FAST SOLUTIONS OF MULTIPLE MONOSTATIC RADAR CROSS SECTION PROBLEMS USING THE RECOMPRESSED ADAPTIVE CROSS APPROXIMATION ALGORITHM

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND

ELECTRONICS ENGINEERING

AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

SeyedMahdi Kazempourradi

January, 2014

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Levent Gürel (Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Tuğrul Dayar

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Sinan Gezici

Approved for the Graduate School of Engineering and Science:

Prof. Dr. Levent Onural
Director of the Graduate School

ABSTRACT

FAST SOLUTIONS OF MULTIPLE MONOSTATIC RADAR CROSS SECTION PROBLEMS USING THE RECOMPRESSED ADAPTIVE CROSS APPROXIMATION ALGORITHM

SeyedMahdi Kazempourradi
M.S. in Electrical and Electronics Engineering
Supervisor: Prof. Dr. Levent Gürel
January, 2014

We developed a method that incorporates an algebraic compression technique to accelerate the computation of multiple monostatic radar cross sections (RCSs) of arbitrary 3-D geometries. Since most radars rely on the backscattering from a target, computing the monostatic RCS (MRCS) is needed more often than the bistatic RCS. Computation of each MRCS value requires a separate solution, which may be costly depending on the size of the problem. The task becomes considerably harder when the goal is to compute multiple MRCS values with high angular resolution. The proposed technique compresses the excitation matrix using the adaptive cross approximation (ACA) algorithm in the first step. A recompression is applied on the matrices obtained from ACA by utilizing the QR decomposition and computing the singular value decomposition in an efficient manner. The solution of each excitation is accelerated by the multilevel fast multipole algorithm. The numerical results demonstrate the efficiency and accuracy of our proposed method.

Keywords: Adaptive cross approximation (ACA), multilevel fast multipole algorithm (MLFMA), radar cross section (RCS), Recompression.

ÖZET

MONOSTATİK RADAR KESİT ALAN PROBLEMLERİNİN ÇÖZÜMLERİNİN TEKRAR SIKIŞTIRILMIŞ YAKLAŞIK ADAPTİF ÇAPRAZ ALGORİTMASIYLA HIZLANDIRILMASI

SeyedMahdi Kazempourradi
Elektrik ve Elektronik Mühendisliği, Yüksek Lisans
Tez Yöneticisi: Prof. Dr. Levent Gürel
Ocak, 2014

Radar kesit alan (RKA) hesaplaması stratejik öneme sahiptir. RKA hesaplamaları için bir yöntem, sayısal elektromanyetik çözüm tekniklerinin kullanılmasıdır. Ancak, gerçek yaşamda önem taşıyan geometrilerin modellenmesi, karşımıza çok büyük sayısal problemler çıkartmaktadır. Böyle büyük sayısal problemleri, matris denklemine çevirip hızlı çözüm yöntemleriyle çözebiliriz. Alıcı ve vericinin yan yana konumlandırıldıgı, hatta bir çok zaman aynı antenin kullanıldıgı radar tiplerinde, monostatik RKA (MRKA) çok önemli sayılır. Tek bir MRKA hesaplanması için bir sayısal çözüm yapılması gerekir ve bu çözüm pahalı olabilir. Değişik ve yüksek çözünürlüklü küresel açılarda ise MRKA hesaplaması çok daha pahalı olabilir. Bu çalışmada birden fazla MRKA hesaplanmasının hızlandırılması amacıyla bir sıkıştırma algoritması geliştirilmiştir. Bu yöntemde, ilk seviyede yaklaşık adaptif çapraz algoritması (YAÇA) kullanılır. Bir sonraki adımda YAÇA'dan çıkan matrisler alınır ve QR algoritmasını uygulanarak az işlemle tekil değer ayrışımı hesaplanır. Belli bir seviyeden düşük olan tekil değerler atıldığında, daha düşük bir etkin rank elde edilir. Bu tezde sıkıştırma algortimasının geliştirilmesi ve farklı problemlere uygulanması anlatılacaktır.

Anahtar sözcükler: Yaklaşık Adaptif Çapraz Algoritması (YAÇA), Radar Kesit Alanı (RKA), Tekrar Sıkıştırma.

Acknowledgement

I would like to express my sincere gratitude to Prof. Dr. Levent Gürel for his supervision, guidance and the suggestions throughout the development of this thesis. It was an honor for me to work with him and benefit from his deep knowledge and experience in electromagnetics.

I also wish to thank Prof. Dr. Tuğrul Dayar and Assoc. Prof. Dr. Sinan Gezici for evaluating and commenting on my thesis as a jury member.

I also like to thank the members of the Bilkent university computational electromagnetics research center (BiLCEM) for their support, collaboration, friendship, and invaluable suggestions in the preparation of this thesis.

My final gratitude is to my family, who gave me endless support in my study.

This thesis is dedicated to my parents for their love, support and encouragement.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation and Related Works

There are many circumstances where the knowledge of the radar cross section (RCS) is crucially important [1]. This information can be used in many ways, such as designing stealth targets [2] or in determining suitable electronic counter measures against a certain target [3].

In some cases, direct measurement of RCSs of a target may not be possible or practical, leaving computation as the viable option. Consequently, accurate and fast RCS simulations are important for predicting actual RCS values [4]. The monostatic RCS (MRCS) values, where the incidence and observation directions are the same, are used more often than the bistatic RCS because most radars depend on the back-scattering from a target. Each MRCS computation requires a separate solution that can be costly depending on the size of the problem. Fast solvers can be utilized to accelerate the solution of large electromagnetic scattering problems by reducing the solution time and the memory consumption. Nevertheless, computing MRCS values for problems that are excited with various incident plane waves from different angles become time-consuming. The task becomes considerably harder when the goal is to compute multiple MRCS values with high angular resolution.

Several methods are developed and used to speed up the computation of RCS problems with multiple right-hand sides (RHSs). By exploiting an adaptive cubic-spline interpolation [5], the desired MRCS values for a given resolution can be computed. Removing redundancies in the excitation matrix by using QR decomposition [6], singular value decomposition (SVD) [7], interpolative decomposition (ID) [8], and adaptive cross approximation (ACA) [13] leads to fast RCS computations. An approach to the solution of RCS problems over a wide range of incident plane waves is the rational function approximation technique. There are three main approaches for creating approximated rational functions: asymptotic waveform evaluation (AWE) [9, 10], the Padé approximation, and model-based parameter estimation (MBPE) [11, 12]. One advantage of ACA is that it requires only partial knowledge of the matrix elements. It also has a low computational cost [14]. However, for a given threshold, the most efficient low-rank approximation a matrix can be obtained by SVD [15]. In this paper, we propose a compression technique that incorporates SVD on the low-rank matrices obtained from ACA to reach a better low-rank decomposition of the RHS matrix. In other words, the main bottleneck of SVD, i.e., its high computational complexity [13], is solved by applying a recompressed ACA (RACA) algorithm. Due to the algebraic nature of the proposed method, it can be used in combination with any fast solver [16, 17, 18, 19] without any major change in the existing code. In this work, the multilevel fast multipole algorithm (MLFMA) is used to accelerate the solution of each excitation [16].

