DISSERTATION

FULL-WAVE AND ASYMPTOTIC COMPUTATIONAL ELECTROMAGNETICS METHODS: ON THEIR USE AND IMPLEMENTATION IN RECEIVED SIGNAL STRENGH, RADAR-CROSS-SECTION, AND UNCERTAINTY QUANTIFICATION PREDICTIONS

Submitted by

Stephen Kasdorf

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 2024

Doctoral Committee:

Advisor: Branislav M. Notaroš

Milan Ilić
Jesse Wilson
Karan Venayagamoorthy

Copyright by Stephen Kasdorf 2024

All Rights Reserved

ABSTRACT

FULL-WAVE AND ASYMPTOTIC COMPUTATIONAL ELECTROMAGNETICS METHODS: ON THEIR USE AND IMPLEMENTATION IN RECEIVED SIGNAL STRENGH, RADAR-CROSS-SECTION, AND UNCERTAINTY QUANTIFICATION PREDICTIONS

We propose and evaluate several improvements to the accuracy of the shooting and bouncing rays (SBR) method for ray-tracing (RT) electromagnetic modeling. A per-ray cone angle calculation is introduced, with the maximum separation angle determined for each individual ray based on local neighbors, allowing the smallest theoretical error in SBR. This enables adaptive ray spawning and provides a unique analysis of the effect of ray cone sizes on accuracy. For conventional uniform angular distribution, we derive an optimal cone angle to further enhance accuracy. Both approaches are integrated with icosahedral ray spawning geometry and a double-counted ray removal technique, which avoids complex ray path searches. The results demonstrate that the advanced SBR method can perform wireless propagation modeling of tunnel environments with accuracy comparable to the image theory RT method, but with much greater efficiency.

To further advance the efficiency of the SBR method, we propose a unified parallelization framework leveraging NVIDIA OptiX Prime programming interfaces on graphics processing units (GPUs). The framework achieves comprehensive parallelization of all components of the SBR algorithm, including traditionally sequential tasks like electric field computation and postprocessing. Through optimization of memory usage and GPU resources, the new SBR method achieves upwards of 99% parallelism under Amdahl's scaling law. This innovative

parallelization yields dramatic speedups without sacrificing the previously enhanced accuracy of the SBR method, demonstrating an unparalleled level of computational efficiency for large-scale electromagnetic propagation simulations.

Finally, we implement and validate several advanced Kriging methodologies for uncertainty quantification (UQ) in computational electromagnetics (CEM). The universal Kriging, Taylor Kriging, and gradient-enhanced Kriging methods are applied to reconstruct probability density functions, offering efficient alternatives to Monte Carlo simulations. We further propose the novel gradient-enhanced Taylor Kriging (GETK) method, which combines the advantages of gradient information and basis functions, yielding superior surrogate function accuracy and faster convergence. Numerical results using higher-order finite-element scattering modeling show that GETK dramatically outperforms other Kriging and non-Kriging methods in UQ problems, accurately predicting the impact of stochastic input parameters, such as material uncertainties, on quantities of interest like radar cross-section.

ACKNOWLEDGEMENTS

First and foremost, I would like to acknowledge my advisor, Prof. Branislav M. Notaroš for his support and guidance during my Ph.D tenure. He was a great help during my studies and helped to shape me as a researcher. I would also like to acknowledge the rest of my committee members, Prof. Jesse Wilson, Prof. Milan Ilić, and Prof. Karan Venayagamoorthy for their advice and help.

I would also like to thank my lab mates, Jake Harmon and Blake Troksa, for their help on the various projects that we worked together on.

Finally, I would like to thank my parents, Roy and Marti Kasdorf, for their support during my time at Colorado State University.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

Introduction

In the rapidly evolving field of computational electromagnetics (CEM), the need for accurate, efficient, and scalable methods to model complex electromagnetic systems has become increasingly pressing. This dissertation focuses on three critical areas of CEM research: the improvement of ray tracing (RT) accuracy, the parallelization of RT methods for enhanced computational performance, and the application of uncertainty quantification (UQ) to electromagnetic simulations. By addressing these areas, this work aims to push the boundaries of CEM modeling, providing novel contributions that significantly enhance the precision and efficiency of simulations in complex, large-scale environments.

The first core theme of this dissertation is improving the accuracy of ray tracing (RT) methods, which are widely used in CEM for modeling electrically large environments. Ray tracing methods discussed here differ from traditional graphics-based RT techniques where graphics-based rays are simply weighted and propagated through the environment, and CEM RT techniques must also associate an electric field with each ray which is coherently summed for all rays at reception points. RT techniques, such as the shooting and bouncing rays (SBR) method, are computationally efficient alternatives to full-wave methods, especially for large and complex environments such as tunnels and urban landscapes [1], [2]. However, despite their efficiency, RT methods, particularly SBR, often suffer from accuracy issues due to phase and magnitude errors introduced by inexact ray paths and improper ray reception handling [3], [4]. Previous studies have shown that while the image theory (IT) method offers greater accuracy by tracing exact ray paths, its computational complexity limits its applicability to electrically large environments [5].

This dissertation introduces novel advancements to the SBR methodology to improve its accuracy, making it comparable to the IT method without significantly increasing computational cost. Specifically, this research proposes optimized ray cone angle calculations and a double-counted ray elimination technique, both of which drastically reduce phase and magnitude errors in SBR simulations. The results demonstrate that the enhanced SBR method can accurately model wireless signal propagation in large-scale environments, such as tunnels, where multipath propagation effects are challenging to simulate [6], [7], [8]. These contributions ensure that SBR remains a practical solution for large-scale CEM problems while achieving accuracy levels traditionally thought unattainable for asymptotic methods.

The second focus of this dissertation is on the parallelization of ray tracing methods to further enhance computational efficiency. The growing complexity of CEM problems, especially in electrically large environments (or environments that are many times larger than the wavelength of interest), has necessitated the use of advanced computational resources, such as graphics processing units (GPUs), to accelerate simulations. GPUs, with their architecture optimized for massive parallel processing, provide an ideal platform for accelerating RT methods. This work presents a unified parallelization framework for the SBR method, integrating it with NVIDIA's OptiX Prime to fully exploit GPU capabilities [9], [10].

A key challenge in parallelizing the SBR method lies in the sequential nature of certain tasks, such as electric field computation, which traditionally limit parallelism [11]. By addressing these limitations, this research achieves upwards of 99% parallelism for SBR computations, significantly reducing simulation times without sacrificing accuracy [12]. The parallelized SBR framework demonstrates substantial computational speedups in complex environments, such as tunnels and urban areas, enabling real-time or near-real-time simulations that would otherwise be infeasible

with traditional methods. This work contributes not only to the field of CEM but also to broader applications requiring high-performance computing, such as wireless network planning and urban modeling.

Finally, this dissertation addresses uncertainty quantification (UQ) in CEM simulations, which is essential for understanding how input uncertainties affect the reliability of electromagnetic models. UQ plays a critical role in real-world applications, where uncertainties in material properties, geometries, and environmental factors are inevitable. Traditional Monte Carlo methods, though widely used for UQ, are often computationally prohibitive due to their high sampling requirements [13], [14]. To overcome these challenges, this work explores advanced surrogate modeling techniques, including gradient-enhanced Kriging (GEK) and the novel gradient-enhanced Taylor Kriging (GETK) [15].

These surrogate models enable efficient UQ analysis by drastically reducing the number of samples needed to accurately reconstruct probability density functions. The development of GEK and GETK methods offers a computationally feasible approach to UQ in CEM, ensuring that high-precision UQ can be applied even in large-scale simulations without excessive computational costs. These advancements provide a robust framework for integrating UQ into goal-oriented simulation objectives, further enhancing the reliability and effectiveness of electromagnetic models in diverse applications, from satellite communications to wireless network design.

By addressing the key areas of ray tracing accuracy, parallelization, and uncertainty quantification, this dissertation presents a comprehensive approach to advancing CEM methodologies. The contributions made here improve both the accuracy and computational efficiency of electromagnetic simulations, offering practical solutions to the challenges faced by researchers and engineers in fields such as wireless communications, aerospace, and urban

infrastructure planning. Through the integration of advanced surrogate models, optimized RT algorithms, and high-performance parallel computing techniques, this work provides a robust foundation for tackling the increasing complexity of modern electromagnetic systems.

Chapter 2

Advancing Accuracy of Shooting and Bouncing Rays Method for Ray-Tracing Propagation Modeling Based on Novel Approaches to Ray Cone Angle Calculation[1]

2.1 Introduction

As communications frequencies become higher and the propagation environments that need to be electromagnetically characterized become physically larger and more interconnected given the increased complexities and demands of emerging wireless system functionalities, these structures become more difficult to model using full-wave computational electromagnetics (CEM) techniques. For example, CEM modeling and simulation of wireless signal propagation in tunnels of underground mines or traffic (e.g., railway or subway) systems presents extraordinary challenges and is an open CEM research problem with unparalleled difficulty in many of its aspects. A mine or traffic tunnel at wireless communication frequencies is electrically an extremely large electromagnetic system, typically spanning thousands of wavelengths. High-frequency asymptotic techniques have been shown to be computationally efficient at modeling such structures [1],[5],[6],[7].

Indeed, there has been a "renaissance" of CEM simulations and computations using high-frequency, asymptotic methodologies and implementations and renewed and growing interest in

[1] This chapter is based on work from Ref. [2]

their applications to wireless propagation modeling within indoor and outdoor communication environments. While asymptotic methods for numerically approximate high-frequency modeling can never be as accurate as the full-wave methods for numerically rigorous field computation, it appears that there are opportunities for the improvement of accuracy of asymptotic modeling in some of its components and aspects. This paper addresses the accuracy of ray-based high-frequency modeling in general and when particularly applied to wireless propagation modeling of tunnel environments.

One of the most important classes of asymptotic techniques for CEM modeling is constituted by the ray tracing (RT) approach, based on the theory of geometrical optics (GO), by which the electric field is discretized into rays, with two common methods used for RT simulations, (i) the image theory method (IT) and (ii) the shooting and bouncing rays (SBR) method. The IT approach uses the method of image transformation to mirror a transmitter across reflection planes in the environment to find exact propagation paths from the transmitter to a receiver. From the accuracy standpoint, this method is advantageous due to the exact paths computed, making the phase error negligible. The only sources of phase error stem from geometric mesh approximations and numerical truncation, the latter typically being insubstantial compared to the former. However, the computational complexity of the IT method is $O(N^K)$ where N is the number of planar facets in the model and K is the number of reflections [16]. In electrically extremely large, closed environments, many rays and reflections are needed for adequate sampling, making IT impractical despite its attractive accuracy characteristics.

The SBR method, in contrast, launches rays from a transmitter in every direction, each ray is tracked as it propagates through the geometric environment according to Snell's law, and its magnitude is adjusted according to Fresnel's coefficients [1]. These paths are then tested to

determine if the ray intersects a receiver, in which case its electric field is added to the field at the receiver point. The ray paths are inexact, so further phase error is introduced in the SBR method compared to the IT method. However, the computational complexity of SBR, $O(NK)$, where N is the number of rays spawned and K is the number of reflections, is substantially lower than that of IT. Additionally, the SBR method allows for the extension of propagation to include refraction and diffraction, which IT alone cannot accomplish. This makes the SBR method much more efficient and versatile for use in large environments.

Previous work has demonstrated the effectiveness of the SBR method in tunnel environments, with findings showing that this method is a useful simulation technique for large, complex environments. The accuracy of the approach, however, is poor when compared to IT and full-wave techniques [7]. In [16], the SBR model was shown to be ineffective at simulating fields in tunnel environments further than a few hundred meters. This is due to issues with ray receptions in the SBR model. The so-called SBR cones method uses reception spheres to model the reception of a wavefront at a receiving antenna, where the sizes of these spheres are extremely important for the accuracy of the SBR method [7]. With large spheres, the results become inaccurate due to multiple counting of propagating wavefronts. Too small of a sphere size, on the other hand, causes discontinuities in the wavefront, known as undercounting. Previous work has been done in determining the appropriate sizes of the reception spheres by using spheres of variable size determined by path length [3],[4],[5],[7],[17],[18],[19]. However, these works assume that the ray distribution is uniform across the three-dimensional (3-D) spawning volume, and this assumption is invalid in 3-D, which can lead to inaccuracies.

This paper presents the development of an SBR model and analysis and improvement of the accuracy of the SBR computation based on three factors and associated novel contributions.

Specifically, we theoretically and empirically analyze importance of the distribution angle of rays launched from the transmitter for phase accuracy, coupled with the importance of ray double count removal for magnitude accuracy. Firstly, we propose per-ray cone angle calculation, with the maximum separation angle calculated for every individual ray based on a set of local neighbors rather than a single global maximum. This allows the smallest theoretical error of the SBR method, with undercounting being completely removed from the model, as well as decreasing the ray cone size which substantially reduces the phase error. Secondly, we also analyze constant angular distributions of the rays, which allow for much more expeditious simulations but make the overall accuracy of the computation extremely sensitive to the choice of the angular distribution. Here, we derive an analytical expression for the optimal choice of this angle to maximize the overall accuracy of the SBR computation, and demonstrate empirically that the SBR simulations generated using the derived optimal constant angle are indistinguishable from those generated using per-ray angles, that is, from the theoretical best-case scenario for simulation accuracy. Thirdly, the two approaches to ray cone angle calculation are coupled with a double counted rays identification and removal approach using icosahedron geometry and neighbor ray sets to identify and remove double counted contribution without the need for lengthy and complicated path searches based on the full geometry of the ray paths, hence coupling improvements of phase and magnitude accuracies. We thus demonstrate how the SBR methodology and implementation can be advanced to produce an SBR approach of similar accuracy as the image theory RT method, in simulations of large tunnels. The results show that, with the proposed, implemented, and analyzed accuracy improvements, the SBR method can perform wireless propagation modeling of planar tunnel environments with the same accuracy as the dramatically less efficient image theory approach. The three contributions to the SBR methodology are completely general, and thus are capable of being applied to any existing

SBR method, model, and simulation. These advancements are positioned and capable of increasing accuracy of an existing SBR technique to a point where the results are almost indistinguishable from the IT approach, as demonstrated in this paper. In fact, this demonstration is the fourth novel contribution of this work, as it has always been thought that the accuracy of the SBR approach can never be brought to the level of the IT accuracy. Note that the basic theory and preliminary accuracy validation of our SBR RT analysis of tunnels are presented in a summary form in [20].

2.2 SBR Methodology

2.2.1 Ray Receptions

Once rays have been tracked as they propagate through a medium, they are tested to see if there is an intersection of the ray and a receiving antenna. The cones technique is advantageous here, because the regular, circular cross-section of the ray bundle. Reception points for the rays are modeled as growing spheres which represent the growing cross-section of the ray cone. An intersection test between the sphere and the center ray of the cone is used, as shown graphically in Fig. 2.1, which is computationally simple and efficient. If this center ray intersects a receiver sphere, one of the rays represented in the ray cone is an exact hit. The linear nature of Maxwell's equations results in the field at any receiver point being the linear superposition of all electric field contributions of rays that intersect this point.

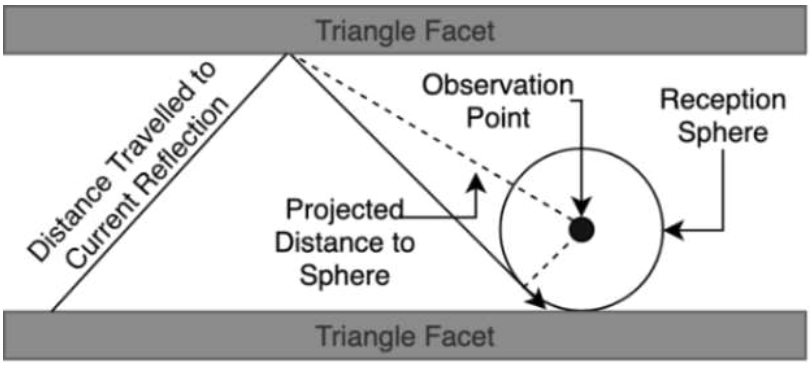


Fig 2.1: Ray–sphere intersection: graphical representation for the scene of a ray approaching a sphere and the corresponding components used in the calculation of the reception sphere radius.

The radius of the ray cones, and thus the reception spheres, is determined for the ith ray by

$$r_{sphere_i} = \frac{\alpha_i d_i}{\sqrt{3}} \tag{2.1}$$

where d_i is the total distance a ray has travelled and α_i is the maximum separation angle between a ray and its neighbors [3],[4],[5],[7],[16],[17],[18],[19]. This equation calculates the minimum possible ray cone radius that fully covers the electric field radiation. In tunnel environments, it is possible to have multiple reflected wavefronts received from the source. However, the propagation distance is tracked individually with every ray, so this can be applied independently for each ray.

2.2.2 Double Count Removal Method

The SBR approach requires launching discrete rays, each of which representing a cone of influence to fully cover the transmitting radiation pattern. Any places where the cones do not overlap would represent a non-physical scenario, where receptions might be missed in the simulation, constituting undercounting. To fully cover the 3-D radiation pattern without discontinuities in the field, the cones must overlap, as shown in Fig. 2.2. This overlap represents a substantial drawback as the overlapping wavefronts can be represented at least twice in the

simulation. Any intersection of the overlapping wavefronts with a receiver will lead to the electric field from a single wavefront contributing twice to the total field, which is referred to as double counting [3],[4],[18]. The double counted fields can be removed but require searching ray paths to find common images. This shows the important role both the ray distribution and cone sizes play in the accurate implementation of the SBR method.

The spawning techniques for the ray tracing application involve sampling points on surfaces that surround some ray origin point. Once these sampled points are determined, each ray for the ray tracing algorithm is launched from the origin to one of the sampled points. While different spawning techniques such as uniform grid sampling and Fibonacci spiral sampling have been tested and provide various advantages regarding sample spacing and ease of formation [1],[6],[21], we use here an icosahedron for the sampling method. Each face of an icosahedron is subdivided into equilateral triangles, where the number of vertices is given according to the *n*th triangular number. The rays are launched on the vertices of the subdivided icosahedron face with the number of rays launched on a face being $n(n+1)/2$, where *n* is the number of subdivisions along an edge of the face. This allows for very uniform distribution of launched rays, which is a requirement for accurate SBR simulations. Additionally, the icosahedron provides important topological information about the rays; the rays can be grouped together into sets of neighbor rays. These neighbor rays can be used to calculate the angle between rays, which proves extremely useful in calculating the sizes of ray cones, as will be discussed in detail in Section 2.3. Similar methods of ray spawning, such as normalized cube which subdivides cubes instead of triangles, offer similar ray topology benefits. The ray neighbor sets can also be used to remove double counted rays in a novel way. Namely, the double counts can only occur with overlapping ray cones, where the overlap represents the same wavefront. Any two ray cones that overlap and take the same path

from the source to the receiver must represent the same wavefront which therefore will be counted twice. So any two rays that are neighbors that take the same number of reflections from the source to the receiver will be double counted, and one is easily removed.

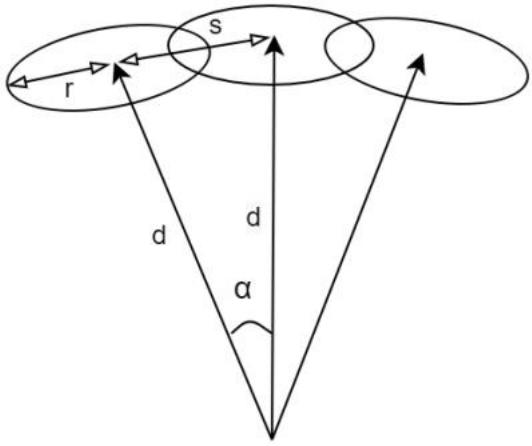


Fig. 2.2: Geometry of showing an adjacent and non-adjacent ray cone.

To prove that overlap, and thus double counted wavefronts, is limited to adjacent ray sets, we can use the geometry shown in Fig. 2.2. The condition that must be satisfied is $s/2 < r < s$. This condition ensures overlap between adjacent rays to provide full radiation coverage, while preventing overlap between non-adjacent rays. From the figure, and using the law of cosines, we get $s = \sqrt{2d^2(1-\cos\alpha)}$. Assuming that α is small and taking the small angle approximation of $\cos\alpha$, we get that $s \approx \alpha d$. Thus, the inequality becomes,

$$\frac{\alpha d}{2} < r < \alpha d. \qquad (2.2)$$

As long as α is chosen as the largest angle between a ray and its neighbors, then the left-hand side of the inequality ensures full radiation pattern coverage. We see that (2.1) satisfies (2.2), showing that it is a good choice for the radius of the ray cone. Equation (2.2) leads to two important conclusions: (i) Because ray cone overlap is limited to adjacent rays, the path search can be limited to only a neighbor ray set, which drastically reduces the computational complexity

of double count removal; (ii) This highlights the importance of choosing α correctly. If this angle is taken to be a constant value for instance, it is possible to cause undercounting since ray distribution is not constant and it is possible that not all ray cones overlap with all adjacent rays. Undercounting is problematic because it cannot be identified in post-processing. Choosing a constant value that is too large can also lead to cone overlap between non-adjacent rays, which expands the search for double count removal.

For the context, previous work in double count removal uses one of two common methods, where in the first approach, a characteristic sequence of a ray's path can be tracked, and any rays at the same receiver with the same characteristic sequence represent a double count, such as in [4]. However, this method quickly exhausts available memory, in addition to lacking parallelizability. The second method uses geometric path searches to identify double counted rays by calculating the distance between rays at a reception and removing those that are within the cone radius, such as in [3],[18]. However, this technique requires a large number of memory accesses and thus represents a bottleneck in the simulation. Our double count removal technique requires no additional memory storage because the adjacent ray sets are calculated only once in the preprocessing portion of the simulation. Additionally, memory checks are limited due to only having to search a single set instead of multiple ray distance accesses. This means that our method not only is computationally more efficient, but it is also more memory friendly, which allows for larger and faster simulations.

