DISSERTATION

IMPROVEMENTS IN COMPUTATIONAL ELECTROMAGNETICS SOLVER EFFICIENCY: THEORETICAL AND DATA-DRIVEN APPROACHES TO ACCELERATE FULL-WAVE AND RAY-BASED METHODS

Submitted by

Cam Key

Department of Electrical and Computer Engineering

In partial fulfillment of the requirements

For the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Fall 2020

Doctoral Committee:

Advisor: Branislav Notaroš

Ali Pezeshki
Donald Estep
Milan Ilić

ABSTRACT

IMPROVEMENTS IN COMPUTATIONAL ELECTROMAGNETICS SOLVER EFFICIENCY: THEORETICAL AND DATA-DRIVEN APPROACHES TO ACCELERATE FULL-WAVE AND RAY-BASED METHODS

Simulation plays an ever-increasing role in modern electrical engineering design. However, the computational electromagnetics solvers on which these simulations rely are often inefficient. For simulations requiring high accuracy, full-wave techniques like finite element method and method of moments dominate, yet existing practices for these techniques frequently allocate degrees of freedom sub-optimally, yielding longer solve times than necessary for a given accuracy. For larger-scale simulations, frequency-asymptotic methods like shooting-bouncing ray tracing dominate, yet existing algorithms suffer from incomplete parallelizability and are consequently unable to take full advantage of modern massively parallel computing resources. We present several approaches, both theoretical and empirical, to address these efficiency problems.

ACKNOWLEDGMENTS

I wish to express my sincere appreciation to my advisor, Prof. Branislav Notaroš, and committee members, Profs. Donald Estep, Ali Pezeshki, and Milan Ilić, for their guidance and mentorship during my graduate studies and throughout my time at Colorado State University. I would also like to thank my friends and collaborators from the Colorado State University Electromagnetics Laboratory. Several of the ideas explored in this dissertation would not have come to fruition without the conversations that broke out across the room while code ran. I would like to thank my wife, Cammy, who has patiently supported me through a lot of long nights, furrowed brows, and grumbling about electromagnetics while completing her own graduate degree.

TABLE OF CONTENTS

INTRODUCTION

This dissertation proposes several improvements to the current state of the art in full-wave frequency-domain and frequency-asymptotic computational electromagnetics (CEM) methods. CEM software increasingly relies on hybrid approaches, so we consider it prudent that improvements to the state of the art consider both full-wave and frequency-asymptotic methods. Full-wave frequency domain methods are the backbone of modern CEM for problems at small to moderate electrical scale. Such approaches, like the finite element method (FEM), method of moments (MoM), and finite difference (FD) method, allow simulation of electromagnetic phenomena governed by Maxwell's equations in high detail for highly varying, complicated propagation environments using a discretized differential or integral description of the underlying physics. The choice in discretization is paramount to the efficiency of such methods. The discretization must resolve the relevant behavior of the analytical solution to achieve an accurate numerical solution, meanwhile, the discretization must remain as simple as possible to yield a tractable and efficiently solvable numerical problem. These competing factors are the major compromise of modern full-wave CEM techniques, and how to best reconcile them is an open problem of great interest. Frequency asymptotic methods, in contrast, are most popular for problems at large electrical scale. Such approaches are substantially faster than full-wave methods for large problems, remaining tractable at much larger electrical lengths. Despite this, even the fastest of the frequency-asymptotic methods, shooting-bouncing-ray (SBR) tracing, suffers from diminishing returns with increased parallel computing power, limiting the maximum practical problem size to which frequency-asymptotic methods can be applied.

Chapter one investigates and evaluates applications of the adjoint problem and its solution in frequency-domain CEM. The chapter establishes and validates adjoint-based applications

including higher-order parameter sampling, a posteriori error estimate evaluation, and p- and h-refinement. These applications can improve efficiency, automation, and robustness of CEM methods. We employ a one-dimensional finite-element-method scattering solver, simplifying implementation, replicability, clarity, and intuitiveness of analysis results and conclusions, which then extend naturally to higher-dimensional solvers and more-complicated CEM problems. While demonstrated with a higher-order solver, the derived techniques apply to low-order methodology as well. This is the first demonstration of applicability of adjoint-based a posteriori error estimation techniques to adaptive discretization refinement in frequency-domain CEM with arbitrary-order basis functions. This work introduces application of dual-weighted residual error estimation and selective adaptivity based on error cancellation. The proposed targeted, adaptive mesh/model p- and/or h-refinement heuristics informed by adjoint element-wise error contribution estimates show near-monotonic reduction of quantity-of-interest error with increased number of refined elements, yielding high accuracy solutions efficiently.

In chapter two, we apply some of the concepts from chapter one to fully three-dimensional CEM, giving examples of the adjoint solution and associated elementwise a posteriori error estimates for a three-dimensional FEM scattering problem. To substantiate our claims that adjoint methods from chapter one may lead to efficiency gains for CEM problems, especially those requiring solution of many related sub-problems, chapter two also explores the correlation between a posteriori error estimates for related sub-problems. We use a set of lossy dielectric scatterer FEM problems with identical discretization but varying relative permittivity to demonstrate that elementwise a posteriori error contribution estimates can be more correlated between sub-problems than even quantities of interest and their gradients. Correlation of a posteriori elementwise error contribution estimates between related subproblems is important for eventual

application of adjoint-based a posteriori error estimation to adaptive refinement of multi-solve problems.

In chapter three, we address one of the main and often-ignored limitations to practical application of adjoint methods to refinement in CEM: meshing. We propose a general, robust surface meshing approach based on discrete surface Ricci flow (DSRF) with iterative adaptive refinement in the parametric domain for the automated generation of high-quality surface meshes of arbitrary element type, (low and high) order, and count. Such a widely-applicable surface meshing approach is crucial to realize application of any h- or hp-refinement based adaptive unknown reallocation scheme, such as that proposed in chapter one, to MoM, the most widely-used full wave frequency domain technique. In the proposed method, surfaces are classified by their Euler characteristic and conformally mapped by DSRF to a canonical parametric domain, allowing a canonical seed mesh to be mapped back to an approximation of the original surface. The new DSRF-based meshing technique provides excellent element quality, corner angle uniformity, and local surface-current basis vector orthogonality, aimed to greatly enhance the accuracy, conditioning properties, stability, robustness, and efficiency of surface integral equation CEM solutions. The ability of the proposed DSRF technique to produce high-quality meshes for complicated, highly-varied surfaces is demonstrated for the NASA almond and a fighter jet model, using triangular, quadrilateral, and discontinuous quadrilateral elements. Other element types are also discussed. Where high-fidelity meshing is desired, the technique can near-perfectly capture fine-scale detail using very few high order elements. Where low-fidelity meshing is desired, DSRF with adaptive refinement can accurately recreate course-scale detail using standard first-order elements (e.g., flat triangular patches). This not only enables practical h- and hp-refinement for complicated surfaces, but also allows future surface integral methodologies to take full advantage

of the inherent efficiency of double-higher-order methods, which have previously been limited due to the substantial mesh generation challenges they present for arbitrary problems.

To address CEM solver efficiency for problems out of reach of full-wave approached, we propose and experimentally validate in chapter four a new ray spawning and associated double count removal (DCR) technique for shooting bouncing ray tracing (SBR). This technique allows, for the first time, efficient parallelization of ray DCR, the major bottleneck and least parallel aspect of modern SBR ray-tracing relying on the ray-cone approximation (RCA). We define non-self-adjacent (NSA) ray classes on a recursively sampled icosahedron, guaranteeing removal of mutual adjacency data dependencies between rays that previously prevented efficient parallelization of ray double count removal and, by extension, SBR. Using a GPU-parallelized implementation of the technique, we demonstrate speedups of DCR over 300×, limited in our testing only by the available hardware. As DCR is the asymptotically dominant contributor to the computation time of SBR-RCA, with respect to the number of parallel processes available, the achieved speedup applies to parallel SBR-RCA as a whole. By enabling substantial acceleration of SBR-RCA, the methods proposed in chapter four not only reduce the time cost of solving existing, electrically large CEM problems, but also extend the size threshold beyond which such problems become intractable for fixed parallel computing resources and time.

In chapter five, we return to methods at the shorter end of the electrical length spectrum. We propose and demonstrate a data-driven, machine learning based approach to accelerate the finite element method, method of moments, finite difference method, and related variational methods while maintaining the attractive properties that have allowed such methods to dominate computational science and engineering fields like computational electromagnetics. We use a neural network to predict a set of macro basis functions for a given problem, using only the solution

to an extremely coarse, and therefore computationally cheap, description of the problem as input. We then solve the problem using the predicted macro basis. Unlike some existing methods, ours does not rely on direct prediction of the solution. We show that our macro basis function approach corrects errors in the raw prediction of the network, achieving a far more accurate solution. Results are presented for a class of finite element scattering problems, with error statistics presented from 1000 validation examples and compared to standard and naïve approaches. These results suggest the described macro basis function approach is superior to machine learning approaches that directly predict the solution. Meanwhile our method achieves comparable accuracy to the full solution while requiring only a fraction of the degrees of freedom. We discuss how the value of this fraction translates to efficiency gains and provide direct time comparisons.

Chapter six explores the feasibility of applying the method proposed in chapter five to problems of higher complexity. The simple one-dimensional FEM slab scattering problem from chapter five poorly captures the complexity of problems of interest to CEM researchers and practitioners. To address this, chapter six explores the feasibility of predicting macro basis functions for vastly more complicated problem classes. In addition, we demonstrate the generality of the predicted macro basis function approach by applying it to MoM, rather than FEM, problems in this chapter. We provide error statistics and representative examples for neural networks trained on simple and complicated datasets of MoM scattering problems with highly varying surface features, shapes, and electrical lengths. Notably, we demonstrate that the trained networks learn generalizable knowledge applicable not only to new problems, but entire problem types on which they were not trained. We conclude that the networks produce encouraging results, especially for cross-validation, and larger training datasets will improve reliability for general scattering problems.

1 A POSTERIORI ERROR ESTIMATION AND ADAPTIVE DISCRETIZATION REFINEMENT USING ADJOINT METHODS IN CEM: A STUDY WITH A ONE-DIMENSIONAL HIGHER-ORDER FEM SCATTERING EXAMPLE

1.1 Introduction

Two dimensional In the majority of CEM methods, numerical discretization relies on low-order techniques, for which the structure of interest is modeled by volume or surface elements that are electrically small, and the fields or currents within the elements are approximated by low-order basis functions, often resulting in large linear system size and high computational overhead. Alternatively, higher-order techniques can greatly reduce the number of unknowns for a given problem and enhance the accuracy and efficiency of the CEM analysis, utilizing higher-order basis functions, e.g., sets of linearly independent polynomials, defined over relatively large geometrical elements [1]. This allows for much greater flexibility in adjusting the resolution of the discretization, including h-refinement where the element size is adjusted, p-refinement where the basis function order is adjusted, and hp-refinement which combines both approaches. However, the practical application of that flexibility still presents a significant challenge. Choosing which subset of elements to p- or h-refine to most-optimally improve solution accuracy remains an open challenge with both the higher-order methodology and low-order techniques.

Previous literature on higher-order CEM techniques has focused mainly on solver algorithm efficiency, computation times, and convergence properties with respect to p- or h-refinement in the contexts of both FEM and MoM based numerical discretization procedures, while offering some general heuristics for discretization (mesh or model) building and discretization refinement [1]-[13]. However, increasing demands of uncertainty quantification for complicated engineering simulations [14] necessitate accurate error estimation of computed

results, preferably using approaches that quantify the contributions to error from various discretization choices involved in the simulation.

A practical issue with FEM and MoM CEM techniques in general is the relative inefficiency of gradient-based optimization. Many of the most effective optimization techniques rely on gradient information—in the CEM case, sensitivity of some property of the solution, the Quantity of Interest (QoI), e.g., radar cross-section (RCS), input impedance, etc., to some parameter of the electromagnetic structure in question (scatterer shape, material permittivity, etc.). This sensitivity information is expressed as a partial derivative of the QoI with respect to the input parameter, which is obtained in the classical approach by introducing a small perturbation to the input parameter and recording the corresponding change in the output quantity. This technique requires a minimum of two complete solves—one with the nominal value and the other with the perturbed value of the parameter and is subject to issues of subtractive cancellation that necessitate accurate solves for the differences. To compound this, CEM optimization problems are often multidimensional, with several parameters forming a basis for the search space. So, the full gradient of the QoI over an n¬-dimensional parameter space requires n+1 full solves. For practical CEM problems, the computation time of which can often be measured in hours and sometimes days, this classical approach can be untenable.

As an optimization algorithm explores the search space, the parameters of the CEM problem being solved may vary substantially, a challenge associated with optimization applied to higher-order and low-order FEM and MoM techniques that has broader implications for both traditional (including gradient information) and gradient-free optimization techniques, like genetic algorithms and particle swarm optimization. For optimization problems with large search spaces, a sufficiently refined mesh for all possible parameter combinations within the search bounds is

often extremely fine, slowing the simulation time for each evaluation of the objective function. Rather than attempting to preconstruct a one-size-fits-all mesh, it is often advantageous to begin with a coarse discretization and refine progressively as parameter changes necessitate. This, however, potentially introduces a remeshing step between successive evaluations of the objective function (FEM or MoM solves). Each remeshing can add significant computational overhead to the optimization algorithm and raises issues regarding which elements to refine for maximum benefit and how to refine them.

We often need to improve the accuracy of an existing CEM solution to a given problem by p-, h-, or hp-refining the model adaptively. In adaptive CEM schemes, the solution is automatically refined step by step, according to an error indicator which can be derived from a posteriori error estimates, computed from the existing numerical solution at each step. Ideally, the adaption to reduce the global error in the QoI would be selective and targeted; an element would be selected for refinement based on its a posteriori error contribution estimate, with selected elements subject to a change in field or current approximation order, split into smaller elements, or both. It is therefore highly desirable to produce an automated, adaptive, targeted refinement algorithm that can not only quickly refine the discretization, but can do so near-optimally, choosing the best K elements in the mesh to p- or h-refine for the largest increase in solution accuracy for a given QoI and a given K.

Adjoint methods, employing a generalization of the notion of a Green's function, and a posteriori error estimation techniques have been widely studied in the field of applied mathematics. Influential references include application to ordinary differential equations [15] and in-depth studies concerning specific and general partial differential equations (PDEs) [16]-[20]. We direct an interested reader to [18] for an excellent summary of the methods and to [16] for a thorough

study of adjoint methods as they pertain to adaptive refinement for the numerical solution of differential equations in general. These methods, although well-explored from a theoretical perspective and more frequently applied in other numerical fields like computational fluid dynamics (CFD), have not seen such widespread utilization in CEM for frequency-domain techniques, and have seen very little application toward higher-order CEM techniques. Previous work on the application of adjoint methods to CEM has most often focused on their implementation and application using time-domain methods [21]-[27]. The majority of this work has focused around sensitivity analysis, the calculation of QoI gradients with respect to various input parameters, and often the application of these gradients toward optimization, either directly through gradient-based approaches or indirectly through the construction of surrogate functions [22], [24]-[29]. Most implementations have relied on finite-difference time-domain (FDTD) approaches with various modifications. Sensitivity analysis has been applied to quantify QoI response to material discontinuities [26], optimize transmission line design [27], perform sensitivity analysis for photonic devices [25], optimize antennas [23], [29], and on similar optimization problems using frequency-domain techniques. The paper [28] notably applies adjoint sensitivity analysis to a higher-order two-dimensional triangular-element FEM solver for design optimization of planar microwave devices. The optimization approach in [28] uses adjoint information only for computation of the gradient of a QoI with respect to various parameters, but it does not apply this information to remesh or quantify numerical error.

Previous work in CEM on the quantification of numerical error has focused predominantly on adjoint-free methods quantifying error in the field solution by estimating a norm directly [30]-[33] or indirectly by convergence of this norm [34]. These methods form an a posteriori error estimate from an established norm. Our approach differs substantially by our consideration of

approximate error in a quantity of interest due to the solution error, rather than a norm of a quantity approximating the solution error itself, and the use of the adjoint solution toward this goal. Use of a norm can lead to a rigorous bound to the solution error, but unfortunately, these bounds often overestimate the true error due to local and global cancellation effects. By neglecting the norm, our work exploits cancellation effects for more-accurate estimates and more precisely targeted refinement of the discretization. While examining the solution error can be useful if the application (i.e., our motivation for solving the PDE) is unknown, we are most often interested in one or few quantities derived from the field solution, e.g., radar cross section of a scatterer. In such cases, computing an approximation of the QoI error from an existing field error estimate is often less accurate than approximating the QoI in the error directly [14], [16]. Refinement based on the former, established approach, therefore tends to oversaturate the discretization, refining more elements than necessary for the given QoI when compared to the latter approach explored in this chapter. Most closely related to the a posteriori error estimation in the present work is that of Monk and Suli [35], [36], in which the adjoint is used to derive a posteriori error bounds for the far-field pattern, with the error estimate then applied to refining the discretization. However, unlike the present work, these papers produce a highly discretization-dependent error estimate specifically for the far-field pattern, limiting its applicability to first-order (linear) finite element approaches with far-field QoI. Meanwhile, the estimate given in this present chapter is presented for a general QoI (with backscattered field given as an example QoI) and is broadly applicable to FEM solvers of arbitrary basis function order. This allows application to h-refinement, similar to [36], and rapidly-convergent p-refinement and hp-refinement. In three-dimensional cases, the error estimates defined in this chapter are straightforwardly extensible to elements of arbitrary geometric shapes, while the estimate given in [35] is defined specifically for tetrahedra. Unlike [30]-[36], the

present work also gives in-depth examples of the broadly applicable sensitivity information that can be attained inexpensively where the adjoint problem is solved (for instance for error estimation) and a discussion of its applications.

In this chapter, we investigate useful applications of the adjoint problem and its solution toward frequency-domain CEM methods. We demonstrate how QoIs can be expressed in inner product form and show how this form can efficiently generate gradient information for a given QoI with any number of parameters using the adjoint solution. We give two useful examples of such parameters using the backscattered field amplitude as the QoI. A one-dimensional higher-order FEM scattering solver is chosen as an ideal testbed to investigate the usefulness of the proposed techniques due to its conceptual and computational simplicity, ease of implementation and replicability, and the clarity with which results from a one-dimensional model can be displayed. Namely, it is advantageous to represent much of the information obtained by adjoint methods as a scalar field over the computational domain, which facilitates displaying data in a useful and intuitive manner and enables straightforward qualitative and quantitative conclusions of the analyses. Results and observations from this model extend naturally to higher-dimensional solvers and more-complicated CEM techniques and problems. Describing how the gradient information can be used to produce a reconstruction of a QoI's response to varying parameters, we invoke the higher-order parameter sampling (HOPS) technique [37] to produce these reconstructions with applications to the example problem. We highlight how such gradient information and response reconstructions can be applied to practical CEM problems requiring many solves, for instance, RCS computation, antenna design, optimization, and Monte Carlo simulation. We introduce a posteriori error estimation techniques using the adjoint solution [38], and apply these error estimates to novel targeted p- and h-refinement schemes. To the best of our

knowledge, this is the first demonstration of the applicability of both adjoint-based a posteriori error estimation and adaptive discretization refinement in frequency-domain CEM using arbitrary-order basis functions. In addition, this chapter introduces to CEM the application of a dual-weighted residual (DWR) estimate to the adjoint-informed a posteriori error estimation, the selective adaptivity based on error cancellation, and p-refinement using the adjoint solution. The adjoint-based DWR technique for CEM proposed in this work produces an accurate, signed error estimate, which is exploited to cancel local error contributions by grouping. This results in rapid reduction in global QoI error with a high selectivity not possible using existing norm-based error estimates in CEM that seek to rigorously (or approximately) bound error in a norm.

For a useful and broadly applicable means of evaluating the performance of different refinement approaches, we introduce a metric based on the degree of monotonicity of a given refinement to quantify its efficacy in comparison with other approaches. Using the same example scattering problem, we propose targeted, adaptive discretization (mesh or model) refinement heuristics informed by adjoint element-wise error contribution estimates. These heuristics perform exceptionally well, greatly reducing error in a QoI for only modest increases in the number of unknowns, while also near-monotonically reducing error with respect to an increasing number of refined elements. The results demonstrate the benefits that adjoint techniques offer for adaptive p- and h-refinement schemes using these heuristics. Although demonstrated with a higher-order solver, all derived and applied techniques generalize to low-order methodology, and the results in this chapter show the usefulness and efficiency of the proposed techniques to low-order methods with h-refinement only. A goal of this work is also to promote the use of adjoint approaches within future CEM techniques and implementations as a means of attaining useful refinement,

optimization, and uncertainty quantification methodologies. Some preliminaries of this chapter are presented in a summary form in [39] and [40].

1.2 One-Dimensional FEM Scattering Problem

We consider a simple electromagnetic scattering problem—the infinite lossy dielectric slab scatterer in a one-dimensional domain—so that the underlying physics, solutions, and numerical method parameters are straightforward to describe. We define the model domain and material subdomains for an infinite (in *y* and *z*) dielectric slab with air and a perfectly matched layer (PML) domain on either side as specified in Table 1.1.

TABLE 1.1. Model domain and material subdomains for scattering from the infinite (in *y* and *z*) lossy dielectric slab truncated by PML.

$-t_{\text{PML}} < x < 0$	Left PML subdomain (t_{PML} is the selected PML thickness)
$0 < x < \lambda$	Left air subdomain (of thickness λ)
$\lambda < x < a$	Lossy dielectric slab subdomain (of thickness $a - \lambda$)
$a < x < L$	Right air subdomain (of thickness $L - a$)
$L < x < L + t_{\text{PML}}$	Right PML subdomain (of thickness t_{PML})

On this domain, we use the double-curl frequency-domain wave equation [1] and the associated boundary condition,

$$\nabla \times \mu_r^{-1} \nabla \times \boldsymbol{E}^{\text{sc}} - k_0^2 \varepsilon_r \boldsymbol{E}^{\text{sc}} = -\nabla \times \mu_r^{-1} \nabla \times \boldsymbol{E}^{\text{inc}} + k_0^2 \varepsilon_r \boldsymbol{E}^{\text{inc}},$$

$$-t_{\text{PML}} < x < L + t_{\text{PML}} \tag{1.1a}$$

$$\boldsymbol{n} \times \boldsymbol{E}^{\text{sc}} = 0,\ x = -t_{\text{PML}},\ x = L + t_{\text{PML}} \tag{1.1b}$$

where ε_r and μ_r are complex relative permittivity and permeability of the inhomogeneous medium (tensors for anisotropic materials), $\mathbf{E}^{\text{inc}}$ and $\mathbf{E}^{\text{sc}}$ are the incident and scattered electric field complex

intensity vectors, $k_0 = \omega\sqrt{\varepsilon_0\mu_0}$ is the free-space wave number, ω is the angular frequency of the implied time-harmonic variation, and **n** is the outward unit normal on the boundary surface. Enforcing homogeneity in the *y* and *z* directions and restricting the incident field to have only a *z* component that depends only on *x*, we simplify (1.1) to

$$-\frac{d}{dx}\frac{1}{\mu_r(x)}\frac{d}{dx}E_z^{\text{sc}}(x) - k_0^2\varepsilon_r(x)E_z^{\text{sc}}(x) = g(x), \quad -t_{\text{PML}} < x < L + t_{\text{PML}} \tag{1.2a}$$

$$g(x) = \frac{d}{dx}\left(\frac{1}{\mu_r(x)} - 1\right)\frac{d}{dx}E_z^{\text{inc}}(x) + k_0^2(\varepsilon_r(x) - 1)E_z^{\text{inc}}(x), \quad -t_{\text{PML}} < x < L + t_{\text{PML}} \tag{1.2b}$$

$$E_z^{\text{sc}}(x) = 0,\ x = -t_{\text{PML}},\ x = L + t_{\text{PML}} \tag{1.3}$$

With the incident field representing a *z*-polarized plane wave propagating forward along the *x*-axis and the standard PML implementation, we have

$$E_z^{\text{inc}}(x) = \begin{cases} Ae^{-jk_0x} & 0 < x < L \\ 0 & -t_{\text{PML}} < x < 0,\quad L < x < L + t_{\text{PML}} \end{cases} \tag{1.4}$$

where we choose $A = 1$ to normalize the field. The material parameter functions in the subdomains defined in Table 3.1.1 are given by:

$$\varepsilon_r(x) = \begin{cases} 1 & 0 < x < \lambda \\ \varepsilon_d & \lambda < x < a \\ 1 & a < x < L \\ 1 - j\alpha_{\text{PML}} & -t_{\text{PML}} < x < 0,\ L < x < L + t_{\text{PML}} \end{cases} \tag{1.5a}$$

$$\mu_r(x) = \begin{cases} 1 & 0 < x < L \\ 1 - j\alpha_{\text{PML}} & -t_{\text{PML}} < x < 0,\ L < x < L + t_{\text{PML}} \end{cases} \tag{1.5b}$$

with ε_d denoting the equivalent complex relative permittivity of the lossy dielectric slab of relative permittivity ε_r and conductivity σ [41],

$$\varepsilon_d = \varepsilon_r - j\frac{\sigma}{\omega\varepsilon_0} \tag{1.6}$$

The lossy dielectric slab scattering problem is solved using a higher-order PML-truncated FEM approach similar to that described in [10], [11], [42], and [43]. The domain is discretized

using line segments along the *x*-axis with scalar basis functions. This geometric simplicity allows for simple *h*-refinement (e.g., an element can be split in half just by adding a new element boundary node at its midpoint).

Like their three-dimensional counterparts in [42], the basis functions used for the one-dimensional solver are defined in a domain parameterized by a single coordinate *s* which ranges from −1 to 1. The element-specific index of the chosen basis function is given by *i*, and the field expansion order for a given element is denoted *M*. Note that the higher-order approach outlined in [42] allows for arbitrary *x*-domain sizes and arbitrary, positive field expansion orders for adjacent elements. This allows adjacent elements to be *h*- and *p*-refined to differing degrees entirely independently of each other. The *i*th basis function for an element is given in the *s* domain as:

$$u_i(s) = \begin{cases} 0.5(1 + (-1)^i s) & i = 0,1 \\ 0.5(1 - s)(1 + s)^{i-1} & 2 \leq i \leq M \end{cases} \tag{1.7}$$

The first and second basis functions maintain field continuity between adjacent elements, while the higher-order basis functions allow for additional *p*-refinement. Note that functions in (1.7) are just one simple choice of higher-order scalar bases on the *s*-parametric domain, and alternative hierarchical polynomial basis functions with improved orthogonality and conditioning properties could also be chosen. For example, a one-dimensional variant of those used in the higher-order FEM-PML method [43] may be easily implemented.

1.3 The Adjoint Problem and the Quantity of Interest

The notion of an adjoint problem generalizes the method of Green's functions [38], [44], [45]. To define the adjoint operator for a given problem, we must cast the problem in linear operator form. For the lossy dielectric slab scattering problem, we consider the Dirichlet boundary

value problem given in (1.2)-(1.3). The differential equation in (1.2a) can be expressed in linear operator form as

$$L\boldsymbol{E}_z^{\mathrm{sc}} = \boldsymbol{g} \tag{1.8}$$

L represents the "forward" operator the forward solution (the scattered electric field), designated $\boldsymbol{E}_z^{\mathrm{sc}}$. The adjoint operator of L is the operator L^{adj} that satisfies the Lagrange identity [38],

$$\left\langle \boldsymbol{E}^{\mathrm{adj}}, L\boldsymbol{E}_z^{\mathrm{sc}} \right\rangle = \left\langle L^{\mathrm{adj}}\boldsymbol{E}^{\mathrm{adj}}, \boldsymbol{E}_z^{\mathrm{sc}} \right\rangle \tag{1.9}$$

with angle brackets denoting the L^2 inner product on functions. The data for the adjoint problem is, in this case, a QoI determined by a linear functional on the forward solution. In inner product form, the QoI is given as

$$\mathrm{QoI} = q[\mathbf{E}_z^{\mathrm{sc}}] = \langle \boldsymbol{E}_z^{\mathrm{sc}}, \boldsymbol{p} \rangle \tag{1.10}$$

where **p** is a function that determines an instrumental or measurement characteristic. For instance, the QoI could be chosen as the field value at a given point in the domain by defining **p** as a Dirac delta function at that point, in which case the adjoint solution is the Green's function [45]. The sampling property of the Dirac delta function when the inner product is applied then evaluates the field at one point. For a given measurement characteristic, the adjoint problem is

$$L^{adj}\boldsymbol{E}^{\mathrm{adj}} = \mathbf{p} \tag{1.11}$$

Note that, in a physical interpretation of (11), the measurement characteristic defining some QoI on the forward solution becomes the excitation of the adjoint problem. As **p** defines a unique QoI, (11) implies that the adjoint equation must be re-solved with a new right-hand-side for each new QoI, analogous to re-solving a forward problem with new incident fields.

We choose a QoI that yields the magnitude of the reflected field from the lossy dielectric slab subject to some incident field. We express the solution in the air-filled region $0 < x < \lambda$ as

$$E_z^{\mathrm{sc}}(x) = E_{i,z}e^{-jk_0x} + E_{r,z}e^{jk_0x},\ 0 < x < \lambda \tag{1.12}$$

where $E_{i,z}$ and $E_{r,z}$ are (complex-valued) numbers characterizing the forward and backward traveling electromagnetic fields, respectively. Since $A = 1$ in (1.4) and the zero-phase point of the reflected field is $x = 0$, $E_{r,z}$ is equal to the complex reflection coefficient. We express the amplitude of the reflected field as a QoI in inner product form (1.10) as

$$q[\boldsymbol{E}_z^{\mathrm{sc}}] = \frac{k_0}{2\pi}\int_0^{2\pi/k_0} e^{-jk_0x}\boldsymbol{E}_z^{\mathrm{sc}}dx$$

$$= \left\langle \boldsymbol{E}_z^{\mathrm{sc}}, \frac{k_0}{2\pi}e^{jk_0x}\left[H(x) - H(x-\frac{2\pi}{k_0})\right]\right\rangle = \langle \boldsymbol{E}_z^{\mathrm{sc}}, \boldsymbol{p}\rangle \tag{1.13}$$

where $H(x)$ is the unit step (Heaviside) function and **p** denotes the defined measurement characteristic. The behavior of this QoI can be evaluated from (1.13) on a function of the form (1.12) yielding

$$q[E_{i,z}e^{-jk_0x} + E_{r,z}e^{jk_0x}] = E_{r,z} \tag{1.14}$$

Note that in (1.14), it is assumed that the surface of the dielectric slab is outside the limits of integration. The idea behind this type of functional evaluation is easily extended to a higher-dimensional scattering problem—different components of the spatial Fourier transform of a scattered electromagnetic wave along a closed surface in free space effectively gives the scattered electromagnetic field in different far-field directions.

To derive the adjoint operator for (1.2a), we apply the Lagrange identity in (1.9), where the left-hand side of (1.9) may be expressed as

$$\int_{-t_{\mathrm{PML}}}^{L+t_{\mathrm{PML}}} E^{\mathrm{adj}^*}(x)\left(-\frac{d}{dx}\frac{1}{\mu_r(x)}\frac{d}{dx}E_z^{\mathrm{sc}}(x)\right)dx -$$

$$\int_{-t_{\mathrm{PML}}}^{L+t_{\mathrm{PML}}} E^{\mathrm{adj}^*}(x)(k_0^2\varepsilon_r(x)E_z^{\mathrm{sc}}(x))dx \tag{1.15}$$

with E^{adj} denoting the unknown adjoint solution. Integrating the first term in (1.15) by parts, it becomes

$$-E^{\text{adj}^*}(x)\frac{1}{\mu_r(x)}\frac{d}{dx}E_z^{\text{sc}}(x)\Big|_{-t_{\text{PML}}}^{L+t_{\text{PML}}}+$$

$$\int_{-t_{\text{PML}}}^{L+t_{\text{PML}}}\frac{1}{\mu_r(x)}\frac{d}{dx}E_z^{\text{sc}}(x)\frac{d}{dx}E^{\text{adj}^*}(x)dx \tag{1.16}$$

Then integration of the second term of (1.16) by parts results in

$$E_z^{\text{sc}}(x)\frac{1}{\mu_r(x)}\frac{d}{dx}E^{\text{adj}^*}(x)\Big|_{-t_{\text{PML}}}^{L+t_{\text{PML}}}-$$

$$\int_{-t_{\text{PML}}}^{L+t_{\text{PML}}}E_z^{\text{sc}}(x)\frac{d}{dx}\frac{1}{\mu_r(x)}\frac{d}{dx}E^{\text{adj}^*}(x)dx \tag{1.17}$$

Applying the forward boundary conditions from (1.3), we cancel the first term of (1.17). By similarly cancelling the first term in (1.16), we enforce the adjoint boundary conditions,

$$E^{\text{adj}}(L+t_{\text{PML}})=0$$

$$E^{\text{adj}}(-t_{\text{PML}})=0 \tag{1.18}$$

The FEM approach studied in this chapter utilizes in general a PML terminated in a perfect electric conductor (PEC) to truncate the computational domain. As such, this treatment of the first term in (1.16) and (1.17) is universally applicable for this method and is analogously true in two and three dimensions. We next rewrite (1.15) by rearranging its second term and applying the results of integration by parts, yielding

$$-\int_{-t_{\text{PML}}}^{L+t_{\text{PML}}}E_z^{\text{sc}}(x)\frac{d}{dx}\frac{1}{\mu_r(x)}\frac{d}{dx}E^{\text{adj}^*}(x)dx-$$

$$\int_{-t_{\text{PML}}}^{L+t_{\text{PML}}}E_z^{\text{sc}}(x)k_0^2\varepsilon_r(x)E^{\text{adj}^*}(x)dx \tag{1.19}$$

from which we recover the form of the adjoint operator on the right-hand side of (1.9) by taking the complex conjugate,

$$L^{adj}E^{\text{adj}}=-\frac{d}{dx}\frac{1}{\mu_r^*(x)}\frac{d}{dx}E^{\text{adj}}(x)-k_0^2\varepsilon_r^*(x)E^{\text{adj}}(x)=p(x) \tag{1.20}$$

Note the similarity of the adjoint equation (1.20) to the forward equation (1.2a). We see that the one-dimensional analogue of the double-curl wave equation (1.1a) is nearly self-adjoint, its adjoint being described entirely by complex conjugation of the material parameters.

1.4 A Posteriori Error Estimation and Refinement

Accurate estimation of error in computational simulation results is a key component in uncertainty quantification [14]. Additionally, adaptive *p*- and *h*-refinement schemes require indicators of the error in a QoI on which the adaption is done, namely, new field/current approximation orders and/or new element sizes are assigned in the new, refined model. Furthermore, in CEM problems requiring many computational simulations, it is useful to have an estimate of whether accuracy of a forward solution for a given parameter set and discretization is sufficient for the desired tolerance, or if refinement is needed for subsequent solves. Such problems, including optimization, antenna design, and radar cross-section determination, are common in CEM. An adjoint-based a posteriori error estimate can address each of these needs.

Due to the Galerkin orthogonality property [45], computing this error estimate essentially involves evaluating a numerical approximation to derivatives of the adjoint solution, and therefore requires that the adjoint problem be solved on a discretization different than that used for the forward solution, for instance, using either finer geometric elements or higher-order basis functions. In general, the adjoint discretization need not present more unknowns than the forward discretization, but for more-accurate estimates it is desirable to compute the adjoint solution using a finer discretization. The use of hierarchical basis functions in this work makes the calculation of many of the required degrees of freedom simpler.

We express the numerical solution of the forward problem on a given mesh as

$$\boldsymbol{E}_z^{\mathrm{sc}} \approx \widetilde{\boldsymbol{E}}_z^{\mathrm{sc}} = \sum_{u_i \in \boldsymbol{V}_b} \alpha_i u_i(x), \tag{1.21}$$

where $\mathbf{V}_\mathrm{b}$ is the space of basis functions for the forward solution. Additionally, we express the numerical solution to the adjoint equation by

$$\boldsymbol{E}^{\mathrm{adj}} \approx \widetilde{\boldsymbol{E}}^{\mathrm{adj}} = \sum_{u_i \in \boldsymbol{V}_{\mathrm{b}'}} \beta_i u_i(x), \tag{1.22}$$

with $\mathbf{V}_{\mathrm{b}'}$ designating the space of basis functions for the adjoint problem, where the M in (1.7) for each element has been increased by 1 from $\mathbf{V}_\mathrm{b}$. Also, we let $\pi_h \widetilde{\boldsymbol{E}}^{\mathrm{adj}}$denote a projection or interpolant of the adjoint solution into the discrete space $\mathbf{V}_\mathrm{b}$ in which we numerically solve the forward problem. In this work, $\pi_h \widetilde{\boldsymbol{E}}^{\mathrm{adj}}$is defined by a least squares approximation of $\widetilde{\boldsymbol{E}}^{\mathrm{adj}}$in the $\mathbf{V}_\mathrm{b}$ subspace of $\mathbf{V}_{\mathrm{b}'}$. Following the arguments expressing the QoI in terms of the adjoint solution above, the a posteriori estimate on the error in the QoI is

$$\begin{aligned}\left\langle \boldsymbol{E}_z^{\mathrm{sc}} - \widetilde{\boldsymbol{E}}_z^{\mathrm{sc}}, \boldsymbol{p} \right\rangle \approx & \left\langle g(x), \widetilde{\boldsymbol{E}}^{\mathrm{adj}} - \pi_h \widetilde{\boldsymbol{E}}^{\mathrm{adj}} \right\rangle \\ & - \left\langle \frac{1}{\mu_r(x)} \frac{d}{dx} \widetilde{\boldsymbol{E}}_z^{\mathrm{sc}}, \frac{d}{dx} \widetilde{\boldsymbol{E}}^{\mathrm{adj}} - \frac{d}{dx} \pi_h \widetilde{\boldsymbol{E}}^{\mathrm{adj}} \right\rangle + k_0^2 \left\langle \varepsilon_r(x) \widetilde{\boldsymbol{E}}_z^{\mathrm{sc}}, \widetilde{\boldsymbol{E}}^{\mathrm{adj}} - \pi_h \widetilde{\boldsymbol{E}}^{\mathrm{adj}} \right\rangle \end{aligned} \tag{1.23}$$

Essentially, (1.23) represents the inner product (in weak form) of the residual of $\widetilde{\boldsymbol{E}}_{\boldsymbol{z}}^{\mathbf{sc}}$ and a weight determined by the adjoint solution $\widetilde{\boldsymbol{E}}^{\mathrm{adj}}$, so it is also called a dual-weighted residual (DWR) estimate. The residual quantifies how well the numerical solution solves the differential equation while the adjoint weight quantifies how the local residual affects the global error [14].

As the adjoint-based a posteriori error estimate requires an additional numerical solve on a finer discretization, it may seem counterintuitive to spend this on an adjoint solve. We recall Richardson extrapolation suggests the classical approach to obtain an error estimate on the accuracy of a forward solution on a given discretization, which is obtained by subtracting the forward solution from a more accurate forward solution obtained from a refined discretization. Because the classic estimate is on the error of the solution rather than a QoI, the level of refinement

needed for reliably accurate estimates using the classical approach is generally higher than needed to compute an accurate a posteriori error estimate [14]. Moreover, the classical approach does not yield an estimate on a QoI that distinguishes residuals determined by local discretization choices and the effects of stability as determined by the adjoint solution, hence the classical estimate is less useful for adaptive discretization [14]. Finally, once obtained, the adjoint solution can be used for other purposes, e.g., optimization and sensitivity analysis.

The standard FEM implementation computes integrals through the domain element-by-element; the inner product integrals in (23) are evaluated in a similar manner, with integrals first computed over each element separately and then summed to obtain the final error estimate. The information at the intermediate step before summation is immensely useful toward remeshing and determining which locations in the mesh are most in need of refinement. This information is referred to as the element-wise error contribution estimate and can be represented as a vector of error contribution estimates from each element

$$\boldsymbol{e} = (e_1, e_2, \ldots, e_N) \tag{1.24}$$

where e_i denotes the error contribution of the ith element and N is the total number of elements. The sum of (3.1.24) then gives the total QoI error estimate.

1.5 Utilizing Gradient Information from the Adjoint Solution

A classical first-order finite difference approach to compute the gradient of a QoI with respect to P independent parameters requires a minimum P+1 solves of the forward problem. Using adjoint methods, the same gradient information is obtained using a single adjoint solve over a finer discretization as expressed in (1.22), requiring only an expression of the partial derivative of the operator with respect to each parameter as shown below. From [38], we use a Taylor

expansion of the QoI, represented here in inner product form as in (1.10), at an arbitrary parameter value, r, in terms of a known value of that QoI for a nominal parameter value, r_0,

$$\langle \boldsymbol{E}_z^{\mathrm{sc}}, \boldsymbol{p} \rangle = \left\langle \boldsymbol{E}_{z,0}^{\mathrm{sc}}, \boldsymbol{p} \right\rangle + \left\langle L D_r \boldsymbol{E}_{z,0}^{\mathrm{sc}} (r - r_0), \boldsymbol{E}^{\mathrm{adj}} \right\rangle + \left\langle \boldsymbol{R}, \boldsymbol{E}^{\mathrm{adj}} \right\rangle \tag{1.25}$$

where $\boldsymbol{E}_{z,0}^{\mathrm{sc}}$represents the forward solution at r_0, D_r denotes the Frechet derivative operator, in this case with respect to r, and **R** is a higher-order remainder term.

Neglecting the remainder term, we obtain a linear approximation for the QoI around the nominal parameter value using the higher-order parameter sampling (HOPS) method [37],

$$\langle \boldsymbol{E}_z^{\mathrm{sc}}, \boldsymbol{p} \rangle \approx \left\langle \boldsymbol{E}_{z,0}^{\mathrm{sc}}, \boldsymbol{p} \right\rangle + \left\langle L D_r \boldsymbol{E}_{z,0}^{\mathrm{sc}} (r - r_0), \boldsymbol{E}^{\mathrm{adj}} \right\rangle \tag{1.26}$$

with the partial derivative of the QoI with respect to the chosen parameter near the nominal parameter value given by

$$\frac{\partial q}{\partial r}(r_0) = \left\langle L D_r \boldsymbol{E}_{z,0}^{\mathrm{sc}}, \boldsymbol{E}^{\mathrm{adj}} \right\rangle \tag{1.27}$$

The gradient of a QoI with respect to multiple parameters, each around a nominal value, can then be formed by a vector of partial derivatives of form (1.27), requiring only the evaluation of inner products with an adjoint solution, rather than numerous additional perturbed solutions of the forward problem.