1.2 Notation

In this thesis, Greek or Roman letters with an italic font are used for scalars. Bold-face, italicized, capital letters with an overbar, such as $\overline{\boldsymbol{A}}$, represent matrices. Vectors are denoted by bold-face, italicized, small letters without a bar, such as $\boldsymbol{a}$. For unit vectors, we use a hat over the vector, such as $\hat{\boldsymbol{n}}$. Bold-face, italicized, capital letters with a tilde, such as $\tilde{\boldsymbol{A}}$, denote approximate matrices. A dot product is used for matrix-matrix and matrix-vector multiplications (MVM).

Chapter 2

Background

2.1 Surface Integral Equations

Surface integral equations (SIEs) are widely used for the solutions of scattering and radiation problems in electromagnetics [20]. Equivalent electric and/or magnetic currents are defined on the surface of a three-dimensional (3-D) complicated structure. The problem can be formulated by applying boundary conditions on the surface of the structure by using the equivalent currents. There are two major testing schemes used in SIEs, namely tangential (T) and normal (N) equations [21]. In tangential equations, boundary conditions are tested directly by sampling the tangential components of fields on the surface. However, in normal equations, fields are tested after a projection on the surface via cross-product operations with the outward normal vector. Applying boundary conditions on the electric field and the magnetic field leads to the electric-field integral equation (EFIE) and the magnetic-field integral equation (MFIE), respectively.

For a perfect electric conductor (PEC) object, any of the T-EFIE, T-MFIE, N-EFIE, and N-MFIE formulations can be used to obtain the induced currents on the surface of the object [22]. The most common formulations in the literature are T-EFIE and N-MFIE, but for closed geometries those formulations suffer from internal resonances. A solution to this issue is to use the combined-field integral

equation (CFIE), which is an appropriate convex combination of T-EFIE and N-MFIE [23].

2.2 T-EFIE Formulation

The total tangential electric field vanishes on a conducting surface, and thus, based on this physical boundary condition, we can declare T-EFIE as

$$\hat{\boldsymbol{t}} \cdot \int_{S'} d\boldsymbol{r'} \overline{\boldsymbol{G}}(\boldsymbol{r}, \boldsymbol{r'}) \cdot \boldsymbol{J}(\boldsymbol{r'}) = \frac{i}{k\eta} \hat{\boldsymbol{t}} \cdot \boldsymbol{E}^{\text{inc}}(\boldsymbol{r}) \quad \boldsymbol{r} \in S', \tag{2.1}$$

where $\boldsymbol{E}^{\text{inc}}(\boldsymbol{r})$ represents the incident electric field and S' is the surface of the geometry. The intrinsic impedance of the medium is $\eta = \sqrt{\mu/\epsilon}$ and k is the wavenumber ($k = 2\pi/\lambda$, where λ is the wavelength). In (2.1), $\hat{\boldsymbol{t}}$ is any tangential unit vector on S' and $\boldsymbol{J}(\boldsymbol{r'})$ is the unknown induced current residing on the surface. The dyadic Green's function is defined as

$$\overline{\boldsymbol{G}}(\boldsymbol{r}, \boldsymbol{r'}) = \left[\overline{\boldsymbol{I}} + \frac{\nabla\nabla}{k^2}\right] g(\boldsymbol{r}, \boldsymbol{r'}), \tag{2.2}$$

where

$$g(\boldsymbol{r}, \boldsymbol{r'}) = \frac{e^{ik|\boldsymbol{r}-\boldsymbol{r'}|}}{4\pi|\boldsymbol{r}-\boldsymbol{r'}|} \tag{2.3}$$

is the scalar Green's function for the 3-D scalar Helmholtz equation. This function represents the response at $\boldsymbol{r}$ due to a point source located at $\boldsymbol{r'}$. If the incident electric field changes, only the RHS of the equation in (2.1) will alter. This point becomes important when we are dealing with problems that are excited with multiple sources.

2.2.1 N-MFIE Formulation

The boundary condition for the tangential magnetic field states that the tangential components of the magnetic field across an interface, along which there exists a surface electric current density $\boldsymbol{J}$, are discontinuous by an amount equal to the electric current density. The N-MFIE formulation can be derived by testing the

magnetic field after a projection on the surface via cross-product operations with the outward normal vector and imposing the physical boundary condition. The expression for N-MFIE can be written as

$$\boldsymbol{J}(\boldsymbol{r}) - \hat{\boldsymbol{n}} \times \int_{S'} d\boldsymbol{r'} \boldsymbol{J}(\boldsymbol{r'}) \times \nabla' g(\boldsymbol{r}, \boldsymbol{r'}) = \hat{\boldsymbol{n}} \times \boldsymbol{H}^{\text{inc}}(\boldsymbol{r}), \tag{2.4}$$

or alternatively:

$$\frac{\Omega_o}{4\pi} \boldsymbol{J}(\boldsymbol{r}) - \hat{\boldsymbol{n}} \times \int_{S'(\text{PV})} d\boldsymbol{r'} \boldsymbol{J}(\boldsymbol{r'}) \times \nabla' g(\boldsymbol{r}, \boldsymbol{r'}) = \hat{\boldsymbol{n}} \times \boldsymbol{H}^{\text{inc}}(\boldsymbol{r}), \tag{2.5}$$

where $\hat{\boldsymbol{n}}$ is any outward unit normal on S' and $\boldsymbol{H}^{\text{inc}}(\boldsymbol{r})$ is the incident magnetic field. In (2.5), Ω_o is the solid angle and the integral on the left-hand side is the principal value. Unlike T-EFIE, which is valid for open and closed geometries, N-MFIE is acceptable only for closed surfaces.

2.2.2 T-N-CFIE Formulation

Unlike T-EFIE, N-MFIE is a second-kind integral equation that leads to well-conditioned matrices after discretization. However, because of the singularity of the N-MFIE kernel and the identity term associated with $\boldsymbol{J}$ term in (2.4), the accuracy of N-MFIE is lower than T-EFIE. T-N-CFIE is a more-accurate integral-equation formulation than N-MFIE [24]. It can be obtained by an appropriate linear combination of T-EFIE and N-MFIE formulations, i.e.,

$$\text{T} - \text{N} - \text{CFIE} = \alpha \text{T} - \text{EFIE} + (1 - \alpha) \text{N} - \text{MFIE}, \tag{2.6}$$

where α is a parameter between 0 and 1. It is shown that $0.2 \leq \alpha \leq 0.3$ yields minimum iteration counts [25]. Along with the noted advantages, one of the most important features of T-N-CFIE is that it is free of internal resonances. T-N-CFIE contains N-MFIE and hence cannot be applied to open geometries.