2.3 Per-Ray and Optimal Constant Ray Cone Angle Calculations

There is a very important relationship between ray cone sizes and simulation accuracy. In the SBR model, the distribution of rays being launched from a receiver becomes vital for the accuracy

of the model, where an even distribution is desired [17]. However, when launching rays in 3-D, it is not possible to have exactly uniform distribution of rays.

We choose the inscribed icosahedron spawning pattern because of its regular and predictable angular distribution of rays. Equation (2.1) gives the radius of the ray cones required for full radiation pattern coverage with the minimum possible overlapping of the cones. This formula requires that the angle α is the maximum angle between neighbor rays. Generally, the distribution of ray angles is assumed to be approximately uniform, and a constant value of α is used to calculate ray cone sizes. On the other hand, as can be seen in Fig. 2.3, the angular distribution of rays is significantly larger at the centers of the icosahedron faces and is smallest at the vertices. For other ray spawning patterns, this discrepancy is typically even more pronounced. This variation in angular distribution between rays can prove to be problematic if a constant α is assumed (unless the proper value for α is chosen).

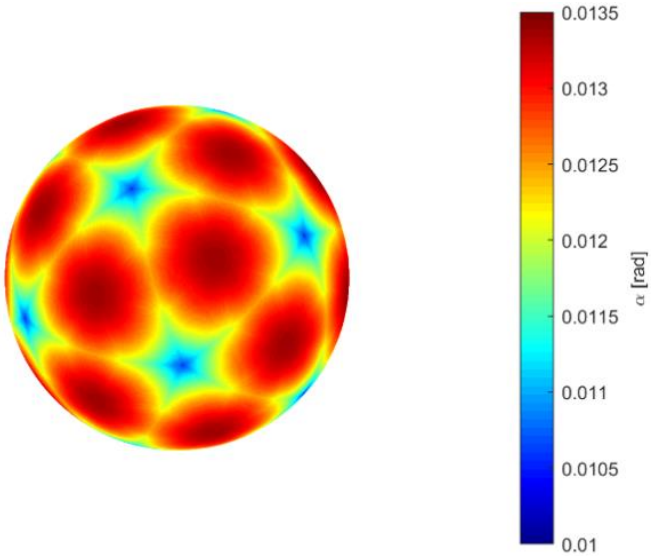
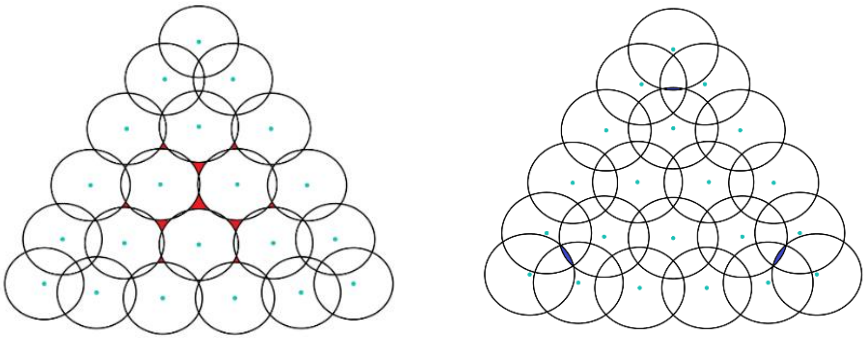


Fig. 2.3: Angular distribution of rays on an icosahedron: distribution of the maximum angle between neighbor rays (α) across an icosahedron spawning procedure with $n = 100$ subdivisions along an edge of the icosahedron face.

Using the adjacent ray sets developed for use in double counted ray identification, we can also calculate the maximum angle between rays for every individual ray. This is advantageous for several reasons. Firstly, using these individual α values allows for the smallest theoretical error of this SBR method, because cone sizes are as small as possible for every ray while ensuring full radiation coverage, which minimizes phase error. In complicated environments where cone cutoff is an issue, this method of calculating cone size will allow for the smallest possible error. Additionally, in adaptive ray spawning procedures, the method of calculating angular distributions for every ray allows for arbitrary refinement of the ray spawning while keeping physical interpretation of ray cone sizes. Lastly, per-ray calculations of angular distributions of rays allows for a unique analysis of the effect of ray cone sizes on the accuracy of the method.

If a constant angular distribution is assumed, this value must be chosen wisely so that error in the method is minimized. If α is chosen to be too small, the cones calculated on rays at the center of an icosahedron face will be too small, leading to a lapse in radiation pattern coverage, namely, undercounting. These field lapses grow as rays propagate in large structures, leading to large errors in the simulation. Fig. 2.4(a) shows a face of the icosahedron with cones inscribed calculated with too small an angle.

On the other hand, if α is chosen to be larger than the optimal value, then cones will become too large towards the vertices of the icosahedron. This means that the cutoff issue will be more significant in these cases. Furthermore, non-adjacent ray cones will start to overlap, leading to triple or quadruple counting of field contributions. These overlaps will not be identified in the double count identification procedures, meaning that this can lead to error in the model. A case with the cones calculated using α greater than ideal is shown in Fig. 2.4(b).



a) (b)

Fig. 2.4. Illustration of ray cones calculated based on (1) with d = 1 and (a) α chosen smaller than the optimal value, with lapses in radiation coverage being filled in red, and (b) α chosen larger than the optimal value, with overlaps between non-adjacent ray cones being filled in blue.

Based on Figs. 3 and 4, the most sensible choice of α for cones with the minimum overlap is the greatest maximum ray angle across the whole face. This ray lies in the center of the icosahedron face as seen in Fig 3. The optimal angle can be found analytically using the golden ratio, with the radius of a sphere inscribed in a regular icosahedron being [22]

$$r = \frac{\phi^2}{2\sqrt{3}}a = \frac{\sqrt{42 + 18\sqrt{5}}}{12}a \approx 0.755a \qquad (2.3)$$

where ϕ denotes the golden ratio and a is the edge length of the icosahedron. The edges are subdivided equally in a regular icosahedron so that the segment length is the edge length divided by the number of segments, $l = a/(n-1)$, with n again standing for the nth triangular number used in icosahedron subdivision. This scenario is shown graphically in Fig. 2.5, from which it is apparent that $\alpha = \tan^{-1}(l/r)$. After some rearranging, we are left with

$$\alpha = \tan^{-1}\frac{2\sqrt{3}}{\phi^2(n-1)} = \tan^{-1}\frac{12}{(n-1)\sqrt{3}\left(3+\sqrt{5}\right)} \qquad (2.4)$$

This equation for optimized α is verified and tested in Section 2.4.

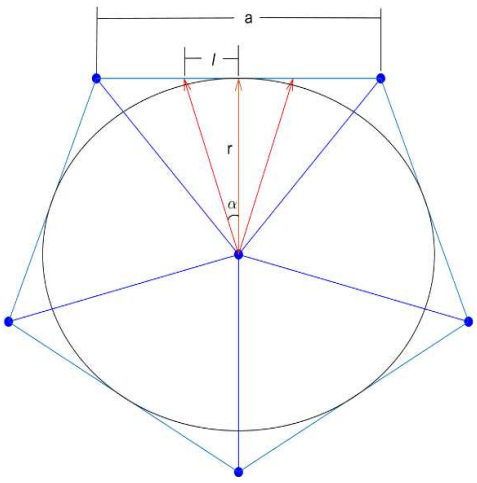


Fig. 2.5: Circle inscribed in an icosahedron: the radius of the inscribed sphere is given in terms of the edge length of an icosahedron face via the golden ratio.

2.4 Results and Discussion

Validation of the advanced SBR RT technique and demonstration of the accuracy improvements are presented here using several tunnels of different uniform cross sections. Tunnels are chosen as a challenging environment for RT solvers, due to the large distances covered and the higher order modes present in propagation. Tunnels offer a corner case enabling a rigorous test for SBR algorithms. The simulations in these tests were performed by creating a line of reception points directed down the long axis (z-axis) of the tunnels at evenly separated intervals. The solutions obtained using the presented SBR RT algorithm are compared with computed results of other asymptotic and full-wave CEM solvers that offer state-of-the-art performance for tunnel simulation.

2.4.1 Validation Against Image Theory

The first set of results is for comparison against an IT simulation. While IT is still an asymptotic approximation, it represents the perfect case for ray-based methods since the phase error is almost entirely eliminated. Our SBR method is compared to IT in a rectangular dielectric waveguide with

cross-sectional dimensions 4 m × 4 m × 1000 m [7]. The rectangular waveguide is particularly appropriate for comparison with SBR because of the accuracy of IT in this structure. In planar geometries, IT produces the most accurate ray tracing results [7]. The waveguide is excited with a vertically polarized isotropic transmitter at 1 GHz located at (x, y, z) = (1.1 m, 2.1 m, 0), with the x, y, and z coordinates being measured from the bottom left corner of the opening of the waveguide when viewed toward the wave propagation. The receivers are also isotropic and located at (1.9 m, 1.7 m, z), chosen to avoid any advantage due to symmetries in the simulation. The tunnel wall parameters are relative permittivity $\varepsilon_r = 5$ and conductivity $\sigma = 0$.

In Fig. 2.6, we see that the SBR solution agrees very well with the IT reference results. The SBR method uses α calculated per ray, showing that this technique is viable for simulating very large-scale environments. In fact, we observe a perfect match of the two sets of results as far as 1,000 m down the tunnel, which is a truly exceptional result that demonstrates an ability of our SBR RT method to correctly compute transmission path lengths at very large distances from the transmitter, a feature traditionally reserved for the dramatically less efficient image-theory RT.

2.4.2 Double Count Removal

Since the SBR results in Fig. 2.6 match so well with IT, this case also seems ideal for examination of the effects of double counting on

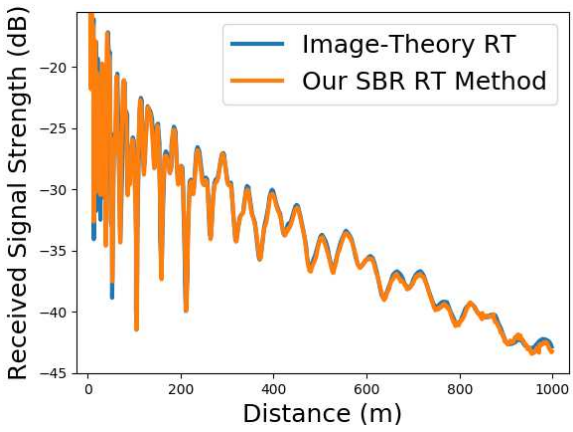


Fig. 2.6: Comparison of the presented SBR method with IT [7] in a dielectric ($\varepsilon_r = 5$, $\sigma = 0$) waveguide at 1 GHz [transmitter at (1.1 m, 2.1 m, 0), receivers at (1.9 m, 1.7 m, z)].

the accuracy of the model. Fig. 2.7 shows the absolute errors of SBR results with respect to reference IT data computed with and without double count removal, respectively. The results emphasize the substantial impact that the double counted rays have on the accuracy of SBR simulations. When compared to results in [3] and [4], we observe a similar error reduction from our technique as those in existing literature. This shows that the proposed technique similarly reduces the computation error while drastically reducing the complexity of the double count removal search.

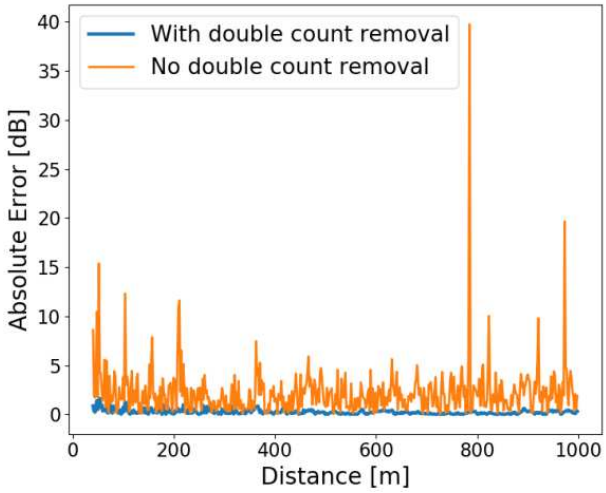
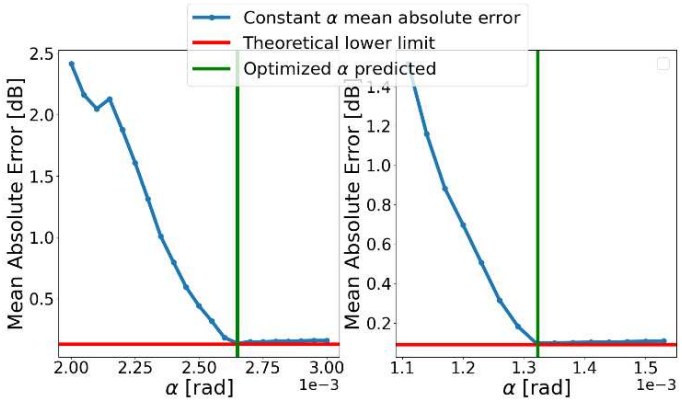


Fig. 2.7: Double count removal results. Absolute error in dB of our SBR simulation with respect to the reference IT results [7] for the dielectric waveguide with and without double count removal.

2.4.3 Ray Cone Angles

Next, we compare the model's ability to use a constant angular distribution given in (2.4). The per-ray α calculations are the theoretical best-case scenario for simulation accuracy, so simulations using constant α are compared to the per-ray α case. We generate simulations with an assumed constant α and plot mean absolute error of the simulation when compared to IT results. In this way, we expect that the mean absolute error is at a minimum when using the analytical expression derived in (2.4). The results of this comparison are shown in Fig. 2.8. We draw several interesting conclusions upon examining these results. Firstly, the value of α predicted by (2.4) perfectly matches with the lowest error in every case. This empirically demonstrates that the derived value of α is indeed optimal. Additionally, we see that the error also increases after this optimal α as expected. The increase in error after the optimal value is smaller than expected. The reason for this comes in two parts. First, with such a regular structure as the rectangular dielectric waveguide, the tube cutoff that is expected with cones that are too large does not happen, so this does not contribute to the error in this case. In an analysis on a more complex and nonuniform structure, we expect the

error to jump up more drastically. Second, we expected an increase in error due to ray cone overlap on non-adjacent rays. In these tests, the face of the icosahedron is oriented down the length of the waveguide, which means that the rays with extra overlap do not penetrate far into the length of the structure. This means that the error will not jump up drastically in this case either.



(a) (b)

Fig. 2.8: Mean absolute error of SBR calculation using a constant angular distribution, with respect to reference IT results, with the vertical line marking the optimal value of α predicted with (2.4) and the horizontal line representing the error using per-ray α: results for (a) $N = 500$ subdivisions and (b) $N = 1001$ subdivisions.

2.4.4 Comparison to Full-Wave Simulation

Next, we compare the SBR simulation to a full-wave technique for a mine tunnel with an arched ceiling presented in [23], where the tunnel is modeled using the surface integral equation (SIE) technique accelerated by the fast multipole method in conjunction with the fast Fourier transform (FMM-FFT). The FMM-FFT accelerated SIE simulations of mine environments are rigorously verified by comparison against measurements [23]. The tunnel considered here has an irregular, curved cross section. However, the SBR methodology encounters difficulty when planar segments are used to approximate curvature, because many segments can lead to many non-physical images produced in the model. Due to these issues, the curvature is approximated using an equivalent rectangular cross-section approach, as shown in [24]. The results of the SBR and FMM-FFT-SIE

simulations, with the material parameters of tunnel walls being $\varepsilon_r = 5.5$ and $\sigma = 0.15$ S/m and the operating frequency 915 MHz, are displayed in Fig. 2.9.

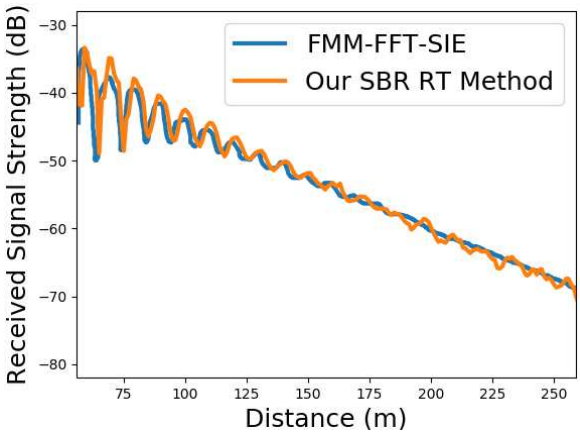


Fig. 2.9: Comparison of the SBR ray-tracing solution to the FMM-FFT accelerated SIE simulation of a mine tunnel with arched ceiling [23]. The tunnel is excited with a 915-MHz vertically polarized Hertzian dipole antenna and the material parameters are $\varepsilon_r = 5.5$ and $\sigma = 0.15$ S/m.

We see an excellent agreement between our SBR method and the full-wave technique. This SBR simulation is generated using α derived in (2.4), but the results generated using per-ray angles are indistinguishable from the results in Fig. 2.9. This shows the robust nature of the proposed SBR technique and its ability to provide accurate modeling of wireless propagation in large environments. Moreover, while providing comparable accuracy in the tunnel simulation, the proposed SBR RT technique is dramatically more efficient than the fast full-wave solver; the FMM-FFT-SIE solver takes over 19 hours to simulate this environment on a multi-GPU cluster [23], compared to less than 1 minute for the SBR technique on a modest single-GPU desktop (making use of parallelism from the Chapter 3).

2.5 Conclusion

This paper has proposed and presented the shooting and bouncing rays method for ray-tracing modeling with improved accuracy, with accuracy improvements coming from several components of the SBR computation. The paper has examined the effect of the calculated sizes of the SBR ray cones on the accuracy of the simulations, proving a very high importance of the proper choice of the distribution angle for the overall performance of the SBR method. We have proposed calculating the ray cone sizes based on the maximum separation angle for every ray, individually. This per-ray calculation is an excellent choice in very complex environments and allows for adaptive ray spawning techniques to be implemented, but it comes at the cost of increased computational complexity in the model. The alternative is to assume a constant angular distribution of the rays, which allows for much more expeditious simulations but makes the overall accuracy of the computation extremely sensitive to the choice of the angular distribution. The paper has given the optimal choice for this angle, which has been shown to be viable in large tunnel environments. Additionally, we have shown the importance of removing double counted phase fronts in the SBR method that implements the ray cones approach. The presented technique uses icosahedron geometry and adjacent ray sets to identify and remove double counted ray contribution without the need for the analysis of the full geometry of the ray paths and lengthy path searches. The improvement of the magnitude accuracy due to the double count removal couples with the improvement of the phase accuracy associated with the optimal distribution angle of rays launched from the transmitter.

The results have demonstrated that the SBR method with the proposed, implemented, and evaluated accuracy improvements can perform wireless propagation modeling of tunnel environments with the same accuracy as the image theory RT, a computationally much slower but

traditionally more accurate solver. We have shown a perfect match of the improved SBR method and the IT method for a kilometer-long tunnel, where it is important to note that correct computation of transmission path lengths at very large distances from the transmitter is a feature traditionally reserved for the dramatically less efficient image-theory RT. We have shown the SBR model's ability to effectively use both the per-ray α calculations and an analytically derived constant angular distribution. The results have empirically demonstrated that the derived value of α is indeed optimal. The SBR simulations generated using the derived optimal constant α are indistinguishable from those generated using per-ray angles, the latter α calculations being the theoretical best-case scenario for simulation accuracy. This demonstrates the robustness and usability of both approaches. The example with comparison of the advanced SBR method with the FMM-FFT accelerated SIE simulations has demonstrated the accuracy of the proposed SBR RT technique in analysis of electromagnetic wave propagation in mine tunnel environments comparable to a full-wave solution, while benefiting from the efficiency of computation characteristic for asymptotic ray-based methods.

Overall, the main contribution of this work is an SBR ray-tracing method of similar accuracy as the IT RT approach. While demonstrated for electromagnetic propagation modeling of large tunnels, the proposed accuracy improvements of this SBR methodology should prove beneficial in other applications, either on its own or hybridized with other CEM approaches, in indoor and outdoor wireless propagation modeling. With growing challenges imposed by emerging communication technologies, e.g., *n*G communication systems, the "renaissance" of asymptotic CEM simulations will just be intensified, and hence a need for this and similar advancements of SBR and other general classes of asymptotic CEM techniques.

Chapter 3

Parallel GPU Optimization of the Shooting and Bouncing Ray Tracing Methodology for Propagation Modeling[2]

3.1 Introduction

Ray tracing (RT) is a high-frequency asymptotic method that provides an alternative approach to solving electrically large computational electromagnetics (CEM) problems that traditionally require prohibitively large computation times using full-wave CEM solvers [2],5],[17]. As a significant advantage in terms of efficiency over full-wave methods, the shooting and bouncing rays (SBR) RT technique features a time complexity of $O(NK)$, where N is the number of rays and K is the number of reflections [1],[2]. The SBR algorithm also has substantially lower computational complexity than the image theory (IT), another RT approach, the complexity of which is $O(N^K)$, where N is the number of planar facets in the model and K is the number of reflections. IT is also limited to planar structures because in curved geometries the concept of images becomes ambiguous with an image from a curved surface being a line instead of a discrete point [7].