To briefly demonstrate where (1.27) comes from and how the $L D_r \boldsymbol{E}_0^{\mathrm{sc}}$ term may be evaluated, we begin by noting

$$D_r q = D_r \langle \boldsymbol{E}_z^{\mathrm{sc}}, \boldsymbol{p} \rangle = \langle D_r \boldsymbol{E}_z^{\mathrm{sc}}, \boldsymbol{p} \rangle \tag{1.28}$$

Applying (1.9) and (1.11), we then get

$$D_r q = \left\langle L D_r \boldsymbol{E}_z^{\mathrm{sc}}, \boldsymbol{E}^{\mathrm{adj}} \right\rangle \tag{1.29}$$

Applying the Frechet derivative operator to both sides of (1.8), we can write

$$D_r [L \boldsymbol{E}_z^{\mathrm{sc}} = \boldsymbol{g}] = [D_r L \boldsymbol{E}_z^{\mathrm{sc}} + L D_r \boldsymbol{E}_z^{\mathrm{sc}} = D_r \boldsymbol{g}]$$

$$L D_r \boldsymbol{E}_z^{\mathrm{sc}} = D_r \boldsymbol{g} - D_r L \boldsymbol{E}_z^{\mathrm{sc}} \tag{1.30}$$

1.6 Numerical Results and Discussion

We first present results for HOPS applied to the lossy dielectric slab scattering problem with the described reflected field amplitude QoI. Implementing HOPS for two parameters, the imaginary part of the slab relative permittivity and the left slab-air interface location, we show the efficacy of this technique in obtaining derivative information and approximate reconstructions of QoI response to changing parameters from only a few sample points. Separating the slab complex relative permittivity into real and imaginary parts,

$$\varepsilon_r(x) = \alpha + \mathrm{j}\beta, \quad \lambda < x < a \tag{1.31}$$

we choose β as the HOPS parameter with nominal value β_0. The linear approximation of the QoI around β_0 can then be expressed in the form of (1.26) as

$$q(\beta) \approx q(\beta_0) + \left\langle [-D_\beta(-k_0^2 \varepsilon_r(x) \boldsymbol{E}_{z,0}^{\mathrm{sc}}) + D_\beta(k_0^2(\varepsilon_r(x) - 1)\boldsymbol{E}_z^{\mathrm{inc}})], \boldsymbol{E}_0^{\mathrm{adj}} \right\rangle (\beta - \beta_0) \tag{1.32}$$

which, writing the inner products in integral form, is given by

$$q(\beta) \approx q(\beta_0) + (\beta - \beta_0) j k_0^2 \int_\lambda^a (E_{z,0}^{\mathrm{sc}}(x) + E_z^{\mathrm{inc}}(x)) E_0^{\mathrm{adj}*}(x) dx \tag{1.33}$$

Note that all integrals over elements within the slab required for evaluation of (1.33) are calculated during assembly of the stiffness matrix for the FEM, provided each element in the integration domain has a homogeneous permittivity value. The HOPS technique is applied to a set of 5 nominal parameter values to reconstruct the response of the QoI to β. Results generated using first-order forward solves and second-order adjoint solves with *h*-uniform elements are overlaid in Fig. 1.1 with the QoI response over the same parameter range obtained by analytical solution of (1.2)-(1.3). Each of the five lines obtained by (1.33) at the five sample points is truncated at

intersections with its left and right neighbors to produce a piecewise-linear reconstruction of the QoI response.

The five-point HOPS results in Fig. 1.1 agree closely with the analytical solution, both in the real component and imaginary component, despite a large parameter domain and low number of sample points. A piece-wise linear approximation of a QoI response in this form has many useful applications. For instance, such an approximation could be used as an inexpensive surrogate function for optimization, requiring fewer expensive direct evaluations of the QoI response by forward solves. The approach in (1.33) can be easily extended to variations in other material parameters and higher-dimensional problems, requiring only an expression for $LD_r\boldsymbol{E}_0^{\mathrm{sc}}$ from the chosen problem.

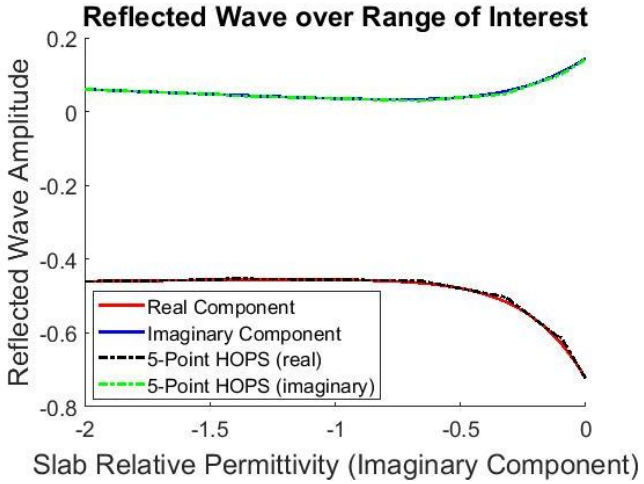


Fig. 1.1. Higher-order parameter sampling using five sample points to produce a linear reconstruction of the response of the reflected field amplitude to the imaginary component of the slab relative permittivity. Results are generated using first-order forward solves and second-order adjoint solves. The domain length is $L = 4$ m, the slab left boundary location is $\lambda = 1$ m, the frequency is $f = 600$ MHz, and the real part of the slab relative permittivity is $\alpha = 7$. Elements are h-uniform with size 0.02 m. Samples were taken at $\beta = [-2.0, -1.0, -0.5, -.025, 0]$.

It is often of great interest in practical CEM problems to determine effects of the location of a material interface on some QoI, for instance the response of the RCS in a given direction to the shape of a scatterer. We give a one-dimensional analogue of this problem here, choosing the

analogous QoI from (1.13) and approximating its response to changes in the left slab interface location λ. The permittivity function in the subdomain $0 < x < L$ can be represented in terms of unit step functions as

$$\varepsilon_r(x) = 1 + (\varepsilon_d - 1)H(x - \lambda_0) - (\varepsilon_d - 1)H(x - a) \tag{1.34}$$

We may approximatively express $q(\lambda)$, again using the HOPS method from (1.26), as

$$q(\lambda) \approx q(\lambda_0) +$$

$$k_0^2\left\langle[-D_\lambda(\varepsilon_r(x)\boldsymbol{E}_{z,0}^{\mathrm{sc}}) + D_\lambda((\varepsilon_r(x) - 1)\boldsymbol{E}_z^{\mathrm{inc}})], \boldsymbol{E}_0^{\mathrm{adj}}\right\rangle(\lambda - \lambda_0) \tag{1.35}$$

which requires the calculation of the derivative of the permittivity function (1.5a) with respect to λ for direct implementation. The Frechet derivative of the permittivity function can be written in terms of the Dirac delta function as

$$D_\lambda \varepsilon_r(x) = -(\varepsilon_d - 1)\delta(x - \lambda_0) \tag{1.36}$$

which, after evaluating the inner product using the sampling property of the Dirac delta, gives a form of (1.35) that can be evaluated directly:

$$q(\lambda) \approx q(\lambda_0) - k_0^2[E_0^{adj*}(x = \lambda_0)(\varepsilon_d - 1)$$

$$(E_{z,0}^{\mathrm{sc}}(x = \lambda_0) + E_z^{\mathrm{inc}}(x = \lambda_0))](\lambda - \lambda_0) \tag{1.37}$$

Note that here we have assumed a piecewise constant permittivity function (1.34). Were the permittivity function instead smooth and continuous, evaluating the corresponding analogue of (1.35) becomes simpler, requiring no use of the sampling property of the Dirac delta as in (1.37). In fact, we only require that a function describing the material renders both sides of (1.8) Fréchet differentiable with respect to the chosen parameter.

Like Fig. 1.1, Fig. 1.2 shows results of a five-sample HOPS reconstruction of the QoI response, this time with respect to the *x*-coordinate of the left slab face. Results are again generated using first-order forward solves, second-order adjoint solves, and *h*-uniform elements. We again

see excellent agreement between the HOPS result and the analytical QoI response over the parameter domain.

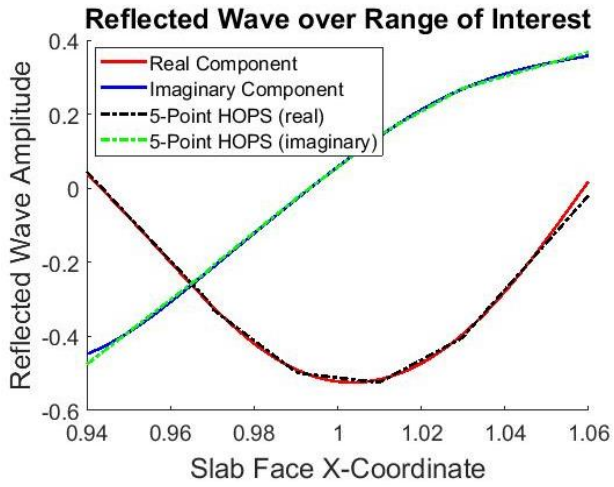


Fig. 1.2. Higher-order parameter sampling using five sample points to produce a linear reconstruction of the response of the reflected field amplitude to the location of the left slab face, using first-order forward solves and second-order adjoint solves, with $L = 4$ m, $\lambda = 1$ m, $f = 600$ MHz, slab relative permittivity $\varepsilon_r = 7 - j1.8$, and h-uniform elements of size 0.02 m. Samples were taken at $\lambda_0 = [0.96, 0.98, 1.0, 1.02, 1.04]$.

We next show the application of the adjoint method to obtain **e**, the element-wise error contribution estimate, for a given problem. Maintaining most parameter values from solves used in the HOPS results and coarsening the element size to 0.05 m, we compute **e** through a partial (un-summed) evaluation of (1.23) as previously described. Fig. 1.3 shows the real and imaginary components of **e** plotted throughout the different material subdomains, with e_i for the ith element plotted at the x-coordinate of the element's midpoint.

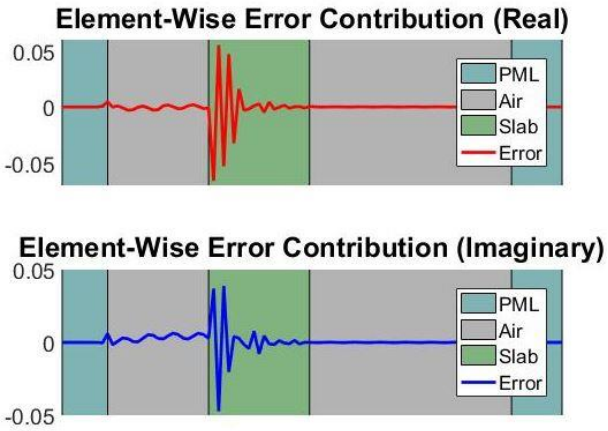


Fig. 1.3. Real and imaginary element-wise error contribution estimates over the computational domain. Results are generated using first-order forward solves and second-order adjoint solves, with $L = 4$ m, $\lambda = 1$ m, $f = 600$ MHz, $\varepsilon_r = 7 - j1.8$, and h-uniform elements of size 0.05 m.

We note several crucial things from the element-wise error contribution in Fig. 1.3. Most importantly, the real and imaginary error contributions are oscillatory, varying from positive to negative values through the domain. As the total error estimate is obtained by the sum of these individual error contribution estimates, we can conclude that there is a high degree of cancelation between error contributions throughout the domain. Clearly this cancelation is not complete, or the error estimate given in (1.23) would be zero. An interesting question then becomes where in the computational domain this lack of cancelation occurs—which elements are contributing the most to the total error estimate. This is a deceptively ill-posed problem, however, see [45]. We cannot directly assign the bulk of the residual error to any one set of elements. To do so suggests some necessary ordering to **e**.

It is difficult if not impossible to apply a universally-applicable and consistent method to identify which elements contribute the most to the total error estimate that relies on the ordering of elements in the spatial domain or the order in which error contributions are summed [45]. No ordering actually exists for the summation of **e**—the error contribution of a given element can be said to cancel with that of any other element or combination of elements, no matter how distant

the elements are in the computational domain. To accurately apply this element-wise error contribution estimate toward targeted, adaptive mesh or model refinement, we must define a means by which to identify a "bad" element that relies on no such ordering.

We show here the application of two un-ordered refinement heuristics and their applications to *h*- and *p*-refinement on a variety of evaluations of error in the QoI for a three-dimensional parameter space consisting of λ, α, and β. The different locations in this search space at which the QoI error estimate is evaluated are given in Table 1.2.

TABLE 1.2. Parameter space locations at which QoI error estimate is evaluated for refinement results.

Plot Identifier	α	β	λ (m)
Case A	7	−1.8	1
Case B	7	−1.8	1.2
Case C	7	−1.8	0.8
Case D	7	−4	1
Case E	7	−4	1.2
Case F	7	−4	0.8
Case G	3	−1.8	1
Case H	3	−1.8	1.2
Case I	3	−1.8	0.8
Case J	3	−4	1
Case K	3	−4	1.2
Case L	3	−4	0.8

A refinement heuristic can be stated as one by which elements in the domain are ranked according to the expected error reduction incurred by their refinement. As described previously, defining such a heuristic is difficult due to error cancellation effects between elements. Therefore, a successful refinement heuristic must in some way take into account the aggregate effects of error cancellation, rather than applying a ranking methodology to each element dependent only on the error contribution of that element. Additionally, the element-wise error contribution for each element does not vary exclusively with its own size or basis function order but is instead dependent on the discretization fineness for all elements. If we have a positive total error estimate, we cannot

simply refine a few of the elements with the largest positive error contributions and hope to sway the sum toward zero. This is in practice a poor approach and will typically result in higher total error despite refinement in the mesh.

Examination of each refinement approach in this chapter is performed by evaluating the relative error of the QoI calculated by a forward solve on a mesh with K refined elements at each of the locations in the parameter space defined in Table 3.1.2 with respect to the analytical QoI at those parameter space locations. A base mesh ($K = 0$) is used for all test cases that contained 100 first-order elements of size 0.05 m. K is then varied from 0 to 100 for each simulation, using the heuristic to select a constant dK elements to refine at each subsequent refinement. p-refinement of an element consisted of increasing M for that element by 1, while h-refinement entailed splitting the element into two elements of size 0.025 m. Relative error calculated in this manner is here referred to simply as error. Adjoint solutions are calculated on meshes of one order higher than the forward solution for each location in the parameter space to obtain an error estimate for each location and K. These error estimates are added back onto the QoI to produce a corrected QoI, and the relative error of this corrected QoI which with respect to the analytical QoI is referred to as the corrected error. All relative error values are given as percentages for clarity. A vector of relative error values, formed separately for uncorrected and corrected results, is recorded for each test case. The entries of these vectors correspond to the relative errors for each K tested. To show general trends, these vectors are averaged for a given heuristic trial, giving for each a vector of average relative errors, $\mathbf{k}$, over the range of K for both uncorrected and corrected errors.

To quantify the efficacy of various refinement heuristics, we define a so-called improvement metric as

$$I(\boldsymbol{k}) = \frac{\sum \text{sign}(\text{diff}(\mathbf{k}))}{\text{length}(\mathbf{k})} \tag{1.38}$$

where **k** represents either the uncorrected or corrected error, length(**k**) is the length of the vector **k**, and diff(**k**) returns a vector of length one lower than **k** containing the differences in value between adjacent entries of **k**. A lower value of this metric implies better performance with −1 or 1 representing a heuristic that always decreases or increases, respectively, the error with increasing number of refined elements. An I value of zero represents a heuristic that has an equal chance to increase or decrease error with additional refined elements.

The first refinement heuristic explored is referred to as the magnitude refinement heuristic. This heuristic simply ranks elements by the absolute value of their error contribution estimate, such that elements with higher error contribution estimate magnitude rank higher. Note that this heuristic does not directly satisfy our earlier desire for a heuristic that considers aggregate error cancellation effects for refinement rather than applying a ranking methodology to each element dependent only on the error contribution of that element. Respective results for uncorrected and corrected relative errors and two different dK values for the magnitude refinement heuristic applied to p-refinement are shown in Fig. 1.4. Fig. 1.5 shows corresponding results for the magnitude refinement heuristic applied to h-refinement.

We see from Fig. 1.4 that the magnitude refinement heuristic informed by adjoint information reduces the QoI relative error from ~30% to <1% by K = 30 when applied to p-refinement. We also see that the corrected QoI obtained by adding the error estimate from (1.23) obtained by the adjoint solve to the forward QoI is vastly more accurate, with an initial relative error of ~0.42% which is reduced below 0.01% by K = 30. For K = 30, we have therefore reduced the initial error in the QoI by over three orders of magnitude with just four solves—two forward and two adjoint. Note also that all refinement performed here is entirely automated, requiring no input from the user other than a desired K, and furthermore that the technique used is not dependent

on the dimension of the problem, variety of element, or volume vs. surface nature of the discretization. This demonstrates the usefulness of adjoint-assisted targeted, adaptive refinement for difficult-to-refine higher-order techniques based on the FEM and/or MoM where efficient automated discretization refinement presents a challenge.

As is inherent to the scaling of FEM and MoM error with *h*-refinement vs. *p*-refinement, we see that convergence is more-gradual in Fig. 1.5 for the magnitude refinement heuristic applied instead to *h*-refinement. We still observe desirable reduction in error, however, with the relative error decreasing from ~30% to ~10% and ~0.4% to <0.1% for $K = 60$. Note that this does not seek to discount the usefulness of *h*- refinement as a technique—a mesh (model) insufficient to describe a given problem certainly requires both *h*- and *p*-refinements to obtain a useful solution efficiently––but rather seeks to point out the power of adjoint-informed *p*-refinement on meshes already *h*-fine enough to describe the problem of interest.

The second refinement heuristic explored is more complicated and will be referred to as the greedy refinement heuristic. The greedy refinement heuristic is an approximate approach to a more-desirable but computationally untenable approach here referred to as minimum sum grouping. Instead of seeking the K elements that should be refined, minimum sum grouping seeks a solution to the problem of which elements should not be refined.

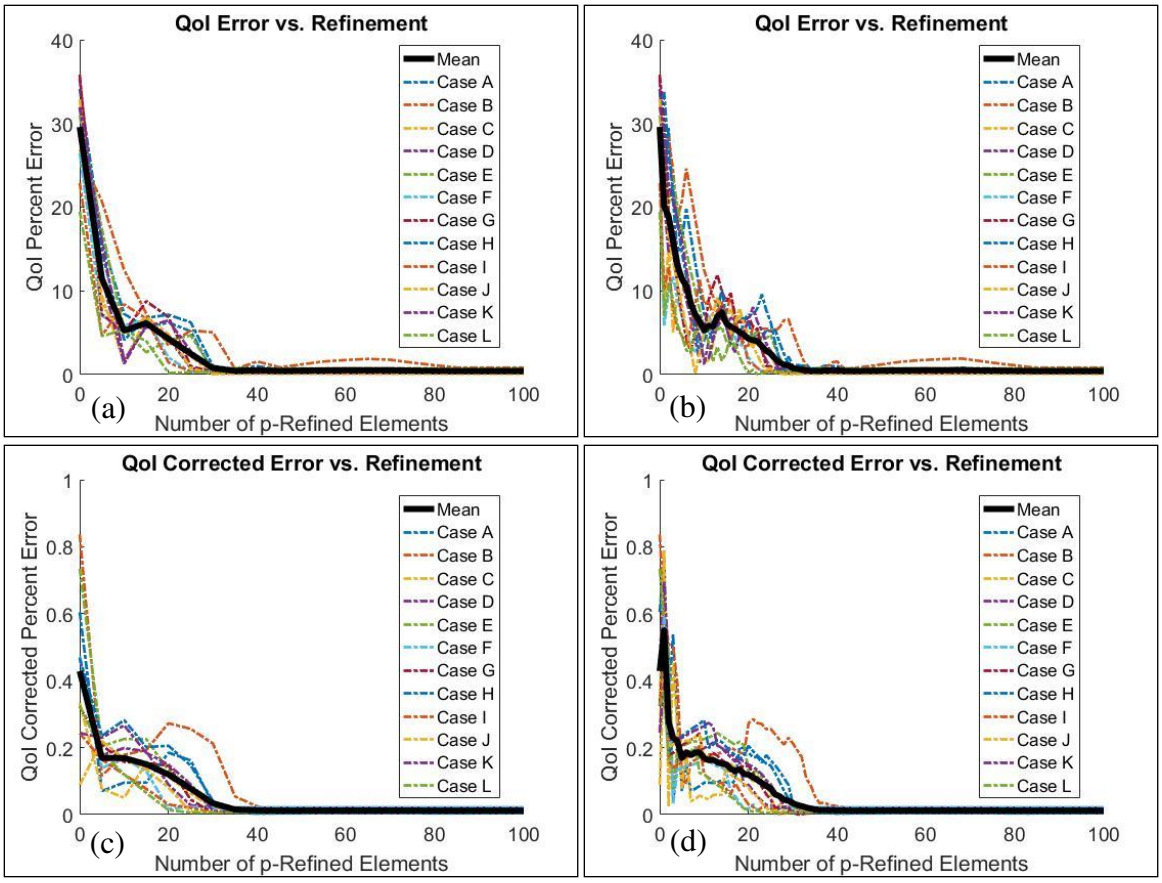


Fig. 1.4. Relative errors of the QoI calculated at each of the locations in the parameter space defined in Table 1.2 with respect to the analytical QoI for the magnitude refinement method implemented with *p*-refinement vs. the number of refined elements: (a) uncorrected error for dK = 5, (b) uncorrected error for dK = 1, (c) corrected error for dK = 5, and (d) corrected error for dK = 1.

In concrete terms, it computes the subset, **e**', of entries in **e** of length length(**e**) − K the absolute value of the sum of which is the smallest possible for a given **e** and K. The K elements selected for refinement by this method are then the elements associated with the remaining K entries in **e** that are not in **e**'.

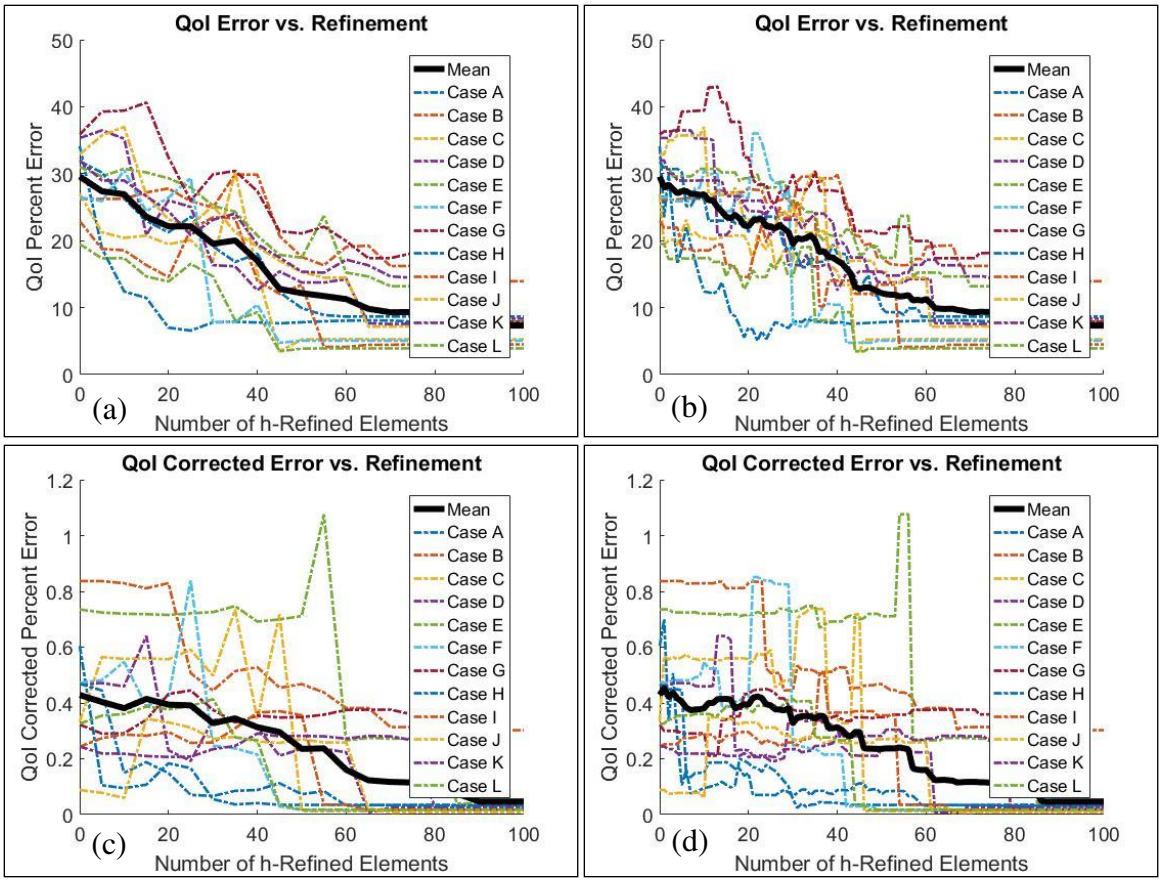


Fig. 1.5. The same as in Fig. 1.4 but for the magnitude refinement method implemented with *h*-refinement.

The subset **e**' is in practice not computable, as it requires evaluating the absolute value of the sum of all possible combinations of entries in **e** of length length(**e**) − *K*. The computational complexity of this problem scales factorially, making the application of minimum sum grouping to problems of even moderate element count computationally infeasible. Hence, we use a "greedy" refinement heuristic that offers a greedy approximate solution to the minimum sum grouping approach. We begin by computing a length(**e**) by length(**e**) matrix **S** from **e** with entries of the form

$$s_{ij} = \text{abs}(e_i + e_j) \tag{1.39}$$

neglecting the diagonal entries ($j = i$), and we find the minimum s_{ij}, appending the corresponding e_i and e_j to an ordered list. We then select the second smallest s_{ij} in **S** that does not include any of the particular **e** entries used previously, appending its corresponding e_i and e_j to the end of the

same ordered list. This is repeated until all entries of **e** have been included in the ordered list, or for odd length(**e**), one entry remains, in which case this entry is appended to the end of the list. The K elements chosen for refinement are then those corresponding to the last K entries in the list. The greedy refinement heuristic approached in this way can be evaluated in polynomial time. Note that the order in which we append e_i and e_j to the list for a given iteration will somewhat affect results. This effect becomes insignificant in practice once $K >> 1$. For the purpose of this chapter, we place e_j before e_i. Figs. 1.6 and 1.7 show the results for the greedy refinement heuristic applied to p- and h-refinements, respectively.

We see from Figs. 1. 6 and 1.7 that the adjoint-informed greedy refinement heuristic performs similarly to the magnitude refinement technique regarding error reduction over the range of tested K values. Comparing these figures to Figs. 1.4 and 1.5, note that, although both approaches trend downward rather smoothly [the improvement metric in (1.38) is strongly negative], this is not true of individual cases. By observation of the heuristic behavior for individual cases, we see there are several instances where the refinement of additional elements increases the error, sometimes substantially. Note, however, that few cases exceed the initial $K = 0$ error for another K, i.e., the error may increase from one K to another, but rarely does it exceed the initial value ($K > 0$ still leads to a reduction in the initial error for almost all K). Exceptions to this, for instance in Fig. 1.6(d), tend to be for narrow ranges of K making it less likely these undesirable K-values will be encountered by chance. This is reflected by a reduced occurrence and severity of these error-increasing K values for higher dK, for instance comparing Fig. 1.6(d) to Fig. 1.6(c).

This highlights and exemplifies the previously-stated desirability of heuristics that take into account aggregate cancellation effects—the tested heuristics perform better for higher dK as

choosing a larger pool of refined elements increases the likelihood the error contribution of a given element will be sufficiently cancelled. Heuristics that group elements in one way or another are therefore often more effective. Note then, that the magnitude refinement heuristic applied to dK > 1 in this way now satisfies the previously stated desire for a heuristic considering aggregate effects.

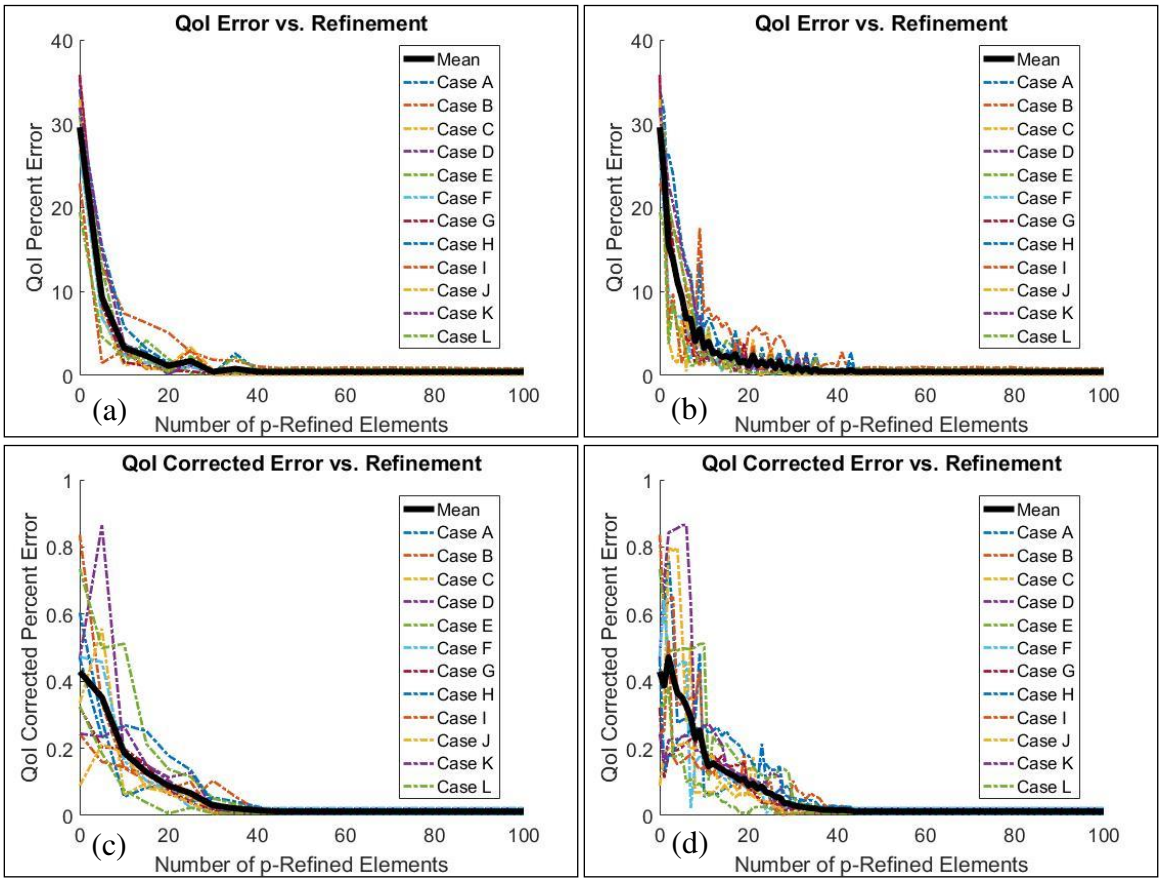


Fig. 1.6. The same as in Fig. 1.4 but for the greedy refinement method (implemented with *p*-refinement).

The greedy refinement heuristic and magnitude refinement heuristic are evaluated against a benchmark refinement method in which K elements are chosen for refinement at random. Results for the random refinement heuristic are shown in Figs. 1.8 and 1.9 as applied to *p*- and *h*-refinements, respectively. Comparing these figures to the results for the two adjoint-informed heuristics, in Figs. 1.4–1.7, we see that, although the random refinement approach of course

achieves the same error reduction for $K = 100$ (all elements refined), error reduction for nearly all other values of K is significantly worse.

Tables 1.3 and 1.4 show improvement metric values, based on (1.38), for the results in Figs. 1.4–1.9 for p- and h-refinements, respectively. Examining the tables, we see that desirable vs. undesirable behavior of the heuristics demonstrated in Figs. 1.4–1.9 can be partially captured by the improvement metric, most notably their degree of monotonicity. Without exception, the metric is poorer (less negative) for applications of the heuristics with $\mathrm{d}K = 1$ rather than $\mathrm{d}K = 5$.

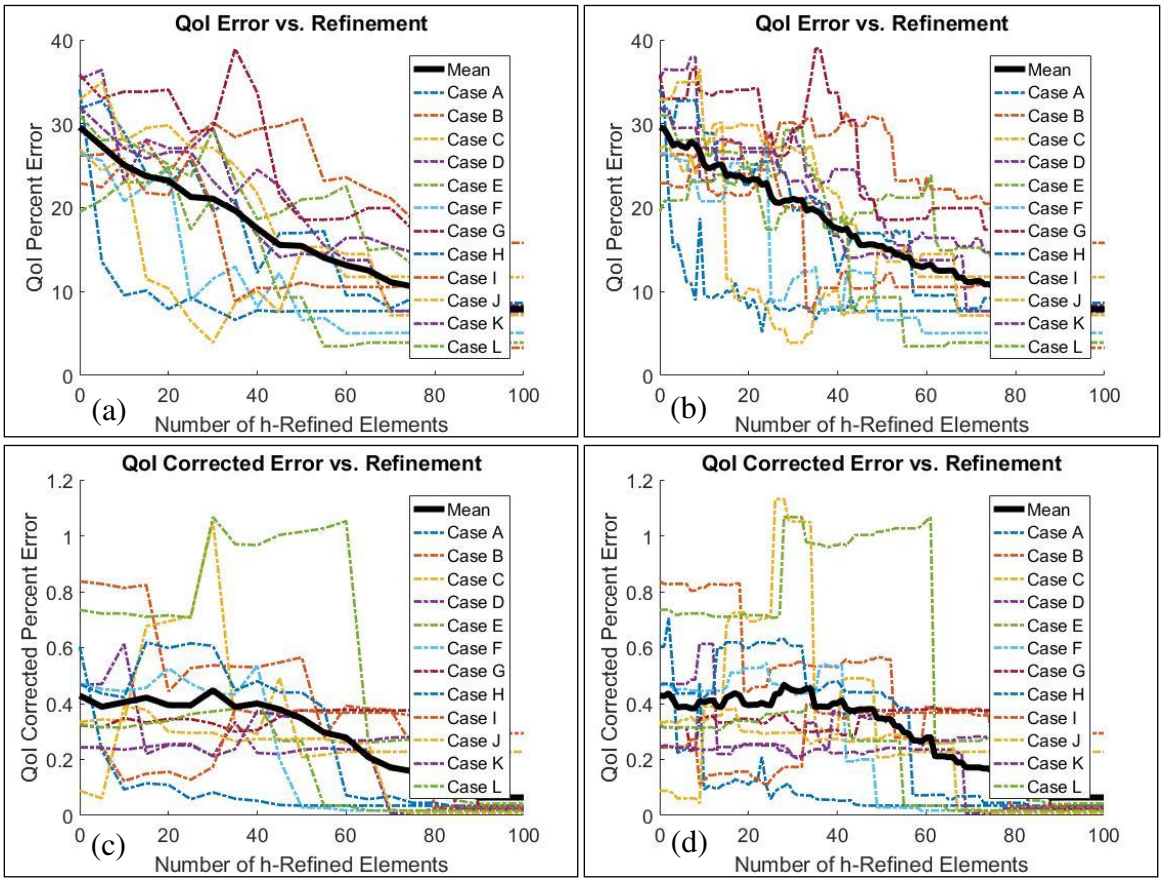


Fig. 1.7. The same as in Fig. 1.4 but for the greedy refinement method implemented with h-refinement.

This reflects the described grouping behavior advantageous to higher $\mathrm{d}K$ values. We also note that the random refinement heuristic presents lower metric values than the two adjoint-informed heuristics in all cases, indicating the superior performance of the adjoint-informed techniques. In

addition, examining the results in Table 1.4 in comparison to those in Table 1.3, we note that, interestingly, although heuristics applied to *h*-refinement reduce error more slowly, they tend to do so somewhat more consistently, reflected in the higher improvement metric values for most entries in Table 1.4 than the corresponding entries in Table 1.3.

TABLE 1.3. Improvement metric values based on (1.38) for magnitude, greedy, and random refinement heuristics applied toward *p*-refinement, i.e., for the results in Figs. 1.4, 1.6, and 1.8. The best value for each column has been bolded.

p-refinement	d$K = 5$		d$K = 1$	
Heuristic	Uncorrected	Corrected	Uncorrected	Corrected
Magnitude	**−0.3**	−0.2	**−0.22**	−0.20
Greedy	**−0.3**	**−0.4**	−0.08	**−0.28**
Random	−0.2	−0.3	−0.02	−0.0

TABLE 1.4. The same as in Table 1.3 but for *h*-refinement, i.e., for the results in Figs. 1.5, 3.1.7, and 1.9.

h-refinement	d$K = 5$		d$K = 1$	
Heuristic	Uncorrected	Corrected	Uncorrected	Corrected
Magnitude	−0.7	**−0.6**	**−0.38**	−0.10
Greedy	**−0.8**	−0.4	−0.22	**−0.12**
Random	−0.2	−0.0	−0.02	−0.04

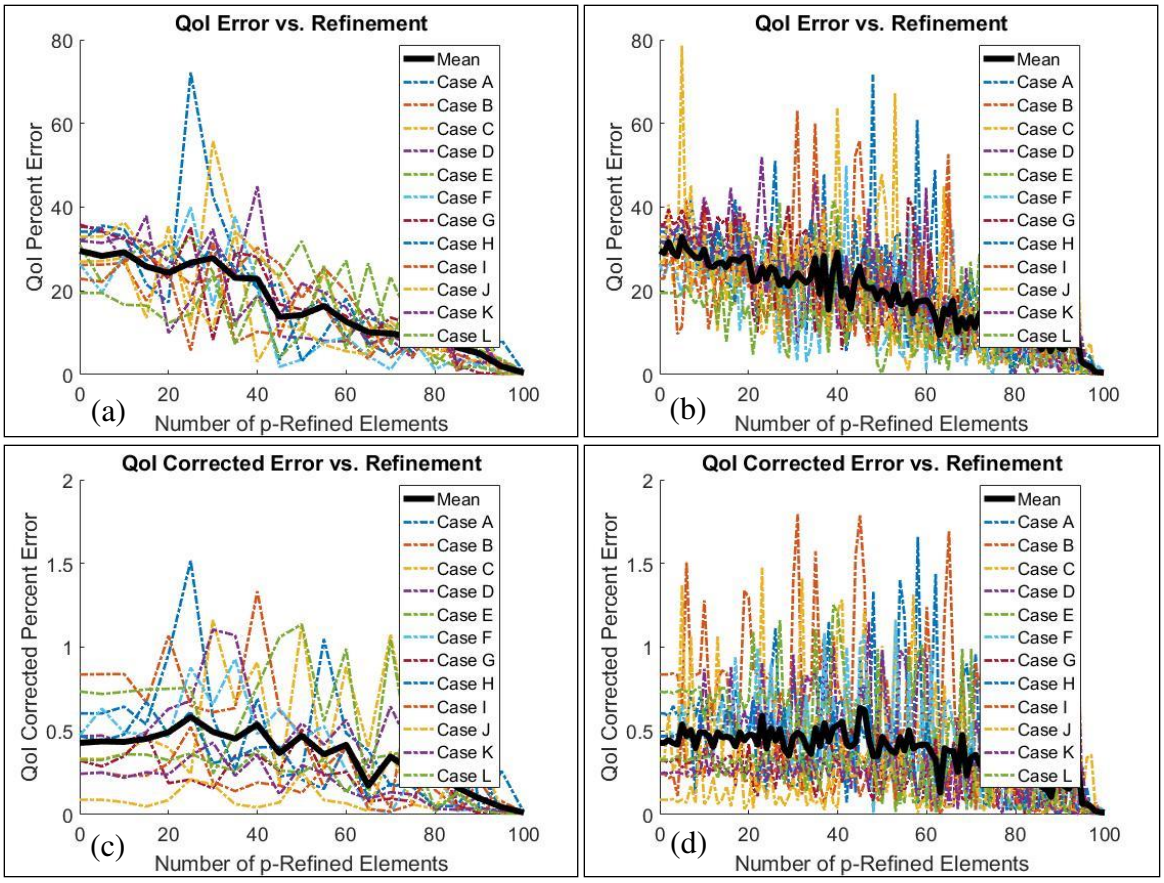


Fig. 1.8. The same as in Fig. 1.4 but for the random refinement method (implemented with *p*-refinement).

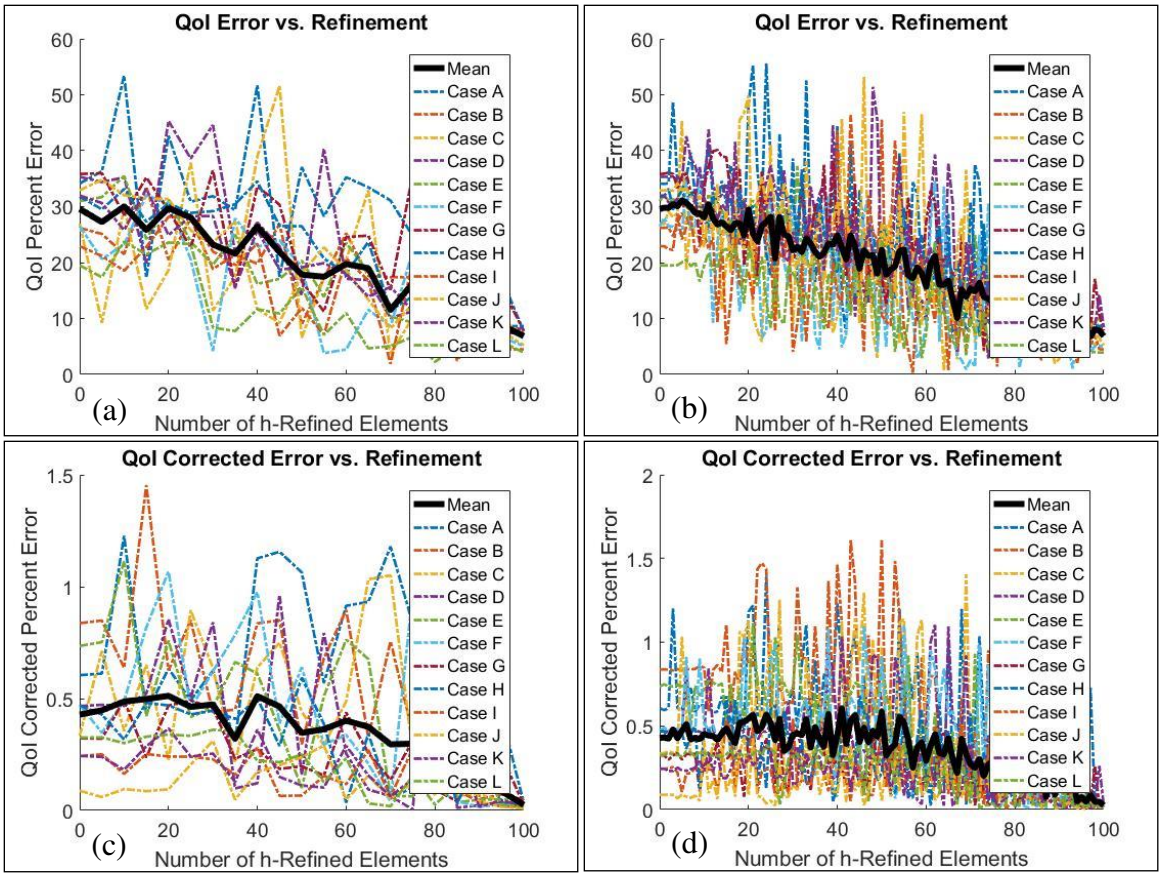


Fig. 1.9. The same as in Fig. 1.4 but for the random refinement method implemented with *h*-refinement.

