August 2023

SINGULAR INTEGRATION BY INTERPOLATION FOR INTEGRAL EQUATIONS

Ioannis Kyriakou
University of Massachusetts Amherst

SINGULAR INTEGRATION BY INTERPOLATION FOR INTEGRAL EQUATIONS

A Dissertation Presented

by

IOANNIS G. KYRIAKOU

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

May 2023

Electrical & Computer Engineering

SINGULAR INTEGRATION BY INTERPOLATION FOR INTEGRAL EQUATIONS

A Dissertation Presented

by

IOANNIS G. KYRIAKOU

Approved as to style and content by:

Marinos N. Vouvakis, Chair

Paul Siqueira, Member

Do-Hoon Kwon, Member

Ramakrishna Janaswamy, Member

Nestor Guillen, Member

Christopher Hollot, Department Chair
Electrical & Computer Engineering

To my beautiful and supportive partner Abigail.

ACKNOWLEDGMENTS

This work marks a big achievement in my life and I would like to wholeheartedly thank all the people, who helped me be the person I am today; as well as all those, who helped me reach this goal.

First and foremost I would like to thank my family for their love, support and teachings, but especially my father, my role model, for his guidance and for showing me how to carve and follow my own path, a path with a heart! Big thanks go to my little brother Xristo too for always and on a regular basis brightening up my weekends. Thanks also go to all my good friends, but especially to my closest friends Vamvakopoulos, Efkarpidis and Chrysselis for all the memorable escapades. A honorary mention also goes out to all the members of the karate club in Umass and especially to my instructors Wilson and Carver and the fellow practitioners Jordan and Kyle for challenging me to better myself as is fundamental in the spirit of the art.

Secondly, I would like to thank all the members, past and present of my academic research group, but especially my colleague Dimitri and of course my advisor Marino for showing a true interest in my progress, but also in my personal endeavors. My advisor has invested a lot of time, efforts, hopes and with a sincere teacher's passion made sure to pass on all of his knowledge and experience to me; all of which I am very grateful for.

An honorary mention goes out to Steve for an inspiring talk that pushed me to make my proposal happen.

Last but certainly not least, I cannot thank enough my beloved partner Abigail for all she has and continues to do for me! Through thick and thin we've shared dreams, hopes, laughter but also endeavors, quarantines and struggles alike!

All these people, directly or indirectly mentioned, made this work possible and therefore deserve a place on this page.

ABSTRACT

SINGULAR INTEGRATION BY INTERPOLATION FOR INTEGRAL EQUATIONS

MAY 2023

IOANNIS G. KYRIAKOU

B.Sc., DEMOCRITUS UNIVERSITY OF THRACE, XANTHI, GREECE

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Marinos N. Vouvakis

Maxwell's equations and the laws of Electromagnetics (EM) govern a plethora of electrical, optical phenomena with applications on wireless, cellular, communications, medical and computer hardware technologies to name a few. A major contributor to the technological progress in these areas has been due to the development of simulation and design tools that enable engineers and scientists to model, analyze and predict the EM interactions in their systems of interest. At the core of such tools is the field of Computational Electromagnetics (CEM), which studies the solution of Maxwell's equations with the aid of computers. The advances in these applications technologies, in return, demand increasingly more efficient and accurate CEM methods.

Among the many CEM methodologies that are currently in broad use, the Bound-ary Element Method (BEM) or surface Method of Moments (MoM), is perhaps the most popular in solving electrically large or electrically small multi-layered structures. In BEM, the surfaces of conductors and dielectrics are discretized to triangular or quadrilateral elements and the equivalent currents on them are convolved with the appropriate Green's function at all observations on the mesh to produce a fully populated impedance matrix to be solved with an appropriate excitation. The reliability, accuracy and speed of BEM, among others, critically depends on the method used to perform the singular four-dimensional convolution integrals between source and observation surface currents through a Green's function, that exhibits a singularity when observation and source elements touch or overlap. Large literature has been devoted in addressing this important issue, and methods involving using singularity subtraction, cancellation or even full 4D integral evaluations. Each of these approaches offer certain advantages, but they tend to require thousands of (often complicated) function evaluations for a single impedance matrix singular integration, it is noted that a typical problem may involve tens or hundreds of millions of such singular integrations.

In this dissertation, an unconventional approach of calculating all weakly singular and near weakly singular integrals, encountered in the BEM solution of the Electric Field Integral Equation (EFIE), as well as near singular integrals encountered in the BEM solution of the Magnetic Field Integral Equation (MFIE) in flat triangular meshes, is presented. Instead of specialized integration rules such as singularity subtraction or cancellation, *universal* look-up-tables and multi-dimensional interpolation are used.

Firstly, frequency independent integral expressions, equivalent to the original EFIE-BEM, MFIE-BEM element matrix expressions are derived, in order to facilitate the construction of said *universal* look-up-tables of integrals. The domain of these functions is discretized by *hp* refinement, i.e., the size, *h* and approximation order, *p*, of the interpolation elements of the entire interpolation domain can be varied independently. Because of the high-dimensional nature of the interpolation domain, from three dimensional to six dimensional, the interpolation over each element is performed with either sparse grids or low-rank tensor train approximations. The integrals are pre-computed into the tables using a state-of-the-art singularity subtraction method at maximum accuracy. Consequently, during run-time, these tables are loaded and any arbitrary singular integral is recovered by multi-dimensional interpolation.

The method is compared to a state-of-the-art singularity subtraction technique for the lowest order Rao-Wilton-Glisson (RWG) basis functions in various PEC flat triangular meshes. For EFIE common triangle, weakly singular, in accuracy, while offering over 150× speed-ups. Similarly for EFIE common edge, near weakly singular, interactions it shows about 50× speed-ups but at a somewhat lower, yet acceptable, accuracy. The tensor decomposition approach improves the accuracy to the level of the state-of-the-art and offers about 20× speed-ups, while it also has a controllable accuracy and speed. Lastly, for MFIE common edge, near hyper singular, interactions accuracy is improved by $1-2$ decimal digits, while offering 20× speed-ups. For a typical BEM run using the single level fast multiple method (FMM) accelerator, the end-to-end set-up time speed improvement with the proposed approach is $15-20\%$.

TABLE OF CONTENTS

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

CHAPTER 1

INTRODUCTION

1.1 Background

This research in this dissertation broadly falls under the discipline of computational modeling of full-wave electromagnetic interactions in arbitrary (shape and topology) structures embedded in an unbounded homogeneous medium such as free space. Such computational electromagnetics (CEM) modeling tools [1–3] aim to rigorously solve, via numerical methods, some form of Maxwell equations [4,5] and their associated boundary conditions [5–7] to produce rather accurate approximations of various electrical or/and optical quantities or properties. Such tools are generally used in the design and virtual prototyping of many devices, components sub-systems and systems in the broader areas of integrated circuits and packaging [8], signal and power integrity [8], antennas and propagation [9], microwave [5,6], millimeter wave [10,11], terahertz [12] or even photonic [13] regimes.

Many effective and reliable CEM methodologies have been proposed over the years, that can be loosely grouped as: differential equations (DE) methods [14,15] and integral equation (IE) methods [3,16]. In the former, the differential (or integral) form of Maxwell equations is numerically solved for the electric or/and magnetic vector field responses, subject to the appropriate EM boundary conditions and some excitation. Whereas in the latter case, an integral equation (equation having the unknown quantity in the integrand) of the vector current or charge density is first analytically derived from Maxwell's equations through the use of an appropriate Green's function [3], and then it is discretized and numerically solved, subject to some known excitation. DE methods are better suited for intricate and heterogeneous structures that produce somewhat spatially localized/confined field responses, whereas IE methods tend to excel at structures that produce fields that propagate outwards to infinity, such as scattering shown in 1.1 and radiation, example shown in 1.2. The most prominent DE CEM methods widely used in commercial tools are the finite element method (FEM) [14] and its variants [17, 18] and the finite difference time-domain (FDTD) [15] and its variants [19, 20]. Among IE methods, surface integral equations [21] based on the electric field integral equation (EFIE) [3, 21] and combined field integral equation (CFIE) [3, 21] for perfect electric conducting structures, and the Poggio, Miller, Chang, Harrington, Wu and Tsai (PMCHWT) [22–24] formulation for dielectric structures, and their variants, are the most popular methods and have been used in commercial tools.

Surface IE (SIE) methods employ unknowns only on surfaces (interfaces between materials and/or background medium) of three-dimensional structures, essentially achieving a form of dimension reduction, that leads to significantly less unknowns than DE methods. However, this benefit comes with intricate inner workings that are present in all (SIE) methods. Those are: (a) different SIE formulations are necessary depending on the type of interface, i.e. PEC to background medium, dielectric to dielectric, dielectric to background medium, etc. (b) Special treatments are necessary to solve for topologically non-simple structures e.g., non-manifold structures, non-simply connected objects, etc. (c) Special care must be taken in evaluating the *singular and oscillatory integrals* encountered in the numerical solution of SIEs. All IE methods for EM, particularly at moderate and high frequencies require an accurate integration of oscillatory integrals something that is usually the topic of fast IE methods [25]. This work focuses on the singular aspect of those integrations, that are particularly critical for the accuracy and efficiency of SIEs, where the singularity

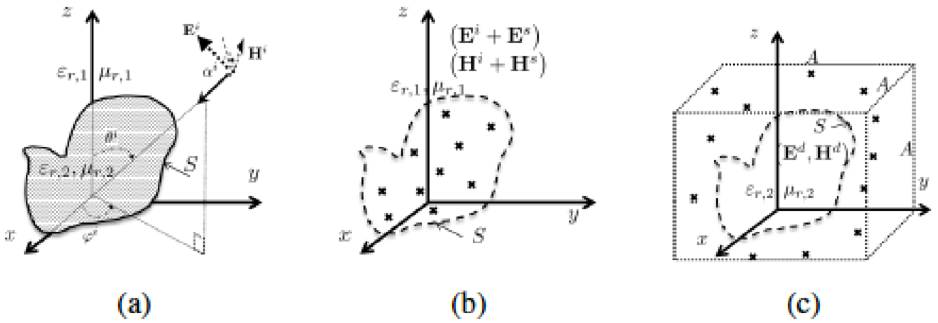

z
$\mathbf{E}^i$
$\mathbf{H}^i$
α^i
$\varepsilon_{r,1}$ $\mu_{r,1}$
θ^i
$\varepsilon_{r,2}$; $\mu_{r,2}$
S
y
x
φ^i
(a)
z
$(\mathbf{E}^i + \mathbf{E}^s)$
$(\mathbf{H}^i + \mathbf{H}^s)$
$\varepsilon_{r,1}$; $\mu_{r,1}$
y
x
S
(b)
z
A
A
S
$(\mathbf{E}^d, \mathbf{H}^d)$
A
$\varepsilon_{r,2}$ $\mu_{r,2}$
y
x
(c)

Figure 1.1. The electromagnetic scattering problem illustration a) Original problem b) External equivalent problem proposing random sources within the scatterer's region c) Internal equivalent problem featuring random sources in a controlled domain outside the scatterer's region, [26]

Figure 1.2. A planar ultra-wideband modular antenna (PUMA) array, [27]

is stronger than that encountered in volume IEs (VIE), due to the dimensionality reduction of the integration domain.

1.2 Problem Statement

The aim of this dissertation research is to develop methodologies and algorithms for the fast, error-controllable, and reliable computation of *large collections of singular interactions* encountered in the numerical solution of discrete integral equations (IE)

[21], through the Boundary Element Method (BEM) [28], aka Method of Moments (MoM) [3], or other similar discretization/solution techniques.

All discrete IE methods explicitly or implicitly form a dense matrix, that in turn is solved with a numerical linear algebra solution method to produce a solution vector that corresponds to the unknown source density distribution, e.g. current or charge density, on (or in) the physical structure to be investigated. For most EM discrete IE problems, each matrix entry represents the pair-wise interaction between two elementary vector current densities situated at two distinct regions on the physical structure. These elementary vector current densities, termed vector basis functions, have finite support, i.e., the current density is non-zero only in a relatively small region of the physical structure. The support regions in most cases are defined by a few elements in a mesh, where a mesh is an approximate discrete covering of the physical structure and is used to define those basis function supports. In most cases, IE methods rely on simplicial surface meshes, where flat triangular patches of arbitrary shape are used. Each of these pair-wise current density interactions entails in computing a few two-dimensional convolutions between the vectorial basis function pair through a global and singular kernel that corresponds to the Green's function (point source response) [3,16] of the underlying integral formulation. The focus of this work will be on the homogeneous unbounded medium Helmholtz equation Green's function and its derivative, as they arise in most widely used IE formulations in EM radiation and scattering.

Most of those pair-wise current density interactions can be computed in a relatively straightforward manner, using numerical cubatures [29,30] of Gaussian type defined on triangles [30]. In practice, because these evaluations are far too many, $O(N^2)$, where N is the number of basis functions, advanced grouping-based approaches employ approximate representations of the Green's function to speed-up evaluations and reduce complexity. The most popular approaches of this kind are the fast multipole method (FMM) [25, 31, 32] that leads to time-complexity $O(N^{1.5})$ and its multilevel counterpart, the multilevel FMM (MLFMM) [33–35] with $O(N \log N)$ complexity. Despite the presence of the Green's function singularity (typically one over distance or distance square), these methods are possible precisely because the interaction integral is regular, i.e. the basis functions are non-singular, and their *regions of support is not overlapping*, to trigger the Green's function singularity. Nevertheless, those interactions alone are not sufficient to produce the correct result of discrete IE. In fact, those non-singular interactions alone are not sufficient to produce a unique solution. The remaining interactions, those involving singular convolution-type integrals are far fewer, but ever important. Those are the interactions that correspond to basis function pairs that have overlapping or/and touching support regions, and their computation is the subject of this dissertation.

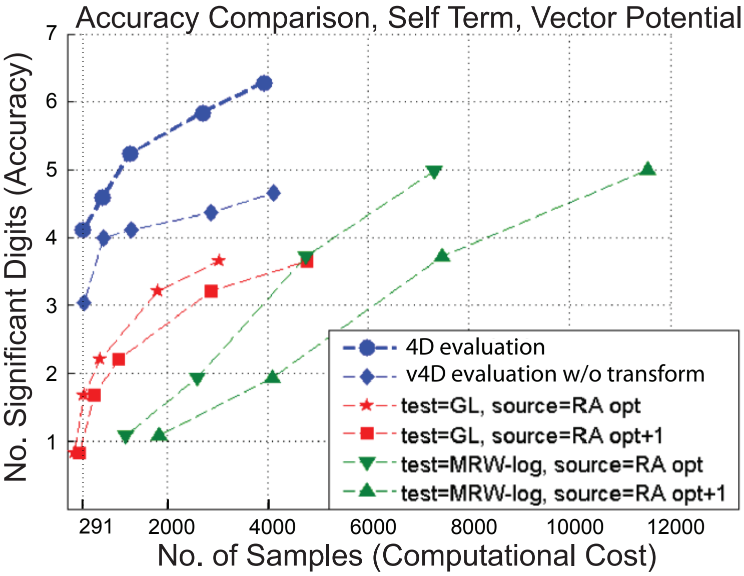


Figure 1.3. Accuracy vs number of samples for the integration of a self term vector potential integral via various methods, [36].

1.3 Importance and Broader Impacts

As will be outlined in this section, the problem of numerically evaluating a *large collection of singular interactions* encountered in discrete integral equation (IE) solution methods directly affects for the efficiency, accuracy, and reliability of those methods. At the same time, it has broad impacts across various methods in CEM and other computational science disciplines.

Although the total number of singular interactions is $O(N)$, which is far smaller than the non-singular ones (near-field and far-field), the computation/approximation of a single singular interaction is vastly more time-consuming that a non-singular one. The singular nature of the integrand necessitates advanced numerical treatments, outlined in the next section, that require large numbers of floating-point operations (FLOP) per integral evaluation. A case in point is the Figure 1.3 adopted from [36], that shows the total number of function evaluations required to reach a certain degree of accuracy for a single overlapping singular integral (a few such integrals are necessary to form a singular interaction). To give a perspective, a typical non-singular evaluation requires less than 50 function evaluations for $4-5$ digits of accuracy, whereas the state-of-the-art method of [36] in Figure 1.3 for a singular integral case, requires upwards of 1000. The situation is further exacerbated by the fact that each of those 1000 function evaluation requires a significantly larger number of FLOPs than that of the non-singular case.

Despite that fact, in full MoM the $O(N^2)$ complexity of the non-singular interactions ends up dominating the computational time in most problems. However, the evaluation of singular interactions can become an important chunk of the computation for fast integral equation methods such as MLFMM. Because of their many benefits, those methods have become the de-facto computational engine for IE-based commercial and academic CEM tools, such as Dassault Systèmes CST Studio Suite, Altair Engineering FEKO, Cadence AWR Axiem or Keysight Momentum. For example, the MLFMM reduces the set-up time complexity of far (non-singular) interactions to $O(N \log N)$, and because the $O(N)$ singular terms have significantly higher cost per term, they can become the dominant set-up time bottleneck.

Speed assessments of discrete IE methods are highly dependent on the degree of requested accuracy, and the degree of simulation reliability. The higher the requested accuracy and the more trustworthy a simulation must be, the longer the run time becomes. However, the degree of overall accuracy and reliability is most closely linked to the singular interaction evaluation precision [36–46]. Singular interactions produce significantly larger (in magnitude) matrix entries than non-singular ones, thus even small evaluation relative errors can produce large absolute errors for these terms. Moreover, because singular interactions, among others, reside on the diagonal entries of the matrix they tend to dominate the matrix condition number and solution accuracy. In fact, any matrix arising from the discretization of fist-kind Fredholm IE [47] would be singular (spectral condition number of infinity) if only those singular interactions were omitted. Same is true for second kind Fredholm IEs [48], however for them the principal value [49] of those singular interactions can be evaluated analytically in a straightforward manner. It must be highlighted that sensitivity to errors in the evaluation of singular interactions is most pronounced on near-field simulation quantities such as current density, near-fields, input reactance, resonant frequencies, etc. Far-field quantities such as radiation pattern of radar cross-section are relatively stable to those errors and perturbations.

This proposed methodology can be used to enhance the performance a rather broad range of discrete IE methods in diverse scientific computing disciplines. However, it will only be presented in the context of high-frequency frequency-domain EM modeling of impenetrable perfect electric conducting (PEC) structures discretized by faceted, simplicial (triangular) meshes. The methodology likely would be effective in the evaluation of large collections of singular interactions of any discrete IE formulation that involves convolutions with translation-invariant kernels, when their evaluation by integration is slow, but the integral and domains can be parametrized with a reasonably small number of variables. The range of applicability spans, but is not limited to, the evaluation of singular interactions in discrete IEs for:

- Different modeling domains, such as time-domain IEs [50] and frequency domain IEs [3, 16]
- Different frequency regimes, i.e., high-frequency [3], static [51] and quasi-static [52].
- Different types of IEs, such as surface IEs (SIE) [21], volume IEs (VIE) [53]
- Different kinds of IE, such as the Fredholm IE of first kind [47], including the electric field integral equation (EFIE) [3, 16, 21] for impenetrable (PEC) objects and the PMCHWT [22–24] penetrable objects, as well as second kind IEs [48] such as the magnetic field IE (MFIE) [3, 21] for PEC objects and the Muller formulation for penetrable objects, and combinations thereof, e.g., the combined field IE (CFIE) [3, 21, 54].
- Vector and scalar IEs and their combinations such as current-charge formulations [55, 56].
- Different discretization schemes such as BEM or MoM [3, 16], collocation [57], and Nystrom [58].
- Different types and orders of basis functions, including composite ones such as loop-charge [59] or loop-star [56]. However, the method is best suitable for the lowest order simplicial meshes functions, as opposed to quadrilateral and higher-order basis functions, and partition of unity or Generalized BEM [60, 61].

- Different Green's functions such as layered media [62], periodic [63], cavity and WG [64] Green's functions. It is expected that performance would depend on the parametrization, thus Green's functions for complicated configurations may inhibit performance.

- Hybrid formulations between differential equation methods and IE methods such as finite element boundary integral FEBI [65] or finite differences and integral equations [66].

- Different disciplines where boundary value problems can be cast in form of IEs, such as mechanics, acoustics, quantum mechanics, etc.

As it will become clear in the following sections, the proposed method is not strictly speaking an singular integration technique. It can be considered as an *unsupervised machine learning* [67] methodology, that after completing an one-time training, it attempts to infer the singular integral results, without ever resorting to any integration but rather multidimensional interpolation. As such, the methodology would be applicable to any case where large collections of slow-to-evaluate integrals are necessary, and that could be well outside the area of boundary value problems and IEs. Example could include integral transforms, statistical inference, data-science, finance, etc.

1.4 Literature Review

Despite the broad reach of the proposed approach, this research will focus on the task of fast, error-controllable, and reliable computation of *large collections of singular interactions* encountered in the numerical solution of discrete IEs by the BEM or MoM. In this context, several methods have been proposed and widely used in commercial tools some of which are outlined in the following paragraphs.

1.4.1 Singularity Subtraction

The singularity subtraction/extraction technique is based on the power series representation of the Green's function and decomposition of the inner double integrals into two parts, [68]. The first part involves the even, non-smooth, singular terms of the power series representation, which are to be evaluated analytically. Hence, separating the singular terms of the first part from the second part, the latter can be made sufficiently smooth for Gaussian quadrature evaluation. Finally, the double outer integrals are also computed by numerical integration. This method is generally accurate, but poses a challenge when higher order basis functions are employed. To this end, in [37] the authors present a method of singularity subtraction, where the basis function and their surface divergences can be expressed as a linear combination of the polynomial shape functions on a triangle. In [69] the authors develop an accurate singularity subtraction technique with a focus on MFIE, while in [70] the authors perform an additional singularity extraction on the outer integral as well.

1.4.2 Singularity Cancellation

In the singularity cancellation technique the integral is treated by use of an optimal change of variables, such that all derivatives of the Green's function exponentially tend to zero for large values, [44, 45] and therefore can be compute via a numerical scheme. Cancellation methods are more recent and unlike subtraction methods are generally independent of basis function kind or order, integration domain shape or curvature and can achieve up to machine precision accuracy; although less efficient for low to medium accuracy levels, [71]. In [46] the authors present an extension of singularity cancellation, the direct evaluation method (DEM); which by a series of coordinate transformations and integration reordering, reduces the quadruple hyper-singular integrals to smooth double integrals, to be evaluated by numerical quadratures; ultimately leading to improved efficiency and up to machine precision accuracy. The method was expanded and generalized to the fully numerical methods (DIRECTFN) [72] to handle coincident, edge adjacent and vertex adjacent singular integrals on planar and curvilinear elements. In [39] the author presents a cancellation method for the near singular integrals of the gradient of the Green's functions (MFIE) using higher order basis functions, over curvilinear triangular elements with controllable accuracy, suitable for even highly skewed triangles. In [73] the author's expand on the Taylor-Duffy method, which considers a specific kernel and linear polynomial and reduces the singular four dimensional integral to a non-singular integral of smaller dimension.

1.4.3 Full 4D Evaluation

The full 4D integral evaluation is based on the application of the divergence theorem on both the inner and the outer double integrals, followed by variable transforms to smoothen the integrands, significantly improving the accuracy as well as the generality of the method with regards to kernels, basis, testing functions and quality of triangle shape. The method is robust and can be readily used with Gauss-Legendre or double exponential quadrature rules for reasonable to arbitrarily high precision, [43]. The approach was then extended beyond well-shaped and touching triangles in [36]. In [71] the authors briefly describe a trend in singular integration akin to [36,43], but state that it is still underdeveloped; thus they propose a singularity subtraction and singularity cancellation hybrid.

1.5 Research Opportunity

Despite the individual merits and drawbacks of each of the outlined methods, the general trend is that singular integrations are getting faster but more involved. After many years of research, perhaps the task of performing a singular interaction integration in MoM is reaching the point of diminishing returns. *The fundamental* *premise of this research is that to extract farther computational efficiencies in this problem, one must recast the singular integration problem at a higher, coarser-gain computational level.* That is, instead of seeking to improve the evaluation efficiency of a single integral, we should strive to improve the evaluation efficiency of a large collection of integrals all together. Conceptually, this is akin to what MLFMM does, i.e., instead of accelerating a single far-field interaction computation, computational efficiencies are sought among grouped interactions.

1.6 Proposed Approach

Strictly speaking, the proposed method is not a singular integration technique of the likes presented in section 1.4. It can be considered as an *unsupervised machine learning* [67] methodology for evaluating collections of singular integrals. To the best of the author's knowledge, this is a first of its kind for this type of application.

The approach relies on two steps. The first step is the one-time training stage that makes use of some of the conventional singular integration approaches outlined in Section 1.4. The result of this stage is a *universal library* of moderate storage footprint, that is to be used during the second step that takes place every time a new MoM matrix setup is invoked (online stage). This universal library is frequency independent and must be capable of capturing every possible singular interaction arrangement, for flat triangles, and any background medium. There are a lot of nuances the go into the generation of this library as will be described in the following chapters, but the most important aspect is maintaining extremely high accuracy and low storage footprint; speed of evaluation is not a major concern because the library is generated only once (off-line stage). The key ingredient in the training stage is to identify suitable high-dimensional parametrizations of the singular integral interactions that are general and lead to a highly accurate, low footprint library. In this work multiple libraries are generated to account for the various singular integral interactions encountered in EFIE, MFIE, CFIE and PMCHWT formulations. Those are:

- *EFIE common triangle* interactions i.e., when source and receiver triangles coincide (fully overlap), and the Green's function has one over distance behavior. Strictly speaking, integrals in this interaction are weakly singular as the Jacobian cancels out the singularity, however numerically even these interactions can hinder efficiency and speed.

- *EFIE common edge* interactions i.e., when source and receiver triangles are touching only along an edge, and the Green's function has one over distance behavior. Again, this is a weakly singular integral that nonetheless is problematic to evaluate numerically.

- *MFIE common edge* interactions i.e., when source and receiver triangles touching only along an edge, and the Green's function has one over distance square behavior.

During the online evaluation, any singular integral in the MoM matrix belonging to one of these three categories is evaluated by interpolating the pre-evaluated interaction integrals in the corresponding library. Advanced multidimensional interpolation methods such as sparse grids [74–76], tensor Tucker decompositions [77,78] and tensor train (TT) decompositions [78] are used to infer the values of any integral expediently and accurately from the limited values stored in the universal library. In other words, *the singular integration problem is converted into a multidimensional interpolation problem*, justifying the name Singular Integration By Interpolation (SIBI). Due to the relatively lower complexity of interpolation, as opposed to that of state-of-the-art singularity subtraction, [37], especially in conjunction with the efficient aforementioned algorithms, the proposed approach was found to be orders of magnitude more accurate and faster than said state-of-the-art in the cases presented herein.

1.7 Contributions

The proposed approach offers an unconventional route towards the singular integration problem in integral equations. Consequently, it has contributed to the problem in the following terms:

- The problem is recast from an integration problem into an interpolation problem. Consequently, it offers a new perspective/paradigm to the problem and paves the way towards new methodologies. Namely resolving *the-curse-dimensionality*, which as is discussed in section 4 and is also evident in 5, is the main issue of the proposed method.

- The proposed approach does not replace, but rather can be paired with any state-of-the-art method towards the library construction of the *off-line stage*. In fact, our research showed that a library of integrals with a higher accuracy positively affected the smoothness of the tabulated functions themselves across the configuration domain. Hence, a machine precision level of accuracy library would likely yield an impressive impact on the mesh performance of the proposed approach.

- The method is general and reliable in terms of kernel, singularity, element shape and quality, whereas most methods in the literature have a narrower application spectrum and are specifically engineered towards that.

- The computation of singular interaction integrals is one of the most complex algorithms to implement in software; an interpolation code on the other hand favors simplicity. Although a singular integration engine is required for the tabulation process (*off-line stage*) the code itself is not.

- The tables could be specifically engineered to particular applications for optimal performance. For example in planar multilayer geometries, the subtended angle between elements is eliminated a a degree of freedom, which would alleviate the *curse-of-dimensionality* improve the performance.

1.8 Thesis Outline

The remaining chapters of this thesis are organized as follows:

In chapter 2 the *off-line stage* is explained, in which said *universal integral libraries* are built. Keep in mind that during the *offline stage* computation time is of little importance; thus the integral calculation for the libraries is purposefully done as accurately as possible, with no regards to computation time. Consequently, the recovery of integrals of any arbitrarily shaped triangle pair during the *online stage* is much faster than the state-of-the-art, [79]! The first part of the approach begins with the parameterization of the geometry of the integration domain, i.e., the source and receiver triangle pair; meaning the BEM element matrices integral expressions are seen as functions of the wavenumber and the triangle pair, the latter of which is consequently substituted with an equivalent series of angles and lengths. In the case of EFIE, equivalent smoother integral expressions based on the scalar nodal shape functions, as opposed to vectorial RWGs, are derived. The next step is the derivation of frequency independent integral expressions for the BEM element matrices, which facilitates the building of universal integral libraries. Following is the introduction and study of the space of these functions, the "configuration" domain, which is transformed from a $3D$ prism/ $6D$ hyper-prism (common triangle/edge case) to a $3D$/ $6D$ rectangle/hyper-rectangle, by means of a Duffy-like [80] transformation, in order to enable high dimensional interpolation schemes, necessary for the common edge case ($6D$ interpolation). There is a different domain for the common triangle and common edge case. Sampling the transformed domain, i.e., computing by use of a state-of-the-art method said frequency independent expressions for various common triangle, edge geometries creates a universal library of integrals and completes the *off-line stage*. In this work, a high accuracy (166^2 quadrature points) third order singularity subtraction rule is used, [37]. The second part, the *online stage*, is explained in chapter 3 and begins with the analysis of the sparse grid interpolation, which involves the grid construction using Smolyak's algorithm and the interpolation utilizing hierarchical basis functions and the barycentric formula. The grid is actually required to be constructed for the *off-line stage* as well, to indicate the locations of the samples in the domain. Due to the higher dimensionality of the interpolation in the common edge case the performance of the proposed approach deteriorates, therefore an alternate implementation is sought out, [81]. Namely, the tensor decomposition approach utilizes a higher order full tensor product grid rather than a sparse one to increase accuracy and deploys a tensor decomposition to compensate for the loss of computational speed, explained in chapter 4. In chapter 5 the proposed approach is compared to the state-of-the-art technique, which was used for the *off-line stage*, in terms of accuracy and efficiency (memory, computational cost). Results are presented for the collection of singular interactions within the BEM matrix of PEC flat triangular meshes as well as the Radar Cross Section (RCS) and surface currents given by the BEM solution of a sphere scattering problem.