2.3 Discretization of Surface Integral Equations

Despite the fact that electromagnetics problems can be formulated rigorously with Maxwell's equations, SIEs can only be solved analytically for a few canonical

objects, such as spheres. To numerically solve problems involving complicated structures, the geometry and the SIEs must be discretized simultaneously. In this way, equivalent surface currents are expanded in a series of basis functions. A dense $N \times N$ matrix equation can be obtained using the method of moments (MOM) and testing the boundary conditions for the electric and the magnetic fields on the surface of the geometry, N being the number of unknown coefficients of the basis functions. Each element of the matrix obtained by discretizing the SIEs presents an electromagnetics interaction between a pair of discretized current elements, i.e., basis and testing functions.

2.3.1 Method of Moments

We can convert the SIEs described in Section 2.1 to a matrix equation by utilizing MOM. The equivalent surface currents are expanded as a summation of basis functions. The coefficients of these functions are calculated by solving the matrix equation obtained by MOM. The linear operator $\mathcal{L}$ is applied on the equivalent surface current and the integral equation can be shown as

$$\mathcal{L}\{\boldsymbol{J}(\boldsymbol{r})\} = \mathbf{g}(\boldsymbol{r}), \tag{2.7}$$

where $\mathbf{g}$ is one of the RHS vectors of T-EFIE and/or N-MFIE. Considering $\boldsymbol{J}$ is unknown, expanding $\boldsymbol{J}$ in a series of known basis functions and unknown coefficients, we obtain,

$$\boldsymbol{J}(\boldsymbol{r}) \approx \sum_{n=1}^{N} a_n \boldsymbol{b}_n(\boldsymbol{r}). \tag{2.8}$$

Projecting (2.8) onto the N-dimensional space span$\{\boldsymbol{t}_1, \boldsymbol{t}_2, \cdots, \boldsymbol{t}_N\}$ yields

$$\langle \boldsymbol{t}_m, \mathcal{L}\{\boldsymbol{J}(\boldsymbol{r})\}\rangle = \langle \boldsymbol{t}_m, \mathbf{g}(\boldsymbol{r})\rangle, \quad m = 1, 2, \cdots, N, \tag{2.9}$$

where

$$\langle \boldsymbol{f}, \mathbf{g}\rangle = \int d\boldsymbol{r}\, \boldsymbol{f}(\boldsymbol{r}) \cdot \mathbf{g}(\boldsymbol{r}) \tag{2.10}$$

denotes the inner product of the two vector functions, $\boldsymbol{f}$ and $\mathbf{g}$. If the testing in (2.10) has been done using the same basis functions as in (2.8) (Galerkin scheme), (2.9) will represent a matrix equation.

In other words, (2.9) can be written as

$$\int d\boldsymbol{r}\boldsymbol{t}_m(\boldsymbol{r}) \cdot \sum_{n=1}^{N} a_n \mathcal{L}\{\boldsymbol{b}_n(\boldsymbol{r})\} = \int d\boldsymbol{r}\boldsymbol{t}_m(\boldsymbol{r}) \cdot \boldsymbol{g}(\boldsymbol{r}). \tag{2.11}$$

Changing the order of summation and integration, the equation becomes

$$\sum_{n=1}^{N} a_n \int d\boldsymbol{r}\boldsymbol{t}_m(\boldsymbol{r}) \cdot \mathcal{L}\{\boldsymbol{b}_n(\boldsymbol{r})\} = \int d\boldsymbol{r}\boldsymbol{t}_m(\boldsymbol{r}) \cdot \boldsymbol{g}(\boldsymbol{r}), \tag{2.12}$$

which yields an $N \times N$ matrix equation. Hence, the coefficient vector $\boldsymbol{a}$ becomes the solution of

$$\overline{\boldsymbol{Z}}_{(N\times N)} \cdot \boldsymbol{a}_{(N\times 1)} = \boldsymbol{v}_{(N\times 1)}, \tag{2.13}$$

where

$$\overline{\boldsymbol{Z}}_{mn} = \langle \boldsymbol{t}_m, \mathcal{L}\{\boldsymbol{b}_n\}\rangle, \quad m, n = 1, 2, \cdots, N, \tag{2.14}$$

and

$$\boldsymbol{v}_m = \langle \boldsymbol{t}_m, \mathbf{g}\rangle. \tag{2.15}$$

The interpretation of each matrix element, i.e., $\overline{\boldsymbol{Z}}_{mn}$, is the electromagnetic interaction between the mth testing and the nth basis function.

2.3.2 Discretization of T-EFIE

By substituting $\mathcal{L}$ with the expression that we derived for T-EFIE, the matrix elements can be obtained as

$$\overline{\boldsymbol{Z}}_{mn}^{\text{T-EFIE}} = \int_{S_m} d\boldsymbol{r}\boldsymbol{t}_m(\boldsymbol{r}) \cdot \int_{S_n} d\boldsymbol{r'}\overline{\boldsymbol{G}}(\boldsymbol{r}, \boldsymbol{r'}) \cdot \boldsymbol{b}_n(\boldsymbol{r'}), \tag{2.16}$$

where $\boldsymbol{t}_m$ is a testing function and $\boldsymbol{b}_n$ is a basis function. The expression in (2.16) can be expanded as [26]

$$\begin{aligned}\overline{\boldsymbol{Z}}_{mn}^{\text{T-EFIE}} &= \int_{S_m} d\boldsymbol{r}\boldsymbol{t}_m(\boldsymbol{r}) \cdot \int_{S_n} d\boldsymbol{r'}\boldsymbol{b}_n(\boldsymbol{r'}) g(\boldsymbol{r}, \boldsymbol{r'}) \\ &- \frac{i}{k^2} \int_{S_m} d\boldsymbol{r}\boldsymbol{t}_m(\boldsymbol{r}) \cdot \int_{S_n} d\boldsymbol{r'}\boldsymbol{b}_n(\boldsymbol{r'}) \cdot [\nabla\nabla' g(\boldsymbol{r}, \boldsymbol{r'})].\end{aligned} \tag{2.17}$$

2.3.3 Discretization of N-MFIE

Using MOM for this discretization, the matrix elements can be achieved via

$$\overline{\boldsymbol{Z}}_{mn}^{\text{N-MFIE}} = \int_{S_m} d\boldsymbol{r}\boldsymbol{t}_m(\boldsymbol{r}) \cdot \boldsymbol{b}_n(\boldsymbol{r}) - \int_{S_m} d\boldsymbol{r}\boldsymbol{t}_m(\boldsymbol{r}) \cdot \hat{\boldsymbol{n}} \times \int_{S_n} d\boldsymbol{r'}\boldsymbol{b}_n(\boldsymbol{r'}) \times \nabla' g(\boldsymbol{r}, \boldsymbol{r'}). \tag{2.18}$$

After performing the singularity extraction technique for the outer integral, (2.18) becomes [27]

$$\overline{\boldsymbol{Z}}_{mn}^{\text{N-MFIE}} = \frac{-1}{2} \int_{S_m} d\boldsymbol{r}\boldsymbol{t}_m(\boldsymbol{r}) \cdot \boldsymbol{b}_n(\boldsymbol{r}) - \int_{S_m,\text{PV}} d\boldsymbol{r}\boldsymbol{t}_m(\boldsymbol{r}) \cdot \hat{\boldsymbol{n}} \times \int_{S_n} d\boldsymbol{r'}\boldsymbol{b}_n(\boldsymbol{r'}) \times \nabla' g(\boldsymbol{r}, \boldsymbol{r'}), \tag{2.19}$$

where PV stands for the principal value of the integral.