1.7 Conclusion

This chapter has investigated and evaluated useful applications of the adjoint problem and its solution for higher-order frequency-domain computational electromagnetics methods. Based on implementation of HOPS, QoI error estimation and error correction, element-wise error contribution estimate evaluation, and adjoint-informed automated targeted p- and h-refinements, this study has established and validated uses of adjoint techniques for improved efficiency, automation, and robustness of higher-order frequency-domain methods. Although the techniques applied in this chapter have been demonstrated using a higher-order solver, all, with the exception of p-refinement, apply with no modification to low-order solvers. We have employed a one-dimensional higher-order PML-truncated FEM scattering solver as an ideal testbed for the ease of

implementation, clarity of displaying the results, and intuitiveness of drawing conclusions from analyses, which then extend naturally to higher-dimensional solvers, more-complicated CEM techniques, adaptive CEM solutions, and problems requiring many solves.

Adjoint-based error estimation determines accurately whether a given discretization sufficiently describes a problem, and such error estimates can be applied to automated h- and p-refinement heuristics with little if any input from the user. Such heuristics reduce error quickly and vastly outperform a random refinement benchmark. On the tested problems, these heuristics are enough to reduce error by several orders of magnitude while only p-refining a modest number of elements (30/100) by one order using hierarchical basis functions and only four solves (two forward and two adjoint). Furthermore, these techniques can reduce error by a factor of more than 4.5 while h-refining roughly half of the elements in the domain (60/100). Most usefully, the adjoint-assisted h- and p-refinement methods we have demonstrated in this chapter are near-monotonic in their error reduction with respect to number of refined elements. The usefulness of the demonstrated adjoint-informed refinement compounds with the p-refinement technique offered by higher-order FEM or MoM frequency-domain methods, especially on meshes already h-fine enough to describe the problem of interest, but nonetheless offers substantial error reduction and excellent convergence properties for low-order methods using h-refinement schemes. In fact, while heuristics applied to h-refinement reduce error more slowly, they result in higher improvement metric values than p-refinement.

To the best of our knowledge, this is the first demonstration of applicability of adjoint a posteriori error estimation techniques to adaptive discretization refinement in the field of CEM using arbitrary-order basis functions. In addition, among the novelties this work introduces to CEM are the application of a dual-weighted residual estimate to the adjoint-based a posteriori error

estimation, the selective adaptivity based on error cancellation, and p-refinement using the adjoint solution. Unlike existing error estimates used in CEM that seek to bound error in a norm, the signed nature of this estimate is exploited to cancel local error contributions by grouping, leading to rapid reduction in global QoI error with a high selectivity. Our work has produced novel targeted model refinement heuristics that quickly and effectively reduce error in a quantity of interest. The study has demonstrated the exceptional benefits that adjoint techniques offer toward targeted, adaptive h- and p-refinement schemes using these heuristics. It has also attained a useful and broadly applicable improvement metric as a figure of merit for different refinement heuristics while providing an instructive discussion of the properties of a refinement heuristic that produce desirable values of this metric. In addition, we have demonstrated how HOPS can be used to obtain useful gradient information with respect to several parameters with vastly fewer additional solves than classical methods, requiring n – 1 fewer solves to compute the gradient, where n is the number of parameters with which the QoI varies. The additional applicability of this technique toward producing surrogate functions for optimization has also been shown. The surrogate functions generated, although piece-wise linear, closely match complicated QoI responses to various parameters.

In general, adjoint techniques are under-utilized in CEM where they could be applied to a wide variety of problems. The simple, one-dimensional FEM solver by which these relatively complicated adjoint techniques have been demonstrated for the purposes of this study should serve as a useful, easily replicable introduction to the described methods. The developed and evaluated adjoint techniques proposed and discussed in this chapter may be used to derive and demonstrate useful applications of adjoint methods to more complicated CEM techniques and solvers. Namely, the methodology described in the chapter generalizes well to higher dimensional problems by

extension of, for instance, (1.11), (1.23), and (1.26) to the double-curl wave equation. In such a case, if three-dimensional problems are considered, the pertinent inner products become volume integrals of vector fields representing the three-dimensional measurement characteristic, forward field, and adjoint field. This study should be especially valuable for future development of adjoint-informed adaptive discretization *p*- and/or *h*-refinement schemes for such CEM techniques, as well as for adjoint-assisted CEM procedures applied to large-scale optimization problems, Monte Carlo simulations, RCS computation, and RF design problems among other uses.

2 APPLICATION OF ADJOINT METHODS TO THREE-DIMENSIONAL FEM SCATTERING PROBLEMS: EXPLORING CORRELATION BETWEEN A POSTERIORI ERROR ESTIMATES FOR RELATED SUB-PROBLEMS

2.1 Introduction

Error estimation, adaptive refinement, and uncertainty quantification are of growing interest in computational electromagnetics (CEM). In chapter one, adjoint-based techniques were demonstrated as effective approaches to these three related research areas. For error estimation problems, adjoint methods excel at producing accurate, signed error estimates for a quantity of interest (QoI), or many, stated as a linear or linearized functional on an approximate field solution obtained by finite element method (FEM), finite different (FD) method, or method of moments (MoM). Such error estimates are typically more accurate than those produced by a priori means or application of a norm. Adjoint-based a posteriori error estimates can be applied effectively to adaptive discretization refinement to dramatically reduce solution error in few solves, for instance using the techniques outlined in chapter one and [46].

Applied to uncertainty quantification, adjoint methods serve as an accurate approach to estimating QoI responses to uncertain model parameters. HOPS, described in chapter one and [37], is the most notable approach. Recent research has demonstrated HOPS can approach the accuracy of Monte Carlo for FEM scattering problems while using two orders of magnitude fewer solves [47], making HOPS a compelling technique for accelerating uncertainty quantification computations in CEM. For complicated uncertainty quantification problems and low error tolerances, HOPS can still require tens or hundreds of solves, however, making HOPS a large, multi-solve problem.

In chapter one, we speculated that adaptive refinement using adjoint-based a posteriori error estimates may be applicable to achieve efficiency gains for large, multi-solve problems like HOPS, Monte Carlo simulation, and radar cross section (RCS) computation, all of which require the solution of many similar sub-problems. Naïvely, we could imagine performing adaptive refinement for each sub-problem separately, perhaps performing a handful of simulations on adapted discretizations to meet an error tolerance for each sub-problem, treating each sub-problem as [46] does. However, as the individual sub-problems composing most multi-solve problems relate closely, we contend so too should their a posteriori elementwise error contribution estimates (EECEs). If this is true, then such refinement techniques could be iterated *across* sub-problems, rather than for each sub-problem, potentially yielding efficiency gains. Accordingly, this chapter presents explores the relatedness (as measured by Pearson's correlation) between EECEs for a set of lossy dielectric sphere FEM scattering problems, a 3D analogue of the lossy dielectric slab scattering problems explored in chapter one.

2.2 Theory and Problem Description

To define the adjoint operator for a given problem, we first put the problem in variational form. If the problem is nonlinear, we use the integral mean value theorem to write the problem in linear form. For the illustrative dielectric scatterer problem, we consider the solution of the Dirichlet boundary value problem:

Find $\mathbf{E}^{\text{sc}} \in \text{H}^1$ (the space of L^2 functions with L^2 first derivatives) such that the variational formulation of the following holds

$$\nabla \times \bar{\bar{\mu}}_r^{-1} \nabla \times \boldsymbol{E}^{\text{sc}} - k_0^2 \bar{\bar{\varepsilon}}_r \boldsymbol{E}^{\text{sc}} = -\nabla \times \bar{\bar{\mu}}_r^{-1} \nabla \times \boldsymbol{E}^{\text{inc}} + k_0^2 \bar{\bar{\varepsilon}}_r \boldsymbol{E}^{\text{inc}}, \tag{2.1a}$$

$$n \times \boldsymbol{E}^{sc} = 0, \tag{2.1b}$$

where the former equation holds through the volume of the domain and the latter holds on the boundary of the domain. Note that (2.1a) and (2.1b) describe the broader class of problems of which one-dimensional slab scattering problems used in chapter one are a small, simplified subset. This may be stated in linear operator form as

$$L\boldsymbol{E}^{\mathrm{sc}} = \boldsymbol{g}, \tag{2.2}$$

a more-general form of (1.8). The adjoint operator is then that which satisfies the following identity:

$$\left\langle \boldsymbol{E}^{\mathrm{adj}}, L\boldsymbol{E}^{\mathrm{sc}} \right\rangle = \left\langle L^{\mathrm{adj}}\boldsymbol{E}^{\mathrm{adj}}, \boldsymbol{E}^{\mathrm{sc}} \right\rangle \tag{2.3}$$

the adjoint problem then given by (1.11).

To generate a small dataset of sub-problems from which to explore the relatedness of EECEs, we simulated a set of 41 lossy dielectric sphere scattering sub-problems. Each sub-problem consisted of a lossy sphere 2 wavelengths in diameter encased in a spherical shell of air 0.3 wavelengths thick. The domain was truncated using a shell of conformal perfectly matched layer (PML) with 0.3 wavelength thickness surrounded by PEC. A single plane wave was used to excite each sub-problem, and double-higher-order frequency domain FEM was used to generate an approximate field solution for each, as in [47]. All discretizations were topologically identical, containing 256 higher-order quadrilateral elements, and used the same basis. Sub-problems varied only in the lossy dielectric sphere real relative permittivity, with relative permittivity values ranging from 4.0-2.0j to 8.0-2.0j. Figure 1 shows the basic geometry of the domain.

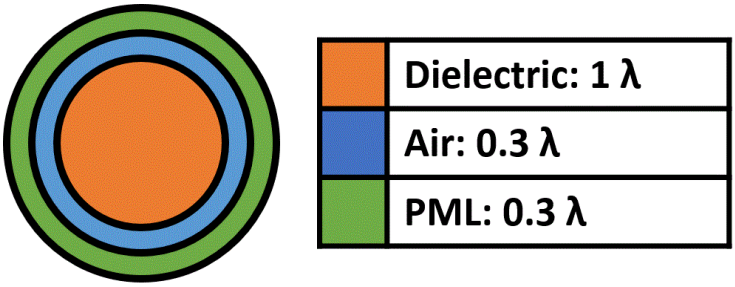


Fig. 2.1. Lossy dielectric sphere problem geometry.

Using the backscattered field QoI from [47], a posteriori EECEs were computed using higher order adjoint solutions, as in as in [46], for all sub-problems. Solution examples for the lossy dielectric sphere scattering problem are shown in Figs. 2.2 and 2.3 (note that a low relative permittivity is used for these figures to produce field features easily identifiable at the chosen figure resolution).

Adjoint solutions were also applied to compute a QoI value and QoI gradient for each sub-problem using the methods of [47]. Note that all QoI values, QoI gradients, and element-wise error contribution estimates are complex-valued.

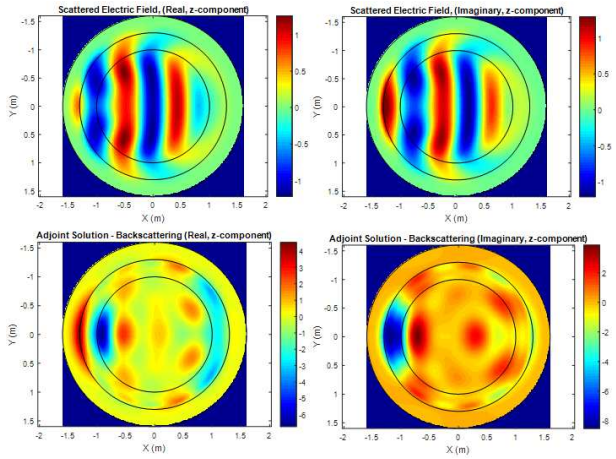


Fig. 2.2. Solutions to the scattering problem for a lossy dielectric sphere of ε_r=2.25-1i and radius 1 meter in free space. Domain terminated with perfectly matched layer with perfect electric conductor (PEC) exterior boundary. Incident wave was chosen to arrive from $\theta = 90°$, $\phi = 0°$ and θ-polarized at 300 MHz with unit electric field amplitude. Solutions to the adjoint problem are given below the corresponding forward solution.

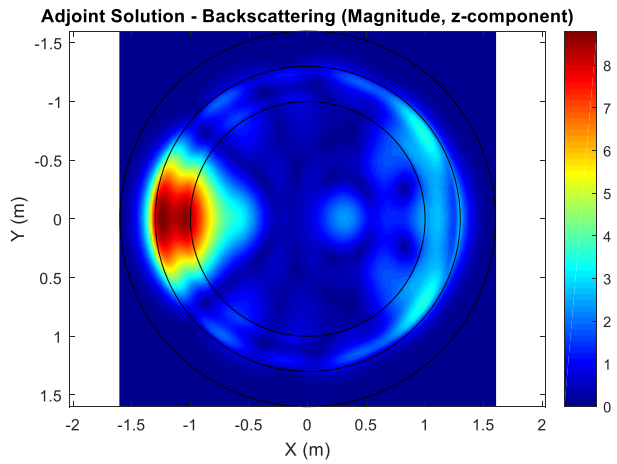


Fig. 2.3. Magnitude of the z-component of the adjoint solution for the problem described in Fig. 1. The areas most affecting the backscattered field lie in the region facing the radar receiver and the Arago spot. Z = 0 cross section is given.

2.3 Results and Discussion

Solving and computing error estimates for all 41 sub-problems produced 41 complex valued EECEs for each element. These EECEs were ordered by real relative permittivity. For each of the first 33 EECE values per element, we cut a subset of length 9, consisting of the EECE value and the next 8 in the list. Performing this for all 256 elements, we had a total of 8,448 9-dimensional observations, the dimensions corresponding to relative indices between a given sub-problem and the 8 next sub problems. We similarly sampled 9-dimensional observations for real QoI values and QoI gradients. We computed correlation coefficients between all 9 dimensions for EECEs, QoI values, and QoI gradients, summarized in Fig. 2.4.

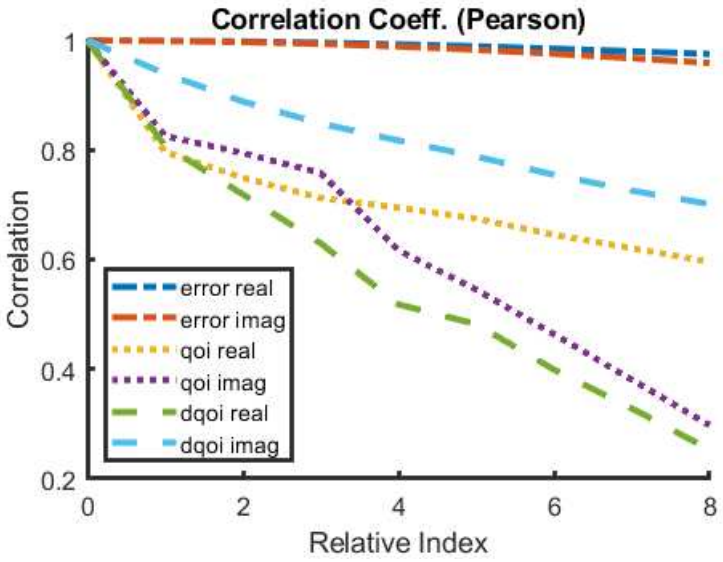


Fig. 2.4. Correlation coefficients for EECEs (error), QoI values, and QoI gradients with respect to relative permittivity (dqoi). Error contribution estimates are substantially more correlated between related sub-problems than QoI values or QoI gradients.

As evident in Fig. 2.4, correlation coefficients for EECEs decay slowly with respect to increased separation between sub-problems (relative index), and therefore with respect to increasing relative permittivity difference. Correlation coefficients for QoI values and QoI gradients decay far more quickly, for reference. However, for applications like adaptive refinement, we care about values of some refinement indicator derived from EECEs, rather than EECE values themselves (complex values cannot be ordered consistently). For a simple example refinement indicator, we use the magnitude of each EECE.

EECE magnitude has often been used as a simple benchmark for adaptive refinement using adjoint-based a posteriori error contribution estimates, for instance in chapter one or [46]. Fig. 2.5 shows EECE magnitudes for all 41 sub-problems and all 256 elements, ordered by the mean EECE magnitude over all sub-problems for each element. Qualitatively, the vertical banding of Fig. 2.5 suggests how slowly EECE magnitudes, and therefore our example refinement heuristic, vary over the range of real relative permittivity values tested.

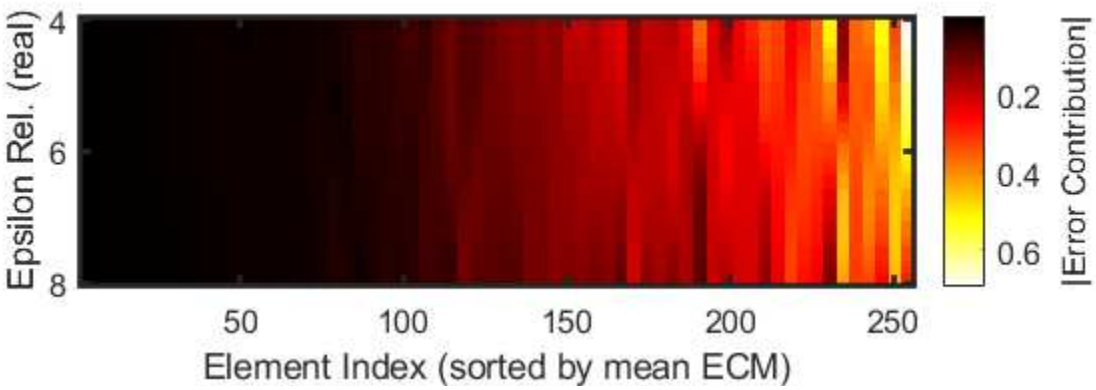


Fig. 2.5. EECE magnitudes. Vertical axis corresponds to sub-problems ordered by real relative permittivity. Horizontal axis corresponds to element index sorted by mean EECE magnitude (ECM). Mean was evaluated for each element over all sub-problems. Note that several elements have nearly identical error trends due to symmetry of the problem.

However, Fig. 2.5 also qualitatively suggests something potentially troubling: elements with higher mean EECE magnitude cover a wider range of EECE magnitude values with varying real relative permittivity. Fig. 2.6 presents this trend qualitatively, showing EECE magnitude standard deviation with respect to EECE magnitude mean. We find the two are correlated with Pearson correlation 0.73.

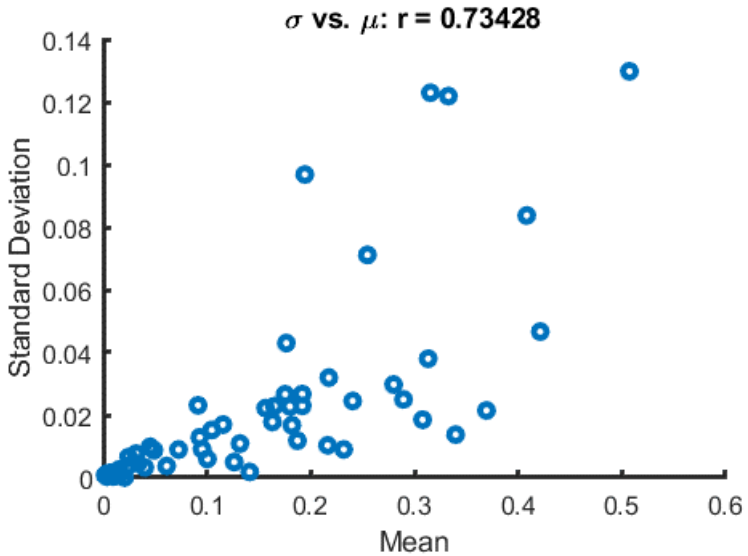


Fig. 2.6. EECE magnitude standard deviation with respect to mean. Standard deviation and mean are correlated.

This suggests that EECE correlation for such elements with high refinement indicator values may decay more quickly. If EECE correlation decays too quickly between related sub-problems, adaptive refinement methods like those presented in [46] may be unable to converge to desired error tolerances when applied across sub-problems. However, in Fig. 2.7, we suggest this may not be the case for existing adaptive refinement schemes. Fig. 2.7, similar to Fig. 2.4, shows correlation coefficients between EECE values for neighboring sub-problems. Unlike Fig. 2.4, 2.7 also shows correlation coefficients for the top and bottom 25% of elements (as ordered by mean EECE magnitude). Even for elements with EECE mean magnitude values within the highest 25%, EECE values remain highly correlated over the relative index range tested.

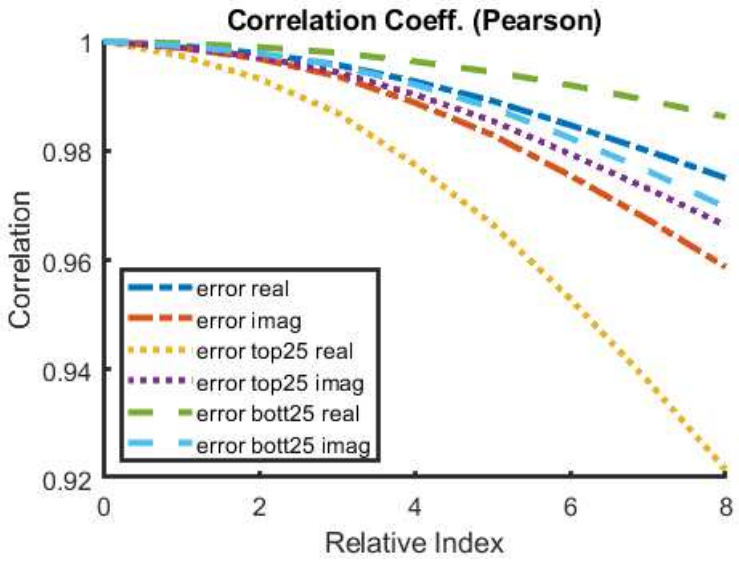


Fig. 2.7. Correlation coefficients for EECE (error) for all problems, problems in the top 25% of EECE magnitude, and problems in the bottom 25% of EECE magnitude. Note the difference in vertical axis scale from Fig. 2.4. Top 25% and bottom 25% have similar correlation between related sub-problems.

As demonstrated in [46], many adaptive refinement approaches can reach low tolerances within a few iterations, so the mild correlation coefficient decay shown in Fig. 2.7 is likely tolerable, even given increased adaptive refinement convergence time due to application across sub-problems.

2.4 Conclusion

Correlation statistics were presented for elementwise error contribution estimate values for 41 lossy dielectric sphere scattering problems with varying relative permittivity. We found that EECE values were highly correlated between related sub-problems, and their correlation coefficients decayed slowly with increasing difference in relative permittivity between problems. These results suggest that application of adjoint-based a posteriori EECE-informed adaptive refinement methods across sub-problems may be able to achieve efficiency gains for multi-solve problems.

3 DISCRETE SURFACE RICCI FLOW FOR GENERAL SURFACE MESHING IN COMPUTATIONAL ELECTROMAGNETICS USING ITERATIVE ADAPTIVE REFINEMENT

3.1 Introduction

Mesh generation is a critical, yet largely neglected, aspect of research in CEM The quality of a surface discretization has substantial impacts on the numerical solution and solution efficiency of electromagnetics problems, yet many researchers and innovators of new surface integral equation (SIE) techniques within the MoM framework seem to mostly defer this aspect due to its difficulty. Meanwhile, as more new simulation techniques emerge and problem sizes grow, a relatively static pool of surface meshing approaches must contend with an ever-increasing variety of surface mesh types, each with unique benefits but added geometric constraints. For many, promising, new MoM-SIE techniques, existing meshing approaches simply cannot automatically produce the required discretizations at any usable quality, limiting the applicability of new research and relegating practitioners of such approaches to heavily involved semi-manual meshing. SIE-based CEM methods also increasingly rely on numerical error estimate-based adaptive refinement techniques to efficiently and dynamically modify problem discretizations during computation [48], necessitating the integration of complicated surface meshing algorithms with existing CEM software. As such, it is of growing importance that CEM researchers and practitioners have access to and understanding of meshing techniques that can be implemented into research codes or commercial software products under development.

There is a variety of existing surface mesh generation approaches available to CEM researchers. However, to our knowledge, none allow for the automatic generation of meshes with user-defined element type, count, and order. Moreover, first-order triangular mesh generation is

well-explored and often simple due to desirable topological properties of triangles as a 2-simplex. For instance, see [49] for surface triangulation from arbitrary point clouds or [50] for improving existing triangular surface meshes. Triangular meshes can also be generated from arbitrary polygonal meshes by subdivision of any polygon [51]. Triangular meshes are ubiquitous in SIE numerical methods, for instance, see [52]. In terms of the best-known and most common mesh type of first order triangular meshes, namely, those built from flat triangles, much first-order triangular surface meshing in CEM has relied on Delaunay triangulation-based meshing approaches due to its simplicity and robustness [53, 54, 55]. Unfortunately, applying the Delaunay triangulation directly limits its applicability strictly to 2D (plate) structures or 2D domains. Other prior work for triangular surface meshing in CEM has focused on refining and improving an existing triangle mesh using various implementations of node addition with local mesh rearrangement [56, 57, 58]; quad-tri conversion [59]; and iterative refinement beginning from user-defined vertex labels [60].

First-order quadrilateral elements are also common in CEM, although less so than first-order triangular, see for instance [61]; meanwhile, first-order quadrilateral mesh generation is somewhat more difficult. Approaches typically rely on direct tri-quad conversion [62, 63], patch-based methods [64, 18], Voronoi-based methods [66], or parametrization-based methods [67, 68, 69]. See [70] for a recent overview of the state of the art in quadrilateral mesh generation. Our proposed method, applied to quadrilateral meshing, fits in the fourth category but maintains generality to other surface mesh types. Discontinuous quadrilateral meshes, in which adjacent quadrilaterals need not share entire edges, are a generalized case of quadrilateral meshes simple to generate at high quality from existing continuous quadrilateral surface meshes by merging or subdividing chosen quadrilaterals [70]; see [71] for the advantages of DG SIE methods. We

therefore first focus on continuous quadrilateral mesh generation but generalize to the discontinuous case by this property. Direct tri-quad conversion typically leads to poor mesh quality for most CEM applications (highly irregular elements with varied corner angles) and can greatly increase the element count of a mesh for little gain in fidelity. Voronoi-based approaches can lead to quad-dominant meshes in which many triangles remain, raising issues for SIE solvers not equipped to handle both quadrilateral and triangular elements simultaneously. Patch-based methods, meanwhile, are not guaranteed to generate a complete mesh and can fail during the generation process [70]. They are therefore unreliable for quintessential surfaces in CEM such as vehicles, antennas, or building environments. The parameterization-based approaches are typically the most robust, but little work has been done to optimize them or apply them for CEM applications with the exception of prior work in their application to parameter sweeping on 2D surface meshes [72]. Other CEM-specific first-order quad meshing work has typically concerned iterative partitioning by sets of node placement rules [73, 74]. More recently, much first-order quad meshing research in CEM has relied on approaches that, although robust, are limited to mixed quad-tri meshes and therefore not applicable for methods relying on meshes of a single type [75, 76, 77, 78].

Beyond the well-known first-order techniques, higher-order methods are of growing interest in CEM. Although such techniques have shown great promise reducing the system dimension for comparable or higher accuracy in MoM-SIE solvers, thereby reducing the computation time and/or increasing accuracy substantially [1, 5], the complexity of generating the needed higher-order quadrilateral or triangular meshes has limited the promised applicability of these techniques from true large-domain modeling. The methodology for generating such meshes is often left out of scope [7], semi-manual, or, at best, unable to effectively generate large-domain

elements, typically relying on the combination of several existing elements into larger high-order elements [1]. This can only be done on highly structured meshes, with most others lacking the topology for merging elements to satisfy common geometric interpolation techniques. No robust meshing process has been developed for this application, and existing parametrization-based techniques, optimized for first-order elements, are not well-suited to the task. Prior work seeking to produce robust and broadly applicable higher-order quadrilateral and triangular surface meshing techniques has used existing high quality first-order meshes of the desired type and subsequently interpolated them [79, 80]. This approach is excellent where such meshes are available but cannot be used when such meshes are unavailable or difficult to produce at high quality, for instance when large-domain quadrilateral meshes are desired. Overall, meshing is an open problem of great relevance, and potentially the most challenging and restrictive component of higher order CEM.

This chapter proposes an efficient and robust general surface meshing technique that is applicable to any of the discussed mesh types and extends easily to others. Able to seamlessly handle higher order and very high order elements, the developed technique is aimed to finally surmount the major barrier to widespread use of otherwise highly efficient higher order methods. Meanwhile, due to its generality, the technique also constitutes a robust and competitive low-order meshing approach useful for first-order triangular (flat triangles) and first-order quadrilateral mesh generation at high quality. Due to its conformal nature, the technique also maintains high local regularity of surface Jacobians and high basis function orthogonality, preventing degeneration of the basis functions and associated increases in condition number. A parametrization-based approach, the technique leverages the discrete surface Ricci flow to map between an arbitrary triangulated mesh and an appropriate canonical parametric domain dependent on the underlying geometry of the original surface and desired mesh properties. A uniform seed mesh, known in the

parametric domain, is then taken by this mapping to the original surface, on which a refinement indicator, defined for the choice of mesh type, is computed. The seed mesh is then refined in the parametric domain based on this indicator and the process is repeated until some stop criterion is met, for instance maximum element size or element count. Although we begin here with a triangular mesh of the original surface, in general we can begin with an arbitrary surface representation, this representation then needing conversion to a high-quality triangulation by sampling, subdivision, or surface reconstruction as appropriate to the type of surface representation, for instance using techniques from [49] or [51].

Overall, this appears to be the first demonstration of a general, robust surface meshing technique able to seamlessly handle arbitrary (low and high) geometric order and element type. Furthermore, this appears to be the first technique able to generate high-quality very high order elements; the lack of such a technique previously constituting the main shortcoming of large-domain methods [1, 7]. We demonstrate the technique for a variety of common mesh types used for the low-order and higher order MoM-SIE methods including triangular, quadrilateral, and discontinuous quadrilateral, and offer suggestions for simple extension to other, less common element types. For each of these types, we maintain generality in element order and show typical results for the most commonly used first-order (lowest-order), e.g., flat triangular patches, as well as higher-order elements. For the latter, we demonstrate meshes using both Lagrange interpolation and cubic spline interpolation, but again offer suggestions for extension to other interpolation methods. We focus here on the application of this technique to MoM-SIE in CEM but do not limit its usefulness to only this application. Some preliminaries of this work are presented in a summary form in [81, 82, 83].

We note a few limitations of the proposed method. Firstly, the method as presented does not enforce continuity for multi-part objects. To work for such cases in practice, the method therefore requires either a discontinuous Galerkin solver or special treatment at part interfaces to enforce continuity. The adaptive refinement we present here also does not guarantee perfect sampling of the original surface on non-differentiable features (i.e., sharp edges). The error due to imperfect sampling of sharp features drops asymptotically to zero with increased iteration but may be unacceptable where perfect preservation of sharp features at otherwise low mesh fidelity is required. Although all refinement methods we present scale as $O(NlogN)$ with the number of iterations, the time complexity of the DSRF with the number of triangles remains formally unknown and is likely the asymptotically dominant factor. We are aware of no study of DSRF scaling with respect to the triangle count, and we consider such a derivation involved enough to warrant its own, separate study. With respect to multiple parameters, DSRF scales formally as $O(N_t N_m N_n)$, where N_t is the number of triangles in the original mesh, N_m is the average number of iterations required to solve the Hessian system to a chosen tolerance, and N_n is the number of Newton method iterations required to meet a tolerance on the curvature error. The open problem is relating N_n and N_m to N_t.

In the rest of the chapter, we begin by recapitulating the mathematics of the discrete surface Ricci flow, tailored specifically to be understandable and useful to the CEM community and having the MoM-SIE modeling goals in mind. In this, the key aspects of the DSRF are outlined as it applies to the present work at a level understandable to those without significant experience in formal topology or geometry. We encourage readers interested in a more-formal, in-depth theoretical discussion with additional implementation details to review [84]. We use the same notation where applicable. We then describe the developed surface meshing technique, with

emphasis on iterative adaptive refinement, and offer specific implementations for common mesh types. Following this, we present a variety of meshes produced by the method as well as statistics on mesh quality. In particular, we present meshing results using the proposed DSRF meshing technique with iterative adaptive refinement for the following two illustrative examples. To demonstrate the proposed method on a well-known CEM test case, we have applied adaptive Ricci flow meshing to the NASA almond, an established CEM benchmarking shape usually used to demonstrate difficulty of surface modeling given its one sharp end. To test the technique on a more-complicated case, we have also produced meshing results for a far more-complicated fighter-jet model. For each of these models, we show continuous triangular, continuous quadrilateral, and discontinuous quadrilateral surface meshes of both low and high geometric order. We conclude further outlining the potential of the new DSRF meshing technique with adaptive refinement.

3.2 Summary of the Method

Since much of the material covered in this chapter is predicated on recent and non-trivial theory work in discrete conformal geometry and computer graphics with which most CEM users and practitioners will have little experience, we begin with a high-level summary of the method to contextualize the details presented in the remainder of the chapter. The basic steps of the method are as follows:

1. If it is not already, convert the surface to be meshed into a triangle mesh
2. Obtain a mapping between the original surface and a simple surface of prescribed curvature by DSRF
    a. Choose a simple surface in which we can easily manipulate mesh topology (e.g. a flat rectangle)

b. Assign a target curvature to each point in the original triangle mesh consistent with the prescribed surface

c. Perform the DSRF to compute locations of vertices from the triangle mesh when flattened to the prescribed surface

3. Apply the mapping to resample the original surface adaptively (adaptive refinement), manipulating mesh topology in the simple prescribed surface (parametric) domain

a. Define a simple seed mesh that covers the prescribed surface

b. Define a refinement indicator and refinement method appropriate for the target mesh type

c. Compute the refinement indicator for the seed mesh

d. Refine the seed mesh using the refinement method, based on computed values of the refinement indicator

e. Iterate steps d and e until some stop criterion is met (number of steps, maximum element size threshold, etc.)

Step 1 is simple for almost all surface descriptions, so we consider it outside the scope of this chapter. As we are resampling the triangle mesh of the original surface, we require that this triangle mesh is sufficiently dense to capture relevant surface features. Note that we require a triangle mesh regardless of the desired output mesh type.

Step 2 is described in section 3.3 in detail, but we offer key considerations here. The surface of prescribed curvature can be arbitrary but should be a surface on which it is simple to manipulate mesh topology. We use the simple and broadly applicable example of a flat Euclidean rectangle

for this chapter. If the original surface is closed (has no boundary), choosing the prescribed surface is substantially easier, as we can assign a constant target curvature to all vertices in the original triangle mesh to flatten it to one of three canonical domains, the sphere, the Euclidean plane, or the hyperbolic plane. If the original surface is not closed, we can distribute its curvature to its boundary to obtain a prescribed surface that is a subset of the Euclidean plane, sphere, or hyperbolic plane. Here, we use a rectangle as an example.

Step 3 is described in section 3.4 in detail with several examples, but we offer the following high level considerations here. The mapping produced by DSRF, although angle-preserving, does not preserve relative area. This can cause details from the original surface to be missed when a uniform sampling of the parametric domain is used to generate element sample points. The goal of adaptive refinement (AR) is to distribute mesh sample points to mitigate this.

Figure 3.4.1 compares quadrilateral surface meshes obtained for an ellipsoid using a nonconformal versus conformal mapping to map a 20×20 grid of square elements to the surface.

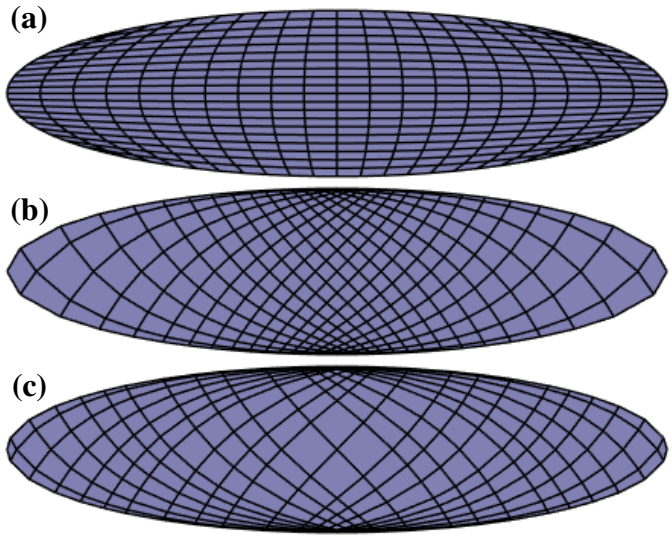


Fig. 3.1. 20×20 structured quadrilateral meshes mapped to an ellipsoid using (a) a nonconformal mapping, (b) a conformal mapping automatically obtained by DSRF, and (c) a conformal mapping using DSRF coupled adaptive refinement to capture more detail in regions of high curvature.

Figure 3.4.2 shows angle histograms for the three meshes in Fig. 3.4.1, demonstrating that the nonconformal mapping preserves corner angles poorly, while the conformal mappings obtained by DSRF achieve good angle preservation.

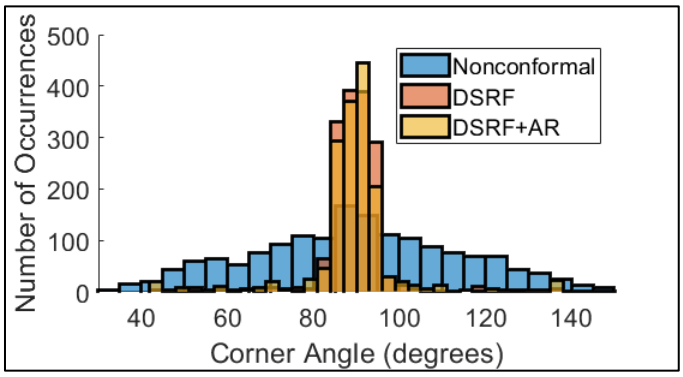


Fig. 3.2. Corner angle histograms for 20×20 structured quadrilateral meshes show in Fig 3.1. Nonconformal mesh has a poor corner angle distribution while meshes obtained using DSRF have distributions concentrated closely around 90 degrees.

Although the nonconformal mapping samples the ellipsoid excellently, it does so while producing poor corner angles for most elements. Meanwhile, the mesh produced with pure DSRF has excellent corner angles, but under-samples the highly curved left and right tips of the ellipsoid, shown Fig. 3.4.1(b). The mesh produced with DSRF and adaptive refinement, on the other hand, maintains the excellent corner angles of the pure DSRF mesh while better distributing corner angles to the left and right ellipsoid tips, shown in Fig. 3.4.1(c).

3.3 Obtaining the Mapping by DSRF

Popularized by its role in Perelman's 2006 proof of the Poincaré conjecture [85], Ricci flow offers a mathematical framework for diffusing irregularities in the metric of a Riemannian manifold. In the context of this work, surface Ricci flow, by the discrete formulation described in [86], allows the generation of a conformal (angle-preserving) mapping between a surface of choice, and a homeomorphic (or non-homeomorphic, given a suitable cut graph) surface of

prescribed Gaussian curvature, here constituting a parametric domain for the mesh and referred to as the prescribed surface. For instance, this allows the NASA almond to be mapped to the unit sphere or cut and mapped to the plane. Information on the prescribed surface can then be conformally mapped back to the original surface. In our application, this information comprises element vertices, and in our higher order cases, element sample points.

We begin with a triangular surface mesh $\Sigma = (V, E, F)$ where V, E, and F are the sets of vertices, edges, and faces composing the mesh, respectively. Here we assume our mesh represents the boundary of a realizable three-dimensional (3D) object, i.e., the surface does not intersect itself, is continuous, and is finite. We refer to this as the initial surface. The initial surface may either be closed or have a boundary (a one-dimensional curve in 3D space) $\partial\Sigma$. We wish to deform this original surface to a much simpler, prescribed surface on which we can easily define and manipulate mesh topology. The prescribed surface has discrete Gaussian curvature given by

$$K(v) = \begin{cases} 2\pi - \sum_{jk} \theta_i^{jk} , & v \notin \partial\Sigma \\ \pi - \sum_{jk} \theta_i^{jk} , & v \in \partial\Sigma \end{cases}, \tag{3.1}$$

where v refers to a given vertex and $\sum_{jk} \theta_i^{jk}$ denotes the sum of all triangle corner angles of which v is a part. Here i denotes the index of v, and θ_i^{jk} denotes the corner angle formed by vertex v and two adjacent (connected by an edge) vertices with indices j and k. We define the Euler characteristic as,

$$X = \mathrm{N_V} - \mathrm{N_E} + \mathrm{N_F} . \tag{3.2}$$

Here $\mathrm{N_V}$, $\mathrm{N_E}$, and $\mathrm{N_F}$ represent the number of vertices, edges, and faces of the surface, respectively. With this, the Gauss-Bonnet theorem asserts,

$$\sum_V K(v) + \epsilon \mathrm{A} = 2\pi \mathrm{X}, \tag{3.3}$$

with A denoting the total surface area of the mesh and the scheme coefficient term ϵ determined by the chosen background geometry, taking a value of 1, 0, or -1, that the discrete Gaussian curvature (3.1) must be conserved during this deformation.