Generally, CEM background and advancements tightly rely on those in (1) electromagnetic/physical formulations and contexts of engineering problems, (2) mathematical

[2] This chapter is based on work from Ref. [12]

and numerical foundations of methods and algorithms, and (3) computing hardware and software infrastructure. Consequently, CEM research is one of the most natural, intense, and advanced combinations of engineering, physics, mathematics, and computer science, with extremely exciting potential and challenges. Indeed, it is the category (3) that has recently provided unprecedented opportunities, with the associated challenges, for the rapid growth of CEM capabilities and modeling technologies needed and desired by electromagnetics application researchers and practitioners, in close synergy, however, with categories (1) and (2). This paper focusses on some of the new ways to better harness the advantages of emerging and growing computing hardware and software infrastructure for CEM, and RT in particular.

Most notably, the development and abundance of graphics processing units (GPUs) has created the enabling technology to achieve massive parallelization. In fact, enormous advances in GPU technology have made these hardware accelerators widely useful outside of the computer graphics world. Originally for graphics applications, GPU architectures are optimized for inherently parallel operation. GPUs are optimized to handle tens of thousands of threads to keep up with the billions of linear algebra calculations required for a high-end video game rendering in real time on an HD monitor. GPUs are therefore ideal candidates when large scale parallelization of simple operations is required [25]. Due to their unique architecture, GPU programming is often done with the compute unified device architecture (CUDA) language, an application programming interface (API) created by GPU manufacturer NVIDIA for use with their products.

In addition to its advantages in terms of memory and computation time requirements, the SBR RT method can benefit immensely from GPU acceleration, due to the linear nature of the ray decomposition, enabling independent tracing of each ray (excluding ray overlap processing). Considerable work has been done to parallelize the method for implementation on GPUs, such as

[9], [10]. However, with the SBR RT application consisting of (1) geometric and physical path calculation of rays as they propagate through a given structure and (2) field calculation of each ray, it is part (1) where good parallelism has been achieved. Portions of the code remain serial, particularly part (2), namely, electric field computations from RT results.

Accuracy improvements of several components of the SBR computation can lead to a SBR method capable of performing wireless propagation modeling of tunnel environments with the same accuracy as the IT RT, a computationally much slower but traditionally more accurate solver [2], [8]. A ray classification method presented in [8] allows for parallelization of the double count removal portion of the SBR algorithm, which traditionally has been very difficult to parallelize.

This present paper proposes a novel unified parallelization framework of algorithms, strategies, and data structures to radically enhance the efficiency of the SBR RT method by integration of SBR with OptiX Prime, comprehensive parallelization of all components of SBR computation, including electric field computation and postprocessing tasks being traditionally limited to sequential operation, and addressing and optimizing memory usage and constraint issues to further advance efficiency of the overall method. The paper shows excellent parallelism of the method, as well as extremely substantial reductions in computation time when compared to the existing methods.

Details on the approach to computing, storing, and updating the electric field for each ray are described in order to develop an understanding of the flow of data and processing on the GPU. We introduce the important optimization strategies used for writing efficient executable code on NVIDIA GPUs. This includes the importance of structured global memory accesses that limit the total number of reads and writes to memory necessary and the inevitability of GPU synchronization. Configuration optimization of the GPU kernels for the RT program is compared

with changes in performance from different configurations. Specific implementations of the RT algorithm to incorporate these strategies are discussed. We believe that the presented algorithms, strategies, and data structures should be very useful to researchers, developers, and practitioners of ray-based and other high-frequency methods and modeling. They should also be of significant interest to the area of parallelism and efficiency enhancement of CEM techniques in general.

The results demonstrate that the new SBR methodology achieves upwards of 99% parallelism, whereas no such scaling performance, or close to it, has been reported for RT computations so far. We assess the level of parallelism using Amdahl's law, a predominant scaling law for systematic prediction, assessment, and understanding of parallelization speedup and parallel scaling properties of a code [25]. This dramatically reduces computation time for convergence in tunnels (tunnels are challenging for RT simulations due to the difficulty in simulating multipath propagation effects in large tunnel environments by RT approaches), for example, as well as in an urban outdoor wireless propagation environment. Overall, the computational accelerations achieved from the very high level of parallelism yield an SBR ray tracing methodology of unparalleled efficiency, without sacrificing the previously advanced and established accuracy of the method [2]. Importantly, the major enhancements of the efficiency (this present work) have been enabled by and addressed synergistically with the improvements of the accuracy [2] of the SBR computation. Note that a portion of this work has been reported in a M.S. Thesis [26]. A very preliminary brief report has also been presented in [20]. Finally, we insist here on a distinction between parallelization and parallel optimization of the code; namely, without the enhancements proposed in this paper, conventional parallelized SBR will result in several orders of magnitude more computation time for the same accuracy.

3.2 OptiX Prime Integration in RT CEM Analysis

In our approach to a RT CEM analysis of a propagation structure or system, the environment is meshed into discrete triangular facets. We use the NVIDIA OptiX application programming interface (API) to compute the ray-facet intersections in order to trace the rays as they propagate through the environment once launched by the antenna. In most scenarios, the medium of propagation is assumed to be air. Once the ray has reflected, the triangles or faces that have been intersected are used to calculate Fresnel coefficients, which are then used to calculate the attenuation of the electric field based on the material composition of the facet. Such generated ray information is used to populate a buffer of ray type objects, and this buffer is one of the main programming components available in the NVIDIA OptiX Prime RT API. The buffer is passed to the OptiX Prime ray intersection program that runs on the GPU and determines the path of a ray from an origin point to a destination point along the direction specified in the spawning. It is indeed advantageous to use the OptiX Prime API given a tremendous research and implementation effort by NVIDIA over the years in developing OptiX Prime as a general-purpose RT engine [11].

An essential feature and a primary reason for using OptiX Prime is the efficient construction of the binary space partition (BSP) tree. The BSP tree is a tree type data structure that stores subdivisions of an environment and allows for efficient ray triangle intersection tests by pruning out sections of the space that are not physically reachable from other sections of the space. In order to construct the BSP-tree and perform ray-facet intersection tests, the OptiX Prime RT API needs a geometry description file to build a mesh. The OptiX Prime API then loads the mesh and executes a closest hit query for every ray in the ray buffer and calculates the nearest intersection with a triangle for that ray. Here, an efficient BSP tree reduces the total number of facets that have to be checked for a ray in order to find which one it hits first [27]. The field evaluation can then be

performed based on the computed intersections and the associated ray reflections. After the ray reflection calculations, the information regarding the previous path segments the ray has traveled is used in postprocessing to determine any intersections the ray may have had with an observation point.

Note that certain anomalous scenarios can occur while a ray is in the geometric portion of the SBR RT algorithm, and they need proper attention and mitigation. The first such scenario occurs when the ray escapes the mesh that it is propagating in, which is a product of numerical precision error that allows rays that hit directly on the boundary of two facets to escape the mesh structure. This happens rarely and is mitigated in the RT application by identifying the escaping rays in the buffer and removing the respective information from the simulation. In the second scenario, a ray leaves the region of interest, or the section where the observation points are located, without leaving the actual mesh. In order to limit this region of interest, a bounding box consisting of multiple "absorbing" planes is created to terminate any ray that intersects the box.

3.3 Optimization of Parallelized SBR Ray Tracing CEM Methodology on GPUs

3.3.1 SBR RT GPU Computation Theory and Implementation

The shooting bouncing RT program involves the spawning, propagation, electric field computation, and electric field summation of millions of rays. The ability to launch millions of rays is highly dependent on the structure and efficiency of the code as well as the level of parallelization achieved. SBR RT is a highly parallel procedure because each ray propagates independently and any computation involving a ray can be performed without interference from other rays. GPUs appear to be ideally suited for an overreaching goal of fully benefiting from the

inherent parallelism of the SBR process, in terms of making the computation as efficient as possible. Specifically, GPUs are effective with this type of highly parallel programming as each basic work unit, or thread, on the GPU can compute, track, and update an individual ray's information. In contrast to central processing units (CPUs), GPUs are optimized for data parallel throughput computations and low latency access to cached data sets. Whereas CPUs are designed to control logic for out of order and abstract execution, GPUs perform best on organized instruction sets with limited control-flow branching. GPUs are also architecturally more tolerant of memory latency and have physically more transistors that can be dedicated to arithmetic computation [27]. For all these reasons, GPUs are highly advantageous for RT applications.

The compute unified device architecture (CUDA) parallel computing platform is the primary language used to program on NVIDIA GPUs, permitting the creation of parallel computational instructions to very substantially increase efficiency from the GPU cores and decrease computation time. The device code for NVIDIA GPUs is written in CUDA C/C++ and the executable code on the GPU runs kernels that are synchronous computation batches containing sequential instruction sets. The sequential code located inside the kernels is executed by the threads that are spawned at the start of the kernel execution.

Understanding the structure and execution procedure of NVIDIA GPUs is essential for efficient parallelization of the SBR RT application. The finest execution unit on the GPU is the thread, with GPU threads being similar to threads on a CPU and executing instructions sequentially. They are grouped into thread blocks and all thread blocks used in the execution of a kernel compose a grid. When a kernel is executed on the GPU, the thread blocks in the grid are distributed to the available streaming multiprocessors (SMs) located on the GPU. Thread blocks do not begin execution until an SM has enough resources for all the threads inside the thread block to execute. The SMs have

shared memory that can be used by all the threads inside that block. The threads located in the block running on the SM cannot all run at one time and the SM typically, in the case of the NVIDIA GTX 1060 GPU used for the SBR RT application in this work, can execute a group of 32 threads, known as a warp, at one time. Warps run in parallel, allowing these groups of threads to execute their instructions rapidly.

Parallel execution, however, introduces new complications: race conditions are extremely frequent in parallel programs and especially in the RT application and can destabilize a program without conscious handling. The main race condition occurs when the electric field at an observation point is modified by adding the electric field from all the rays that have intersected the same observation point. Since many threads on the GPU are all executing concurrently, many of the threads would be modifying the memory containing the electric field for the observation point simultaneously. This means that the electric field for each observation point may not be updated properly by the threads performing the postprocessing calculations. To eliminate this race condition, the threads must synchronize to ensure mutual exclusion when reading or writing to the memory where the electric field for the observation points is stored. The downfall with this synchronization is a loss in computation time due to threads having to wait for one another to update the electric field.

Effective GPU programing ensures that work is given to all the pipelines available on the GPU and increases computation throughput by hiding the latency of the memory pipelines. To maximize efficiency on the GPU, ideally all the SMs should be running code continuously with other computations running asynchronously. The amount of memory available on the GPU and on the host CPU are the primary limiting factors of the SBR RT application. The memory requirements of the GPU only allow a fixed number of rays to be computed at a time. Because of this constraint,

for some larger structures, the rays need to be split into batches before being sent to the GPU. The first type of batching involves the subdivision of the faces of the icosahedron used to launch the rays initially (we employ an icosahedron for the sampling method in our spawning technique for the RT application) [2]. The other type of batching utilized in the SBR RT application is the batching of the reflection calculations for each of the rays. If the ray's reflection calculations are not batched, then storing all the path information for a ray over hundreds of reflections would overflow the memory. This is because each ray reflection consists of the ray's previous and current intersection points each represented as three floating point numbers and the identification of the triangle that was hit represented by an integer type. This stored information equates to 28 bytes of memory for each reflection, and for 100 million rays, this would be 2.8 gigabytes of storage per reflection. Hence, as the number of reflections increases, only information about a ray's current reflection segment is tracked to limit this memory footprint, and after all paths in a batch of rays have been computed, the next batch is processed. This procedure is repeated until no remaining rays need to be traced for intersections with the observation spheres or the limit on the number of reflections has been reached.

Maximum efficiency with any kind of parallelization, especially on GPUs, requires the optimization of the program to get the best performance possible. The first type of optimization prevalent with all NVIDIA CUDA GPU programs is the launch configurations of the kernel. The launch configurations refer to the number of blocks and number of threads per block that are created to run the code within the kernel. The downside of having too many spawned threads is that some threads will be idle while the other threads are doing all the work. Conversely, if there are too few threads then there could be resources on the GPU such as CUDA cores and SMs that are not being utilized for computation. To get the best efficiency, all compute units on the

GPU should remain occupied with computation. The RT application is memory bound, meaning that the bottleneck in simulations is primarily due to the memory limits available to hold data, and the simulation therefore requires higher thread occupancy in order to achieve better efficiency. In our work, various kernel launch configurations were tested beginning with a block size of 128 threads and adjusted to the number of threads per block in increments of 32. The best results, in general, were obtained with 128 threads per block. The number of blocks is also an important consideration, and typically 1,000 or more thread blocks are necessary as this enables the code to be evenly distributed among the GPU hardware, but also highly modular and scalable with additional GPUs.

3.3.2 Optimal Parallelization of Ray Tracing Method

Fig. 3.1 shows the flow between the GPU and CPU in our SBR RT method, depicting the transfers between the GPU and CPU, which can be a severe bottleneck when running simulations. As can be seen, the method starts with loading the environment mesh into the CPU, which has all pertinent information about surface normals and locations. The ray-initialization-icosahedron is then generated, and the ray-neighbor adjacency map is calculated, which, in turn, is used to find exact ray cone sizes for computing ray-receiver intersections more exactly. This is the bulk of the work performed on the CPU, which remains constant as ray numbers and reflection counts increase. The batching loop is then started, with the 20 faces of the icosahedron being the default number of batches in the program. Initial ray directions are generated across one face of the icosahedron, subdivided according to the n^{th} triangular number [2], and these are copied to the GPU memory.

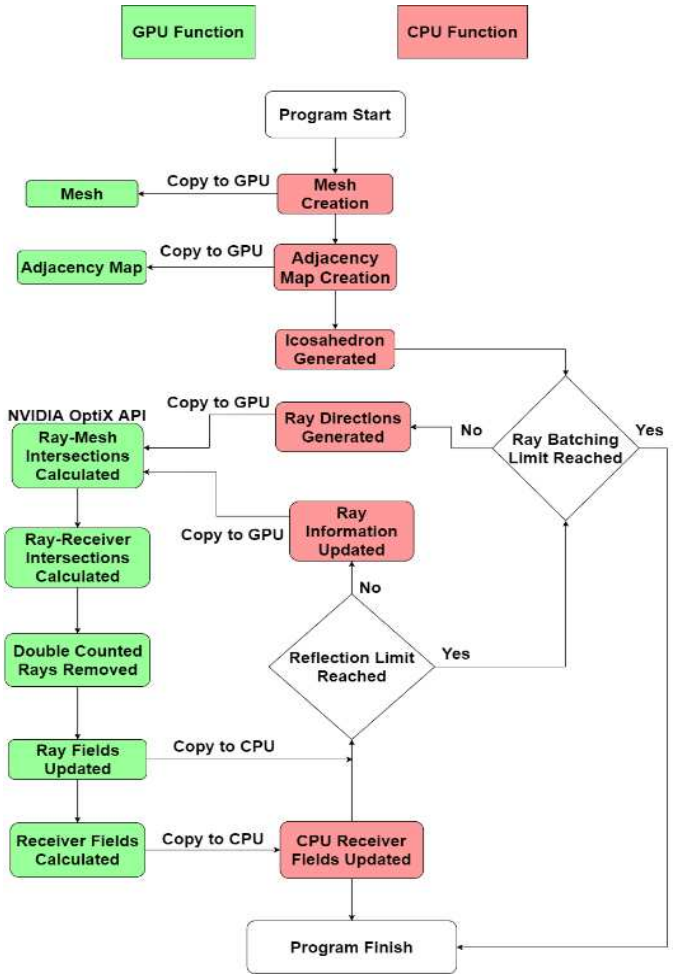


Fig. 3.1: RT methodology: Flowchart showing basic processes of the proposed RT method as well as transfers between the GPU and CPU.

Ray-facet intersections are calculated using the NVIDIA OptiX Prime API. In this routine, we calculate the intersections of these rays with set receiver points, and remove double counted rays by means of a new method described in [2], [8], which uses information from the ray adjacency map already created. After the ray double count removal, the ray fields are employed to update the receiver fields at every point, according to Fresnel's coefficients and reflection geometry. The ray information and receiver field arrays are copied to the CPU memory, which is the main transfer bottleneck – not severe, however, since both arrays contain a minimal amount of information. The copied receiver fields are added to the master copy on the CPU. If the reflection limit has not been

reached, new ray directions are calculated, and the loop is repeated. If the reflection limit is reached, the program moves onto the next batch of rays, and the process repeats. Once all the batches have been completed, the master CPU copy of receiver fields is output to a text file. As we can see from this process, the bulk of the calculations are done on the GPU, which helps to vastly accelerate the program. Bottlenecks between the GPU and CPU are limited, which also helps to increase speed.

Global memory (memory allocated on the GPU device from the host) throughput is an important optimization consideration when computing on a GPU. Transferring information from host memory to device memory is very slow and the number of copies needs to be minimized. In the RT application, the data for each ray is transferred to the GPU only once before the postprocessing after each reflection. The largest memory storage on the GPU is global memory which is up to 6 GB in the case of NVIDIA GTX 1060 GPU used in this work. Global memory is accessible by all threads running across all thread blocks in the grid, but it has very high access latency [28]. All memory operations performed in the execution of the kernel are issued per warp and the warp accesses memory in 32-word chunks. If the 32 threads in a warp access continuous addresses in memory, then the latency for that group of threads is reduced. These types of accesses to global memory are called coalesced accesses and they ensure all threads access a continuous chunk in memory and therefore reduce the total time for memory access. In order to achieve the coalesced memory accesses, our SBR RT application avoids scattered address patterns and global memory storage with large strides between accesses.

Ray double counting is a well-known issue with the SBR method, with different methods proposed to address this problem [3],[4],[29]. Recently, we proposed a double count removal method using icosahedron geometry and ray cones approach [2]. As illustrated in Fig. 3.2, if two

neighbor rays reach the same observation point in the same number of reflections, they represent the same wavefront, and one is easily removed. Efficient parallelization of the double count removal procedure is a very important aspect of the GPU optimization for the SBR code. However, the double count removal methods still suffer from the need to examine neighbor ray paths to identify the double counted rays, which is inherently not parallel. To parallelize this portion of the code, adjacent rays need to be split into different classes. The icosahedron spawning pattern allows for grouping rays into three distinct classes, in which no adjacent rays will be a member of the same class. Fig. 3.3 shows an example of the different classes displayed as three colors plotted over the faces of the icosahedron. Specifically, we split the adjacent rays into separate classes to ensure that no two adjacent rays that double count on the same observation point are processed at the same time, and this classing method is described in detail in [8]. If the GPU threads were to execute the double count removal simultaneously, namely, if the classing method were not used, two rays that have intersected an observation point and represent a double count would not be properly flagged, and the double count would be missed.

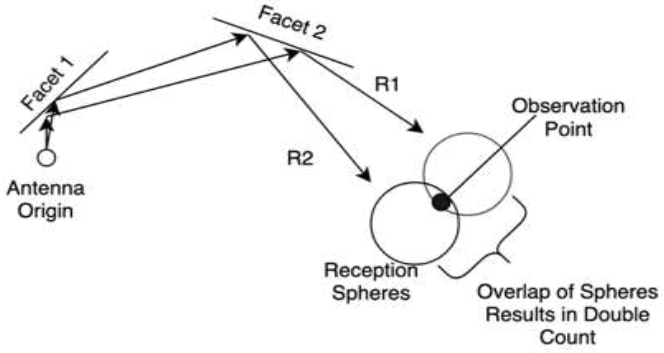


Fig. 3.2: Ray cone double counted wavefront**:** Illustration of overlapping ray cones, with multiple adjacent cones intersecting the same observation point (the black dot) due to the overlap.

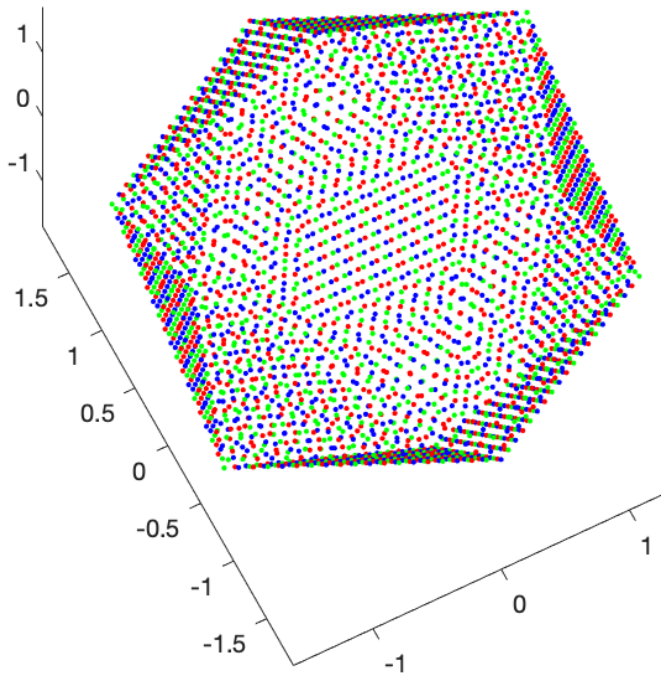


Fig. 3.3: Adjacent classes of the RT icosahedron**:** The three distinct classes displayed as red, green, and blue dots plotted over the icosahedron display how adjacent rays are members of different classes.

3.3.3 SBR Electric Field Computation Accleration

Typical SBR algorithms use the physical optics (PO) radiation approximation for the electric field computations. Essentially, the rays are used to "paint" surface currents in the environment, which are then entered in the PO radiation integral to compute scattered fields [17]. These integrations can be carried out for each mesh facet individually [9], which does allow for some degree of parallelization. However, the numerical integration is still computationally heavy for GPU implementation, which can result in an efficiency bottleneck. In the method presented here, instead of using the PO approximation, the SBR technique implements a ray-sphere intersection method, as detailed in [2], where ray field contributions are summed directly at reception points. The advantage is that each ray-sphere test is entirely independent from one another, as well as computationally efficient. This results in an electric field computation scheme that allows for embarrassingly parallel GPU methodology.