2.3.4 Discretization of T-N-CFIE

T-N-CFIE is the linear combination of T-EFIE and N-MFIE. Based on this fact, both formulations must be used to form T-N-CFIE. The elements of T-N-CFIE can be derived as

$$\overline{\boldsymbol{Z}}_{mn}^{\text{T-N-CFIE}} = (\alpha)\overline{\boldsymbol{Z}}_{mn}^{\text{T-EFIE}} + (1-\alpha)\overline{\boldsymbol{Z}}_{mn}^{\text{N-MFIE}}. \tag{2.20}$$

2.3.5 Computation of the RHS Vectors

Each element of the RHS vector for T-EFIE can be calculated by testing the incident electric field as

$$\boldsymbol{v}_m^{\text{T-EFIE}} = \frac{-i}{k\eta} \int_{S_m} d\boldsymbol{r}\boldsymbol{t}_m(\boldsymbol{r}) \cdot \boldsymbol{E}^{\text{inc}}(\boldsymbol{r}). \tag{2.21}$$

Similarly, the RHS vector for N-MFIE can be obtained using

$$\boldsymbol{v}_m^{\text{N-MFIE}} = -\int_{S_m} d\boldsymbol{r}\boldsymbol{t}_m(\boldsymbol{r}) \cdot \hat{\boldsymbol{n}} \times \boldsymbol{H}^{\text{inc}}(\boldsymbol{r}). \tag{2.22}$$

Finally, the linear combination of (2.21) and (2.22) yields the RHS vector for T-N-CFIE, i.e.,

$$\boldsymbol{v}_m^{\text{T-N-CFIE}} = \alpha(\boldsymbol{v}_m^{\text{T-EFIE}}) + (1-\alpha)(\boldsymbol{v}_m^{\text{N-MFIE}}). \tag{2.23}$$

In the multiple excitation case, the RHS vector will become a matrix and each column of the matrix can be calculated using (2.23).

2.3.6 RWG Functions

The expansion of the unknown surface current density can be performed using the oriented Rao-Wilton-Glisson (RWG) [26] functions on small planar triangles. It is important to mention that although triangles are planar, two triangles are not necessarily co-planar. To implement a Galerkin approach for discretization, we use RWG functions for basis and testing functions. Surfaces of geometries are meshed with planar triangles in accordance with the RWG functions. Each basis function is associated with an edge, hence the number of unknowns, N, is equal to the total number of edges in the triangulation. For high accuracy, triangle sizes must be small, but this leads to problems with large numbers of unknowns. As a rule of thumb, we choose the average size of the mesh to be about one-tenth of the wavelength ($\lambda/10$).

The RWG set of basis and testing functions exhibits constant normal and linear tangential variations along the edge. The normal (perpendicular) component of the RWG function on the edge is maximum and equal to unity. The normal component of the RWG basis function is continuous on the common edge. These properties will lead to a piecewise constant expansion of the surface current density [26].

The triangular basis functions are defined as

$$\boldsymbol{b}_n(\boldsymbol{r}) = \begin{cases} \dfrac{l_n}{2A_n^+}(\boldsymbol{r} - \boldsymbol{r}_n^+), & \boldsymbol{r} \in S_n^+ \\ \dfrac{l_n}{2A_n^-}(\boldsymbol{r}_n^- - \boldsymbol{r}), & \boldsymbol{r} \in S_n^- \\ 0, & \text{otherwise.} \end{cases} \tag{2.24}$$

Equation (2.24) can be rewritten as

$$\boldsymbol{b}_{ik}(\boldsymbol{r}) = \pm\frac{l_{ik}}{2A_i}(\boldsymbol{r} - \boldsymbol{r}_{ik})\delta_i(\boldsymbol{r}). \tag{2.25}$$

In the equation above, $\delta_i(\boldsymbol{r})$ is defined as

$$\delta_i(\boldsymbol{r}) = \begin{cases} 1, & \boldsymbol{r} \in A_i \\ 0, & \text{otherwise.} \end{cases} \tag{2.26}$$

An important property of RWG functions is that their divergence is finite everywhere:

$$\nabla \cdot \boldsymbol{b}_n(\boldsymbol{r}) = \begin{cases} \dfrac{l_n}{A_n^+}, & \boldsymbol{r} \in S_n^+ \\ -\dfrac{l_n}{A_n^-}, & \boldsymbol{r} \in S_n^- \\ 0, & \text{otherwise.} \end{cases} \tag{2.27}$$

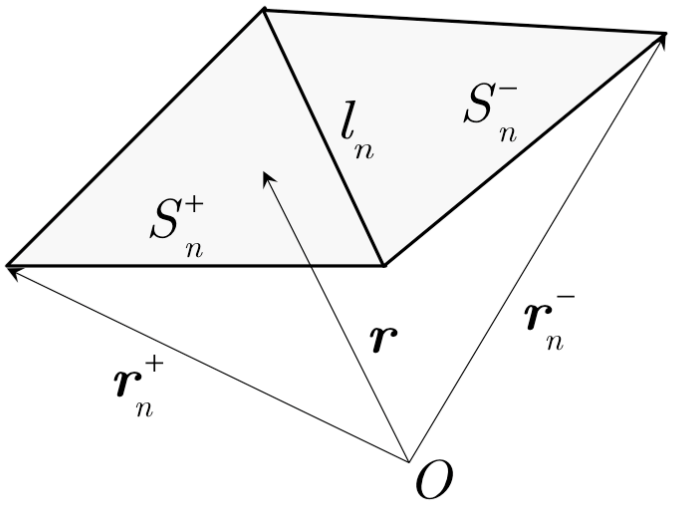

Figure 2.1: The RWG function defined on triangular domains.

2.4 Iterative Solvers

In this section, we give a brief introduction to iterative solvers. Direct solution of a dense $N \times N$ matrix equations using conventional methods, such as Gaussian

elimination, requires $\mathcal{O}(N^3)$ operations and is very costly [15]. The most important reason for using iterative solutions is to reduce the computational complexity of the matrix-equation solution. In the ideal case, we would like to have a complexity of $\mathcal{O}(N)$ or $\mathcal{O}(N \log N)$ both in memory and execution time.

The iterative solver, requires at least one MVM at each iteration. Hence, reducing the complexity of the MVM will lead to a low-complexity solution. The iterations continue until the approximate solution, which is obtained in each iteration, becomes accurate enough to satisfy the practical criteria.

One of the methods that can reduce the complexity of MVM is the multilevel fast multipole algorithm (MLFMA) [16]. This algorithm has both the memory and the CPU time complexities of $\mathcal{O}(N \log N)$.