To proceed, we must specify the background geometry. For our purposes and for intuition, we can consider the background geometry as a geometric perspective we choose for a given surface. For instance, consider the approximately spherical surface of the Earth. Locally, any part of the Earth looks like the Euclidean plane, but globally, the Earth has properties not possible from the Euclidean perspective. Consider we are standing on the equator facing north, and we draw two parallel lines, both at right angles to the equator, also facing north. In the Euclidean plane, parallel lines never intersect, yet, consider we extend these two straight lines as far north as possible: they will intersect at the north pole. This quirk, and similar properties, are unique to the spherical geometry of the Earth. It is therefore natural to understand global properties of the Earth's surface from a spherical perspective. This is true even though the Earth is not a sphere (rather an oblate spheroid). This does not prevent us from choosing a Euclidean description of the Earth's surface instead, for instance, a Mercator-projected map. However, we note that all points along the north edge of the Mercator map correspond to just one point on the Earth: the north pole (this is similarly true for the south edge and south pole). By choosing a Euclidean perspective for an inherently spherical surface, we have lost some of the properties of that surface (in this case, uniqueness of the north pole, among others not mentioned) and have introduced singularities at the poles. By the discrete uniformization theorem [84], any surface naturally admits one of three potential background geometries: spherical, Euclidean, or hyperbolic, shown in Fig. 3.3 and corresponding to ϵ values of 1, 0, and -1, respectively. A natural choice in background geometry depends on the topological properties of the object [86]. This can be determined by examining the Euler

characteristic (3.2) of the initial surface and comparing to the Euler characteristic of the prescribed surface.

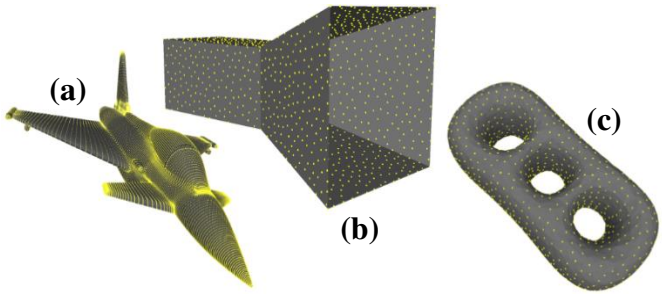


Fig. 3.3. Examples of objects naturally admitting the three background geometries: (a) spherical, (b) Euclidean, and (c) hyperbolic geometry.

To avoid singularities, Euclidean background geometries are typically most useful for closed surfaces containing one hole (homeomorphic to the 1-torus) and having Euler characteristic 0. Due to its simplicity, it is also favorable for open surfaces (those having a boundary and Euler characteristic 1) for our application. Spherical background geometries are naturally applied to closed surfaces containing no holes (those homeomorphic to the sphere), for instance, the NASA almond, with Euler characteristic 2. Hyperbolic background geometries, meanwhile, are useful for more-complicated closed surfaces containing several holes (homeomorphic to n-tori) and having Euler characteristic $-2n$ for integer n 1 or higher. A singularity will occur when our original surface and prescribed surface have different natural background geometry. Note that closed surfaces can alternatively be cut (specifying a topological boundary without removing faces or vertices) to produce an open surface with Euler characteristic 1. By Ricci flow, such a surface can be mapped to the Euclidean plane, offering attractive choices for the seed mesh. Such cuttings will admit singularities that in practice do not substantially impact the quality of the mesh, introducing perhaps a handful of elements with poor corner angles, so long as the arc length along the cut is nonzero on both the original and prescribed surface. This is not true, for instance, in the case of the sphere and Mercator projection, which can be considered to involve an infinitesimal cut of zero

arc length at each pole of the sphere to produce a cylinder (which is then cut once more to produce a plane). The cuts of zero arc length at the poles introduces the substantial degeneracy of the Mercator projection at the poles.

Figure 3.4(a) shows an initial surface, in this case a fighter jet. Here we chose to apply a cut to the surface along its plane of symmetry and prescribe a flat, rectangular section of the Euclidean plane as the prescribed surface. Euclidean Ricci flow was chosen for this task, and the total discrete Gaussian curvature (3.1) was divided evenly among four points along the boundary to satisfy the Gauss-Bonnet theorem (3.2) for the

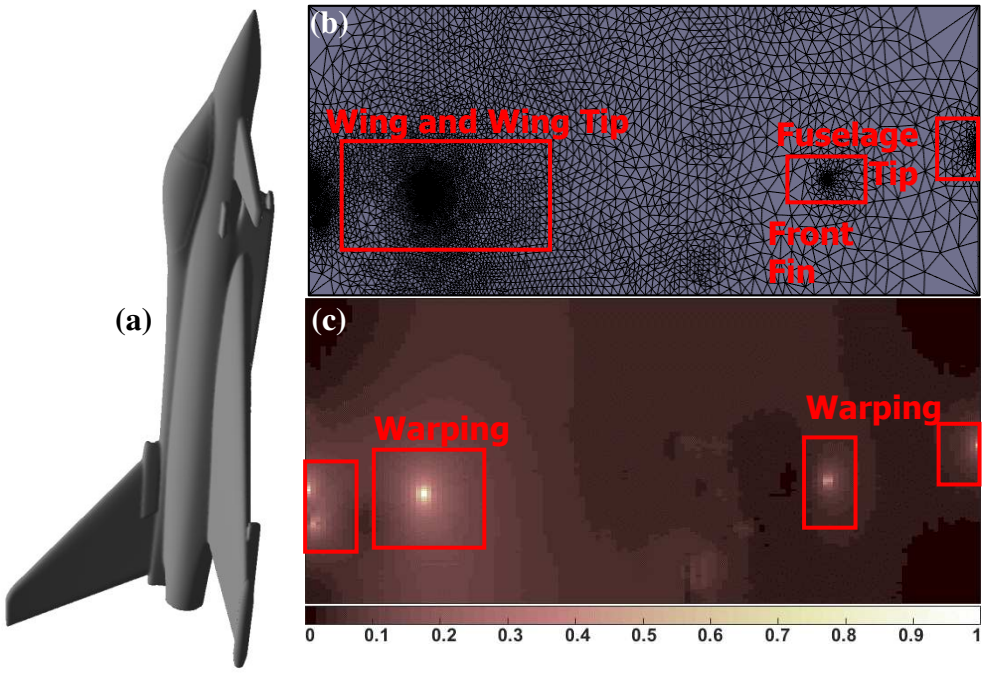


Fig. 3.4. (a) Initial fighter jet surface. (b) Triangle mesh of fighter jet cut and conformally flattened to the plane with highly warped areas boxed in red. (c) Refinement indicator (normalized) for each quadrilateral element demonstrating high degree of warping at fighter jet fin, wing, and fuselage tip when uniform sampling is used.

chosen prescribed surface. Figure 3.4(b) shows the flattening of the initial fighter jet triangle mesh to the parametric domain by solution of the discrete surface Ricci flow system with curvature concentrated at four boundary vertices to produce corners. The parent jet mesh contained

115,967 triangles, and the DSRF took 14.3 seconds to compute on an i7 3770k at 3.50 GHz with a fully parallelized implementation.

Once a background geometry and prescribed surface are selected, a target curvature $\bar{K}(v)$ is chosen constrained by (3.2). For instance, using Euclidean background geometry on an open surface and mapping to the Euclidean plane, the total curvature can be allocated entirely to four boundary vertices. These vertices will then become the vertices of a rectangular image of the original surface in the Euclidean plane after mapping. Once the target curvature is selected, the discrete surface Ricci flow system

$$\frac{du_i(t)}{dt} = \bar{K}_i - K_i(t) \tag{3.4}$$

can be solved by any number of standard numerical methods to obtain the final conformal mapping between surfaces. The above semi-discrete form retains continuous time, t, which in practice is discretized into a finite set of iterations. A nonlinear equation, (3.4) must be solved iteratively. For a thorough background on computing the Hessian of the Ricci energy for this system and applying it through Newton's method to obtain the final mapping, see [84]. We give a brief overview here.

To compute and solve the DSRF system, we must first define several discrete parameters and structures over Σ. We first define a circle packing metric. We associate with each vertex $v_i \in V$ nonnegative radius γ_i corresponding to a circle centered on v_i. We also define a real-valued discrete conformal structure coefficient on E denoted η. Together with the scheme coefficient from (3.2), our circle packing metric is then defined by the tuple $(\Sigma, \gamma, \eta, \epsilon)$, from which we can then determine any edge length $l_{ij} \in E$ between vertices $v_i, v_j \in V$. Defining the discrete conformal factor u_i, dependent on the background geometry,

$$u_i = \begin{cases} \log \gamma_i & \mathbb{E}^2 \\ \log \tanh 0.5\gamma_i & \mathbb{H}^2 \\ \log \tan 0.5\gamma_i & \mathbb{S}^2 \end{cases} \tag{3.5}$$

where $\mathbb{E}^2$, $\mathbb{H}^2$, and $\mathbb{S}^2$ denote Euclidean, hyperbolic, and spherical background geometry, respectively, we can compute l_{ij} in general by

$$\begin{cases} l_{ij}^2 = 2\eta_{ij}e^{u_i+u_j} + \varepsilon_i e^{2u_i} + \varepsilon_j e^{2u_j} & \mathbb{E}^2 \\ \cosh l_{ij} = \frac{4\eta_{ij}e^{u_i+u_j}+(1+\varepsilon_i e^{2u_i})(1+\varepsilon_j e^{2u_j})}{(1-\varepsilon_i e^{2u_i})(1-\varepsilon_j e^{2u_j})} & \mathbb{H}^2 \\ \cos l_{ij} = \frac{-4\eta_{ij}e^{u_i+u_j}+(1-\varepsilon_i e^{2u_i})(1-\varepsilon_j e^{2u_j})}{(1+\varepsilon_i e^{2u_i})(1+\varepsilon_j e^{2u_j})} & \mathbb{S}^2 \end{cases} \tag{3.6}$$

The ε coefficients and range of the conformal structure coefficient for a few common circle packing schemes [84] are defined in Table 3.4.1. We use inversive distance circle packing for the results presented in this chapter.

TABLE 3.1. Range of conformal structure coefficient and ε coefficient values for common circle packing schemes.

Scheme	η_{ij}	ε_i	ε_j
Thurston's	[0,1]	+1	+1
Tangential	+1	+1	+1
Virtual radius	> 0	-1	-1
Inversive Distance	> 0	+1	+1

With a circle packing scheme defined and chosen, we can now solve the DSRF system (3.4) iteratively as follows. At every iteration we begin by computing all circle radii γ_i from the discrete conformal factor (3.5). Following this, we use η and γ values to compute all edge lengths by application of (3.6). From the edge lengths, we compute the corner angles θ_i^{jk} from the cosine law appropriate to the chosen background geometry

$$\begin{cases} l_k^2 = \gamma_i^2 + \gamma_j^2 - 2l_i l_j \cos\theta_k & \mathbb{E}^2 \\ \cosh l_k = \cosh l_i \cosh l_j - \sinh l_i \sinh l_j \cos\theta_k & \mathbb{H}^2 \\ \cos l_k = \cos l_i \cos l_j - \sin l_i \sin l_j \cos\theta_k & \mathbb{S}^2 \end{cases} \tag{3.7}$$

and subsequently the vertex curvature K from the angle deficit (3.1). We then compute the Hessian matrix, H, from the local (face) Hessian matrices:

$$\frac{\partial(\theta_i,\theta_j,\theta_k)}{\partial(u_i,u_j,u_k)} = -\frac{1}{2A} L\Theta L^{-1} D \tag{3.8}$$

where

$$L = \begin{bmatrix} s(l_i) & 0 & 0 \\ 0 & s(l_j) & 0 \\ 0 & 0 & s(l_k) \end{bmatrix} \tag{3.9}$$

$$A = \sin\theta_i s(l_j) s(l_k) \tag{3.10}$$

and

$$D = \begin{bmatrix} 0 & \tau(i,j,k) & \tau(i,k,j) \\ \tau(j,i,k) & 0 & \tau(j,k,i) \\ \tau(k,i,j) & \tau(k,j,i) & 0 \end{bmatrix} \tag{3.11}$$

Note, as in [84], we define here for convenience between background geometries

$$s(x) = \begin{cases} x & \mathbb{E}^2 \\ \sinh x & \mathbb{H}^2 \\ \sin x & \mathbb{S}^2 \end{cases} \tag{3.12}$$

and

$$\tau(i,j,k) = \begin{cases} \frac{1}{2}(l_i^2 + \epsilon_j \gamma_j^2 + \epsilon_k \gamma_k^2) & \mathbb{E}^2 \\ \cosh l_i \cosh^{\epsilon_j} \gamma_j - \cosh^{\epsilon_k} \gamma_k & \mathbb{H}^2 \\ \cos l_i \cos^{\epsilon_j} \gamma_j - \cos^{\epsilon_k} \gamma_k & \mathbb{S}^2 \end{cases} \tag{3.13}$$

Finally, we solve the linear system

$$H\delta u = \bar{K} - K \tag{3.14}$$

for δu, updating the discrete conformal factor to be used in the next iteration as

$$u \leftarrow u - \delta t \delta u. \tag{3.15}$$

This process is iterated until a convergence criterion is met, most simply until the maximum difference between the current and target discrete curvature falls below some threshold, i.e., until

$$\max_i |\overline{K}_i - K_i| < threshold \tag{3.16}$$

From the final η, γ, u, and θ values, we can compute the final vertex locations in the target domain by flattening from a seed face as in [86]. For additional discussion of convergence rate, stability, and modifications to improve the robustness of the above approach, see [84, 87, 88].

Once the locations of all vertices are known in the parametric domain, any point within that domain can be mapped back to the initial surface using barycentric coordinates, defining a piecewise-linear inverse of M. A point p with parametric coordinate (u_0, w_0) in the parametric domain is found to lie in face f. If f has parametric vertices v_1, v_2, and v_3, each with parametric coordinate of form (u_i, w_i) and nonparametric coordinate of form (x_i, y_i, z_i), the image of p on the original surface, here denoted p' with coordinate (x_0, y_0, z_0), is given by

$$(x_0, y_0, z_0) = \sum_{i=1}^{3} k_i (x_i, y_i, z_i), \tag{3.17}$$

where addition is understood component-wise and the Barycentric coordinates are given by

$$k_1 = \frac{\begin{vmatrix} u_3 - u_0 & v_3 - v_0 \\ u_2 - u_3 & v_2 - v_3 \end{vmatrix}}{s} \tag{3.18a}$$

$$k_2 = \frac{\begin{vmatrix} u_1 - u_0 & v_1 - v_0 \\ u_3 - u_1 & v_3 - v_1 \end{vmatrix}}{s} \tag{3.18b}$$

$$k_3 = \frac{\begin{vmatrix} u_2 - u_0 & v_2 - v_0 \\ u_1 - u_2 & v_1 - v_2 \end{vmatrix}}{s} \tag{3.18c}$$

with scaling factor

$$s = \begin{vmatrix} u_1 - u_2 & v_1 - v_2 \\ u_3 - u_1 & v_3 - v_1 \end{vmatrix} \tag{3.19}$$

By choosing a simple prescribed surface on which element sample points in the seed mesh can be easily defined and manipulated, we exert a high degree of control over the resultant re-mapping. However, the mapping produced by Ricci flow preserves only angles, not relative areas, so simply mapping a uniform grid of sample points from the prescribed surface to the surface of

choice produces poor results for our application, leading to wide discrepancies in mesh fidelity between minimally-warped and highly-warped portions of the resulting surface mesh, as illustrated in Fig. 3.4(c). This is the motivation for beginning with an initial seed mesh and iteratively refining, an approach that allows the unknown degree of local warping to be compensated for adaptively.

3.4 Iterative Adaptive Refinement

3.4.1 Overview

We describe here how to construct a mesh informed by some refinement indicator, in general motivated by either geometric error or numerical solution error estimates. Focusing on the geometric properties of the method, we offer specific examples of refinement indicators to reduce geometric error but maintain generality for easy application of the method to adaptive refinement (AR) based instead on solution error.

With an initial triangular surface mesh $\Sigma = (V, E, F)$, a mapping of this surface to the parametric domain, $\bar{\Sigma} = (\bar{V}, \bar{E}, \bar{F})$, and an associated map $M: \bar{\Sigma} \to \Sigma$ from the parametric domain to the initial surface, we wish to construct a new surface mesh, $\tilde{\Sigma} = (\tilde{V}, \tilde{E}, \tilde{F})$, of arbitrary type. Beginning with a seed mesh $\tilde{\Sigma}_0$ of the chosen mesh type in the parametric domain, we must define a refinement indicator, $R_n(\tilde{\Sigma}, M)$ and a refinement method $Q(R_n, \tilde{\Sigma}_n) = \tilde{\Sigma}_{n+1}$. We may then iterate Q on $\tilde{\Sigma}_0$ N times, updating R_n at each iteration, to produce a final surface mesh $\tilde{\Sigma} = \tilde{\Sigma}_N$. N may be user-defined or may be a function of $\tilde{\Sigma}$, e.g., some stop criterion like total element count or maximum element size. We give examples of such R_n and Q for a variety of common mesh types and offer suggestions to extend these to other mesh types. We also define the simplest seed mesh for each mesh type covered if Euclidean DSRF is used to map to the plane. Note that the refinement methods described here have linearithmic time complexity with the number of refinement iterations.

3.4.2 Continuous Structured Quadrilateral Meshes

Here we give an example of a refinement indicator and associated refinement method for the continuous quadrilateral case using Euclidean DSRF. For every edge $e \in \tilde{E}$, we find the Euclidean distance, d, between its endpoints, (x_1, y_1, z_1) and (x_2, y_2, z_2) in the nonparametric domain

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \tag{3.20}$$

We assign such a distance to each $e \in \tilde{E}$, constituting $R_n(\tilde{\Sigma}, M)$ with domain $\tilde{E}$.

Starting from a seed mesh in the parametric domain consisting of one quadrilateral element aligned with the parametric coordinate axes as in Fig. 3.5(a), we find the row and column containing the edge with highest d for the vertical ($w_1 = w_2$) and horizontal ($u_1 = u_2$) edges, respectively. We then subdivide the appropriate row and column in half in the parametric domain, taking one row to two rows and one column to two columns. This can be repeated N times and constitutes a simple and effective example of one possible $Q\left(R_n, \tilde{\Sigma}_n\right) = \tilde{\Sigma}_{n+1}$. Figures 3.5(b)-(d) show this subdivision process in the parametric domain for the fighter jet mesh from Fig. 3.4(a) for $N = 2$, $N = 10$, and $N = 20$ iterations, respectively.

Note that the example Q given here constitutes only a simple and informative refinement method to produce a structured quadrilateral mesh. If unstructured meshes are permissible, any existing quadrilateral mesh refinement method could be used in the parametric domain, with the resulting mesh then mapped back to the original surface conformally. For instance, node-placement schemes like those in [73], [74], and [89] could be adapted to serve as the refinement method.

If higher order elements are chosen, we may subsample $\tilde{\Sigma}_N$ in the parametric domain to produce the necessary interpolation nodes. For instance, if high order elements requiring a grid of $k*k$ nodes per element are chosen, we may split each row and column of $\tilde{\Sigma}_N$ k-2 times to obtain

the necessary sample density. In this chapter, to improve accuracy and maintain adaptivity for the given examples, we do this implicitly. If an $L*L$ grid of higher order elements, each requiring $k*k$ nodes, is chosen, we define N to be $L*(k\text{-}1)\text{-}1$ to obtain the necessary sample points for all elements. For the given examples, we order higher order sample points for quadrilateral elements as defined in [1].

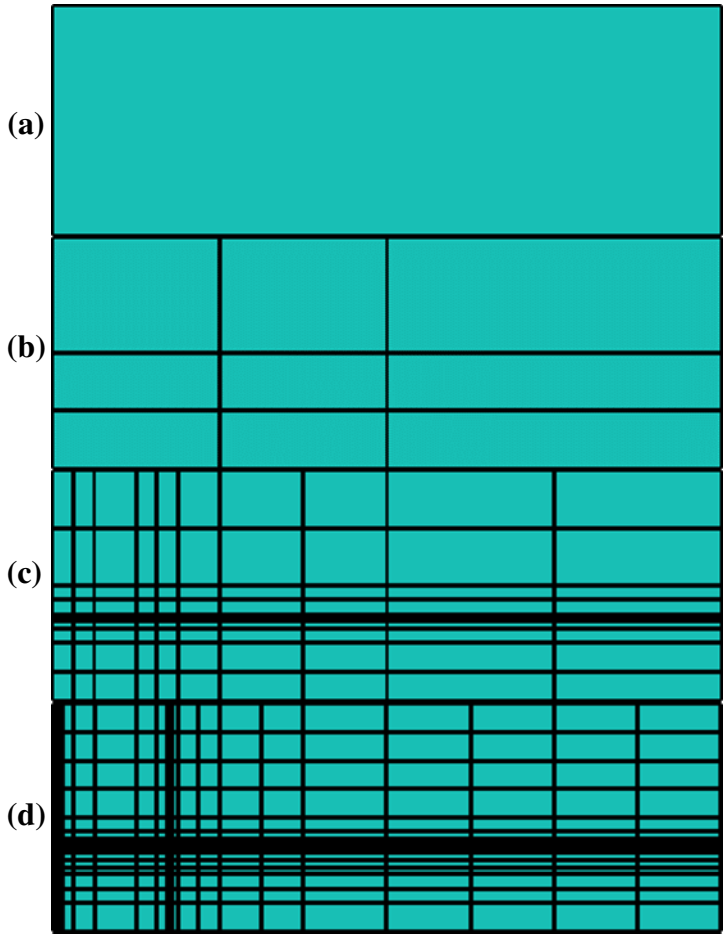


Fig. 3.5. Iterative adaptive refinement in the parametric domain for a continuous curved quadrilateral mesh of a fighter jet in Fig. 3.4(a) to increase mesh quality intelligently, with more elements being allocated to high density areas in the parametric domain, leading to a more-uniform final jet mesh: (a) seed mesh with one element aligned with coordinate axes in parametric domain, (b) refined mesh with $N = 2$ iterations, (c) refined mesh with $N = 10$ iterations, and (d) refined mesh with $N = 20$ iterations.

3.4.3 Continuous Triangular Meshes

We now give a similar example for the continuous triangular case using Euclidean DSRF. For every edge $e \in \tilde{E}$, we again find the Euclidean distance, d, between its endpoints in the nonparametric domain (3.20). We assign such a distance to each $e \in \tilde{E}$, again constituting $R_n(\tilde{\Sigma}, M)$ with domain $\tilde{E}$. For $e \in \tilde{E}$ with maximum d and parametric endpoints (u_1, w_1) and (u_2, w_2), we compute the parametric midpoint p as

$$p = \left(\frac{u_1+u_2}{2}, \frac{v_1+v_2}{2}\right), \tag{3.21}$$

and include it in the set of existing vertices in $\tilde{\Sigma}$. We then update a Delaunay tessellation of this augmented $\tilde{V}$ in the parametric domain to update $\tilde{E}$ and $\tilde{F}$ to include the added vertex.

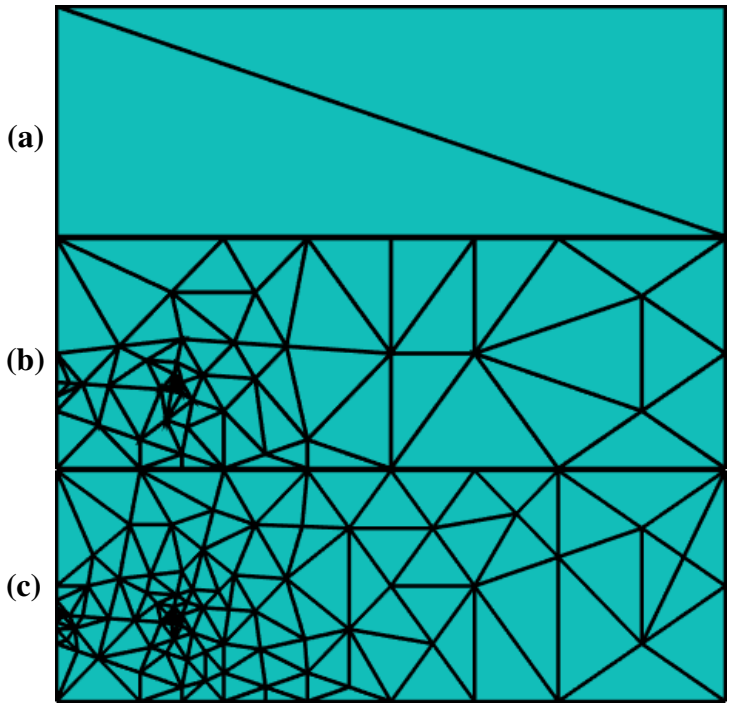


Fig. 3.6. Iterative adaptive refinement in the parametric domain for a continuous curved triangular mesh of a fighter jet in Fig. 3.4(a): (a) seed mesh with two elements in the parametric domain, (b) refined mesh with $N = 50$ elements, and (c) refined mesh with $N = 100$ iterations.

This can be repeated N times and constitutes a possible $Q\left(R_n, \tilde{\Sigma}_n\right) = \tilde{\Sigma}_{n+1}$. Figures 3.6(a)-(c) show similar parametric-domain adaptive refinement results for N = 2, 50, and 100 iterations, respectively.

As in the continuous quadrilateral case, extension to higher order elements is simple, requiring only additional sampling of the mapping at interpolation nodes. Interpolation nodes can be chosen for each element in the parametric domain and mapped by M as before. For the examples given in this chapter, we define higher order sample points for each triangle as in [1].

3.4.4 Discontinuous Quadrilateral Meshes

To define a suitable refinement indicator in the discontinuous quadrilateral case using Euclidean DSRF, we again compute (3.20) for all $e \in \tilde{E}$, beginning from the seed mesh defined for the continuous quadrilateral case and shown in Fig. 3.5(a). For e with maximum d, we split an adjacent face in the direction perpendicular to such e, introducing two new vertices and one new edge. Note that, although an edge may have two adjacent faces, it is in practice inconsequential which face is split on a given iteration, as the unsplit face is guaranteed to be refined on a subsequent (typically the next) iteration, dependent on the number of edges with identical maximum d. Such face splitting constitutes a possible $Q\left(R_n, \tilde{\Sigma}_n\right) = \tilde{\Sigma}_{n+1}$ and introduces one new face per iteration. Additional constraints could be imposed on which element to split at each iteration to satisfy potential requirements of specific discontinuous quadrilateral codes. For instance, if an implementation requires that one edge joins to at most two, conditions that would violate this if refined could be detected at each iteration, with e with the next highest d chosen instead. Figure 3.7 demonstrates this refinement method for various N on the jet fighter mesh with no such additional constraints. For extension to higher order, we sample on a quadrilateral-by-

quadrilateral basis and again use the sampling convention defined for higher order quadrilaterals in [1].

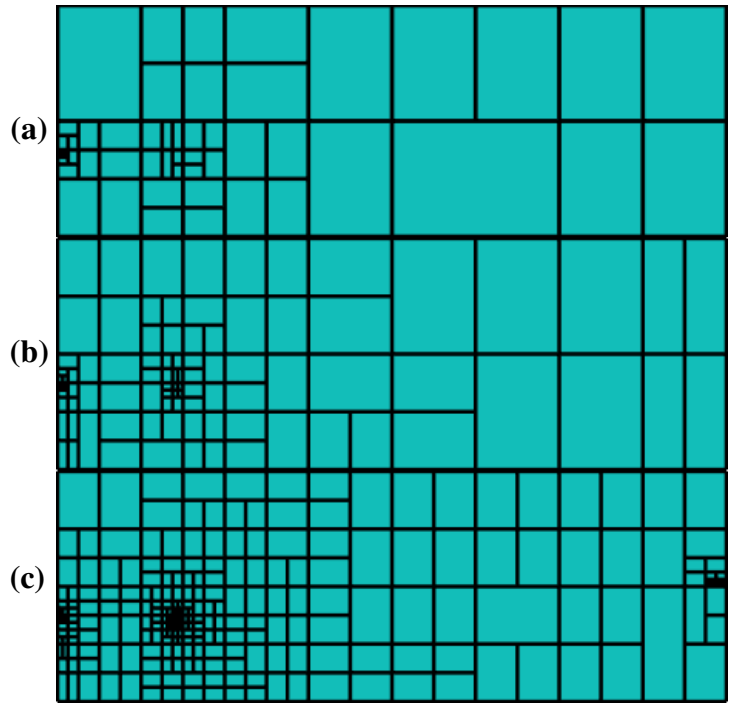


Fig. 3.7. Adaptive refinement for discontinuous quadrilaterals in the parametric domain: (a) N = 50 , (b) N = 100, and (c) N = 300 iterations.

3.4.5 Continuous Unstructured Quadrilateral Meshes

Here we demonstrate how DSRF-AR can be used to convert an existing 2D meshing technique into a 3D surface meshing technique, in this case for continuous structured quadrilateral meshes. We begin with the 2D continuous quadrilateral subdivision method described in [89]. The method in [89] first refines elements uniformly by splitting each refined element into a 3×3 grid of quadrilaterals. A set of 4 irregular subdivision patterns is then applied to adjacent elements to repair any discontinuities introduced during refinement. To apply this method to 3D surfaces using our DSRF-AR approach, we again compute (3.20) for all $e \in \tilde{E}$. For e with maximum d, we split an adjacent face into a 3×3 grid of quadrilaterals. Any neighboring faces of the refined face are also refined uniformly if needed to maintain the criterion that no edge joins to more than 3. This

is iterated until some stop criterion is met, after which the irregular subdivision templates from [89] are applied to repair all discontinuities. Note that no irregular element is ever subdivided, as this would lead to unbounded mesh quality deterioration. This constitutes another possible $Q(R_n, \tilde{\Sigma}_n) = \tilde{\Sigma}_{n+1}$. Figure 3.4.8 demonstrates this refinement method on the fighter jet mesh for several *N*. As in previous quadrilateral examples, extension to higher order constitutes quadrilateral-by-quadrilateral resampling using the convention defined in [1]. Note that, since [89] assumes square elements to maintain reasonable corner angles in the irregular subdivision templates, we use a different seed mesh here, splitting the rectangular parametric domain into approximately square elements (here 3). This can be automated by comparing the width and height of the rectangular parametric domain, subdividing it appropriately.

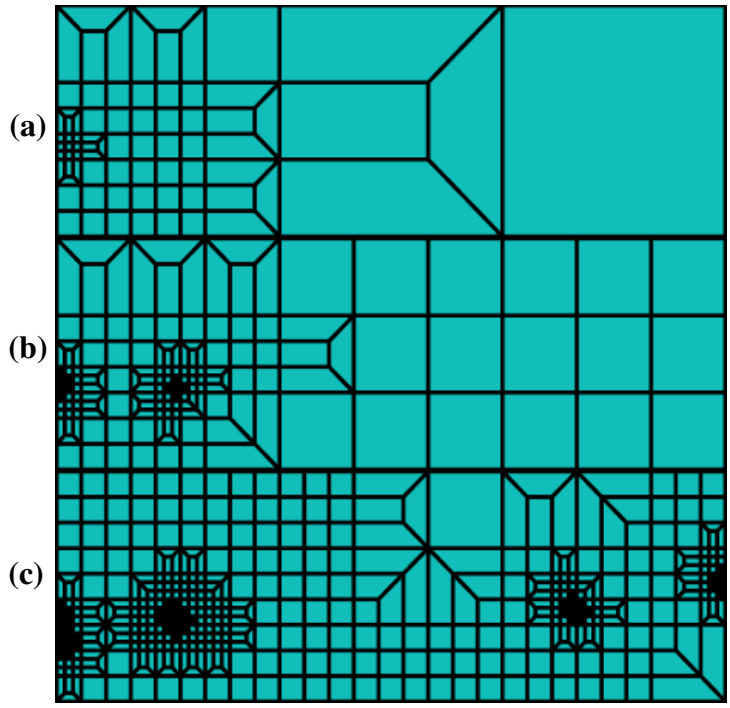


Fig. 3.8. Adaptive refinement for continuous unstructured quadrilaterals in the parametric domain: (a) *N* = 5 (b) *N* = 10, and (c) *N* = 50 iterations.

3.4.6 Generalization to Mesh Types Not Covered

Although we have covered three common mesh types, we by no means wish to limit the applicability of DSRF with adaptive refinement to production of only continuous quadrilateral, continuous triangular, discontinuous quadrilateral meshes. We hope the given examples offer clear guidance for generalizing to other mesh types, but we additionally give recommendations for generalization by offering a few notes on surface mesh features and pertinent aspects of DSRF with adaptive refinement not otherwise covered in this chapter.

We have chosen in this chapter to use the Euclidean case of DSRF and map to a rectangular subspace of the Euclidean plane. This choice was made for simplicity not only in the intuitiveness of Euclidean space, but also the simplicity of defining and visualizing seed meshes on a flat rectangular domain. In practice, spherical and hyperbolic DSRF have excellent advantages when applied to complicated topologies to reduce or eliminate singularities. As mentioned, these singularities often have only minor impacts on the conformality of the computed mapping but tend to produce some singularity-adjacent elements for which angles are poorly preserved when mapping from the parametric domain. For most applications, this is not an issue, as the vast majority (> 99%) of elements are typically well-mapped. However, for methods extremely sensitive to even small proportions of poorly shaped elements, we suggest avoiding singularities by using spherical or hyperbolic DSRF where needed.

As stated, we chose to map to a rectangle in the Euclidean plane for the given examples. In practice, other regions and other spaces can be chosen, and indeed can produce superior results depending on the geometry of Σ, the shape of its boundary, and its topology. The difficulty in choosing more-complicated prescribed surfaces is definition of the seed mesh. For other subsets of the Euclidean plane, we recommend using an existing 2D mesher for the chosen mesh type to

produce a coarse seed mesh of the prescribed surface. If mapping to the unit sphere using spherical DSRF, we recommend choosing a spherical polyhedron of the appropriate element type as the seed mesh. If using hyperbolic DSRF, we recommend choosing a polygonal tiling of a region in the Poincaré disk.

It is furthermore crucial to choose $R_n(\tilde{\Sigma}, M)$ and especially $Q\left(R_n, \tilde{\Sigma}_n\right) = \tilde{\Sigma}_{n+1}$. to prevent the creation of malformed elements during refinement in the parametric domain. "Malformed" depends on the mesh type and user application, but typically is related to the regularity of corner angles and local Jacobian within and between elements. As $M(p) = p'$ is conformal, malformed elements in the parametric domain become malformed elements in the final surface mesh.

3.5 Results and Discussion

3.5.1 Example Meshes Produced by DSRF with AR

Here we demonstrate meshing by DSRF with adaptive refinement for the well-known NASA almond model and a complicated fighter jet model. Higher order elements were reconstructed using Lagrange interpolation as in [1] or cubic spline interpolation of the sample points on an element-by-element basis. The same initial Σ triangular meshes were used for all cases (one for the almond, one for the fighter jet). The DSRF was computed on these initial meshes and used to generate one $M(p) = p'$ for each model. The parametric domain adaptive refinement methods described previously were then iterated through these mappings to produce a variety of surface meshes shown in Figs. 3.9-3.20. We show the robustness of the proposed technique to recreate complicated surfaces for arbitrary mesh types with arbitrary element counts and orders, for instance accurately representing the fighter jet model with as few as 32 elements. We are not aware of any other meshing technique that can reliably produce such large-domain meshes.

Figure 3.9(a) shows a high-resolution higher order continuous quadrilateral surface mesh generated using DSRF with the refinement scheme outlined in section 3.4.2.

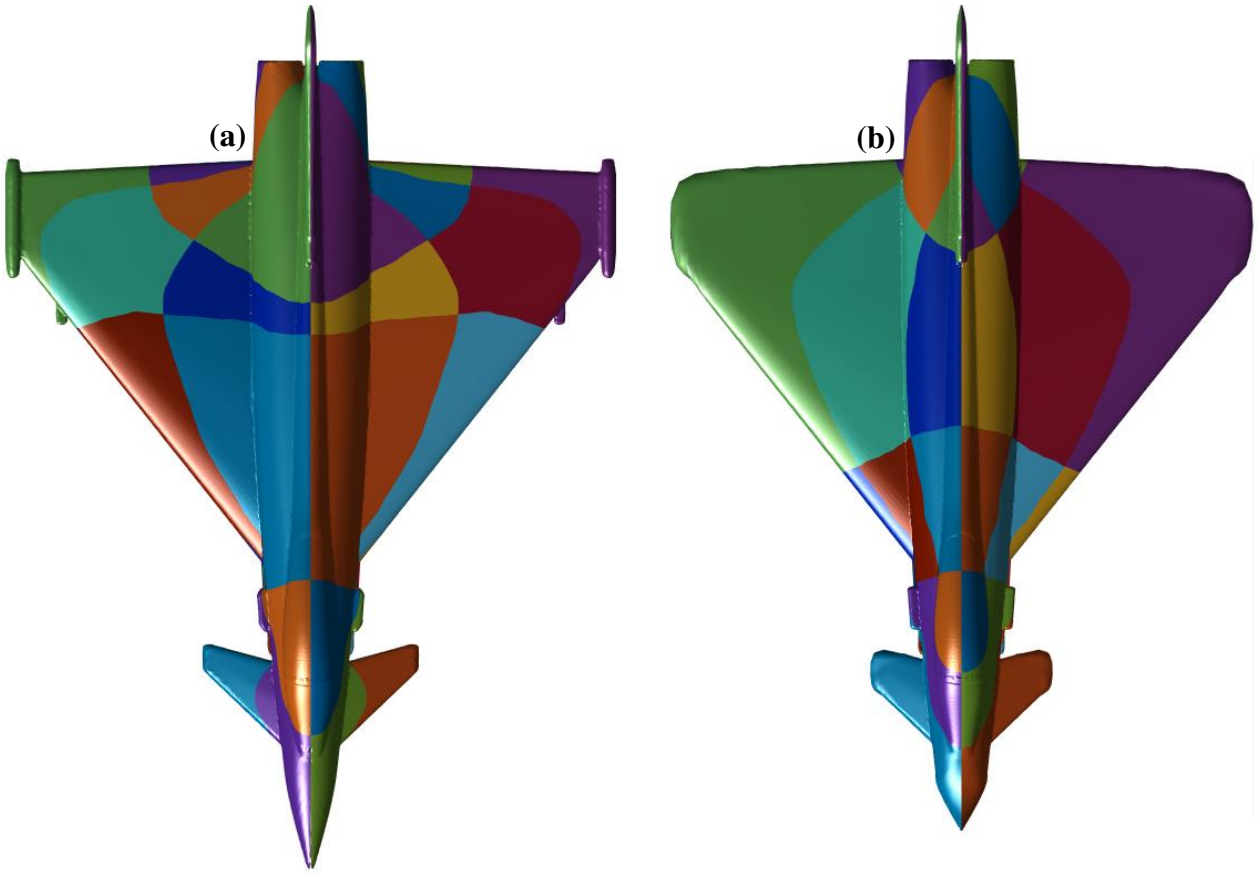


Fig. 3.9. Comparing effects of adaptive iterative refinement vs. uniform sampling on mesh quality: (a) continuous quadrilateral mesh of a fighter jet in Fig. 3.4(a) with 32 64^{th}-order elements using DSRF with iterative adaptive refinement outlined in Fig. 3.5 and (b) the same using uniform sampling.

Figure 3.9(b) shows the equivalent surface mesh instead using uniform sampling in the parametric domain. All parameters including element count, element order, Σ, and M were identical between Figs. 3.9(a) and 3.9(b). Spline interpolation was chosen in both cases. Extreme loss of fidelity can be seen around the fuselage tip and wing tips in the uniformly sampled case, these details meanwhile excellently captured in the adaptive case. This shows not only the importance but also the effectiveness of the proposed adaptive sampling methods for accurately capturing detail in the desired model.

Figure 3.10 shows the same mesh as Fig. 3.9(a) from an oblique angle, making the high fidelity with which the adaptive sampling technique captures fine detail in the initial surface apparent. A comparison between this higher order continuous quadrilateral mesh and the 1st-order triangular mesh (chosen for Σ and shown in Fig. 3.4(a)) shows the near perfection with which this instance of $\tilde{\Sigma}$ recreates the original surface. A similar result is shown in Fig. 3.11 for the NASA almond, here using 32 16th-order continuous quadrilateral elements with Lagrange interpolation. The parent almond mesh contained 2,023 triangles and the DSRF took 0.168 seconds to compute. Adaptive sampling was also used for Fig. 3.11 as outlined in section 3.4.2. Note that, for most practical use cases, such large, curved elements would be supported by extremely high-order current expansions, most importantly to compensate for their large electrical size.

Figures 3.12 and 3.13 show 1st-order discontinuous quadrilateral meshes generated using the technique outlined in Section 3.4.4. Figure 3.12 shows the fighter jet model recreated using 6490 1st-order elements, while Fig. 3.13 shows the NASA almond featuring 2000 elements of the same type. In both cases, the original surface is well-reconstructed.

Figures 3.14 and 3.15 show higher-order analogues of Figs. 3.12 and 3.13, now using 300 30th-order and 300 10th-order discontinuous quadrilateral elements, respectively. Spline interpolation was used in both cases. Note that roughness present on the almond surface in Fig. 3.15 is not an artifact of the proposed meshing technique, but rather shows an accurate recreation of roughness due to 1st-order triangular facets in the original almond mesh chosen as Σ.