Our methodology uses the NVIDIA CUDA API to search ray path intersections in parallel, where every ray is checked for intersection with observation point spheres, which represent the radius of the ray-cone. Since it is possible for a ray to intersect multiple observation points, every ray must be checked with every sphere, which results in a computational complexity of O(*NP*), where *N* is the number of rays and *P* is the number of observation points. Due to the nature of arrays on GPUs, it is impossible to allocate an array in memory without knowledge of the array size, and, since the number of ray-observation intersections cannot be known a priori, ray-observation checks need to be conducted twice to achieve parallelism, as follows. Our technique checks for the ray-observation intersection, and if found true, it uses atomic add to create a total count of intersections, which is then used to allocate an array on the GPU. The same intersection checks are done again, but now we fill the array with electric field data that is used to calculate the total electric field.

While this unavoidable doubling of the ray-observation checks, for GPU parallelism, certainly decreases the efficiency of the analysis per se, given that typically both *N* and *P* are very large in RT applications, and so is O(*NP*), the overall order of magnitude for the computation time remains the same, and hence this method still massively benefits from parallelization. To illustrate this, Fig. 3.4. shows the comparison of the ray-sphere intersection method time when done on a GPU versus a CPU. In the CPU intersections, C++ vectors are leveraged so that the intersections do not need to be computed twice. Instead, a vector push back is used to extend the container for each valid intersection. However, due to the inefficient nature of vector reallocation, as well as the fully parallel computations on the GPU, we see an increase of up to 125× for just this portion of the code. This is also indicative of a massive increase in computational efficiency of the

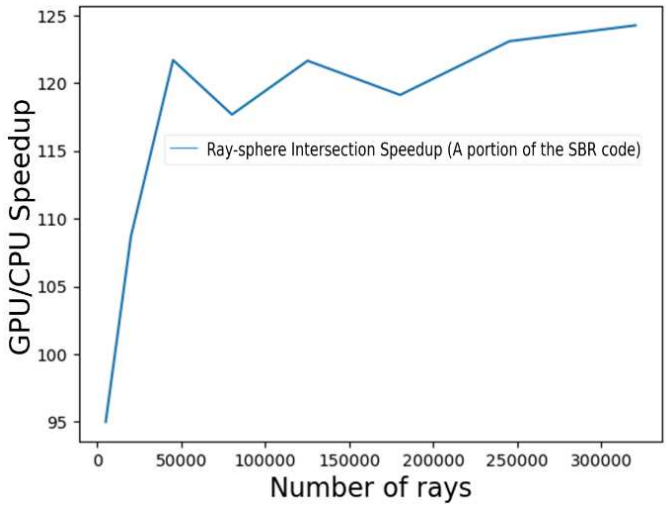


Fig. 3.4: Speedup of the ray-observation intersection method (only a portion of the overall SBR RT code), vs. the number of rays, as implemented on a GPU in parallel, with two passes of the intersection checks method, relative to the implementation on a CPU in serial, with intersections computed once.

overall SBR RT methodology with the proposed GPU parallelization, resulting in an unparalleled total computation time relative to other approaches.

3.4 Numerical Results and Discussion

As the first example of the computational speedup from the proposed novel NVIDIA OptiX Prime CUDA GPU parallelization framework for SBR RT CEM propagation modeling, Fig. 3.5 shows strong scaling results of the SBR code when compared to Amdahl's parallelization scaling law at 99.3% parallelism, when using 2.5 million rays and 20 reflections, which are typical values required for convergence of the SBR method. The number of CUDA blocks is set to 1 while increasing the number of compute threads per block from 1 to 1024, to increase parallel computations, where 1024 threads per block is a limit for this comparison imposed by CUDA. These strong scaling results indicate that the code has achieved about 99% parallelism, namely, a 99% parallel portion with 1% sequential rest. This is an excellent outcome, generally extremely difficult to realize due to many postprocessing tasks being traditionally limited to sequential

operation. We also perform the evaluation of the level of parallelism using Amdahl's law specifically for the electric field computation portion of the SBR algorithm and code, along with the parallelism for the entire algorithm and code. The results show very similar parallelism levels, of about 99%, for both the entire code and the field computation part.

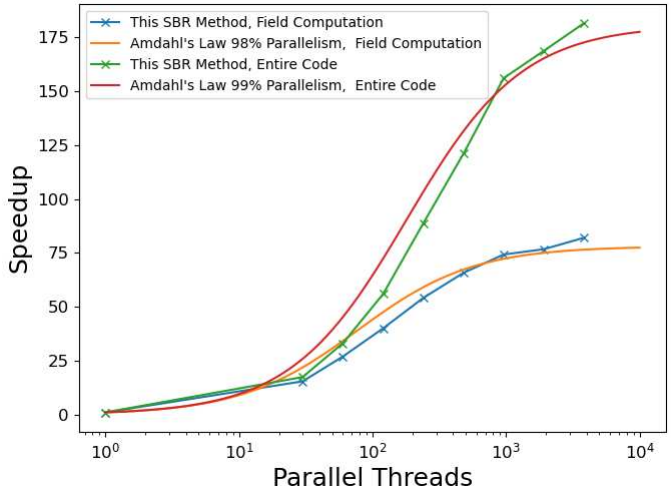


Fig. 3.5: Comparison of strong scaling SBR RT results with 2.5 million rays and 20 reflections to Amdahl's law with 99% parallelism, curve-fitted through the SBR data points. The GPU computations are performed a variable number of compute threads (per block).

As the next code efficiency demonstration, Fig. 3.6 shows a comparison with Amdahl's law of strong scaling results obtained with the same accelerated SBR framework as in Fig. 3.5 but for the number of threads per block set to 128 and the number of CUDA blocks swept from 1 to 4096. We observe from this experiment a strong speedup and approximately 97% parallelism in the code even when a larger number of total compute threads are used. We also see the speedup plateau at around 1000 blocks, which signifies full occupancy of the GPU (once the memory is at full occupancy, speedup is effectively bound) and numerically confirms related theoretical and empirical considerations.

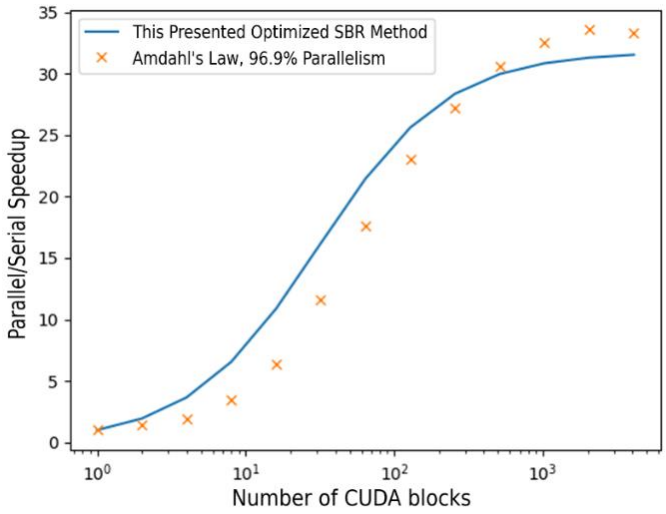


Fig. 3.6: Comparison of strong scaling results to Amdahl's law with 96.9% parallelism (curve fit through SBR points) for the same accelerated SBR framework as in Fig. 3.5 but with a fixed number (128) of compute threads per CUDA block and a variable number of blocks.

Next, Table I shows comparison of total computation times of the SBR RT propagation analysis conducted in serial and parallel, that is, for the results generated using a serial CPU implementation and with the parallelized GPU code, respectively. The simulations are performed using optimal values for compute threads and CUDA blocks for individual numbers of rays, on a standard workstation computer with an NVIDIA Geforce 1060 (6 GB) GPU. We observe massive parallel vs. serial speedups, which demonstrates the effectiveness of the proposed methodology in truly optimizing the SBR RT code and taking full advantage of NVIDIA OptiX Prime API and CUDA GPU programing interface for propagation modeling.

Table 3.1: Comparison of Serial and parallel Computation Times for SBR RT Propagation Computation.

Number of Rays (thousands)	100	400	900	1600
Serial Computation Time (seconds)	852	6181	29638	86214
Parallel Computation Time (seconds)	1	2.6	4.5	6.5
Speedup (×)	852	2372	6565	13263

As the next example, Fig. 3.7 shows propagation modeling results for a rectangular dielectric waveguide (tunnel) from [7], with dimensions 4 m × 4 m × 1000 m, tunnel wall dielectric parameters $\varepsilon_r = 5$ and $\sigma = 0$, and excitation by a vertically polarized isotropic transmitter at 1 GHz. Tunnels represent a challenging environment for RT solvers, due to the large distances covered and the higher order modes present in propagation, and thus an excellent test case. The SBR RT results are compared with the IT RT solution (image theory is known to produce the most accurate ray tracing results) and the results obtained by Ansys Savant SBR+ [30], which uses the PO approximation for electric field computations and is parallel compute enabled. We observe excellent agreement of the results obtained by the three methods. This confirms that the vast parallel optimization of all computational aspects of the SBR RT methodology proposed and described in this work, enabling such high parallelism and unparalleled efficiency, have not affected the previously advanced and established accuracy of the method (on par with the drastically less efficient IT RT approach) [2]. In fact, the presented major efficiency optimizations and enhancements have been enabled by and addressed synergistically with the improvements of the accuracy of the SBR computation.

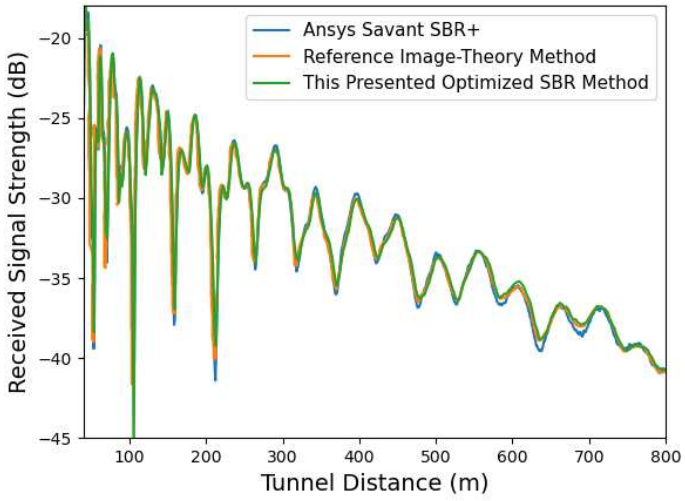


Fig. 3.7: Comparison of the presented optimally parallelized SBR method with IT [7] and Ansys Savant SBR+ [30] in a 4 m × 4 m × 1000 m waveguide tunnel with dielectric ($\varepsilon_r = 5, \sigma = 0$) walls at 1 GHz [transmitter at (1.1 m, 2.1 m, 0), receivers at (1.9 m, 1.7 m, z)].

As the final test case, we consider an example of an urban wireless propagation environment, the so-called "Madrid grid" urban scenario, often referenced as the METIS project scenario, which has been commonly used for evaluation and characterization of wireless propagation and parametrization of radio channel modeling in outdoor urban scenarios, typically via ray-tracing simulations [31],[32],[33],[34]. This test case is aimed at evaluation and demonstration of both the parallelism of the new proposed optimized SBR methodology and the accuracy and validity of the presented SBR computations. We consider the geometry shown in Fig. 3.8, in which both the buildings and ground plane are assumed to be concrete, with permittivity of 4.5 and conductivity of 0.09 S/m respectively. The heights of the buildings are adopted to be 50 m. The path from A to B to C represents a line-of-sight radio channel while the path B to D is a non-line-of-sight channel. The transmitting antenna (Tx), at location A, is a vertically polarized half-wave dipole at a frequency of 5 GHz, at a height of 10 m with respect to the ground. The receiving points are placed at heights of 5 m. Note that the presented SBR methodology is a pure SBR RT approach, and hence diffraction from edges is not included in the model.

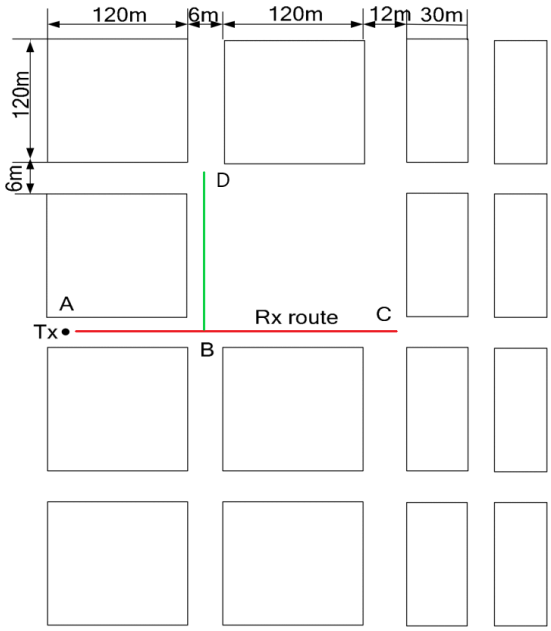


Fig. 3.8: Geometry of the "Madrid grid" (METIS project) outdoor urban scenario. The path A–B–C represents a line-of-sight radio channel while the path B–D is a non-line-of-sight channel. The transmitting antenna (Tx) at 5 GHz is at location A.

The validation of our SBR computations for this urban radio channel modeling scenario is performed using Ansys SBR+ as shown in Fig. 3.9. We observe excellent agreement of the present SBR method with Ansys SBR+ in both the line-of-sight and non-line-of-sight channels.

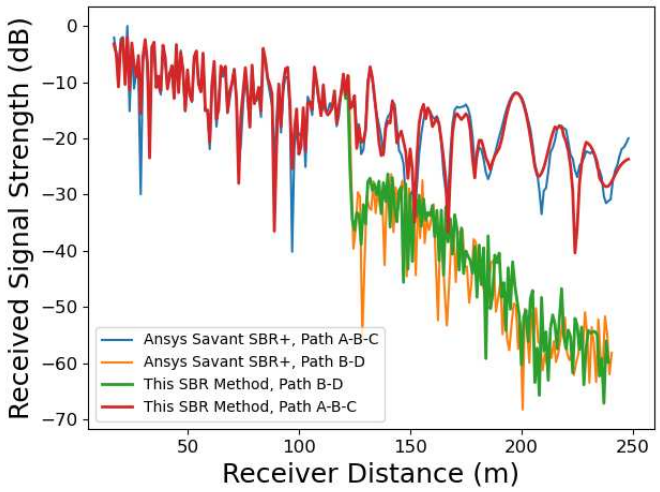


Fig. 3.9: Results for the METIS test case geometry from Fig. 8. A vertically polarized half-wave dipole is launched from Tx and traced through the Rx path shown at 5 GHz. The buildings and ground are both assumed to be cement with a relative permittivity of 4.5 and conductivity of 0.09 S/m.

While showing this excellent agreement of the results, we also demonstrate massive parallelization for this test case. The results in Fig. 3.10 show very high parallelism levels, namely, upwards of 99% parallelism, for both the entire code and the field computation part in the "Madrid grid" urban scenario SBR simulations, similar to the tunnel test case.

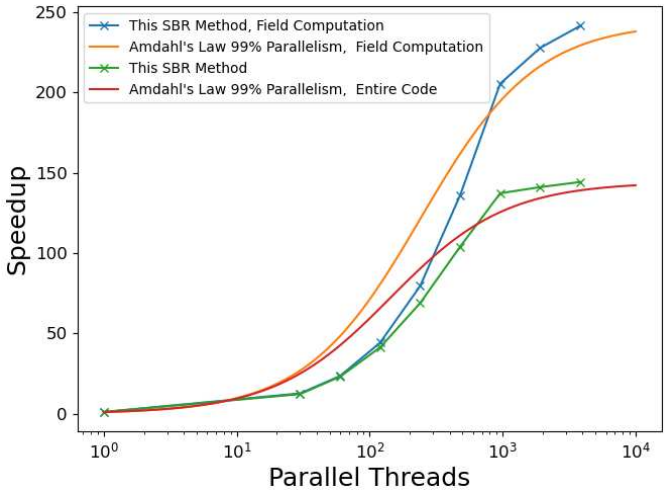


Fig. 3.10: Strong scaling for the METIS test case. We show scaling for the electric field computation portion of the code as well as the total scaling for the SBR method (the entire code).

3.5 Conclusion

This paper has proposed and described a novel parallelization methodology to maximally enhance the efficiency of the developed cone-based shooting and bouncing ray tracing technique for electromagnetic propagation modeling, which incorporates procedures such as ray batching and double count removal to improve the accuracy of the solution. The parallel optimization of the SBR code is achieved by fully exploiting the inherent features of NVIDIA OptiX Prime API and CUDA GPU programing interfaces suitable for SBR computations, comprehensive parallelization of all components of the SBR algorithm, including electric field calculation as well as postprocessing tasks being traditionally limited to sequential operation, and addressing and

optimizing memory usage and constraints to further advance efficiency of the method. The paper has presented and explained in detail a novel unified framework of algorithms, strategies, and data structures applied to SBR RT to further acceleration on the GPU without sacrificing the previously advanced and established accuracy of the method. Importantly, the major enhancements of the accuracy and the efficiency are tightly coupled and had to be addressed and pursued synergistically.

The presented numerical results have demonstrated that the new proposed optimized SBR methodology achieves massive parallel vs. serial speedups and upwards of 99% parallelism under Amdahl's parallelization scaling law, which is a rather unique performance outcome. The strategic use of GPUs and the meticulous optimization of all computational aspects of the code have resulted in a solver that is extremely efficient in launching millions of rays, computing, storing, and updating the electric field for each ray, and converging to accurate solutions for electrically large structures smoothly and rapidly. The efficiency gained from the acceleration, achieved from the very high level of parallelism, has created an SBR algorithm with a favorable performance and dramatic reduction of computation time for convergence in long tunnels, as a challenging test case, as well as in an urban outdoor wireless propagation environment. We believe that the algorithms, strategies, and data structures proposed and described in this work should be of significant interest and value to researchers, developers, and practitioners in the field of ray-based and high-frequency asymptotic methods and modeling, as well as for all related considerations and advancements of parallelism and efficiency of CEM techniques in general

Chapter 4
Kriging Methodology for Uncertainty Quantification in Computational Electromagnetics[3]

4.1 Introduction

Among the most important topics in computational science and engineering, and particularly so in computational electromagnetics (CEM), uncertainty quantification (UQ) is at the forefront of research due to the far-ranging academic and industrial uses [13]. Through effective and rigorous UQ, the quality of analyses and designs may be improved drastically, both in terms of effectiveness and reliability. Indeed, from the position of satellites to the manufacture of the simplest resistor or tomorrow's weather, uncertainty is unavoidable. Further, uncertainty is essential in CEM as material parameters of an antenna/scatterer, object shape/size, mutual positions/ orientations, etc. are all uncertain input parameters, which may be known only within a specific tolerance. UQ in CEM involves studies of how the input parameter uncertainty translates into uncertainty in the generated electric or magnetic field, for example, where the UQ-CEM aims at assessing the sensitivity of the field to the uncertain input parameter. Troublesome sources of uncertainty also include experimental uncertainty, inadequacy and randomness of a model or some of its components, discretization error, and other sources of numerical error, e.g., numerical integration and finite-precision arithmetic.

In CEM, the finite element method (FEM) and the method of moments (MoM), along with similar alternatives such as finite difference methods, provide the underlying model for solving

[3] This chapter is based on work from Ref. [15]

the relevant partial differential equation or integral equation problems and therefore the structure by which the uncertainty influences or perturbs the solution and the quantities of interest (QoIs) computed from the solution. Namely, in practical applications, we are often interested in specific QoIs (i.e., functionals of the solution). With the solution itself being a random quantity, by tailoring results to the set of necessary computables, the parameter space may be explored more effectively, driving higher quality UQ analysis for the goal-oriented simulation objectives and the same or lower computational cost. For a waveguide structure, a typical QoI that we are interested in studying could be the cutoff frequency or the propagation coefficient; for a scattering target, the QoI could be the radar cross section (RCS) or a specific component of the scattered field. In other words, according to our goal-oriented approach, rather than the solution directly, our primary objective is the study of a goal functional (or a collection of goal functionals) [14], [35]. The general UQ problem we are considering in this work can be formulated as predicting and quantifying the probability density function of a QoI in a CEM simulation given the probability density of uncertain (stochastic) input parameters to the CEM structure and model. Specifically, the probability density of the uncertain input parameter (such as material parameter) is propagated through the system, in this case being FEM, so that an output probability density can be constructed. This output probability density represents uncertainty of some calculated or measured quantity given uncertainty in the input. Specifically, in this work, we will consider uncertainty in a scatterer material parameter and propagate this uncertainty into the simulated monostatic radar cross section.

Traditional Monte Carlo methods – the conventional and dominant approach to UQ – are straightforward, nonintrusive (the deterministic forward solver can be used as a "black box" and only postprocessing needs to be done on simulation results), and can easily be applied to higher-

dimensional parameter spaces [36]–[38]. However, the number of samples required restricts applicability to low-cost computations or when high-precision UQ is not needed, as the convergence of Monte Carlo simulation is proportional to the square root of the number of samples. For example, in [39]–[41], standard Monte Carlo methods were applied to rough surface scattering. Given the precision required of modern devices and objectives, simulations require similar resolution, demanding more expensive computational models. Hence, when model complexity and precision require a large number of unknowns (degrees of freedom), each Monte Carlo solve can take large (often prohibitive) computation time. Thousands (or many more) of points are usually required for Monte Carlo convergence, which makes this simulation practically unattainable in many cases.