2.5 Multilevel Fast Multipole Algorithm

Discretization of an integral-equation formulation with MOM leads to a dense $N \times N$ matrix equation. The matrix equation can be solved using direct or iterative solvers. As stated in the previous section, ordinary direct solution methods are too expensive due to their high computational complexity of $\mathcal{O}(N^3)$. On the other hand, typical iterative solvers have the complexity of $\mathcal{O}(N^2)$ for memory requirements and execution time, because they require MVM at each iteration. Many methods have been proposed to accelerate the solutions of matrix equations. Most of these so-called "fast" methods take advantage of redundancies in the MOM impedance matrix.

One of the most powerful methods is the MLFMA. Due to its reduced computational complexity of $\mathcal{O}(N \log N)$, this method has been widely used in recent years to solve extremely large electromagnetics problems. In MLFMA, an oct-tree structure of L levels is constructed by placing the object in a cubic box and dividing the geometry into clusters (sub-boxes), recursively. The recursion continues until the the minimum size of the lowest-level clusters becomes approximately

one-fourth of a wavelength ($\lambda/4$). Then, MLFMA computes the interaction between testing and basis functions in a group-by-group manner. The required MVM is decomposed into two parts as

$$\overline{\boldsymbol{Z}}_{(N\times N)} \cdot \boldsymbol{a} = \overline{\boldsymbol{Z}}^{\mathrm{NF}}_{(N\times N)} \cdot \boldsymbol{a} + \overline{\boldsymbol{Z}}^{\mathrm{FF}}_{(N\times N)} \cdot \boldsymbol{a}. \tag{2.28}$$

In (2.28), $\overline{\boldsymbol{Z}}^{\mathrm{NF}}_{(N\times N)}$ denotes the near-field interactions (NFI) matrix, whereas $\overline{\boldsymbol{Z}}^{\mathrm{FF}}_{(N\times N)}$ stands for the far-field interactions (FFI) matrix. The NFI matrix is calculated directly using the ordinary MOM and stored in memory in a data-sparse format. MLFMA efficiently performs the multiplication of the coefficient vector and the FFI matrix in three stages: aggregation, translation, and disaggregation. In MVM, these stages are performed on the tree structure in a multilevel scheme [16].

In the aggregation stage, the radiated fields of clusters are calculated from the bottom of the tree structure to the highest level. At the lowest level, the radiated field of a cluster is achieved by adding the radiation patterns of the basis functions inside each cluster. These radiation patterns are computed and stored in memory in the setup phase of the program. At higher levels, the radiated field of each cluster originates from the fields of its sub-clusters. These radiated fields are translated into incoming fields in the translation stage. Finally, in the disaggregation stage, the total incoming fields at the cluster centers are calculated from the top of the tree to the lowest level. In the lowest level, the incoming fields are received by testing functions [28].

In MLFMA, all fields are sampled on the unit sphere as a function of θ and ϕ. The number of samples required for each cluster is proportional to the size of the cluster and depends on the desired number of accurate digits. Because we set the minimum cluster size at the lowest level (as 0.25λ), the cluster size at level ℓ is $a_\ell = 2^{\ell-3}\lambda$. The total number of sampling points is of $\mathcal{O}(T_\ell^2)$. In the worst-case scenario, T_ℓ can be computed via

$$T_\ell \approx 1.73ka_\ell + 2.16(d_0)^{2/3}(ka_\ell)^{1/3}, \tag{2.29}$$

where d_0 is the number of accurate digits desired [28].

2.6 Surface Operators and Secondary Fields

To formulate scattering and radiation problems, three different operators are defined:

$$\boldsymbol{\mathcal{T}}\{\boldsymbol{X}\}(\boldsymbol{r}) = ik\int_S d\boldsymbol{r}'\big[\boldsymbol{X}(\boldsymbol{r}') + \frac{1}{k^2}\nabla'\cdot\boldsymbol{X}(\boldsymbol{r}')\nabla\big]g(\boldsymbol{r},\boldsymbol{r'}), \tag{2.30}$$

$$\boldsymbol{\mathcal{K}}\{\boldsymbol{X}\}(\boldsymbol{r}) = \int_S d\boldsymbol{r}'\boldsymbol{X}(\boldsymbol{r}')\times\nabla' g(\boldsymbol{r},\boldsymbol{r'}) \text{ and} \tag{2.31}$$

$$\boldsymbol{\mathcal{I}}\{\boldsymbol{X}\}(\boldsymbol{r}) = \boldsymbol{X}(\boldsymbol{r}), \tag{2.32}$$

where S is the surface of the 3-D object with an arbitrary shape. In (2.30)-(2.32), $\boldsymbol{X}$ is either the equivalent electric current ($\boldsymbol{J}$) or the equivalent magnetic current ($\boldsymbol{M}$) on the surface, $k = \omega\sqrt{\mu\epsilon}$ is the wavenumber, and $g(\boldsymbol{r},\boldsymbol{r'})$ denotes the scalar Green's function. Decomposing the operator $\boldsymbol{\mathcal{K}}$ in (2.31) into its principal value and limit parts yields

$$\boldsymbol{\mathcal{K}}\{\boldsymbol{X}\}(\boldsymbol{r}) = \boldsymbol{\mathcal{K}}_{\mathrm{PV}}\{\boldsymbol{X}\}(\boldsymbol{r}) - \frac{\Omega_i}{4\pi}\boldsymbol{\mathcal{I}}^{\times n}\{\boldsymbol{X}\}(\boldsymbol{r}), \tag{2.33}$$

where Ω_i denotes the internal solid angle, and

$$\boldsymbol{\mathcal{I}}^{\times n}\{\boldsymbol{X}\}(\boldsymbol{r}) = \hat{\boldsymbol{n}}\times\boldsymbol{X}(\boldsymbol{r}). \tag{2.34}$$

In (2.34), $\hat{\boldsymbol{n}}$ is the outward unit vector. By imposing the boundary condition on the electric field ($\boldsymbol{E}$), the radiated electric field can be calculated as

$$\boldsymbol{E}(\boldsymbol{r}) = \eta\boldsymbol{\mathcal{T}}\{\boldsymbol{J}\}(\boldsymbol{r}) - \boldsymbol{\mathcal{K}}_{\mathrm{PV}}\{\boldsymbol{M}\}(\boldsymbol{r}) + \frac{\Omega_i}{4\pi}\boldsymbol{\mathcal{I}}^{\times n}\{\boldsymbol{M}\}(\boldsymbol{r}), \tag{2.35}$$

and a dual expression results by doing the same on the magnetic field ($\boldsymbol{H}$):

$$\boldsymbol{H}(\boldsymbol{r}) = \frac{1}{\eta}\boldsymbol{\mathcal{T}}\{\boldsymbol{M}\}(\boldsymbol{r}) + \boldsymbol{\mathcal{K}}_{\mathrm{PV}}\{\boldsymbol{J}\}(\boldsymbol{r}) - \frac{\Omega_i}{4\pi}\boldsymbol{\mathcal{I}}^{\times n}\{\boldsymbol{J}\}(\boldsymbol{r}), \tag{2.36}$$

where η is the wave impedance. In (2.35) and (2.36), $\boldsymbol{J}$ and $\boldsymbol{M}$ are the equivalent surface currents, defined as

$$\boldsymbol{J}(\boldsymbol{r}) = \boldsymbol{\mathcal{I}}^{\times n}\{\boldsymbol{H}\}(\boldsymbol{r}) = \hat{\boldsymbol{n}}\times\boldsymbol{H}(\boldsymbol{r}) \text{ and} \tag{2.37}$$

$$\boldsymbol{M}(\boldsymbol{r}) = -\boldsymbol{\mathcal{I}}^{\times n}\{\boldsymbol{E}\}(\boldsymbol{r}) = -\hat{\boldsymbol{n}}\times\boldsymbol{E}(\boldsymbol{r}). \tag{2.38}$$

The fields in (2.35) and (2.36) can be expressed in terms of the spherical components or can be referred to a Cartesian system [22].