CHAPTER 2

SINGULAR INTEGRATION BY INTERPOLATION

In this chapter the main principles of the proposed Singular Integration by Interpolation (SIBI) are explained. The BEM element matrix expressions for the electric and magnetic field integral equations (EFIE, MFIE) are manipulated such that frequency and material independent but geometrical parametric tabulation of the singular integrals is feasible. Namely, equivalent integral expressions are derived, that are sufficiently smooth and are defined in an appropriate 'parametrization domain' such that they be suitable for efficient high-dimensional interpolation,

2.1 BEM Element Matrix Definitions

The conventional integral expressions that lead to the assembly of the impedance matrix element entries of the BEM for the Electric Field Integral Equation (EFIE) and Magnetic Field Integral Equation (MFIE) in BEM (aka MoM) for perfect electric conducting (PEC) structures embedded in unbounded and homogeneous background media are [3, 82]:

$$\begin{aligned}\mathbf{Z}_{mn}^{(e)\;\mathcal{L}} = & \int_T \boldsymbol{\alpha}_m^t(\mathbf{r}) \cdot \int_{T'} \boldsymbol{\alpha}_n(\mathbf{r}') \frac{e^{-jk|\mathbf{r}-\mathbf{r}'|}}{4\pi|\mathbf{r}-\mathbf{r}'|} d\mathbf{r}'^2 d\mathbf{r}^2 \\ & -\frac{1}{k^2}\int_T \nabla_\Gamma \cdot \boldsymbol{\alpha}_m^t(\mathbf{r}) \int_{T'} \nabla'_\Gamma \cdot \boldsymbol{\alpha}_n(\mathbf{r}') \frac{e^{-jk|\mathbf{r}-\mathbf{r}'|}}{4\pi|\mathbf{r}-\mathbf{r}'|} d\mathbf{r}'^2 d\mathbf{r}^2, \quad m,n=1,2,3 \qquad (2.1)\end{aligned}$$

$$\begin{aligned}\mathbf{Z}_{mn}^{(e)\;\mathcal{K}} = & \frac{1}{2}\int_T \boldsymbol{\alpha}_m^t(\mathbf{r}) \cdot \hat{\boldsymbol{n}} \times \boldsymbol{\alpha}_n(\mathbf{r}) d\mathbf{r}^2 \\ & + \int_T \fint_{T'} \nabla' \frac{e^{-jk|\mathbf{r}-\mathbf{r}'|}}{4\pi|\mathbf{r}-\mathbf{r}'|} \cdot \left(\boldsymbol{\alpha}_m^t(\mathbf{r}) \times \boldsymbol{\alpha}_n(\mathbf{r}')\right) d\mathbf{r}'^2 d\mathbf{r}^2, \quad m,n=1,2,3 \qquad (2.2)\end{aligned}$$

where T and T' is a pair of observation and source flat elements (triangles in this dissertation), $\mathbf{r}$ and $\mathbf{r}'$ are the observation and source coordinates, $\boldsymbol{\alpha}_m^t(\mathbf{r})$ and $\boldsymbol{\alpha}_n(\mathbf{r}')$ are the divergence conforming current test and trial vector basis function [82] (Rao-Wilton-Glisson (RWG) in this disseratation), $\hat{\boldsymbol{n}}$ the unit vector normal to the observation triangle surface and k the wavenumber. The $\mathbf{Z}_{mn}^{(e)}$ in (2.1) and (2.2) signify that these are the element matrices, i.e., interaction between a single source and observation pair, and are not fully assembled into the full impedance matrix as one matrix entry would involve the summation of multiple such interactions doe to the fact that each trial and test function is defined on two mesh elements (triangles). It is noted that although (2.1) and (2.2) are presented in this dissertation in the context of EFIE and MFIE for perfect electric conducting (PEC) structures, the same integral kernels $\mathcal{L}$ and $\mathcal{K}$ and convolution integrals are present in IEs for dielectric structures e.g. PMCHWT [22–24].

Because the fist term in (2.2) is not singular it is advisable to be integrated separately either analytical or conventional Gauss quadratures. The second term of (2.2) and is zero when the source and observation elements are the same (overlap) and the elements/patches are flat, thus does pose a challenge. However, when the source and observation elements touch at an edge or a node, and the triangles are not coplanar, it leads to a near-singular integration that is particularly problematic. For those cases we will only focus on this singular integral defined as:

$$\mathbf{Z}_{mn}^{(e)\ \mathcal{K}_{pv}} = \int_{T} \fint_{T'} \nabla' \frac{e^{-jk|\mathbf{r}-\mathbf{r}'|}}{4\pi|\mathbf{r}-\mathbf{r}'|} \cdot \left(\boldsymbol{\alpha}_m^t(\mathbf{r}) \times \boldsymbol{\alpha}_n(\mathbf{r}')\right) d\mathbf{r}'^2 d\mathbf{r}^2 \tag{2.3}$$

The aim of this dissertation is to numerically evaluate these integrals accurately, reliably and efficiently for all the cases in a mesh when the source and observation elements T and T' are overlapping or touching at an edge.

For the sake of brevity, evaluation of (2.1) and (2.2) will be presented only for RWG basis functions as they are the most versitile and popular option. The various vector conventions for the definition of RWG basis functions are shown in Fig. 2.1. Using those vectors each RWG basis can be written in terms of the barycentric coordinate ζ gradients as:

$$\begin{aligned}
\boldsymbol{w}_0 &= \zeta_1 \nabla \zeta_2 - \zeta_2 \nabla \zeta_1 && (2.4)\\
\boldsymbol{w}_1 &= \zeta_0 \nabla \zeta_2 - \zeta_2 \nabla \zeta_0 && (2.5)\\
\boldsymbol{w}_2 &= \zeta_0 \nabla \zeta_1 - \zeta_1 \nabla \zeta_0 && (2.6)\\
\nabla \zeta_j &= \frac{\hat{\boldsymbol{h}}_j \ell_j}{2A} = \frac{\hat{\boldsymbol{h}}_j}{h_j} && (2.7)\\
\boldsymbol{\alpha}_0 &= \hat{\boldsymbol{n}} \times \boldsymbol{w}_0 = \frac{1}{2A}\left(\zeta_1 \boldsymbol{\ell}_2 - \zeta_2 \boldsymbol{\ell}_1\right) && (2.8)\\
\boldsymbol{\alpha}_1 &= \hat{\boldsymbol{n}} \times \boldsymbol{w}_1 = \frac{1}{2A}\left(\zeta_2 \boldsymbol{\ell}_0 - \zeta_0 \boldsymbol{\ell}_2\right) && (2.9)\\
\boldsymbol{\alpha}_2 &= \hat{\boldsymbol{n}} \times \boldsymbol{w}_2 = \frac{1}{2A}\left(\zeta_0 \boldsymbol{\ell}_1 - \zeta_1 \boldsymbol{\ell}_0\right) && (2.10)
\end{aligned}$$

where A is the triangle area, ζ the barycentric coordinates and $\hat{\boldsymbol{h}}, \boldsymbol{\ell}$ are the height, edge vectors of the triangle. It is noted that (2.5)-(2.7), are the tangentially continuous 'edge' element used in finite elements and are related with RWGs thought the fist part of equations (2.8)-(2.10).

2.1.1 Scalar Element Integral Terms

Although the proposed approach can be directly formulated using (2.1) and (2.3), it was found that for the case of the L operator of the EFIE to express the singular integral element matrix in terms of integrals over scalar basis functions (see appendix A). These expressions were found to be smoother than their equivalent, vectorial RWG ones and thus are desirable for the proposed method. In the case of EFIE the BEM element matrix integrals in (2.1), can easily be expressed as integrals where the basis functions are scalar rather than vectorial. These expressions were found

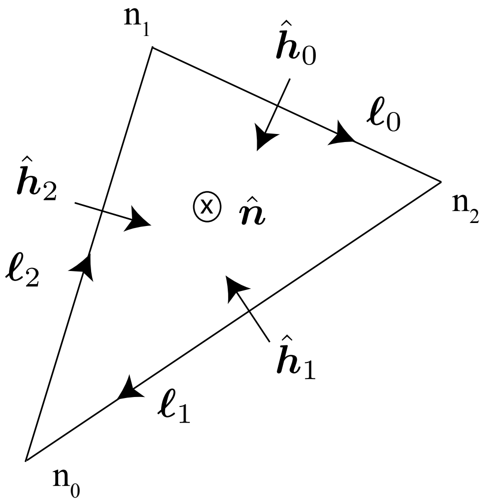


Figure 2.1. Various vector quantities and orientations used for the definition of the RWG basis functions in this dissertation.

to be smoother than their equivalent, vectorial RWG and thus are desirable for the proposed method. Substituting the basis functions $\boldsymbol{\alpha}^t, \boldsymbol{\alpha}$ and using $\zeta_0 + \zeta_1 + \zeta_2 = 1$ we get:

$$\mathbf{Z}_{mn}^{(e)\ \mathcal{L}} = \frac{1}{4A^2} \sum_{\substack{p=0 \\ p \neq m}}^{2} \sum_{\substack{q=0 \\ q \neq n}}^{2} c_{pq} \boldsymbol{\mathcal{I}}_{pq} - \frac{d_{mn}}{k^2} \sum_{p=0}^{2} \sum_{q=0}^{2} \boldsymbol{\mathcal{I}}_{pq}$$

where c_{pq}, d_{mn} are constants dependent on the heights and lengths of the triangles T, T', which are given in the Appendix and

$$\boldsymbol{\mathcal{I}}_{pq} = \int_T \zeta_p(\mathbf{r}) \int_{T'} \zeta_q(\mathbf{r}') \frac{e^{-jk|\mathbf{r}-\mathbf{r}'|}}{4\pi|\mathbf{r}-\mathbf{r}'|} d\mathbf{r}'^2 d\mathbf{r}^2 \tag{2.11}$$

where $\zeta_p(\mathbf{r}), \zeta_q(\mathbf{r}')$ are the barycentric coordinates of the triangles T, T' respectively.

2.2 Integral Parametrization

The key difference of the proposed approach to conventional singular integration approaches is that we seek to create a (behavioral) model of $\mathbf{Z}^{K_{pv}}$ in (2.3) and $\mathbf{I}$ in

(2.11). That is we seek to create a function of $\mathbf{Z}^{K_{pv}}$ and $\mathbf{I}$ in terms of some geometrical parameters that represent an element matrix interaction between source and observation triangles. Because those integrals also involve the wavenumber (material and frequency) they can be expressed in terms of k, T and T' as:

$$\boldsymbol{\mathcal{I}}_{pq}(k,T,T') = \int_T \zeta_p(\mathbf{r}) \int_{T'} \zeta_q(\mathbf{r}') \frac{e^{-jk|\mathbf{r}-\mathbf{r}'|}}{4\pi|\mathbf{r}-\mathbf{r}'|} d\mathbf{r}'^2 d\mathbf{r}^2 \tag{2.12}$$

$$\mathbf{Z}^{(e)\,\mathcal{K}_{pv}}_{mn}(k,T,T') = \int_T \fint_{T'} \nabla' \frac{e^{-jk|\mathbf{r}-\mathbf{r}'|}}{4\pi|\mathbf{r}-\mathbf{r}'|} \cdot \left(\boldsymbol{\alpha}^t_m(\mathbf{r}) \times \boldsymbol{\alpha}_n(\mathbf{r}')\right) d\mathbf{r}'^2 d\mathbf{r}^2 \tag{2.13}$$

where T and T' are some *parallelizations* that represent geometry of the observation and source triangles, respectively. The proposed approach deals with two of the most common and important of source-observation triangle interactions; when the triangles coincide, termed *common triangle* interaction or share an edge, termed *common edge* interaction. The geometries of these interactions are shown in Fig. 2.2. It is noted that the third singular case, that of common vertex, is not considered herein, primely because the parametrization has high dimension, and conventional approaches, even carefully designed Gaussian quadrature, perform reasonably well. The common triangle case in Fig. 2.2a the integral I can be parametrized as:

$$\boldsymbol{\mathcal{I}}_{pq}(k,T,T') \equiv \boldsymbol{\mathcal{I}}_{pq}(k,T) \equiv \boldsymbol{\mathcal{I}}_{pq}(k,\ell,\phi_1,\phi_2) \tag{2.14}$$

where *ell* is the length of any triangle edge and ϕ_1 and ϕ_2 are the two angles incident on that edge. The integrals in the common edge case ($\partial T' \cup \partial T \neq 0$), similarly can be parametrized in terms of the wavenumber k, the angle ϕ_5 between the common edge triangle pair, the length of the common edge ℓ and the four angles formed by the common edge and the four other triangle edges denoted as ϕ_1, ϕ_2, ϕ_3

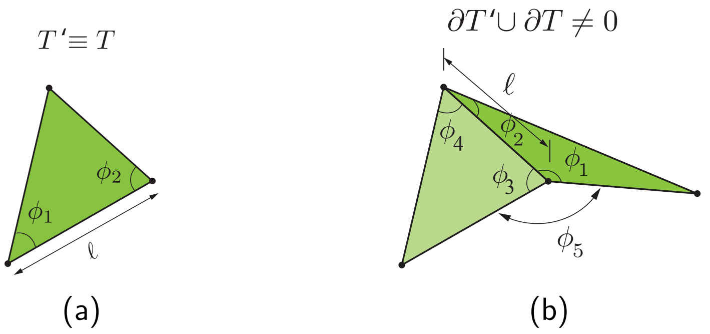


Figure 2.2. Geometry parametrization for the a) common triangle and b) common edge singular cases.

and ϕ_4, as shown in Fig. 2.2b. In the common edge case the parametrization is written as:

$$\boldsymbol{\mathcal{I}}_{pq}(k, T, T') \equiv \boldsymbol{\mathcal{I}}_{pq}(k, \ell, \phi_1, \phi_2, \phi_3, \phi_4, \phi_5) \tag{2.15}$$

$$\mathbf{Z}_{mn}^{(e)\ \mathcal{K}_{pv}}(k, T, T') \equiv \mathbf{Z}_{mn}^{(e)\ \mathcal{K}_{pv}}(k, \ell, \phi_1, \phi_2, \phi_3, \phi_4, \phi_5) \tag{2.16}$$

for the EFIE and MFIE, respectively.

2.3 Wavenumber Independent Parametrization

The other key enabler to the proposed approach is the existence of a wavenumber independent parametrization for the integrals in (2.12-2.13). Hence, the following derivations assume non-dispersive media. Although the approach can be expanded to such media as an extra parameter at the expense of an extra parametrization dimension, it will not be covered in this dissertation. The frequency independent integrals are derived below for the cases of EFIE common triangle, EFIE common edge and MFIE common edge.

2.3.1 EFIE common triangle

The key principle in this derivation is to transform all the quantities involved in (2.11) in terms of angles, that is, lengths and areas must be expressed in electrical lengths and electrial areas respectively. This can be achieved by applying the following change of variables to the (2.12) :

$$\tilde{x} = kx, \quad \tilde{y} = ky, \quad \tilde{z} = kz, \quad \Rightarrow \tilde{\mathbf{r}} = k\mathbf{r}, \quad \tilde{\mathbf{r}}' = k\mathbf{r}' \tag{2.17}$$

The integral can be rewritten as:

$$\begin{aligned}\boldsymbol{\mathcal{I}}_{pq}(k, \ell, \phi_1, \phi_2) &= \int_{\tilde{T}} \zeta_p(\tilde{\mathbf{r}}) \int_{\tilde{T}} \zeta_q(\tilde{\mathbf{r}}') k \frac{e^{-j|\tilde{\mathbf{r}}-\tilde{\mathbf{r}}'|}}{4\pi|\tilde{\mathbf{r}} - \tilde{\mathbf{r}}'|} \frac{1}{k^2} d\tilde{\mathbf{r}}'^2 \frac{1}{k^2} d\tilde{\mathbf{r}}^2 \\ &= k^{-3} \tilde{\boldsymbol{\mathcal{I}}}_{pq}(\phi_0, \phi_1, \phi_2) \end{aligned} \tag{2.18}$$

where $\phi_0 = k\ell$ is the electrical length of one of the edges of the scaled parametrized common triangle $\tilde{T}$, as shown in Fig. 2.3a; $\tilde{\mathbf{r}}, \tilde{\mathbf{r}}'$ are the observation and source coordinates and $\zeta_p(\tilde{\mathbf{r}}), \zeta_q(\tilde{\mathbf{r}}')$ are the barycentric coordinates of $\tilde{T}$ and the integral $\tilde{\boldsymbol{\mathcal{I}}}_{pq}$ is given as:

$$\tilde{\boldsymbol{\mathcal{I}}}_{pq}(\phi_0, \phi_1, \phi_2) = \int_{\tilde{T}} \zeta_p(\tilde{\mathbf{r}}) \int_{\tilde{T}} \zeta_q(\tilde{\mathbf{r}}') \frac{e^{-j|\tilde{\mathbf{r}}-\tilde{\mathbf{r}}'|}}{4\pi|\tilde{\mathbf{r}} - \tilde{\mathbf{r}}'|} d\tilde{\mathbf{r}}'^2 d\tilde{\mathbf{r}}^2 \tag{2.19}$$

This integral is evaluated over the scaled common triangle $\tilde{T}$, and is independent of the wavenumber (frequency) and is related to the original integral through:

$$\boldsymbol{\mathcal{I}}_{pq}(k, \ell, \phi_1, \phi_2) = k^{-3} \tilde{\boldsymbol{\mathcal{I}}}_{pq}(\phi_0, \phi_1, \phi_2) \tag{2.20}$$

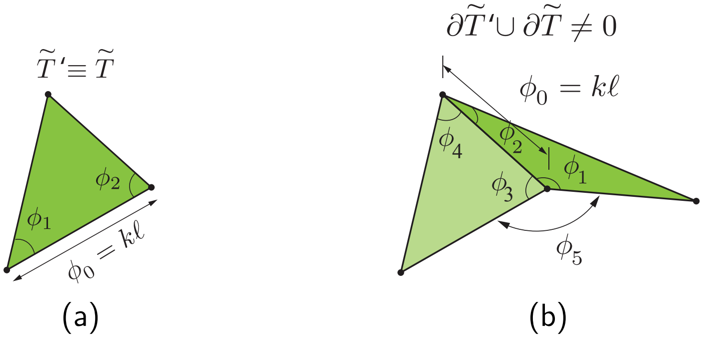

$\widetilde{T}' \equiv \widetilde{T}$
ϕ_2
ϕ_1
$\phi_0 = k\ell$
(a)
$\partial\widetilde{T}' \cup \partial\widetilde{T} \neq 0$
$\phi_0 = k\ell$
ϕ_4
ϕ_2
ϕ_1
ϕ_3
ϕ_5
(b)

Figure 2.3. Scaled geometry parametrization for the a) common triangle and b) common edge singular cases.

2.3.2 EFIE common edge

The process for the common edge case is identical to the common triangle. The wavenumber (frequency) independent integral is given as:

$$\tilde{\boldsymbol{\mathcal{I}}}_{pq}(\phi_0,\phi_1,\phi_2,\phi_3,\phi_4,\phi_5) = \int_{\tilde{T}} \zeta_p(\tilde{\mathbf{r}}) \int_{\tilde{T}'} \zeta_q(\tilde{\mathbf{r}}') \frac{e^{-j|\tilde{\mathbf{r}}-\tilde{\mathbf{r}}'|}}{4\pi|\tilde{\mathbf{r}}-\tilde{\mathbf{r}}'|} d\tilde{\mathbf{r}}'^2 d\tilde{\mathbf{r}}^2 \tag{2.21}$$

This integral is evaluated over the scaled common edge triangle pair $\tilde{T}, \tilde{T}'$, shown in Fig. 2.3b and is related to the original integral through:

$$\boldsymbol{\mathcal{I}}_{pq}(k,\ell,\phi_1,\phi_2,\phi_3,\phi_4,\phi_5) = k^{-3}\tilde{\boldsymbol{\mathcal{I}}}_{pq}(\phi_0,\phi_1,\phi_2,\phi_3,\phi_4,\phi_5) \tag{2.22}$$

2.3.3 MFIE common edge

The same change of variables in (2.17), is applied to (2.13) for the MFIE common edge case. The wavenumber independent integral is given by:

$$\tilde{\mathbf{Z}}_{mn}^{(e)\ \mathcal{K}_{pv}}(\phi_0,\phi_1,\cdots,\phi_5) = \int_{\tilde{T}} \fint_{\tilde{T}'} \tilde{\nabla}' \frac{e^{-j|\tilde{\mathbf{r}}-\tilde{\mathbf{r}}'|}}{4\pi|\tilde{\mathbf{r}}-\tilde{\mathbf{r}}'|} \cdot \Big(\boldsymbol{\alpha}_m^t(\tilde{\mathbf{r}}) \times \boldsymbol{\alpha}_n(\tilde{\mathbf{r}}')\Big) d\tilde{\mathbf{r}}'^2 d\tilde{\mathbf{r}}^2 \tag{2.23}$$

Where $⨍$ denotes the evaluation in the Cauchy integral sense, [49]. The above expression is associated with the original integral through:

$$\mathbf{Z}_{mn}^{(e)\ \mathcal{K}_{pv}}(k, \ell, \phi_1, \phi_2, \phi_3, \phi_4, \phi_5) \quad = \quad \tilde{\mathbf{Z}}_{mn}^{(e)\ \mathcal{K}_{pv}}(\phi_0, \phi_1, \phi_2, \phi_3, \phi_4, \phi_5) \tag{2.24}$$

The frequency (and/or materials) independent integrals in (2.19,2.21,2.23) are functions only of angles and electrical lengths in the geometry of the triangles as shown in Fig. 2.3 and thus can be tabulated in a universal library of integrals *once-and-for-all*. The original integrals in (2.12-2.13) can be retrieved from the frequency independent ones in the library at any time, using the (2.20,2.22,2.24).

It was found that a slightly different normalization for the EFIE cases may be more beneficial, as it leads to smoother models, the normalizations are:

$$\tilde{\boldsymbol{\mathcal{I}}}_{pq}^{n}(\phi_0, \phi_1, \phi_2) = \frac{\tilde{\boldsymbol{\mathcal{I}}}_{pq}(\phi_0, \phi_1, \phi_2)}{\tilde{A}_s^{1.5}} \tag{2.25}$$

$$\tilde{\boldsymbol{\mathcal{I}}}_{pq}^{n}(\phi_0, \phi_1, \phi_2, \phi_3, \phi_4, \phi_5) = \frac{\tilde{\boldsymbol{\mathcal{I}}}_{pq}(\phi_0, \phi_1, \phi_2, \phi_3, \phi_4, \phi_5)}{\tilde{A}_s^{0.75}\tilde{A}_r^{0.75}} \tag{2.26}$$

where the superscript "n" indicates normalized integral model, and $\tilde{A}_s, \tilde{A}_r$: are the areas of source, receiver triangles $\tilde{T}'$ and $\tilde{T}$ respectively. We note that $\tilde{A}_s = k^2 A_s and \tilde{A}_r = k^2 A_r$, thus:

$$\frac{A_s^{0.75} A_r^{0.75}}{\tilde{A}_s^{0.75}\tilde{A}_r^{0.75}} = k^{-3}$$

Finally the EFIE integrals $\tilde{\boldsymbol{\mathcal{I}}}_{pq}$ in (2.19,2.21) are replaced with $\tilde{\boldsymbol{\mathcal{I}}}^{n}_{pq}$, can be written in terms of the normalized ones as:

$$\boldsymbol{\mathcal{I}}_{pq}(k, \ell, \phi_1, \phi_2) = A_s^{1.5}\tilde{\boldsymbol{\mathcal{I}}}^{n}_{pq}(\phi_0, \phi_1, \phi_2) \tag{2.27}$$

$$\boldsymbol{\mathcal{I}}_{pq}(k, \ell, \phi_1, \phi_2, \phi_3, \phi_4, \phi_5) = A_s^{0.75}A_r^{0.75}\tilde{\boldsymbol{\mathcal{I}}}^{n}_{pq}(\phi_0, \phi_1, \phi_2, \phi_3, \phi_4, \phi_5) \tag{2.28}$$

2.4 Parametrization Domain

The angle ϕ_1, ϕ_2 and ϕ_3 in the common triangle case and $\phi_1 - \phi_6$ in the common edge cases define a three-dimensional and six-dimensional parametrization domains respectively. Because of geometrical restrictions, those domains are prisms and hyperprisms respectively, as the following on the ϕ variables apply:

$$\phi_0, \phi_1, \phi_2, \phi_3, \phi_4, \phi_5 > 0^\circ \tag{2.29}$$

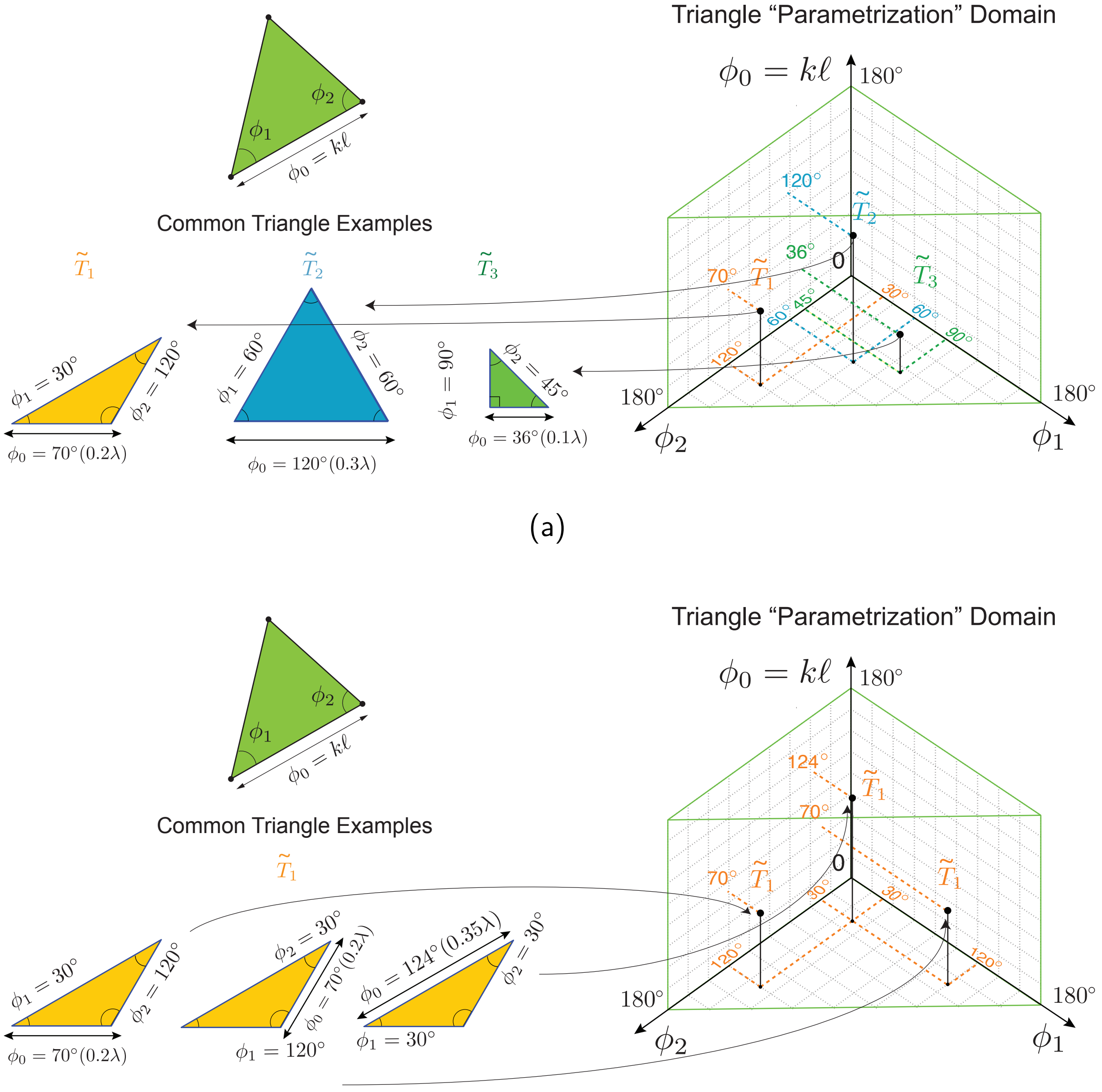
$$\phi_5 \leq 180^\circ \tag{2.30}$$

$$\phi_1 + \phi_2 < 180^\circ \tag{2.31}$$

$$\phi_3 + \phi_4 < 180^\circ \tag{2.32}$$

In theory ϕ_0, the electrical length of a triangle's edge, can be infinitely large; however in practice, due to approximation reasons, it is usually bounded to electrical lengths of half-wavelength thus the bound for $\phi_0 < 180^\circ$.

The parametrization domain for the common triangle case is shown in Fig. 2.4a, where each point in the domain corresponds to a scaled common triangle geometry. Fig. 2.4b suggests that further symmetries can be leveraged to reduce the parametrization volume, thus lowering universal table memory and the computation effort at the interpolation stage.



(a)

(b)

Figure 2.4. The "parametrization" domain for the common triangle case. a) Each point inside the parametrization domain corresponds to a particular scaled common triangle geometry. b) Symmetries in the parametrization domain, each scaled common triangle geometry can be mapped to three points.

2.4.1 Smoothness of Integral Models I (2.25, 2.26) and Z (2.23)

As eluted in the introduction, we will attempt to construct a model for those integrals by sampling i.e. evaluating the singular integrals once and for all, at various carefully chosen situations i.e. geometrical configurations. Prior to sampling the integral functions of (2.25,2.26,2.23) it is necessary to the have a rough idea about the degree of smoothness of the underlying integral models we attempt to sample (tabulate) and then interpolate. They would guide us in choosing a sampling strategy and interpolation method. Fig. 2.5 plots the distribution of the magnitude (in logarithmic scale) of the scaled and normalized integral $\tilde{\boldsymbol{\mathcal{I}}}_{pq}^{n}$ for the EFIE common triangle case. The distribution is plotted along a 2D slice of the parameter domain that represent all possible triangles with triangle edge length one tenth of a wavelength or $\phi_0 = 36°$. It is emphasized that each point on this graph corresponds to a particular common triangle geometry. For example the point $(30, 120)$ corresponds to a triangle that has an edge of electrical length 36° and two adjacent angles of 30° and 120°. Notice the plane of symmetry along the $\phi_1 = \phi_2$ line; that is because triangles such as $(30, 120)$ and $(120, 30)$ are identical and thus their singular integral interaction are same. This symmetry will later on prove essential in building low footprint libraries for the tabulated models. The magnitude of the normalized integral is close to zero along the edges of the plot and close to its maximum along the hypotenuse, where the triangles are highly skewed and appear to be sliver triangles. Not surprisingly, the integrals of these poor mesh-quality triangles will prove to be very challenging to compute accurately. Nonetheless the function is generally very smooth, which is important so that the sampling of the domain need not be very fine, helping with computational efficiency and library footprint. Similarly, in Fig. 2.6 the magnitude distribution of the EFIE common edge $\tilde{\boldsymbol{\mathcal{I}}}_{pq}^{n}$ from (2.26) is plotted along the same parameter space cut plane, i.e., fixing the electrical length of the common edge to $\phi_0 = 36°$, the subtended angle to $\phi_5 = 90°$ and the adjacent to the common edge angles of the source triangle to $\phi_3 = \phi_4 = 45°$, while varying the adjacent to the common edge angles of the receiver triangle ϕ_1, ϕ_2. Each point on this plot corresponds to a particular common edge triangle pair, where only the shape of the observation triangle changes. Again the function is generally smooth across the distribution, but becomes less smooth at

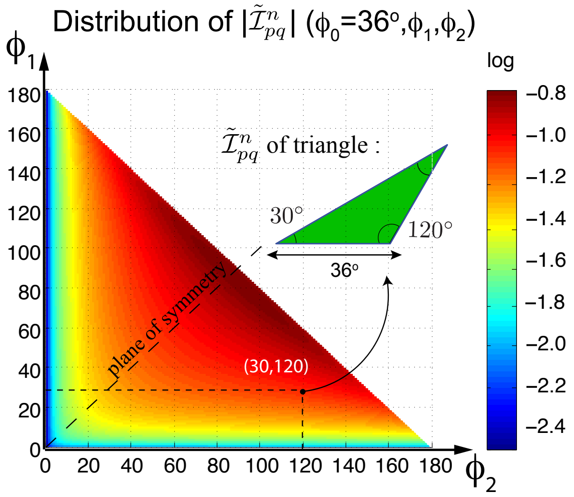
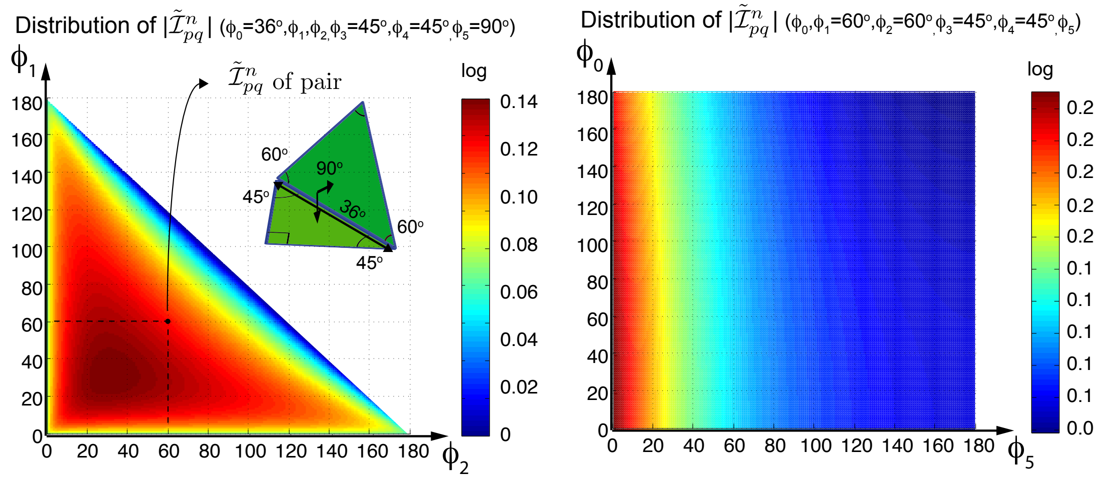


Figure 2.5. Smoothness of the tabulated integral function $\tilde{\boldsymbol{\mathcal{I}}}^n_{pq}$, (2.25) for the EFIE common triangle case; slice along $\phi_0 = 36°$ (edge $h = \lambda/10$). The plot shows the magnitude of normalized and scaled integral in logarithmic color-scale, showing that is quite smooth and can be interpolated with relative ease.

the edges. Moreover, there is again a symmetry plane along the line of $\phi_1 = \phi_2$. It is interesting to note that the distribution for $\mathbf{Z}^{(e)\ \mathcal{K}_{pv}}_{mn}$ (MFIE) from (2.23), which is shown in Fig. 2.7 goes to zero on the symmetry plane.