Given the potential unsuitability of standard Monte Carlo methods, then, a significant body of research has been devoted to stochastic spectral Galerkin approaches, e.g., polynomial chaos expansion (PCE) [42]–[49], and stochastic collocation approaches [50]–[56]. Some of these methods are intrusive techniques, requiring substantial modifications to the deterministic CEM solver and thus being undesirable due to the implementation complexity required. Some methods suffer the curse of dimensionality, i.e., the rate of convergence deteriorates with the increase of dimension of the parameter space), an important issue in UQ that is not addressed in this work, nor we discuss ways of its mitigation.

Additionally, with the increasing involvement of machine learning and optimization techniques in engineering projects, robust and automated uncertainty quantification is of vital importance. Recently, given the necessity to consider uncertainty as outlined above, researchers and practitioners in design optimization have begun to consider uncertainty as an integral component of the applied procedures. For instance, as outlined in [57]–[60], UQ-augmented machine learning

techniques for optimizing designs in electromagnetics have resulted in substantial improvements for real-world applications. While promising, in most cases, these techniques rely on standard Monte Carlo methods, therefore suffering the same weaknesses hampering other areas of engineering and numerical methods.

In recent years, sensitivity analysis and optimization methods have been increasingly based on surrogate models primarily due to the reduced computational cost when compared with Monte Carlo methods. Surrogate models replace expensive function evaluations with an analytical model which is constructed using samples of the physical (original) simulation (on the actual model) [61], [62], [38]. If at some point the surrogate evaluations become sufficiently accurate when compared to the actual (physical) model, the surrogate can now be used instead of the prohibitively expensive function evaluations. Several surrogate models, such as the least square polynomial, multi-layer perceptron, radial basis function, and Kriging models and methods were examined in [63], and Kriging and radial basis function methods showed the most promise.

This paper presents the implementation and use of the Kriging methodology, i.e., surrogate models based on Kriging interpolation, also referred to as Gaussian process regression, in UQ in CEM. Kriging methods have been applied in optimization and UQ problems in other fields of science and engineering, such as aeronautics, aerospace, geostatistics, fluid dynamics, etc. [38], [61]–[67] as well as in optimization of designs in magnetics [68] and CEM [69]–[73]. In particular, [69] proposes a combination of the universal Kriging with DIRECT algorithm to increase the accuracy of the surrogate reconstruction by selecting optimum points and using the surrogate model to optimize the size of a slot-array frequency selective surface. In [70], the combination of domain confinement with gradient enhanced ordinary nested Kriging is proposed to reduce the size of the domain of interest in the surrogate reconstruction within the optimization of the

parameters of an antenna. Further applications of Kriging interpolation to antenna optimization are presented in [71]–[73]. In addition, notable examples of UQ of electronic circuits using Kriging surrogate models and SPICE circuit solvers are given in [74], [75].

We present consistent and unified description, derivation, implementation, and use of several advanced Kriging approaches, namely, the universal Kriging, Taylor Kriging, and gradient-enhanced Kriging methods, for reconstruction of probability density function in CEM problems. We provide in-depth and comprehensive description of Kriging and the necessary methods to calculate covariance, adjoints, correlation, etc. Here, the universal Kriging (UK) is a surrogate method that is advanced from the ordinary Kriging (OK) by the use of basis functions in the form of (typically) power functions in the construction of the Kriging predictor. The Taylor Kriging (TK) method is a further advancement by means of Taylor series polynomial basis functions centered at the average of sample points. The gradient-enhanced Kriging (GEK) formulation uses gradient values to advance the accuracy of the ordinary Kriging approach. Note that some preliminary results using one of these techniques (TK) are presented in a summary form in [76].

Moreover, this paper proposes a completely new Kriging methodology, the gradient-enhanced Taylor Kriging (GETK) technique. Namely, gradient enhancements as shown in [64], [61], [77], [62] have all been proposed and built on the foundation of the ordinary Kriging method which has no basis functions involved with the regression scheme. In addition, GEK has been implemented in [61] and [64] specifically in the context of UQ and optimization, and was shown to be significantly more accurate than even advanced versions of Kriging such as UK. In this work, we propose, derive (using optimization principles), develop, test, and demonstrate a GEK methodology based on the TK approach which has been shown to generally be the most accurate form of non-gradient-enhanced Kriging methods. The novel GETK methodology has the

advantage of making use of gradient information, such as with the GEK method, and basis functions, much like in the UK and TK formulations (and more similarly to TK as we use Taylor series bases). The proposed GETK methodology is novel not only to UQ in CEM but to science and engineering in general, and not only to UQ and general numerical analysis but to all possible uses of Kriging interpolations, of which there are many indeed.

Furthermore, for gradient enhanced surrogate functions in this work, i.e., the GEK and GETK models, the gradients are calculated using our adjoint CEM methodology [14], [35], [78]–[81], that is, the gradients are obtained from the suitable adjoint problem that relates the deterministic forward problem to a specific QoI. The benefit of using adjoints to solve for gradients is that the solution requires exactly (only) one additional simulation solve regardless of the dimensionality of the parameter space. Note that adjoint solutions have become increasingly popular in computational methods in general in recent years due to their excellent scaling. Computational thrift is achieved through a goal-oriented focus, with the method built around user-specified quantities of interest.

The paper presents validation and comparative numerical studies of different Kriging methodologies for UQ CEM analysis. We compare and discuss OK, UK, TK, GEK, and GETK solutions to UQ scattering problems and examine the accuracy and convergence of the solution and performance of various approaches when reconstructing an output probability density function given an input probability density from some uncertain quantity, with emphasis on predicting material uncertainty impact on finite element method (FEM) RCS computations. We also validate, evaluate, and discuss the accuracy of various Kriging methods against two well-established non-Kriging UQ approaches. The first is the general adjoint-based gradient-enhanced higher-order parameter sampling (HOPS) UQ methodology [14], [35]. The second is a PCE approach, the

arbitrary polynomial chaos (aPC) technique, which is nonintrusive and has been shown to work rather well for UQ elsewhere in applied mathematics [82]. Numerical results show that Kriging methods provide excellent and very useful tools to construct surrogate functions in UQ CEM simulations, with very small numbers of input sample points, as a very efficient alternative to Monte Carlo computations. The results also demonstrate that the novel GETK method outperforms all other approaches, Kriging and non-Kriging, by much faster convergence with increasing the number of sample points and dramatically more accurate probability density reconstruction in all examples.

4.2 General Principles

4.2.1 The Kriging System

To introduce and explain the principles that all Kriging interpolation methods share and establish the Kriging system of equations in general terms, consider a stochastic function of the following form, which may represent some complex-valued scattered field based on some input $\boldsymbol{X}$ [83], [65], [68],

$$Z(\boldsymbol{X}) = \mu(\boldsymbol{X}) + \varepsilon(\boldsymbol{X}), \tag{4.1}$$

where $\boldsymbol{X}$ is referred to as a position vector in the parameter space, $\mu(\boldsymbol{X})$ is a mean or drift function, and $\varepsilon(\boldsymbol{X})$ is a random error term with the expected value $E[\varepsilon(\boldsymbol{X})] = 0$. With N observed, or measured, values, $Z(\boldsymbol{X_1}), Z(\boldsymbol{X_2}), \dots, Z(\boldsymbol{X_N})$, the Kriging predictor, $\hat{Z}(\boldsymbol{X})$, is given as a linear sum of these values multiplied by unknown weights (λ), which can in general be complex,

$$\hat{Z}(\boldsymbol{X}) = \sum_{i=1}^{N} \lambda_i \cdot Z(\boldsymbol{X_i}), \tag{4.2}$$

where, generally, the hatted, $\widehat{(.)}$, symbology is used to indicate an estimated or predicted quantity value instead of the true value. The weights λ_i are selected based on two conditions combined [83]. Firstly, $\hat{Z}(\boldsymbol{X})$ is the unbiased estimator of $Z(\boldsymbol{X})$, that is, $E[Z(\boldsymbol{X})] = E[\hat{Z}(\boldsymbol{X})]$. Secondly, the variance of the estimator error, $Var\left[\hat{Z}(\boldsymbol{X}) - Z(\boldsymbol{X})\right] = Var[\epsilon(\boldsymbol{X})]$, is minimized, which can be expanded into the following expression:

$$\begin{aligned} \text{Minimize: } \; Var[\epsilon(\boldsymbol{X})] = &\sum_{i=1}^{N} \lambda_i \sum_{j=1}^{N} \lambda_j \, Cov\left[Z(\boldsymbol{X}_i),\, Z\left(\boldsymbol{X}_j\right)\right] \\ &-2\sum_{i=1}^{N} \lambda_i \, Cov[Z(\boldsymbol{X}_i),\, Z(\boldsymbol{X})] + Var[Z(\boldsymbol{X})] \end{aligned} \tag{4.3}$$

where $Cov\left[Z(\boldsymbol{X}_i),\, Z\left(\boldsymbol{X}_j\right)\right]$ is the covariance between samples $Z(\boldsymbol{X}_i)$ and $Z\left(\boldsymbol{X}_j\right)$, at input sample points $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ (also referred to as training points), the total number of which is *N*, and $\hat{Z}(\boldsymbol{X})$ is the surrogate output reconstruction point. From the first requirement, we take the expected value of (4.2) and use the assumption that $E[Z(\boldsymbol{X})] = \mu(\boldsymbol{X})$, i.e., a stationary process, to come up with the Kriging system constraint

$$\mu(\boldsymbol{X}) = \sum_{i=1}^{N} \lambda_i \cdot \mu(\boldsymbol{X}_i). \tag{4.4}$$

In the case of ordinary Kriging, $\mu(\boldsymbol{X})$ is assumed to be constant, yielding the OK system constraint as follows:

$$\sum_{i=1}^{N} \lambda_i = 1. \tag{4.5}$$

On the other hand, in advanced Kriging approaches, this assumption is abandoned; namely, in the universal Kriging, Taylor Kriging, and gradient-enhanced Taylor Kriging methods, rather than a constant, we instead assume that $\mu(\boldsymbol{X})$ can be expressed as a sum of basis functions up to order M,

$$\mu(\boldsymbol{X}) = \widehat{\boldsymbol{a}}^{\boldsymbol{T}}\boldsymbol{b}(\boldsymbol{X}) = \sum_{l=0}^{M} \widehat{\boldsymbol{a}}_l b_l(\boldsymbol{X}), \tag{4.6}$$

where $b_l(\boldsymbol{X})$ are basis functions and $\widehat{\boldsymbol{a}}_l$ are unknown coefficients. Thus (4.4) can be expressed as follows:

$$b_l(\boldsymbol{X}) = \sum_{i=1}^{N} \boldsymbol{\lambda}_i b_l(\boldsymbol{X_i}), \;\; l = 0,1, \dots, M. \tag{4.7}$$

4.2.2 Empirical Covariance Function

Having specified the Kriging mean, we now outline the covariance function. We consider two modalities: one based on an empirical covariance function and the other on maximum likelihood estimation (MLE) [64], [68], [84], [85], [61]. The covariance used in Kriging is typically assumed to be homogenous and isotropic [83], meaning that two points in the parameter space have the

same covariance if the distance between them is equal and covariance is independent of direction. This allows for the construction of a covariance function $C(h)$ that can be evaluated for any pair of $Z(\boldsymbol{X}_i)$ and $Z(\boldsymbol{X}_j)$, where $h = |\boldsymbol{X}_i - \boldsymbol{X}_j|$ is the Euclidean norm. Note that anisotropic covariance functions, where the function is no longer independent of direction within the input parameter space, can also be included. While this can improve performance in specific scenarios where data correlation trends appear in specific directions, isotropic covariance functions, especially the squared exponential kernel, are typically chosen due to generally good performance in vast majority of scenarios as well as flexibility and ease of use.

In the empirical approach, the distances are separated into discrete bins with every pair of samples used for the Kriging system being placed into one of these bins with $N(h)$ representing the number of sample points that fall within a given bin, and an empirical covariance function is evaluated as [84], [85]

$$C(h) = \frac{1}{|N(h)|} \sum_{(i,j) \in N(h)} Z(\boldsymbol{X}_i) \cdot Z(\boldsymbol{X}_{\boldsymbol{i+h}}) - \, m_i \cdot m_{i+h} \,. \tag{4.8}$$

Here, m_i is the mean of function values at distance h given by

$$m_i = \frac{1}{|N(h)|} \sum_{i \in N(h)} Z(\boldsymbol{X}_i) \,. \tag{4.9}$$

Once the empirical covariance function is created, it is fitted according to one of several models, including Gaussian, exponential, cubic, and spherical among others. The commonly used Gaussian fit function is given by

$$C(h) = C_0 e^{-\left(\frac{h}{d}\right)^2}, \tag{4.10}$$

where C_0 and d are hyperparameters to be fitted.

4.2.3 Maximum Likelihood Estimation

Within the MLE approach, rather than constructing an empirical function, we instead aim to maximize a likelihood function given the observed data, namely, to make the likelihood function the most likely realization of the data as in a Bayesian approach. The likelihood is given by

$$L(\theta) = \prod_{d=1}^{D} f(X_d|\theta_d), \tag{4.11}$$

where $f(\cdot)$ is the probability distribution of the data, $\boldsymbol{X}$ is the input vector, whose dimension is D, and $\boldsymbol{\theta}$ is the vector of hyperparameters to be optimized via MLE. However, the likelihood function is difficult to optimize directly, and instead we use the log-likelihood function, i.e., the logarithm of the likelihood function,

$$LL(\theta) = log\, L(\theta) = log \prod_{k=1}^{D} f(X_k|\theta_d)$$
$$= \sum_{k=1}^{D} \log f(X_k|\theta_d)\,. \qquad (4.12)$$

Given that the logarithm function is monotonic (always increasing), the hyperparameter that maximizes the likelihood function will also maximize the log-likelihood function, and the computational benefit from invoking the log-likelihood becomes evident at higher dimensions, where we turn the multiplication into a summation that can be independently optimized. Additionally, many probability distributions involve exponential functions, which are cancelled by the logarithm function.

The log-likelihood function considered in this work represents a Gaussian prior distribution likelihood given by

$$LL(\theta) = -\sum_{k=1}^{D}[\log(2\pi\hat{\sigma}^2)^{\frac{N}{2}}\sqrt{|\boldsymbol{R}|} - \frac{(\boldsymbol{z}-\boldsymbol{B}\hat{\boldsymbol{a}})^T\boldsymbol{R}^{-1}(\boldsymbol{z}-\boldsymbol{B}\hat{\boldsymbol{a}})}{2\hat{\sigma}^2}], \qquad (4.13)$$

where $\boldsymbol{z}$ is a vector of observed data, $\boldsymbol{B}$ is a matrix of basis functions, $\hat{\sigma}^2$ is the variance of the data, and $\hat{\boldsymbol{a}}$ are the regression coefficients. $\boldsymbol{R}$ is a correlation function adopted as

$$R(\boldsymbol{X}_i, \boldsymbol{X}_j, \theta) = e^{-\theta(X_i - X_j)^2}, \qquad (4.14)$$

and the covariance function is obtained as

$$Cov\left[Z(\boldsymbol{X}_i),\, Z\left(\boldsymbol{X}_j\right)\right] = \sigma^2 R\left(\boldsymbol{X}_i, \boldsymbol{X}_j,\, \theta\right). \tag{4.15}$$

While there are many possible choices for the correlation function, the Kriging systems solutions presented in this paper will all use the Gaussian (squared exponential) correlation function given in (4.14).

Maximizing (13) represents an MLE set of *D* nonlinear optimization problems, with *D* standing for the dimension of the parameter space in (4.11). To solve the optimization problems, we use derivatives of (4.13) with respect to $\widehat{\boldsymbol{a}}$ and $\hat{\sigma}^2$, respectively, leading to

$$\widehat{\boldsymbol{a}} = (\boldsymbol{B}^T \boldsymbol{R}^{-1} \boldsymbol{B})^{-1} \boldsymbol{B}^T \boldsymbol{R}^{-1} \boldsymbol{z} \tag{4.16}$$

$$\hat{\sigma}^2 = \frac{(\boldsymbol{z} - \boldsymbol{B}\widehat{\boldsymbol{a}})^T R^{-1} (\boldsymbol{z} - \boldsymbol{B}\widehat{\boldsymbol{a}})}{n}. \tag{4.17}$$

Substituting these expressions back into (4.13), we perform the optimization as follows:

$$Maximize: \quad -\frac{[N\, log\, \hat{\sigma}^2 + log|\boldsymbol{R}|]}{2}. \tag{4.18}$$

This regression scheme is calculated using generalized least squares (GLS), since GLS also requires a covariance matrix for regression solution.

4.2.4 Adjoint Solution

For gradient enhanced Kriging models in this work, the gradients are obtained from the suitable adjoint problem [14], [35], [78]–[81], where the solution does not suffer from "the curse of dimensionality," meaning it requires one additional simulation solve for any number of uncertain input parameters to solve for all gradients. Here we present the solution to the adjoint problem in more general terms, so that the result can be used for alternate choices of QoI if desired. Then we narrow this adjoint solution to extract the gradient specifically for material parameters, which is directly applicable to the material parameter uncertainty as a specific focus of this work.

For clarity with the numerical results section (though other partial differential equations may be treated similarly), we start with the double-curl wave equation to solve for the scattered electric field ($\boldsymbol{E}^{sc}$):

$$\nabla \times \mu_{\mathrm{r}}^{-1} \nabla \times \mathbf{E}^{sc} - k_0^2 \varepsilon_r \boldsymbol{E}^{sc} = -\nabla \times \mu_{\mathrm{r}}^{-1} \nabla \times \mathbf{E}^{\mathrm{inc}} + k_0^2 \varepsilon_r \boldsymbol{E}^{inc}, \tag{4.19}$$

where ε_r and μ_r are the relative permittivity and permeability, respectively, of the medium and k_0 is the free space wave number, which we recast using a linear operator, $\mathcal{L}$, and the corresponding excitation function $\boldsymbol{G}$,

$$\mathcal{L}\boldsymbol{E}^{sc} = \boldsymbol{G}. \tag{4.20}$$

The adjoint operator, $\mathcal{L}^*$, is defined using the Lagrange identity [86],

$$\langle \mathcal{L}\boldsymbol{u}, \boldsymbol{v} \rangle = \langle \boldsymbol{u}, \mathcal{L}^* \boldsymbol{v} \rangle, \tag{4.21}$$

where $\langle \cdot, \cdot \rangle$ denotes the standard L^2 inner-product.

We consider a linear functional J of the forward solution for $\boldsymbol{E}^{sc}$ as the QoI [14],

$$J[\boldsymbol{E}^{sc}] = \langle \boldsymbol{E^{sc}}, \boldsymbol{p} \rangle, \tag{4.22}$$

where $\boldsymbol{p}$ is the excitation to the adjoint problem as defined by $\mathcal{L}^* \boldsymbol{v} = \boldsymbol{p}$. The choice of $\boldsymbol{p}$ can be modified for different choices of QoI, and $\boldsymbol{v}$ is the respective solution to the adjoint problem. Following [14], if we then perturb some component of the forward problem, a perturbed forward problem is expressed as

$$\tilde{\mathcal{L}} \widetilde{\boldsymbol{E}}^{sc} = \widetilde{\boldsymbol{G}}, \tag{4.23}$$

where $\widetilde{\boldsymbol{E}}^{sc}$ is the perturbed solution with respect to the deterministic reference forward solution, $\boldsymbol{E}^{sc} = \boldsymbol{E}^{ref}$,

$$\widetilde{\boldsymbol{E}}^{sc} = \boldsymbol{E}^{ref} + \delta \boldsymbol{E}^{ref}, \tag{4.24}$$

with δ standing for a linear operator that signifies the effects of the perturbation, $\tilde{\mathcal{L}} = \mathcal{L} + \delta \mathcal{L}$, and similarly for $\widetilde{\boldsymbol{G}}$. Taking the inner product of (4.23) with the adjoint solution $\boldsymbol{v}$ and applying the Lagrange identity,

$$\left\langle \widetilde{\boldsymbol{E}}^{sc}, \boldsymbol{p} \right\rangle = \langle \boldsymbol{E}^{sc}, \boldsymbol{p} \rangle + \left\langle \delta \boldsymbol{G} - \delta \mathcal{L} \widetilde{\boldsymbol{E}}^{sc}, \boldsymbol{v} \right\rangle. \tag{4.25}$$

In a practical case of relatively small perturbations, we apply the definition of the QoI in (4.24) to obtain

$$J\left[\widetilde{\boldsymbol{E}}^{sc}\right] \approx J[\boldsymbol{E}^{ref}] + \left\langle \delta \boldsymbol{G} - \delta \mathcal{L} \boldsymbol{E}^{ref}, \boldsymbol{v} \right\rangle. \tag{4.26}$$

We focus now on the perturbation on the material parameters, say on the reference relative permittivity of a scatterer, $\varepsilon_r = \varepsilon_r^{ref}$, and a perturbation of $\Delta\varepsilon_r = \widetilde{\varepsilon_r} - \varepsilon_r^{ref}$. The effects of the perturbation $\delta \mathcal{L} \boldsymbol{E}^{ref}$ and $\delta \boldsymbol{G}$ are calculated by taking a derivative of the left- and right-hand side of (4.19), respectively, with respect to the perturbed parameter and multiplying it by the perturbation. Substituting these effects into (4.26), we arrive at

$$J\left[\widetilde{\boldsymbol{E}}^{sc}\right] \approx J[\boldsymbol{E}^{ref}] + k_0^2 \langle (\boldsymbol{E}^{\boldsymbol{inc}} - \boldsymbol{E}^{ref})\, \Delta\varepsilon_r, \boldsymbol{v} \rangle. \tag{4.27}$$

In the case of multiple quantities being perturbed, the perturbation becomes a vector, and the terms $\delta \mathcal{L} \boldsymbol{E}^{ref}$ and $\delta \boldsymbol{G}$ acquire matrix form. If we represent (4.27) in the form of a first order Taylor series expansion,

$$J\left[\widetilde{\boldsymbol{E}}^{sc}\right] \approx J[\boldsymbol{E}^{ref}] + \Delta\varepsilon_r \nabla J[\boldsymbol{E}^{ref}], \tag{4.28}$$

we can extract the gradient term, namely, the following expression for the gradient of the QoI:

$$\nabla J[\boldsymbol{E}^{ref}] = k_0^2 \langle \boldsymbol{E}^{inc} - \boldsymbol{E}^{ref}, \boldsymbol{v} \rangle, \tag{4.29}$$

to be used in our GEK and GETK methods in this work. Importantly, we can implement this result into computation of any gradient enhanced surrogate function, with a requirement of only one additional simulation solution solve.