2.7 Radar Cross Section

Once we solve the unknown coefficients using MLFMA, the induced surface electric currents are determined. We can compute the secondary electric field (the scattered field) on the surface of a PEC scatterer using (2.35):

$$\boldsymbol{E}^s(\boldsymbol{r}) = \eta \boldsymbol{\mathcal{T}}\{\boldsymbol{J}\}(\boldsymbol{r}). \tag{2.39}$$

Then, the copolar and cross-polar components of the RCS are computed as

$$\begin{bmatrix} \sigma_{\phi\phi} & \sigma_{\phi\theta} \\ \sigma_{\theta\phi} & \sigma_{\theta\theta} \end{bmatrix} = \lim_{r\to\infty} 4\pi r^2 \begin{bmatrix} \left|\frac{\boldsymbol{E}_\phi^{s\,2}(\boldsymbol{r})}{\boldsymbol{E}_\phi^{i\,2}(\boldsymbol{r})}\right| & \left|\frac{\boldsymbol{E}_\phi^{s\,2}(\boldsymbol{r})}{\boldsymbol{E}_\theta^{i\,2}(\boldsymbol{r})}\right| \\ \left|\frac{\boldsymbol{E}_\theta^{s\,2}(\boldsymbol{r})}{\boldsymbol{E}_\phi^{i\,2}(\boldsymbol{r})}\right| & \left|\frac{\boldsymbol{E}_\theta^{s\,2}(\boldsymbol{r})}{\boldsymbol{E}_\theta^{i\,2}(\boldsymbol{r})}\right| \end{bmatrix}, \tag{2.40}$$

where $\boldsymbol{E}^i(\boldsymbol{r})$ is the incident field, and the scattered field $\boldsymbol{E}^s(\boldsymbol{r})$ is computed in the far zone of the target [4].

The unit of the RCS is area, and one the most common references is one meter squared. Therefore, a designation is dB/m^2 (or dBsm) for a 3-D RCS. When the transmitter and receiver are at the same location, the RCS is usually referred to as monostatic (or back-scattered), and as bistatic when the two are at different locations. A plot of the RCS as a function of the space coordinates is usually referred to as the RCS pattern [29].

2.8 Angular Resolution

The minimum number of monostatic observation angles to obtain a full representation of back-scattered fields can be determined by the Nyquist-Shannon sampling theorem. The maximum angular resolution just depends on the geometry of object. For an object with a maximum diameter of d, the maximum angular resolution can be calculated via

$$\Delta\theta_{\max} = \Delta\phi_{\max} = \frac{\lambda}{2d}, \tag{2.41}$$

where λ is the corresponding wavelength of the incident plane wave. In some applications, we require a specific angular response resolution, which means a

higher sampling rate of incident angles is needed. Hence, we enforce high linear dependencies between columns of the RHS matrix [30].

Chapter 3

Implementation

3.1 Adaptive Cross Approximation

The ACA algorithm [14] takes advantage of the rank-deficient nature of matrices that contain linear dependencies and computes a low-rank decomposition successively with a finite number of rank-1 approximations. Hence, a matrix of this type can be approximated by a low-rank representation [19]. A schematic view of the ACA algorithm is illustrated in Fig. 3.1.

SVD has a computational complexity of $\mathcal{O}(N^3)$, where N is the number of unknowns. The considered full matrix must also be computed and stored explicitly [15].

One main advantage of the ACA algorithm is its purely algebraic nature. As opposed to SVD, development and implementation of ACA do not depend on the full knowledge of the matrix [32]. Moreover, due to its algebraic nature, ACA can be modular and very easily integrated into existing codes. The complexity of ACA is $\mathcal{O}(k^2(m+n))$ and needs only $k(n+m)$ elements to be calculated.

The aim of the ACA algorithm is to decompose a rank-deficient matrix $\overline{\boldsymbol{A}}$

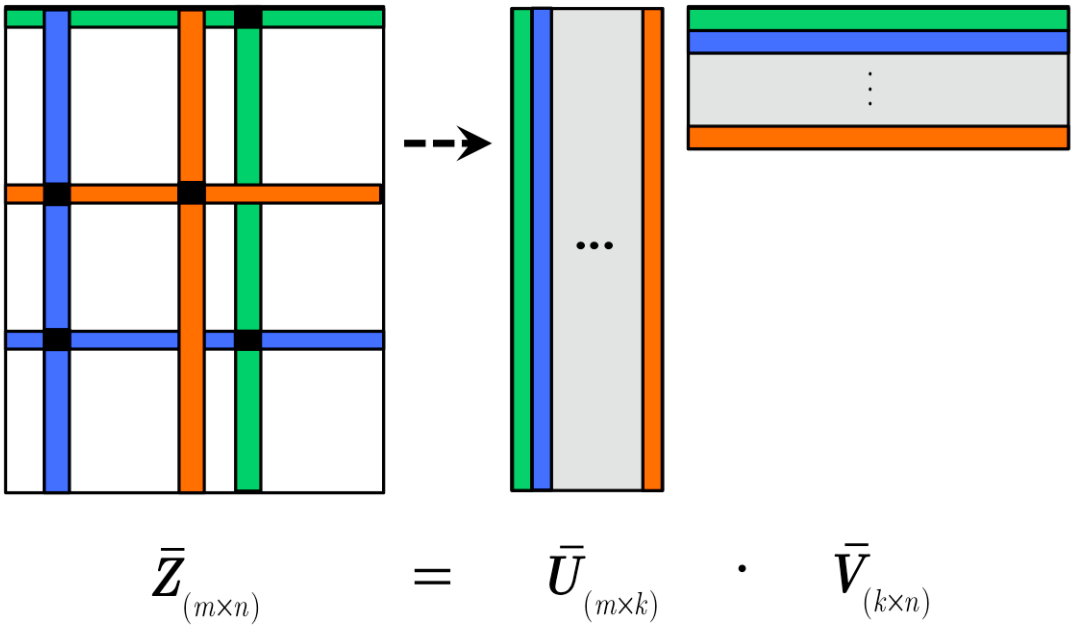

Figure 3.1: A schematic view of the ACA algorithm.