2.4.2 Geometrical Symmetries

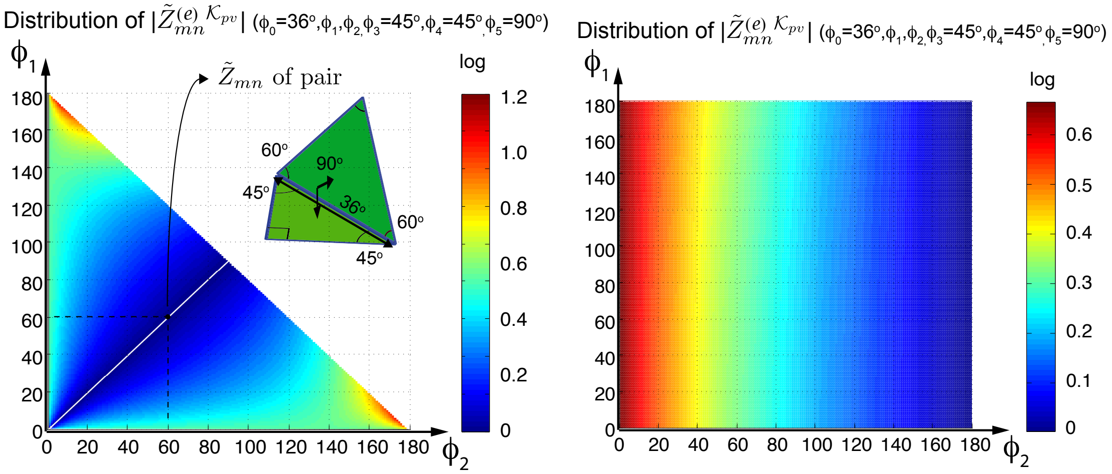
As seen in Fig. 2.4b, a common triangle, geometrical configuration can be represented by multiple points in the parametrized domain. Thus for each 3×3 scalar integral $\tilde{\boldsymbol{\mathcal{I}}}^n$ or vector $\tilde{\mathbf{Z}}^{(e)\ \mathcal{K}_{pv}}$ element matrix not all 9 entries (integrals) need be stored, leading to a significant reduction of library footprint and by extension memory and computation time savings. As it will be detailed below, for common triangle EFIE case, and the common edge EFIE and MFIE $2, 5, 4$ integrals are tabulated, respectively, reducing the size of the library by $22.2\%, 55.5\%, 44.4\%$ respectively. To demonstrate that, consider the node and edge numbering in Fig. 2.8 and the various equivalent parameter permutations, as shown in Fig. 2.9. Using these conventions the element matrices for EFIE common triangle can be written as:



(a)

(b)

Figure 2.6. Smoothness of the tabulated integral function $\tilde{\boldsymbol{\mathcal{I}}}^n_{pq}$, (2.26) for the EFIE common edge case; a) slice along $\phi_0 = 36°, \phi_3 = 45°, \phi_4 = 45°, \phi_5 = 90°$ b) slice along $\phi_1 = 60°, \phi_2 = 60°, \phi_3 = 45°, \phi_4 = 45°$. The plot shows the magnitude of normalized and scaled integral in logarithmic color-scale, showing that is quite smooth and can be interpolated with relative ease.



(a)

(b)

Figure 2.7. Smoothness plot of the tabulated integral function $\mathbf{Z}^{(e)\,\mathcal{K}_{pv}}_{mn}$, (2.23) for the MFIE common edge case; a) slice along $\phi_0 = 36°, \phi_3 = 45°, \phi_4 = 45°, \phi_5 = 90°$ b) slice along $\phi_1 = 60°, \phi_2 = 30°, \phi_3 = 45°, \phi_4 = 60°$. The plot shows the magnitude of scaled integral in logarithmic color-scale, showing that is quite smooth and can be interpolated with relative ease.

$$\tilde{\boldsymbol{\mathcal{I}}}^n_{CT} = \begin{bmatrix} \tilde{\mathcal{I}}^n_{0'0}(\tilde{\boldsymbol{\phi}}_1) & \tilde{\mathcal{I}}^n_{0'1}(\tilde{\boldsymbol{\phi}}_1) & \tilde{\mathcal{I}}^n_{0'2}(\tilde{\boldsymbol{\phi}}_1) \\ \tilde{\mathcal{I}}^n_{1'0}(\tilde{\boldsymbol{\phi}}_1) & \tilde{\mathcal{I}}^n_{1'1}(\tilde{\boldsymbol{\phi}}_1) & \tilde{\mathcal{I}}^n_{1'2}(\tilde{\boldsymbol{\phi}}_1) \\ \tilde{\mathcal{I}}^n_{2'0}(\tilde{\boldsymbol{\phi}}_1) & \tilde{\mathcal{I}}^n_{2'1}(\tilde{\boldsymbol{\phi}}_1) & \tilde{\mathcal{I}}^n_{2'2}(\tilde{\boldsymbol{\phi}}_1) \end{bmatrix} = \begin{bmatrix} \tilde{\mathcal{I}}^n_{2'2}(\tilde{\boldsymbol{\phi}}_3) & \tilde{\mathcal{I}}^n_{2'1}(\tilde{\boldsymbol{\phi}}_3) & \tilde{\mathcal{I}}^n_{2'1}(\tilde{\boldsymbol{\phi}}_2) \\ \tilde{\mathcal{I}}^n_{2'1}(\tilde{\boldsymbol{\phi}}_3) & \tilde{\mathcal{I}}^n_{2'2}(\tilde{\boldsymbol{\phi}}_4) & \tilde{\mathcal{I}}^n_{2'1}(\tilde{\boldsymbol{\phi}}_1) \\ \tilde{\mathcal{I}}^n_{2'1}(\tilde{\boldsymbol{\phi}}_2) & \tilde{\mathcal{I}}^n_{2'1}(\tilde{\boldsymbol{\phi}}_1) & \tilde{\mathcal{I}}^n_{2'2}(\tilde{\boldsymbol{\phi}}_1) \end{bmatrix} \tag{2.33}$$

Where for the common edge edge EFIE the symmetires lead to

$$\tilde{\boldsymbol{\mathcal{I}}}^n_{CE} = \begin{bmatrix} \tilde{\mathcal{I}}^n_{0'1}(\tilde{\boldsymbol{\phi}}_2) & \tilde{\mathcal{I}}^n_{0'1}(\tilde{\boldsymbol{\phi}}_1) & \tilde{\mathcal{I}}^n_{0'2}(\tilde{\boldsymbol{\phi}}_1) \\ \tilde{\mathcal{I}}^n_{2'1}(\tilde{\boldsymbol{\phi}}_2) & \tilde{\mathcal{I}}^n_{1'1}(\tilde{\boldsymbol{\phi}}_1) & \tilde{\mathcal{I}}^n_{2'2}(\tilde{\boldsymbol{\phi}}_2) \\ \tilde{\mathcal{I}}^n_{1'1}(\tilde{\boldsymbol{\phi}}_2) & \tilde{\mathcal{I}}^n_{2'1}(\tilde{\boldsymbol{\phi}}_1) & \tilde{\mathcal{I}}^n_{2'2}(\tilde{\boldsymbol{\phi}}_1) \end{bmatrix} = \begin{bmatrix} \tilde{\mathcal{I}}^n_{0'1}(\tilde{\boldsymbol{\phi}}_2) & \tilde{\mathcal{I}}^n_{0'1}(\tilde{\boldsymbol{\phi}}_1) & \tilde{\mathcal{I}}^n_{0'2}(\tilde{\boldsymbol{\phi}}_1) \\ \tilde{\mathcal{I}}^n_{2'1}(\tilde{\boldsymbol{\phi}}_2) & \tilde{\mathcal{I}}^n_{1'1}(\tilde{\boldsymbol{\phi}}_1) & \tilde{\mathcal{I}}^n_{2'2}(\tilde{\boldsymbol{\phi}}_2) \\ \tilde{\mathcal{I}}^n_{1'1}(\tilde{\boldsymbol{\phi}}_2) & \tilde{\mathcal{I}}^n_{2'1}(\tilde{\boldsymbol{\phi}}_1) & \tilde{\mathcal{I}}^n_{2'2}(\tilde{\boldsymbol{\phi}}_1) \end{bmatrix} \tag{2.34}$$

And for the common edge MFIE case

$$\tilde{\mathbf{Z}}^{(e)\ \mathcal{K}_{pv}} = \begin{bmatrix} \tilde{Z}_{0'0}(\tilde{\boldsymbol{\phi}}_1) & 0 & \tilde{Z}_{0'2}(\tilde{\boldsymbol{\phi}}_1) \\ \tilde{Z}_{1'0}(\tilde{\boldsymbol{\phi}}_1) & \tilde{Z}_{1'1}(\tilde{\boldsymbol{\phi}}_1) & 0 \\ \tilde{Z}_{2'0}(\tilde{\boldsymbol{\phi}}_1) & \tilde{Z}_{2'1}(\tilde{\boldsymbol{\phi}}_1) & \tilde{Z}_{2'2}(\tilde{\boldsymbol{\phi}}_1) \end{bmatrix} = \begin{bmatrix} \tilde{Z}_{0'0}(\tilde{\boldsymbol{\phi}}_1) & 0 & \tilde{Z}_{0'2}(\tilde{\boldsymbol{\phi}}_1) \\ -\tilde{Z}_{0'0}(\tilde{\boldsymbol{\phi}}_2) & \tilde{Z}_{0'2}(\tilde{\boldsymbol{\phi}}_2) & 0 \\ \tilde{Z}_{2'0}(\tilde{\boldsymbol{\phi}}_1) & \tilde{Z}_{2'1}(\tilde{\boldsymbol{\phi}}_1) & -\tilde{Z}_{2'1}(\tilde{\boldsymbol{\phi}}_2) \end{bmatrix} \tag{2.35}$$

The subscripts of the integrals denote the source, observation triangle edges respectively, while the subscript of the interpolant $\tilde{\boldsymbol{\phi}}$ denotes a particular parametrization of the triangle as shown in Fig. 2.9. The relations shown above were verified by use of an in-house BEM code deploying Gaussian quadrature, [30] for regular integration and singularity subtraction integration, [37]. Reference results were provided by Professor D. R. Wilton for the validation of the singular integration, as well as radar cross section (RCS) results provided by the analytical Mie series, [4]. The geometrical symmetries of the integration domain (the triangle pairs) translate to symmetries in the BEM element matrices according to the operator. The relationships given above speak exactly to that. In the case of the MFIE K operator there are some negative

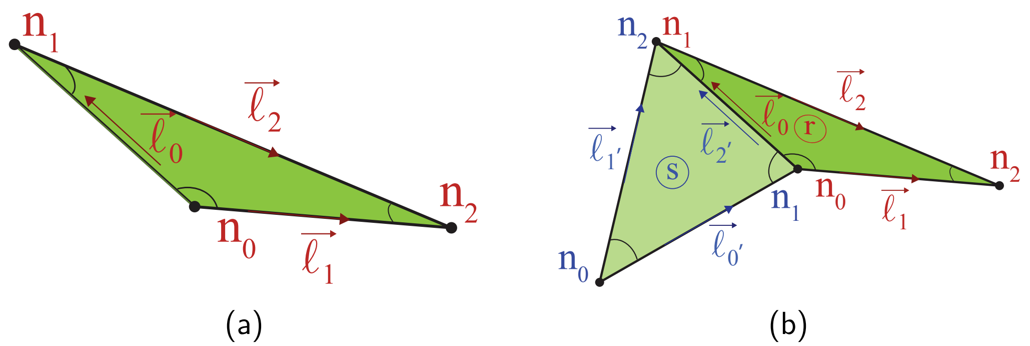


Figure 2.8. Node and edge numbering for a) common triangle b) common edge.

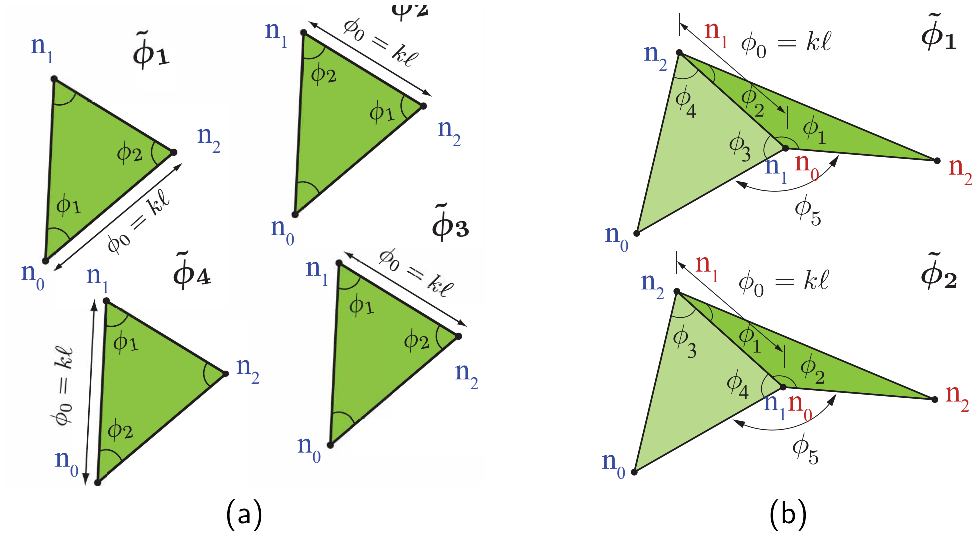


Figure 2.9. Equivalent parametrization orderings that are used to identify symmetries in the element matrices. a) Common triangle case. b) Common edge case.

symmetries because of the inner product. Notice that one particular entry was found to always be equal to zero.

2.4.3 Parameter Domain Transformation

Transforming the parametrization domain from a prism to a cube or a six-dimensional (6D) hyper prism into a 6D hyper-cube it becomes more convenient to sample and perform interpolation by emplying the tensor product properties of such domains. For the sake of consistency the domain was transformed in the common triangle case

Table 2.1. Interpolants of Figure 2.9. The superscript s denotes the source triangle and r the receiver.

	interpolant	ϕ_0	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5
CT	$\tilde{\boldsymbol{\phi}}_{\mathbf{1}}$	$k\ell_1$	$\widehat{n_1 n_0 n_2}$	$\widehat{n_0 n_2 n_1}$	-	-	-
	$\tilde{\boldsymbol{\phi}}_{\mathbf{2}}$	$k\ell_2$	$\widehat{n_0 n_2 n_1}$	$\widehat{n_2 n_1 n_0}$	-	-	-
	$\tilde{\boldsymbol{\phi}}_{\mathbf{3}}$	$k\ell_2$	$\widehat{n_2 n_1 n_0}$	$\widehat{n_0 n_2 n_1}$	-	-	-
	$\tilde{\boldsymbol{\phi}}_{\mathbf{4}}$	$k\ell_0$	$\widehat{n_2 n_1 n_0}$	$\widehat{n_1 n_0 n_2}$	-	-	-
CE	$\tilde{\boldsymbol{\phi}}_{\mathbf{1}}$	$k\ell_0$	$\widehat{n_2^r n_0^r n_1^r}$	$\widehat{n_2^r n_1^r n_0^r}$	$\widehat{n_0^s n_1^s n_2^s}$	$\widehat{n_0^s n_2^s n_1^s}$	$180° - \widehat{\boldsymbol{n}^r, \boldsymbol{n}^s}$
	$\tilde{\boldsymbol{\phi}}_{\mathbf{2}}$	$k\ell_0$	$\widehat{n_2^r n_1^r n_0^r}$	$\widehat{n_2^r n_0^r n_1^r}$	$\widehat{n_0^s n_2^s n_1^s}$	$\widehat{n_0^s n_1^s n_2^s}$	$180° - \widehat{\boldsymbol{n}^r, \boldsymbol{n}^s}$

as well, as is shown in Fig. 2.10. Essentially the domain is rotated 90° around the ϕ_0 axis and the origin is expanded from a point to a line via a Duffy-like transformation, [80]. In doing so a very fine sampling rate is achieved near the origin region and the adjacent edges, as is explained in detail in the next section.

The forward transformation is given by:

$$\tilde{\phi}_0 = \phi_0 \tag{2.36}$$

$$\tilde{\phi}_1 = \frac{\sqrt{2}}{2}\left(\phi_1 + \phi_2\right) \tag{2.37}$$

$$\tilde{\phi}_2 = 180\frac{\sqrt{2}}{2}\frac{\left(\phi_2 - \phi_1\right)}{\left(\phi_2 + \phi_1\right)} \tag{2.38}$$

$$\tilde{\phi}_3 = \frac{\sqrt{2}}{2}\left(\phi_3 + \phi_4\right) \tag{2.39}$$

$$\tilde{\phi}_4 = 180\frac{\sqrt{2}}{2}\frac{\left(\phi_4 - \phi_3\right)}{\left(\phi_4 + \phi_3\right)} \tag{2.40}$$

$$\tilde{\phi}_5 = \phi_5 \tag{2.41}$$

whereas the inverse transformation is:

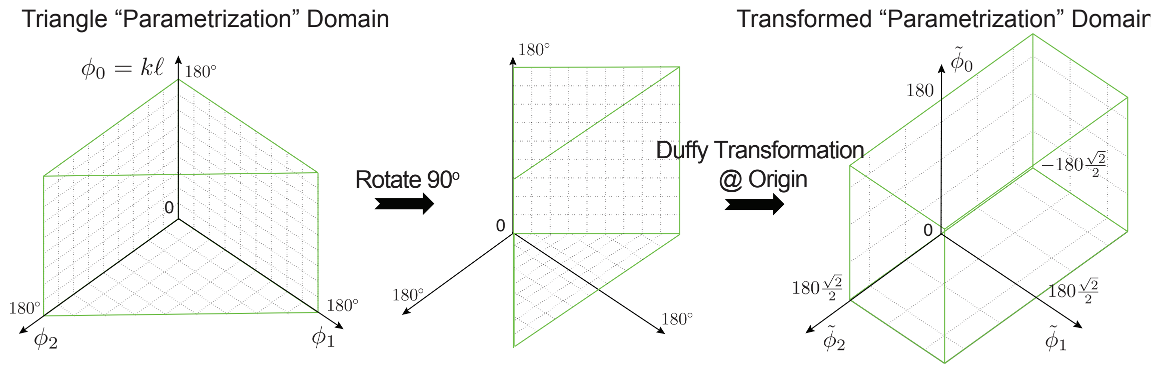


Figure 2.10. Transformation of the parametrization domain to a rectangular (or hyper-rectangular) one. This transformation is more convenient because it enables the use of tensor-product, sparse grid and low-rank tensor interpolation schemes.

$$\phi_0 = \tilde{\phi}_0 \tag{2.42}$$

$$\phi_1 = \frac{\sqrt{2}}{2}\tilde{\phi}_1 - \tilde{\phi}_1\tilde{\phi}_2\frac{1}{180} \tag{2.43}$$

$$\phi_2 = \frac{\sqrt{2}}{2}\tilde{\phi}_1 + \tilde{\phi}_1\tilde{\phi}_2\frac{1}{180} \tag{2.44}$$

$$\phi_3 = \frac{\sqrt{2}}{2}\tilde{\phi}_3 - \tilde{\phi}_3\tilde{\phi}_4\frac{1}{180} \tag{2.45}$$

$$\phi_4 = \frac{\sqrt{2}}{2}\tilde{\phi}_3 + \tilde{\phi}_3\tilde{\phi}_4\frac{1}{180} \tag{2.46}$$

$$\phi_5 = \tilde{\phi}_5 \tag{2.47}$$

It is noted that for the common triangle case the forward or inverse transformation make use of the first three equation pairs i.e., (2.36-2.38) and (2.42-2.44) respectively.

2.4.4 Parametrization Domain Discretization

The process of sampling the "parametrization" domain and building a universal library of integrals, begins with *hp*-refinement of the **transformed** "parametrization" domain into elements, where *h* pertains to the variable size of the elements and *p* denotes the variable interpolation order among the elements. It is noted that the *hp*-refinement process can be automated like in the cases of adaptive finite element method (FEM). However, since in this case the underlying function is fixed, a nonadaptive, ad-hoc, hp-refinement approach was employed. This semi-automated trial-and-error process ended up being one of the most time-consuming steps in this work. The *hp*-refinement for the EFIE common triangle case is shown in Fig. 2.11; on the right it shows a top view of the refinement along the first two axes as well as the interpolation orders assigned to each element; on the left it shows the refinement in the last axis. Each element is a $3D$ cube, for example the first element is : $[0.36, 2] \times [2, 33] \times [-125.865, -95]$. The smoothness plots provide information into how the refinement should be; for example in Fig. 2.5 the function seems less smooth at the edges or the region close to the origin and should be sampled more finely at those regions, hence there are more elements along the top edge. Moreover, the transformation itself contributes to this sampling scheme; for example the 6 elements on the bottom of Fig. 2.11 all correspond to the small region close to the origin in the "parametrization" domain (original or non-transformed), which is shown in Fig. 2.12. If the sample locations on these elements were to be inverse-transformed and plotted on the "parametrization" domain the not so smooth region close to the origin would prove to be very finely sampled. The *hp* refinement presented herein was obtained through an iterative trial-and-error process. Namely at each iteration, a finer refinement was used to build a new library, which was then tested on a set of realistic example meshes (see the error histograms of Chapter 5) and the average and maximum relative error on each element was observed. Similarly, the *hp*-refinement for the EFIE common edge case is shown in Fig. 2.13, where each subfigure shows the top view of the refinement in a different pair of axes. Each element is a $6D$ hyper-cube, for example the first element is : $[0.36, 120] \times [5, 32] \times [-120.2082, -75] \times [5, 32] \times [-120.2082, -75] \times [5, 40]$. Likewise, the refinement for MFIE is shown in Fig. 2.14. A noteworthy point is the definition of the parametrization domain boundary, marked with red in all figures and defined by the angle limits given in (2.29-2.32). The parametrization domain does not extend to 0 and 180 degrees, meaning there are

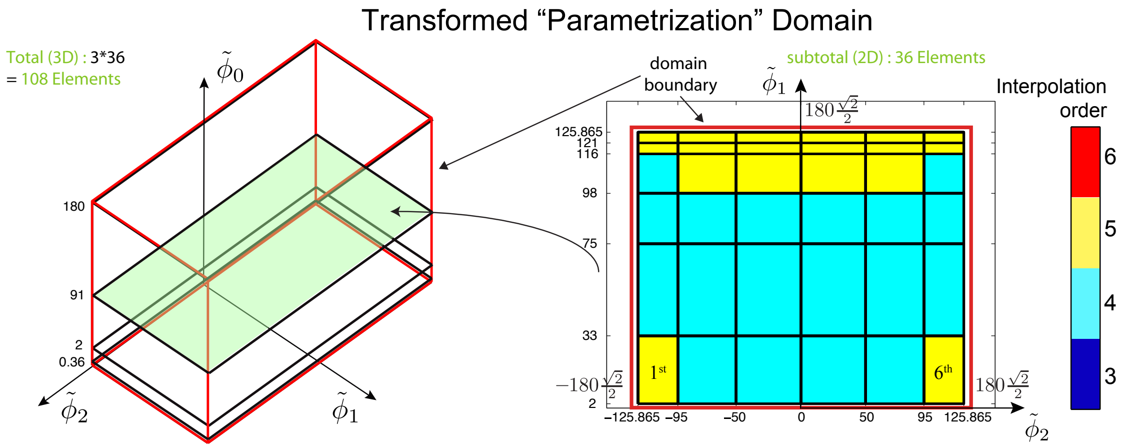


Figure 2.11. Domain *hp*-refinement (discretization) for the EFIE common triangle case. The orders for a delayed sparse grids range from $p = 6$ to 10.

no elements very close to the boundary. That is because in this region the function varies rapidly, as a result of the triangles being so highly skewed (alternatively they have very poor triangle quality) that the *hp* refinement would have to be so fine as to ultimately become very inefficient. If a mesh contains interactions that lie in this region, it would be preferable to use an alternative singular integration technique; and provide a warning that the mesh contains poorly shaped triangles that do not have good approximation properties. Moreover, notice how in the case of EFIE common edge, Fig. 2.13 the angle ϕ_5 goes up to 180°; this limit however, is not strict and in the case of MFIE, shown in Fig. 2.14 it is extended to 190° to improve the interpolation error for common edge interactions with $\phi_5 \in [170, 180]$.

2.4.5 Element Sampling

Each element in the transformed parametrization domain is sampled (amounts to evaluating the underling singular integral with a slow but high-accuracy quadruple precision singularity subtraction method) using a grid pattern. The sampling is performed in a reference element that is centered at $0.5 \times 0.5 \times 0.5$ in 3D (and 0.5^6 in 6D) and is having unit length along each side. This reference element is afine transformed

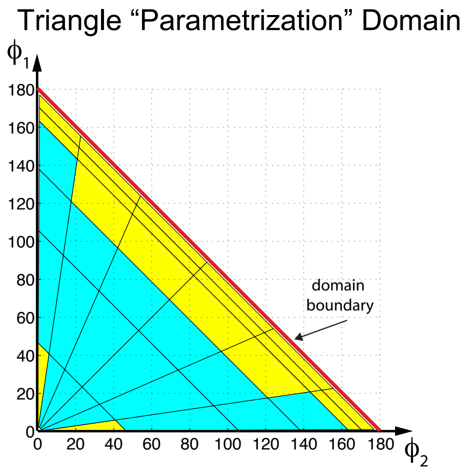


Figure 2.12. *hp*-refinement on the right of Fig. 2.11 inverse-transformed to the original parametrization domain.

to the actual parametrization domain element location, akin to finite element assembly process [14]. In all cases in this dissertation the sampling has tensor product or sparse grid form and as such all sampling can be derived from 1D sampling. The sampling abscissae vector $\boldsymbol{x}$ contains the interpolation samples in the $1D$ case, which in this work range from 0 to 1. The Gauss-Patterson [74] abscissae were found to be optimal among a selection of others (uniform, Chebyshev, Clenshaw-Curtis); for an order of $p = 4$ they are given as, [74]:

$$\begin{aligned}\boldsymbol{x}^4 = \{&0.0005, 0.0030, 0.0092, 0.0198, 0.0352, 0.0557, 0.0816, 0.1127, 0.1488, 0.1894,\\ &0.2343, 0.2828, 0.3344, 0.3883, 0.4438, 0.5000, 0.5562, 0.6116, 0.6656, 0.7171,\\ &0.7657, 0.8105, 0.8512, 0.8872, 0.9184, 0.9442, 0.9648, 0.9802, 0.9908, 0.9969,\\ &0.9995\}\end{aligned}$$

Therefore, the unit cell for a $p = 4$ interpolation order element of the common triangle case, would be given as $\boldsymbol{x}^4 \otimes \boldsymbol{x}^4 \otimes \boldsymbol{x}^4$, as shown in Fig. 2.16a. Remember that to sample means to compute the singular integrals of one of (2.25,2.26,2.23). Each grid

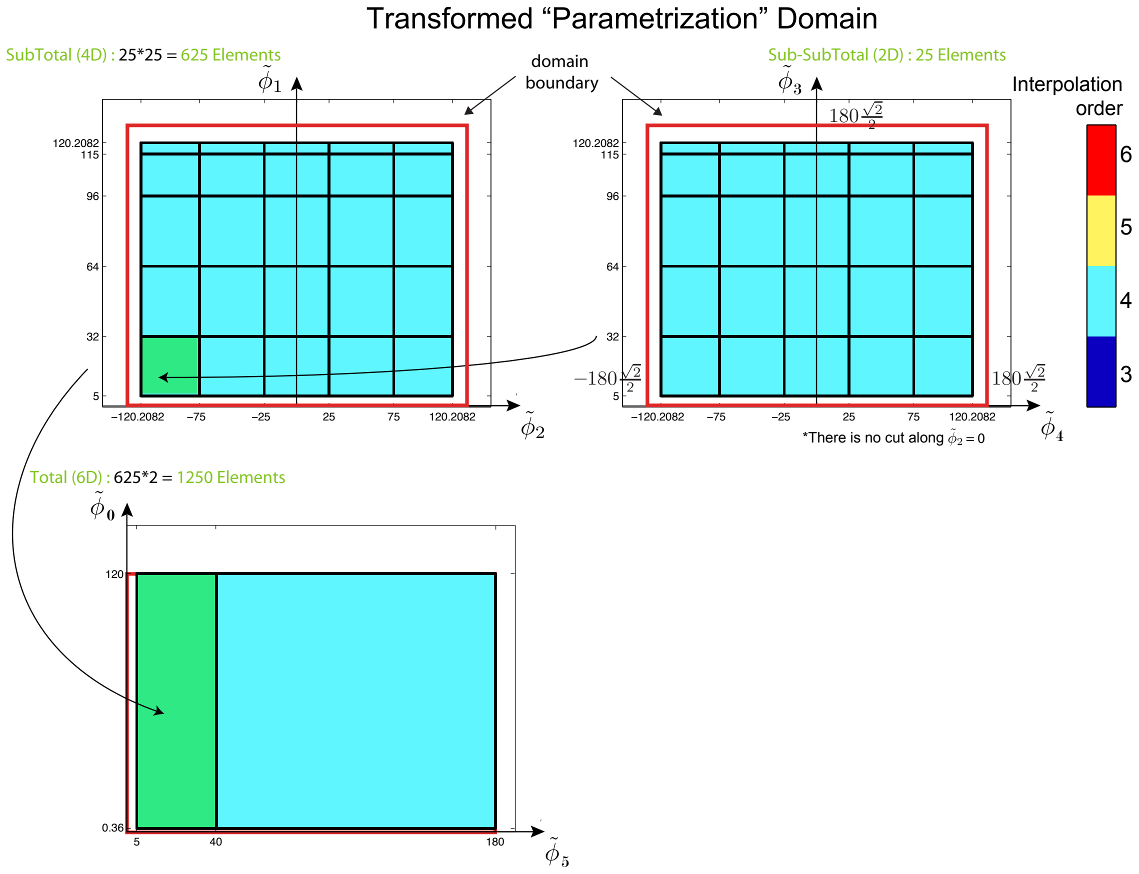


Figure 2.13. Parametrization domain *hp*-refinement for the EFIE common edge case. The orders for a delayed sparse grids range from $p = 6$ to 10.

point is scaled by its element's size to a vector $\tilde{\boldsymbol{\phi}}$ and then inverse-transformed via (2.42-2.47) to a vector of angles $\boldsymbol{\phi}$, which corresponds to a common triangle/edge geometry, whose singular integrals can be computed via any state-of-the-art method; in this manuscript the singularity subtraction of [37] is used. This completes the process of building a library/look up table of integrals, which are *universal* (frequency, materials independent) and is built *once-and-for-al*. This work employes three different libraries, one for each case: the EFIE common triangle case and the EFIE, MFIE common edge cases. Subsequently, these libraries can be loaded and the integrals of any arbitrary common triangle/edge interactions of a BEM mesh can be computed via interpolating among those finite sample points. It is emphasized that the interpolation