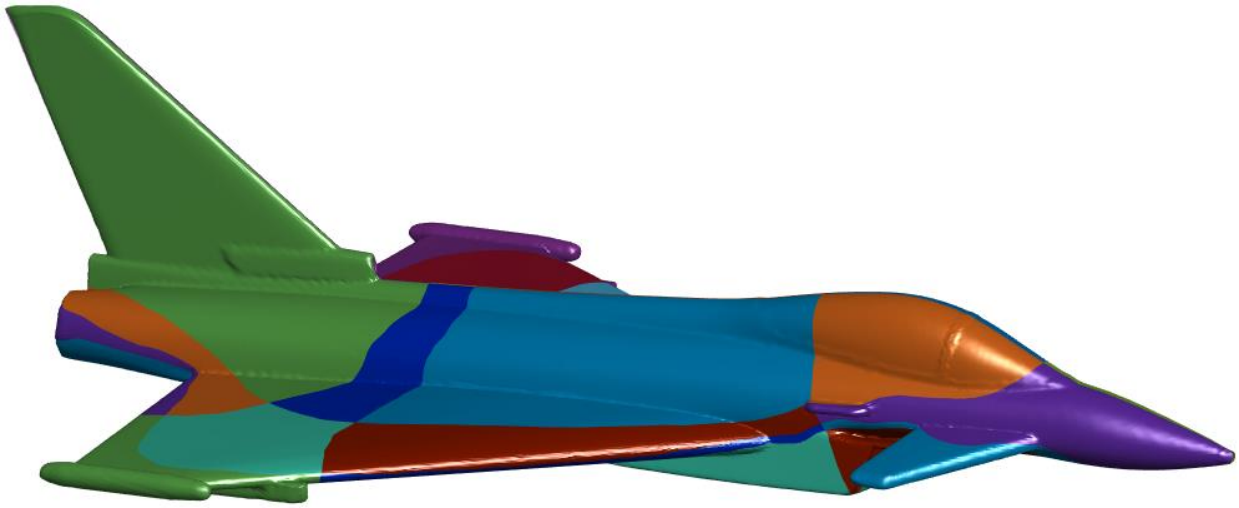


Fig. 3.10. Fighter jet model featuring as few as 32 64^{th}-order quadrilateral elements in Fig. 3.9(a) viewed from oblique angle. Note excellent curvature/detail modeling with hyper-large hyper-curved quadrilateral patches.

Fig. 3.11. NASA almond model using adaptive refinement from Fig. 3.5 with only 32 16^{th}-order continuous quadrilateral elements.

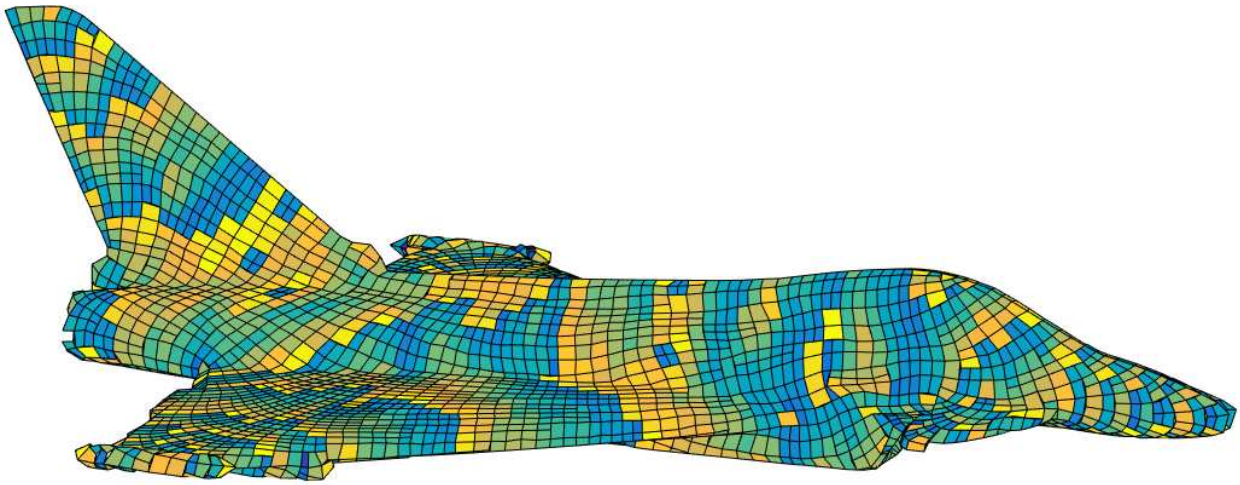


Fig. 3.12. Fighter jet model constructed from 6490 1st-order discontinuous quadrilateral elements using iterative adaptive refinement from Fig. 3.7.

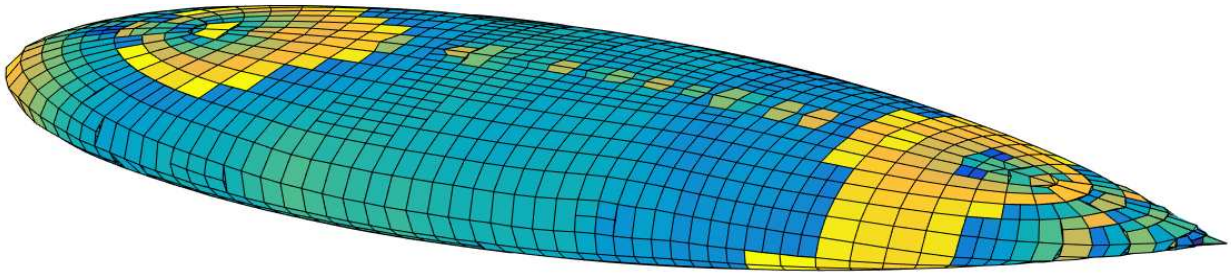


Fig. 3.13. NASA almond model featuring 2000 1st-order discontinuous quadrilateral elements obtained by the adaptive refinement technique from Fig. 3.7.

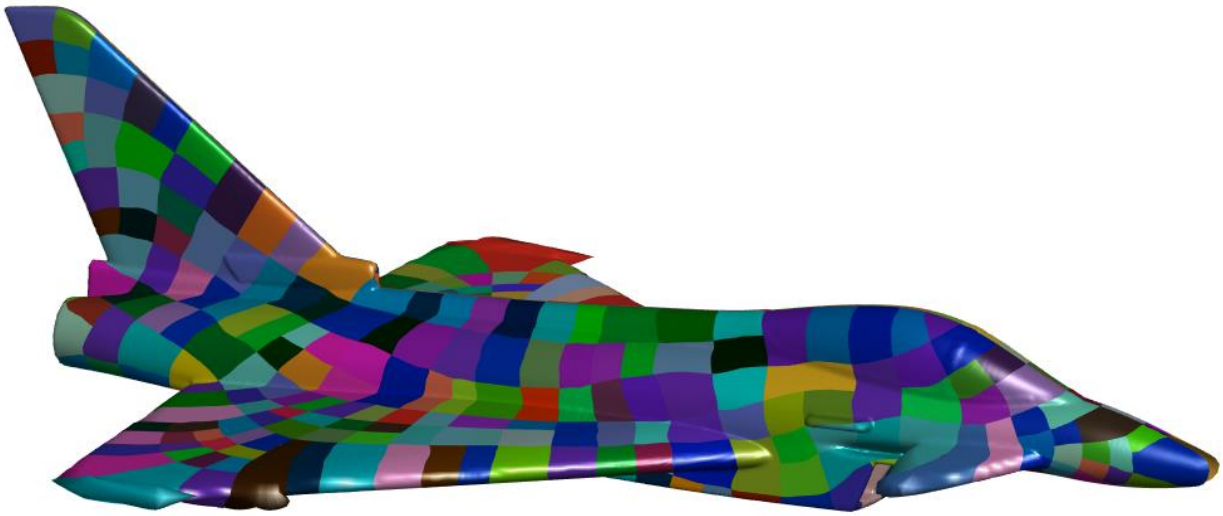


Fig. 3.14. Fighter jet model composed of 300 30^{th}-order discontinuous quadrilateral elements by the technique in Fig. 3.7.

Figure 3.16 shows a low-resolution meshing of the fighter jet model using 2898 first-order triangular elements. Adaptive sampling was used as outlined in section 3.4.3. Despite the low element count and lowest possible geometric order, the model is well represented at coarse-scale, showing that the proposed method works well even as a first-order triangular mesher.

Figures 3.17 and 3.18 show higher order triangular meshes for the fighter jet and almond, respectively, using the technique outlined in Section 3.4.3. The fighter jet was meshed using 3702 10^{th}-order elements interpolated by cubic spline, while the almond was meshed using 1098 10^{th}-order elements interpolated using Lagrange polynomials. We see excellent fidelity in both cases. Note that roughness from facets in Σ can again be seen in Fig. 3.18, like Fig. 3.15.

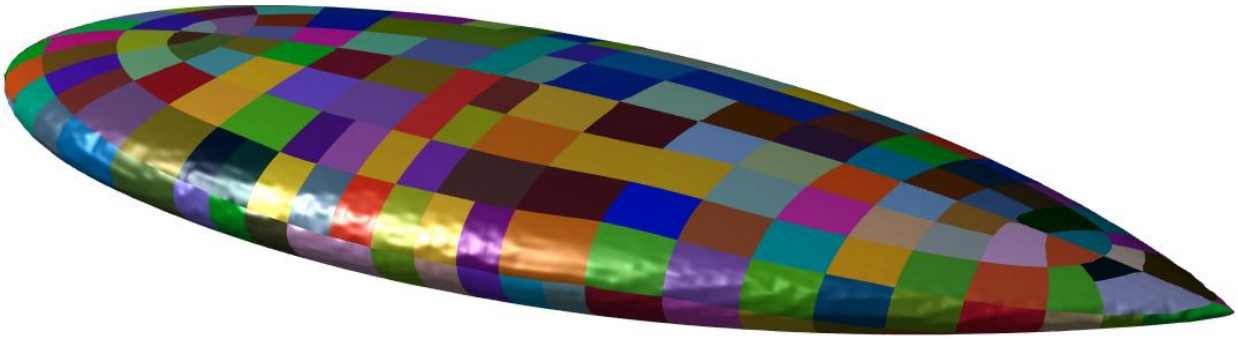


Fig. 3.15. NASA almond model using refinement from Fig. 3.7 with 300 10^{th}-order discontinuous quadrilateral elements.

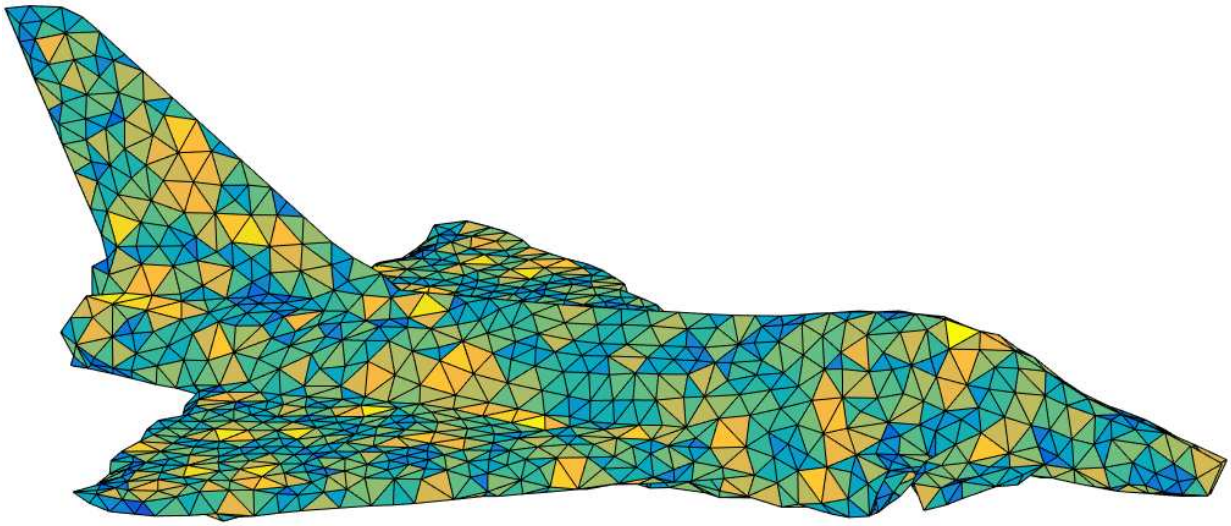


Fig. 3.16. Low-resolution fighter jet model using 2898 1^{st}-order continuous triangular elements with adaptive refinement as in Fig. 3.6.

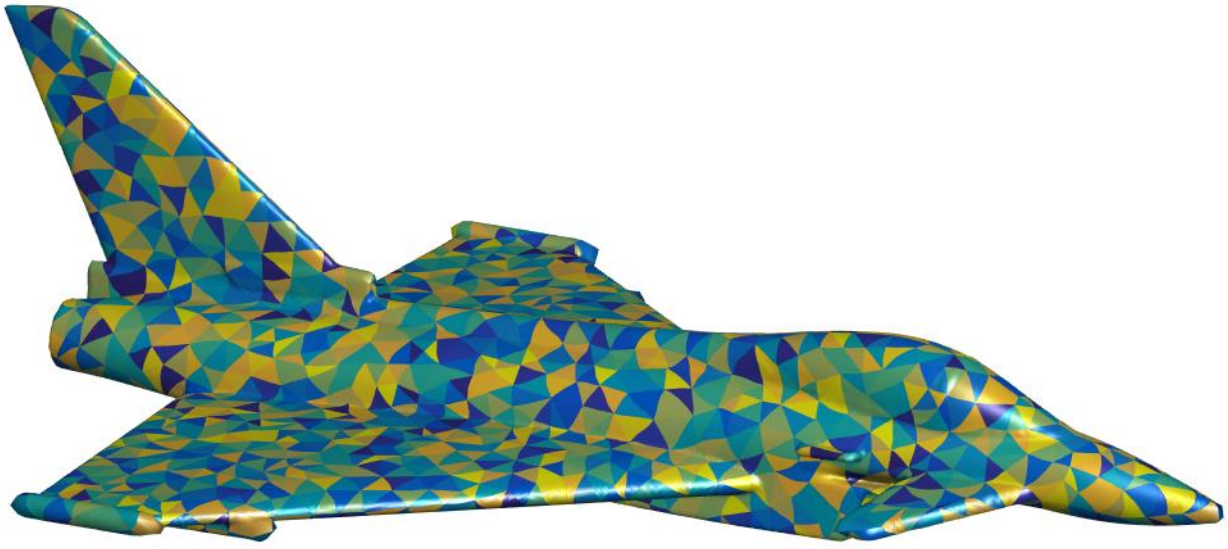


Fig. 3.17. Fighter jet model featuring 3702 10^{th}-order continuous triangular elements generated by the iterative adaptive refinement technique in Fig. 3.6.

Figures 3.19 and 3.20 show first order quadrilateral meshes for the fighter jet and almond, respectively, using the technique from Section 3.4.5. The fighter jet was meshed using 4562 first order elements while the almond was meshed using 1544 first order elements. Detail from the original surfaces is captured well in these continuous quadrilateral meshes despite their low element count and low order.

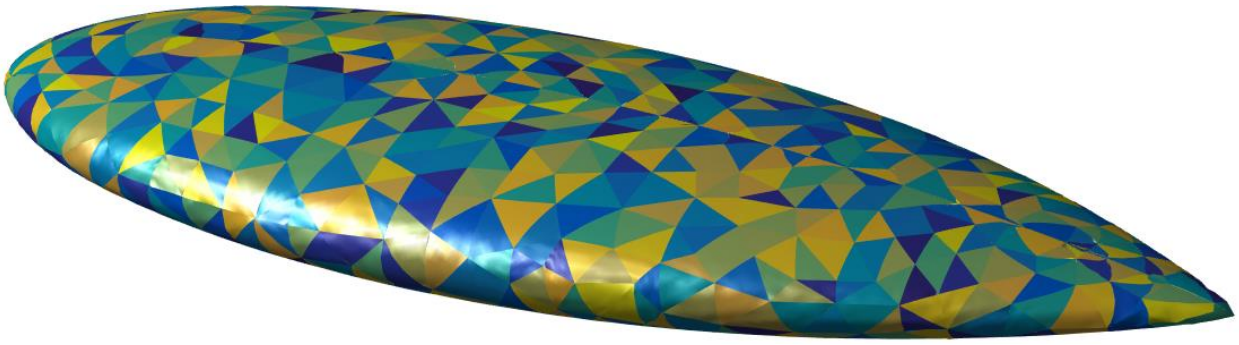


Fig. 3.18. NASA almond model containing 1098 10^{th}-order continuous triangular elements based on the adaptive refinement from Fig. 3.6.

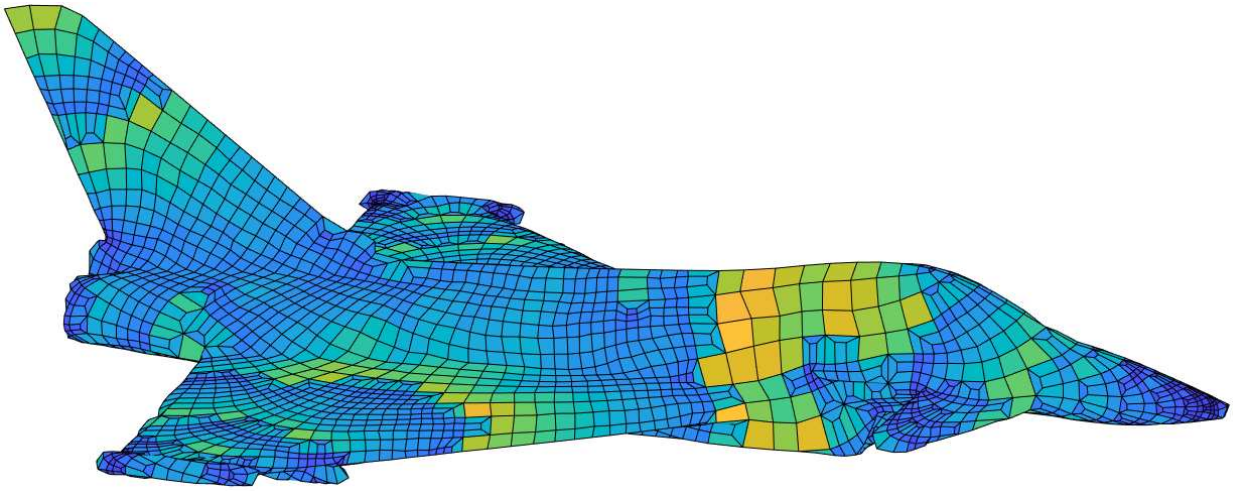


Fig. 3.19. Fighter jet model featuring 4562 1^{st}-order continuous quadrilateral elements generated by the iterative adaptive refinement technique in Fig. 3.8.

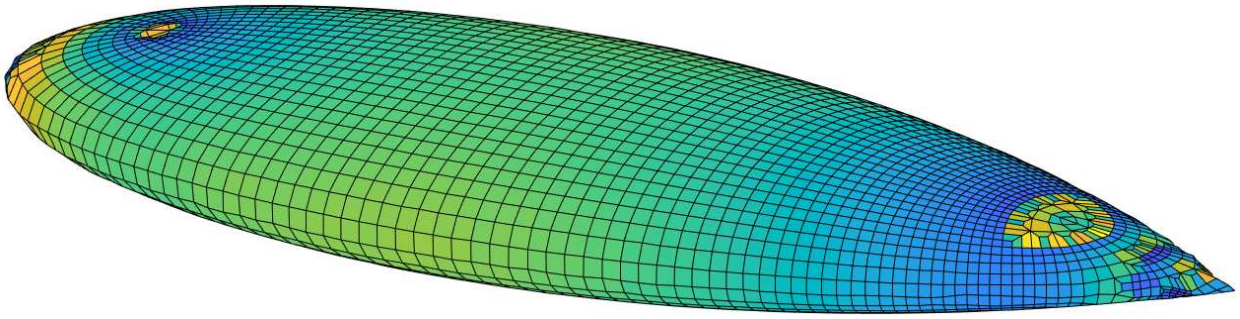


Fig. 3.20. NASA almond model containing 1544 1st-order continuous quadrilateral elements based on the adaptive refinement from Fig. 3.8.

3.5.2 Corner Angle Statistics

Here we demonstrate the conformality of the DSRF method with adaptive refinement for the almond and fighter jet models. For the given examples, we sample both the jet and the almond in the parametric domain adaptively using the refinement indicator and refinement method given for the continuous quadrilateral case in Section 3.4.2. Both meshes were sampled using $N = 256$ to obtain a dense sampling of the conformality of $M(p)$ for both models. Corner angles were computed for every vertex $v \in \tilde{V}$ in the resulting surface meshes, and histograms were produced from the resulting set of corner angles for each mesh and are shown in Fig. 3.21. Note that almost all corner angles are close or equal to 90°, indicating excellent conformality of the DSRF method. This is of utmost importance for many singularity-extraction techniques used in MoM that are not robust to poor corner angles, but otherwise offer excellent accuracy [1]. Additionally, the conformality of the method is critical to maintain high local orthogonality of the basis functions, thereby controlling system condition number.

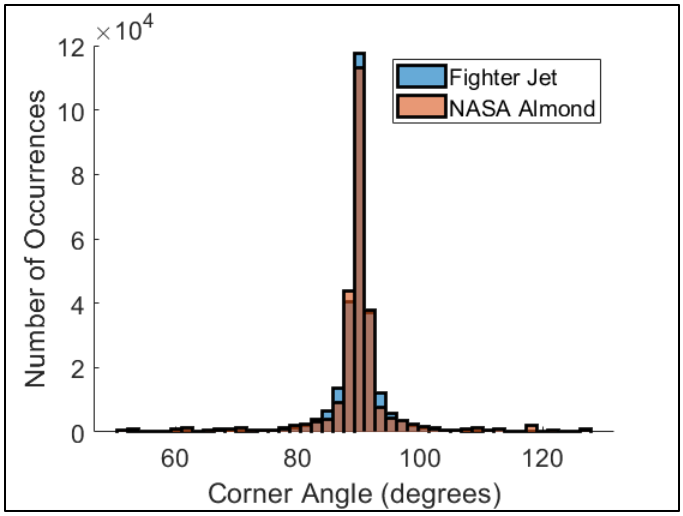


Fig. 3.21. Corner angle histograms for mappings generated using DSRF on continuous quadrilateral meshes. Mappings were sampled using a 256×256 grid of sample points as in Fig. 3.5 for the fighter jet in Fig. 3.4(a) and the NASA almond, respectively.

3.6 Conclusion

This chapter has addressed a crucial but largely under-investigated aspect of modern computational electromagnetics research – high quality surface mesh generation. We have introduced a general, robust surface meshing approach intended here for use as a geometric discretization technique for MoM-SIE problems in electromagnetics but easily extensible to other applications. The proposed technique makes use of new mathematics in the field of topology that has, to our knowledge, not previously breached the field of CEM. The method uses the discrete surface Ricci flow to generate a high-quality discrete conformal mapping from an arbitrary input surface of arbitrary geometry to a parametric domain in which a seed mesh is defined. Iterative adaptive refinement is then used to refine the seed mesh, from which the final surface mesh is produced by an inversion of the mapping using barycentric interpolation. The novel proposed technique has been demonstrated capable of high-quality mesh generation for a variety of surface mesh types, given suitable refinement indicators and methods, including triangular, continuous quadrilateral, and discontinuous quadrilateral of both low and high order. We have defined such

refinement indicators and methods for the studied mesh types and have offered guidelines for extension to mesh types not covered in this chapter.

The mesh generation results presented have shown that DSRF with adaptive refinement easily recreates even complicated initial surfaces using several mesh types over a large range of orders and fidelities. The ability of the new DSRF-based meshing technique to produce high quality meshes even for complicated, highly varied surfaces has been demonstrated for the NASA almond and a fighter jet model. Where high-fidelity meshing is desired, the proposed DSRF technique has been able to near-perfectly capture fine-scale detail using very few high order elements, here demonstrated with as few as 32 elements of up to 64^{th}-order, unprecedented in the field of CEM. Where low-fidelity meshing is desired, DSRF with adaptive refinement has been able to accurately recreate course-scale detail using standard first-order elements. Corner angle measurements have shown that the generated discrete mappings are highly conformal, leading to excellent angle conservation between parametric and final surface meshes when inverse mapped, yielding meshes ideal of angle-sensitive singularity extraction techniques used in MoM.

While the DSRF method with adaptive refinement has been shown a highly effective general surface meshing technique, we consider this the first work in a relatively experimental line of research and appropriately, we have noted some drawbacks of the method as presented. The method is not applicable to complicated multi-part objects when continuity between meshes of individual parts is required. It is also not formally applicable to non-differentiable surfaces where perfect preservation of sharp (non-differentiable) features is required. The method can only asymptotically approach preservation of sharp features, so the error introduced may be unacceptable where sharp feature preservation on otherwise low fidelity meshes is required. We therefore anticipate several areas for future work including improvement of refinement methods

to include sharp-feature preference; extension to multi-part objects while maintaining mesh continuity between parts; improvements to the potentially poor computational scaling of the DSRF; and utilization with simulation-derived error data for adaptive refinement to mitigate not only geometric error during the meshing process, but also numerical error in CEM solvers.

Overall, by leveraging the DSRF, we can provide a unified framework for generating low- or high-order surface meshes of arbitrary element type that integrates with any existing mesh reconstruction tool, to quickly remesh, refine, and optimize. Our DSRF technique enforces high-quality discretizations, even for sub-optimal parent meshes. It provides a mathematical guarantee of element quality, corner angle uniformity, and local surface current basis vector orthogonality, aimed to greatly enhance the accuracy, conditioning properties, stability, robustness, and efficiency of the CEM solution. The ability to automatically generate geometrically ultra-high order elements of high quality demonstrates significant advantages for practical application in CEM, both in reducing the number of unknowns and improving accuracy and robustness. Additionally, mesh refinement or full reconstruction (e.g., first-order triangle to ultra-high order quadrilateral and vice versa) is extremely inexpensive. A precomputed map from the parent surface and its parameterization enables this low-cost reconstruction and may assist many other common and desirable goals such as optimization. As such, DSRF meshes can be locally or globally refined efficiently motivated by geometric constraints, solution error constraints, or both.

4 NON-SELF-ADJACENT RAY CLASSES FOR PARALLELIZABLE SHOOTING BOUNCING RAY TRACING DOUBLE COUNT REMOVAL

4.1 Introduction

Ray tracing is an old and simple computational electromagnetics (CEM) technique that has seen renewed interest in recent years due to increased computing power and demand for fast propagation modeling in electrically large, complicated environments. A frequency-asymptotic technique, ray tracing is well-suited to the types of propagation problems to which classical full-wave techniques like method of moments (MoM), finite difference (FD), and finite element method (FEM) are least suited. As such, ray tracing fills an important gap in the toolkit of methods available to CEM researchers and practitioners for diverse applications including 5G planning, propagation modeling in tunnel environments, and received signal strength (RSS) mapping [90]-[93]. As ray tracing is applied to a broader suite of increasingly demanding applications, the efficiency and scalability of the technique is now, more than ever, paramount to its usefulness.

For electrically large propagation environments with linear, homogeneous media, ray tracing techniques have predominantly relied on image theory (IT) or the shooting-bouncing rays method (SBR); see [94] for an overview of these methods. In both cases, rather than explicitly solving variational formulations of Maxwell's equations and resulting linear systems, as full-wave techniques do, ray tracing iteratively constructs an approximate solution by propagating rays, each representing radiation from a source over a differential solid angle, and recording their interaction with the environment constrained as modeled by high frequency approximations like the Fresnel coefficients and theory of geometric optics (GO). For an excellent historical and theoretical background, see [95].

Image theory computes the paths rays follow from a source to a given receiver by recursively reflecting a source over all boundaries visible from that source to produce a set of image sources—each image source then treated as a new source. This process is continued to some maximum number of reflections, $N_{reflections}$, at which point any valid paths from source to receiver with up to $N_{reflections}$ reflections can be computed from the set of image sources—see [95] for a good overview. The advantage of this approach is that all possible paths between source and receiver with $N_{reflections}$ or fewer are captured exactly, reducing phase error. However, the computational complexity of IT is $O\left(N_{faces}{}^{N_{reflections}}\right)$, where N_{faces} is the number of flat surfaces used to represent material discontinuities in the propagation environment. Since, for modern problems, N_{faces} is large, IT quickly becomes computationally untenable, even for small numbers of reflections. We note, however, that some techniques like reflection spaces or illumination zones can somewhat reduce the computational cost of IT.

SBR overcomes the computational shortcomings of IT by instead choosing a set of ray directions and a fixed number of rays a priori, then propagating each ray through the environment until it has made $N_{reflections}$ reflections or some other stop criterion is met, e.g., the ray leaving some region of interest. This yields linear complexity with the number of rays, and, using domain partitioning methods like the binary space partition (BSP), logarithmic complexity with respect to the number of facets [96]. However, this does not produce a set of exact paths between source point and receiver point, necessitating a method to decide which rays' field contributions should be counted at a given receiver. This is typically resolved by applying either the ray-cone approximation (RCA) [97]-[98] or ray-tube launching (RTL) [99]. Note that we only consider flat, triangular facets in this chapter.

RTL has the advantage of exactly tiling the sphere of possible initial ray directions with no overlap. However, RTL introduces cases where ray tubes are split when only part of a tube's subtended solid angle reflects from a given face. Handling such cases introduces computational overhead and heavily conditional execution, yielding a ray count that grows dynamically with reflection order in a way that cannot quickly be predicted a priori. RTL can therefore not be parallelized in an efficient, synchronous manner, making it a poor choice for modern SBR applications where scalability on synchronous, parallel hardware like graphics processing units (GPUs) is critical [100].

RCA also suffers from a barrier to efficient and complete parallelization: double count removal (DCR). DCR is necessary when using RCA due to inherent overlap between ray cones in the three-dimensional (3D) domain [100]. If a receiver point falls within the overlap of two cones from the same source, the field contribution from that source may be counted twice, leading to significant error in the resulting received power [100]. This necessitates a method to either prevent such cases a priori or detect them and remove them during computation. Many DCR approaches have been proposed in the past, but none have been developed with scalability on modern parallel hardware in mind. In [101], the authors present a DCR method by which rays are described by a characteristic sequence of planes hit, such that two rays with the same characteristic sequence when arriving at the same receiver are duplicates, necessitating the removal of one. This requires a comparison of characteristic sequences between all rays that arrive at a receiver to detect identical characteristic sequences—leading to a worst case complexity of $O\left({N_{rays}}^2\right)$ and producing a mutual data-dependency between rays that prevents effective parallelization. This also suffers from additional computational overhead where multiple coplanar, adjacent facets are present and therefore need to be tracked as the same object to maintain uniqueness of the characteristic

sequence of a unique ray. A similar method is described in [102] that relies on information about each ray's number of reflections, distance traveled, and angle of transmission to detect and remove double counts. This is essentially a continuous version of the characteristic sequence from [101], with which we identify ray paths by continuous-valued properties of their propagation paths rather than discrete indices. The method in [102] suffers from the same mutual data-dependency between rays that hinders the method in [101] from effective parallelization. The most common type of DCR is described well in [103], which uses explicit geometric calculations to determine if two rays that have arrived at the same reception sphere contain the reception point in the overlap of their ray cones, indicating a double count. This approach is fast and reliable for sequential execution, but, as with previous methods, suffers from a mutual data dependency between rays that hinders its parallel performance and scalability. A useful structured sampling method is described in [97] that constrains the number of neighboring rays for any given ray, limiting double count checks to a known set of neighbor rays by sampling recursively on the icosahedron. This is useful to reduce the worst-case complexity of DCR to $O(N_{rays})$. However, the DCR method described in [97] still introduces a mutual data dependency between neighboring rays that prevents efficient parallelization. We elaborate on what we mean by a data dependency and why it makes efficient parallelization difficult in Section 4.5.3.

This chapter proposes an efficient method of double count removal in SBR ray tracing that is highly parallelizable and removes the last major bottleneck to efficient parallel scaling of SBR applied to CEM. We take a similar sampling approach as [97] to limit potential double counts for each ray to a set of known neighbor candidates and maintain an $O(N_{rays})$ worst case run time, but introduce a new DCR method that does not suffer from the mutual data dependency between rays that prevents effective parallelization of previous DCR methods. We introduce non-self-adjacent

(NSA) classes of rays on the structured icosahedral and octahedral samplings such that no two rays in the same class are neighbors. When only one NSA class is processed at a time, no ray has mutual data dependency with any other ray currently being processed, removing the major barrier presented by previous methods to effective parallelization of SBR. Due to the structure of the sampling we use and the way we define the NSA classes, information from at most six neighbor rays needs to be checked for double counting at the time a given ray is processed. The number of neighbor rays that need to be checked and their indices is known a priori for any ray. The NSA classes we introduce have useful properties like symmetry, asymptotic inter-class isotropy, and simple definition yielding easy implementation. We present a four-class NSA formulation on the icosahedron, maintaining complete non-self-adjacency at the minor expense of inter-class isotropy. We also present two three-class NSA formulations: one on the icosahedron maintaining inter-class isotropy but only asymptotic NSA, and one on the octahedron, maintaining inter-class isotropy and full NSA at the expense of decreased global sampling regularity.

In the rest of the chapter, we introduce these NSA classes and associated DCR methodology. We begin with a review of the icosahedral sampling technique, followed by a description of the introduced NSA classes, along with their definition, useful properties, and relative advantages. We next discuss application of the introduced NSA classes to highly scalable DCR, offering a theoretical discussion of the asymptotic correctness of our simple DCR method in terms of sampling the SBR image space. We introduce the image space with motivating examples to facilitate this theoretical discussion. We then present speedup, computation time, and scaling results demonstrating efficacy of the proposed method, achieving over 300× speedup. We conclude by further outlining the potential of the new DCR technique using NSA classes for efficient and scalable SBR.

4.2 The Non-Self-Adjacency Property

To facilitate simple discussion of NSA classes, we make a variety of useful definitions and assumptions while noting a few important consequences. We begin denoting by R the set of all rays to be processed and by $r_i \in R$ the i^{th} ray in R. We assume here that each ray is unique, or formally that $r_i \neq r_j, i \neq j$. We then define the total number of rays, $N_{rays} = |R|$. We also denote by K the set of ray classes, and by $C_i \in K$ a specific ray class. A ray class is a set of rays. The total number of classes is $N_{classes} = |K|$. For all classes, we enforce completeness $\bigcup_{i=1}^{N_{classes}} C_i = R$, and independence $C_i \cap C_j = \emptyset, i \neq j$. In general, we denote the neighborhood (set of neighbors) of r_i as N_i with only the constraint that $r_i \notin N_i$. The most useful choice of N_i for RCA is the set of spherical Voronoi neighbors of r_i, denoted here V_i. However, we maintain generality in the choice of neighbors wherever we use N_i. We formally define the NSA property as

$$\left\{r_i \middle| r_i \in N_j, r_i \in C_k, r_j \in C_k\right\} = \emptyset, \ \ \forall C_k \in K, \tag{4.1}$$

and similarly, the asymptotic NSA property as

$$\lim_{N_{rays}\to\infty} \frac{\left|\{r_i | r_i \in N_j, r_i \in C_k, r_j \in C_k\}\right|}{|\{r_i | r_i \in C_k\}|} = 0, \ \ \forall C_k \in K. \tag{4.2}$$

4.3 Non-Self-Adjacent Ray Classes

4.3.1 Requirements and Motivation

NSA ray classes are those that satisfy (4.1). Structured DCR methods like ours or [97] limit the DCR data dependency to a known neighbor set. For such DCR methods, ray classes that satisfy (4.1) guarantee that no rays within a given class are dependent, allowing all members of a class to be processed in parallel. Any ray class can satisfy (4.1) with the correct neighbor sets, most simply

and least usefully $N_i = \emptyset, \forall r_i \in R$. In competition with this, the specific choice $N_i = V_i, \forall r_i \in R$ is geometrically correct for detecting SBR double counts but constrains the possible classes that satisfy (4.1) for a given sampling pattern. Satisfaction of (4.2) gives an easy solution to this problem. For ray classes with $N_i = V_i, \forall r_i \in R$ that satisfy (4.2) but not (4.1), we can ignore possible double counts between neighbors in the same class, allowing members of the same class to be processed in parallel while introducing only minimal error. In practice, this is done by excluding from the neighbor set of a ray any Voronoi neighbors that share the class of that ray. Because of this, ray classes based on $N_i = V_i, \forall r_i \in R$ that satisfy only (4.2) are almost as useful as those that satisfy (4.1). We show three useful ray class definitions based on $N_i = V_i, \forall r_i \in R$ that satisfy (4.2) or (4.1).

Among the possible methods to define NSA ray classes for SBR, the simplest is to assign each ray to its own class. Since, for removal of the SBR DCR data dependency, we require ray classes to be processed sequentially, assignment of each ray to its own class is equivalent to fully sequential SBR. This may seem trivial but reveals an important consideration for the number of rays per class: if the number of rays per class is less than the number of rays our given hardware can process in parallel, then ray classing presents a computational bottleneck. To maximize the minimum value of N_{rays} for which this bottleneck occurs, it is desirable to choose the minimum number of ray classes possible—the fewer ray classes, the more rays per class. In choosing the minimum number of ray classes, it is easy to see that neither one class nor two classes can give us the necessary NSA property. For $N_{classes} = 1$, (4.1) is not satisfied unless $N_i = \emptyset, \forall r_i \in R$, otherwise a ray and its neighbors are in the same class. For $N_{classes} = 2$, (4.1) is not satisfied unless neighbors of a ray are themselves never neighbors, or in other words, the graph, G, constructed by connecting each $r_i \in R$ to its neighbors contains no topological triangles. G with

no topological triangles can exist in general, but for the most useful case of $N_i = V_i, \forall r_i \in R$, G is the spherical Delaunay triangulation of R, which contains *only* triangles for $N_{rays} > 2$.

For (4.1) to hold when $N_i = V_i, \forall r_i \in R$, neighbors of any $r_i \in R$ cannot be in the same class as r_i and no adjacent neighbors can be in the same class as each other. This requires, at minimum, $N_{classes} = 3$ to fully satisfy (4.1). Since V_i lie on a topological circle around r_i we require

$$|V_i| \; mod \; 2 = 0, \forall r_i \in R, \tag{4.3}$$

or in other words, an even number of neighbors for each ray. A few regular neighborhoods with varying neighbor counts are shown in Fig. 4.1.

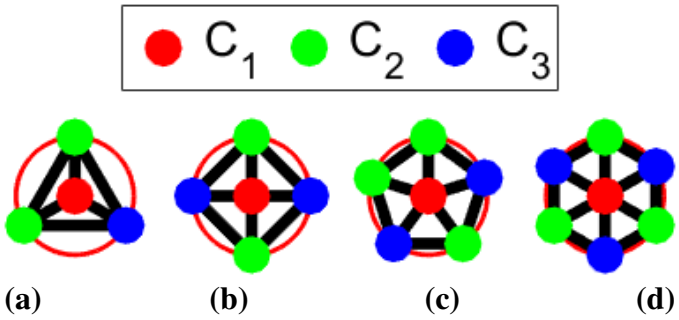


Fig. 4.1. Examples of uniform local topology with three classes: (a) triangular, (b) square, (c) pentagonal, and (d) hexagonal. Neighbors of central cell lie on a topological circle.

For global sampling uniformity, we desire the Voronoi cells of all rays to be identical, regular polygons. In the Euclidean plane, we could simply tile with either squares (Fig. 4.1b) or regular hexagons (Fig. 4.1d) and the class patterns from Fig. 4.1 to satisfy sampling uniformity and (4.1). However, satisfying sampling uniformity on the sphere is only possible for the five platonic solids, offering at most 20 sample points in the case of the dodecahedron (sampled on vertices) or the icosahedron (sampled on face centroids). This motivates methods like the icosahedral subdivision

approach in [97] that, more generally speaking, sample at the vertices of high-frequency geodesic polyhedra to maximize sampling uniformity in a structured way.

4.3.2 Three Classes in Icosahedral Topology

Unfortunately, geodesic polyhedra with icosahedral symmetry never emit a topology that can satisfy (4.1) with $N_{classes} = 3$; they contain 12 vertices with $N_i = 5$, necessitating $N_i \cap N_j \neq \emptyset, i \neq j$ in some cases. However, this defect need only occur at the edges of the original (pre-subdivision) icosahedron. The number of samples that lie on the original icosahedral edges grows linearly with the number of subdivisions while the total number of sample points grows quadratically. The asymptotic NSA property (4.2) is therefore satisfied with $N_{classes} = 3$. We show a simple method here.

We can easily define a set of possible sample points on any triangle and three associated classes that satisfy (4.1) when only points on that triangle, t, are considered. We denote by $\{a_t, b_t, c_t\}$ the set of vertex locations of the triangle, and by $N_{divisions}$ the desired number of subdivisions (an edge of t is split into $N_{divisions}$ new edges). Each sample point on the triangle is then given by

$$s_{i,j}^t = a_t + \frac{i}{N_{divisions}}(b_t - a_t) + \frac{j}{N_{divisions}}(c_t - a_t), \tag{4.4}$$

with indices defined by

$$i, j \in \mathbb{N}_0, i, j \leq N_{divisions} + 1, i + j \leq N_{divisions} + 1. \tag{4.5}$$

The classes on t are then given by

$$C_k^t = \left\{ s_{i,j}^t | (j - i)\ mod\ 3 = k - 1 \right\}, k \in \{1,2,3\}. \tag{4.6}$$

Figure 4.2 shows these classes on a triangle for $N_{divisions} = 11$.

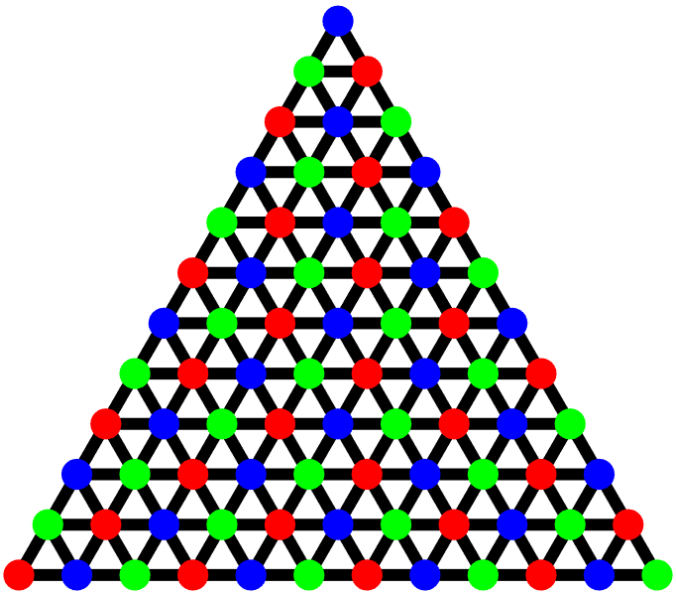


Fig. 4.2. NSA classes defined by (6) for $N_{divisions} = 11$.

Since (4.6) may assign different classes to a point lying on an edge depending on which adjacent triangle we consider, we require an extra step to maintain class independence $C_i \cap C_j = \emptyset, i \neq j$ and expand classes from (4.6) to the entirety of a geodesic polyhedron by a union of (4.6) over its triangles.