into

$$\overline{\boldsymbol{A}}_{(m\times n)} = \overline{\boldsymbol{U}}_{(m\times k)} \cdot \overline{\boldsymbol{V}}_{(k\times n)} + \overline{\boldsymbol{R}}_{(m\times n)}, \tag{3.1}$$

where $\overline{\boldsymbol{U}} \in \mathbb{C}^{m\times k}$, $\overline{\boldsymbol{V}} \in \mathbb{C}^{k\times n}$, and $\overline{\boldsymbol{R}} \in \mathbb{C}^{m\times n}$ is the error matrix derived from the approximation. The decomposition provided in (3.1) approximates $\overline{\boldsymbol{A}}$ with $\tilde{\boldsymbol{A}}$ by multiplying two matrices of rank at most k. Details of the ACA algorithm are as follows:

Initialize the first row index, I_1, and set $\tilde{\boldsymbol{A}} = 0$. Then, initialize the first row of the approximate error matrix as

$$\tilde{\boldsymbol{R}}(I_1, :) = \overline{\boldsymbol{A}}(I_1, :). \tag{3.2}$$

The first column index, J_1, should be chosen as the column index of the maximum element of $\tilde{\boldsymbol{R}}(I_1, :)$. Normalizing $\tilde{\boldsymbol{R}}(I_1, :)$ with $\tilde{\boldsymbol{R}}(I_1, J_1)$ results in the first row of $\overline{\boldsymbol{V}}$, i.e.,

$$\overline{\boldsymbol{V}}(1, :) = \tilde{\boldsymbol{R}}(I_1, :)/\tilde{\boldsymbol{R}}(I_1, J_1). \tag{3.3}$$

Fill the first column of the approximate error matrix using the J_1th column of $\overline{\boldsymbol{A}}$. This forms the first column of $\overline{\boldsymbol{U}}$. The second row index is the index of

Algorithm 1 Partially Pivoted Adaptive Cross Approximation

Let $k := 1$;
$F := 0$
repeat
  find i_k
  $\tilde{\boldsymbol{v}}_k := \overline{\boldsymbol{A}}_{(m\times n)}(i_k, 1:n)$
  for $\ell := 1, ..., k-1$ **do** $\tilde{\boldsymbol{v}}_k := \tilde{\boldsymbol{v}}_k - (\boldsymbol{u}_\ell(i_k))\boldsymbol{v}_\ell$
    $F := F \cup \{i_k\}$
    if $\tilde{\boldsymbol{v}}_k$ does not vanish **then**
      $j_k := \mathbf{argmax}_{j=1,...,n}|\tilde{\boldsymbol{v}}_k(j)|$
      $\boldsymbol{v}_k := (\tilde{\boldsymbol{v}}_k)(j_k)^{-1}\tilde{\boldsymbol{v}}_k$
      $\boldsymbol{u}_k := \overline{\boldsymbol{A}}_{(m\times n)}(1:m, j_k)$
      for $\ell := 1...k-1$ **do** $\tilde{\boldsymbol{v}}_k := \boldsymbol{u}_k - (\boldsymbol{v}_\ell(j_k))\boldsymbol{u}_\ell$
      end for
      $k := k + 1$
    end if
  end for
until the stoping criterion (3.6) is fulfilled or $F = \{1, ..., m\}$.

the maximum element in the first column of the approximate error matrix. We should calculate the norm of the two rank-1 approximants of $\overline{\boldsymbol{A}}$ and check the stopping criterion. If the criterion is not satisfied, the iterations will start. At the rth iteration, we update the I_rth row of the approximate error matrix as

$$\tilde{\boldsymbol{R}}(I_r, :) = \overline{\boldsymbol{A}}(I_r, :) - \sum_{\ell=1}^{r-1} \overline{\boldsymbol{U}}(I_r, \ell) \cdot \overline{\boldsymbol{V}}(\ell, :). \tag{3.4}$$

The index of the maximum element of $\tilde{\boldsymbol{R}}(I_r, :)$ is J_r. Similar to the initialization step, the rth row of $\overline{\boldsymbol{V}}$ is the I_rth row of the approximate matrix that is normalized by $\tilde{\boldsymbol{R}}(I_r, :)/\tilde{\boldsymbol{R}}(I_r, J_r)$. In the next step, the J_rth column of $\overline{\boldsymbol{U}}$ is updated and we obtain the rth column of $\overline{\boldsymbol{U}}$, i.e.,

$$\overline{\boldsymbol{U}}(:, J_r) = \tilde{\boldsymbol{R}}(:, J_r) = \overline{\boldsymbol{A}}(:, J_r) - \sum_{\ell=1}^{r-1} \overline{\boldsymbol{V}}(\ell, J_r) \cdot \overline{\boldsymbol{U}}(:, \ell). \tag{3.5}$$

The stopping criterion controls wheter the condition

$$||\overline{\boldsymbol{U}}(:, J_r) \cdot \overline{\boldsymbol{V}}(I_r, :)|| \leq \epsilon ||\tilde{\boldsymbol{A}}^{(r)}|| \tag{3.6}$$

is satisfied for a given threshold ϵ or not. In (3.6), $||\cdot||$ denotes the matrix

Frobenius norm and $||\tilde{\boldsymbol{A}}||$ is computed as

$$\begin{aligned} ||\tilde{\boldsymbol{A}}^{(r)}||^2 &= ||\tilde{\boldsymbol{A}}^{(r-1)}||^2 \\ &+ 2\sum_{j=1}^{r-1} |\overline{\boldsymbol{U}}^H(:,j)\cdot\overline{\boldsymbol{U}}(:,r)|\cdot|\overline{\boldsymbol{V}}^H(j,:)\cdot\overline{\boldsymbol{V}}(r,:)| \\ &+ ||\overline{\boldsymbol{U}}(:,r)||^2\cdot||\overline{\boldsymbol{V}}(r,:)||^2. \end{aligned} \tag{3.7}$$

The iterations continue until the criterion in (3.6) is fulfilled. In each iteration, we choose rows and columns of $\overline{\boldsymbol{A}}$ that are different from previous iterations.

The ACA algorithm terminates when the error matrix (not the approximate error matrix) at the kth iteration has a norm less than the product of the norm of $\overline{\boldsymbol{A}}$ and a prescribed threshold ϵ, i.e.,

$$||\overline{\boldsymbol{R}}|| \leq \epsilon||\overline{\boldsymbol{A}}||. \tag{3.8}$$

The criterion defined in (3.8) requires complete knowledge of $\overline{\boldsymbol{A}}$, which is why we replace this criterion with the one in (3.6). In this way, a low-rank approximation of the original matrix is obtained by requiring only partial knowledge of $\overline{\boldsymbol{A}}$. In other words, the memory and the CPU time scale as $\mathcal{O}(k(m+n))$ and $\mathcal{O}(k^2(m+n))$, respectively. The efficient rank k will be further reduced by means of the recompression technique [32].