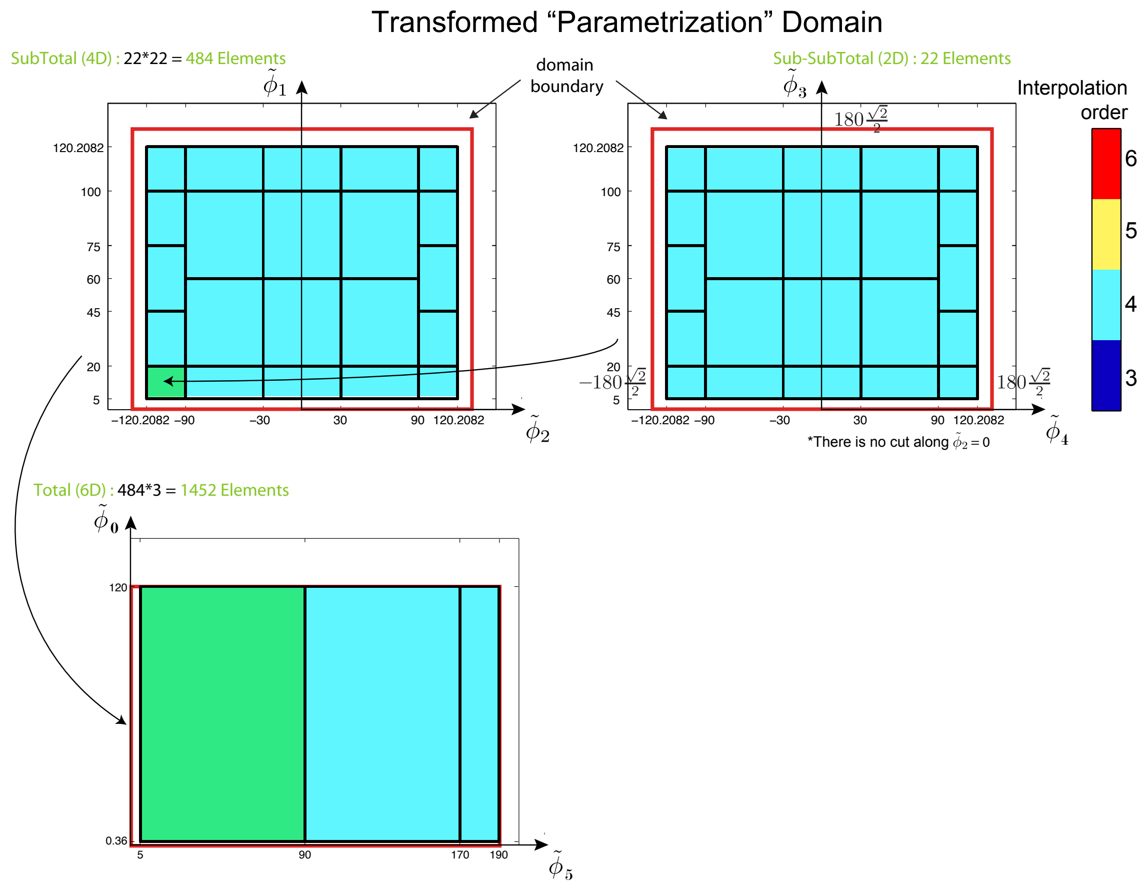


Figure 2.14. Parametrization domain *hp*-refinement for the MFIE common edge case. The orders for a delayed sparse grids range from $p = 6$ to 10.

is local and is performed on each element once for all its interpolants; such an example is shown in Fig. 2.15. Finally, the interpolated frequency independent integrals of (2.25,2.26,2.23) are mapped back to the original integrals of (2.14,2.15-2.16) using (2.20,2.22,2.24) respectively. The reference element interpolation/sampling scheme is crucial to the performance SIBI. The number and location of sample points must be selected carefully to maximize interpolation accuracy, while minimizing computation and memory cost. Generally, p refinement (increasing interpolation order) marginally increases accuracy in exchange for computation cost, while h refinement increases accuracy somewhat in exchange for memory. It is generally preferable to prioritize accuracy and computation cost. However, the number of grid points in each

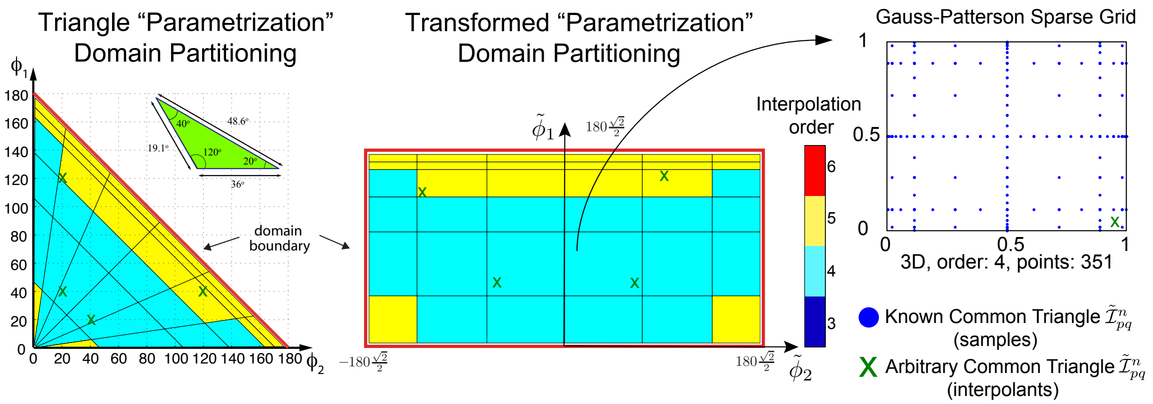

Triangle “Parametrization” Domain Partitioning
ϕ_1
180
160
140
120
100
80
60
40
20
0
0 20 40 60 80 100 120 140 160 180
ϕ_2
40°
48.6°
19.1°
120°
20°
36°
domain boundary
Transformed “Parametrization” Domain Partitioning
$\tilde{\phi}_1$
$180\frac{\sqrt{2}}{2}$
$-180\frac{\sqrt{2}}{2}$
$180\frac{\sqrt{2}}{2}$
$\tilde{\phi}_2$
Interpolation order
6
5
4
3
Gauss-Patterson Sparse Grid
1
0.5
0
0
0.5
1
3D, order: 4, points: 351
Known Common Triangle $\tilde{\mathcal{I}}^n_{pq}$ (samples)
Arbitrary Common Triangle $\tilde{\mathcal{I}}^n_{pq}$ (interpolants)

Figure 2.15. Sampling/Interpolation strategy of SIBI. A particular common triangle interaction is mapped to four interpolants in the parametized model domain, which are mapped to various elements in the transformed (hypercube) parametrized domain. Each element in the parametrized domain has a pre-defined sampling pattern (sparse grid in this case), where the integrals have been pre-evaluated and tabulated once and for all. The specific interaction (interpolant) can be interpolated localy within the element using the values of the pre-evaluated samples.

element quickly becomes very large even in the common triangle case ($3D$) using tensor product sampling strategies, based on 1D Gauss-Patternson abscissas, as shown in Fig. 2.16a. Therefore, two more efficient sampling/interpolation methods will be detailed in the following chapters. Chapter 3 details the sparse grid [74] and delayed sparse grid [83] sampling/interpolation, followed by more advanced low-rank tensor interpolation based on tensor Tucker [78] and tensor-trains [78] decompositions, in Chapter 4.

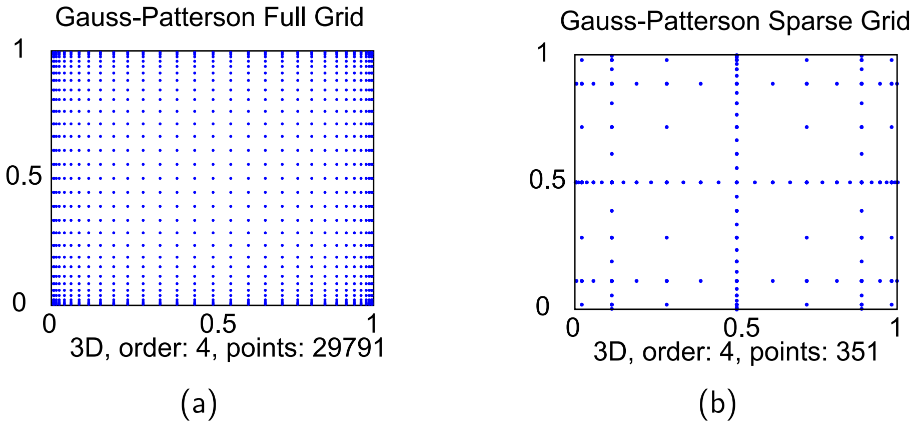


Figure 2.16. Sampling pattern of a 3D reference element (2D cut shown) with Gauss-Patterson abscissae a) tensor product grid b) sparse grid.

CHAPTER 3
SPARSE GRIDS INTERPOLATION

Although the SIBI relies on a local interpolation i.e. each element in the parametrized domain is interpolated independently of the others leading to large seving, it requires a high dimension interpolation within the elements. For the case of common triangle, where the parametrization domain is 3D, the interpolation can be extremely fast even with rudimentary tensor product 1D interpolation strategies. However, for the common edge cases, where the parametrization domain is 6D, the reference element sampling and interpolation scheme plays critical role in accuracy and efficiency. Essentially the challenge has to do with coping with the increasing dimensionality that exponentially increases the number of samples and interpolation computations within the transformed element hypercubes.

There are several known approaches that can cope reasonably well with moderate dimensionality order (less than 10) [REFs], and one of those is Smolyak's sparse grid appoach [74].

Sparse grids reference element retains the accuracy of a high order full tensor product one, while the number of grid points is significantly reduced leading to a memory cost reduction and more importantly a computation cost reduction (see Fig. 2.16). One of the key characteristics and the main reason that allows sparse grids to save points lies in the “nested form” of the abscissae. The other key characteristic is that the sparse grid is built up hierarchically in levels in such a way that multiple re-use of samples in the previous levels is levereged. This works also explores the use of delayed sparse grids that are a modification of the standard sparse grid algorithms and leads to notisable savings in our application.

3.1 Smolyak's Grid Construction

A d dimensional sparse grid of order p has $p+1$ levels and is build by adding up all the sub-grids of each level ℓ. Let $\boldsymbol{i} = \{i_0, i_1, \ldots, i_{d-1}\}$ be a multi-index, then a level ℓ is the set of multi-indices $\boldsymbol{i}$ such that $|\boldsymbol{i}| = i_0 + i_1 + \ldots + i_{d-1} = \ell$. Let also $\mathbf{x}$ be the abscissae or $1D$ interpolation rules, which are vectors of real valued scalars that dictate the sampling location and usually range from 0 to 1 or -1 to 1. A sub-grid is then given as the tensor product of the abscissae vectors $\boldsymbol{x}^{i_0} \otimes \boldsymbol{x}^{i_1} \otimes \ldots \otimes \boldsymbol{x}^{i_{d-1}}$. The Gauss-Patterson abscissae in $[0,1]$, which were found to have better performance than other options, such as uniform, Chebyshev, Clenshaw-Curtis are given as, [74]:

$$\boldsymbol{x}^0 = \{0.5\} \tag{3.1}$$

$$\boldsymbol{x}^1 = \{0.1127, 0.5, 0.8872\} \tag{3.2}$$

$$\boldsymbol{x}^2 = \{0.01975, 0.1127, 0.2828, 0.5, 0.7171, 0.8872, 0.9802\} \tag{3.3}$$

$$\begin{aligned}\boldsymbol{x}^3 = \{&0.0030, 0.01975, 0.0557, 0.1127, 0.1894, 0.2828, 0.3883, 0.5, 0.6116, 0.7171,\\ &0.8105, 0.8872, 0.9442, 0.9802, 0.9969\}\end{aligned} \tag{3.4}$$

$$\begin{aligned}\boldsymbol{x}^4 = \{&0.0005, 0.0030, 0.0092, 0.0198, 0.0352, 0.0557, 0.0816, 0.1127, 0.1488, 0.1894,\\ &0.2343, 0.2828, 0.3344, 0.3883, 0.4438, 0.5000, 0.5562, 0.6116, 0.6656, 0.7171,\\ &0.7657, 0.8105, 0.8512, 0.8872, 0.9184, 0.9442, 0.9648, 0.9802, 0.9908, 0.9969,\\ &0.9995\}\end{aligned} \tag{3.5}$$

Notice that abscissae are nested/hierarchical, meaning that abscisse of $\boldsymbol{x}^1$ are included in $\boldsymbol{x}^2$ and $\boldsymbol{x}^2$ in $\boldsymbol{x}^3$ and so forth. Each sub-grid is a tensor product of any of these abscissae; for example in the common triangle case, where the reference element is three dimensional, i.e. $d = 3$ the very first sub-grid is $\boldsymbol{x}^0 \otimes \boldsymbol{x}^0 \otimes \boldsymbol{x}^0$, which is the point $(0.5, 0.5, 0.5)$; another example could be $\boldsymbol{x}^0 \otimes \boldsymbol{x}^1 \otimes \boldsymbol{x}^0$, which has the points $(0.5, 0.1127, 0.5)$, $(0.5, 0.5, 0.5)$, $(0.5, 0.8872, 0.5)$. Notice that the hierarchical/nested nature of abscissae enables a sigificant reduction in the size of the grids. In Fig. 3.1 a sparse grid of dimension $d = 2$ and order $p = 3$ is constructed from the addition of all sub-grids up to level $\ell = 3$. For level $\ell = 0 = |\boldsymbol{i}|$ there can only be one multi-index, $\boldsymbol{i} = \{0, 0\}$, but for level $\ell = 1 = |\boldsymbol{i}|$ the index could be $\boldsymbol{i} = \{0, 1\}$ or $\boldsymbol{i} = \{1, 0\}$ and so forth; diagonal lines connecting $i_0 = i_1$ mark the level and its sub-grids. It is important to note is that the gray points on every sub-grid, which are points that have already appeared on previous levels, will not be added a second time to the total grid, but are used in the "hierarchical" basis functions definition (explained later).

3.2 Hierarchical Basis Functions and Barycentric Interpolation Formula

Another key feature of the sparse grid interpolation, other than Smolyak's grid construction are the hierarchical interpolation basis functions, which are shown in Fig. 3.2b and compared to interpolatory basis functions in Fig. 3.2a for the Gauss-Patterson abscissae and order $p = 2$ and dimension $d = 1$. In both cases, there are as many basis functions as there are abscissae. Each basis function has a value of 1 at its "base node" and 0 at all other "support nodes". In the case of interpolatory basis functions, shown in Fig. 3.2a, the basis functions are defined solely using the abscissae $\boldsymbol{x}^2$; while in the case of hierarchical, shown in Fig. 3.2b, they are defined on three levels using $\boldsymbol{x}^0, \boldsymbol{x}^1$ and $\boldsymbol{x}^2$ for levels $\ell = 0, 1, 2$ respectively. Therefore on level $\ell = 0$ there is one constant basis function with base node 0.5 and no support nodes; on level $\ell = 1$ there are two basis functions with base nodes 0.1172 and 0.8872 respectively and support nodes $0.5, 0.8872$ and $0.1172, 0.5$ respectively (and so forth

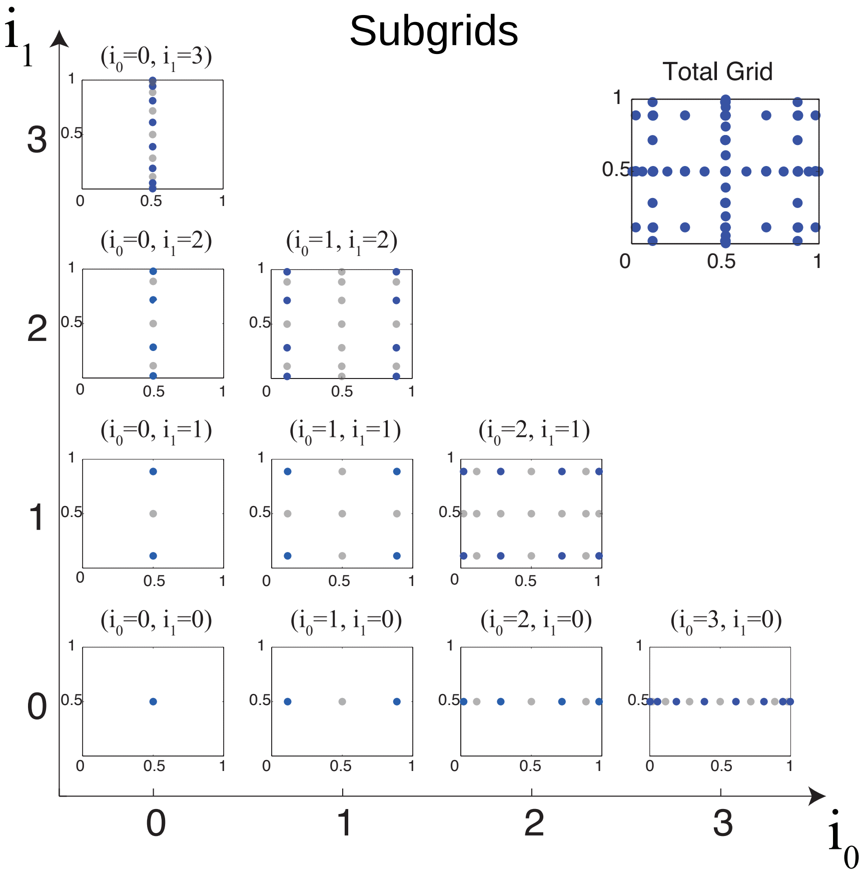
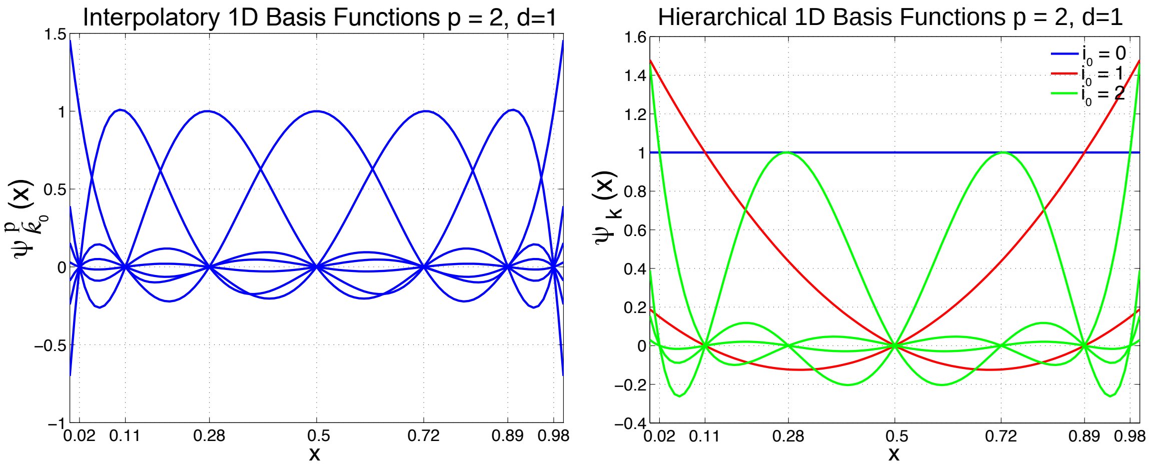


Figure 3.1. Smolyak's sparse grid construction of a $2D$, 3^{rd} order sparse grid. The gray dots mark inactive subgrid points. The horizontal i_0 axis represent the interpolation order along the first e.g., x, dimension, whereas the vertical, i_1 axis is that for the second dimension. A sparse grid (top right figure) is the direct sum of all active abscissae (shown in blue) in each level (level is considered the sum of each dimensional order to be same i.e., in this case $i_0 + i_1 = \ell$, where ℓ is the level)

for level $\ell = 2$). Notice that in level $\ell = 1$ there is no basis function with a base node 0.5, because that was already accounted for (added) in level $\ell = 0$; it is a gray, "inactive" point in the Smolyak grid construction. Inactive points are defined only in the hierarchical case and are used as support nodes only for all higher level basis functions; for example 0.5 will be used as a support node for all levels, in this case $\ell = 1, 2$.



(a) (b)

Figure 3.2. Basis functions of the 2^{nd} barycentric formula with Gauss-Patterson abscissae a) interpolatory basis b) hierarchical basis.

3.2.1 Barycentric Formula

In this work the hierarchical basis functions are coupled with the 2^{nd} barycentric formula [84], which is a more stable and numerically faster version of common Lagrange polynomial evaluations. For a reference element sparse grid of order p and dimension d, let $j = \{0, 1, 2, \ldots, d-1\}$ be a dimension index and $\boldsymbol{i}$ be the multi-index of level ℓ of Smolyak's algorithm, $\tilde{\boldsymbol{\phi}}$ be the interpolant and $\tilde{\boldsymbol{x}} \in [0, 1]^d$ be the abscissae. Then i_j is the multi-index in dimension j and denotes the abscissae, weight vector to be used in dimension j; for example if $\ell = 1, d = 3, \boldsymbol{i} = \{0, 1, 0\}$ then $\boldsymbol{x}^{i_0} = \boldsymbol{x}^0, \boldsymbol{x}^{i_1} = \boldsymbol{x}^1, \boldsymbol{x}^{i_2} = \boldsymbol{x}^0$. Finally let $k_j = \{0, 2, 4, \ldots, M^{i_j} - 1\}$ be the index of the base node in dimension j, where M^{i_j} is the size of the abscissae vector $\boldsymbol{x}^{i_j}$. Then the k^{th} basis function, psi^i_k for the sub-grid is given as, [74]:

$$\psi_k^{\boldsymbol{i}}(\tilde{\boldsymbol{x}}) = \prod_{j=0}^{d-1} \psi_{k_j}^{i_j}(\tilde{x}_j) \tag{3.6}$$

$$\psi_{k_j}^{i_j}(\tilde{x}_j) = \frac{f_{k_j}(\tilde{x}_j)}{\sum_{m=0}^{M^{i_j}-1} f_m(\tilde{x}_j)}, \quad \psi^0 = 1 \tag{3.7}$$

$$f_m(\tilde{x}_j) = \frac{w_m^{i_j}}{\tilde{x}_j - x_m^{i_j}}, \quad \tilde{x}_j \in [0,1] \tag{3.8}$$

Where x_i are the Gauss-Patterson abscissae and w_i are the Gauss-Patterson weights given by, [74]:

$$\boldsymbol{w}^0 = \{1\} \tag{3.9}$$

$$\boldsymbol{w}^1 = \{0.5, -1, 0.5\} \tag{3.10}$$

$$\boldsymbol{w}^2 = \{0.23895, -0.65544, 0.91649, -1, 0.91649, -0.65544, 0.23895\} \tag{3.11}$$

$$\begin{aligned}\boldsymbol{w}^3 = \{&0.30857, -0.77596, 0.97305, -1, 0.96511, -0.92145, 0.89041,\\ &-0.87948, 0.89041, -0.92145, 0.96511, -1, 0.97305, -0.77596,\\ &0.30857\}\end{aligned} \tag{3.12}$$

$$\begin{aligned}\boldsymbol{w}^4 = \{&0.42847, -0.96749, 1, -0.80737, 0.5959, -0.42841, 0.30902, -0.2271,\\ &0.17152, -0.13381, 0.10815, -0.09073, 0.07909, -0.07168, 0.06756,\\ &-0.06624, 0.06756, -0.07168, 0.07909, -0.09073, 0.10815, -0.13381,\\ &0.17152, -0.2271, 0.30902, -0.42841, 0.5959, -0.80737, 1, -0.96749,\\ &0.42847\}\end{aligned} \tag{3.13}$$

Equation (3.6) represents the Smolyak sparse grid product, whereas (3.8) and (3.8) are the one dimensional hierarchical basis functions defined though the barycentric interpolation formula.

3.3 Sparse Grid Interpolation Formula

Having defined the sparse grids and the hierarchical basis on it, we are ready to show how an integral $\mathcal{I}$ can be interpolated inside a d-dimensional sparse grid reference element. If a particular interpolant i.e. interaction geometry (mapped to the reference element) is denoted by $\tilde{\boldsymbol{\phi}}$, then the integral can be evaluated from the Gauss-Pattersons sparse grid samples as, [74]:

$$\mathcal{I}_{p,d}(\tilde{\boldsymbol{\phi}}) = \mathcal{I}_L(\tilde{\boldsymbol{\phi}}^{\mathbf{0}}) + \sum_{\ell=1}^{p} \sum_{|\mathbf{i}|=\ell} \psi_k^{\boldsymbol{i}}(\tilde{\boldsymbol{x}}) \Delta\mathcal{I}_L(\tilde{\boldsymbol{\phi}}^{\mathbf{k}}) \tag{3.14}$$

$$\Delta\mathcal{I}_L(\tilde{\boldsymbol{\phi}}^{\mathbf{k}}) = \mathcal{I}_L(\tilde{\boldsymbol{\phi}}^{\mathbf{k}}) - \mathcal{I}_{p-1,d}(\tilde{\boldsymbol{\phi}}^{\mathbf{k}}) \tag{3.15}$$

Where $\mathcal{I}$ denotes one of the wavenumber independent integrals of $\tilde{\boldsymbol{\mathcal{I}}}^n$, $\tilde{\mathbf{Z}}$ (2.25,2.26,2.23), $\mathcal{I}_L$ is a wavenumber independent integral stored in a library, $\mathcal{I}_{p,d}$ is the interpolated integral result on the d-dimensional, p-order reference element, $\tilde{\boldsymbol{\phi}}^{\mathbf{0}}(0.5, 0.5, \ldots, 0.5)$ is the node of level $\ell = 0$ and $\tilde{\boldsymbol{\phi}}^{\mathbf{k}}$ is the k^{th} node of the sub-grid $\boldsymbol{i}$ of level ℓ, $\Delta\mathcal{I}_L$ is the "hierarchical surplus" data. The surplus data for level $\ell = 0$ is identical to the non-surplus data. To compute the surpluses one would use (3.14), with interpolation order $p = 0$ and the interpolants would be the active nodes of all subgrids on level $\ell = 1$. The results of the interpolation would be subtracted from the non-surplus data and form the "hierachical surpluses" of level $\ell = 1$. The process continues with order $p = 1$ and the active nodes of all subgrids on level $\ell = 2$ as the interpolants; and so on and so forth until the final level is reached and all data has been converted to "hierarchical surplus" data. Once the surplus data has been computed the non-surplus data is no longer needed; hence the surplus data can be stored in a new library replacing the old one and thus significantly improving computational cost, without compromising memory or accuracy in any way.

3.4 Delayed Sparse Grids

As mentioned before the size of the interpolation reference element is crucial to the accuracy and computational cost of sparse grid interpolation. However, even with the sparse grid construction this size increases exponentially with the dimension of interpolation, thus its performance is severely deteriorated; a phenomenon known as the "curse-of-dimensionality", [75, 76]. Although the curse-of-dimensionality is a serious problem for virtually all known interpolation methods to moderate dimensions, for sparse grids, one option to better cope with it is to introduce "delayed" sparse grids. According to the observation of Novak & Ritter [83] a level ℓ sparse grid can exactly reconstruct polynomials of total degree $2\ell + 1$, i.e. posses the Gauss interpolation/integration property. Therefore, it is possible to construct a grid which follows the same nested construction as sparse grids, but delays the introduction of higher order abscissae $\boldsymbol{x}$ until the Novak & Ritter exactness constraint requires them. The grid construction follows the same Smolyak's algorithm, but utilizes a map for the indices $i_0, i_1, \ldots, i_{d-1}$ that folllows the pattern:

i_j regular	0	1	2	3	4	5	6	7	8	9	10
i_j delayed	0	1	1	2	2	2	3	3	3	3	3

The table above, shows that when Smolyak's sparse grid algorithm asks for $\boldsymbol{x}^10$, the delayed sparse grids employ only $\boldsymbol{x}^3$, leading to significant savings. This further reduces the number of total grid points, especially as dimension increases, while retaining the accuracy, [83]. This also allows for the use of higher order p, whereas regular Gauss-Patterson sparse grids are restricted to $p \leq 6$. The Smolyak's grid construction for a delayed sparse grid of dimension $d = 2$ and order $p = 3$ is shown in Fig. 3.3. Notice how some of the higher order sub-grids have been replaced with lower order ones, thus more points are being re-used and the total number of points (size of reference element) has overall been reduced. A comparison delayed sparse grid with the conventional sparse grid is shown in Figure 3.4 for a six dimensional reference element of order $p = 4$. Delayed sparse grids reduce the number of sample points from 2561 in the conventional sparse grid to 1889, corresponding to approximately 25% sample reduction. Figure 3.5 shows a results comparison, between delayed sparse grids and regular sparse grids; the details are explained in Chapter 5. The results show a clearly superior accuracy of the delayed sparse grids versus the regular sparse grids.

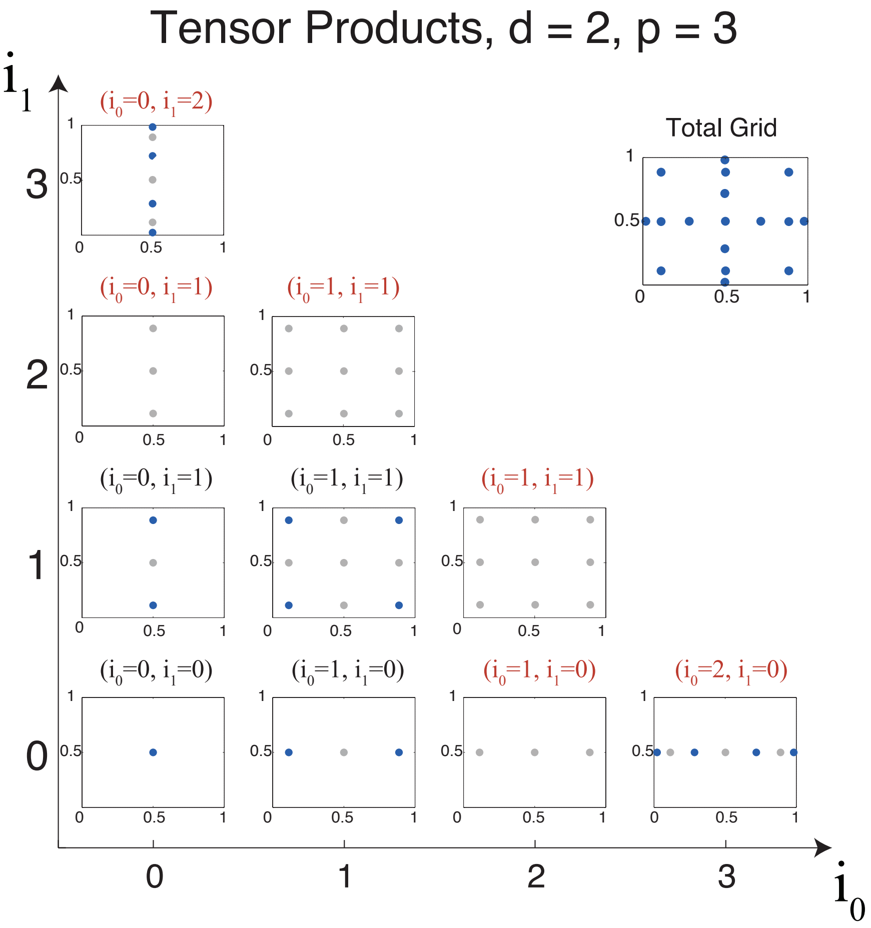
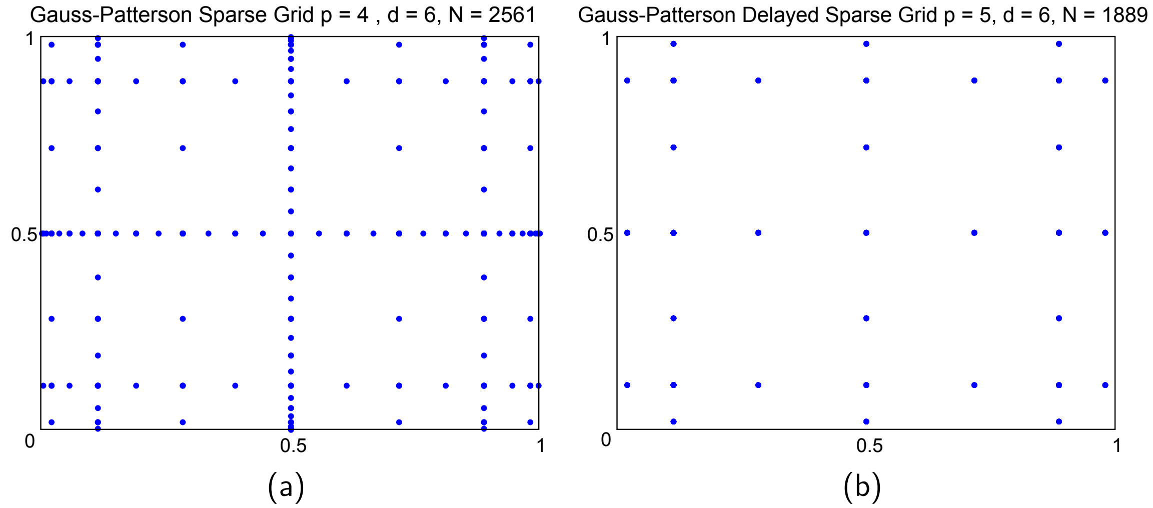
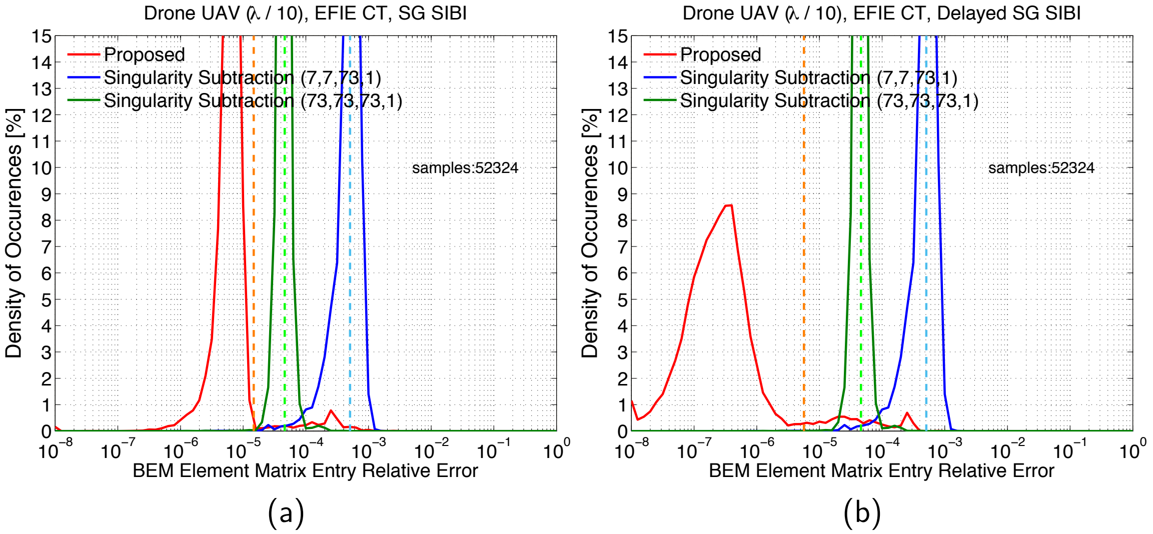


Figure 3.3. Delayed sparse grid construction of a $2D$ element of order 3.