We denote by G the set of edges, by P the set of vertices, and by T the set of triangular facets of an arbitrary polyhedron with triangular faces. Each vertex $p \in P$ has a set of incident edges. We specify that a given p is a member of only one of its incident edges. Similarly, each edge $g \in G$ separates two triangular faces. We specify that points on a given g are a member of only one of the two triangles it separates. By these definitions, each point on the geodesic polyhedron is a member of one and only one $t \in T$. If $s_{i,j}^t \in t$, we say t is the parent triangle of $s_{i,j}^t$. The parent triangle of any sample point is unique. The classes on the geodesic polyhedron are then given by

$$C_k = \bigcup_{t \in T} C_k^t, \tag{4.7}$$

where indices are as defined in (4.5). Figure 4.3 shows these classes on the icosahedron with $N_{divisions} = 13$.

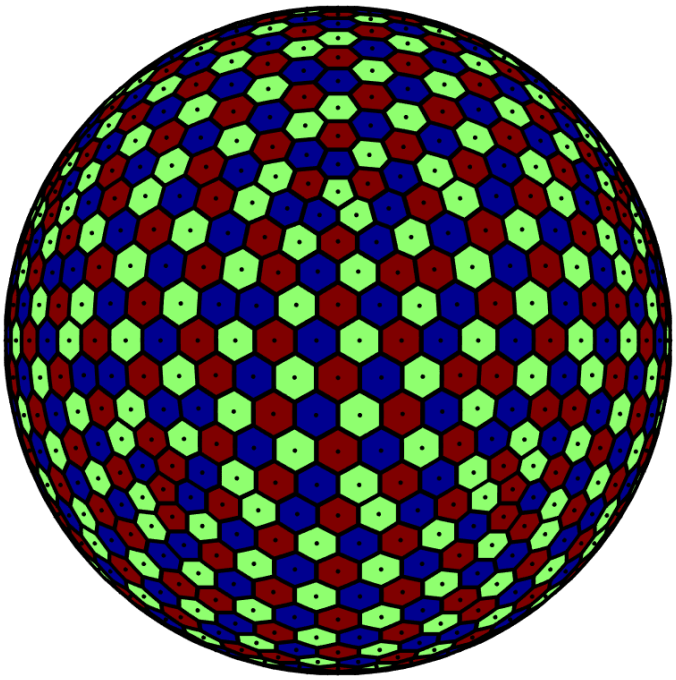


Fig. 4.3. Asymptotically-NSA classes on the icosahedron with $N_{divisions} = 13$ and $N_{classes} = 3$. Voronoi-adjacent rays in the same class are common here due to very low ray count.

This choice of classes and sample points offers an excellent foundation for parallel SBR-DCR. Although only (4.2) is satisfied, the points that violate (4.1) are constrained to those lying on the edges of the original icosahedron and their immediate neighbors. These points represent a proportion of the total N_{rays} that decreases linearly with increased $N_{divisions}$. This makes double counting between adjacent same-class neighbors inconsequential at the high ray counts typically used in most SBR applications.

4.3.3 Three Classes in Octahedral Topology

The presence of twelve points with five neighbors on geodesic polyhedra with icosahedral topology prevents such polyhedra from satisfying (4.3) at all vertices. Octahedral geodesic polyhedra, on the other hand, contain only points that satisfy (4.3), making them a good option for sampling where perfect non-self-adjacency is desired with the minimum number of classes.

Figure 4.4 shows (4.6) and (4.7) applied to the octahedron with $N_{classes} = 3$. We use an ordering of $\{a_t, b_t, c_t\}$ for each triangle that maintains class independence on edges between triangles regardless of parent triangle assignment. Many such orderings exist on the octahedron, so we do not specify one here. These octahedral classes have the advantage of fully satisfying (4.1) with only three classes, but at the cost of somewhat reduced sample uniformity compared to the icosahedron.

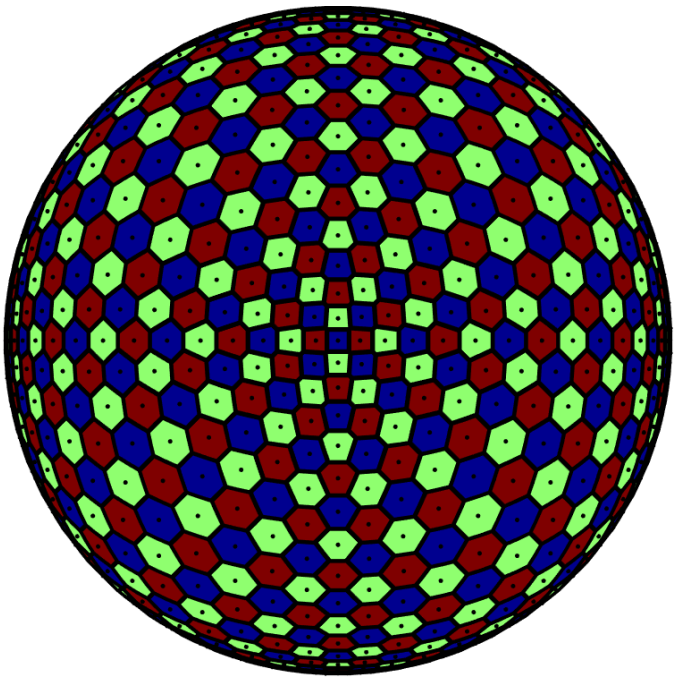


Fig. 4.4. Perfect NSA classes on the icosahedron with $N_{divisions} = 17$ and $N_{classes} = 3$. No Voronoi-adjacent rays are in the same class.

4.3.4 Four Classes in Icosahedral Topology

To achieve both high sampling uniformity and satisfaction of (4.1), we can define fully NSA classes on the icosahedron with $N_{classes} = 4$. We choose one of many four-color vertex colorings of the icosahedron, with colors corresponding to class indices $k \in \{1,2,3,4\}$. This assigns to each of the icosahedron's twelve vertices one of four classes, such that no adjacent vertices share a class.

On a given triangle, we again denote by $\{a_t, b_t, c_t\}$ the set of vertex locations, and now by $\{k_{a_t}, k_{b_t}, k_{c_t}\}$ the set of corresponding class indices. For simplicity, we define the vector $h = \langle k_{a_t}, k_{b_t}, k_{c_t} \rangle$, with $h(l)$ denoting its l^{th} entry. Sample points are again defined by (4.4) with indices defined by (4.5). However, instead of (4.6), the classes on t are now given by

$$C^t_{h(l)} = \{s^t_{i,j} | (j - i)\ mod\ 3 = h(l) - 1\}, l \in \{1,2,3\}. \tag{4.8}$$

Assigning parent triangles as before to maintain class independence, the four classes on the icosahedral geodesic polyhedron are again given by (4.7). Note that $N_{divisions}$ must be one less than an integer multiple of 3 to maintain class independence. Figure 4.5 shows (4.7) and (4.8) applied to the icosahedron with $N_{divisions} = 13$ and $N_{classes} = 4$. These classes fully satisfy (4.1) and have the same sampling uniformity as those from Fig. 4.3, but each class no longer samples the entire sphere.

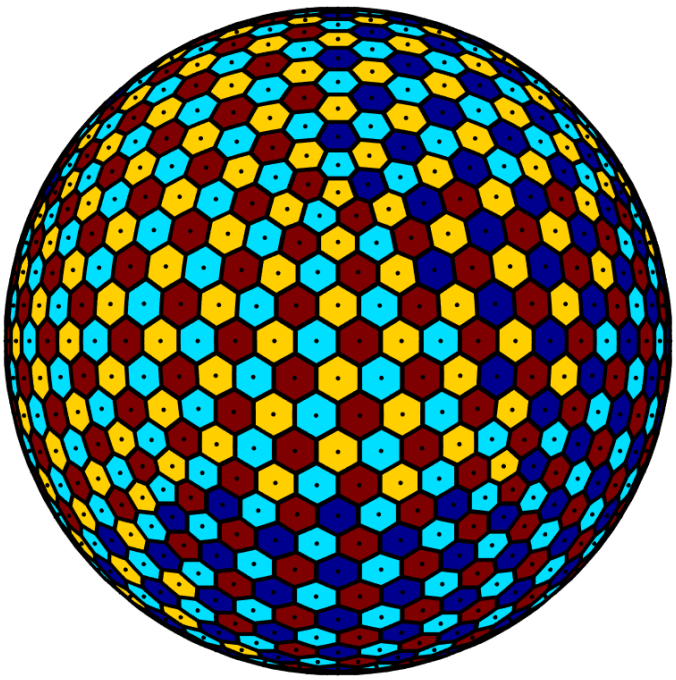


Fig. 4.5. Perfect NSA classes on the icosahedron with $N_{divisions} = 13$ and $N_{classes} = 4$. No Voronoi-adjacent rays are in the same class.

4.4 SBR as a Sampling of the Image Space

To facilitate a discussion of SBR DCR, we introduce the concepts of the environment space and image space. The environment space, E, is the physical space in which we are modeling propagation. Any point $e \in E$ is given by a scalar real-valued triplet with spherical coordinates $\langle \rho, \theta, \phi \rangle, \rho \in [0, \infty),\ \theta \in [0, \pi],\ \phi \in (-\pi, \pi]$. A ray, r, with initial direction $\langle \theta_0, \phi_0 \rangle$ follows a curve, s, through E, parametrized by d such that $s(d) = e$ is the point on s at which the ray has traveled d distance along s. The curve s is a straight line radiating from the origin if no reflections occur, a continuous path composed of line segments of reflections occur in homogeneous media, or a general continuous, curved path in inhomogeneous media. We consider only the first two cases here. The image space, Q, represents the space in which paths taken by rays follow straight lines radiating from the origin regardless of their reflections in the environment space. Note that we consider only reflections here, not transmission. Any point $q \in Q$ is also given by a real-valued

triplet with spherical coordinates $\langle d, \theta_0, \phi_0 \rangle, d \in [0, D_{max}],\ \theta \in [0,2\pi],\ \phi \in (-\pi, \pi]$, where d is the distance traveled in E for the ray with initial direction $\langle \theta_0, \phi_0 \rangle$. D_{max} gives the maximum propagation distance considered. The ray source is the origin of both spaces. We define a map M such that $M(q) = e = s(d)$. Note that M is in general not invertible: $\exists e \in E\ s.t. \{q | M(q) = e\} = \emptyset$; we call such e occluded. Note that $E = Q$ in homogeneous media with no reflections and $D_{max} = \infty$.

For clarity, we give a few examples, shown in Fig. 4.6, in two dimensions of E and the associated Q. To produce these plots, we constrained $\phi = 0$ and uniformly distributed initial ray directions in θ. Since a given ray only samples E along a given path, in turn sampling Q only along a straight radial path from the origin, we interpolate Q between rays using RCA and assigning any q not on a ray path the properties of the nearest q on a ray path. This produces a piecewise approximation of M: $\widetilde{M}$. Rays were propagated for a fixed, constant distance. We chose $N_{rays} = 1000$ so no defects due to the $\widetilde{M}$ approximation are visible at the chosen figure resolution and propagation distance. To demonstrate the relationship between E and Q, we assign hues to $e \in E$ corresponding to θ and opacity increasing with ρ. Each $q \in Q$ is then assigned the hue and opacity of $\widetilde{M}(q)$.

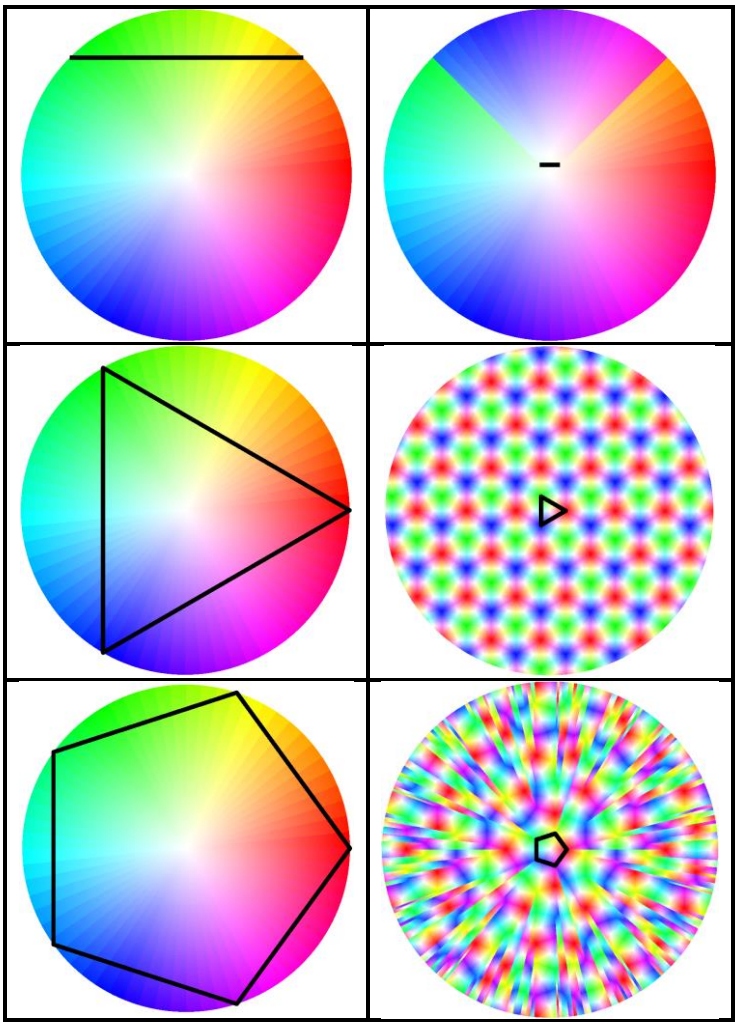


Fig. 4.6. Examples of environment and associated image spaces. Reflectors are shown in black. The left column shows three examples of environment spaces: a single plane, a triangle, and a pentagon. The right column shows the associated image spaces.

We quantify the geometric error introduced by approximating the mapping M as

$$\varepsilon_{geometric} = \frac{3}{4\pi D_{max}^4} \int_Q \left\| M(q) - \widetilde{M}(q) \right\| dq. \tag{4.9}$$

Note that the 4th rather than 3rd power in (4.9) comes from normalizing with respect to D_{max} in addition to the volume of integration. We define geometric convergence of SBR as the property that

$$\lim_{N_{rays} \to \infty} \varepsilon_{geometric} = 0. \tag{4.10}$$

SBR has geometric convergence if, for an arbitrary region, Ω, in θ, ϕ on the sample sphere surface,

$$\lim_{N_{rays}\to\infty} |\{r|\langle\theta_0, \phi_0\rangle \in \Omega\}| = \infty. \tag{4.11}$$

It is easy to show from (4.4) that the sampling patterns in Section 4.3 enforce (4.11) and therefore (4.10).

4.5 Efficient, Parallel Double Count Removal

4.5.1 The Proposed Method

To present our DCR method, we first make some definitions for clarity. We have a set of observation points O with $o \in E, \forall o \in O$ and $N_{observations} = |O|$. We denote by $N_{reflections}$ the maximum number of reflections considered for all rays. The goal of SBR is to compute the field at all observation points due to a set of source points. We consider only one source point at a time, combining fields at observation points by superposition when multiple source points are present. To compute from which rays contributions are considered at a given $o \in O$, we use the dynamically-sized sphere intersection method from [103]. We choose α from [103] for a given $r_i \in R$ as the maximum angle between r_i and any $r_j \in V_i$. This prevents any gaps between ray cones, allowing errors only in the form of overlap (double counts) between neighbors.

Our DCR technique is more straightforward than those in [97]-[103] and can be summarized simply when implementation details are ignored: We process only one ray class and only the n^{th} reflection for rays in that class at a time, recording any ray-observation pairs for sphere intersections that occur between the n^{th} and $(n+1)^{\text{th}}$ reflection. We only keep a ray-observation pair containing r_i and o_j if no neighbors of r_i are members of pairs containing o_j. Note that, for NSA classes like those in Section 4.3.2 that satisfy (4.2) but not (4.1), we do not consider Voronoi-neighboring rays in the same class as neighbors for the purpose of DCR. This introduces an error

that is asymptotically negligible, as discussed in Section 4.3.2.

This DCR method is extremely simple, and with the NSA classes from Section 4.3, highly parallelizable and scalable. We give a comparison to existing DCR methods as well as a pseudocode example for one possible implementation in Section 4.5.3. First, however, we consider the glaring omission we make in defining such a simple method. Our method introduces an obvious error that previous methods have mitigated with more-complicated techniques. Where neighboring rays hit different, non-coplanar facets but intersect the same observation sphere with the same reflection count, our method will detect a false double-count not detected by more-rigorous DCR methods. A simple example of the type of false double count detected by our method is shown in Fig. 4.7. If rays r_1 and r_2 are neighbors, only one of their field contributions will be counted at o after having reflected one time, even though these reflections were from different, non-coplanar facets. The field contributions of r_1 and r_2 in this case represent different image sources, so both should be counted.

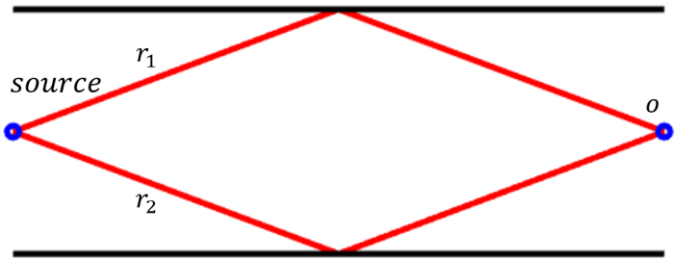


Fig. 4.7. A simple example of how a false double count may be detected by our method. If the two rays are neighbors, our method will count only one of their contributions at o, despite the rays representing unique images.

Production of false double counts, superficially, seem like a major flaw with our DCR method. However, we show here that the proportion of false double counts drops asymptotically to 0 with increased N_{rays}, and by extension the asymptotic correctness of our DCR method.

4.5.2 Asymptotic Correctness

To show asymptotic correctness of our method, it suffices to show the proportion of neighboring rays that hit the same triangular facets in the same order after having traveled some finite maximum distance D_{max} approaches 1 asymptotically as $N_{rays} \to \infty$. Satisfaction of this property can be shown using the notion of the image space as follows.

Denote by EG the set of points in E on facet edges and $QG = \{q | M(q) \in EG\}$. Projecting QG in the d direction onto the unit sphere gives QG'. If the domain contains a finite number of reflecting facets and D_{max} is finite, QG' partitions the unit sphere into a finite number of polygonal regions. Note that these partitions correspond to the largest possible polygonal cone boundaries of ray tubes in RTL after splitting if rays are only traced until D_{max}. By (4.11) and the observation that the region boundaries subtend only an infinitesimal solid angle, the proportion of neighboring rays that hit the same triangular facets in the same order by their D_{max} approaches 1 and our DCR method introduces a proportion of false double counts that decreases to 0 as $N_{rays} \to \infty$.

Convergence is also apparent from (4.11) and the perspective of image theory. Since (4.11) implies the solid angle subtended by each ray cone decreases asymptotically toward zero as $N_{rays} \to \infty$, the probability of neighboring rays hitting different facets at their first reflection (necessary but not sufficient for a false double count) also decreases toward zero. In the case where rays hit the same facet at their first reflection, the resulting reflected rays can be considered to radiate from the same image source. The rays' second reflection can then be treated as a first reflection, yielding an inductive proof of convergence for arbitrary reflection count or D_{max}.

4.5.3 Pseudocode and Comparison to Existing DCR Methods

To understand our DCR approach with NSA classes and why it allows for efficient parallelization, it is useful to understand why existing approaches make this more difficult. Generically speaking, existing DCR methods attempt to apply some function $DCR(r_i, o, c)$, to determine whether a ray causes a double count at a given observation o for a given context c and, if so, resolve that double count in some data structure that tracks ray-observation intersections. All ray-observation intersections remaining after DCR are counted in the final field computations for the corresponding observation. Our method is no different in this regard. Consider the case, however, where r_1, r_2, and r_3 are mutual neighbors; i.e. $r_1, r_2 \in N_3$, $r_1, r_3 \in N_2$, $r_2, r_3 \in N_1$. As noted in Section 4.3, the correct neighbor choice $N_i = V_i, \forall r_i \in R$ yields the spherical Delaunay triangulation for G, so cases like this occur for every ray regardless of the sampling method chosen. Consider also that o falls in the overlap of all three rays' cones. Only one of these rays should be counted, although each of the three is equally valid under RCA. For a given context, we have three potential instances of DCR to process: $DCR(r_1, o, c)$, $DCR(r_2, o, c)$, and $DCR(r_3, o, c)$. To correctly resolve this situation by counting only one of the three rays, the outputs of the three processes must be consistent, e.g. if r_1 is counted, r_2 and r_3 cannot be counted. This requires that, for instance, computation of $DCR(r_2, o, c)$ and $DCR(r_3, o, c)$ is dependent on the result of $DCR(r_1, o, c)$, so the three processes cannot complete execution simultaneously. This is equally true if DCR constitutes a simple comparison of characteristic sequences [101] as it is for geometric computations between rays [103]. The problem lies in how synchronization mechanisms like mutexes that allow such data dependencies to be handled in a parallel execution environment delay process completion; a given thread must wait for others on which it is dependent. Such

parallelization approaches are inefficient, since processor cycles are wasted while waiting, or, in more complicated approaches, while switching between threads.

To further illustrate the problem presented by adjacent ray data dependencies, we give below two examples of pseudocode, one for our DCR approach with NSA classes, and another for a generic DCR approach without NSA classes. For both examples, we assume that neighbor sets are defined and known ahead of time, as in our method or e.g. [97], since this is already a common approach in recent literature to limit the data dependency to only a small neighbor set. We also assume that indices of observations intersected by a given ray, r_i, are recorded in a hitlist denoted HL_i. We denote a generic observation point index as idx. There are many ways to manage this information, but we consider this the simplest and most illustrative.

Algorithm 4.1. Example of DCR with NSA classes.

```
1 for k ∈ [1..N_classes] do
2           for i ∈ [1..|C_k|] do in parallel
3                 for r_j ∈ N_i do
4                       conflicts ← {idx | idx ∈ HL_i, idx ∈ HL_j}
5                       for idx ∈ conflicts do
6                             HL_i ← HL_i \ {idx}
```

Algorithm 4.2. Example of DCR with no NSA classes.

```
1 for i ∈ [1..N_rays] do in parallel
2           for r_j ∈ N_i do
3                 if i > 1 then
3                       while not available_j do
4                             wait
5                 conflicts ← {idx | idx ∈ HL_i, idx ∈ HL_j}
6                 for idx ∈ conflicts do
7                       HL_i ← HL_i \ {idx}
8     available_i ← true
```

Using NSA classes as in Algorithm 4.1, only non-neighboring rays are processed in parallel, so $conflicts$ can be readily computed for each parallel instance. Note that lines 3 through

6 are effectively an implementation of $DCR(r_i, o, c)$. Consider, in contrast, if we use no NSA classes. We must somehow resolve cases like $r_1, r_2 \in N_3$, $r_1, r_3 \in N_2$, $r_2, r_3 \in N_1$. One way to do this is Algorithm 4.2. Rays processed in parallel may now be dependent on each other, so we define the Boolean variable $available_j$ to keep track of whether double count removal has been completed for r_j. Here, lines 2 through 8 are effectively an implementation of $DCR(r_i, o, c)$. The process for r_1 can execute immediately, but other threads must wait until their dependencies are resolved. In fact, in this implementation, most threads will spend most of the total computation time waiting.

For simplicity, both examples intentionally ignore context. This is appropriate for our DCR method, but not for existing methods. Context is information other than ray index and observation index that identifies a unique field contribution. For instance, in [101], c is a characteristic sequence of facets hit by a ray before registering a hit for an observation. To consider context in general, updates to $conflicts$ would require comparison of e.g. characteristic sequences [101] or geometry information [103], adding a layer of complexity and reducing performance.

Algorithm 4.3. Example of ordering of our DCR method relative to other SBR processes.

```
1 for n ∈ [1..N_reflections] do
2           initialize HL_i to empty ∀i ∈ [1..N_rays]
3           for k ∈ [1..N_classes] do
4                 for i ∈ [1..|C_k|] do in parallel
5                       trace r_i to n^th reflection
6                       compute sphere intersections
7                       fill HL_i
8           DCR
```

Our DCR method avoids this by defining c as the number of reflections taken by a ray before encountering an observation sphere. We then only calculate hits for the n^{th} reflection of all rays simultaneously, resetting hitlists before the $(n+1)^{\text{th}}$ reflection. Since all entries in HL_i correspond

to the same c, our method allows c to be ignored during DCR. A simple example of how we order our DCR method relative to other SBR processes is presented in Algorithm 4.3.

4.6 Results and Discussion

To demonstrate the scalability of the proposed NSA class-based parallel DCR method and its practical advantages over inherently sequential approaches, we produced an efficient GPU-based implementation. As a baseline, we also produced an efficient but fully sequential CPU-based implementation of the method. The CPU-based implementation performs the same operations from Section 4.5, but processes only one ray at a time, rather than rays in each class in parallel. Both implementations used the same parallel, GPU-based SBR ray propagation, sphere intersection, and field computation implementations, the computation times of which were included in the total computation time. We show results using the 3- and 4-class icosahedral schemes from Section 4.3, denoted Ico3 and Ico4, respectively. Our intention in presenting results for both Ico3 and Ico4 here is to demonstrate the bottleneck introduced by NSA classes does not occur for 3-class or 4-class schemes over the range of typical parameters tested (as low as $N_{rays} = 10^3$). Note that, since our DCR technique requires fewer operations to detect and handle double counts than existing methods, its use as a sequential benchmark here likely underestimates the computation time of most existing DCR approaches. Also note that, with good implementation, no pre-process step is required for management of ray class designations. Each ray's class can be determined in constant time from its parent triangle index and its indices within that triangle. All results were produced on a mid-range (as of 2019) consumer workstation equipped with an Intel i7-3770 3.4 GHz CPU and an NVIDIA GeForce GTX 1060 6GB GPU with 1280 CUDA cores. A 4×4×1000-meter waveguide was used as the propagation environment. Since our initial implementation is targeted to CUDA-enabled GPUs, we are not able to include a strong scaling

plot (i.e., scaling with respect to core count) since threads are automatically distributed to GPU streaming multiprocessors in the CUDA paradigm, offering us little control over how many are used simultaneously. We hope to present a strong scaling plot in future work once we have an efficient CPU implementation.

Figure 4.8 shows the computation time taken only by DCR for both the sequential and parallel implementations with respect to increasing N_{rays}. We chose to test a wide range of N_{rays} values that we believe is representative of the range of ray counts used for most practical applications. We see vastly improved performance and scaling of parallel DCR over the sequential implementation, with parallel DCR outperforming sequential for all N_{rays} tested and a maximum observed speedup over 300×. We observe the largest speedups for the highest N_{rays} tested, with the speedup for lower ray counts likely constrained by host-device communication overhead below $N_{rays} = 10^6$.

Figure 4.9 shows the fraction of the total computation time taken by DCR for each approach with respect to N_{rays}. The sequential example takes roughly 50% of the total computation time by 100 million rays. The parallel examples, meanwhile, take less than 1% of the total time. Measuring the time proportionality of DCR is useful because it offers a simple, relative comparison of DCR to other important steps of the SBR algorithm. Encouragingly, the results of Fig. 4.9 indicate that time taken by our parallel implementation is asymptotically non-dominant with respect to increasing ray count. Furthermore, the fact that Ico3 and Ico4 agree almost perfectly shows that no bottleneck is introduced by NSA ray classing over the wide range of ray counts tested.

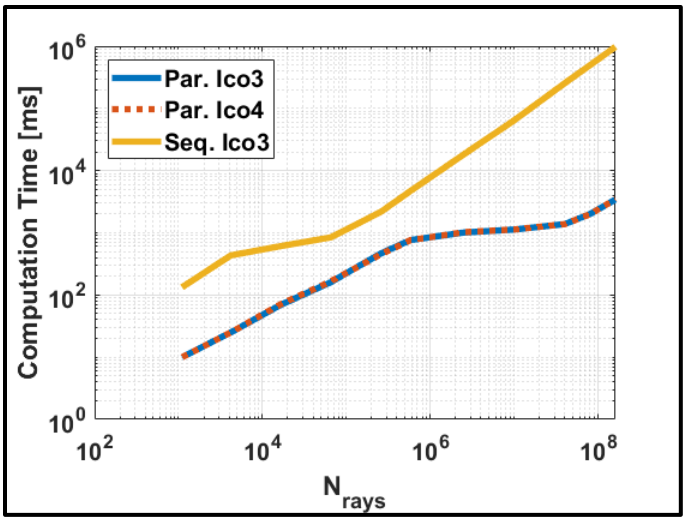


Fig. 4.8. Computation time of sequential vs. parallel DCR with respect to N_{rays}. Other parameters were constant: $N_{reflections} = 20$, $N_{observations} = 500$.

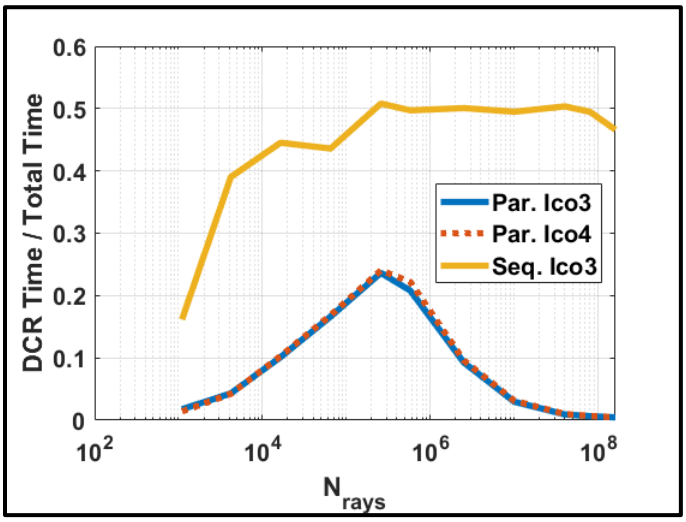


Fig. 4.9. Proportion of total SBR computation time taken by sequential vs. parallel DCR with respect to N_{rays}. Other parameters were constant: $N_{reflections} = 20$, $N_{observations} = 500$.

We also note that the time proportionality peak in Fig. 4.9 around $N_{rays} = 10^6$ lends evidence to our belief that non-asymptotic effects like communication overhead constrain the speedup in Fig. 4.8 below this value.

Figure 4.10 shows similar results to figure 4.8, but with respect to the maximum number of reflections simulated for any given ray. We again chose a range of values that we consider

typical for most practical applications. The parallel examples are once again faster in all cases, even at high reflection orders, with a maximum observed speedup over 100× for the parameter values tested. We note that the observed speedup becomes lower at higher reflection orders. We believe this is due to memory limitations of our GPU hardware at high reflection orders necessitating host-device communication.

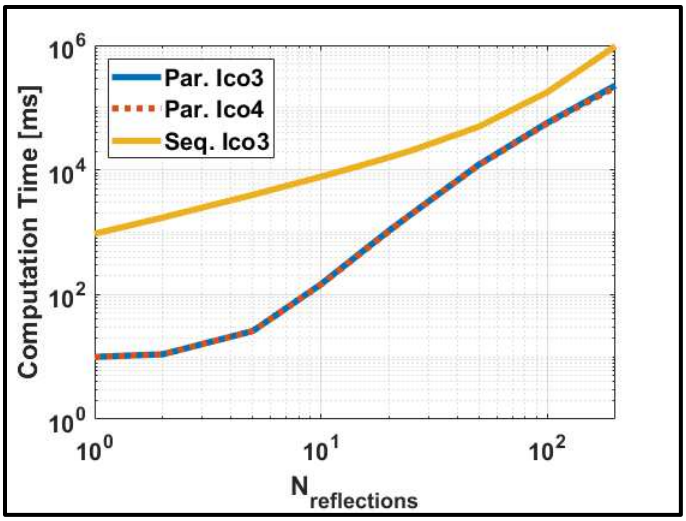


Fig. 4.10. Computation time of sequential vs. parallel DCR with respect to $N_{reflections}$. Other parameters were constant: $N_{rays} = 2{,}505{,}000$, $N_{observations} = 500$.

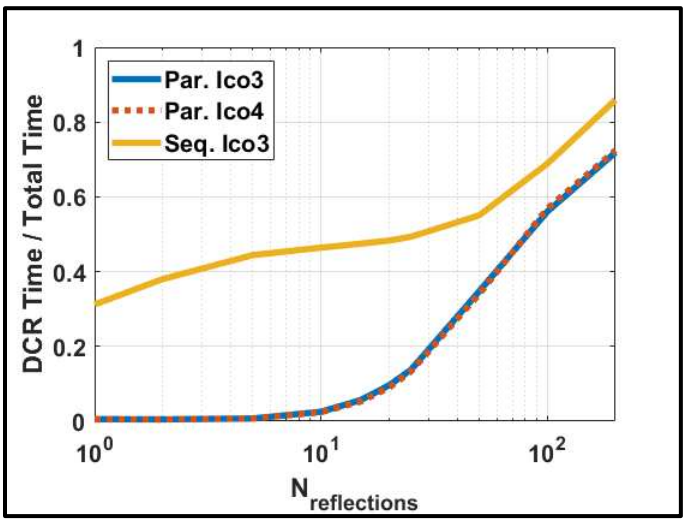


Fig. 4.11. Proportion of total SBR computation time taken by sequential vs. parallel DCR with respect to $N_{reflections}$. Other parameters were constant: $N_{rays} = 2{,}505{,}000$, $N_{observations} = 500$.

Figure 4.11, analogous to Fig. 4.9, shows the proportion of the total computation time taken by each example. Although it appears in Fig. 4.11 that asymptotic behavior of the time proportionality has begun to dominate (we observe a linear trend on the semilog scale by around $N_{reflections} = 100$), this is unlikely to be the case. The proportion of the total time taken by DCR is limited to 1, so the observed trend is misleading (all three curves must level out at some point). As with Fig. 4.10, we believe the reduced efficiency at higher reflection orders can be attributed to memory limitations of our GPU hardware and associated host-device communication overhead.

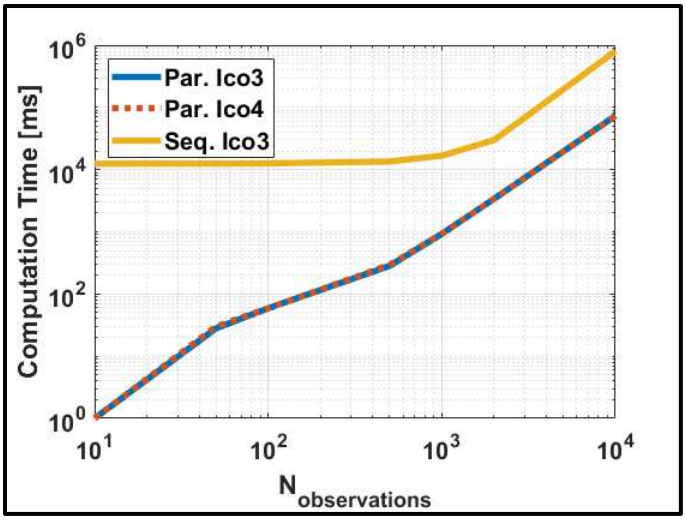


Fig. 4.12. Computation time of sequential vs. parallel DCR with respect to $N_{observations}$. Other parameters were constant: $N_{rays} = 2{,}505{,}000$, $N_{reflections} = 20$.

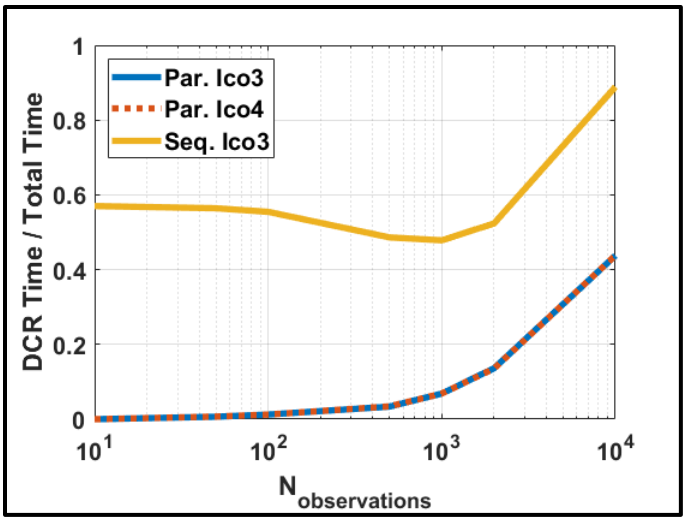


Fig. 4.13. Proportion of total SBR computation time taken by sequential vs. parallel DCR with respect to $N_{observations}$. Other parameters were constant: $N_{rays} = 2{,}505{,}000$, $N_{reflections} = 20$.

Figures 4.12 and 4.13 are analogous to figures 4.8 and4. 9 but with respect to the number of field observation points. The parallel examples tested for Fig. 4.12 achieve a maximum observed speedup over 10,000×, although this is for very low $N_{observations}$. At high $N_{observations}$, the observed speedup levels out to about 10× on our test hardware. Like Figs. 4.10 and 4.11, we believe the reduced efficiency in Figs. 4.12 and 4.13 for high $N_{observations}$ is due to host-device communication overhead.

4.7 Conclusion

This chapter has introduced non-self-adjacent ray classes for efficient, parallelizable shooting-bouncing-ray tracing double count removal. Unlike previous DCR methods, the approach made possible by the NSA ray classes introduced in this chapter can take advantage of modern, parallel computing hardware, e.g., GPUs, that was not available in ray tracing's theoretical infancy. Predominantly geometric aspects of SBR like ray path computation and ray intersection tests have long been efficiently parallelizable, and most modern SBR approaches have taken advantage of

this. However, the parallel approach to DCR enabled by the present work removes the last and final barrier to fully parallel, large-scale SBR simulations. This is crucial as problem sizes continue to grow, necessitating highly parallel and efficient CEM algorithms.

5 DATA-ENABLED ADVANCEMENT OF COMPUTATION IN ENGINEERING: A ROBUST MACHINE LEARNING APPROACH TO ACCELERATING VARIATIONAL METHODS IN ELECTROMAGNETICS AND OTHER DISCIPLINES

5.1 Introduction

Variational techniques like FEM, MoM, and FD method are dominant for solving numerical physics problems in CEM and computational science/engineering (CSE) due to their flexibility, robustness, and rigorous mathematical underpinnings. The principal shortcoming of these methods is their poor scaling and high computational cost. We introduce a broadly applicable method by which neural networks can be applied to speed up variational methods without sacrificing their desirable characteristics. Rather than predicting solutions to these problems directly, we use neural networks to guess a highly simplified basis on which to solve the problem rigorously using existing techniques.

Previous work seeking to use neural networks to make predictions about the solutions to computational physics or CSE problems has capitalized on the strong predictive power of well-trained neural networks but has not addressed the shortcomings of using such an inherently empirical approach for real-world engineering problems. This has limited the real-world usefulness of such results. Most previous work has focused on predicting quantities derived from a numerical solution given a description of the physical problem, typically material parameters in the computational domain and excitations for the problem [104]–[108]. There has been occasional work that uses a neural network to predict the solution itself, rather than a derived quantity [109]–[110]. In this way, such research has sought to effectively replace variational methods with neural networks as the numerical tool used to solve computational physics problems.

In contrast, some of the biggest breakthroughs and substantial applications of neural networks to perform challenging tasks with the accuracy needed for industry use have used existing, mathematically formal methods guided by the intuitive predictive capability of neural networks to achieve speedup and even improve accuracy [111]–[112]. We believe this is critical to the application of machine learning in most engineering contexts. We have found no existing research that has coupled neural networks with variational methods in a broadly applicable, robust way. The closest we have found is the use of neural networks to predict bulk material parameters for faster multi-scale FEM simulations in structural mechanics [113]–[118]. We consider this excellent work and in line with the philosophy of using neural networks to guide more-rigorous methods, but unfortunately the method described is specific to structural mechanics problems.

Predicting basis functions directly, rather than trying to predict solutions or derived quantities, we exploit the crucial strength of neural networks: the ability to efficiently and accurately learn low-dimensional representations of complicated, high-dimensional datasets to understand underlying correlations. In the context of variational methods for CSE, this means learning not only the fundamental physical behavior of problems, but also larger emergent trends that define the aggregate behavior of a physical structure under simulation. By using predicted bases to rigorously solve a given problem, we maintain the key strengths of variational methods: rigorous bounding of solution error, accurate error estimation, and well-substantiated methods to improve solution accuracy when solution error is found to be excessive for the given application. These benefits are crucial for any numerical method applied in an engineering context. Meanwhile, the proposed method avoids the downsides traditionally associated with empirical, data-driven predictors like neural networks, namely their black box nature and unpredictability when subject to inputs dissimilar to those used for training.

5.2 Theory

We consider in general a discretized linear (or linearized) differential or integral equation-based problem with solution S, set of basis functions F, and linear system of form $[A]x = b$, where $f_i \in F$ and x_i denote the i^{th} basis function and associated solution weight, respectively. This system may be Galerkin-weighted, but we do not impose this. The weak solution to the problem with N basis functions in this notation is given by

$$\tilde{S} = \sum_{i \in [1..N]} x_i f_i \approx S \tag{5.1}$$

Construction and solution of the linear system for large problems is computationally time consuming and memory intensive. With N basis functions, solution of the system has time complexity $O(N^2)$ for iterative methods or $O(N^3)$ for direct methods [1]. Meanwhile, construction of the system, typically dominated by performing the necessary integrations, has complexity $O(N^2)$ for boundary integral methods due to global coupling of the basis functions and $O(N)$ for finite element and finite difference methods due to local coupling.

Convolutional neural networks (CNNs) have seen an explosion in popularity in recent years due to advances in parallel computing power and network architecture that have, together, enabled applicability of CNNs to a broad range of complicated tasks from playing board games [111] to classifying images with record accuracy [119]. For an excellent overview of the theory and concept of modern CNN architectures, see [120]. CNNs take advantage of spatial correlation in data to efficiently learn complicated underlying trends more effectively than classical fully connected neural networks. If our data have d discrete spatial dimensions, input to a CNN is an array with $d + 1$ dimensions; the extra dimension of the array corresponding to the number of

input channels, c. We denote by $n_i, i \in [1..d]$ the size of the input array in the i^{th} spatial dimension. The total number of scalar inputs to a CNN is then

$$N_{inputs} = c * \prod_{i \in [1..d]} n_i \tag{5.2}$$

The time complexity of evaluating a CNN is $O(N_{inputs})$, a substantial improvement over the $O(N_{inputs}^2)$ complexity of evaluating a fully connected neural network, assuming a typical case where the fully connected network has a similar number of neurons in a hidden layer as the number of inputs [121].