4.3 Kriging System Solutions

4.3.1 Universal Kriging

The variance of the estimator in the universal Kriging (UK) method is given by (4.3), subject to (4.7), where basis functions ($\boldsymbol{b}$) are introduced as an advancement over the ordinary Kriging (OK). This can easily be transformed into an unconstrained optimization problem using Lagrange multipliers (δ_l), leading to [83], [65], [68]:

Minimize:

$$\sum_{i=1}^{N} \lambda_i \sum_{j=1}^{N} \lambda_j \, Cov\left[Z(\boldsymbol{X}_i),\, Z(\boldsymbol{X}_j)\right] - 2\sum_{i=1}^{N} \lambda_i \, Cov[Z(\boldsymbol{X}_i),\, Z(\boldsymbol{X})]$$

$$+Var(Z(\boldsymbol{X}) + 2\sum_{l=1}^{M} \delta_l \left[\sum_{i=1}^{N} \lambda_i b_l(\boldsymbol{X}_i - b_l(\boldsymbol{X})\right]$$

$$+Var\left(Z(\boldsymbol{X}) + 2\sum_{l=1}^{M} \delta_l \left[\sum_{i=1}^{N} \lambda_i b_l(\boldsymbol{X}_i - b_l(\boldsymbol{X})\right]\right) \tag{4.30}$$

Taking partial derivatives with respect to λ_i and δ_l, we come up with the following system of matrix equations that can be solved for λ values:

$$\begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\delta} \end{bmatrix} = \begin{bmatrix} \boldsymbol{C} & \boldsymbol{B} \\ \boldsymbol{B}^T & \boldsymbol{0} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{c}_{x_i x} \\ \boldsymbol{b}_x \end{bmatrix}, \tag{4.31}$$

$$\boldsymbol{C} = \begin{bmatrix} C_{x_1x_1} & \cdots & C_{x_1x_N} \\ \vdots & \ddots & \vdots \\ C_{x_Nx_1} & \cdots & C_{x_Nx_N} \end{bmatrix}, \tag{4.32}$$

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{b_1}(\boldsymbol{x_1}) & \cdots & \boldsymbol{b}_M(\boldsymbol{x_1}) \\ \vdots & \ddots & \vdots \\ \boldsymbol{b}_1(\boldsymbol{x_N}) & \cdots & \boldsymbol{b}_M(\boldsymbol{x_N}) \end{bmatrix}. \tag{4.33}$$

$\boldsymbol{C}$ is a covariance matrix, where $C_{x_ix_j}$ is the covariance between $Z(\boldsymbol{X}_i)$ and $Z(\boldsymbol{X}_j)$ and $\boldsymbol{c}_{x_ix}$ is a vector of covariances evaluated between $Z(\boldsymbol{X_i})$ and $Z(\boldsymbol{X})$, $\boldsymbol{B}$ is a basis function matrix, and $\boldsymbol{b}_x$ represents a vector of basis functions evaluated at $\boldsymbol{X}$.

In the UK technique, the basis functions are given by

$$b_l(\boldsymbol{X}) = \boldsymbol{X}^l, \qquad l = 0, 1, \dots, M, \tag{4.34}$$

where the number of basis functions, M, is limited by the following expression involving the dimension of the parameter space, D, and the number of input sample (training) points, N [68], [87]:

$$\frac{(D+M)!}{M!\,D!} \leq N. \tag{4.35}$$

With all the details above, the UK system of equations in (4.31) is solved, and the computed weights λ_i are used to construct the Kriging predictor $\hat{Z}(\boldsymbol{X})$ in (4.2).

4.3.2 Taylor Kriging

The Taylor Kriging (TK) methodology is an advancement of the UK technique by means of special Taylor series polynomial basis functions, instead of the power functions in (4.34), that are

adjusted to be more sensitive to sample points, namely, they are centered at the average of sample points. The M^{th} order Taylor series expansion of the mean about $\boldsymbol{X}_0$, where $\boldsymbol{X}_0$ is the average of all sample inputs, is given by [83]

$$\mu(\boldsymbol{X}) = \widehat{\boldsymbol{a}}^{\boldsymbol{T}}\boldsymbol{b}(\boldsymbol{X}) = \mu(\boldsymbol{X_0}) + \mu^{(1)}(\boldsymbol{X_0})(\boldsymbol{X} - \boldsymbol{X_0}) + \cdots + \frac{\mu^{(\boldsymbol{M})}(\boldsymbol{X_0})(\boldsymbol{X} - \boldsymbol{X_0})^{\boldsymbol{M}}}{K!}. \quad (4.36)$$

With reference to (4.6), the choices for $b_l(\boldsymbol{X})$ and $\boldsymbol{a}$ in (4.36) are

$$b_l(\boldsymbol{X}) = (\boldsymbol{X} - \boldsymbol{X_0})^{\boldsymbol{l}}, \quad a_l = \mu(\boldsymbol{X_0})^{(l)}, \quad l = 0,1, \dots, M. \quad (4.37)$$

4.3.3 Gradient Enhanced Kriging

As another advancement of the OK method, the gradient-enhanced Kriging (GEK) formulation [64], [62], [77], [61], which uses gradient values to enhance the accuracy of the method, results in the following Kriging matrix equation [62]:

$$\begin{bmatrix} \boldsymbol{\lambda} \\ \widetilde{\boldsymbol{\delta}} \end{bmatrix} = \begin{bmatrix} \dot{\boldsymbol{C}} & \boldsymbol{F} \\ \boldsymbol{F}^T & \boldsymbol{0} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{c}_{x_i x} \\ \boldsymbol{1} \end{bmatrix} \quad (4.38)$$

Here, $\dot{\boldsymbol{C}}$ is a composite matrix of the covariance matrix in (4.32) and its derivatives, of size $N(D+1) \times N(D+1)$, that contains the covariances between function values and gradients, $\boldsymbol{c}_{x_i x}$ is the same covariance vector as in (4.31), $\boldsymbol{F} = [1, \dots 1, 0, \dots 0]$ (a vector with N unity values and the rest being zeros), and $\widetilde{\boldsymbol{\delta}}$ is the associated Lagrange multiplier [62]. Note that $\boldsymbol{F}$ in the GEK method can be thought of as a rudimentary version of the basis functions in UK or TK methods with the order $M = 0$.

4.3.4 Gradient Enhanced Taylor Kriging

The use of adjoints to find gradient information has recently become more explored and exploited in computational science and engineering, so the utilization of gradients in surrogate

functions is increasingly of interest. However, the gradient-enhanced Kriging method has been developed typically based on the ordinary Kriging technique. The goal here is to develop a gradient enhanced Kriging methodology based on the TK approach instead. The novel gradient-enhanced Taylor Kriging (GETK) method would then benefit from both the gradient information and the basis functions in the regression scheme. Moreover, in our approach, the gradients are obtained from the adjoint solve to the deterministic forward problem for a specific QoI.

Expanding the general Kriging predictor form in (4.2), we introduce a GETK predictor given by

$$\hat{Z}(\boldsymbol{X}) = \sum_{i=1}^{N} [\lambda_i \, Z(\boldsymbol{X}_i) \; + \; \kappa_i \frac{\partial Z(\boldsymbol{X}_i)}{\partial \boldsymbol{X}_i}] \,, \tag{4.39}$$

thus introducing the function gradient multiplied by a second weight vector, κ. Satisfying the unbiased estimator condition, $E[Z(\boldsymbol{X})] = E[\hat{Z}(\boldsymbol{X})]$, (4.7) is now expanded to

$$b_l(\boldsymbol{X}) = \sum_{i=1}^{N} [\lambda_i b_l(\boldsymbol{X}_i) + \kappa_i \frac{\partial b_l(X_i)}{\partial \boldsymbol{X}_i}], \quad l = 0,1, \dots, M. \tag{4.40}$$

Deriving $Var\big[\hat{Z}(\boldsymbol{X}) - Z(\boldsymbol{X})\big]$, (4.3) now acquires a substantially more complex form,

$$\text{Minimize: } \; Var[\epsilon(\boldsymbol{X})] = \sum_{i=1}^{N} \lambda_i \sum_{j=1}^{N} \lambda_j \, Cov\big[Z(\boldsymbol{X}_i), \, Z\big(\boldsymbol{X}_j\big)\big]$$

$$+\sum_{i=1}^{N}\kappa_i\sum_{j=1}^{N}\lambda_j\,Cov\left[\frac{\partial Z(\boldsymbol{X}_i)}{\partial \boldsymbol{X}_i},\,Z\left(\boldsymbol{X}_j\right)\right]$$

$$+\sum_{i=1}^{N}\kappa_i\sum_{j=1}^{N}\kappa_j\,Cov\left[\frac{\partial Z(\boldsymbol{X}_i)}{\partial \boldsymbol{X}_i},\frac{\partial Z\left(\boldsymbol{X}_j\right)}{\partial \boldsymbol{X}_j}\right]+Var[Z(\boldsymbol{X})] \tag{4.41}$$

$$+\sum_{i=1}^{N}\lambda_i\sum_{j=1}^{N}\kappa_j Cov\left[Z(\boldsymbol{X}_i),\frac{\partial Z\left(\boldsymbol{X}_j\right)}{\partial \boldsymbol{X}_j}\right]-$$

$$2\sum_{i=1}^{N}\lambda_i\,Cov[Z(\boldsymbol{X}_i),\,Z(\boldsymbol{X})]-2\sum_{i=1}^{N}\kappa_i\,Cov\left[\frac{\partial Z(\boldsymbol{X}_i)}{\partial \boldsymbol{X}_i},\,Z(\boldsymbol{X})\right].$$

Using the same Lagrange multipliers ($\boldsymbol{\delta}$) approach and the fact that

$$Cov\left[\tfrac{\partial Z(\boldsymbol{X}_i)}{\partial \boldsymbol{X}_i},\,Z\left(\boldsymbol{X}_j\right)\right]=\tfrac{\partial}{\partial \boldsymbol{X}_i}Cov[Z(\boldsymbol{X}_i),Z\left(\boldsymbol{X}_j\right)]\,, \tag{4.42}$$

the system in (4.41) can be recast as

$$\begin{bmatrix}\boldsymbol{\lambda}\\ \boldsymbol{\kappa}\\ \boldsymbol{\delta}\end{bmatrix}=\begin{bmatrix}\boldsymbol{C} & \dfrac{\partial \boldsymbol{C}}{\partial \boldsymbol{X}_j} & \boldsymbol{B}\\ \dfrac{\partial \boldsymbol{C}^{\boldsymbol{T}}}{\partial \boldsymbol{X}_i} & \dfrac{\partial^2 \boldsymbol{C}}{\partial \boldsymbol{X}_i\partial \boldsymbol{X}_j} & \dfrac{\partial \boldsymbol{B}}{\partial \boldsymbol{X}_i}\\ \boldsymbol{B}^{\boldsymbol{T}} & \dfrac{\partial \boldsymbol{B}^{\boldsymbol{T}}}{\partial \boldsymbol{X}_i} & \boldsymbol{0}\end{bmatrix}^{-1}\begin{bmatrix}\boldsymbol{c}_{x_ix}\\ \dfrac{\partial c_{x_ix}}{\partial x_i}\\ \boldsymbol{b}_{\boldsymbol{x}}\end{bmatrix}. \tag{4.43}$$

The so obtained GETK matrix system can then be solved for weights $\boldsymbol{\lambda}$ and $\boldsymbol{\kappa}$, which constitute the solution for the GETK predictor $\hat{Z}(\boldsymbol{X})$ in (4.39).

The GETK system main matrix can be formally interpreted as the GEK one in (4.38) with $\dot{\boldsymbol{C}}$ and $\boldsymbol{F}$ now containing covariances and basis functions, respectively, as well as their derivatives,

$$\dot{\boldsymbol{C}} = \begin{bmatrix} \boldsymbol{C} & \dfrac{\partial \boldsymbol{C}}{\partial \boldsymbol{X}_i} \\ \dfrac{\partial \boldsymbol{C}}{\partial \boldsymbol{X}_j} & \dfrac{\partial^2 \boldsymbol{C}}{\partial \boldsymbol{X}_i \partial \boldsymbol{X}_j} \end{bmatrix}, \qquad \boldsymbol{F} = \begin{bmatrix} \boldsymbol{B} \\ \dfrac{\partial \boldsymbol{B}}{\partial \boldsymbol{X}_i} \end{bmatrix}. \tag{4.44}$$

Note, however, that the largest computational load in performing both GEK and GETK methods comes from inverting the composite covariance matrix, $\dot{\boldsymbol{C}}$, which is the same size for both methods. This means that there is very little extra computational cost for using GETK as compared to GEK.

The MLE formulation for covariance function, described in Section 4.2.3, has an additional benefit when used within the GEK and GETK methods, where the gradients are considered. Since we have an explicit form for the covariance function as opposed to a fitted function in the case of the empirical covariance, we can explicitly evaluate covariance function derivatives. The derivatives in (4.43), which are Jacobians in general dimension, are calculated using the form of covariance in (4.14) and (4.15) as follows:

$$\frac{\partial \boldsymbol{C}}{\partial \boldsymbol{X}_i} = \left[-2\theta^{(k)} \left(X_{\boldsymbol{i}}^{(\boldsymbol{k})} - X_{\boldsymbol{j}}^{(\boldsymbol{k})} \right) \sigma^2 R\left(\boldsymbol{X}_i, \boldsymbol{X}_j, \theta \right) \right]_{k=1,\dots,D} \tag{4.45}$$

$$\frac{\partial \boldsymbol{C}}{\partial \boldsymbol{X}_j} = \left[2\theta \left(X_{\boldsymbol{i}}^{(\boldsymbol{k})} - X_{\boldsymbol{j}}^{(\boldsymbol{k})} \right) \sigma^2 R\left(\boldsymbol{X}_i, \boldsymbol{X}_j, \theta \right) \right]_{k=1,\dots,D} \tag{4.46}$$

$$\frac{\partial^2 \boldsymbol{C}}{\partial \boldsymbol{X}_i \partial \boldsymbol{X}_j} =$$

$$\begin{cases} \left[\begin{pmatrix} 2\theta X_{\boldsymbol{i}}^{(\boldsymbol{k})^2} - 4\theta X_i^{(k)} X_j^{(k)} \\ +2\theta X^{(k)}{}_j^2 - 1 \\ \cdot -2\theta^{(k)}\sigma^2 R(\boldsymbol{X}_i,\, \boldsymbol{X}_j,\, \theta) \end{pmatrix}\right]_{k,l=1,\ldots,D} & \text{for } k = l \\ \left[\begin{matrix} -4\theta^{(k)}\theta^{(l)} \\ \cdot \left(X_i^{(k)} - X_j^{(k)}\right) \\ \cdot \left(X_i^{(l)} - X_j^{(l)}\right)\sigma^2 R(\boldsymbol{X}_i,\, \boldsymbol{X}_j,\, \theta) \end{matrix}\right]_{k,l=1,\ldots,D} & \text{otherwise} \end{cases}, \quad (4.47)$$

where the sizes of these Jacobian terms are $N \times ND$ for (4.45) and (4.46) and $ND \times ND$ for (4.47). On the other hand, when using the empirical covariance method, the derivatives must be computed numerically using finite differences or similar techniques. Additionally, matrix conditioning can be a concern in many regression and interpolation schemes, and to counter this issue in the Kriging methods, a small term is added to the diagonal of the composite covariance matrix, $\dot{\boldsymbol{C}}$, which helps with numerical stability of the output.

It is interesting to note that the Kriging model itself is inherently a stochastic model. So, while we propagate uncertainty from a stochastic input variable to the output, the model itself adds to the overall uncertainty, compared with the Monte Carlo solution which is deterministic. A full coverage of this can be found in [74], where we see that the first moment of uncertainty, i.e., the mean, is unaffected, whereas the variance of UQ is biased by the stochasticity of the model. Additionally, this additional uncertainty due to the model becomes very small as the number of samples increases, and the study of the exact uncertainty of the model and its effect on the overall uncertainty of the problem becomes intractable as input parameter dimension becomes large.

4.4 Results and Discussion

We now present numerical results obtained in comparison of five different Kriging approaches, namely, OK, UK, TK, GEK, and GETK methods, as well as two well-established non-Kriging UQ

methodologies, the HOPS adjoint sensitivity analysis method [14], [35] and the PCE method implemented as an aPC technique [82], [88], in UQ scattering problems. The performance of each of the presented methods and solutions is assessed in comparison with the 1000-point Monte Carlo (MC) simulation as a reference solution and a validation norm.

The scattering analysis is done using higher order curvilinear FEM modeling [89]–[91] to solve the double-curl wave equation (4.19) for the scattered field $\boldsymbol{E}^{sc}$ with an anisotropic locally-conformal perfectly matched layer (PML) for the computational domain termination [92]. The QoI considered here is the monostatic radar cross section (RCS) of a lossy dielectric sphere, with a goal of predicting material uncertainty impact on FEM-PML RCS computations.

In all of the Kriging results in this work, we use MLE to construct the covariance function (Section 4.2.3) as an alternative to the empirical covariance function (Section 4.2.2). In UK, TK, and GETK implementations and simulations, we adopt the number of basis functions, M, according to the maximum limit allowed by (4.35) given the dimension of the parameter space, D, and the adopted number of input sample points, N, in particular test cases and analyses. In addition, we separately limit the bases to a maximum of $M = 5$, to avoid ringing effects from high degree polynomial basis functions, also known as Runge phenomenon [93]. The OK and GEK implementations are inherently limited to 0^{th} order bases ($M = 0$).

In all cases tested here, we use a uniform sampling distribution of (N) input sample (training) points. While individual Kriging implementations frequently construct an initial discretization using Latin hypercube sampling (LHS) or similar method and then adaptively and optimally refine the distribution of points using a variety of techniques such as adaptive sampling, an even distribution of sample points where each method performs reconstruction using the same set of points was deemed to be the most appropriate for comparative numerical studies and evaluation of

different techniques in this work. Here, even sampling becomes intractable in higher dimensions since the parameter space volume grows exponentially with the dimension increase. Instead, LHS or similar sampling schemes are typically chosen for the initial data set. This also means that to sample the input space with the same density with an increase in dimension, N must increase exponentially as well. Additionally, the size of the Kriging system matrix, whose inversion represents the bulk of the computational load for the model, grows as $N(D+1) \times N(D+1)$, which can also become quite large as both N and D increase. This is known as the curse of dimensionality, mentioned earlier, a well-known phenomenon with many models involving sampling, including aPC and Kriging. The mitigation of these effects is a well-researched problem, which, however, is outside of the scope of this work.

We also note that the initial set of sampled training data can have a significant effect on the predicted value from the trained Kriging model. In some cases, where sampling does not adequately sample the parameter space, the initial predicted value can have poor performance. By either increasing the number of samples or changing the initial sample distribution, we can dramatically increase the accuracy of the initial Kriging prediction. Latin Hypercube Sampling (LHS) is a model that generates random test points that fully sample the input parameter space, and is thus typically used to sample parameter space in high dimension. In this work, we avoid using such sampling in order to keep an even distribution so that all comparisons of different methods are done on equal base.

In all cases, the input variable is assumed to be a random variable, and we sample 1000 points to form the input distribution. For the Monte Carlo reference simulation, we compute the solution at each of the 1000 sample points using higher order FEM modeling, which forms the deterministic solution; however, this can very quickly become intractable with computationally heavy

simulations. The same sampled input distribution is then used to predict the RCS and the probability density function using the trained Kriging model, where we evaluate the error with respect to the MC method as our metric.

4.4.1 Test Case with Smooth Output Function

As our first test case, we consider the scattered field of a spherical dielectric scatterer 2 m in diameter excited by a plane electromagnetic wave at 70 MHz. The scatterer material property constitutes the uncertain input parameter, with the real part of the relative permittivity, ε_r, described by a normal random variable (the dimension of the parameter space is $D = 1$). The complex relative permittivity is given by $\varepsilon_r = \{\sim N(6, 1.5), -2\}$, where the real part of ε_r is described as a normal random distribution with mean equal to 6 and variance 1.5 and the imaginary part is taken to be constant with a value of -2. We establish the validity and examine the performance of each of the presented methods in comparison with a reference 1000-point Monte Carlo simulation, treated as an "exact" solution.

In Fig. 4.1, we examine the convergence of the root mean squared error (RMSE) of the surrogate reconstruction of the probability density function versus the number of input sample points by means of the five presented Kriging methods, as well as HOPS and PCE methods, with respect to the reference Monte Carlo simulation. We observe relatively good convergence for many methods, coming from the fact that a well-behaved, smooth, output function is relatively easy to predict using different methods. However, even with this well-behaved result, we still see that the novel GETK method outperforms all other approaches by orders of magnitude. In addition, note that the HOPS method is used, along with the PCE method, as a non-Kriging comparison case for these results, but it should be emphasized that while this excellent UQ approach might require a larger number of sample points to achieve accurate results, with large input vectors, its true strength lies

in the ability to accurately and efficiently predict UQ problems with higher dimensions of parameter spaces [14], [35].