(a) (b)

Figure 3.4. Comparison of a conventional spares grid with a delayed sparse grid for a $6D$ element of 4^{th} order ($d = 6, p = 4$). The pictures show a two dimensional cut along the hypercubes. a) sparse grid leads to $N = 2561$ samples b) delayed sparse grid leads to $N = 1889$ samples.



(a) (b)

Figure 3.5. EFIE common triangle case for drone UAV mesh with average length $\lambda/10$ (500MHz). Plots show histograms of the relative error in each entry of the BEM matrix that involves common triangle integration. The SG-SIBI/DSG-SIBI is compared with a 'typical' version of singularity subtraction using $(7, 7, 73, 1)$ rule (see Chapter 5 text for details) and a 'high accuracy' $(73, 73, 73, 1)$ rule. a) **Sparse Grid SIBI (SG-SIBI)** b) **Delayed Sparse Grid SIBI (DSG-SIBI)** .

CHAPTER 4
TENSOR DECOMPOSITION

Another option in alleviating the curse of dimensionality encountered in the interpolation of high-dimensional spaces/domains is use low-rank decomposition of full tensors i.e., products tensor (tensor is a generalization of a dense matrix in multiple dimensions), in a similar manner than singular value decomposition (SVD) [REF trefethen book] does for matrices. In the case of singular integration by interpolation (SIBI), this chapter will detail an approach that leverages tensor decompositions for the two common edge cases where sampling/interpolation of six dimensional parametrization elements are performed. The common triangle case, where the interpolation is in three dimensions is more straightforward and full tensor interpolation works very well with reasonable computational effort.

A tensor is a generalization of a dense matrix in more than two dimensions, [77, 85]. A tensor decomposition is a multi-dimensional Singular Value Decomposition (SVD) that decomposes the original tensor into a product of smaller sized tensors and/or matrices (termed factors or carriages), that can lead to significant savings, [78]. The size of the factors depends on the tensor numerical ranks, that is controlled by a prescribed approximation tolerance that plays the role of interpolation order in polynomial interpolation. In SIBI, the tensor decomposition representation replaces the full tensor or sparse grid representations of parametrization elements.

This chapter will explore two tensor decomposition approximations, the tensor Tucker decompositions (TD) and the tensor train (TT) decompositions. The latter approach was found more efficient than the former one, and in most cases better than the sparse grids of Chapter 3.

4.1 Tensor Tucker Decomposition

The Tucker decomposition (TD) of a $6D$ tensor consist of a product of seven factors, one smaller size $6D$ *core* tensor and six *side* matrices (2D tenors). The TD in this research is not rank-revealing, meaning that the full 6D tensor has to first be formed, and then decomposted via a series of SVDs. The reconstructed approximation is given as, [78]:

$$\boldsymbol{\mathcal{I}}(\tilde{\phi}_0, \tilde{\phi}_1, \ldots, \tilde{\phi}_5) \approx \sum_{\alpha_0,\ldots,\alpha_5} \mathbf{G}^{r_0\times\cdots\times r_5}(\alpha_0, \ldots, \alpha_5)\mathbf{U}_0^{n_0\times r_0}(\tilde{\phi}_0, \alpha_0)\cdots\mathbf{U}_5^{n_5\times r_5}(\tilde{\phi}_5, \alpha_5) \tag{4.1}$$

where $\mathcal{I}$ is the reconstructed full tensor of a wavenumber independent integral from one of eq. (2.26,2.23), $\mathbf{G}$ is the $6D$ *core* tensor of the decomposition, $\mathbf{U}_0, \ldots, \mathbf{U}_5$ are the *side* matrices of the decomposition, $\alpha_0, \ldots, \alpha_5$ summation/auxiliary indices, $r_0, \ldots, r_5$ are the compression ranks and $n_0, \ldots, n_5$ are the full tensor sizes. The superscript notation in (4.1) denotes the tensor dimensionality and size, i.e., $\mathbf{A}^{a\times b\times c}$ denotes a 3D tensor with sizes a in the first dimension, b in the second and c in the third. The decomposition or the reconstructed approximation using the Tensor Tucker format is depicted in Fig. 4.1.

The decomposition can be performed only after the full tensor has been computed and is a very time consuming computation. The grid for each element using full tensor has a tremendous number of points, as mentioned before and in the previous chapters, where each grid point entails lengthy singular integrations computed at maximum available accuracy. However, each element is constructed and decomposed

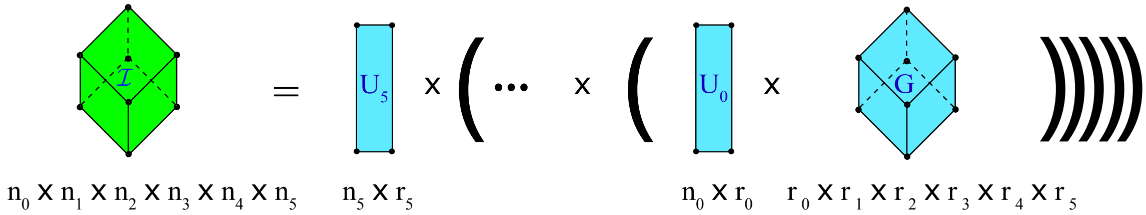


Figure 4.1. Tensor Tucker decomposition/reconstruction overview. A large hypecube is decomposed into a product of large 2D arrays and one small hybercube of the same dimension as the original one. If the dimension (rank) of the core-hypercube is small, this decomposition can lead to large savings in moderate dimensions.

independently of other elements, hence this computation is fully parallelizable on an element level, and is again done only once.

Once the TD for all elements have been formed in the off-line stage, the interpolation can be performed in the online stage. The TD interpolation in a parametrized element for a single interpolant is a series of matrix-matrix/vector multiplications, which is depicted in Fig. 4.2 given by, [78]:

$$\boldsymbol{\mathcal{I}}(\tilde{x}_0, \tilde{x}_1, \ldots, \tilde{x}_5) \approx (\boldsymbol{\psi}(\tilde{x}_5)\mathbf{U}_5)\,(\,\cdots\,(\,(\boldsymbol{\psi}(\tilde{x}_1)\mathbf{U}_1)\,(\,(\,\boldsymbol{\psi}(\tilde{x}_0)\mathbf{U}_0)\mathbf{G}\,)))) \tag{4.2}$$

Note that in Tensor Tucker interpolation the basis functions ψ are nodal, shown in Fig. 3.2a. In practice, it is possible to compute all interpolants in each parametrization domain element at once, as an efficient matrix-matrix multiplication.

4.2 Tensor Train Decomposition

In the tensor train singular integration by interpolation (TT-SIBI) the $6D$ tensor in common edge cases are decomposed into six factors, four $3D$ tensors and two matrices via a series of SVDs and/or QR decompositions. The reconstructed approximation is given by, [78]:

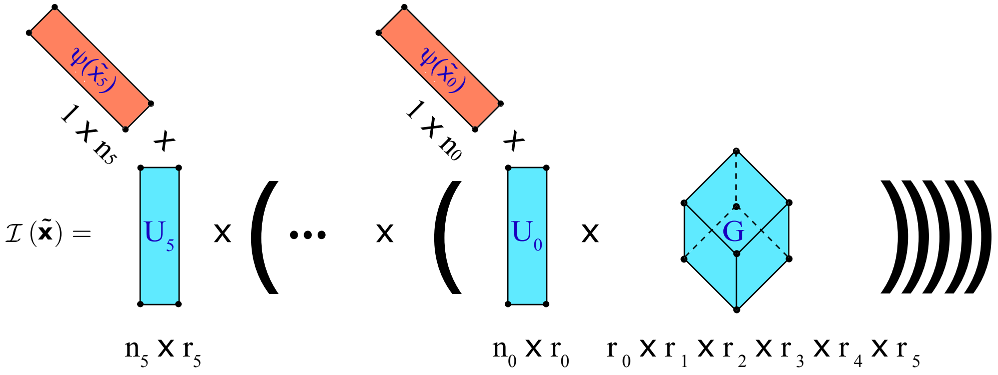


Figure 4.2. Tensor Tucker interpolation overview. The red vectors represent basis functions evaluated at the location of an interpolant. In the case of the more efficient vectorized implementations, multiple interpolations are processed at once, and the red vector become matrices.

$$\boldsymbol{\mathcal{I}}(\tilde{\phi}_0, \tilde{\phi}_1, \ldots, \tilde{\phi}_5) \approx \sum_{\alpha_0,\ldots,\alpha_4} \mathbf{G}_0^{n_0 \times r_0}(\tilde{\phi}_0, \alpha_0)\mathbf{G}_1^{r_0 \times n_1 \times r_1}(\alpha_0, \tilde{\phi}_1, \alpha_1) \cdots \mathbf{G}_5^{r_4 \times n_5}(\tilde{\phi}_4, \alpha_5) \tag{4.3}$$

where $\mathcal{I}$ is a wavenumber independent integral from one of eq. (2.26,2.23), $\mathbf{G}_0, \ldots, \mathbf{G}_5$ are the $2D$, $3D$ tensors, $\alpha_0, \ldots, \alpha_4$ summation/auxiliary indices, $r_0, \ldots, r_4$ are the compression ranks and $n_0, \ldots, n_5$ are the full tensor sizes. The same notation for tensor as in (4.2) is used. The decomposition or the reconstructed approximation using the TT format is depicted in Fig. 4.3. The frequency independent integral of an arbitrary common edge interaction is given by tensor interpolation as a series of matrix-matrix/vector multiplications, which is depicted in Fig. 4.4 and its formula is given below, [78]. Note that in Tensor Train interpolation the basis functions ψ are nodal, shown in Fig. 3.2a.

$$\boldsymbol{\mathcal{I}}(\tilde{x}_0, \tilde{x}_1, \ldots, \tilde{x}_5) \approx \Big\{\Big[\boldsymbol{\psi}(\tilde{x}_4)\Big[\boldsymbol{\psi}(\tilde{x}_3)\Big[\boldsymbol{\psi}(\tilde{x}_2)[\boldsymbol{\psi}(\tilde{x}_1)[(\boldsymbol{\psi}(\tilde{x}_0)\mathbf{G}_0)\mathbf{G}_1]]\mathbf{G}_2\Big]\mathbf{G}_3\Big]\mathbf{G}_4\Big]\mathbf{G}_5\Big\}\boldsymbol{\psi}(\tilde{x}_5) \tag{4.4}$$

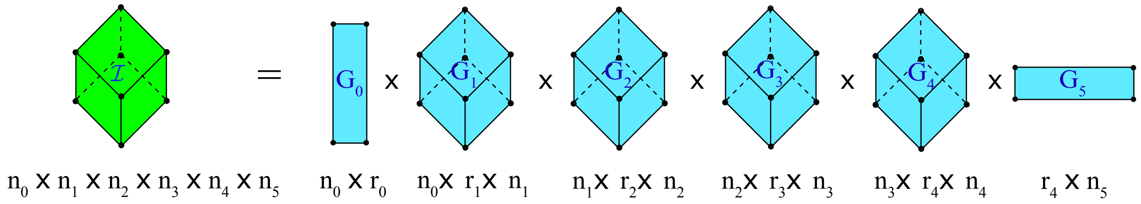

$\mathcal{I}$
=
G_0
x
G_1
x
G_2
x
G_3
x
G_4
x
G_5
$n_0 \times n_1 \times n_2 \times n_3 \times n_4 \times n_5$
$n_0 \times r_0$
$n_0 \times r_1 \times n_1$
$n_1 \times r_2 \times n_2$
$n_2 \times r_3 \times n_3$
$n_3 \times r_4 \times n_4$
$r_4 \times n_5$

Figure 4.3. Tensor Train decomposition/reconstruction overview. A large hypecube is decomposed into a product of two $2D$ arrays and a series (train) of small $3D$ arrays (cubes). If the dimension (ranks) of the core-cubes is small, this decomposition can lead to large savings in high-dimension.

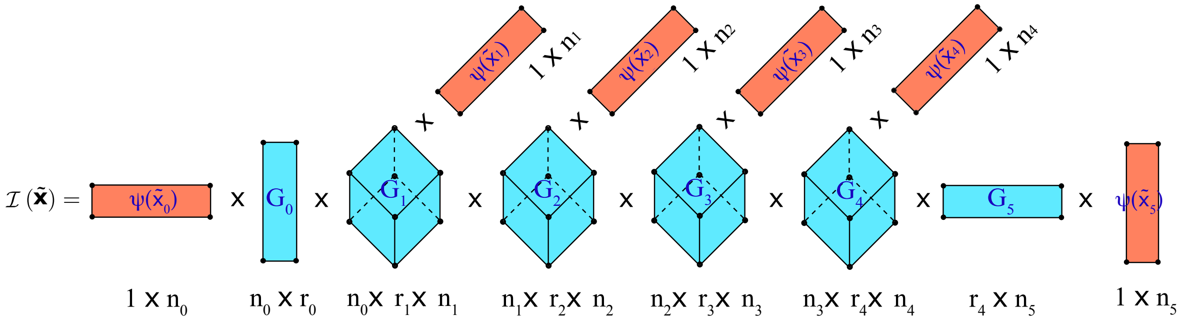


Figure 4.4. Tensor Train interpolation overview. The red vectors represent basis functions evaluated at the location of an interpolant. In the case of the more efficient vectorized implementations, multiple interpolations are processed at once, and the red vector become matrices.

The *hp*-refinement of the parametrized domain was found to be different than that of the sparse grids. This is mainly due to the fact that tensor representations are more suitable for using anisotropic order interpolations, i.e. tensor product interpolation schemes where the order of interpolation is different among dimensions. This particular refinement has significantly less elements (169 versus 1250 or 1452 for the sparse grids in the EFIE and MFIE case, respectively) because of the greatly increased number of points per element. Specifically in this refinement each element has anisotropic order $p = \{3, 2, 2, 2, 2, 3\}$, which leads to $15^2 \cdot 7^4 = 5.4 \cdot 10^5$ full grid points/samples per element, for a total of $169 \cdot 5.4 \cdot 10^5 = 9.1 \cdot 10^7$ points for the entire parametrization domain (library), as opposed to $1250 \cdot 2561 = 3.2 \cdot 10^6$ for

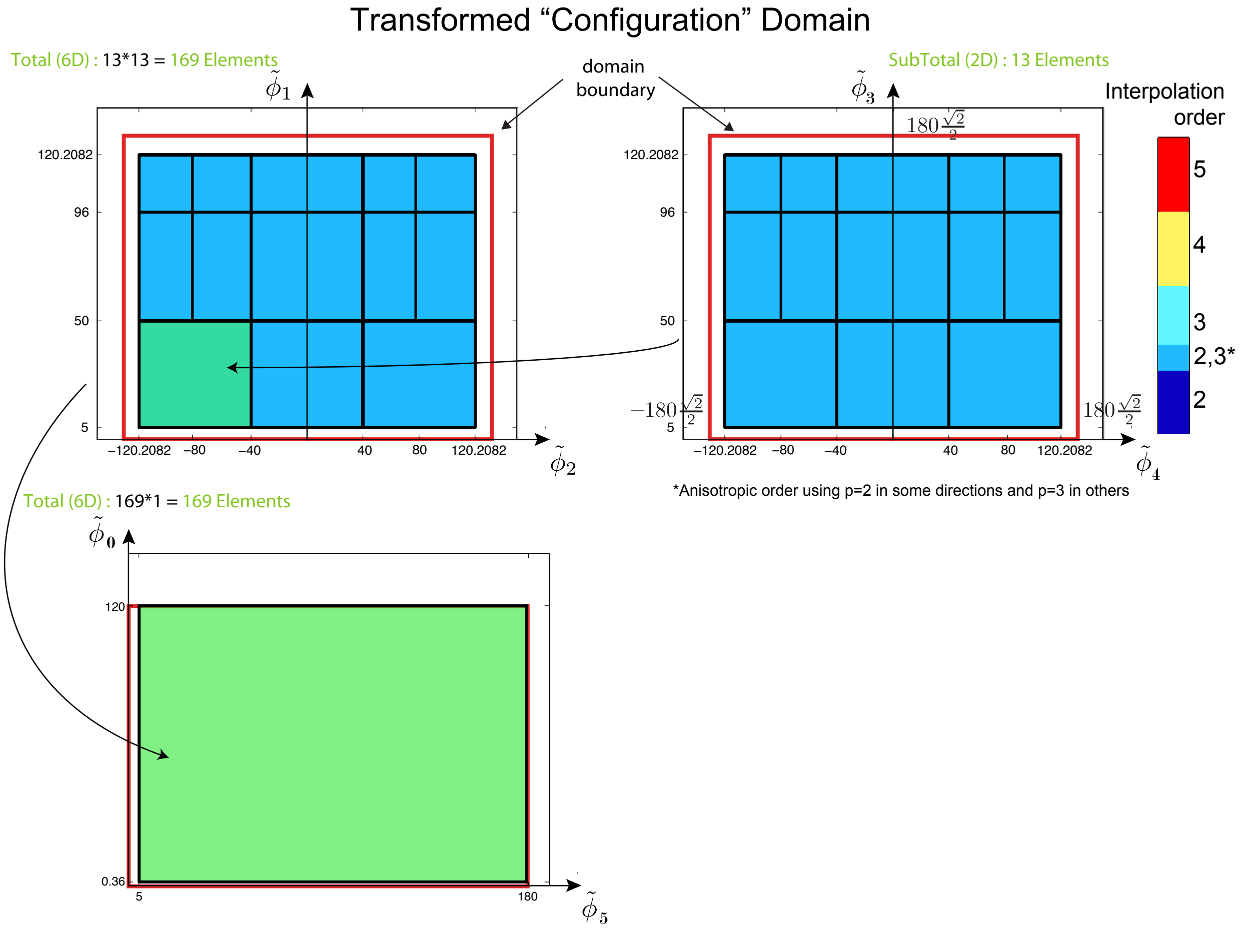


Figure 4.5. Domain *hp*-partitioning in the EFIE, MFIE common edge case to be used with a full tensor product grid unit cell and Tensor Decomposition.

the domain partitioning of fig. 2.13 using the regular sparse grid unit cell. Thus the two partitionings have similar size and *setup* computation times. Moreover note that for the tensor decomposition approach each full tensor element must first be built and stored, ie. every sample computed by a state-of-the-art rule at the highest available accuracy, before it is decomposed. In this regard the tensor decomposition approach could potentially offer far greater interpolation accuracy by increasing the number of elements (h refinement) and interpolation order (p refinement) with little to no speed compromise; whereas the sparse grids approach performance limits out, since h refinement was tested and showed little improvement and further p refinement

Table 4.1. Error, memory and run time for Drone UAV ($\lambda/10, 500MHz$) common edge interactions

		Error	Memory
Equation	Method	mean	[MB]
EFIE	Sparse Grid SIBI	$3.3 \cdot 10^{-4}$	488.5
	Tensor Tucker SIBI (core rank $\{5, 4, 4, 4, 4, 5\}$)	$3.5 \cdot 10^{-4}$	85.9
	Tensor Tucker SIBI (core rank $\{6, 6, 6, 6, 6, 6\}$)	$7.5 \cdot 10^{-5}$	290.6
	Tensor Train SIBI (tol 10^{-4})	$4.4 \cdot 10^{-5}$	39.3
	Tensor Train SIBI (tol 10^{-5})	$2.5 \cdot 10^{-5}$	93.3
MFIE	Sparse Grid SIBI	$3.3 \cdot 10^{-3}$	227
	Tensor Tucker SIBI (core rank $\{5, 4, 4, 4, 4, 5\}$)	$5.1 \cdot 10^{-3}$	68.7
	Tensor Tucker SIBI (core rank $\{6, 6, 6, 6, 6, 6\}$)	$2.0 \cdot 10^{-3}$	484.8
	Tensor Train SIBI (tol 10^{-3})	$1.3 \cdot 10^{-2}$	29.9
	Tensor Train SIBI (tol 10^{-4})	$2.4 \cdot 10^{-3}$	76.8
	Tensor Train SIBI (tol 10^{-5})	$7.0 \cdot 10^{-4}$	139.5

adversely affects speed. Table 4.1 shows a comparison of the average error of all common edge interactions in a Drone UAV mesh using sparse grids SIBI, tensor Tucker SIBI and tensor train SIBI, as well as the memory required by each method. The results show that the decomposition methods, tensor Tucker SIBI and tensor train SIBI utilizing increased decomposition tolerances achieve greater accuracy as well as increased memory cost.

CHAPTER 5

RESULTS

In this chapter the efficiency and performance of SIBI is evaluated to determine its competitveness with the conventional methods in the literature. For this purpose, singular integrations are performed for collections of triangles and triangle pairs that can be handled by SIBI. The calculations are done using the various versions of SIBI and their accuracy and efficiency is compared amongst them. The results show an improvement versus a conventional singular integration method. The singular interactions handled by SIBI account for a small percentage of the BEM matrix, but that does not necessarily extend to a small percentage of the assembly time; therefore the overall impact of SIBI in end-to-end results is investigated solving the scattering problem of a perfect electric conductor (PEC) sphere.

The calculations were done sequentially on MacOS 10.5, Intel(R) Xeon(R) Gold 6140 CPU @2.3GHz, with 190.5GB RAM, [86] and using double precision arithmetic. The SIBI algorithm calculates all of the interpolants for all common triangle, common edge interactions (see Table 2.4.2 and Figure 2.9) in the mesh and distributes them across the parametrized domain elements (see Figure 2.15). The interpolation is consequently calculated for all interpolants on each element. In the case of sparse grid SIBI (SG-SIBI) an interpolation matrix is constructed sequentially (although parallelizable) according to (3.6) - (3.8), which when multiplied by the multi-rhs (one column per integral function) completes the interpolation for the element. In the case of tensor train SIBI (TT-SIBI) or tensor Tucker decomposition (TD-SIBI) the interpolation is done sequentially for all interpolants and all integral functions according to (4.2, 4.4). Although similar to the SG-SIBI case the interpolation could be done altogether for all interpolants and integral functions at once as a matrix-matrix multiplication for improved computation efficiency. For each interaction, all the interpolated integral function results (see 2.4.2) are then collected from various different elements to fill up the BEM element matrix. The code was compiled by the Intel compiler. The libraries were constructed in element groups in parallel on a cluster of 10 such computers, although the calculations can be parallelized on an element level as well. The tensor decomposition was computed sequentially element after element using a Matlab tensor decomposition toolbox, [77], although it can be parallelized on an library level (in groups of elements).

5.1 Computational benchmarks

In this section SIBI is compared to the state-of-the-art implementation of the singularity subtraction technique, [37] in terms of accuracy, computational cost and memory overhead. The BEM element matrix entries of all common triangle or common edge interactions (other interactions are assumed to be zero) of two sample problems (meshes) are computed by the singularity subtraction method and the proposed method (SIBI) and the relative error is reported. The results presented are for perfect electric conducting (PEC) a drone UAV in free space, of typical discretization and good quality as well as a smaller BEM mesh of a PEC air intake (cavity) of typical discretization and poor quality, shown in Figs. 5.1a, 5.2a respectively. The quality of the meshes is assensed by the triangle quality factor, [87]:

$$Q = \frac{4A\sqrt{3}}{\ell_1^2 + \ell_2^2 + \ell_3^2} \tag{5.1}$$

where A: is the triangle area, ℓ: are the triangle edge lengths. Other definition of the triangle quality factor exist, but typically produce similar results are (5.1). A

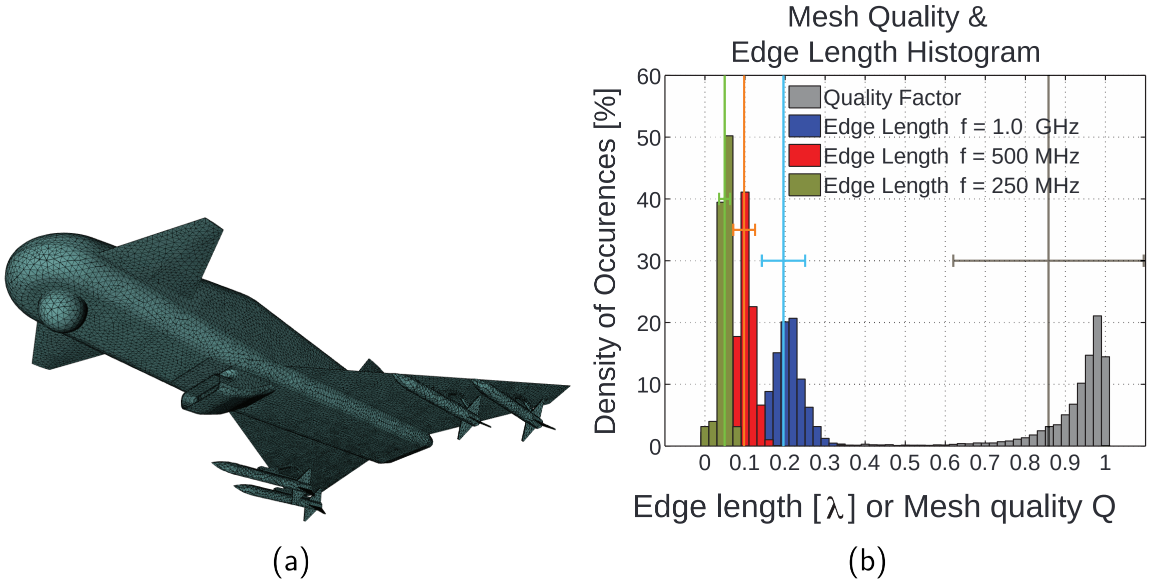
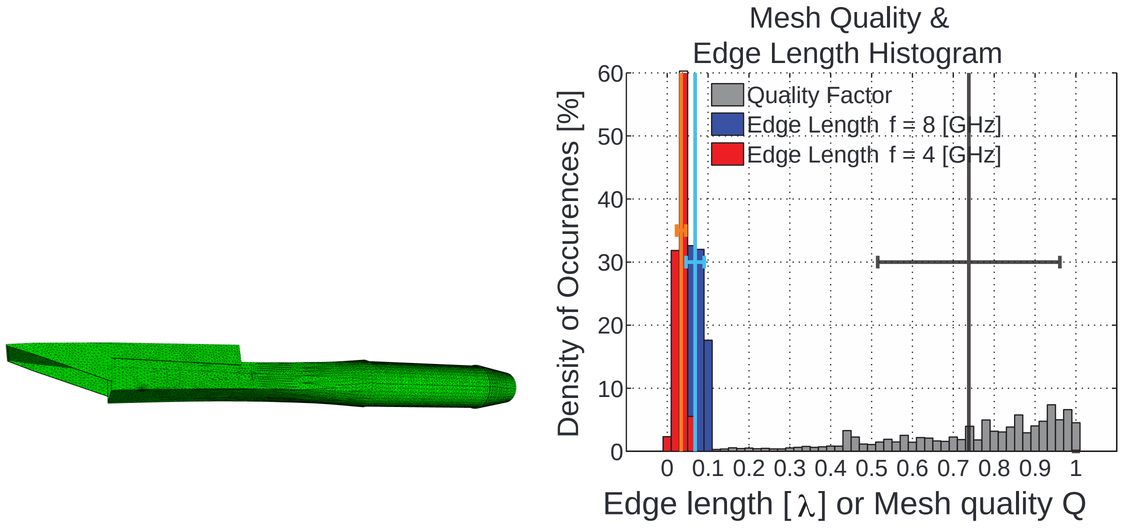


Figure 5.1. Mesh statistics showing histograms of the triangle quality factor and edge electrical lengths for frequencies 250MHz, 500MHz, 1GHz.

quality factor of $Q = 1$ is ideal as it represents an equilateral triangle, that has very good approximation properites, while a factor of $Q \to 0$ is a sliver-like triangle, that is undesirable as it has very poor approximation properties. Generally it is held that, if a triangle has $Q < 0.6$ it is considered to be of poor quality, [87] and thus it is advisable to be avoided in FEM or BEM commutations because the approximation properties are deteriorated, a phenomenon that is not related to integration approximation. However, in real life scenarios poor quality meshes like this may arise, and thus we have decided to use this challenging case as a benchmark. Figs. 5.1bb, 5.2b show histograms of the triangle quality factor of the mesh's triangles and the edge electrical length for 250MHz, 500MHz and 1GHz. In Fig. 5.1b about 20% of triangles has triangle quality factor $Q = 0.98$ indicating a very good mesh quality.