TABLE 5.1. Asymptotic speedup using CNNs for various cases.

Solver	**MoM**	**FEM**
Direct Solver	$speedup \propto \frac{1}{\gamma^3}$	$speedup \propto \frac{1}{\gamma^3}$
Iterative Solver	$speedup \propto \frac{1}{\gamma^2}$	$speedup \propto \frac{1}{\gamma^2}$

We propose to use a deep convolutional neural network to predict the solution weights, x, to complicated FEM, MoM, and FD problems given only the solution to a computationally inexpensive analogue of the problem solved on a reduced basis. Most simply, the reduced basis, $\check{F}$, would constitute a small subset of the complete basis, F. A reduced basis is easy to conceive of for FEM and MoM, especially using higher-order bases. We can simply reduce the number of polynomial basis functions allocated to each element. A reduced basis is less obvious at first for FD, since we typically do not consider the concept of basis functions when working with finite difference techniques, but rather sample points. However, we can consider the sample points used in FD as a weighting of Dirac-delta basis functions centered on the spatial locations of the sample points. In this sense, FD is a special case of FEM given a particular choice of basis and a particular

quadrature rule. From this perspective, a reduced basis is easy to conceive of: a coarser grid, the sample points of which are a subset of the original grid.

If a CNN can predict x from $\check{x}$, the solution to the problem discretized using basis $\check{F}$ containing $\check{N} = \gamma N, \gamma \in (0,1]$ basis functions, then the achieved speedup of solving for x is asymptotically proportional to expressions given in Table 5.1.

A difficulty with CNNs and applicability of their results for certain tasks is their black box nature. A trained CNN is a purely empirical model, typically with little theoretical underpinning nor theoretical guarantee on the accuracy of its output. To counteract this to quickly obtain accurate FEM, FD, and MoM solutions, we propose to use x predicted by the network not as the final solution, but rather to generate a set of macro basis functions that can be used to re-solve the problem at comparable accuracy to the approximation using F, but instead using a substantially smaller number of basis functions. We define a macro basis function in general as a linear combination of basis functions from F

$$f_{macro} = \sum_{i \in [1..N]} \alpha_i f_i \tag{5.3}$$

where α coefficients are specific to a particular macro basis function. We denote the set of macro basis functions F_{macro}.

By this approach, we can guarantee that the solution obtained using the CNN's prediction exactly and rigorously solves a weak formulation of the problem. We also guarantee that the solution satisfies the boundary conditions of the problem by careful choice of the original basis function in F and careful definition of the macro basis functions. We denote by $F_{boundary} \subset F$ the set of basis functions in the original basis that are nonzero wherever a boundary condition is imposed in the original problem. We also denote by $F_{remainder} = F - F_{boundary}$ the remaining

basis functions in the original basis. We then place the additional constraint on any macro basis function that it contains no contribution from basis functions in $F_{boundary}$

$$f_{macro} = \sum_{I_{macro}} \alpha_i f_i \,, \qquad (5.4)$$
$$I_{macro} = \left\{ i \in [1..N] \middle| \, f_i \notin F_{boundary} \right\}$$

We then solve the problem with the modified basis $\bar{F} = F_{boundary} \cup F_{macro}$. The macro basis function approach scales as before but with specification that $\gamma = |\bar{F}|/|F|$ where vertical brackets denote set cardinality.

5.3 Numerical Results and Discussion

We demonstrate here the usefulness of the proposed macro basis function approach for FEM. We randomly generated a dataset of 1000 lossy dielectric slab scattering FEM problems as in chapter one. Both slab location and slab material parameters were varied and randomly sampled from a uniform distribution, with slab location varied over 3 wavelengths, slab real relative permittivity varied between 1 and 10, and slab imaginary relative permittivity varied between 0j and -5j. The domain was PML-truncated. F for this test was a set of 6$^{\text{th}}$-order polynomial basis functions as defined in [1]. $\check{F}$ comprised only the linear subset of F, giving $\gamma = 0.33$. A simple feedforward CNN was trained on all 1000 examples to predict x from $\check{x}$. We used a simple four-layer CNN with 3×1 filters and 64 filters per layer. Convolution was performed only in the spatial dimension of the data, with basis functions of different orders encoded as different input or output channels. N_{inputs} for this network was 27, and the network had 162 outputs. For each element, a single macro basis function was constructed as a linear combination of higher-order basis functions with α coefficients equal to predicted complex solution weights in x. To validate, 1000 new problems were generated from the same distribution. For each, the problem was solved using $\bar{F}$

predicted by the network. Fig. 5.1 compares the solution obtained using $\bar{F}$ to the solution using F and to a 2nd-order solution for a typical example. Note that the 2nd-order basis and $\bar{F}$ have the same number of basis functions.

We see poor agreement between the 2nd-order solution and the 6th-order solution. Meanwhile, despite yielding the same linear system size and structure as the 2nd-order solution, the solution using $\bar{F}$ agrees well with the full 6th-order solution. To further demonstrate the strength of the proposed macro basis function approach, we used the raw output of the neural network (a prediction of the solution weights) to plot a "naïve" predicted solution without re-solving the system. This serves as a benchmark for the somewhat common approach in existing literature to predict a solution directly. Fig. 5.2 compares this solution with the actual solution and the predicted solution using the proposed method.

Although the naïve predicted solution agrees with the actual solution somewhat better than the 2nd-order solution, we see various inaccuracies in the network's prediction manifest themselves directly in the form of amplitude errors (for instance in the imaginary plot around 2.5 meters) and even substantial errors in solution behavior (around 1.3 meters in the imaginary plot). Because we have taken the solution predicted directly by the neural network at face value as the naïve predicted solution, these errors go uncorrected. Meanwhile, our proposed macro basis function approach compensates for inaccuracies and misconceptions of the network to produce a substantially more accurate solution with the same number of basis functions as the 2nd-order solution.

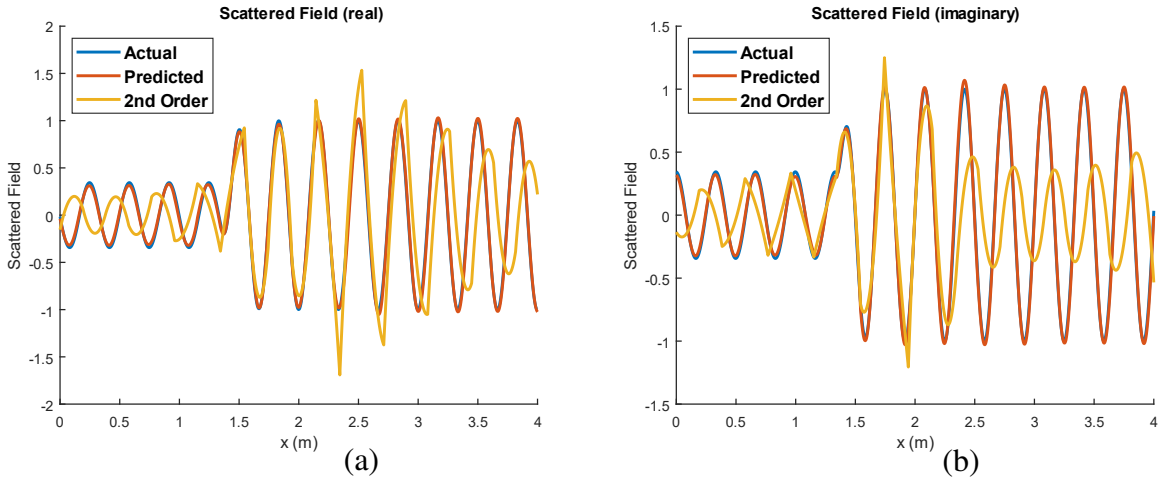


Fig. 5.1. Scattered field (z-directed) solution comparison between weak solution obtained using predicted macro basis functions, actual solution, and 2nd order weak solution: (a) real component and (b) imaginary component. Predicted solution using the proposed macro basis function approach agrees almost perfectly with the actual solution, despite using only 14% as many basis functions. The 2nd-order solution shown uses the same number of basis functions as the predicted solution but does not agree with the actual solution.

Fig. 5.3 shows the root mean square (RMS) error with respect to the 6th-order solution for all 1000 validation problems. The validation problem from Figs. 5.1 and 5.2 was chosen to fall at the peak of the real predicted RMS error histogram, i.e., an example with typical error.

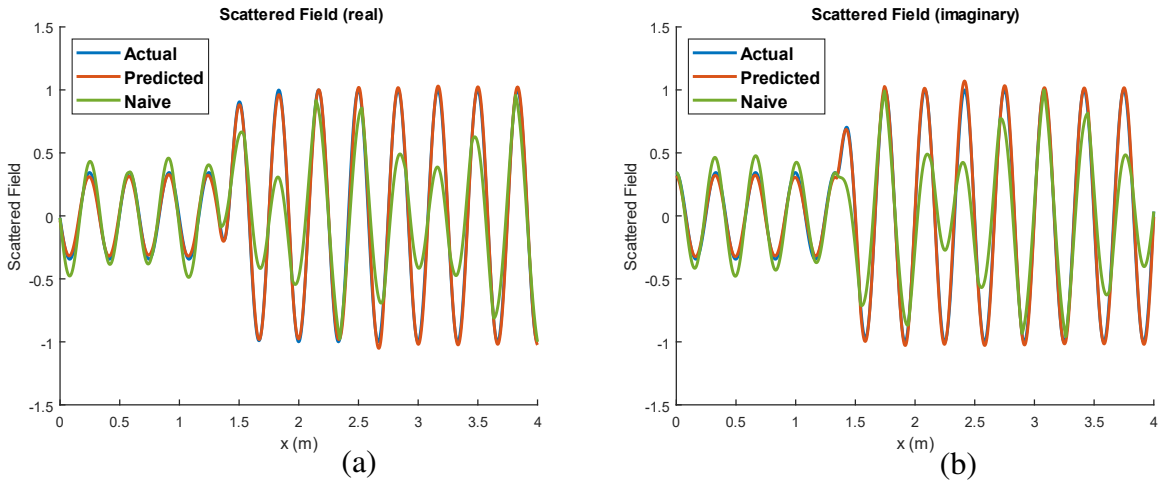
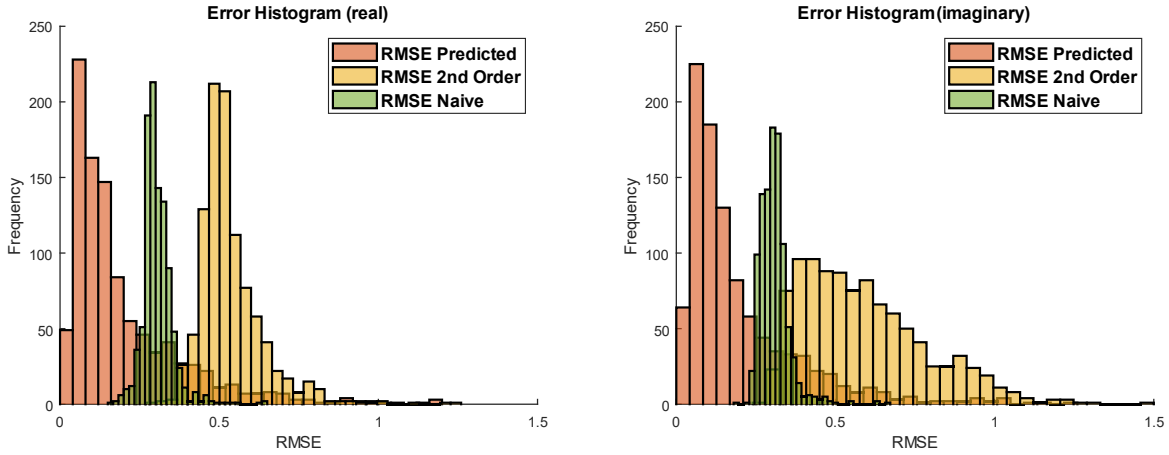


Fig. 5.2. Scattered field (z-directed) solution comparison between weak solution obtained using predicted macro basis functions, actual solution, and naïve predicted solution: (a) real component and (b) imaginary component. Naïve predicted solution is obtained by plotting the solution directly predicted by the network without the macro-basis function approach.

The error at the peak of the histograms for the solutions obtained using the predicted macro basis functions is approximately an order of magnitude less than that for the 2nd order solutions. The proposed method also dominates the naïve predictive approach. In no case does the naïve approach have error equal to or lower than the peak of the distribution for the proposed method. This demonstrates the potential of the proposed predicted macro basis function approach over both neural network predicted solutions and variational method solutions in isolation.

We also present a direct computation time comparison between the 2nd-order, 6th-order, naïve, and proposed macro basis function methods. Table 5.2 gives the time taken by our implementation of each method to solve 1000 randomly generated validation problems. Note that direct time comparisons are highly implementation-dependent, so, although we believe our implementations are efficient, we present Table 5.2 with that in mind.



(a) (b)

Fig. 5.3. Real (a) and imaginary (b) RMS error histograms for all 1000 validation problems. Predicted case is for the proposed macro basis function approach. Naïve case gives the error of the solutions directly predicted with the network using no macro basis function approach (the typical, existing approach). 2nd-order case serves as a comparison to the proposed approach. The 2nd-order case and predicted case use the same number of basis functions, but the proposed method yields error an order of magnitude lower.

TABLE 5.2: Direct time comparisons for 1000 random problems.

Method	**2nd Order**	**6th Order**	**Naïve Method**	**Proposed Method**
Time (ms)	41.9	439.7	110.9	152.1

5.4 Conclusion

This chapter has introduced a robust data-enabled machine learning approach to accelerate CEM and CSE variational methods like FEM, MoM, and FD techniques. Predicting macro basis functions by which a weak formulation can be solved rigorously, the described approach substantially reduces the number of unknowns required to solve a given problem, offering an asymptotic speedup over pure FEM, MoM, or FD solutions while maintaining the rigorousness, accuracy, and broad applicability of these methods. The described method has been demonstrated on a class of FEM problems and rigorously validated on a set of 1000 unseen validation problems. Compared to the naïve approach of predicting the solution directly, our method obtains substantially higher accuracy, its solution typically almost indistinguishable from the true solution. Our method also obtains far higher accuracy than a typical (no predicted macro basis) solution with an equal number of unknowns.

6 PREDICTING MACRO BASIS FUNCTIONS FOR METHOD OF MOMENTS SCATTERING PROBLEMS

6.1 Introduction

Neural networks have seen widespread use in various areas of science and engineering in recent years due to increased computing power and improved network architectures. Neural networks excel at learning complicated, non-obvious correlations between data to make useful predictions for tasks including image classification, computer vision, text processing, image synthesis, text synthesis, robotic motion planning, language processing, and more. See [119] for an in-depth overview of neural network applications and the current state of the art. Few studies have sought to apply machine learning to computational electromagnetics (CEM) and computational science and engineering (CSE), especially in the modern context of deep learning [104]–[110].

In contrast to numerically rigorous CEM methods, neural networks take a strongly empirical approach to making predictions, with little mathematical underpinning other than that of the backpropagation and gradient descent typically used to train them. This is not to downplay the strength of neural networks for many tasks—they excel, given sufficiently high quality training data, at distilling a process or dataset to its fundamental, most crucial components, from which the network can efficiently (computationally speaking) make inferences that are often times of substantially higher quality than those of more-rigorous techniques. However, neural networks are effectively black boxes once trained, and their empirical nature makes them unpredictable and sometimes subject to substantial error for cases not similar to their training set [119], [122]–[128]. This makes their direct application to CEM challenging and somewhat questionable. Indirect application, on the other hand, offers a possible avenue for more-rigorous use of neural networks

in CEM. Through indirect application, we seek to couple the efficient predictive capabilities of neural networks with the robustness and rigorousness of existing CEM methods.

In general, out of various modern network architectures, deep residual networks have shown excellent performance for highly complicated tasks [119], [120], [111] and can be structured for regression problems to be fully convolutional. Fully convolutional neural networks can accept arbitrarily sized gridded inputs, a useful trait for CEM applications requiring arbitrarily sized meshes. For simplicity, we do not exploit this trait here, using only fixed-sized meshes, but we consider it prudent to perform the present study with the reality of arbitrary-sized meshes in mind.

Most prior work in applying machine learning to CEM and CSE has focused on predicting quantities derived from a numerical solution given a problem description, often material parameters throughout a domain and an excitation [104]–[108]. Others have focused on predicting the numerical solution itself [109]–[110], effectively attempting to replace the existing methodology (e.g. MoM) with learned models.

In chapter five, we proposed a robust application of neural networks for accelerating variational methods like finite element method (FEM), method of moments (MoM) and finite difference (FD) method for CEM and CSE problems. Rather than predicting a solution directly, our approach uses networks to predict sets of macro basis functions on which the problem is then re-solved. We demonstrated in chapter five that such an approach meaningfully reduces the number of unknowns (and therefore runtime) of simple one-dimensional (1-D) FEM scattering problems while producing solutions of comparable accuracy to classical solutions of substantially higher dimensionality. Although the 1-D FEM slab scattering problem set explored in chapter five was ideal for proof of concept, conclusions about the usefulness and versatility of our method

cannot be drawn from such a simple case. In this chapter, we explore the ability of neural networks to predict macro basis functions for more-complicated sets of 3-D MoM perfect electric conductor (PEC) scattering problems. In fact, the main challenge in MoM for radiation and scattering applications relates to the appearance of large matrices resulting from discretization of large and complicated electromagnetic problems. Any significant and systemic reduction of the number of basis functions and unknowns in MoM computations is therefore highly beneficial. We consider a double-higher-order surface integral equation (SIE) approach [7], with arbitrarily curved surface elements, higher order polynomial basis functions, and complex-valued solution weights, to be a general and challenging example on which to test and demonstrate the proposed method for MoM. The MoM-SIE problem datasets in this study are: a set of 10,000 warped cylinders, tori, plates, and spheroids, with electrical sizes ranging over two orders of magnitude subject to incident plane waves from varying directions; and a comparably simple set of 1,000 bent plates. These datasets of scatterers with diverse shapes and both large-scale and small-scale features capture a greater variety and complexity of the hypothetical set of all scattering problems of interest to CEM practitioners, advancing toward validation and a generalizable application of our method.

Our predicted macro basis function approach, in contrast to the naïve approach of predicting the solution directly, leverages learned models to reduce the dimensionality of CEM and CSE problems, then solved by classical methods like MoM. This is inspired by some significant breakthroughs (outside of CEM and CSE) in application of neural networks to completing challenging tasks with high accuracy. Such breakthroughs, like [111] and [112], have used existing mathematically formal methods, guided by the predictive capability of neural networks. We believe this is critical to the application of machine learning in most engineering contexts. Although we have found little existing research that couples neural networks with

variational methods in a broadly applicable way, [113]–[118] are in a similar vein of reasoning, applying networks to predict bulk material parameters for faster multi-scale FEM structural mechanics simulations.

The chapter is organized as follows. Section 6.2 outlines the theoretical background of our approach to predicting macro basis functions for method of moments scattering and radiation problems using deep learning. Section 6.3 describes and discusses generation, characteristics, and use of datasets for this study. This section also explains the implemented network architecture, network versions, and training and testing procedures. Section 6.4 gives extensive and clear numerical results which assess network performance and ability to accurately predict macro basis functions for 3-D MoM scattering problems. It provides comparisons between predicted and actual basis functions, error statistics, and representative examples for networks trained on simple and complicated datasets. Section 6.5 further evaluates application of networks to problem types on which they were not trained, establishing useful generalizability of networks' learned knowledge. Section 6.6 then summarizes the main conclusions of the study.

6.2 Background Theory and Context

As in chapter five, we consider a discretized linear integral or differential equation-based problem with solution S, set of basis functions F, and linear system of form $[A]x = b$, where $f_i \in F$ and x_i denote the i^{th} basis function and associated solution weight, respectively. The weak solution to such a problem with N basis functions is

$$\tilde{S} = \sum_{F} x_i f_i \approx S \tag{6.1}$$

We apply neural networks to predict solution weights, x, given the solution, $\check{x}$, to a computationally inexpensive analogue of the problem solved on a reduced basis, $\check{F}$, most simply, a small subset of

the complete basis, F. To compensate for errors in predicted solution weights, we use x predicted by the network not as the final solution, but to generate a set of macro basis functions that can be used to re-solve the problem. We define each macro basis function as some linear combination of basis functions from F,

$$f_{macro} = \sum_{F} \alpha_i f_i \tag{6.2}$$

To satisfy boundary conditions, we require careful definition of the macro basis functions. We denote by $F_{boundary} \subset F$ the set of basis functions in the original basis that directly contribute to satisfaction of boundary conditions and by $F_{remainder} = F - F_{boundary}$ the remaining basis functions from the original basis. We impose a simple constraint on each macro basis function:

$$\begin{aligned} &f_{macro} = \sum_{F_m \subset F_a} \alpha_i f_i, \\ &F_a \in \{F_{remainder}, F_{boundary}\} \end{aligned} \tag{6.3}$$

Note that this constraint is relaxed from that in chapter five, allowing macro basis functions to include $f_i \in F_{boundary}$ so long as they do not include $f_i \in F_{remainder}$. We then solve the problem with the modified basis $\bar{F} \subset F_{macro}$, a chosen subset of all macro basis functions using the network's prediction of x_i as α_i for each f_i present in a macro basis function. Our simple approach to forming macro basis functions is therefore limited by the network's ability to predict x accurately from $\check{x}$, so we seek to evaluate that ability in this chapter.

6.3 Datasets and Networks

Meshes used for general MoM problems are highly varied both geometrically and topologically, describing a broad range of surfaces and consisting of a variety of surface element types in a wide range of configurations. Our goal here is only to explore the ability of neural networks to predict x for complicated MoM scattering problems, so we chose to constrain all

models in our datasets to a single mesh topology: a 20×20 grid of rectangular surface elements. In the MoM implementation used to generate training data, we discretized the electric-field SIE over PEC surfaces [7], using generalized curved quadrilaterals of arbitrary geometrical orders k_u and k_v, shown in Fig. 6.1 [1],

$$\mathbf{r}(u,v) = \sum_{m=0}^{k_u} \sum_{n=0}^{k_v} \mathbf{r}_{mn} L_{k_u,m}(u) L_{k_v,n}(v), \quad -1 \le u, v, \le 1 \tag{6.4}$$

where L represents Lagrange interpolation polynomials and $\mathbf{r}_{mn}$ are the position vectors of interpolation nodes. We approximated the electric surface current, $\mathbf{J}_s$, over quadrilaterals by means of hierarchical vector basis functions of arbitrarily high current-expansion orders n_u and n_v [7] arranged in a maximally orthogonalized fashion [130]. For instance, the u-oriented current expansion over a quadrilateral in Fig. 6.1 is given by

$$\mathbf{J}_u(u,v) = \sum_{j=0}^{n_u} \sum_{k=0}^{n_v-1} \alpha_{u,jk} S_j(u) P_k(v) \frac{\mathbf{a}_u}{\mathfrak{I}} \tag{6.5}$$

where S and P are the respective divergence-conforming maximally orthogonal functions from [130], $\{\alpha\}$ are unknown current-distribution coefficients, $\mathfrak{I} = |\mathbf{a}_u \times \mathbf{a}_v|$ is the Jacobian of the covariant transformation, and $\mathbf{a}_u = \partial\mathbf{r}/\partial u$ and $\mathbf{a}_v = \partial\mathbf{r}/\partial v$ are unitary vectors along the parametric coordinates in Fig. 6.1. Note that the expansion is one order lower in the parametric direction perpendicular to its orientation. An analogous form exists for the v-directed expansion, $\mathbf{J}_v$.

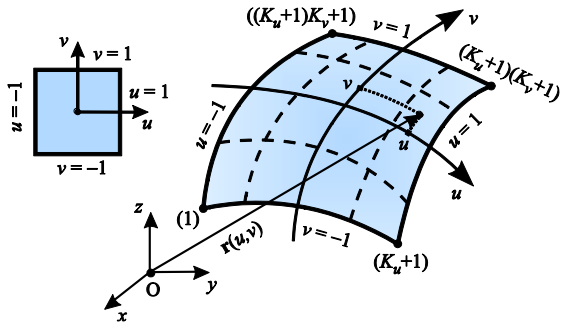


Fig. 6.1. Generalized curved parametric quadrilateral patch for higher order MoM-SIE modeling, defined in (6.4). Square parent domain is also shown.

In this study, each element was geometrically 2nd order [$k_u = k_v = 2$ in (6.4) and Fig. 6.1] and current expansion orders of 5 were chosen for all elements [$n_u = n_v = 5$ in (6.5)] for the complete basis F. Each model in our datasets therefore had the same number of elements and basis functions, all with the same relative ordering in two parametric dimensions. This allows us to ignore challenges associated with mapping surface currents on arbitrary meshes to network inputs (which we consider out of scope for the goals of this study), although we note some methods like [131] exist. Such a 20×20 grid of rectangular elements is surprisingly versatile and can be warped and glued to describe a variety of surface types using parametric mappings.

Fig. 2 shows representative objects from the complicated dataset of 10,000 warped plates, cylinders, tori, and spheroids. These objects vary in length scale from 0.1 to 10 wavelengths (λ), measured in an object's longest dimension, and have a wide distribution of aspect ratios and deformations on multiple scales. Fig. 6.3 shows representative objects from the simple dataset of 1,000 bent plates. These objects have simple deformation patterns and length scales on the order of a wavelength. All objects in the simple dataset shared the same orientation with respect to the incident field and the same aspect ratio (square) in the non-vertical directions. Note that objects in Fig. 6. 2 and Fig. 6.3 are rendered at the same scale, and all objects are shown from the direction of the incident plane wave used as excitation.

The basis function weights for each element can be displayed intuitively in a 6×5 grid. For symmetry, we present weights in a 6×6 grid for both u- and v-directed current expansions, with the rightmost column left empty (or set to zero) to represent the column of missing basis functions in the perpendicular direction due to the lower sum limit in (6.5). The complete basis F for each element therefore contains 60 basis functions (30 for $\mathbf{J}_u$ and 30 for $\mathbf{J}_v$). Per-element basis function grid patterns are shown in Fig. 6.4. Note that each element will have two such grids of complex-valued basis functions weights associated with it: one for u- and one for v-directed current expansions. Gray cells represent weights in $\check{F}$. Green, orange, and blue cells represent weights to be predicted.

Weights in the first two rows in Fig. 6.4 constitute $F_{boundary}$, so either the first or second row of weights are zero for elements without neighbors (present on plates and cylinders). For such elements, zero-valued weights in the A and B or C and D regions were still input to the network, but network predictions for I or J, appropriately, were ignored when computing error statistics (as they are necessarily set to zero when re-solving the system).

Functions in cells H, I, and J are those intended to be merged into macro basis functions weighted by network predictions. These regions were chosen for simplicity. If we were attempting to build a solution directly from the weights predicted by the network, we would desire a network that predicted these values on some common scale. However, we intend to use these weights to form macro basis functions of form (6.3) for re-solving of the problem on $\bar{F}$, so, ignoring numerical constraints, we do not care about the relative value of weights not in the same region.

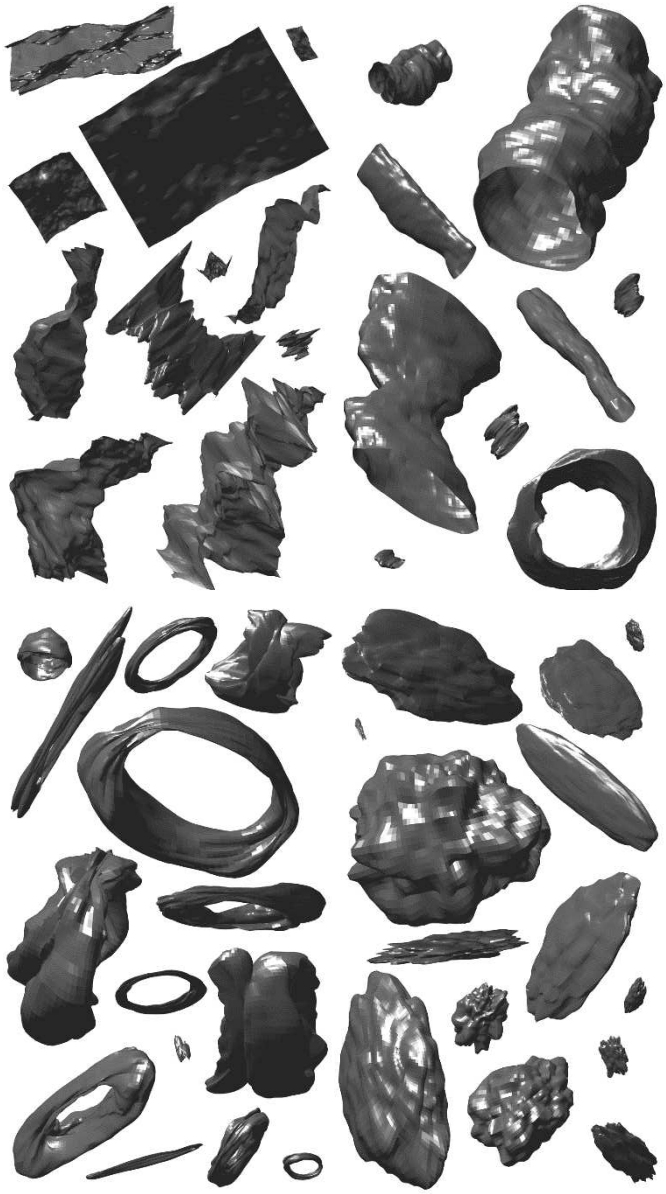


Fig. 6.2. Representative objects from the complicated dataset of 10,000 randomly generated PEC scatterers, with length variation from 0.1 to 10 λ. All objects are at the same scale. These objects fit into 4 categories among which the dataset was uniformly divided: plate-like, cylinder-like, torus-like, and spheroid-like objects. All objects had varying degrees of randomly generated large-scale and small-scale features added to their surfaces. MoM-SIE analysis was performed for these surfaces, discretized using a 20×20 grid of 2nd-order quadrilateral elements ($k_u = k_v = 2$, Fig. 6.1) with 5th-order basis functions [$n_u = n_v = 5$, see (6.5)] constituting the complete basis, F, in response to an incident plane wave from a randomly-generated direction (one direction per model). Each object is viewed from the direction of the incident excitation. Plates shown top left; cylinders top right; tori bottom left; spheroids bottom right.

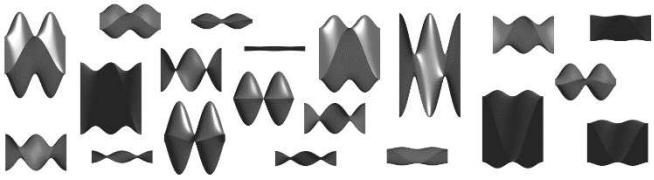


Fig. 6.3. Representative objects from the simple dataset of 1,000 bent plates, with length scales on the order of λ and simple deformation patterns. Each object is viewed from the direction of the incident wave.

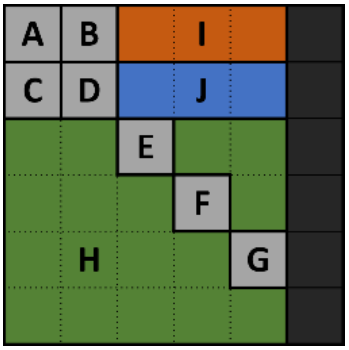


Fig. 6.4. Basis function weight grids. Function weights in A-G were given to network as input. Weights in regions H, I, and J were predicted by the network.

Therefore, we normalized all weights within regions H, I, and J such that the value of highest magnitude in each region has unit magnitude. Basis functions in regions A through G, corresponding to functions in $\check{F}$, were included uncombined in $\bar{F}$ for examples in this chapter. This is a degenerate case of (6.3) where only one α_i is nonzero per f_{macro}, rendering such f_{macro} equivalent to $f_i \in \check{F}$.

The same network architecture was used for both the simple and complicated datasets. The architecture consisted of a simple 16-block residual network [120] with no batch normalization and a hyperbolic tangent nonlinearity after the final convolutional layer. We used 128 filters per layer. This architecture was not optimized for the problem but was sufficient for proof of concept. One version of the network was trained exclusively on 800 examples from the simple dataset, the other 200 held-out for testing. We denote this Network A. Another version, Network B, was trained

exclusively on 9,800 examples from the complicated dataset, evenly partitioned between the four subcategories. The remaining 200 objects were held-out as a test set. For both networks, the test sets were drawn randomly. There was no overlap between the training dataset and the test dataset. No object appeared in both the test and training datasets, meaning objects in the test dataset, not just their solutions for a particular excitation, had never been seen by the network. Both networks were trained using the Adam optimization algorithm [132] with a batch size of 10 and learning rate of 1e-4 for 150 and 500 epochs, respectively. Interestingly, we noted substantial overfitting for longer training runs, indicating these networks will benefit from larger training datasets.

Input to each network was 20×20×28, the 28 channels corresponding to real and imaginary values for both u- and v- directed basis functions in cells A-G. The 20×20 spatial grid corresponds to the 20×20 grid of quadrilateral elements comprising each mesh. Similarly, the output was 20×20×92, the 92 channels corresponding to real and imaginary values for both u- and v-directed basis functions in H, I, and J. Network loss was computed as mean squared error between predicted and actual values.

6.4 Numerical Results and Discussion

To accurately assess network performance, both networks were evaluated on their test sets after training. Both networks were also tested on the opposite test set. To present results, we split the 20×20×92 output for each test problem into 400 1×1×92 vectors of element-wise weights. We then further subdivided each vector into four 1×1×23 vectors, for real u, imaginary u, real v, and imaginary v, respectively. Each vector was then rearranged and padded into a 6×6 grid corresponding to weight assignments for H, I, and J in Fig. 6.4. Values in these grids corresponding to input weights A-G and values in the right-most column were set to zero for presentation of

representative examples. Error statistics were computed between these grids and corresponding grids constructed from the true solution weights (normalized in the same way). Fig. 6.5 shows 15-bin root mean square error (RMSE) histograms. Weight boundaries are the same for all four sub-plots. Fig. 6.5(a) shows application of Network A to the simple test set, while Fig. 6.5(b) shows application of Network A to the complicated test set. Similarly, Figs. 6.5(c) and 6.5(d) show application of Network B to the complicated and simple test sets, respectively.

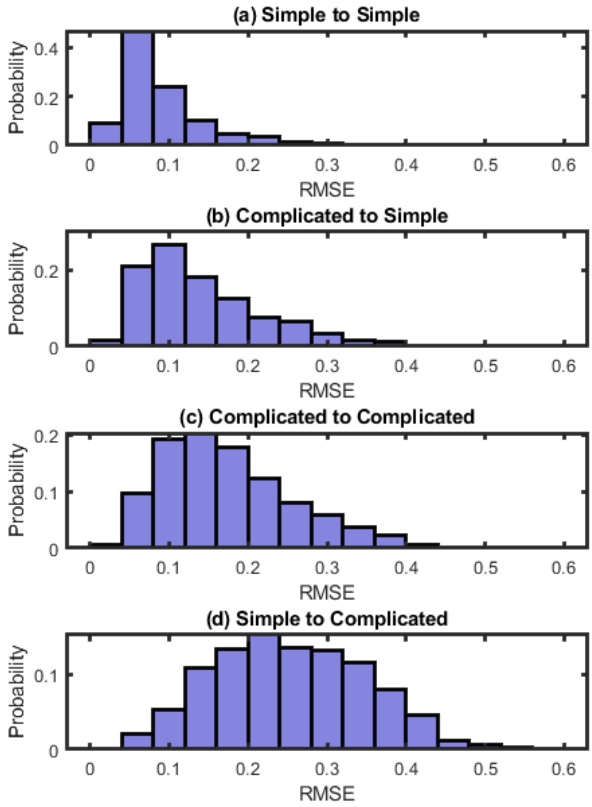


Fig. 6.5. 15-bin RMS error (of basis function weights prediction with respect to actual weights) histograms for both networks on both validation datasets: (a) Network A on the simple set (Fig. 6.3); (b) Network B on the simple set; (c) Network B on the complicated set (Fig. 6.2); (d) Network A on the complicated set.

Figs 6.6 and 6.7 show representative examples for each bin, corresponding to histograms in Fig. 6.5. Fig. 6.6 shows bin examples for Figs. 6.5(a) and (b), while Fig. 6.7 shows bin examples for Figs. 6.5(c) and (d). For each, the bin of highest probability (mode) is highlighted green.

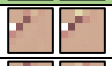
	Network A		Network B	
Bin#	Predicted/Actual	Error	Predicted/Actual	Error
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				

Fig. 6.6. Representative bin examples for histograms in Fig. 6.5(a), left, and Fig. 6.5(b), right, for testing of Network A and Network B, respectively, on the simple dataset (Fig. 6.3): predicted weights, actual weights, and errors for per-element basis functions (Fig. 6.4). Some entries were left blank for Network A on this dataset, as no examples fell in bins 10-11 for this case. Neither case produced examples that fell in bins 12 or higher, so these bins were omitted from this plot. The bin of highest probability is highlighted green.

Network A, trained only on the simple dataset, was able to predict weights for objects in the simple test set accurately. This is especially apparent in the representative mode example (green) for Network A in Fig. 6.6. Network B, trained on the complicated dataset, performed similarly well for the complicated test set, although with somewhat higher error overall. Both results are expected, but we are pleased with the accuracy with which the networks predicted basis function weights on their corresponding test sets. More interestingly, however, were the cross-validation results. Despite the complicated dataset containing no objects similar to those found in the simple dataset (evident by comparison of the plates in Fig. 6.2 to those in Fig. 6.3), Network B performed better on the simple test set than it did on its own (the complicated) test set.

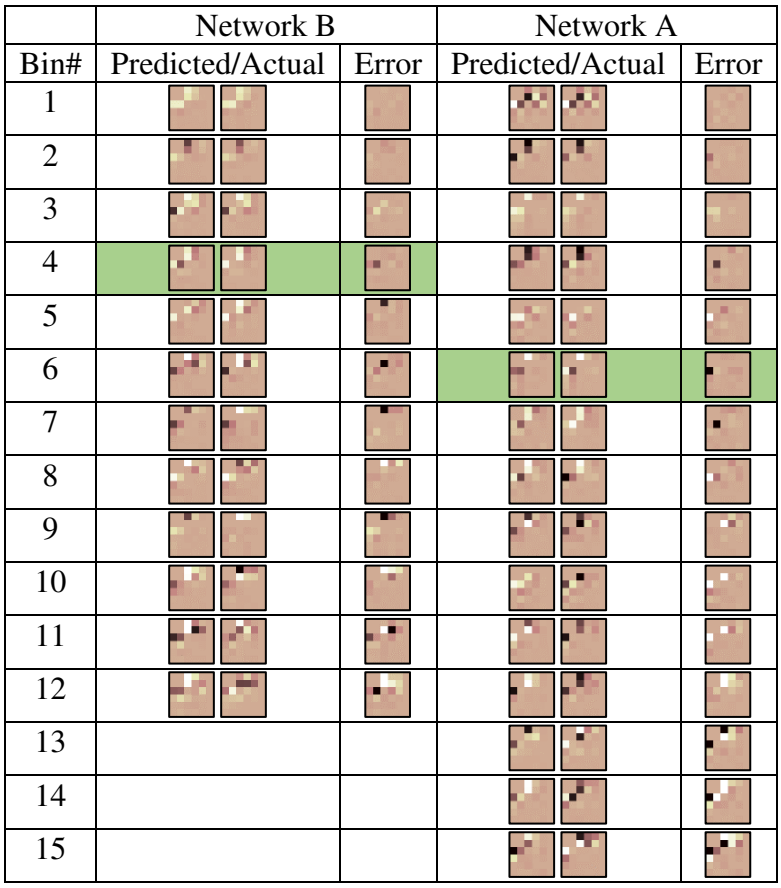


Fig. 6.7. Similar to Fig. 6.6. Representative bin examples for histograms in Fig. 6.5(c), left, and Fig. 6.5(d), right, for validation of Network B and Network A, respectively, on the complicated dataset (Fig. 2). Some entries were left blank for Network B on this dataset, as no examples fell in bins 13-15 for this case.

This indicates that Network B learned generalizable knowledge applicable beyond the complicated dataset on which it was trained. Network A clearly also learned generalizable knowledge, demonstrated in Figs. 6.5(d) and 6.7, but at far higher error.

6.5 Conclusion

This chapter has presented development, implementation, analysis, and validation of a machine learning approach to predicting macro basis functions for method of moments 3-D scattering and radiation problems using deep neural networks. We generated and used two datasets

of MoM PEC scattering problems, a complicated dataset of 10,000 randomly warped cylinders, tori, plates, and spheroids of high variability of electrical sizes and wide distribution of aspect ratios and deformations on multiple scales; and a comparably simple set of 1,000 bent plates, simulated by the double-higher-order MoM-SIE technique. Two deep residual networks were trained on the two respective datasets. To accurately assess network performance, both networks were evaluated on their test sets after training, as well as on the opposite test set for cross-validation.

The results have shown that the networks are able to accurately predict basis function weights on their own test sets. Even more importantly, cross-validations have demonstrated networks' ability to learn generalizable knowledge applicable beyond the types of objects on which they were trained.

Overall, the numerical examples have shown that deep neural networks are a promising approach to macro basis function prediction for MoM scattering and radiation problems. The performed study has demonstrated that these networks have the capacity to understand complicated, widely varying 3-D MoM problems, which, although intuitively plausible, is far from an obvious conclusion without such a study. In particular, the predicted macro basis function approach leverages learned models to considerably reduce the total number of basis functions and number of unknowns in MoM computations and thus MoM matrix size. Meanwhile, the appearance of large matrices generally constitutes the main problem in MoM for practical radiation and scattering applications. While the implemented and analyzed networks have produced highly encouraging results in this study, especially for cross-validation, larger training datasets will improve reliability for general antenna and scattering problems.

CONCLUSION

This dissertation has presented several improvements, both theoretical and empirical, to the state of the art in computational electromagnetics solver efficiency. As CEM problem sizes have continue to grow, so too have the demands placed on the techniques at the core of tools used by CEM practitioners, spurring hybridization of previously disparate methods to cover the wide range of problem types and scales of interest. It has therefore been of increasing importance that improvements to CEM methods consider the plethora of CEM techniques in common use. Accordingly, this dissertation has explored efficiency gains for the two most common classes of CEM solver: full-wave and ray-based.