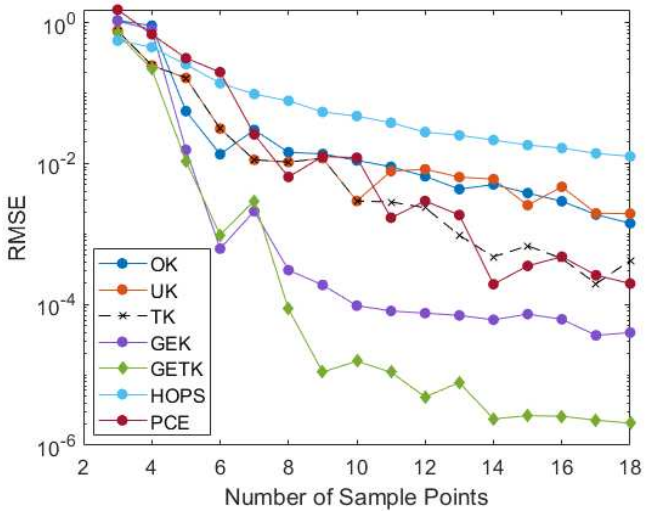


Fig. 4.1: Convergence of the root mean squared error of the RCS surrogate reconstruction of a dielectric spherical scatterer 2 m in diameter, with uncertain relative permittivity given by a normal random distribution $\varepsilon_r = \{\sim N(6, 1.5), -2\}$, excited by a plane wave at 70 MHz: comparison of results obtained by five different Kriging methods, HOPS, and PCE, with RMSE computed with respect to the reference 1000-point Monte Carlo simulation considered as an "exact" solution.

4.4.2 Test Case with Rapidly Varying Output Function

Our second test case deals with the RCS of a 2-m diameter spherical dielectric scatterer with $D = 1$ and $\varepsilon_r = \{\sim N(6, 1.5), -1\}$ at 115 MHz. The goal of this second comparative study is to examine the convergence of different Kriging methods with a rapidly varying output solution, as this represents a difficult case for any data reconstruction method. The RMSE probability density function reconstruction results in Fig. 4.2 demonstrate that the GETK method indeed converges much faster than all other techniques tested. We see that the GEK method performs quite well also, but still considerably underperforms the GETK results. Overall, for each of the presented methods, we observe much higher accuracy in Fig. 4.1, in the well-behaved output function case, than in Fig. 4.2 for the same number of sample points. Namely, in this case we see just how difficult it is to use a surrogate model when the output gradients vary significantly. Finally, a study like the one

in Fig. 4.2 indeed gives a quantitative measure of how much more accurate a UQ method (the GETK method here) is in reconstructing the uncertainty of CEM problems, that is, in outputting probability density prediction for a quantity of interest given the probability density of an uncertain (stochastic) input parameter.

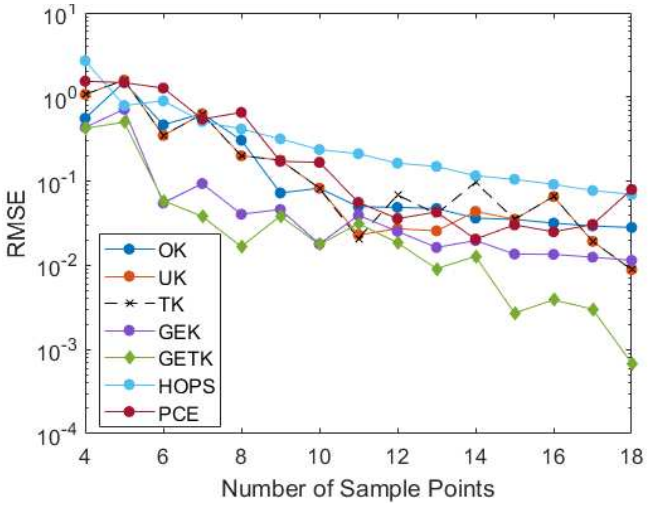


Fig. 4.2: RCS reconstruction RMSE convergence for the five Kriging methods, HOPS, and PCE for a 2-m diameter spherical dielectric scatterer with input uncertain relative permittivity given by $\varepsilon_r = \{\sim N(6, 1.5), -1\}$ at 115 MHz. taking the reference 1000-point Monte Carlo results as an "exact" solution.

With the previously presented convergence studies having established the novel GETK method as clearly the best performing choice, Fig. 4.3 shows the GETK reconstruction of the RCS QoI given only 6 sample points, for both the smoothly and rapidly varying test cases of Figs. 4.1 and 4.2, respectively. We see an almost exact agreement between GETK surrogate function results, at an extremely small number of sample points, and the reference 1000-point Monte Carlo solution, which demonstrates excellent reconstruction capability of the GETK method.

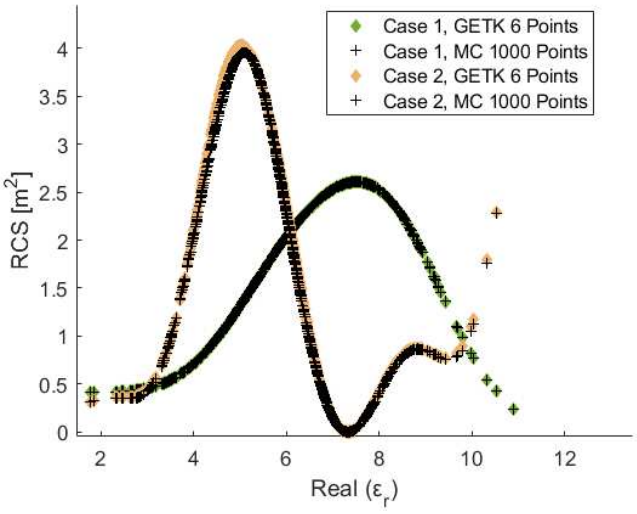


Fig. 4.3: Comparison of output RCS surrogate reconstructions obtained by the novel GETK method, at 6 reconstruction points, with the reference 1000-point Monte Carlo (MC) simulations for the dielectric spherical scatterer analyzed in Fig. 4.1 (case 1) and that from Fig. 4.2 (case 2).

Whereas the error convergence studies (in Figs. 4.1 and 4.2) are by all means most relevant for validation and comparative evaluation of different numerical approaches and techniques, the explicit results for the probability density function reconstruction as a function of the RCS QoI shown in Fig. 4.4 are perhaps the most practically important, as the principal goal of a UQ method is to reproduce an output probability density that most closely matches the true value. These results show that the probability density reconstruction by means of the novel GETK method is extremely accurate even with as few as 6 sample points in both test cases, when compared to the reference 1000-point Monte Carlo solution.

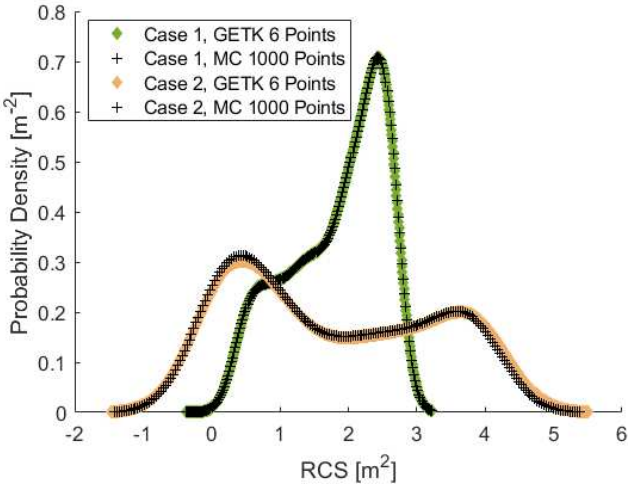


Fig. 4.4: Predicted output probability density function for the RCS QoI of two dielectric spherical scatterers (cases 1 and 2) analyzed in Fig. 4.3: comparison of results obtained by the novel GETK method at 6 reconstruction points with the reference 1000-point Monte Carlo simulations.

4.4.3 Test Case with Multi-Parameter Uncertainty

Our third test case deals with multi-parameter or higher-dimensional uncertainty, namely, with a problem where more than one input parameter in the model undergoes random variation. As this work compares and discusses different Kriging (and non-Kriging) solutions to UQ scattering problems and examines the accuracy and convergence of the solution and performance of various approaches with emphasis on predicting material uncertainty impact on FEM RCS computations, the most natural multi-parameter uncertainty and higher-dimensional input in this context seems to be a 2-dimensional ($D = 2$) perturbation of both the real and imaginary components of the dielectric permittivity of the scatterer. Hence, we consider the RCS of a 2-m diameter spherical dielectric scatterer with both the real and imaginary parts of ε_r being described independently by a normal random variable, $\varepsilon_r = \{\sim N(4.5, 0.25), \sim N(-1, 0.25)\}$, at 50 MHz, to verify the accuracy of the presented UQ approaches and examine the convergence of different Kriging methods. The output Monte Carlo RCS reconstruction for this test case is the classical "saddle" shape, shown in Fig. 4.5, which represents an excellent test case for many numerical methods due to an unstable critical point that acts as both a local minimum and maximum depending on the

axis. This test case should be challenging for surrogate functions due to both the higher dimensionality of the input as well as the output shape.

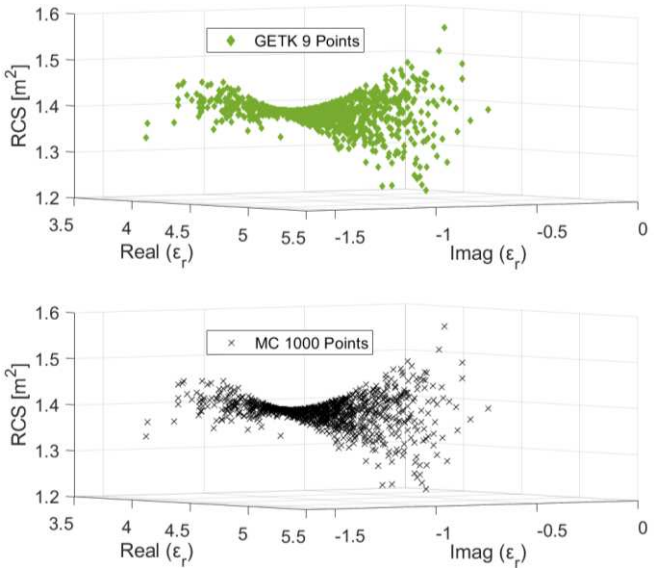


Fig. 4.5: Comparison of the output RCS GETK surrogate reconstruction, at 9 reconstruction points, with the 1000-point Monte Carlo results for a 2-m diameter spherical dielectric scatterer with both the real and imaginary components of the permittivity independently undergoing random variations, $\varepsilon_r = \{\sim N(4.5, 0.25), \sim N(-1, 0.25)\}$, at 50 MHz.

Fig. 4.5 also shows the RCS reconstruction of the surrogate function using the novel GETK method with only 9 sample points, where we observe a perfect match with the reference 1000-point Monte Carlo simulation. In fact, the agreement is so close that the 3-D rendering for this figure when the two sets of results are shown on top of each other would have a difficult time distinguishing between the two "saddle" surfaces. Hence, the results in Fig. 4.5 are presented in two separate figures, to better show the two data sets while still demonstrating how similar/exact they are when compared to each other and how the GETK RCS reconstruction is almost an exact match with MC solution even with an extremely small number of training input points.

Inspecting the convergence of the RMSE of the probability density function reconstruction versus the number of sample points using the five Kriging methods, HOPS, and PCE in Fig. 4.6, we conclude that the novel GETK method indeed converges much faster than all other techniques

tested. We also see that the PCE method, which is considered to be an excellent surrogate model for UQ, starts to perform quite well at large numbers of sample points; however, a much more rapid convergence and lower error of GEK and especially GETK methods with lower numbers of points are observed.

Fig. 4.7 shows the probability density distribution of the 2-D random case predicted using GETK from 9 sample points compared with the 1000-point Monte Carlo simulation. We observe that the novel GETK method matches the reference solution remarkably well and predicts a practically "exact" output density function for the RCS QoI with an extremely small number of points. Additionally, the figure includes for reference the results for the 1-D random case in which the input probability density has been collapsed to the mean in the imaginary part of ε_r. This demonstrates the importance of modeling all uncertain input variables in the model, as the addition of a second randomly varied parameter significantly affects the output probability density function for the considered quantity of interest.

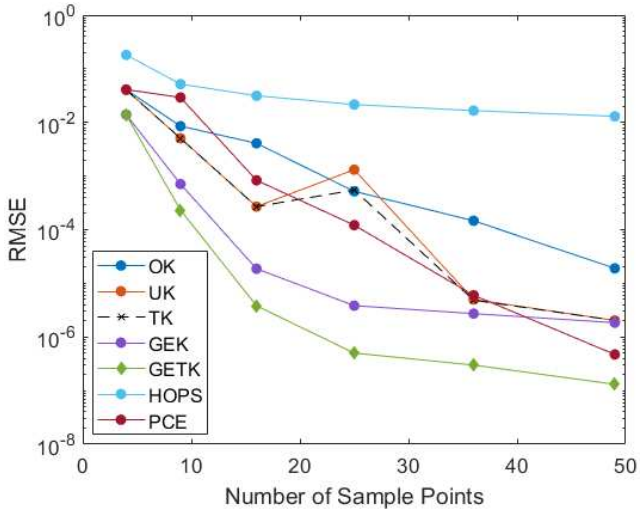


Fig. 4.6: Convergence of the RMSE of the probability density function reconstruction versus the number of sample points using the five Kriging methods, HOPS, and PCE for the dielectric spherical scatterer with 2-D uncertainty analyzed in Fig. 4.5, with the 1000-point Monte Carlo simulation as a validation norm.

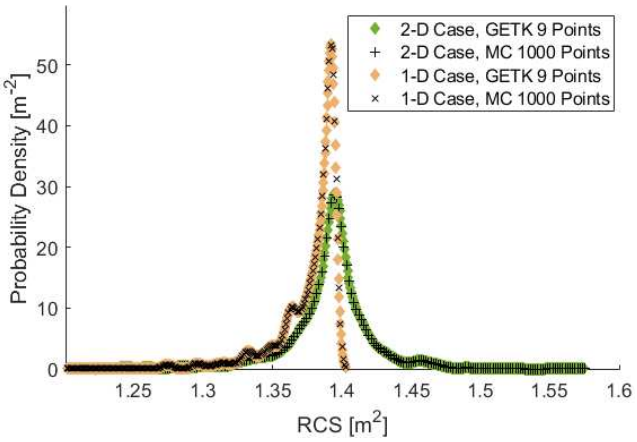


Fig. 4.7: Comparison of the probability density reconstruction as a function of RCS QoI for the dielectric spherical scatterer with the 2-D parameter space from Fig. 4.5 generated by the novel GETK method at 9 reconstruction points with the reference 1000-point Monte Carlo solution. Overlayed are also the results for the 1-D random case with the probability density of the imaginary part of ε_r collapsed to its mean, $\varepsilon_r = \{\sim N(4.5, 0.25), -1\}$.

4.5 Conclusion

This paper has presented the implementation and use of the Kriging methodology in uncertainty quantification in computational electromagnetics. It has provided consistent and unified description, derivation, implementation, and use of several advanced Kriging approaches, namely, the universal Kriging, Taylor Kriging, and gradient-enhanced Kriging methods, for reconstruction of probability density function in CEM problems. The paper has offered in-depth and comprehensive description of Kriging, as well as the necessary methods to calculate covariance, adjoints, correlation, etc., so that interested readers could use this material to construct their own Kriging method with little to no extra resources needed.

The paper has also proposed, derived, and demonstrated the gradient-enhanced Taylor Kriging methodology, novel to science and engineering in general. The novel GETK approach combines the advantages of all other methods. It benefits from both the gradient information, like with the GEK method, and basis functions in the regression scheme, as in the UK and TK formulations. Moreover, in our approach, the gradients are calculated from the suitable adjoint problem that

relates the deterministic forward problem to a specific quantity of interest. This is the first application of gradient enhanced Kriging with basis functions to CEM problems.

The paper has presented validation and comparative numerical studies of five different Kriging solutions to UQ scattering problems and has examined the accuracy and convergence of the solution and performance of various approaches. Numerical results using higher order curvilinear FEM-PML scattering modeling have shown that the presented Kriging methods are able to accurately and efficiently output probability density prediction for a quantity of interest (RCS of a scatterer) given the probability density of stochastic input parameters (material uncertainties), with very small numbers of sample (training) points, as a very efficient alternative to Monte Carlo simulations.

The novel GETK method has vastly outperformed all other approaches, Kriging (OK, UK, TK, and GEK) and non-Kriging (HOPS and PCE), in terms of surrogate function accuracy and convergence with increasing the number of sample points in all examples. While this work has shown Kriging and other results for 1-dimensional and 2-dimensional uncertainties that seemed to be the most natural inputs in the context of predicting material uncertainty impact on FEM RCS computations, the demonstrated advancement of the accuracy with the novel GETK method over all other approaches should translate to higher dimensional surrogate problems as well and is not limited to lower dimensional inputs.

Chapter 5

Conclusion

This dissertation has advanced the state of computational electromagnetics (CEM) by addressing critical challenges in ray tracing (RT) accuracy, parallelization, and uncertainty quantification (UQ) using Kriging methods. Through a series of novel methodologies and approaches, this work has significantly enhanced both the accuracy and efficiency of CEM simulations, particularly in large and complex environments.

The first key contribution is the improvement of ray tracing accuracy in the shooting and bouncing rays (SBR) method. By introducing optimized techniques for calculating ray cone sizes based on maximum separation angles and removing double-counted rays, this research has successfully demonstrated that SBR can achieve accuracy comparable to the more computationally expensive image theory (IT) method. The improvements in phase and magnitude accuracy, especially in large tunnel environments, enable SBR to match IT performance in scenarios where the correct computation of transmission path lengths at long distances is critical. The results show that SBR simulations using the derived optimal constant angular distribution are indistinguishable from those using per-ray angles, providing a robust and efficient solution for electromagnetic wave propagation modeling in tunnel environments. Overall, this enhanced SBR methodology offers a versatile, accurate, and computationally efficient alternative for wireless propagation modeling in both indoor and outdoor environments, supporting the growing demands of next-generation communication systems.

The second major contribution is the development of a parallelized SBR methodology that achieves unprecedented levels of efficiency through GPU acceleration. By leveraging NVIDIA's

OptiX Prime and CUDA programming interfaces, this research has created a unified parallelization framework that enables comprehensive parallelism across all components of the SBR algorithm, including the traditionally sequential tasks of electric field computation and postprocessing. The result is an SBR method with upwards of 99% parallelism, as demonstrated by Amdahl's parallelization scaling law, and a dramatic reduction in computation time for complex environments, such as long tunnels and urban landscapes. The optimized parallelized SBR method has proven to be highly efficient, handling millions of rays and converging to accurate solutions quickly and smoothly. These advancements in parallelism and efficiency will be invaluable to researchers and practitioners using ray-based methods in CEM and other high-frequency modeling applications, offering substantial improvements in speed without sacrificing accuracy.

Finally, this dissertation has made significant strides in uncertainty quantification (UQ) using advanced Kriging methodologies. Through the implementation and validation of several Kriging approaches—universal Kriging (UK), Taylor Kriging (TK), gradient-enhanced Kriging (GEK), and the novel gradient-enhanced Taylor Kriging (GETK)—this research has provided a highly efficient and accurate framework for UQ in CEM. The GETK method, in particular, has outperformed all other Kriging and non-Kriging approaches in terms of surrogate function accuracy and convergence with increasing sample points, demonstrating its superiority as a predictive tool for handling stochastic input parameters. By leveraging gradient information and basis functions, the GETK method has proven to be highly effective for predicting the impact of material uncertainties on quantities of interest, such as radar cross section (RCS) in scattering problems. The results show that the GETK approach not only improves accuracy for low-dimensional inputs but is also scalable to higher-dimensional uncertainty problems, making it a valuable alternative to computationally expensive Monte Carlo simulations.

In conclusion, the advancements in RT accuracy, parallelization, and UQ using Kriging methods presented in this dissertation collectively push the boundaries of CEM modeling. These contributions offer practical solutions for efficiently and accurately simulating complex electromagnetic environments, ensuring that future CEM applications—ranging from wireless communication systems to aerospace engineering—can meet the increasing demands for precision and computational feasibility.

References

[1] H. Ling, R.-C. Chou, and S.-W. Lee, "Shooting and bouncing rays: calculating the RCS of an arbitrarily shaped cavity," in *IEEE Transactions on Antennas and Propagation*, vol. 37, no. 2, pp. 194-205, Feb. 1989.

[2] S. Kasdorf, B. Troksa, C. Key, J. Harmon, and B. M. Notaros, "Advancing Accuracy of Shooting and Bouncing Rays Method for Ray-Tracing Propagation Modeling Based on Novel Approaches to Ray Cone Angle Calculation," IEEE Transactions on Antennas and Propagation, vol. 69, no. 8, Aug. 2021, pp. 4808-4815.

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

[4] Z. Yun, M. F. Iskander and Z. Zhang, "Development of a new shooting-and-bouncing ray (SBR) tracing method that avoids ray double counting," *IEEE Antennas and Propagation Society International Symposium. 2001 Digest*. Boston, MA, USA, 2001, pp. 464-467.

[5] S.-H. Chen and S.-K. Jeng, "SBR image approach for radio wave propagation in tunnels with and without traffic," *IEEE Transactions on Vehicular Technology*, vol. 45, no. 3, Aug. 1996, pp. 570-578.