In this section, the various versions of SIBI are compared to two singularity subtractions rules, as shown for example in figures starting with Fig. 5.3. The first is a typical singularity subtraction rule of $(7, 7, 73, 1)$. The quadruplets indicate the number of samples in the inner and outer quadrature rules of the subtracted term,



(a) (b)

Figure 5.2. Mesh statistics showing histograms of the triangle quality factor and edge electrical lengths for frequencies 4GHz, 8GHz.

followed by the outer quadrature rule of the analytical term and lastly by the order of singularity subtraction (similar performance has been observed when compared to second order singularity subtraction). The second rule is a 'high accuracy' rule with $(73, 73, 73, 1)$. In these plots red, blue and green lines represent how frequently a particular error (horizontal axis) occurred in the mesh. This error is the relative error for all BEM matrix entries that involve singular integration (common triangle or edge, depending on the case plotted). The reference evaluate is considered a singularity subtraction method with 2 subtraction terms and quadruple perversion integrations i.e. $(166, 166, 166, 2)$. These type of plots, are considerably more comprehensive that the typical ones found in literature, because it gives a picture of the error for all singular interactions in the mesh, including one with poor quality triangles, typically presented as tails in those distributions. Vertical dashed lines of lighter hue show the mean value of each method. In the error histograms for the EFIE common triangle case, shown in Fig. 5.3 the proposed approach is two orders of magnitude more accurate, while only the maximum error of the higher order singularity subtraction rule

$(73, 73, 73, 1)$ is a little better than that of the proposed. The case of delayed sparse grids, shown in Fig. 5.4 further improves the accuracy of the proposed method, by one to two orders of magnitude.

The performance of the SG-SIBI is significantly degraded in common edge cases, due to the higher dimensionality of the interpolation problem; as shown in Fig. 5.5a, 5.11a the accuracy of the proposed method is not as good as that of the singularity subtraction rules, although it is still at an acceptable level of about $3 - 4$ decimal digits in average. The delayed sparse grids SIBI (DSG-SIBI), shown in Fig. 5.6a, 5.12a does not improve the accuracy.

The tensor train SIBI (TT-SIBI) approach outlined in Chapter 4 however, has imporved accuracy. Results of TT-SIBI are shown in Figs. 5.9a, 5.10a, 5.15a, 5.16a. For all these cases the TT-SIBI has similar accuracy as singularity subtraction and in some outperforms it. But it is noted that the tails (worst quality triangles) are still a bit heavier than singularity subtraction indicating that in these cases TT-SIBI should be used with some caution. The tensor Tucker decomposition approach shown in Figs. 5.8, 5.14 shows a similar improvement. In the MFIE case, which can be seen in Fig. 5.5b, 5.11b, the proposed method outperforms singularity subtraction regarding accuracy and an improvement of up to one order of magnitude in both average and maximum error is achieved. This fact is likely attributed to the stronger singular behavior of the integrands, which makes accurate integration all the more challenging. Note that in the case of MFIE coplanar common edge interactions yield a zero element matrix and thus these interactions are filtered out of the error histograms, which is why the number of samples differ from left to right in Fig. 5.5, 5.11. Moreover, the proposed method handles near-coplanar interactions better than singularity subtraction and is generally about an order of magnitude more accurate. Delayed sparse grids, shown in Fig. 5.6b, 5.12b, hardly improves accuracy, while tensor train, shown in Fig. 5.9b, 5.10b, 5.15b, 5.16b, can improve accuracy up to one order of magnitude.

In this section we are comparing the computational performance of the various SIBI implementations with the singularity subtraction methods outlined in the above. In the Tables 5.1, 5.2 the proposed method is compared in terms of computational and memory cost; the average and maximum error. "Memory" refers to the loaded library size of the proposed method given in MB, while "time", refers to the average run time to perform one singular integral normalized by the average run time to perform one far-field (regular) integral evaluated by a 7-point inner, 7-point outer $(7,7)$ Gauss quadrature rule. For example $t = 2[t_{FF}]$ is two times slower than the $(7,7)$ Gaussian quadrature rule. This normalization is used to factor out the computer hardware and performance, thus providing a more universal benchmark metric. Note that on each column the best entry/method with regards to average error, maximum error, memory or computation cost is shown in bold. The tables show that in the EFIE common triangle case, the delayed sparse grids SIBI is the most efficient method. In comparison to the state-of-the-art, although the maximum error is almost unchanged, the average error is about $1-2$ decimal digits improved and the computational cost improvement is over 200 fold, while the memory cost of 2.2MB is negligible. Note that the proposed method is about $4-6$ times faster than even the far-field rule! Since SIBI's performance depends on the parametrization dimension, the method is expected to perform worst for the common edge cases that involve $6D$ interpolation. For the both EFIE and MFIE common edge case time and memory cost have increased significantly, while the accuracy has decreased. However, the proposed approach still offers an acceptable accuracy of 4 decimal digits in average (about 1 decimal digit lower than state-of-the-art) at a computational time that is about 47 times faster than that of the typical singularity subtraction rule. Delayed sparse grids further improve computational cost, but do not solve the accuracy problem, whereas tensor train SIBI

(TT-SIBI) significantly improves the accuracy, by about one order of magnitude at the cost of somewhat slower runs. In addition, the memory cost increases as well with tolerance, while the accuracy improvement from 10^{-4} to 10^{-5} is not too great. The Tucker decomposition SIBI (TD-SIBI) on the other hand offers marginal accuracy improvement and slower run times. Thus, in the case of EFIE common edge the most efficient method appears to be tensor train SIBI with tolerance 10^{-4}, since it offers accuracy comparable to state-of-the-art singularity subtraction at an improved by about 20 times fold computational cost. In the case of MFIE common edge however, despite the higher dimensionality all versions of SIBI (SG-SIBI, DSG-SIBI, TT-SIBI) improve accuracy by about 1 decimal digit, as well as computational cost by 24 times fold. It is noted that the accuracy of the singularity subtraction rules has degraded severely, likely due to the stronger singular behavior of the integrand. Similarly to EFIE common edge case the delayed sparse grids approach does not improve accuracy, but it does improve computational cost and memory, while tensor train SIBI improves accuracy, while still offering better run times that singularity subraction.

To conclude, the optimal SIBI method for each case is as follows:

- DSG-SIBI for EFIE common triangle case, is the the fastest and most accurate method

- TT-SIBI ($tol = 10^{-4}$) for EFIE common edge case, is the best middle line between speed and accuracy; a little slower than SG-SIBI but considerably more accurate

- DSG-SIBI for MFIE common edge cas, is the fastest and most accurate method

5.2 Results for Start-to-End BEM computation for a PEC Sphere

In this section various sizes $(0.2\lambda, 2\lambda, 12\lambda)$ are considered. The geometry was chosen because the analytical Mie series solution is known, [4]. The proposed SIBI is integrated with an EFIE, MFIE and combined field IE (CFIE) BEM and compared with a singularity subtraction of $(7, 7, 82, 3)$. In the proposed approach, the common triangle cases were computed with delayed sparse grid SIBI, the common edge EFIE cases with tensor train SIBI of tolerance 10^-4, and common edge MFIE with Sparse Grid SIBI.

Figures 5.17, 5.18 and 5.19 show the radar cross section of the sphere with sizes $0.2\lambda, 2\lambda$ and 12λ respectively. On the left is the result of the analytical solution, Mie series and on the right is the relative error of the BEM result, with respect to the analytical solution, using the singularity subtraction BEM and SIBI BEM. The results show that the BEM result is $2-3$ decimal digits accurate, while the conventional and proposed approach yield about the same accuracy. In the previous sections the results showed that the proposed approach is almost as accurate, if not more accurate than the state-of-the-art singularity subtraction and therefore this similarity in results is to be expected. Keep in mind that the proposed approach cannot deal with all near field singular interactions yet; one such example is the common vertex case, which would require $9-10$D interpolation and therefore poses an even bigger challenge than common edge (6D interpolation). In Table 5.3 the proposed approach solution of the sphere problem is compared to the conventional one in terms accuracy, memory and computation time. In this case time refers to the near field assembly time and was recorded in seconds, when the problems were run sequentially on the hardware Intel(R) Xeon(R) Gold 6140 CPU @2.3GHz, with 190.5GB RAM, [86]. It is noted that the far field were computed with the single level Fast Multipole Method (FMM), [25] for the problems of sphere size $2\lambda, 12\lambda$. The memory cost is increased by a small amount in the proposed approach, which becomes negligible with increasing problem size, while the assembly / setup time is improved by about $11 - 20\%$; this time improvement surprisingly becomes quite smaller, around $2 - 3\%$ in the CFIE case however.

In conclusion, SIBI improves the accuracy and speed of singular integration in the cases of common triangle and common edge interactions. SIBI is general, in that it can compute the integrals of any kernel and even more for any arbitrarily shaped triangle or pair of triangles, regardless of the triangle quality factor. It is likely that SIBI is more accurate than conventional integration for poor quality triangles; keep in mind that the reference for these integrations was a high accuracy conventional method. However, the cost of SIBI scales exponentially with the dimensionality of the parametrized domain, deteriorating its performance.

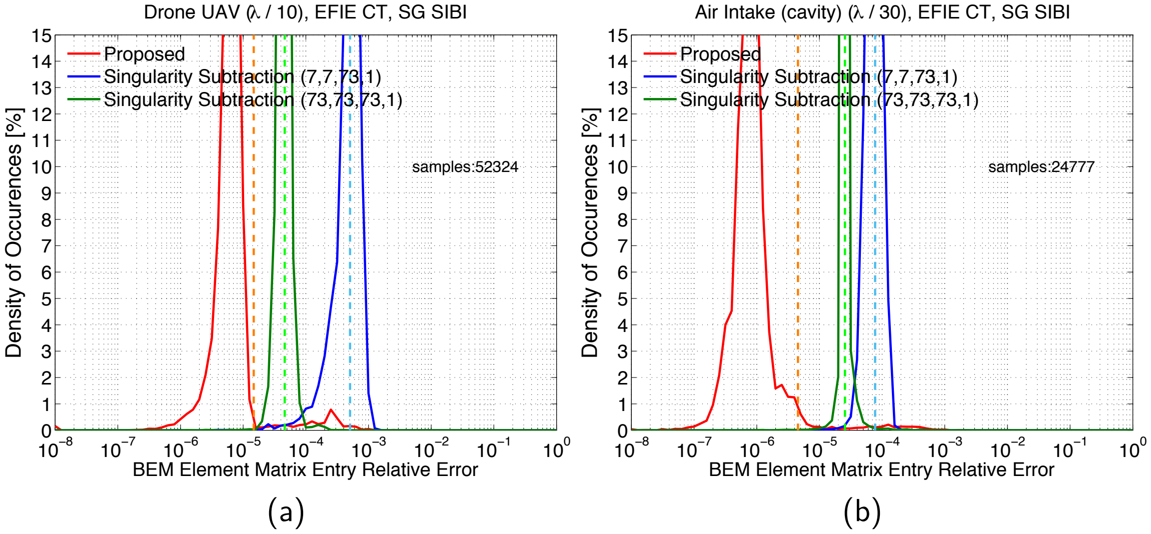
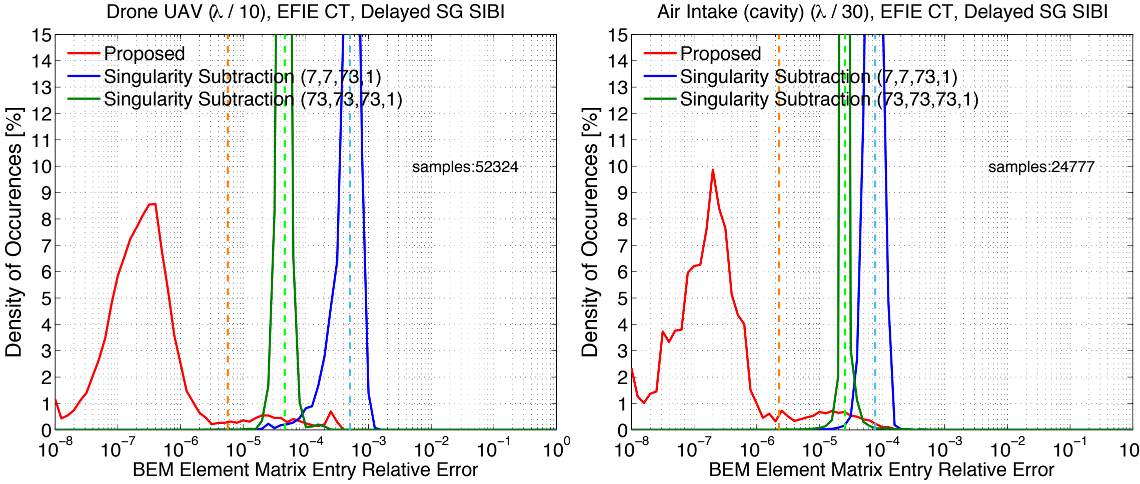
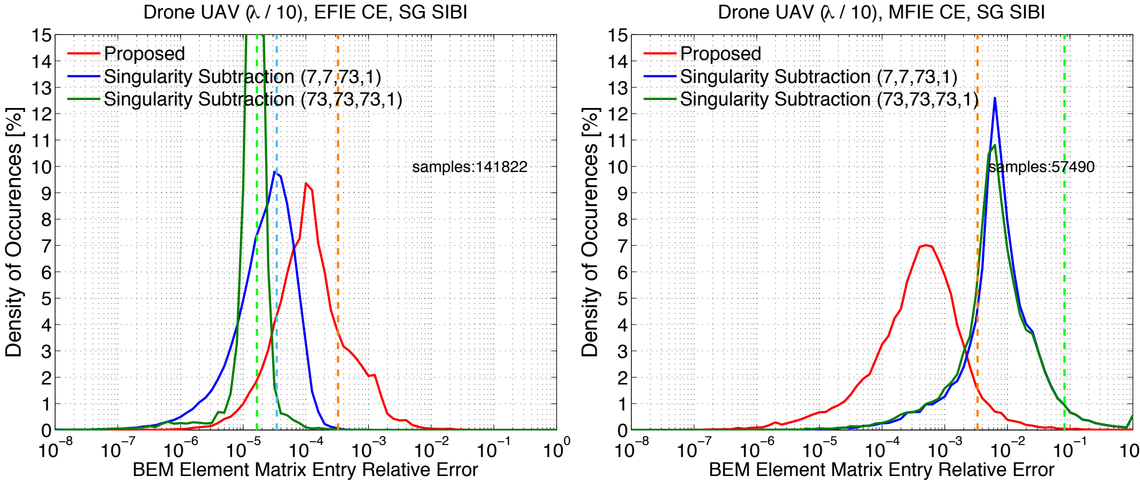


Figure 5.3. EFIE common triangle case for **Sparse Grid SIBI (SG-SIBI)**. Plots show histograms of the relative error in each entry of the BEM matrix that involves common triangle integration. The SG-SIBI is compared with a 'typical' version of singularity subtraction using $(7, 7, 73, 1)$ rule (see text for details) and a 'high accuracy' $(73, 73, 73, 1)$ rule. a) drone UAV mesh with average length $\lambda/10$ (500MHz) b) air intake (cavity) mesh with average length $\lambda/30$ (4GHz).



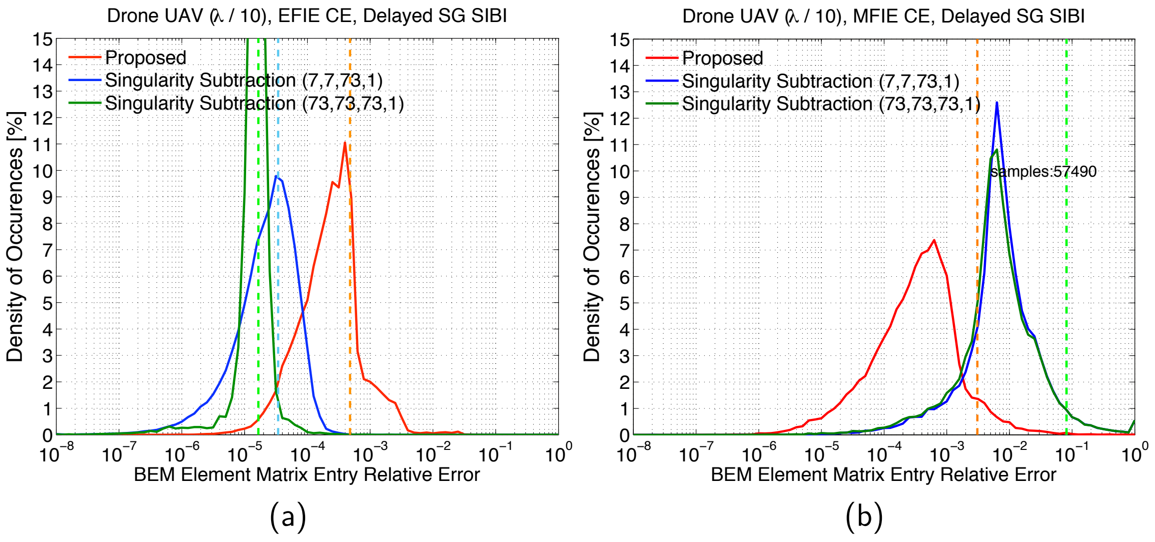
(a) (b)

Figure 5.4. EFIE common triangle case for **Delayed Sparse Grid SIBI (DSG-SIBI)**. Plots show histograms of the relative error in each entry of the BEM matrix that involves common triangle integration. The DSG-SIBI is compared with a 'typical' version of singularity subtraction using $(7, 7, 73, 1)$ rule (see text for details) and a 'high accuracy' $(73, 73, 73, 1)$ rule. a) drone UAV mesh with average length $\lambda/10$ (500MHz) b) air intake (cavity) mesh with average length $\lambda/30$ (4GHz).



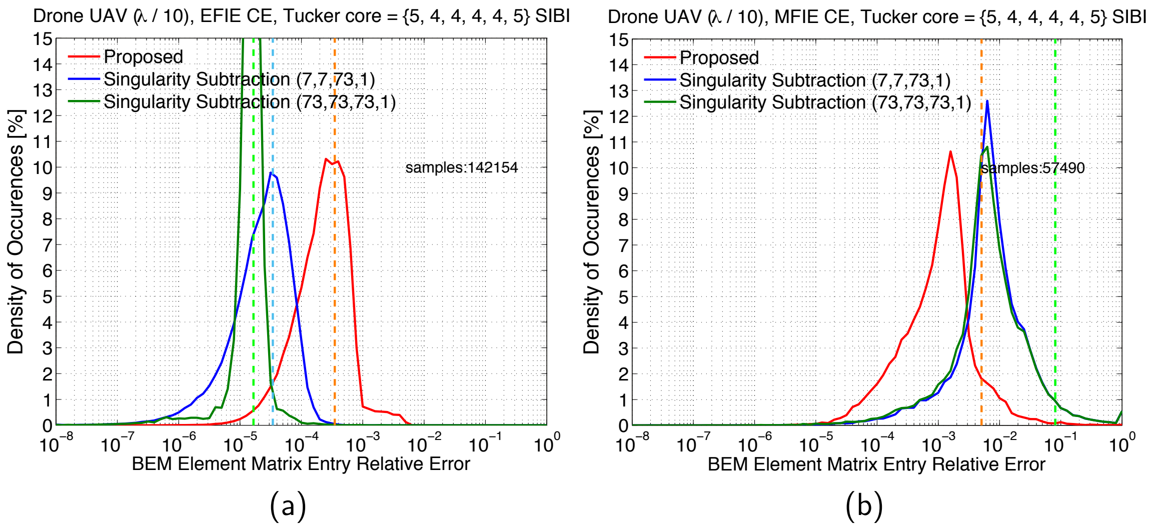
(a) (b)

Figure 5.5. Common edge case for **Sparse Grid SIBI (SG-SIBI)** for Drone UAV problem. Plots show histograms of the relative error in each entry of the BEM matrix that involves common edge integration. (a) EFIE common edge case. (b) MFIE common edge case.



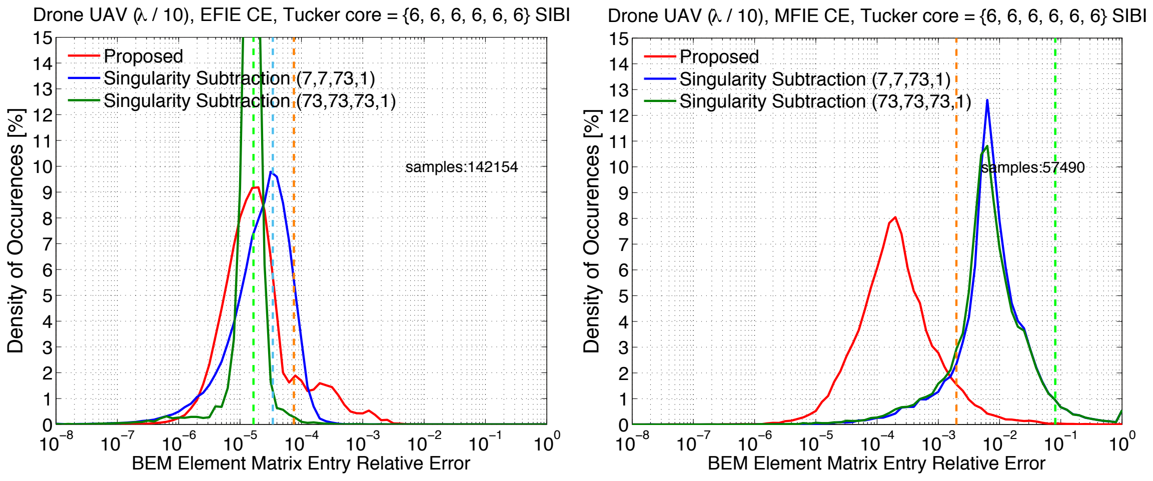
(a) (b)

Figure 5.6. Common edge case for **Delayed Sparse Grid SIBI (DSG-SIBI)** for Drone UAV problem. Plots show histograms of the relative error in each entry of the BEM matrix that involves common edge integration. (a) EFIE common edge case. (b) MFIE common edge case.



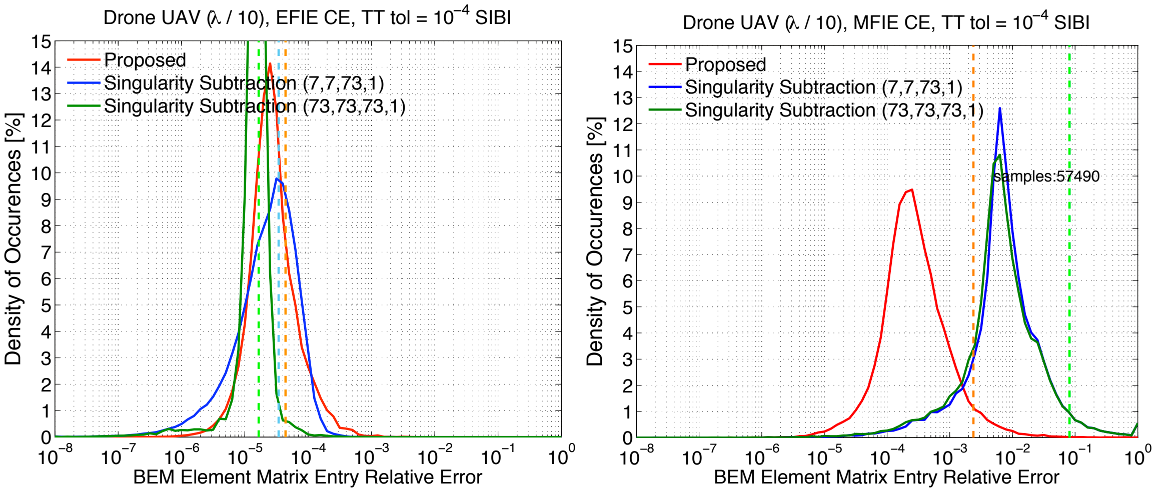
(a) (b)

Figure 5.7. Common edge case for **Tucker decomposition SIBI (TD-SIBI)** for Drone UAV problem. Decomposition core used is $\{5, 4, 4, 4, 4, 5\}$. Plots show histograms of the relative error in each entry of the BEM matrix that involves common edge integration. (a) EFIE common edge case. (b) MFIE common edge case.



(a) (b)

Figure 5.8. Common edge case for **Tucker decomposition SIBI (TD-SIBI)** for Drone UAV problem. Decomposition core used is $\{6, 6, 6, 6, 6, 6\}$. Plots show histograms of the relative error in each entry of the BEM matrix that involves common edge integration. (a) EFIE common edge case. (b) MFIE common edge case.



(a) (b)

Figure 5.9. Common edge case for **Tensor Train SIBI (TT-SIBI)** for Drone UAV problem. Decomposition tolerance used is 10^{-4}. Plots show histograms of the relative error in each entry of the BEM matrix that involves common edge integration. (a) EFIE common edge case. (b) MFIE common edge case.

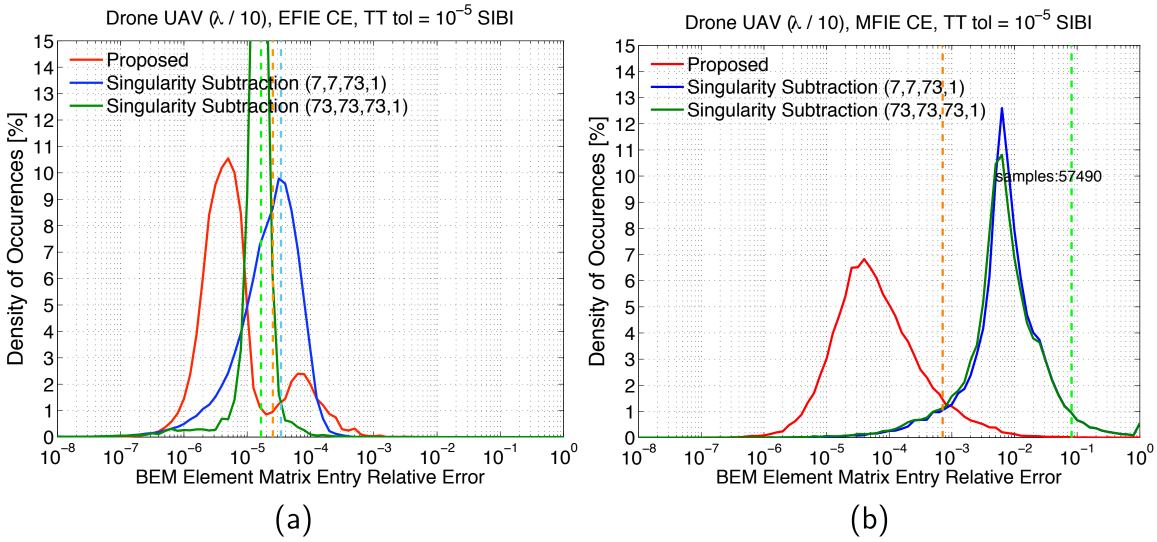


Figure 5.10. Common edge case for **Tensor Train SIBI (TT-SIBI)** for Drone UAV problem. Decomposition tolerance used is 10^{-5}. Plots show histograms of the relative error in each entry of the BEM matrix that involves common edge integration. (a) EFIE common edge case. (b) MFIE common edge case.

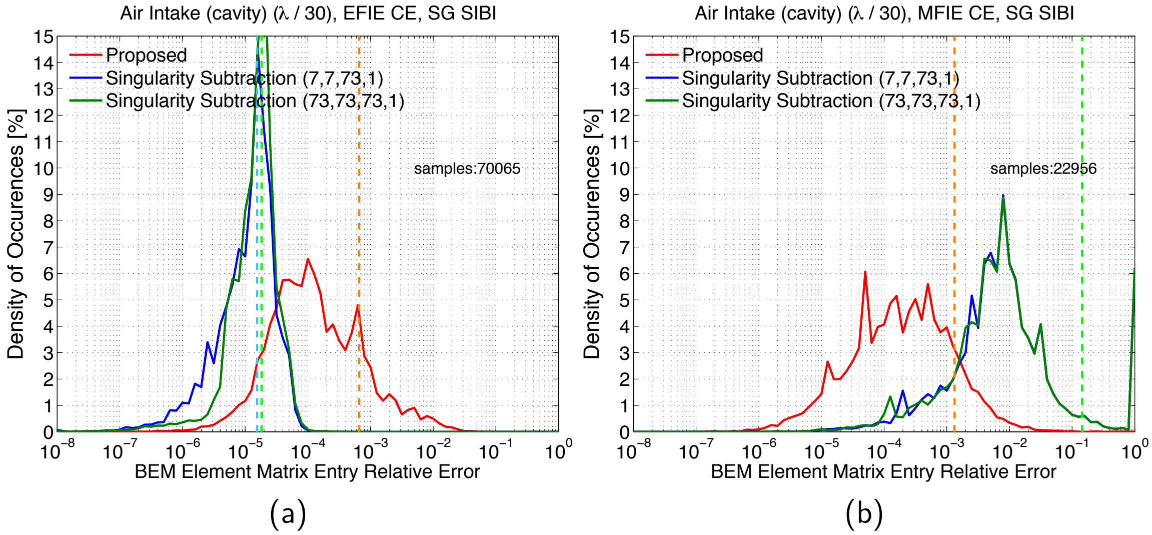
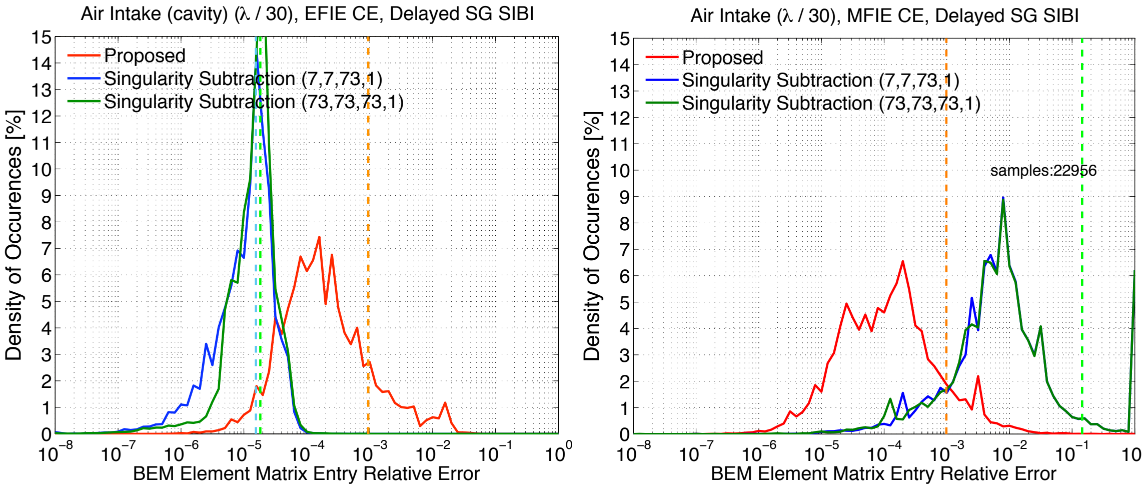


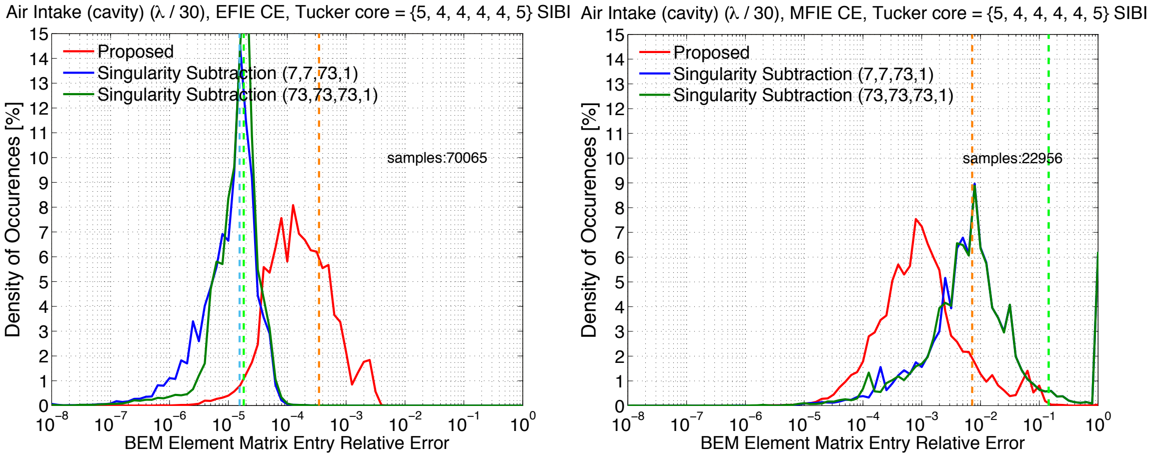
Figure 5.11. Common edge case for **Sparse Grid SIBI (SG-SIBI)** for Air Intake (cavity) problem. Plots show histograms of the relative error in each entry of the BEM matrix that involves common edge integration. (a) EFIE common edge case. (b) MFIE common edge case.