In chapter one, we applied and demonstrated the use of adjoint-based a posteriori error estimation and adaptive discretization refinement to CEM using a simple, one-dimensional higher order FEM scattering example as an ideal testbed. This chapter demonstrated how the adjoint solution can be leveraged to efficiently compute multi-dimensional gradient information with a static (with respect to the dimensionality of the gradient) number of solves. This is immensely useful in nearly all modern contexts where CEM is used. Efficient computation of gradients was demonstrated for the one-dimensional lossy dielectric slab scattering problem using real and imaginary relative permittivity values as the problem dimensions with respect to which gradient information was computed. We further showed how QoI responses can accurately be reconstructed using only a few forward and adjoint solves from which gradient information is obtained, allowing efficient construction of surrogate models for several engineering contexts, including optimization and sensitivity analysis. In addition to leveraging adjoint methods for the computation of gradient information, we demonstrated how a higher order adjoint solution can be used to compute accurate, signed error contribution estimates for every element in a domain. We then described and

experimentally validated how such elementwise error contribution estimates can be used not only to quantify solution error, but also to adaptively refine a problem discretization to achieve far higher solution accuracy for only a modest increase in discretization dimension. These findings indicate that adjoint methods, previously rarely studied in CEM contexts, can benefit accuracy and efficiency of CEM solves as they have partial differential equation solvers in a variety of other disciplines like structural mechanics and computational fluid dynamics.

In chapter two, we applied the adjoint methods from chapter one to a more-complicated three-dimensional lossy dielectric sphere scattering problem set—a 3D analogue of the one-dimensional slab scattering problem set explored in chapter one. In chapter one, we claimed that the benefits of adjoint methods to CEM solver efficiency may be compounded for large, multi-solve problems in which many related sub-problems must be solved accurately. Such multi-solve problems are ubiquitous in CEM and include radar cross-section computation, design optimization, and uncertainty quantification. For such problems, we note that, in addition to adjoint methods enabling efficient computation of useful gradient information, the elementwise error contribution estimates they provide may also be applicable across sub-problems for dynamic discretization refinement without sub-problem repetition. We remarked in chapter two how such efficiency gains for multi-solve problems are contingent on the relatedness of EECEs between sub-problems. Chapter two, therefore, substantiated our claims about efficiency gains for multi-solve problems by demonstrating that EECEs can be considerably more correlated between sub problems than even QoIs or their derivatives. We further demonstrated that, for our example problem set, EECEs remain highly correlated between sub-problems across their spectrum of magnitudes, indicating that application of EECE-based refinement schemes across sub-problems is likely feasible.

In chapter three, we addressed one of the major barriers to application of adaptive refinement schemes to real-world CEM problems solved on surface discretizations: the difficulty of producing and manipulating surface meshes at high quality. Leveraging the discrete surface Ricci flow, we described a technique by which an arbitrary surface can be conformally mapped to a simple parametric domain, allowing complicated meshing operations to be performed in e.g. a rectangular domain rather than on the surface of a fighter jet. The conformal nature of the produced mappings maintains relative element shape between domains, allowing high quality meshes in the parametric domain to be mapped to high quality meshes of the original surface. We further defined a simple iterative adaptive refinement technique to mitigate area distortion inherent to conformal mappings and provided a variety of examples for common CEM mesh types. In addition to enabling application of adaptive refinement schemes, like those outlined in chapter one, to complicated surface meshes, the techniques introduced in chapter three also allowed, for the first time, application of highly efficient double-higher-order quadrilateral surface element-based techniques to arbitrary surfaces. Such techniques have been shown, for geometrically simple test problems like spheres and cubes, to greatly reduce the number of unknowns required to meet a certain error tolerance. The work in chapter three, therefore, will allow the efficiency gains of double-higher-order techniques to finally be realized for realistically complicated surfaces.

Chapter four addressed scaling problems of the most popular large-domain simulation technique in CEM: SBR. The chapter discussed how data dependencies inherent to shooting bouncing ray double count removal hinder efficient implementation of SBR on parallel computing platforms and therefore SBR's scalability, despite perfect parallelism of all other aspects of the method. To remedy this, we proposed in chapter four the concept of non-self-adjacent ray classes. By partitioning the set of rays into NSA ray classes, we showed in chapter four that the data

dependency can be circumvented, so long as the number of rays to be processed is greater than three times the number of executing threads. Chapter four provides a detailed theoretical description of our NSA ray classing approach and subsequently provides numerical results demonstrating the efficiency gains achievable using NSA ray classes for an example problem, implementation, and hardware.

Chapter five returned to efficiency gains for variational methods in CEM including MoM, FEM, and FD, this time from a semi-empirical perspective. In contrast to the formal techniques in chapters one and two, which leveraged information in a higher order adjoint solution to produce error information from which the discretization could be refined (or coarsened) to efficiently allocate dimensions in the problem space, chapter five described a technique by which the dimension of the problem space can be substantially reduced a priori using prior knowledge. Specifically, the technique outlined in chapter five leverages the predictive capability of deep neural networks to generate a set of macro basis functions on which to project the problem, achieving accurate solutions at substantially lower computational cost than standard bases. In chapter five, we demonstrated this technique for the same type of lossy dielectric slab scattering problem used in chapter one and provided several error statistics for its application to a set of 1,000 such problems. Results indicated the predicted macro basis function approach was highly effective at producing efficient bases on which to solve a given lossy dielectric slab scattering problem.

Chapter six explored the feasibility of applying the predicted macro basis function approach to substantially more-complicated CEM problems in the context of MoM. Although results presented in chapter five were encouraging, their usefulness was limited to proof-of-concept due to the simplicity of the lossy dielectric slab scattering problem set. Real-world CEM problems of interest to CEM practitioners and engineers are more complicated, describing

propagation phenomena in the presence of objects like printed circuit boards, aircraft, and mine tunnels. To better capture this variety, we presented in chapter six a dataset of 10,000 randomly generated PEC scatterers with varied surface features over a broad range of shapes and sizes. In addition, we presented a comparably simple set of 1,000 warped PEC plates. One scattering solution was computed for each object in response to an incident plane wave. As in chapter five, deep neural networks were trained to predict macro basis functions from these solutions. One network was trained per dataset. Chapter six showed that, despite the complexity of the datasets tested, networks were able to predict solutions and therefore predict macro basis functions at high accuracy, demonstrating the applicability of our predicted macro basis function approach to complicated problems.

REFERENCES

[1] B. M. Notaroš, "Higher order frequency-domain computational electromagnetics," Special Issue on Large and Multiscale Computational Electromagnetics, IEEE Transactions on Antennas and Propagation, vol. 56, no. 8, pp. 2251-2276, August 2008.

[2] J. M. Jin, K. C. Donepudi, J. Liu, G. Kang, J. M. Song, and W. C. Chew, "High-Order Methods in Computational Electromagnetics," in Fast and Efficient Algorithms in Computational Electromagnetics, W. C. Chew et al, Ed. Norwood, MA: Artech House, 2001.

[3] B. M. Kolundzija and A. R. Djordjević, "Electromagnetic Modeling of Composite Metallic and Dielectric Structures", Norwood, MA: Artech House, 2002.

[4] A. F. Peterson, "Mapped Vector Basis Functions for Electromagnetic Integral Equations", Morgan & Claypool Publishers, 2006.

[5] 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, March 1997.

[6] G. Kang, J. Song, W. C. Chew, K. C. Donepudi, and J. M. Jin, "A novel grid-robust higher order vector basis function for the method of moments," IEEE Transactions on Antennas and Propagation, vol. 49, pp. 908-915, June 2001.

[7] M. Djordjevic and B. M. Notaros, "Double higher order method of moments for surface integral equation modeling of metallic and dielectric antennas and scatterers," IEEE Transactions on Antennas and Propagation, vol. 52, no. 8, pp. 2118-2129, August 2004.

[8] E. Jørgensen, J. L. Volakis, P. Meincke, and O. Breinbjerg, "Higher order hierarchical Legendre basis functions for electromagnetic modeling," IEEE Transactions on Antennas and Propagation, vol. 52, pp. 2985-2995, November 2004.

[9] J. P. Webb, "Hierarchical vector basis functions of arbitrary order for triangular and tetrahedral finite elements", IEEE Transactions on Antennas and Propagation, vol. 47, no. 8, pp. 1244-1253, August 1999.G. Kron, Diakoptics: The Piecewise Solution of Large Scale Systems, MacDonald, 1963.

[10] M. M. Ilic and B. M. Notaros, "Higher order hierarchical curved hexahedral vector finite elements for electromagnetic modeling," IEEE Transactions on Microwave Theory and Techniques, vol. 51, no. 3, pp. 1026-1033, March 2003.

[11] M. M. Ilic and B. M. Notaros, "Higher order large-domain hierarchical FEM technique for electromagnetic modeling using Legendre basis functions on generalized hexahedra," Electromagnetics, vol. 26, no. 7, pp. 517-529, October 2006.

[12] M. M. Ilić, M. Djordjević, A. Ž. Ilić, and B. M. Notaroš, "Higher order hybrid FEM-MoM technique for analysis of antennas and scatterers," IEEE Transactions on Antennas and Propagation, vol. 57, pp. 1452-1460, May 2009.

[13] E. M. Klopf, N. J. Sekeljic, M. M. Ilić, and B. M. Notaroš, "Optimal Modeling Parameters for Higher Order MoM-SIE and FEM-MoM Electromagnetic Simulations," IEEE Transactions on Antennas and Propagation, 60(6), 2790-2801, 2012

[14] D. E. Keyes, L. C. McInnes, C. Woodward, W. Gropp, E. Myra, M. Pernice, J. Bell, J. Brown, A. Clo, J. Connors, E. Constantinescu, D. Estep, K. Evans, C. Farhat, A. Hakim, G. Hammond, G. Hansen, J. Hill, T. Isaac, X. Jiao, K. Jordan, D. Kaushik, E. Kaxiras, A. Koniges, K. Lee, A. Lott, Q. Lu, J. Magerlein, R. Maxwell, M. McCourt, M. Mehl, R. Pawlowski, A. Peters Randles, D. Reynolds, B. Riviere, U. Ruede, T. Scheibe, J. Shadid, B. Sheehan, M. Shephard, A. Siegel, B. Smith, X. Tang, C. Wilson, and B. Wohlmuth, "Multiphysics simulations: Challenges and opportunities," International Journal of High Performance Computing Applications, vol. 27, no. 1, pp. 4-83, 2013.

[15] D. Estep. "A posteriori error bounds and global error control for approximations of ordinary differential equations." SIAM J. Numer. Anal., 32:1–48, 1995.

[16] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson. Introduction to adaptive methods for differential equations. Acta Numerica, pages 105–158, 1995.

[17] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson. Computational Differential Equations. Cambridge University Press, New York, 1996.

[18] M. Giles and E. Suli, "Adjoint methods for PDEs: A posteriori error analysis and postprocessing by duality," Acta Numerica, pp. 145-236, 2002.

[19] R. Becker and R. Rannacher, "An optimal control approach to a posteriori error estimation in finite element methods," Acta Numerica, pp. 1-102, 2001.

[20] W. Bangerth and R. Rannacher. Adaptive Finite Element Methods for Differential Equations Birkhauser, Boston, 2003.

[21] M. H. Bakr, Nonlinear Optimization in Electrical Engineering with Applications in MATLAB, IET, September 2013.

[22] M. H. Bakr and N. K. Nikolova, "An adjoint variable method for time domain TLM with fixed structured grids," IEEE Transactions on Microwave Theory and Techniques, vol. 52, pp 554-559, 2004.

[23] N. K. Nikolova, H. W. Tam, and M. H. Bakr, "Sensitivity analysis with the FDTD method on structures grids," IEEE Transactions on Microwave Theory and Techniques, vol. 52, no2, pp. 1207-1216, 2004.

[24] S. M. Ali, N. K. Nikolova, and M. H. Bakr, “A central adjoint variable method for sensitivity analysis,” IEEE Transactions on Magnetics, vol. 40, no. 4, pp. 1969-1971, 2004.

[25] M. A. Swillam, M. H. Bakr, and X. Li, “Accurate sensitivity analysis of photon devices exploiting the finite-difference time-domain central adjoint variable method,” Journal of Applied Optics, vol. 46, no. 9, pp. 1492-1499, 2007.

[26] M. A. Swillam, M. H. Bakr, N. K. Nikolova, and X. Li, “Adjoint sensitivity analysis of dielectric discontinuities using FDTD,” Journal of Electromagnetics, vol. 27, no. 2, pp 123-140, 2007.

[27] M. H. Bakr and N. K. Nikolova, “An adjoint variable method for time-domain transmission-line modeling with fixed structured grids,” IEEE Transactions on Microwave Theory and Techniques, vol. 52, no. 2, pp. 449-554, 2004.

[28] P. Garcia, and J. P. Webb, “Optimization of planar devices by the finite element method,” IEEE Transactions on Microwave Theory and Techniques, vol. 38, pp. 48-53, 1990.

[29] S. Koziel and A. Bekasiewicz, “Fast EM-driven size reduction of antenna structures by means of adjoint sensitivities and trust regions,” IEEE Antennas and Wireless Propagation Letters, vol. 14, pp. 1681-1684, 2015.

[30] M. M. Botha and D. B. Davidson, "An explicit a posteriori error indicator for electromagnetic, finite element-boundary integral analysis," in IEEE Transactions on Antennas and Propagation, vol. 53, no. 11, pp. 3717-3725, Nov. 2005.

[31] K. C. Chellamuthu and N. Ida, "'A posteriori' element by element local error estimation technique and 2D & 3D adaptive finite element mesh refinement," IEEE Transactions on Magnetics, vol. 30, no. 5, pp. 3527-3530, Sept. 1994.

[32] S. M. Schnepp, “Error-driven dynamical hp-meshes with the Discontinuous Galerkin Method for three-dimensional wave propagation problems,” Journal of Computational and Applied Mathematics, vol. 270, pp. 353-368, 2014.

[33] F. C Meyer and D. B. Davidson, “A posteriori error estimates for two-dimensional electromagnetic field computations: boundary elements and finite elements,” ACES Journal, vol. 11, no. 2, pp. 40-54, 1996.

[34] K. C. Chellamuthu and N. Ida, "Reliability assessment of an 'a posteriori' error estimate for adaptive computation of electromagnetic field problems," in IEEE Transactions on Magnetics, vol. 31, no. 3, pp. 1761-1764, May 1995.

[35] P. Monk, “A posteriori error indicators for Maxwell’s equations,” Journal of Computational and Applied Mathematics, vol. 100, no. 2, pp. 173-190, Dec. 1998.

[36] P. Monk and E. Suli, “The adaptive computation of far-field patterns by a posteriori error estimation of linear functionals,” SIAM Journal of Numerical Analysis, vol. 36 no. 1, pp. 251-274, 1998.

[37] D. Estep and D. Neckels, “Fast and Reliable Methods for Determining the Evolution of Uncertain Parameters in Differential Equations,” J. Comp. Physics, vol. 213, pp. 530-556, April 2006.

[38] G. I. Marchuk, Adjoint Equations and Analysis of Complex Systems, Netherlands: Kluwer Academic Publishers; 1995.

[39] C. Key, A. Smull, B. M. Notaroš, D. Estep, and T. Butler, “Adjoint Methods for Uncertainty Quantification in Applied Computational Electromagnetics: FEM Scattering Examples,” Proceedings of the 2018 International Applied Computational Electromagnetics Society Symposium – ACES2018, March 25–29, 2018, Denver, Colorado, USA.

[40] C. Key, A. Smull, D. Estep, T. Butler, and B. M. Notaroš, “A Posteriori Element-wise Error Quantification for FEM Solvers Using Higher Order Basis Functions,” Proceedings of the 2018 IEEE International Symposium on Antennas and Propagation, July 8–13, 2018, Boston, MA, USA, pp. 1319–1320.

[41] B. M. Notaroš, Electromagnetics, New Jersey : PEARSON Prentice Hall; 2010.

[42] A. B. Manić, S. B. Manić, M. M. Ilić, and B. M. Notaroš, “Large anisotropic inhomogeneous higher order hierarchical generalized hexahedral finite elements for 3-D electromagnetic modeling of scattering and waveguide structures,” Microw. Opt. Technol. Lett., vol. 54, pp. 1644–1649, Jul. 2012.

[43] A. P. Smull, A.B. Manic, S.B. Manic, and B.M. Notaros, “Anisotropic Locally Conformal Perfectly Matched Layer for Higher Order Cuvilinear Finite Element Modeling,” IEEE Transactions on Antennas and Propagation, vol. 65, no. 12, pp. 7157-7165, December 2017.

[44] D. Estep, M. Larson, and R. Williams, “Estimating the error of numerical solutions of systems of nonlinear reaction-diffusion equations,” Memoirs of the American Mathematical Society, vol. 696, pp. 1-109, 2000.

[45] J. H. Chaudhry, D. Estep, S. Tavener, V. Carey, and J. Sandelin, “A Posteriori Error Analysis of Two-Stage Computation Methods with Application to Efficient Discretization and the Parareal Algorithm,” SIAM Journal on Numerical Analysis, vol. 54, pp. 2729-3122, 2016.

[46] J. J. Harmon, C. Key, D. Estep, T. Butler, B. M. Notaros, “Adjoint-based Accelerated Adaptive Refinement in Frequency Domain 3-D Finite Element Method Scattering Problems,” IEEE Transactions on Antennas and Propagation, *in print.*

[47] J. J. Harmon, C. Key, D. Estep, T. Butler, B. M. Notaros, “Sensitivity Analysis for Uncertain Material Parameters in Frequency Domain 3-D FEM,” IEEE Transactions on Antennas and Propagation, *in prep.*

[48] S. Kim, “Error estimation and adaptive refinement technique in the method of moments,” Doctoral Thesis, Georgia Institute of Technology, May 2017.

[49] J. Giesen and M. John, “Surface reconstruction based on a dynamical system,” Computer Graphics Forum, vol. 21, no. 3, pp. 363-371, 2002.

[50] R. Renka, “Two Simple Methods for Improving a Triangle Mesh Surface,” Computer Graphics Forum, vol. 35, no. 6, pp. 46-58, 2016.

[51] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, “Computational Geometry (2nd revised edition),” Springer-Verlag, 2000.

[52] S. M. Rao, D. R. Wilton, and A. W. Glisson, “Electromagnetic scattering by surfaces of arbitrary shape,” IEEE Transactions on Antennas Propagation, vol. 30, no. 5, pp. 409–418, May 1982.

[53] J. Ikaheimo, K. Forsman, and L. Kettunen, “Adaptive Mesh Generation in 2D Magnetostatic Integral Formulations,” IEEE Transactions on Magnetics, vol. 33, no. 2, pp. 1736-1739, Mar 1997.

[54] Z. J. Cendes, D. Shenton, and H. Shahnasser, “Magnetic Field Computation Using Delaunary Triangulation and Complementary Finite Element Method,” IEEE Transactions on Magnetics, vol. mag-19, no. 6, pp. 2551-2554, Nov 1983.

[55] M. M. Sakamoto, J. R. Cardoso, J. M, Machado, and M. Salles, “A 2-D Delaunay Refinement Algorithm Using an Initial Prerefinement From the Boundary Mesh,” IEEE Transactions on Magnetics, vol. 44, no. 6, pp. 1418-1421, Jun 2008.

[56] H. Tsuboi, T. Asahara, F. Kobayashi, and T. Misaki, “Adaptive Triangular Mesh Generation for Boundary Element Method in Three-Dimensional Electrostatic Problems,” IEEE Transactions on Magnetics, vol. 34, no. 5, pp. 3379-3382, Sept 1998.

[57] L. Janicke and A. Kost, “Error Estimation and Adaptive Mesh Generation in the 2D and 3D Finite Element Method,” IEEE Transactions on Magnetics, vol. 32, no. 3, pp. 1332-1337, May 1996.

[58] S. Dafour, G. Vinsard, B. Laporte, and R. Moretti, “Mesh Improvement in 2-D Eddy-Current Problems,” IEEE Transactions on Magnetics, vol. 38, no. 2, pp. 377-380, Mar 2002.

[59] K. Virga and Y. Rahmat-Samii, “RCS Characterization of a Finite Ground Plane with Perforated Apertures: Simulations and Measurements,” IEEE Transactions on Antennas and Propagation, vol. 42, no. 11, pp. 1491-1501, Nov 1994.

[60] D. A. Lindholm, “Automatic Triangular Mesh Generation on Surfaces of Polyhedra,” IEEE Transactions on Magnetics, vol. mag-19, no. 6, pp. 2539-2542, Nov 1983.

[61] M. Kostic, B. M. Kolundzija, D. S. Sumic, and B. L. Mrdakovic, (2010). "Optimized quadrilateral mesh for higher order method of moment based on triangular mesh decimation," Proceedings of the 2010 Antennas and Propagation Symposium, 2010.

[62] E. Catmull and J. Clark, "Recursively generated b-spline surfaces on arbitrary topological meshes," Computer-aided Design, vol. 10, no. 6, pp. 350–355, 1978.

[63] L. Velho and D. Zorin, "4-8 subdivision," Computer Aided Geometric Design, vol. 18, no. 5, pp. 397–427, 2001.

[64] J. Xia, I. Garcia, Y. He, S. Xin, and G. Patow, "Editable polycube map for gpu-based subdivision surfaces," Proceedings of the Symposium on Interactive 3D Graphics and Games, 2011, New York, NY, USA, pp. 151–158.

[65] I. Boier-Martin, H. Rushmeier, and J. Jin, "Parameterization of triangle meshes over quadrilateral domains," Proceedings of the Eurographics Symposium on Geometry Processing, 2004, Nice, France, pp. 197–208.

[66] B. Lévy, and Y. Liu, "LP Centroidal Voronoi Tesselation and its applications," ACM Transactions on Graphics, vol. 29, no. 4, pp. 101-A119, 2010.

[67] F. Kälberer, M. Nieser, and K. Polthier, "Quadcover - surface parameterization using branched coverings," Computer Graphics Forum, vol. 26, no. 3, pp. 375–384, 2007.

[68] N. Ray, W. C. Li, B. Lévy, A. Sheffer, and P. Alliez, "Periodic global parameterization," ACM Transactions on Graphics, vol. 25, pp. 1460–1485, Oct. 2006.

[69] D. Bommes, H. Zimmer, and L. Kobbelt, "Mixed-integer quadrangulation," ACM Transactions on Graphics, vol. 28, pp. 77-78, Jul. 2009.

[70] D. Bommes, B. Lévy, N. Pietroni, E. Puppo, C. Silva, M. Tarini, and D. Zorin, "Quad-Mesh Generation and Processing: A Survey," Computer Graphics Forum, vol. 32, no. 6, pp. 51-76, 2013.

[71] G. Xiao and Y. Hou, "Intuitive Formulation of Discontinuous Galerkin Surface Integral Equations for Electromagnetic Scattering Problems," IEEE Transactions on Antennas and Propagation, vol. 65, no. 1, pp. 287-294, Jan 2017.

[72] S. L. Ho, Y. Zhao, and W. N. Fu, "An Efficient Parameterized Mesh Method for Large Shape Variation in Optimal Designs of Electromagnetic Devices," IEEE Transactions on Magnetics, vol. 48, no. 11, pp. 4507-4510, Nov 2012.

[73] B. M. Kolundzija, "Automatic Mesh Generation Using Single- and Double-Node Segmentation Techniques," IEEE Antennas and Propagation Magazine, vol. 40, no. 4, pp. 30-38, Aug 1998.

[74] E. H. Newman and P. Tulyathan, "A Surface Patch Model for Polygonal Plates," IEEE Transactions on Antennas and Propagation, vol. ap-30, no. 4, pp. 588-593, Jul 1982.

[75] T. A. Linkowski and P.M. Slobodzian, "Contour- and Grid-Based Algorithm for Mixed Triangular-Rectangular Planar Mesh Generation," Progress in Electromagnetics Research B, vol. 40, pp. 201-220, 2012.

[76] J. Moreno, M. J. Algar, I Gonzalez Diego, and F. Catedra, "A New Mesh Generator Optimized for Electromagnetic Analysis," Proceedings of the 5th European Conference on Antennas and Propagation, pp. 1734-1738, 2011.

[77] J. Moreno, M. J. Algar, I. Gonzalez, F. Catedra, "Redesign and Optimization of the Paving Algorithm Applied to Electromagnetic Tools," Progress in Electromagnetics Research B, vol. 29, pp. 409-429, 2011.

[78] H. Borouchaki and P. Frey, "Adaptive triangular-quadrilateral mesh generation," International Journal for Numerical Methods In Engineering, vol. 41 no. 5, pp. 915-934, 1998.

[79] Z.Q. Xie, R. Sevilla, O. Hassan, and K. Morgan, "The generation of arbitrary order curved meshes for 3d finite element analysis," Computational Mechanics, vol. 51, no. 3, pp. 361-374, 2013.

[80] J. C. Young, "Higher-Order Mesh Generation Using Linear Meshes [~rfm~EM Programmer's Notebook]," IEEE Antennas and Propagation Magazine, vol. 61, no. 2, pp. 120-126, Apr 2019.

[81] J. Harmon, C. Key, and B. M. Notaros, "Geometrically Conformal Quadrilateral Surface-Reconstruction for MoM-SIE Simulations," Proceedings of the 2019 International Applied Computational Electromagnetics Society (ACES) Symposium – ACES2019, April 15–19, 2019, Miami, Florida, USA.

[82] C. Key and B. M. Notaros, "Automatic Generalized Quadrilateral Surface Meshing in Computational Electromagnetics by Discrete Surface Ricci Flow," accepted for the 2019 IEEE International Symposium on Antennas and Propagation, July 7–12, 2019, Atlanta, GA.

[83] J. Harmon, C. Key, S. B. Manic, and B. M. Notaros, "Construction and Application of Geometrically Optimal Curvilinear Surface Elements for Double Higher-Order MoM-SIE Modeling," accepted for the 2019 USNC-URSI Radio Science Meeting (joint with the IEEE AP-S International Symposium), July 7–12, 2019, Atlanta, GA.

[84] G. Zhang, L. Zeng, and G. Yau, "The unified discrete surface Ricci flow," Graphical Models, vol. 76 no. 5, pp. 321-339, 2014.

[85] H. D. Cao and X. P. Zhu, "A Complete Proof of the Poincaré and Geometrization Conjectures - Application of the Hamilton-Perelman Theory of the Ricci Flow," Asian Journal of Mathematics, vol. 10, no. 2, pp. 165-492, June 2006.

[86] M. Jin, J. Kim, F. Luo, and X. Gu, “Discrete Surface Ricci Flow,“ IEEE Transactions on Visualization and Computer Graphics, vol. 14, no. 5, Oct 2008.

[87] X. Gu, F. Luo, J. Sun, and T. Wu, “A Discrete Uniformization Theorem for Polyhedral Surfaces,” Journal of Differential Geometry, vol. 109, no. 2, 2013.

[88] X. Gu, R. Guo, F. Luo, J. Sun, and T. Wu, “A Discrete Uniformization Theorem for Polyhedral Surfaces II,” Journal of Differential Geometry, vol. 109, no. 3, 2014.

[89] R. Garimella, “Conformal Refinement of Unstructured Quadrilateral Meshes,” Proceedings of the 18th International Meshing Roundtable, pp. 30-44, 2009.

[90] F. Hossain, T. Geok, T. Rahman, M. Hindia, K. Dimyati, S. Ahmed, C. Tso, and N. Abd Rahman, “An efficient 3-D ray tracing method: prediction of indoor radio propagation at 28 GHz in 5G network,” *MDPI Electronics,* vol. 8, no. 1, pp. 286-306, Mar. 2019.

[91] B. Troksa, C. Key, F. Kunkel, S. Savic, M. Ilic, and B. Notaros, “Ray tracing using shooting-bouncing technique to model mine tunnels: theory and verification for a PEC waveguide,” *ACES Journal,* vol. 34, no.2, Feb. 2019.

[92] N. Sood, L. Liang, S. V. Hum, and C. D. Sarris, "Ray-tracing based modeling of ultrawideband pulse propagation in railway tunnels", *Proc. IEEE APS/URSI Int. Symp.,* pp. 2383-2386, Jul. 2011.

[93] L. Kanaris, A. Kokkinis, M. Raspopoulos, A. Liotta, and S. Stavrou, “Improving RSS fingerprint-based localization using directional antennas,” *Antennas and Propagation (EuCAP) 20148th European Conference,* pp. 2174-2177, Aug. 2014.

[94] M. Catedra and J. Perez, Cell Planning for Wireless Communications, Norwood, MA, USA: Artech House, 1999.

[95] D. McNamara, C. Pistorius, and J. Malherbe, Introduction to the Uniform Geometrical Theory of Diffraction, Norwood, MA, USA: Artech House, 1990.

[96] T. Ize, I. Wald, and S. Parker, “Ray tracing with the BSP tree,” *Proceedings of IEEE Symposium on Interactive Ray Tracing,* pp. 159-166, 2008.

[97] S. Seidel, T. Rappaport, “Site-specific propagation prediction for wireless in-building personal communication system design,” *IEEE Transactions on Vehicular Technology,* vol. 43, pp. 879-891, Nov. 1994.

[98] S. Chen and S. Jeng, “An SBR/image approach for radio wave propagation in indoor environments with metallic furniture,” *“IEEE Transactions on Antennas and Propagation,* vol. 45, pp. 98-106, Jan. 1997.

[99] C. Yang, B. Wu, and C. Ko, “A ray-tracing method for modeling indoor wave propagation and penetration,” *IEEE Transactions on Antennas and Propagation,* vol. 46, pp. 907-919, Jun. 1998.

[100] Z. Yun and M. Iskander, “Ray tracing for radio propagation modeling: Principles and applications,” *IEEE Access,* vol. 3, pp. 1089-1100, Sep. 2015.

[101] Z. Yun, M Iskander, and Z. Zhang, “Development of a new shooting-and-bouncing ray (SBR) tracing method that avoid ray double counting,” *IEEE APS International Symposium Digest,* vol. 1, pp. 464-467, Jul. 2001.

[102] D. Didascalou, T. M. Schäfer, F. Weinmann, and W. Wiesbeck, "Ray density normalization for ray-optical wave propagation modeling in arbitrarily shaped tunnels", *IEEE Trans. Antennas Propagat*., vol. 48, pp. 1316-1325, Sept. 2000.

[103] N. Noori, A. A. Shishegar, E. Jedari, "A New Double Counting Cancellation Technique for Three-Dimensional Ray Launching Method", *IEEE Antennas and Propagation Society International Symposium*, pp. 2185-2188, 2006.

[104] R. Mishra, “An overview of neural network methods in computational electromagnetics,” International Journal of RF and Microwave Computer-Aided Engineering, vol. 12, no. 1, Dec 2001.

[105] T. Nguyen-Thien and T. Tran-Cong, “Approximation of functions and their derivatives: A neural network implementation with applications,” Applied Mathematical Modelling, vol. 23, no. 9, pp. 687-704, 1999.

[106] L. Lanzi, C. Bisagni, and S. Ricci, “Neural network systems to reproduce crash behavior of structural components,” Computers & Structures, vol. 82, no. 1, pp. 93–108, 2004.

[107] J. Ghaboussi, J. H. Garrett, and X. Wu, “Knowledge-based modeling of material behavior with neural networks,” Journal of Engineering Mechanics, vol. 117, no. 1, pp. 132–153, 1991.

[108] J. D. Martín-Guerrero, M. J. Rupérez-Moreno, F. Martinez-Martínez, D. Lorente-Garrido, A. J. Serrano-López, C. Monserrat, S. Martínez-Sanchis, and M. Martínez-Sober, “Machine Learning for Modeling the Biomechanical Behavior of Human Soft Tissue” IEEE 16th International Conference on Data Mining Workshops (ICDMW), Barcelona, 2016, pp. 247-253, 2016.

[109] I. Lagaris, A. Likas, D. Fotiadis, “Artificial neural networks for solving ordinary and partial differential equations,” IEEE Transactions on Neural Networks, vol. 9, no. 5, pp. 987-1000, Sep 1998.

[110] E. Soliman, M. Bakr, N. Nikolova, “Neural networks-method of moments (NN-MoM) for the efficient filling of the coupling matrix,” IEEE Transactions on Antennas and Propagation, vol. 52, no. 6, Jun 2004.

[111] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabisal, ‘‘Mastering the game of go without human knowledge,’’ Nature, vol. 550, no. 7676, pp. 354–359, 2017.

[112] R. Evans, J. Jumper, J. Kirkpatrick, L. Sifre, T. Green, C. Qin, A. Žídek, S. Nelson, A. Bridgland, H. Penedones, S. Petersen, K. Simonyan, S. Crossan, D. Jones, D. Silver, K. Kavukcuoglu, D. Hassabis, and A. Senior, "De novo structure prediction with deep-learning based scoring," Thirteenth Critical Assessment of Techniques for Protein Structure Prediction, pp. 1-4, Dec 2018.

[113] A. R. Shahani, S. Setayeshi, S. A. Nodamaie, M. A. Asadi, and S. Rezaie, "Prediction of influence parameters on the hot rolling process using finite element method and neural network," Journal of Materials Processing Technology, vol. 209, no. 4, pp. 1920–1935, 2009.

[114] X. Wu and J. Ghaboussi, "Neural network-based material modeling," Thesis, University of Illinois, 1991.

[115] W. Ruijter, R. Spallino, L. Warnet, and A. de Boer, "Optimization of composite panels using neural networks and genetic algorithms," Second MIT Conference on Compurational Fluid and Solid Mechanics, K. J. Bathe, Ed. Elsevier Science Ltd, pp. 2359–2363, 2003.

[116] A. A. Javadi, T. P. Tan, and M. Zhang. "Neural network for constitutive modelling in finite element analysis", Computer Assisted Mechanics and Engineering Sciences, Vol. 10, pp. 523-5299, 2003.

[117] R. I. Levin and N. A. J. Lieven, "Dynamic finite element model updating using neural networks," Journal of Sound and Vibration, vol. 210, no. 5, pp. 593–607, 1998.

[118] Y. Hashash, S. Jung, and J. Ghaboussi, "Numerical implementation of a neural network based material model in finite element analysis," International Journal for numerical methods in engineering, vol. 59, no. 7, pp. 989–1005, 2004.

[119] J. Schmidhuber, "Deep Learning in Neural Networks: An Overview," Neural Networks, vol. 61, pp. 85-117, Jan 2015.

[120] K. He, X. Zhang, S. Ren, J. Sun, "Deep Residual Learning for Image Recognition," IEEE Computer Vision and Pattern Recognition Proceedings, pp. 770-778, 2016.

[121] K. He and J. Sun, "Convolutional neural networks at constrained time cost," Computer Vision and Pattern Recognition Proceedings, pp. 5353-5360, 2015.

[122] S. M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, ''Universal adversarial perturbations,'' *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 86–94, Jul 2017.

[123] S. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, ''DeepFool: A simple and accurate method to fool deep neural networks,'' *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 2574–2582, 2016.

[124] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, ''The limitations of deep learning in adversarial settings,'' *Proc. IEEE Eur. Symp. Secur. Privacy*, pp. 372–387, 2016.

[125] A. Rozsa, M. Günther, and T. E. Boult, ‘‘Are accuracy and robustness correlated,’’ *Proc. IEEE Int. Conf. Mach. Learn. Appl.,* pp. 227–232, 2016.

[126] J. H. Metzen, M. C. Kumar, T. Brox, and V. Fischer, ‘‘Universal adversarial perturbations against semantic image segmentation,’’ *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 2755–2764, Oct 2017.

[127] X. Li and F. Li, ‘‘Adversarial examples detection in deep networks with convolutional filter statistics,’’ *Proc. Int. Conf. Comput. Vis.*, pp. 1–9, 2017.

[128] A. Nguyen, J. Yosinski, and J. Clune, ‘‘Deep neural networks are easily fooled: High confidence predictions for unrecognizable images,’’ *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 427–436 2015.

[129] C. Key and B. M. Notaros, “Data-Enabled Advancement of Computation in Engineering: A Robust Machine Learning Approach to Accelerating Variational Methods in Electromagnetics and Other Disciplines,” *IEEE Antennas and Wireless Propagation Letters,* Vol. 19, No. 4, April 2020, pp. 626-630.

[130] M. M. Kostic and B. M. Kolundzija, “Maximally Orthogonalized Higher Order Bases Over Generalized Wired, Quadrilaterals, and Hexahedra,” *IEEE Transactions on Antennas and Propagation,* Vol. 61, No. 6, June 2013, pp. 3135-3148.

[131] Z. Sun, E. Rooke, J. Charton, Y. He, J. Lu, and S. Baek, “ZerNet: Convolutional Neural Networks on Arbitrary Surfaces Via Zernike Local Tangent Space Estimation,” *Computer Graphics Forum,* Vol. 00, No. 0, 2020, pp. 1-13.

[132] D. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” *International Conference on Learning Representations,* 2014.

PUBLICATIONS

Journal Papers

C. Key and B. Notaroš, "Predicting Macro Basis Functions for Method of Moments Scattering Problems Using Deep Neural Networks," IEEE Transactions on Antennas and Propagation, *in review*.

C. Key and B. Notaroš, "Data-Enabled Advancement of Computation in Engineering: A Robust Machine Learning Approach to Accelerating Variational Methods in Electromagnetics and Other Disciplines," IEEE Antennas and Wireless Propagation Letters, vol. 19, no. 4, pp. 626-630, April 2020.

C. Key, B. Troksa, S. Kasdorf, and B. Notaros, "Non-Self-Adjacent Ray Classes for Parallelizable Shooting Bouncing Ray Tracing Double Count Removal," IEEE Journal on Multiscale and Multiphysics Computational Techniques, *in review*.

C. Key, A. Smull, D. Estep, T. Butler, and B. Notaroš, "A Posteriori Error Estimation and Adaptive Discretization Refinement Using Adjoint Methods in CEM: A Study with a One-Dimensional Higher-Order FEM Scattering Example," IEEE Transactions on Antennas and Propagations, vol. 68, no. 5, pp. 3791-3806, May 2020.

C. Key, A. Smull, B. M. Notaroš, D. Estep, and T. Butler, "Adjoint Methods for Uncertainty Quantification in Applied Computational Electromagnetics: FEM Scattering Examples," Applied Computational Electromagnetics Society Journal, Vol. 34, No. 2, 2019.

C. Key, J. Harmon, and B. Notaroš, "Discrete Surface Ricci Flow for General Surface Meshing in Computational Electromagnetics Using Iterative Adaptive Refinement," IEEE Transactions on Antennas and Propagations, *in print*.

B. A. Troksa, C. L. Key, F. B. Kunkel, S. V. Savić, M. M. Ilic, and B. M. Notaroš, "Ray Tracing Using Shooting-Bouncing Technique to Model Mine Tunnels: Theory and Verification for a PEC Waveguide," Applied Computational Electromagnetics Society Journal, Vol. 34, No. 2, 2019.

S. Kasdorf, B. Troksa, C. Key, J. Harmon, and B. Notaroš, "Advancing Accuracy of Shooting and Bouncing Rays Method for Ray-Tracing Propagation Modeling on Novel Approaches to Ray Cone Angle Calculation," IEEE Transactions on Antennas and Propagation, *in review*.

J. Harmon, C. Key, D. Estep, T. Butler, and B. Notaros, "Adjoint-based Accelerated Adaptive Refinement in Frequency Domain 3-D Finite Element Method Scattering Problems," IEEE Transactions on Antennas and Propagation, *in review*.

J. J. Harmon, C. Key, D. Estep, T. Butler, B. M. Notaros, "Sensitivity Analysis for Uncertain Material Parameters in Frequency Domain 3-D FEM," IEEE Transactions on Antennas and Propagation, *in review*.

Peer-Reviewed Conference Papers and Abstracts

C. Key and B. M. Notaros, "Automatic Generalized Quadrilateral Surface Meshing in Computational Electromagnetics by Discrete Surface Ricci Flow," accepted for the 2019 IEEE International Symposium on Antennas and Propagation, July 7–12, 2019, Atlanta, GA.

C. Key, J. Harmon, B. Troksa, and B. M. Notaroš, "Fast Sphere Intersection Tests for Shooting-Bouncing Ray Tracing: Space Partitioning and Ray Path Voxelization," Proc. 2019 USNC-URSI National Radio Science Meeting, January 9-12, 2019, Boulder, CO.

C. Key, J. Harmon, B. Troksa, and B. M. Notaroš, "Adjoint-based A Posteriori Error Estimation and its Applications in CEM: DHO Techniques and the 3D Scattering Problem," Proc. 2019 USNC-URSI National Radio Science Meeting, January 9-12, 2019, Boulder, CO.

C. Key, J. Harmon, B. Troksa, and B. M. Notaroš, "Applications of Shooting-Bouncing Ray Tracing to Modeling Propagation in Underground Mines," Proc. 2019 USNC-URSI National Radio Science Meeting, January 9-12, 2019, Boulder, CO.

C. Key, A. Smull, B. M. Notaroš, D. Estep, and T. Butler, "Adjoint Methods for Uncertainty Quantification in Applied Computational Electromagnetics: FEM Scattering Examples," Proceedings of the 2018 International Applied Computational Electromagnetics Society (ACES) Symposium – ACES2018, March 25–29, 2018, Denver, CO.

C. Key, A. Smull, D. Estep, T. Butler, and B. M. Notaroš, "A Posteriori Element-wise Error Quantification for FEM Solvers Using Higher Order Basis Functions," Proceedings of the 2018 IEEE International Symposium on Antennas and Propagation, July 8–13, 2018, Boston, MA, pp. 1319–1320.

C. Key, B. Troksa, S. Savić, M. M. Ilic, and B. M. Notaroš, "EM Simulation and Characterization of Underground Mines Using Ray Tracing, Vector Parabolic Equation, and Hybrid Approaches," Proc. 2018 USNC-URSI National Radio Science Meeting, January 4-7, 2018, Boulder, CO.

J. Harmon, C. Key, B. M. Notaroš, D. Estep, and T. Butler, "Adjoint-Based Uncertainty Quantification in Frequency-Domain Double Higher-Order FEM," Proceedings of the 2019 International Applied Computational Electromagnetics Society (ACES) Symposium – ACES2019, April 14–18, 2019, Miami, FL.

J. Harmon, C. Key, B. M. Notaroš, "Geometrically Conformal Quadrilateral Surface-Reconstruction for MoM-SIE Simulations," Proceedings of the 2019 International Applied Computational Electromagnetics Society (ACES) Symposium – ACES2019, April 14–18, 2019, Miami, FL.