[6] B. Troksa, C. Key, F. Kunkel, S. V. Savic, M. M. Ilic, and B. M. Notaros, "Ray Tracing Using Shooting-Bouncing Technique to Model Mine Tunnels: Theory and Verification for a PEC Waveguide," invited paper, Special Issue on Advanced Computational Electromagnetic Methodologies and Techniques, *ACES Journal*, Vol. 34, No. 2, February 2019, pp. 224–225.

[7] D. Didascalou, "Ray Optical Wave Propagation Modelling in Arbitrarily Shaped Tunnels," Ph.D. dissertation, Dept. Elect. Eng., *Universität Karlsruhe*, Karlsruhe, Germany, 2000.

[8] C. Key, B. A. Troksa, S. Kasdorf, and B. M. Notaros, "Non-Self-Adjacent Ray Classes for Parallelizable Shooting–Bouncing Ray Tracing Double Count Removal," IEEE Journal on Multiscale and Multiphysics Computational Techniques, vol. 5, pp. 245-254, 2020.

[9] C. Y. Kee and C. Wang, "Efficient GPU Implementation of the High-Frequency SBR-PO Method," *IEEE Antennas and Wireless Propagation Letters*, vol. 12, pp. 941-944, 2013.

[10] K. Xu, D. Ding and R. Chen, "Programmable graphics processing units (GPUs) accelerated SBR method for analyzing the scattering of open cavities," *2008 Asia-Pacific Microwave Conference*, pp. 1-4, 2008.

[11] S. G. Parker, J. Bigler, A. Dietrich, et al., "OptiX: a general-purpose ray tracing engine," ACM Transactions on Graphics, vol. 29, no. 4, pp. 1-13, 2010.

[12] S. Kasdorf, B. Troksa, C. Key, J. Harmon, S. Pasricha and B. M. Notaroš, "Parallel GPU Optimization of the Shooting and Bouncing Ray Tracing Methodology for Propagation Modeling," in *IEEE Transactions on Antennas and Propagation*, vol. 72, no. 1, pp. 174-182, Jan. 2024.

[13] R. Smith, "Uncertainty Quantification: Theory, Implementation, and Applications," SIAM, Philadelphia, 2014.

[14] J. J. Harmon, C. Key, D. Estep, T. Butler and B. M. Notaros, "Adjoint Sensitivity Analysis for Uncertain Material Parameters in Frequency Domain 3-D FEM," IEEE Transactions on Antennas and Propagation, vol. 69, no. 10, pp. 6669-6679, Oct. 2021.

[15] S. Kasdorf, J. J. Harmon and B. M. Notaroš, "Kriging Methodology for Uncertainty Quantification in Computational Electromagnetics," in *IEEE Open Journal of Antennas and Propagation*, vol. 5, no. 2, pp. 474-486, April 2024

[16] N. Sood, "Realistic Assessment of Novel Wireless Systems with Ray-Tracing Based Techniques," MASc Thesis, Dept. Elect. Comp. Eng., *Univ. of Toronto*, Toronto, Canada, 2012.

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

[18] V. Mohtashami and A. A. Shishegar, "A new double—counting cancellation technique for ray tracing using separation angle distribution," *2008 IEEE International RF and Microwave Conference*, Kuala Lumpur, 2008, pp. 306-310.

[19] Z. Yun and M. F. Iskander, "Ray Tracing for Radio Propagation Modeling: Principles and Applications," *IEEE Access*, vol. 3, pp. 1089-1100, 2015.

[20] S. Kasdorf, B. Troksa, J. Harmon, C. Key, and B. M. Notaros, "Shooting-Bouncing-Rays Technique to Model Mine Tunnels: Theory and Accuracy Validation," *Proceedings of the 2020 International Applied Computational Electromagnetics Society (ACES) Symposium – ACES2020*, July 27–31, 2020, Online Conference.

[21] C. Key, B. Troksa, F. Kunkel, S. V. Savic, M. M. Ilic, and B. M. Notaros, "Comparison of Three Sampling Methods for Shooting-Bouncing Ray Tracing Using a Simple Waveguide Model," *Proceedings of the 2018 IEEE International Symposium on Antennas and Propagation*, July 8–13, 2018, Boston, MA, USA, pp. 2273–2274.

[22] V. Orlovsky, "Decimal expansion of radius of inscribed sphere about a regular icosahedron with edge = 1," The On-Line Encyclopedia of Integer Sequences. Jul. 2010.

[23] A. C. Yucel, W. Sheng, C. Zhou, Y. Liu, H. Bagci and E. Michielssen, "An FMM-FFT Accelerated SIE Simulator for Analyzing EM Wave Propagation in Mine Environments Loaded With Conductors," in *IEEE Journal on Multiscale and Multiphysics Computational Techniques*, vol. 3, pp. 3-15, 2018.

[24] D. G. Dudley, M. Lienard, S. F. Mahmoud and P. Degauque, "Wireless propagation in tunnels," *IEEE Antennas and Propagation Magazine*, vol. 49, no. 2, pp. 11-26, April 2007.

[25] B. M. Notaros and L. C. Kempel, "Computational Electromagnetics for Antennas," in *Antenna Engineering Handbook, 5th edition*, J. L. Volakis (ed.), *McGraw-Hill Education*, pp. 1343–1375, 2018.

[26] B. A. Troksa, "GPU Accelerated Cone Based Shooting Bouncing Ray Tracing," M.S. Thesis, Department of Electrical and Computer Engineering, Colorado State University, 2019.

[27] A. Santos, J. M. Teixeira, T. Farias, V. Teichrieb, and J. Kelner, "Understanding the Efficiency of kD-tree Ray-Traversal Techniques over a GPGPU Architecture," *International Journal of Parallel Programming*, vol. 40, no. 3, pp. 331–352, 2011.

[28] C Woolley, "GPU optimization fundamentals," NVIDIA Corp., Dev. Tech. Eng., 2013.

[29] M. F. Iskander and Z. Yun, "Propagation prediction models for wireless communication systems," *IEEE Transactions on Microwave Theory and Techniques*, vol. 50, no. 3, pp. 662-673, March 2002.

[30] Ansys Savant 2023 R1, ANSYS, Inc. Available: http://www.ansys.com. Accessed June 28, 2023.

[31] J. F. Monserrat and M. Fallgren, *METIS Deliverable D6.1 Simulation guidelines[J]*, October 2013, [online] Available: http://ww.metis2020.com/.

[32] A. W. Mbugua, Y. Chen, L. Raschkowski, L. Thiele, S. Jaeckel and W. Fan, "Review on Ray Tracing Channel Simulation Accuracy in Sub-6 GHz Outdoor Deployment Scenarios," in *IEEE Open Journal of Antennas and Propagation*, vol. 2, pp. 22-37, 2021

[33] G. Steinböck, A. Karstensen, P. Kyösti and A. Hekkala, "A 5G hybrid channel model considering rays and geometric stochastic propagation graph," *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Valencia, Spain, 2016, pp. 1-6

[34] C. Feng and L. Yuan-jian, "Study on propagation characteristics for typical outdoor environment," *2015 Asia-Pacific Microwave Conference (APMC)*, Nanjing, China, 2015, pp. 1-3.

[35] J. J. Harmon and B. M. Notaros, "Reducing the Computational Expense of Uncertainty Quantification in Computational Electromagnetics: A Goal-Oriented Perspective," *Invited Paper*, Special Session "International Standards Development and Applications," *Proceedings of the 2022 IEEE International Symposium on Antennas and Propagation*, July 10–15, 2022, Denver, Colorado, pp. 659–660.

[36] D. Estep and D. Neckels, "Fast methods for determining the evolution of uncertain parameters in differential equations," *J. Comput. Phys.*, vol. 213, no. 2, pp. 530-556, April 2006.

[37] D. Estep and D. Neckels, "Fast methods for determining the evolution of uncertain parameters in reaction-diffusion equations," *Computer Methods in Applied Mechanics and Engineering*, vol. 196, pp. 3967-3979, 08 2007.

[38] J. H. S. Baar, "Stochastic Surrogates for Measurements and Computer Models of Fluids," Ph.D. Dissertaition, Delft University of Technology, Dec. 2014.

[39] R. L. Wagner, J. Song, and W. C. Chew, "Monte Carlo simulation of electromagnetic scattering from two-dimensional random rough surfaces," *IEEE Transactions on Antennas and Propagation*, vol. 45, no. 2, pp. 235–245, 1997.

[40] Q. Li, H. Chan, and L. Tsang, "Monte Carlo simulations of wave scattering from lossy dielectric random rough surfaces using the physics-based two-grid method and the canonical-grid method," *IEEE Transactions on Antennas and Propagation*, vol. 47, no. 4, pp. 752–763, 1999.

[41] N. Garcia and E. Stoll, "Monte Carlo calculation for electromagnetic wave scattering from random rough surfaces," *Phys. Rev. Lett.*, vol. 52, pp. 1798–1801, May 1984.

[42] S. M. Smith and C. Furse, "Stochastic FDTD for analysis of statistical variation in electromagnetic fields," *IEEE Transactions on Antennas and Propagation*, vol. 60, no. 7, pp. 3343–3350, 2012.

[43] K. Masumnia-Bisheh, K. Forooraghi, and M. Ghaffari-Miab, "Electromagnetic uncertainty analysis using stochastic FDFD method," *IEEE Transactions on Antennas and Propagation*, vol. 67, no. 5, pp. 3268– 3277, 2019.

[44] A. C. M. Austin and C. D. Sarris, "Efficient analysis of geometrical uncertainty in the FDTD method using polynomial chaos with application to microwave circuits," *IEEE Transactions on Microwave Theory and Techniques*, vol. 61, no. 12, pp. 4293–4301, 2013.

[45] R. S. Edwards, A. C. Marvin, and S. J. Porter, "Uncertainty analyses in the finite-difference time-domain method," *IEEE Transactions on Electromagnetic Compatibility*, vol. 52, no. 1, pp. 155–163, 2010.

[46] U. K. Khankhoje and S. Padhy, "Stochastic solutions to rough surface scattering using the finite element method," *IEEE Transactions on Antennas and Propagation*, vol. 65, no. 8, pp. 4170–4180, 2017.

[47] G. J. K. Tomy and K. J. Vinoy, "A fast polynomial chaos expansion for uncertainty quantification in stochastic electromagnetic problems," *IEEE Antennas and Wireless Propagation Letters*, vol. 18, no. 10, pp. 2120–2124, 2019.

[48] G. J. K. Tomy and K. J. Vinoy, "Neumann-expansion-based FEM for uncertainty quantification of permittivity variations," *IEEE Antennas and Wireless Propagation Letters*, vol. 19, no. 4, pp. 561–565, 2020.

[49] M. Branicki and A. Majda, "Fundamental limitations of polynomial chaos for uncertainty quantification in systems with intermittent instabilities," *Communications in Mathematical Sciences,* vol. 11, 2013, pp. 55–103.

[50] D. Voyer, F. Musy, L. Nicolas, and R. Perrussel, "Comparison of methods for modeling uncertainties in a 2D hyperthermia problem," *Progress In Electromagnetics Research B*, vol. 11, pp. 189–204, 2009.

[51] M. A. Drissaoui, S. Lanteri, P. Leveque, F. Musy, L. Nicolas, R. Perrussel, and D. Voyer, "A stochastic collocation method combined with a reduced basis method to compute uncertainties in numerical dosimetry," *IEEE Transactions on Magnetics*, vol. 48, no. 2, pp. 563–566, 2012.

[52] J. Ochoa and A. Cangellaris, "Macro-modeling of electromagnetic domains exhibiting geometric and material uncertainty," *Applied Computational Electromagnetics Society Journal*, vol. 27, pp. 80–87, 2012.

[53] M. Gossye, G. Gordebeke, K. Y. Kapusuz, D. V. Ginste, and H. Rogier, "Uncertainty quantification of waveguide dispersion using sparse grid stochastic testing," *IEEE Transactions on Microwave Theory and Techniques*, vol. 68, no. 7, pp. 2485–2494, 2020.

[54] D. Poljak, S. Šesnić, M. Cvetković, S. Lallechere and K. El Khamlichi Drissi, "On some applications of stochastic collocation method in computational electromagnetics: Applications in ground penetrating radar, bioelectromagnetics, grounding systems and buried lines," in *2016 24th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, 2016, pp. 1–5.

[55] J. Son and Y. Du, "Comparison of intrusive and nonintrusive polynomial chaos expansion-based approaches for high dimensional parametric uncertainty quantification and propagation," *Computers & Chemical Engineerin*g, vol. 134, p. 106685, 2020.

[56] M. Eldred, "Recent advances in non-intrusive polynomial chaos and stochastic collocation methods for uncertainty analysis and design," *50th AIAA/ASME/ASCE/AHS/ASC*

Structures, Structural Dynamics, and Materials Conference, 4-7 May 2009, Palm Springs, California, pp. 1–37, 05 2009.

[57] Y. Li, G. Lei, G. Bramerdorfer, S. Peng, X. Sun, and J. Zhu, "Machine Learning for Design Optimization of Electromagnetic Devices: Recent Developments and Future Directions," *Applied Sciences,* vol. 11, 2021.

[58] M. O. Akinsolu, K. K. Mistry, B. Liu, P. I. Lazaridis and P. Excell, "Machine Learning-assisted Antenna Design optimization: A Review and the State-of-the-art," *2020 14th European Conference on Antennas and Propagation (EuCAP)*, 2020, pp. 1-5.

[59] Q. Wu, H. Wang and W. Hong, "Multistage Collaborative Machine Learning and its Application to Antenna Modeling and Optimization," *IEEE Transactions on Antennas and Propagation*, vol. 68, no. 5, pp. 3397-3409, May 2020.

[60] D. Kan, D. Spina, S. De Ridder, F. Grassi, H. Rogier and D. V. Ginste, "A Machine-Learning-Based Epistemic Modeling Framework for Textile Antenna Design," *IEEE Antennas and Wireless Propagation Letters*, vol. 18, no. 11, pp. 2292-2296, Nov. 2019.

[61] R. P. Dwight and Z.-H. Han, "Efficient Uncertainty Quantification using Gradient-Enhanced Kriging," *11th AIAA Non-Deterministic Approaches Conference,* 2009.

[62] W. Yamazaki, M. P. Rumpfkeil and D. J. Mavriplis, "Design Optimization Utilizing Gradient/Hessian Enhanced Surrogate Model," *American Institute of Aeronautics and Astronautics, 28th AIAA Applied Aerodynamics Conference,* 28 June - 1 July 2010, Chicago, IL, pp.1-23.

[63] J. Peter and M. Marcelet, "Comparison of Surrogate Models for Turbomachinery Design," *WSEAS Transactions on Fluid Mechanics,* vol. 3, no. 1, pp. 10-17, Jan. 2008.

[64] H.-S. Chung and J. J. Alonso, "Using Gradients to Construct Cokriging Approximation Models for High-Dimensional Design Optimization Problems," *American Institute of Aeronautics and Astronautics, 40th AIAA Aerospace Sciences Meeting and Exhibit*, January 14–17, 2002, Reno, NV, pp. 1-15.

[65] L. Zhao, K. K. Choi and I. Lee, "Metamodeling Method Using Dynamic Kriging for Design Optimization," *AIAA Journal,* vol. 49, no. 9, Sept. 2011.

[66] S. Bhattacharjee, P. Mitra and S. K. Ghosh, "Spatial Interpolation to Predict Missing Attributes in GIS Using Semantic Kriging," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 52, no. 8, pp. 4771-4780, Aug. 2014

[67] S. Yu and Y. Li, "Active Learning Kriging Model With Adaptive Uniform Design for Time-Dependent Reliability Analysis," *IEEE Access,* vol. 9, pp. 91625-91634, 2021.

[68] B. Xia, T. Lee, K. Choi and C. Koh, "A Novel Adaptive Dynamic Taylor Kriging and Its Application to Optimal Design of Electromagnetic Devices," *IEEE Transactions on Magnetics,* vol. 52, no. 3, pp. 1-4, Mar. 2016.

[69] E. S. Siah, M. Sasena, J. L. Volakis, P. Y. Papalambros and R. W. Wiese, "Fast parameter optimization of large-scale electromagnetic objects using DIRECT with Kriging metamodeling," *IEEE Transactions on Microwave Theory and Techniques*, vol. 52, no. 1, pp. 276-285, Jan. 2004.

[70] A. Pietrenko-Dabrowska, S. Koziel and M. Al-Hasan, "Accurate Modeling of Antenna Structures by Means of Domain Confinement and Gradient-Enhanced Kriging," *2021 15th European Conference on Antennas and Propagation (EuCAP)*, Dusseldorf, Germany, 2021, pp. 1-5.

[71] A. Pietrenko-Dabrowska and S. Koziel, “Antenna Modeling Using Variable-Fidelity EM Simulations and Constrained Co-Kriging,” *IEEE Access*, vol. 8, pp. 91048-91056, 2020.

[72] S. Koziel and S. Ogurtsov, “Multi-Objective Design of Antennas Using Variable-Fidelity Simulations and Surrogate Models,” *IEEE Transactions on Antennas and Propagation*, vol. 61, no. 12, pp. 5931-5939, Dec. 2013.

[73] A. Pietrenko-Dabrowska, S. Koziel, and M. Al-Hasan, “Cost-Efficient Bi-Layer Modeling of Antenna Input Characteristics Using Gradient Kriging Surrogates,” *IEEE Access*, vol. 8, pp. 140831-140839, 2020.

[74] P. Manfredi and R. Trinchero, “A Probabilistic Machine Learning Approach for the Uncertainty Quantification of Electronic Circuits Based on Gaussian Process Regression,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 8, pp. 2638-2651, Aug. 2022

[75] P. Manfredi, “Probabilistic Uncertainty Quantification of Microwave Circuits Using Gaussian Processes,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 71, no. 6, pp. 2360-2372, June 2023.

[76] S. Kasdorf, J. Harmon, and B. M. Notaros, “Kriging Methodology for Predicting Material Uncertainty Impact on FEM Scattering Computations,” *Proc. 2023 IEEE International Symposium on Antennas and Propagation*, July 23–28, 2023, Portland, Oregon, pp. 53-54.

[77] S. Ulaganathan, I. Couckuyt, T. Dhaene, E. Laermans and J. Degroote, "On the use of gradients in Kriging Surrogate Models," *Proceedings of the 2014 Winter Simulation Conference*, Savannah, GA, USA, 2014, pp. 2692-2701.

[78] C. Key, A. Smull, D. Estep, T. Butler, and B. M. Notaros, “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 Propagation*, Vol. 68, No. 5, May 2020, pp. 3791–3806.

[79] J. J. Harmon, C. Key, D. Estep, T. Butler, and B. M. Notaros, “Adjoint-based Accelerated Adaptive Refinement in Frequency Domain 3-D Finite Element Method Scattering Problems,” *IEEE Transactions on Antennas and Propagation*, Vol. 69, No. 2, February 2021, pp. 940–949.

[80] C. Key, J. J. Harmon, and B. M. Notaros, “Correlations in A posteriori Error Trends for the Finite Element Method in the Presence of Changing Material Parameters,” *IEEE Antennas and Wireless Propagation Letters*, Vol. 20, No. 12, December 2021, pp. 2516–2518.

[81] J. J. Harmon and B. M. Notaroš, "Accelerated Adaptive Error Control and Refinement for SIE Scattering Problems," *IEEE Transactions on Antennas and Propagation,* vol. 70, no. 10, pp. 9497-9510, Oct. 2022.

[82] S. Oladyshkin and W. Nowak, “Data-driven uncertainty quantification using the arbitrary polynomial chaos expansion”, *Reliability Engineering & System Safety*, vol. 106, 2012, pp 179-190.

[83] H. Liu, "Taylor Kriging Metamodeling For Simulation Interpolation, Sensitivity Analysis, and Optimization," Ph.D. Dissertation*,* Dept. of Industrial and Systems Engineering, University of Auburn, May 2009.

[84] M. Diago-Mosquera, A. Aragon-Zavala, L. Azpilicueta, R. Shubair and F. Falcone, "A 3-D Indoor Analysis of Path Loss Modeling Using Kriging Techniques," *IEEE Antennas and Wireless Propagation Letters,* vol. 21, no. 6, 2022.

[85] P. Wang, Z. Feng, Y. Tang and Y. Zhang, "A fingerprint database reconstruction method based on ordinary Kriging algorithm for indoor localization," *International Conference on Intelligent Transportation, Big Data & Smart City*, 2019.

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

[87] B. Xia, N. Baatar, Z. Ren and C. Koh, "A Numerically Efficient Multi-Objective Optimization Algorithm: Combination of Dynamic Taylor Kriging and Differential Evolution," *IEEE Transactions on Magnetics,* vol. 51, no. 3, 2015.

[88] S. Oladyshkin, "aPC Matlab Toolbox: Data-driven Arbitrary Polynomial Chaos" (https://www.mathworks.com/matlabcentral/fileexchange/72014 -apc-matlab-toolbox-data-driven-arbitrary-polynomial-chaos), MATLAB Central File Exchange. Retrieved July 5, 2023.

[89] 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.

[90] 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, pp. 517-529, 2006.

[91] B. M. Notaros, "Higher Order Frequency-Domain Computational Electromagnetics," *Invited Review Paper*, Special Issue on Large and Multiscale Computational Electromagnetics, *IEEE Transactions on Antennas and Propagation*, Vol. 56, No. 8, pp. 2251-2276, August 2008.

[92] A. P. Smull, A. B. Manic, S. B. Manic, and B. M. Notaros, "Anisotropic Locally-Conformal Perfectly Matched Layer for Higher Order Curvilinear Finite-Element Modeling," *IEEE Transactions on Antennas and Propagation*, Vol. 65, No. 12, pp. 7157–7165, December 2017.

[93] J. Epperson, "On the Runge Example," *The American Mathematical Monthly*, vol. 94, 1987, pp. 329-341.