(a)

(b)

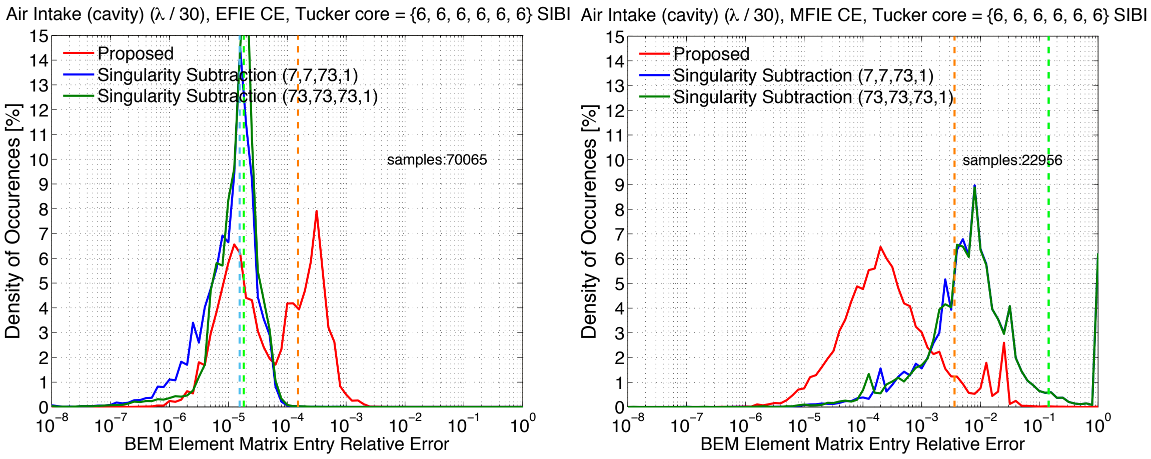
Figure 5.12. Common edge case for **Delayed Sparse Grid SIBI (DSG-SIBI)** for Air Intake (cavity) problem. Plots show histograms of the relative error in each entry of the BEM matrix that involves common edge integration. (a) EFIE common edge case. (b) MFIE common edge case.



(a)

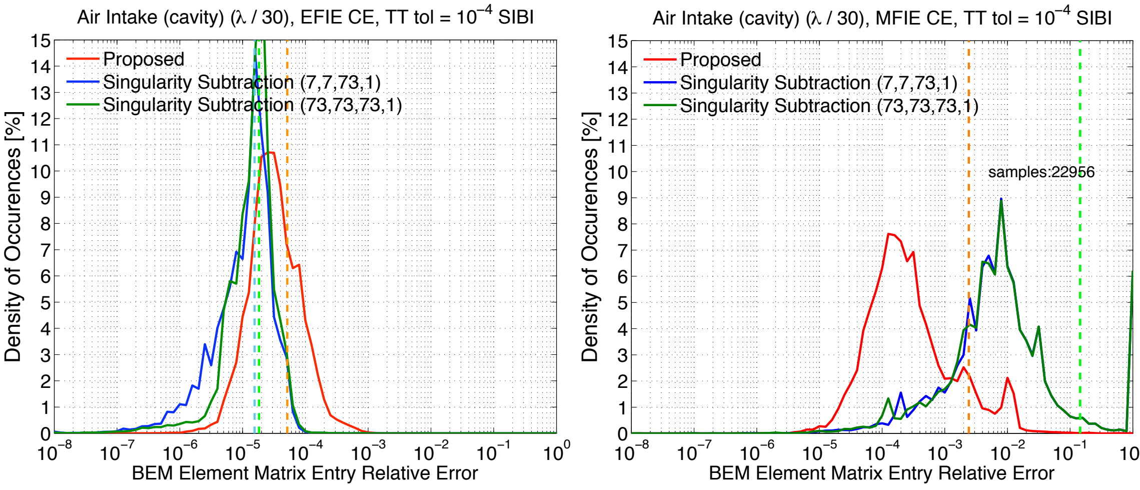
(b)

Figure 5.13. Common edge case for **Tucker decomposition SIBI (TD-SIBI)** for Air Intake (cavity) problem. Decomposition core used is $\{5, 4, 4, 4, 4, 5\}$. Plots show histograms of the relative error in each entry of the BEM matrix that involves common edge integration. (a) EFIE common edge case. (b) MFIE common edge case.



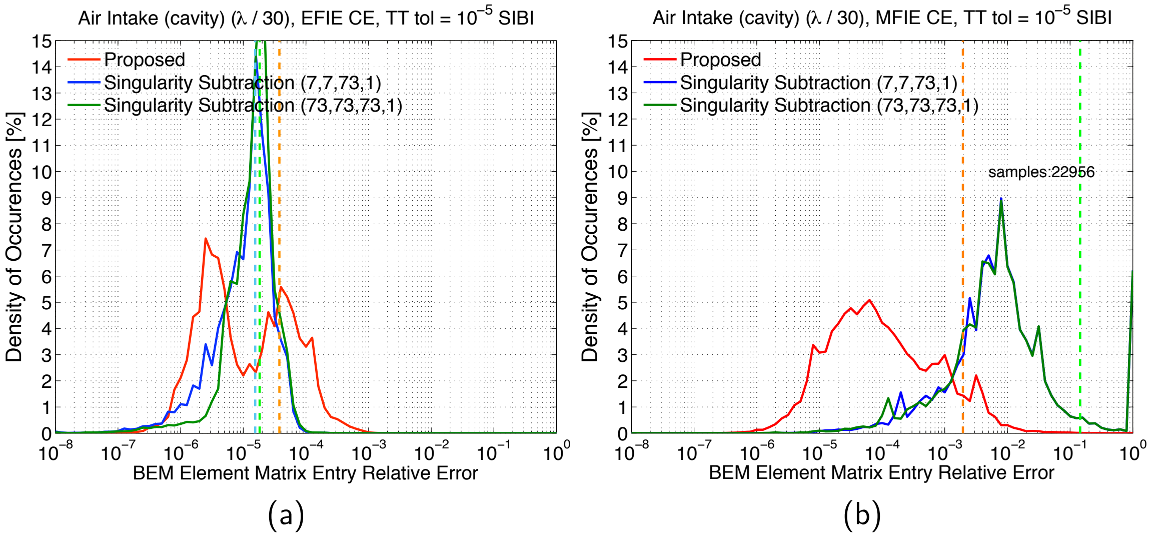
(a) (b)

Figure 5.14. Common edge case for **Tucker decomposition SIBI (TD-SIBI)** for Air Intake (cavity) problem. Decomposition core used is $\{6, 6, 6, 6, 6, 6\}$. Plots show histograms of the relative error in each entry of the BEM matrix that involves common edge integration. (a) EFIE common edge case. (b) MFIE common edge case.



(a) (b)

Figure 5.15. Common edge case for **Tensor Train SIBI (TT-SIBI)** for Air Intake (cavity) problem. Decomposition tolerance used is 10^{-4}. Plots show histograms of the relative error in each entry of the BEM matrix that involves common edge integration. (a) EFIE common edge case. (b) MFIE common edge case.



(a)

(b)

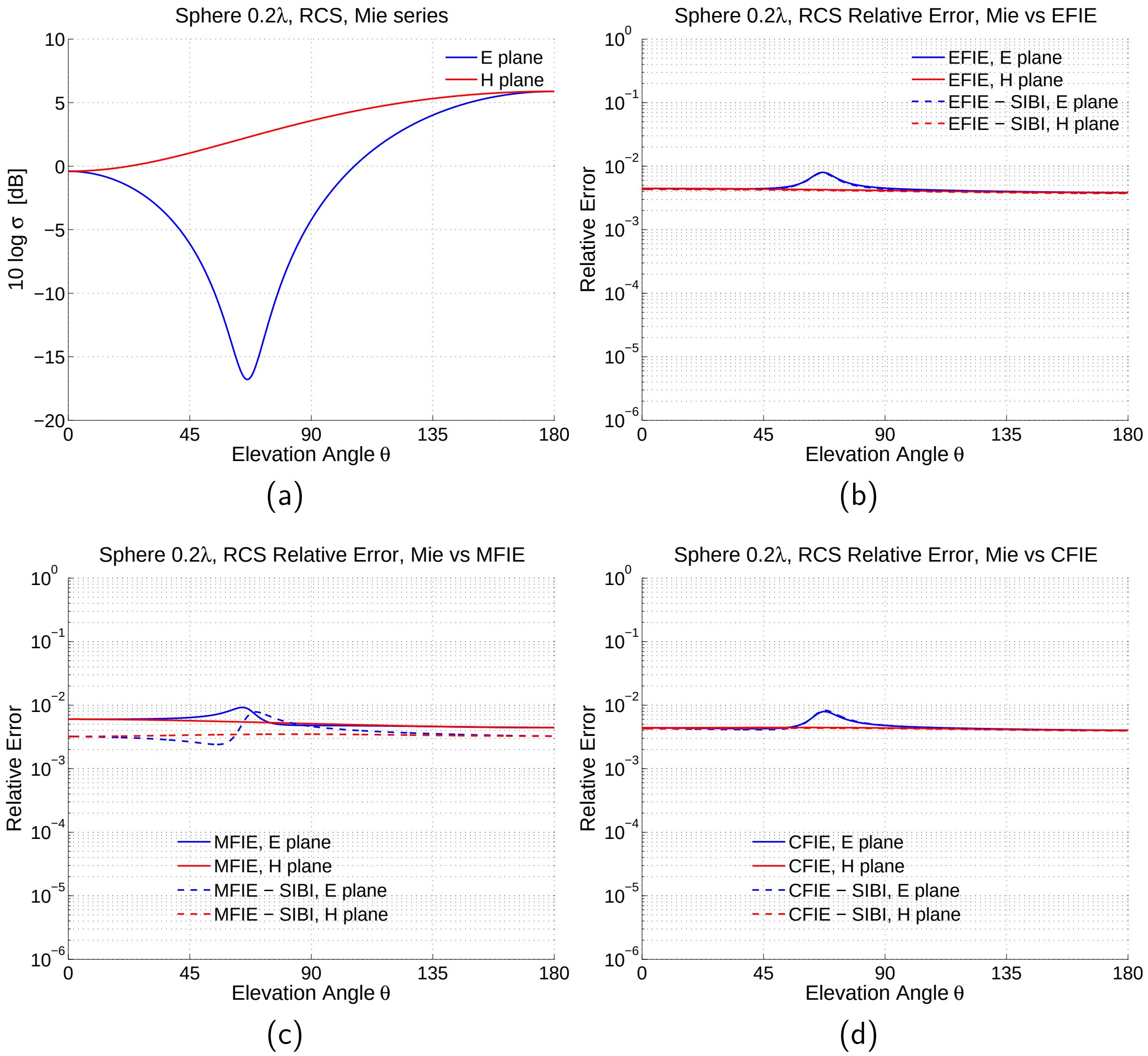
Figure 5.16. Common edge case for **Tensor Train SIBI (TT-SIBI)** for Air Intake (cavity) problem. Decomposition tolerance used is 10^{-5}. Plots show histograms of the relative error in each entry of the BEM matrix that involves common edge integration. (a) EFIE common edge case. (b) MFIE common edge case.

Table 5.1. Error, memory and run time for Drone UAV ($\lambda/10, 500MHz$)

Drone UAV ($\lambda/10, 500MHz$)		Error		Memory	Time
Equation	Method	mean	max	[MB]	[t_{FF}]
EFIE CT	Sparse Grid SIBI	$1.5 \cdot 10^{-5}$	$2.8 \cdot 10^{-3}$	2.3	0.25
	Delayed Sparse Grid SIBI	$\mathbf{5.7 \cdot 10^{-6}}$	$8.0 \cdot 10^{-4}$	2.2	**0.16**
	Singularity Subtraction $(7, 7, 73, 1)$	$5.0 \cdot 10^{-4}$	$1.6 \cdot 10^{-3}$	**0**	48.3
	Singularity Subtraction $(73, 73, 73, 1)$	$4.6 \cdot 10^{-5}$	$\mathbf{4.0 \cdot 10^{-4}}$	**0**	76.7
EFIE CE	Sparse Grid SIBI	$3.3 \cdot 10^{-4}$	$5.7 \cdot 10^{-2}$	488.5	1.0
	Delayed Sparse Grid SIBI	$4.8 \cdot 10^{-4}$	$5.7 \cdot 10^{-2}$	360.3	**0.72**
	Tensor Tucker SIBI (core rank $\{5, 4, 4, 4, 4, 5\}$)	$3.5 \cdot 10^{-4}$	$7.0 \cdot 10^{-3}$	85.9	3.3
	Tensor Tucker SIBI (core rank $\{6, 6, 6, 6, 6, 6\}$)	$7.5 \cdot 10^{-5}$	$5.7 \cdot 10^{-3}$	290.6	20.7
	Tensor Train SIBI (tol 10^{-4})	$4.4 \cdot 10^{-5}$	$4.0 \cdot 10^{-3}$	39.3	2.0
	Tensor Train SIBI (tol 10^{-5})	$2.5 \cdot 10^{-5}$	$4.0 \cdot 10^{-3}$	93.3	3.7
	Singularity Subtraction $(7, 7, 73, 1)$	$3.4 \cdot 10^{-5}$	$2.0 \cdot 10^{-3}$	**0**	47.0
	Singularity Subtraction $(73, 73, 73, 1)$	$\mathbf{1.7 \cdot 10^{-5}}$	$\mathbf{2.0 \cdot 10^{-3}}$	**0**	75.3
MFIE CE	Sparse Grid SIBI	$3.3 \cdot 10^{-3}$	77.1	227	1.0
	Delayed Sparse Grid SIBI	$3.0 \cdot 10^{-3}$	61.3	167.4	**0.43**
	Tensor Tucker SIBI (core rank $\{5, 4, 4, 4, 4, 5\}$)	$5.1 \cdot 10^{-3}$	48.7	68.7	2.8
	Tensor Tucker SIBI (core rank $\{6, 6, 6, 6, 6, 6\}$)	$2.0 \cdot 10^{-3}$	22.1	484.8	19.7
	Tensor Train SIBI (tol 10^{-3})	$1.3 \cdot 10^{-2}$	288.5	29.9	1.8
	Tensor Train SIBI (tol 10^{-4})	$2.4 \cdot 10^{-3}$	71.6	76.8	3.1
	Tensor Train SIBI (tol 10^{-5})	$\mathbf{7.0 \cdot 10^{-4}}$	**6.6**	139.5	5.0
	Singularity Subtraction $(7, 7, 73, 1)$	$8.1 \cdot 10^{-2}$	2199	**0**	24.7
	Singularity Subtraction $(73, 73, 73, 1)$	$8.1 \cdot 10^{-2}$	2211	**0**	77.0

Table 5.2. Error, memory and run time for Air Intake (cavity) $(\lambda/30, 4GHz)$

Air Intake (cavity) $(\lambda/30, 4GHz)$		Error		Memory	Time
Equation	Method	mean	max	[MB]	$[t_{FF}]$
EFIE CT	Sparse Grid SIBI	$4.5 \cdot 10^{-6}$	$2.0 \cdot 10^{-3}$	2.3	0.31
	Delayed Sparse Grid SIBI	$\mathbf{2.3 \cdot 10^{-6}}$	$5.6 \cdot 10^{-4}$	2.2	**0.24**
	Singularity Subtraction $(7, 7, 73, 1)$	$7.7 \cdot 10^{-5}$	$\mathbf{5.6 \cdot 10^{-4}}$	**0**	49.0
	Singularity Subtraction $(73, 73, 73, 1)$	$2.5 \cdot 10^{-5}$	$5.8 \cdot 10^{-4}$	**0**	76.3
EFIE CE	Sparse Grid SIBI	$6.6 \cdot 10^{-4}$	$5.8 \cdot 10^{-2}$	488.5	1.1
	Delayed Sparse Grid SIBI	$9.5 \cdot 10^{-4}$	$6.3 \cdot 10^{-2}$	360.3	**0.74**
	Tensor Tucker SIBI (core rank $\{5, 4, 4, 4, 4, 5\}$)	$3.4 \cdot 10^{-4}$	$4.7 \cdot 10^{-3}$	85.9	3.4
	Tensor Tucker SIBI (core rank $\{6, 6, 6, 6, 6, 6\}$)	$1.5 \cdot 10^{-4}$	$4.0 \cdot 10^{-3}$	290.6	23.0
	Tensor Train SIBI (tol 10^{-4})	$5.1 \cdot 10^{-5}$	$4.0 \cdot 10^{-3}$	39.3	2.1
	Tensor Train SIBI (tol 10^{-5})	$3.8 \cdot 10^{-5}$	$4.1 \cdot 10^{-3}$	93.3	4.5
	Singularity Subtraction $(7, 7, 73, 1)$	$\mathbf{1.6 \cdot 10^{-5}}$	$\mathbf{8.0 \cdot 10^{-4}}$	**0**	45.0
	Singularity Subtraction $(73, 73, 73, 1)$	$1.8 \cdot 10^{-5}$	$8.0 \cdot 10^{-4}$	**0**	74.3
MFIE CE	Sparse Grid SIBI	$1.3 \cdot 10^{-3}$	5.8	227	1.1
	Delayed Sparse Grid SIBI	$\mathbf{9.8 \cdot 10^{-4}}$	**4.6**	167.4	**0.78**
	Tensor Tucker SIBI (core rank $\{5, 4, 4, 4, 4, 5\}$)	$7.2 \cdot 10^{-3}$	17.6	68.7	3.3
	Tensor Tucker SIBI (core rank $\{6, 6, 6, 6, 6, 6\}$)	$3.6 \cdot 10^{-3}$	27.7	484.8	20.5
	Tensor Train SIBI (tol 10^{-3})	$1.0 \cdot 10^{-2}$	31.5	29.9	1.7
	Tensor Train SIBI (tol 10^{-4})	$2.4 \cdot 10^{-3}$	22.5	76.8	3.5
	Tensor Train SIBI (tol 10^{-5})	$1.9 \cdot 10^{-3}$	25.1	139.5	5.8
	Singularity Subtraction $(7, 7, 73, 1)$	$1.4 \cdot 10^{-1}$	190.0	**0**	24.6
	Singularity Subtraction $(73, 73, 73, 1)$	$1.4 \cdot 10^{-1}$	190.1	**0**	76.5



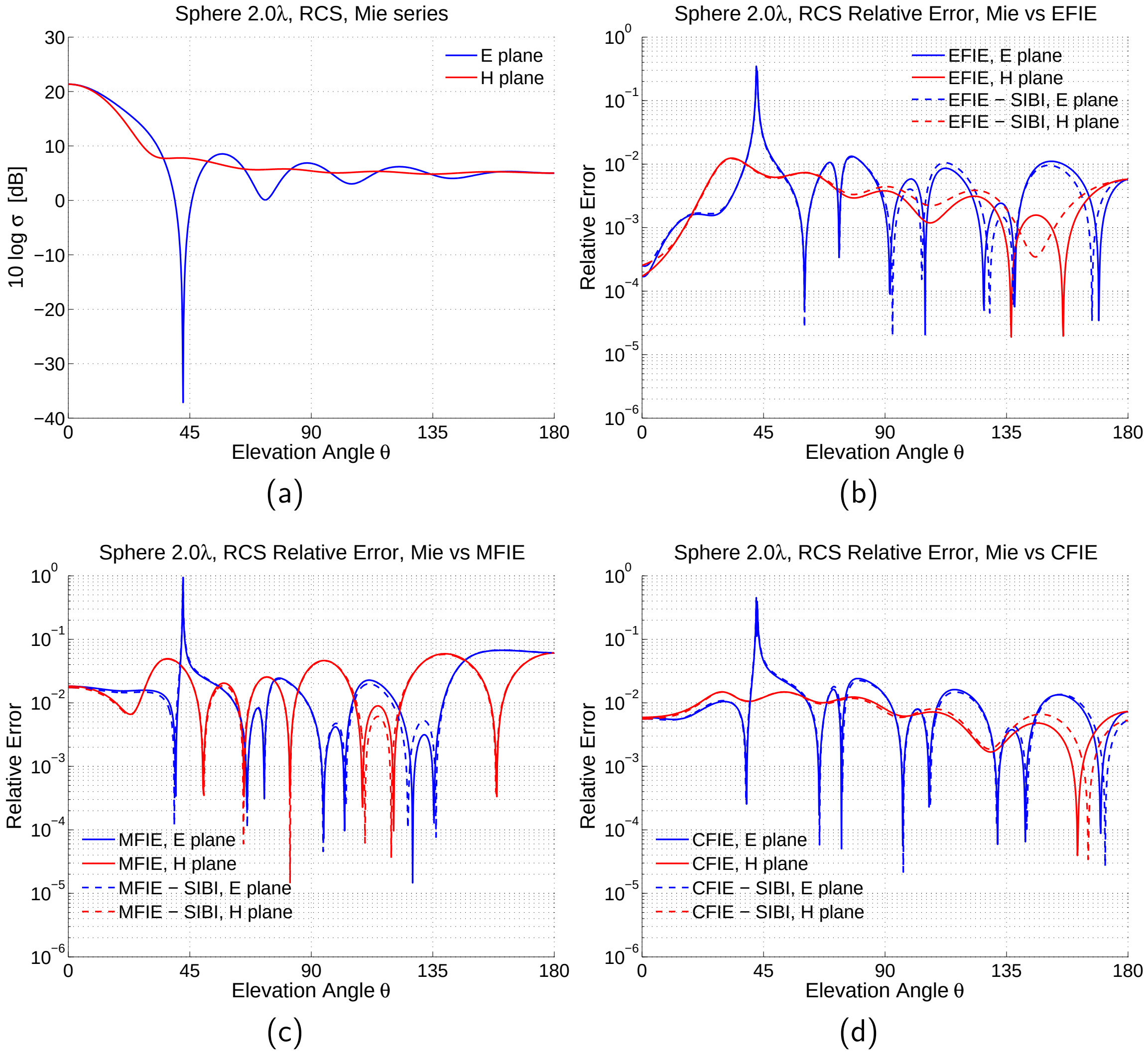
(a)

(b)

(c)

(d)

Figure 5.17. RCS of sphere mesh, size 0.2λ (30MHz) a) Mie series (reference) b) relative error of EFIE and EFIE-SIBI c) relative error of MFIE and MFIE-SIBI b) relative error of CFIE and CFIE-SIBI.



Sphere 2.0λ, RCS, Mie series

(a)

Sphere 2.0λ, RCS Relative Error, Mie vs EFIE

(b)

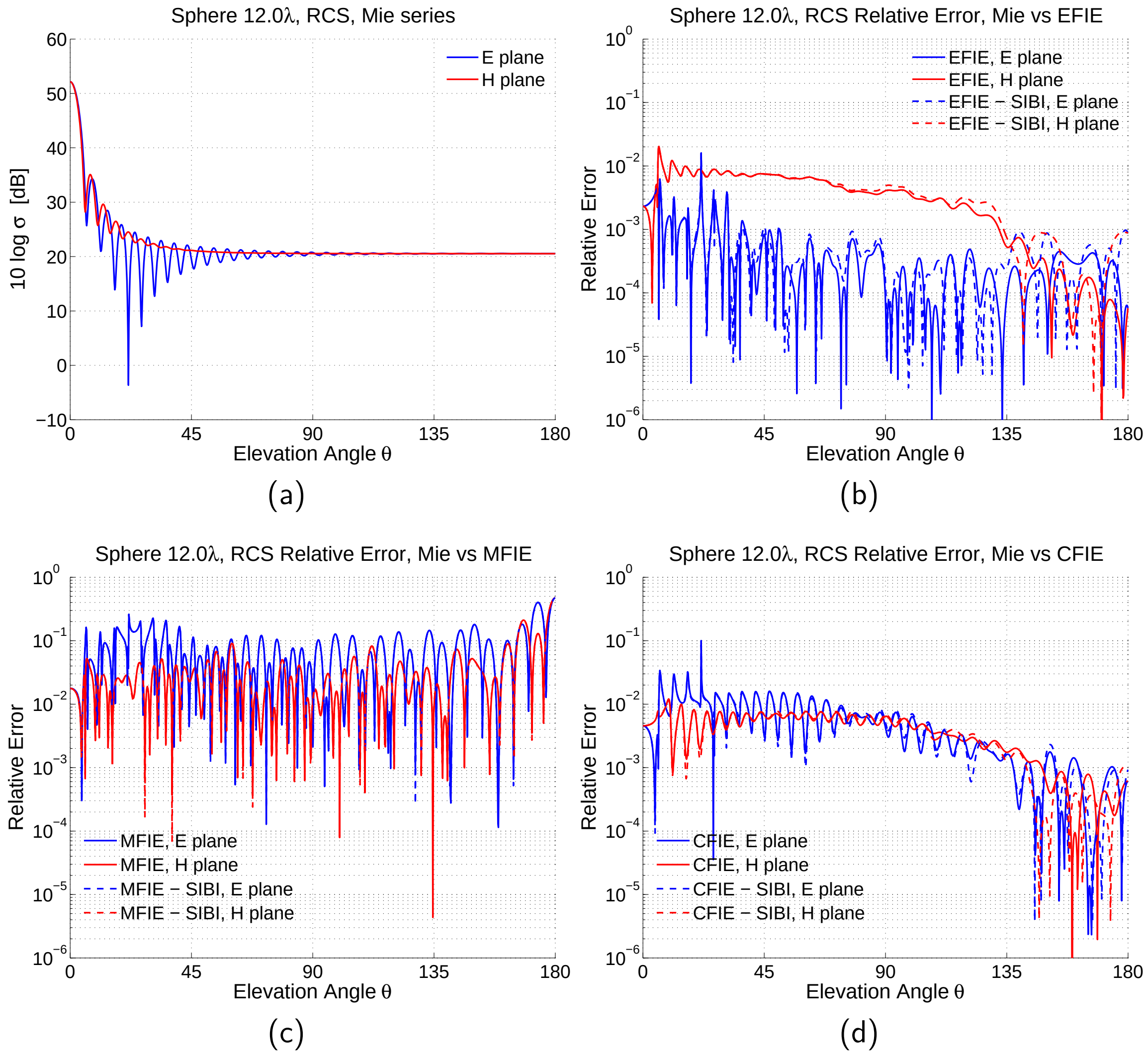
Sphere 2.0λ, RCS Relative Error, Mie vs MFIE

(c)

Sphere 2.0λ, RCS Relative Error, Mie vs CFIE

(d)

Figure 5.18. RCS of sphere mesh, size 2λ (300MHz) a) Mie series (reference) b) relative error of EFIE and EFIE-SIBI c) relative error of MFIE and MFIE-SIBI b) relative error of CFIE and CFIE-SIBI.



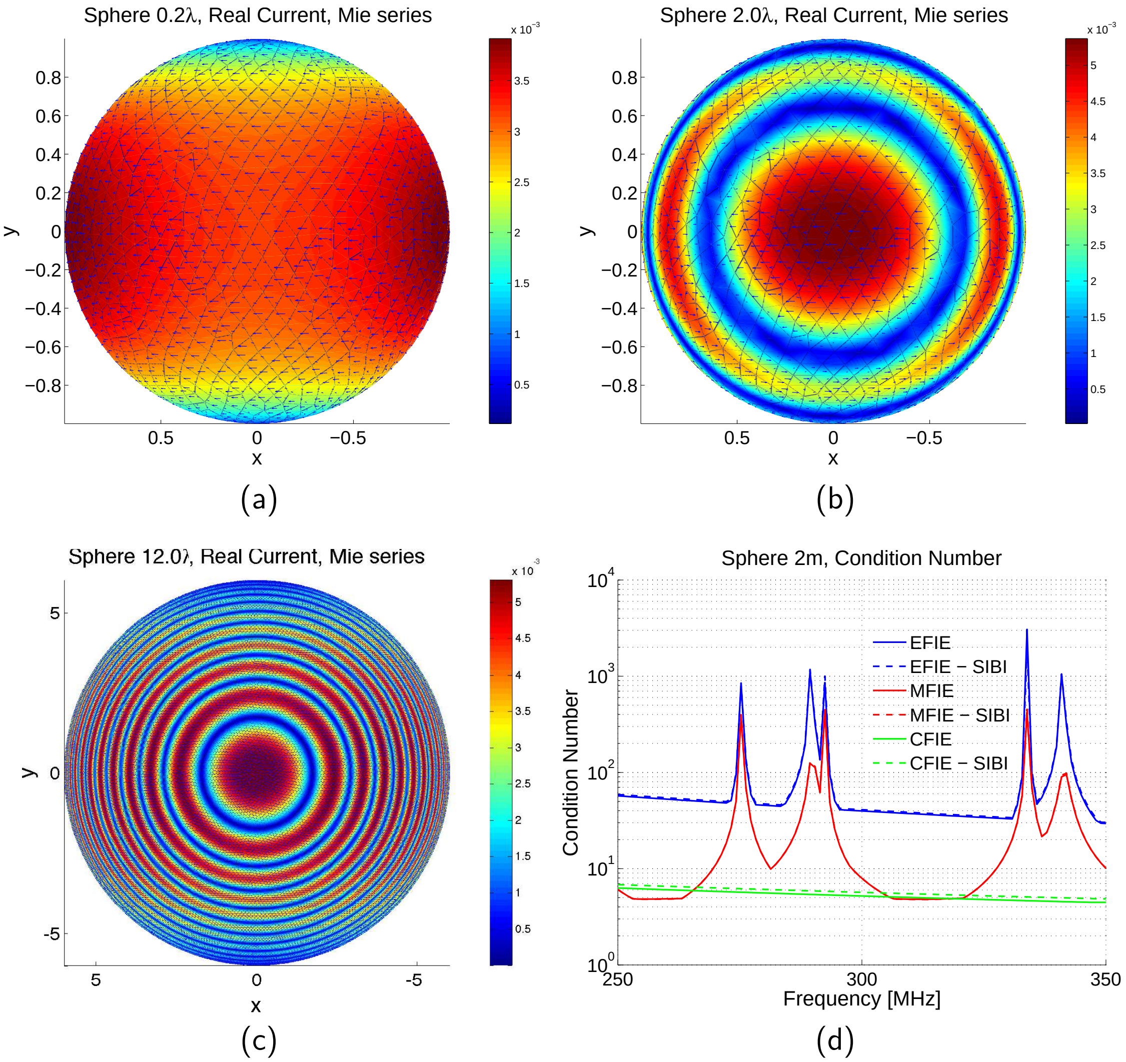
(a)

(b)

(c)

(d)

Figure 5.19. RCS of sphere mesh, size 12λ (300MHz) a) Mie series (reference) b) relative error of EFIE and EFIE-SIBI c) relative error of MFIE and MFIE-SIBI b) relative error of CFIE and CFIE-SIBI.



(a)

(b)

(c)

(d)

Figure 5.20. Current distributon (real part) on PEC sphere when illuminated by $x-$polarized plane wave, computed by the analytical Mie series for sizes a) 0.2λ b) 2λ c) 12λ d) BEM matrix condition number of sphere mesh; conventional vs SIBI

Table 5.3. Error, memory and assembly run time for sphere scattering problem when singular integrals were computed using SIBI and the singularity subtraction $(7, 7, 166, 3)$ conventional and proposed methods

Sphere Problem		RCS Error		Current Error		Memory	Time
Size	Method	mean	max	mean	max	[MB]	[sec]
0.2λ	EFIE	$4.3 \cdot 10^{-3}$	$8.0 \cdot 10^{-3}$	$9.9 \cdot 10^{-3}$	$3.7 \cdot 10^{-2}$	145	35.3
	EFIE - SIBI	$4.2 \cdot 10^{-3}$	$7.8 \cdot 10^{-3}$	$1.1 \cdot 10^{-2}$	$5.9 \cdot 10^{-2}$	214	31.3
	MFIE	$5.3 \cdot 10^{-3}$	$9.2 \cdot 10^{-3}$	$9.7 \cdot 10^{-3}$	$2.9 \cdot 10^{-2}$	234	79.8
	MFIE - SIBI	$3.5 \cdot 10^{-3}$	$7.8 \cdot 10^{-3}$	$9.7 \cdot 10^{-3}$	$3.1 \cdot 10^{-2}$	472	72.6
	CFIE	$4.4 \cdot 10^{-3}$	$7.9 \cdot 10^{-3}$	$9.7 \cdot 10^{-3}$	$3.6 \cdot 10^{-2}$	267	283.7
	CFIE - SIBI	$4.4 \cdot 10^{-3}$	$8.3 \cdot 10^{-3}$	$9.6 \cdot 10^{-3}$	$3.6 \cdot 10^{-2}$	559	286.9
2λ	EFIE	$5.4 \cdot 10^{-3}$	$3.5 \cdot 10^{-1}$	$1.0 \cdot 10^{-1}$	$7.4 \cdot 10^{-1}$	116	18.4
	EFIE - SIBI	$5.4 \cdot 10^{-3}$	$3.3 \cdot 10^{-1}$	$1.0 \cdot 10^{-1}$	$7.2 \cdot 10^{-1}$	197	13.6
	MFIE	$2.5 \cdot 10^{-2}$	$9.4 \cdot 10^{-1}$	$1.0 \cdot 10^{-1}$	$7.2 \cdot 10^{-1}$	286	79.5
	MFIE - SIBI	$2.5 \cdot 10^{-2}$	$9.4 \cdot 10^{-1}$	$1.0 \cdot 10^{-1}$	$7.1 \cdot 10^{-1}$	365	73.7
	CFIE	$9.9 \cdot 10^{-3}$	$4.1 \cdot 10^{-1}$	$1.0 \cdot 10^{-1}$	$6.7 \cdot 10^{-1}$	173	194.7
	CFIE - SIBI	$1.0 \cdot 10^{-2}$	$4.5 \cdot 10^{-1}$	$1.0 \cdot 10^{-1}$	$6.7 \cdot 10^{-1}$	478	187.6
12λ	EFIE	$2.2 \cdot 10^{-3}$	$2.0 \cdot 10^{-2}$	$9.6 \cdot 10^{-2}$	3.3	8672	1018.9
	EFIE - SIBI	$2.3 \cdot 10^{-3}$	$2.0 \cdot 10^{-2}$	$9.6 \cdot 10^{-2}$	3.1	9404	878.9
	MFIE	$5.8 \cdot 10^{-2}$	$4.7 \cdot 10^{-1}$	$1.8 \cdot 10^{-1}$	10.7	10401	2006.9
	MFIE - SIBI	$5.8 \cdot 10^{-2}$	$4.7 \cdot 10^{-1}$	$1.8 \cdot 10^{-1}$	10.7	11061	1813.2
	CFIE	$4.8 \cdot 10^{-3}$	$1.0 \cdot 10^{-1}$	$8.9 \cdot 10^{-2}$	1.3	13811	12718.7
	CFIE - SIBI	$4.8 \cdot 10^{-3}$	$8.5 \cdot 10^{-2}$	$8.9 \cdot 10^{-2}$	1.3	14590	12468.2

CHAPTER 6

EPILOGUE

Efficient and accurate singular integration encountered in the BEM solution of integral equations (EFIE, MFIE) is a challenging research topic. Even the most prominent of techniques, the state-of-the-art, such as singularity subtraction, singularity cancellation and full 4D integral evaluation greatly compromise computational cost/speed in favor of decent accuracy. This dissertation proposes an unconventional approach, where construction of *universal* libraries/look-up tables of integrals precomputed to high precision, while disregarding computational cost is sought at an off-line stage, that is performed once. Subsequently the integral is recovered by means of multidimensional interpolation, which governs the computational cost as well as the accuracy of the overall method. This singular integration by interpolation (SIBI) apporach was combined with advance multidimensional interpolation/sampling steerages such as sparse grids, delayed sparse grids, low-rank tensor train decompositions to accelerate the $6D$ integrations encountered in the common edge singular case. Moreover, all the memory costs due to the loaded libraries reported are generally not significant. However, the speed and accuracy of SIBI is superior to conventional approaches for the common triangle and common edge cases. The Tensor Train SIBI was found to be much more efficient than all other options for the common edge EFIE case whereas the delayed sparse grids and sparse grids where best for the common triangle EFIE and common edge MFIE respectively. An interesting observation about the proposed approach is that using sparse grids the performance is mostly controlled via the interpolation order, while using the tensor decomposition that control parameter is the decomposition tolerance. Specifically, the interpolation order controls the full tensor interpolation accuracy (maximum), while the decomposition tolerance controls how close the interpolation accuracy is to said maximum accuracy at the price of computational cost. Consequently, in future work a higher interpolation order ($p = 3$ or $p = 4$) could be used for the full tensor in the tensor decomposition approach, which could potentially greatly increase the accuracy without significantly affecting the computational cost; in practice building such a library could take months though. Additionally in future work, the h refinement of the utilized domain partitioning for the tensor decomposition approach could be increased to significantly improve accuracy compromising only memory somewhat, but not the computational cost.

The performance of the proposed approach, using its optimal implementations (Delayed) Sparse Grids and Tensor Train was also compared to that of the conventional approach to solve the electromagnetic scattering problem of a sphere. In the former the near field singular interactions were computed by the optimal SIBI implementation and in the latter by a state-of-the-art singularity subtraction. Both approaches showed similar accuracy and memory cost for the radar cross section with regards to the analytical Mie series solution, but the proposed approach improved computation time by about $11 - 20\%$.

In future work different singular integration rules can be used to construct the *universal* libraries. It was found that the smoothness of the integral functions depended on the accuracy of the integration rule. Thus, a rule with up to machine precision accuracy could yield significant improvement to the performance of SIBI. The principle of SIBI is general and could be applied to different kernels, therefore future research in this topic could also involve cases such as lossy media, or used with second order basis function or used with curvilinear triangles. In addition, research into interpolation methods that scale slowly with dimension could allow SIBI to be extended to higher dimensionality cases such as the common vertex case. An efficient automatic implementation of iterative *hp* refinement of the parametrization domain could improve the performance of SIBI, especially in the case of Tensor Train, where the decompositions were fairly costly and tedius without the use of an in-house Tensor Train decomposition code.

Appendices

APPENDIX A

VECTOR TO SCALAR INTEGRALS

In the case of EFIE the BEM element matrix integrals in (2.1), can easily be expressed as integrals where the basis functions are scalar rather than vectorial (see Chapter 2). Substituting the basis functions $\boldsymbol{\alpha}^t, \boldsymbol{\alpha}$ and using $\zeta_0 + \zeta_1 + \zeta_2 = 1$ we get:

$$Z_{mn}^{(e)\ \mathcal{L}} = \frac{1}{4A^2} \sum_{\substack{p=0 \\ p \neq m}}^{2} \sum_{\substack{q=0 \\ q \neq n}}^{2} c_{pq} \mathcal{I}_{pq} - \frac{d_{mn}}{k^2} \sum_{p=0}^{2} \sum_{q=0}^{2} \mathcal{I}_{pq}$$

where $c_{pq} = s(m,p)\, s(n,q)\, \boldsymbol{\ell}_{f(m,p)} \cdot \boldsymbol{\ell}_{f(n,q)}$ and

$$s(m,p) = \begin{cases} (-1)^{p+1}, & m = 0 \\ (-1)^{p/2+1}, & m = 1 \\ (-1)^{p}, & m = 2 \end{cases} \tag{A.1}$$

$$f(m,p) = \begin{cases} 3|m-p| - 1, & m + p = 1 \\ |m-p| - 1, & m + p \neq 1 \end{cases} \tag{A.2}$$

$$\begin{aligned} d_{mn} = {} & \frac{\hat{\boldsymbol{h}}_{m+1}}{h_{m+1}} \cdot \boldsymbol{\ell}_{m-1} \frac{\hat{\boldsymbol{h}}_{n+1}}{h_{n+1}} \cdot \boldsymbol{\ell}_{n-1} - \frac{\hat{\boldsymbol{h}}_{m+1}}{h_{m+1}} \cdot \boldsymbol{\ell}_{m-1} \frac{\hat{\boldsymbol{h}}_{n-1}}{h_{n-1}} \cdot \boldsymbol{\ell}_{n+1} - \\ & \frac{\hat{\boldsymbol{h}}_{m-1}}{h_{m-1}} \cdot \boldsymbol{\ell}_{m+1} \frac{\hat{\boldsymbol{h}}_{n+1}}{h_{n+1}} \cdot \boldsymbol{\ell}_{n-1} + \frac{\hat{\boldsymbol{h}}_{m-1}}{h_{m-1}} \cdot \boldsymbol{\ell}_{m+1} \frac{\hat{\boldsymbol{h}}_{n-1}}{h_{n-1}} \cdot \boldsymbol{\ell}_{n+1} \end{aligned} \tag{A.3}$$

and

$$\mathcal{I}_{pq} = \int_T \zeta_p(\mathbf{r}) \int_{T'} \zeta_q(\mathbf{r}') \frac{e^{-jk|\mathbf{r}-\mathbf{r}'|}}{4\pi|\mathbf{r}-\mathbf{r}'|} d\mathbf{r}'^2 d\mathbf{r}^2 \tag{A.4}$$

where $\boldsymbol{\ell}_m, \boldsymbol{\ell}_n$ are the edge vectors, $\boldsymbol{h}_m, \boldsymbol{h}_n$ are the height vectors and $\zeta_p(\mathbf{r}), \zeta_q(\mathbf{r}')$ are the barycentric coordinates of the triangles T, T' respectively.

For example for $m = 0, n = 1 \Rightarrow p = \{\cancel{0}, 1, 2\}, q = \{0, \cancel{1}, 2\}$

The first term after substituting the basis function $\boldsymbol{\alpha}_0, \boldsymbol{\alpha}_1$ is:

$$\int_T \frac{1}{2A}(\zeta_1\boldsymbol{\ell}_2 - \zeta_2\boldsymbol{\ell}_1) \cdot \int_{T'} \frac{1}{2A}(\zeta_2\boldsymbol{\ell}_0 - \zeta_0\boldsymbol{\ell}_2) G(\mathbf{r}, \mathbf{r}') d\mathbf{r}'^2 d\mathbf{r}^2 =$$

$$\int_T \int_{T'} \frac{1}{4A^2} \left(\zeta_1\zeta_2\boldsymbol{\ell}_2 \cdot \boldsymbol{\ell}_0 - \zeta_1\zeta_0\boldsymbol{\ell}_2 \cdot \boldsymbol{\ell}_2 - \zeta_2\zeta_2\boldsymbol{\ell}_1 \cdot \boldsymbol{\ell}_0 + \zeta_2\zeta_0\boldsymbol{\ell}_1 \cdot \boldsymbol{\ell}_2\right) G(\mathbf{r}, \mathbf{r}') d\mathbf{r}'^2 d\mathbf{r}^2 =$$

$$\frac{1}{4A^2}\Big(\overbrace{\boldsymbol{\ell}_2 \cdot \boldsymbol{\ell}_0}^{c_{12}} \mathcal{I}_{12} \overbrace{-\boldsymbol{\ell}_2 \cdot \boldsymbol{\ell}_2}^{c_{10}} \mathcal{I}_{10} \overbrace{-\boldsymbol{\ell}_1 \cdot \boldsymbol{\ell}_0}^{c_{22}} \mathcal{I}_{22} + \overbrace{\boldsymbol{\ell}_1 \cdot \boldsymbol{\ell}_2}^{c_{20}} \mathcal{I}_{20}\Big)$$

where

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

The second term after substituting the basis function $\boldsymbol{\alpha}_0, \boldsymbol{\alpha}_1$ is:

$$\int_T \int_{T'} \frac{1}{4A^2} \left(\frac{\hat{\boldsymbol{h}}_1}{h_1} \cdot \boldsymbol{\ell}_2 \frac{\hat{\boldsymbol{h}}_2}{h_2} \cdot \boldsymbol{\ell}_0 - \frac{\hat{\boldsymbol{h}}_1}{h_1} \cdot \boldsymbol{\ell}_2 \frac{\hat{\boldsymbol{h}}_0}{h_0} \cdot \boldsymbol{\ell}_2 - \frac{\hat{\boldsymbol{h}}_2}{h_2} \cdot \boldsymbol{\ell}_1 \frac{\hat{\boldsymbol{h}}_2}{h_2} \cdot \boldsymbol{\ell}_0 + \frac{\hat{\boldsymbol{h}}_2}{h_2} \cdot \boldsymbol{\ell}_1 \frac{\hat{\boldsymbol{h}}_0}{h_0} \cdot \boldsymbol{\ell}_2\right) G(\mathbf{r}, \mathbf{r}') d\mathbf{r}'^2 d\mathbf{r}^2$$

$$= d_{mn} \int_T \int_{T'} G(\mathbf{r}, \mathbf{r}') d\mathbf{r}'^2 d\mathbf{r}^2$$

Then after substituting $\zeta_0 + \zeta_1 + \zeta_2 = 1$ in both source and test integrals:

$$d_{mn} \int_T (\zeta_0 + \zeta_1 + \zeta_2) \int_{T'} (\zeta_0 + \zeta_1 + \zeta_2) G(\mathbf{r}, \mathbf{r}') d\mathbf{r}'^2 d\mathbf{r}^2 = d_{mn} \sum_{p=0}^{2} \sum_{q=0}^{2} \mathcal{I}_{pq}$$

BIBLIOGRAPHY

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

[2] T. Rylander, P. Ingelström, and A. Bondeson, *Computational Electromagnetics.* Springer, 2013.

[3] W. C. Gibson, *The Method of Moments in Electromagnetics.* CRC Press, USA: Boca Raton, 2015.

[4] J.-M. Jin, *Theory and Computation of Electromagnetic Fields.* John Wiley & Sons, Inc., Hoboken New Jersey, USA, 2010.

[5] D. M. Pozar, *Microwave Engineering.* John Wiley & Sons, Inc., 2011.

[6] R. E. Collins, *Foundations for Microwave Engineering.* MacGraw-Hill, 1992.

[7] F. T. Ulaby and U. Ravaioli, *Fundamentals of Applied Electromagnetics.* Pearson, 2022.

[8] E.-P. Li, *Electrical Modeling and Design for 3D System Integration: 3D Integrated Circuits and Packaging, Signal Integrity, Power Integrity and EMC.* John Wiley & Sons, Inc., 2012.

[9] C. A. Balanis, *Antenna Theory: Analysis and Design.* Wiley Blackwell, 2016.

[10] P. de Maag, R. Gonzalo, Y. C. Vardaxoglou, and J.-M. Baracco, “Electromagnetic bandgap antennas and components for microwave and (sub) millimeter wave applications,” *Transactions on Antennas and Propagation Magazine, IEEE*, vol. 51, no. 10, pp. 2667–2677, 2003.

[11] M. R. Castellanos, Y. Liu, D. J. Love, B. Peleato, J.-M. Jin, and B. M. Hochwald, “Signal-level models of pointwise electromagnetic exposure for millimeter wave communication,” *Transactions on Antennas and Propagation Magazine, IEEE*, vol. 68, no. 5, pp. 3963–3977, 2020.

[12] W. Withayachumnankul and D. Abbott, “Metamaterials in the terahertz regime,” *Photonics Journal, IEEE*, vol. 1, no. 2, pp. 99–118, 2009.

[13] T. Maka, D. N. Chigrin, S. G. Romanov, and C. M. Torres, “Three dimensional photonic crystals in the visible regime,” *Progress In Electromagnetics Research*, vol. 41, pp. 307–335, 2003.

[14] J.-M. Jin, *The Finite Element Method in Electromagnetics.* John Wiley & Sons, Inc., 2015.

[15] S. D. Gedney, *Introduction to the Finite-Difference Time-Domain (FDTD) Method for Electromagnetics.* Morgan & Claypool Publishers, 2011.

[16] R. F. Harrington, *Field Computation by Moment Methods.* The IEEE Press Series on Electromagnetic Wave Theory, 1993.

[17] T. Strouboulis, K. Copps, and I. Babuška, "The generalized finite element method," *Computer Methods In Applied Mechanics and Engineering*, vol. 190, no. 32, pp. 4081–4193, 2001.

[18] D. J. Benson, Y. Bazilevs, E. D. Luycker, M.-C. Hsu, T. R. Hughes, and T. Belytschko, "A generalized finite element formulation for arbitrary basis functions: from isogeometric analysis to XFEM," *International Journal for Numerical Methods in Engineering*, vol. 83, no. 6, pp. 765–785, 2010.

[19] M. Clemens and T. Weiland, "Discrete electromagnetism with the finite integration technique," *Progress in Electromagnetics Research*, vol. 32, pp. 65–87, 2001.

[20] H. Jahandari and C. G. Farquharson, "A finite-volume solution to the geophysical electromagnetic forward problem using unstructured grids," *Geophysics*, vol. 79, no. 6, pp. 287–302, 2014.

[21] K. F. Warnick, *Numerical Analysis for Electromagnetics Integral Equation.* Artech House, 2008.

[22] A. J. Poggio and E. K. Miller, *Computer Techniques for Electromagnetics.* Pergamon, 1973, ch. 4, Integral Equation Solutions for Three-Dimensional Scattering Problems.

[23] Y. Chang and R. F. Harrington, "A surface formulation for characteristic modes of material bodies," *Transactions on Antennas and Propagation Magazine, IEEE*, vol. 25, no. 6, pp. 789–795, 1977.

[24] T.-K. Wu and L. L. Tsai, "Scattering from arbitrarily-shaped lossy dielectric bodies of revolution," *Radio Science*, vol. 12, no. 5, pp. 709–718, 1977.

[25] C. Cecka and E. Darve, "Fourier based fast multipole method for the Helmholtz equation," *SIAM Journal on Scientific Computing*, vol. 35, pp. A79–A103, 2013.

[26] M. A. Moharram and A. A. Kishk, "Efficient electromagnetic scattering computation using the random auxiliary sources method for multiple composite 3D arbitrary objects," *Transactions on Antennas and Propagation Magazine, IEEE*, vol. 63, no. 8, pp. 3621–3633, 2015.

[27] C. S. Merola and M. N. Vouvakis, "UHF planar ultrawideband modular antenna (puma) arrays," *Transactions on Antennas and Propagation Magazine, IEEE*, pp. 1803–1804, 2017.

[28] W. Smigaj, T. Betcke, S. Arridge, J. Phillips, and M. Schweiger, "Solving boundary integral problems with BEM++," *ACM Transactions on Mathematics Software*, vol. 41, pp. 1–40, 2015.

[29] L. Zhang, T. Cui, and H. Liu, "A set of symmetric quadrature rules on triangles and tetrahedra," *Journal of Computational Mathematics*, vol. 27, no. 1, pp. 89–96, 2009.

[30] D. A. Dunavant, "High degree efficient symmetrical Gaussian quadrature rules for the triangle," *International Journal for Numerical Methods in Engineering*, vol. 21, pp. 1129–1148, 1985.

[31] W. B. Lu, T. J. Cui, and H. Zhao, "Acceleration of fast multipole method for large-scale periodic structures with finite sizes using sub-entire-domain basis functions," *Transactions on Antennas and Propagation Magazine, IEEE*, vol. 55, no. 2, pp. 414–421, 2007.

[32] X. C. Wei, E. P. Li, and Y. J. Zhang, "Efficient solution to the large scattering and radiation problem using the improved finite-element fast multipole method," *Transactions on Magnetics Magazine, IEEE*, vol. 41, no. 5, pp. 1684–1687, 2005.

[33] O. Ergul, T. Malas, C. Yavuz, A. Unal, and L. Gurel, "Computational analysis of complicated metamaterial structures using MLFMA and nested preconditioners," in *The Second European Conference on Antennas and Propagation, EuCAP*. IEEE, 2007, pp. 1–6.

[34] S. He, Z. Nie, and J. Hu, "Electromagnetic solution for dielectric objects with multilevel fast multipole algorithm and singular value decomposition," in *Antennas and Propagation Society International Symposium, IEEE.* IEEE, 2009, pp. 1–4.

[35] D. T. Schobert and T. F. Eibert, "Fast solution of finite element/boundary integral problems employing hierarchical Green's function interpolation combined with multilevel fast multipole method," in *International Conference on Electromagnetics in Advanced Applications, IEEE.* IEEE, 2012, pp. 694–697.

[36] J. Rivero, F. Vipiana, D. R. Wilton, and W. A. Johnson, "Evaluation of 4-D reaction integrals via double application of the divergence theorem," *Transactions on Antennas and Propagation Magazine, IEEE*, vol. 67, no. 2, pp. 1131–1142, 2019.

[37] S. Järvenpää, M. Taskinen, and P. Ylä-Oijala, "Singularity subtraction technique for high-order polynomial vector basis functions on planar triangles," *Transactions on Antennas and Propagation Magazine, IEEE*, vol. 54, no. 1, pp. 42–49, 2006.

[38] A. G. Polimeridis, F. Vipiana, J. R. Mosig, and D. R. Wilton, "A general framework for high precision computation of singular integrals in Galerkin SIE formulations," in *Antennas and Propagation Society International Symposium (APSURSI), IEEE*. IEEE, 2013, pp. 454–455.

[39] M. M. Botha, "Numerical integration scheme for the near-singular Green function gradient on general triangles," *Transactions on Antennas and Propagation Magazine, IEEE*, vol. 63, pp. 4435–4445, 2015.

[40] N. V. Nair, A. J. Pray, J. Villa-GIron, B. Shanker, and D. R. Wilton, "A singularity cancellation technique on arbitrary higher order patch descriptions," in *Antennas and Propagation Magazine Conference, IEEE*. IEEE, 2012.

[41] M. A. Khayat and D. R. Wilton, "Numerical evaluation of singular and near-singular potential integrals," *Transactions on Antennas and Propagation Magazine, IEEE*, vol. 53, pp. 3180–3190, 2005.

[42] F. Vipiana and D. R. Wilton, "Numerical evaluation via singularity cancellation schemes of near-singular integrals involving the gradient of Helmholtz-type potentials," *Transactions on Antennas and Propagation Magazine, IEEE*, vol. 61, pp. 1255–1265, 2013.

[43] D. R. Wilton, F. Vipiana, and W. A. Johnson, "Evaluation of 4-D reaction integrals in the method of moments: Coplanar element case," *Transactions on Antennas and Propagation Magazine, IEEE*, vol. 65, no. 5, pp. 2479–2493, 2017.

[44] A. G. Polimeridis and T. V. Yioultsis, "On the direct evaluation of weakly singular integrals in Galerkin potential integral equation formulations," *Transactions on Antennas and Propagation Magazine, IEEE*, vol. 56, pp. 3011–3019, 2008.

[45] A. G. Polimeridis, J. Tamayo, J. M. Rius, and J. R. Mosig, "Evaluation of weakly singular integrals via generalized cartesian product rules based on the double exponential formula," *Transactions on Antennas and Propagation Magazine, IEEE*, vol. 58, pp. 1980–1988, 2010.

[46] A. G. Polimeridis, J. M. Tamayo, J. M. Rius, and J. R. Mosig, "Fast and accurate computation of hyper-singular integrals in Galerkin surface integral equation formulations via the direct evaluation method," *Transactions on Antennas and Propagation Magazine, IEEE*, vol. 59, pp. 2329–2340, 2011.

[47] P. C. Hansen, "Numerical tools for analysis and solution of Fredholm integral equations of the first kind," *Inverse Problems*, vol. 8, pp. 849–872, 1992.

[48] K. Atkinson and W. Han, *Theoretical Numerical Analysis: A Functional Analysis Framework*. Springer, 2009.

[49] R. Piessens, "Numerical evaluation of Cauchy principal value integrals," *Numerical Mathematics, BIT*, vol. 10, no. 4, pp. 467–480, 1970.

[50] D. S. Weile, G. Pisharody, N.-W. Chen, B. Shanker, and E. Michielssen, "A novel scheme for the solution of the time-domain integral equations of electromagnetics," *Transactions on Antennas and Propagation Magazine, IEEE*, vol. 52, no. 1, pp. 283–295, 2004.

[51] S. Rao, A. Glisson, D. Wilton, and B. Vidula, "A simple numerical solution procedure for statics problems involving arbitrary-shaped surfaces," *Transactions on Antennas and Propagation Magazine, IEEE*, vol. 27, no. 5, pp. 604–608, 1979.

[52] A. Menshov and V. I. Okhmatovski, "Surface-volume-surface electric field integral equation for magneto-quasi-static analysis of complex 3-D interconnects," *Transactions on Antennas and Propagation Magazine, IEEE*, vol. 62, no. 11, pp. 2563–2573, 2014.

[53] P. Ylä-Oijala, J. Markkanen, S. Järvenpää, and S. P. Kiminki, "Surface and volume integral equation methods for time-harmonic solutions of Maxwell's equations," *Progress In Electromagnetics Research*, vol. 149, pp. 15–44, 2014.

[54] P. Ylä-Oijala and M. Taskinen, "Well conditioned Müller formulation for electromagnetic scattering by dielectric objects," *Transactions on Antennas and Propagation Magazine, IEEE*, vol. 53, no. 10, pp. 3316–3323, 2005.

[55] T. E. Roth and W. C. Chew, "A current and charge integral equation for dielectric regions in the time domain," *International Symposium on Antennas and Propagation and North American Radio Science Meeting*, pp. 1011–1012, 2020.

[56] M. Taskinen and P. Ylä-Oijala, "Current and charge integral equation formulation," *Transactions on Antennas and Propagation Magazine, IEEE*, vol. 54, no. 1, pp. 58–67, 2006.

[57] D. Conte and B. Paternoster, "Multistep collocation methods for Volterra integral equations," *Applied Numerical Mathematics*, vol. 59, no. 8, pp. 1721–1736, 2009.

[58] J. Bremer, "On the Nystr om discretization of integral equations on planar curves with corners," *Applied and Computational Harmonic Analysis*, vol. 32, no. 1, pp. 45–64, 2012.

[59] V. I. Okhmatovski, J. D. morsey, and A. C. Cangellaris, "Enhancement of the numerical stability of the adaptive integral method at low frequencies through a loop-charge formulation of the method-of-moments approximation," *Transactions on Antennas and Propagation Magazine, IEEE*, vol. 52, no. 3, pp. 962–970, 2004.

[60] A. R. Hall, *Generalized Method of Moments.* OUP Oxford, 2004.

[61] D. L. Dault, N. V. Nair, J. Li, and B. Shanker, "The generalized method of moments for electromagnetic boundary integral equations," *Transactions on Antennas and Propagation Magazine, IEEE*, vol. 62, no. 6, pp. 3174–3188, 2014.

[62] K. A. Michalski and J. R. Mosig, “Multilayered media Green’s functions in integral equation formulations,” *Transactions on Antennas and Propagation Magazine, IEEE*, vol. 45, no. 3, pp. 58–519, 1997.

[63] R. E. Jorgenson and R. Mittra, “Efficient calculation of the free-space periodic Green’s function,” *Transactions on Antennas and Propagation Magazine, IEEE*, vol. 38, no. 5, pp. 633–642, 1990.

[64] E. G. Williams, “On Green’s functions for a cylindrical cavity,” *The Journal of the Acoustical Society of America*, vol. 102, no. 6, pp. 3300–3307, 1997.

[65] A. Tzoulis and T. F. Eibert, “A hybrid FEBI-MLFMM-UTD method for numerical solutions of electromagnetic problems including arbitrarily shaped and electrically large objects,” *Transactions on Antennas and Propagation Magazine, IEEE*, vol. 53, no. 10, pp. 3358–3366, 2005.

[66] M. Zaslavsky, V. Druskin, S. Davydycheva, L. Knizhnerman, A. Abubakar, and T. Habashy, “Hybrid finite-difference integral equation solver for 3D frequency domain anisotropic electromagnetic problems,” *Geophysics*, vol. 76, no. 2, pp. 123–137, 2011.

[67] R. Gentleman and V. J. Carey, “Unsupervised machine learning,” *Bioconductor Case Studies*, pp. 137–157, 2008.

[68] P. Ylä-Oijala and M. Taskinen, “Calculation of CFIE impedance matrix elements with RWG and $n\times$ RWG functions,” *Transactions on Antennas and Propagation Magazine, IEEE*, vol. 51, pp. 1837–1846, 2003.

[69] L. Zhang, A. Deng, and M. .Wang, “Using an accurate singularity extraction technique to the MFIE for the scattering analysis of small objects,” *Engineering Analysis with Boundary Elements*, vol. 33, pp. 442–446, 2009.

[70] L. Gürel and O. Ergül, “Singularity of the magnetic-field integral equation and its extraction,” *Antennas and Wireless Propagation Letters, IEEE*, vol. 4, pp. 229–232, 2005.

[71] J. Rivero, F. Vipiana, D. R. Wilton, and W. A. Johnson, “Hybrid integration scheme for the evaluation of strongly singular and near-singular integrals in surface integral equations,” *Transactions on Antennas and Propagation Magazine, IEEE*, vol. 67, no. 10, pp. 6532–6540, 2019.

[72] A. G. Polimeridis, F. Vipiana, J. R. Mosig, and D. R. Wilton, “DIRECTFN: Fully numerical algorithms for high precision computation of singular integrals Galerkin SIE methods,” *Transactions on Antennas and Propagation Magazine, IEEE*, vol. 61, no. 6, pp. 3112–3122, 2013.

[73] M. T. H. Reid, J. K. White, and S. Jonhson, “Generalized Taylor-Duffy method for efficient evaluation of Galerkin integrals boundary-element method computations,” *Transactions on Antennas and Propagation Magazine, IEEE*, vol. 63, no. 1, pp. 195–209, 2015.

[74] W. A. Klimke, “Uncertainty modeling using fuzzy arithmetic and sparse grids,” Ph.D. dissertation, Universität Stuttgart, 2006.

[75] T. Gerstner and M. Griebel, “Numerical integration using sparse grids,” *Numerical Algorithms*, vol. 18, pp. 209–232, 1998.

[76] V. Barthelmann, E. Novak, and K. Ritter, “High dimensional polynomial interpolation on sparse grids,” *Advances in Computational Mathematics*, vol. 12, pp. 273–288, 2000.

[77] T. G. Kolda and B. W. Bader, “Tensor decomposition and applications,” *Society for Industrial and Applied Mathematics Review*, vol. 51, pp. 455–500, 2009.

[78] I. Oseledets and E. Tyrtyshnikov, “TT-cross approximation for multidimensional arrays,” *Linear Algebra and its Applications*, vol. 432, pp. 70–80, 2010.

[79] I. G. Kyriakou, C. L. Zekios, and M. N. Vouvakis, “Singular integration in BEM by interpolation: The EFIE case,” in *International Symposium on Antennas and Propagation & USNC/URSI National Radio Science Meeting, IEEE.* IEEE, 2018, pp. 2313–2314.

[80] M. G. Duffy, “Quadrature over a pyramid or cube of integrands with a singularity at a vertex,” *Journal of Numerical Analysis, SIAM*, vol. 19, pp. 1260–1262, 1982.

[81] I. G. Kyriakou and M. N. Vouvakis, “Singular integration by interpolation (SIBI) for integral equations via tensor trains,” in *International Symposium on Antennas and Propagation and North American Radio Science Meeting, IEEE.* IEEE, 2020, pp. 1053–1054.

[82] S. M. Rao, D. R. Wilton, and A. W. Glisson, “Electromagnetic scattering by surfaces of arbitrary shape,” *Transactions on Antennas and Propagation Magazine, IEEE*, vol. 30, pp. 409–418, 1982.

[83] J. Burkardt, “Slow exponential growth for Gauss Patterson sparse grids,” 2014.

[84] J. P. Berrut and L. N. Trefethen, “Barycentric Lagrange interpolation,” *Society for Industrial & Applied Mathematics, Review*, vol. 46, pp. 501–517, 2004.

[85] A. Chichocki, D. Mandic, A.-H. Phan, C. Caiafa, G. Zhou, Q. Zhao, and L. D. Lathauwer, “Tensor decomposition for signal processing applications: From two-way to multiway component analysis,” *Signal Processing Magazine, IEEE*, vol. 32, no. 2, pp. 145–163, 2015.

[86] J. Jeffers and J. Reinders, *Intel Xeon Phi Coprocessor High Performance Programming.* Newnes, 2013.

[87] W. Mitchell and R. Bank, “PLTMG: A software package for solving elliptic partial differential equations, user’s guide 7.0,” *Mathematics of Computation, SIAM*, vol. 64, p. 1343, 1995.