3.2 QR Decomposition

The QR factorization of a matrix $\overline{\boldsymbol{A}} \in \mathbb{C}^{m\times n}$, with $m \geq n$, is given by

$$\overline{\boldsymbol{A}} = \overline{\boldsymbol{Q}}\cdot\overline{\boldsymbol{R}} = \overline{\boldsymbol{Q}}\cdot\begin{bmatrix}\overline{\boldsymbol{R}}_1\\ 0\end{bmatrix}, \tag{3.9}$$

where $\overline{\boldsymbol{Q}} \in \mathbb{C}^{m\times m}$ is a unitary matrix, $\overline{\boldsymbol{R}} \in \mathbb{C}^{m\times n}$, and $\overline{\boldsymbol{R}}_1 \in \mathbb{C}^{n\times n}$ is an upper triangular matrix. If $\overline{\boldsymbol{A}}$ is non-singular, then $\overline{\boldsymbol{R}}_1$ is of full rank n. The factorization in (3.9) can be represented as

$$\overline{\boldsymbol{A}} = \begin{bmatrix}\overline{\boldsymbol{Q}}_1 & \overline{\boldsymbol{Q}}_2\end{bmatrix}\cdot\begin{bmatrix}\overline{\boldsymbol{R}}_1\\ 0\end{bmatrix} = \overline{\boldsymbol{Q}}_1\cdot\overline{\boldsymbol{R}}_1, \tag{3.10}$$

where $\overline{\boldsymbol{Q}}_1 \in \mathbb{C}^{m\times n}$ consists of the first n columns of $\overline{\boldsymbol{Q}}$.

The matrices $\overline{\boldsymbol{Q}}$ and $\overline{\boldsymbol{R}}$ can be constructed by means of Householder transformations. It is known that the Householder method is numerically stable because of the orthogonalization process. The unitary matrix $\overline{\boldsymbol{Q}}$ can be written as a product of the elementary Householder matrices:

$$\overline{\boldsymbol{Q}} = \overline{\boldsymbol{H}}_1 \cdot \overline{\boldsymbol{H}}_2 \cdots \overline{\boldsymbol{H}}_s, \tag{3.11}$$

where $s = \min(n-1, m-1)$. In (3.11), $\overline{\boldsymbol{H}}_i$ ($i \in \{1, 2, \cdots, s\}$) is an unitary matrix of the form

$$\overline{\boldsymbol{H}}_i = \overline{\boldsymbol{I}}_n - \frac{2}{||\boldsymbol{v}||^2}\boldsymbol{v}\boldsymbol{v}^H, \tag{3.12}$$

where $\boldsymbol{v}$ is a column vector (called the Householder vector). For a rank-deficient matrix $\overline{\boldsymbol{A}}$, QR factorization will not provide a basis for the range of $\overline{\boldsymbol{A}}$. To overcome this problem, a column-permuted version of $\overline{\boldsymbol{A}}$ can be factorized [15].

3.3 Singular Value Decomposition

The SVD of the matrix $\overline{\boldsymbol{A}} \in \mathbb{C}^{m\times n}$ is

$$\overline{\boldsymbol{A}}_{(m\times n)} = \overline{\boldsymbol{U}}_{\Sigma(m\times m)} \cdot \overline{\boldsymbol{\Sigma}}_{(m\times n)} \cdot \overline{\boldsymbol{V}}^H_{\Sigma(n\times n)}, \tag{3.13}$$

where $\overline{\boldsymbol{U}}_\Sigma \in \mathbb{C}^{m\times m}$ and $\overline{\boldsymbol{V}}_\Sigma \in \mathbb{C}^{n\times n}$ are unitary matrices. In (3.13), $\overline{\boldsymbol{\Sigma}} \in \mathbb{C}^{m\times n}$ is

$$\overline{\boldsymbol{\Sigma}} = \mathrm{diag}(\sigma_1, \cdots, \sigma_p), \tag{3.14}$$

where $p = \min\{m, n\}$ and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0$. Defining q as the rank of $\overline{\boldsymbol{A}}$ by

$$\sigma_1 \geq \cdots \geq \sigma_q > \sigma_{q+1} = \cdots = \sigma_p = 0, \tag{3.15}$$

the SVD of $\overline{\boldsymbol{A}}$ can be represented as the summation of the outer product of q rank-1 matrices. The summation can be truncated by $k' < q = \mathrm{rank}(\overline{\boldsymbol{A}})$ as

$$\tilde{\boldsymbol{A}} = \sum_{i=1}^{k'} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^H, \tag{3.16}$$

where $\tilde{\boldsymbol{A}}$ is the low-rank approximation of $\overline{\boldsymbol{A}}$ with an effective rank k'.

The matrix $\tilde{\boldsymbol{A}}$ is called a truncation of $\overline{\boldsymbol{A}}$ to rank k'. We will refer to this truncation operation as a reduced SVD (RSVD). The SVD method is not a practical approach for computing low-rank approximations because it has a high computational cost and requires complete knowledge of the all matrix elements. However, we will take advantage of SVD to recompress the matrices obtained from ACA [15].

3.4 Recompression Technique

In this section, we present a recompression technique that reduces the required amount of storage and hence accelerates the solution of the problem. Although ACA generates high-quality approximations, the amount of required storage can still be reduced [33, 34].

Using RSVD, we present an efficient algorithm (Algorithm 2) for the recompression of a low-rank matrix $\overline{\boldsymbol{A}}_{(m\times n)} \approx \overline{\boldsymbol{U}}_{(m\times k)} \cdot \overline{\boldsymbol{V}}_{(k\times n)}$, such that $\tilde{\boldsymbol{A}}_{(m\times n)} \approx \tilde{\boldsymbol{U}}_{(m\times k')} \cdot \tilde{\boldsymbol{V}}_{(k'\times n)}$ with rank k' is less than k. In other words, rank k' will be replaced by rank k, obtained from ACA, by comparing the singular values to the given threshold ϵ_{SVD}. The method uses the QR decomposition and calculates the SVD in a very efficient manner [33]. Assume we have computed the QR decompositions

$$\overline{\boldsymbol{U}} = \overline{\boldsymbol{Q}}_U \cdot \overline{\boldsymbol{R}}_U \tag{3.17}$$

and

$$\overline{\boldsymbol{V}} = \overline{\boldsymbol{R}}_V^H \cdot \overline{\boldsymbol{Q}}_V^H \tag{3.18}$$

of $\overline{\boldsymbol{U}} \in \mathbb{C}^{n\times k}$ and $\overline{\boldsymbol{V}} \in \mathbb{C}^{k\times m}$, respectively. According to [15], this can be done with $4k^2(m+n) - \frac{8}{3}k^3$ operations. The outer product $\overline{\boldsymbol{R}}_U \cdot \overline{\boldsymbol{R}}_V^H$ of the two upper triangular matrices $\overline{\boldsymbol{R}}_U$ and $\overline{\boldsymbol{R}}_V$ needs $\frac{k}{3}(2k^2 + \frac{11}{2}k - 1)$ operations [33]. The result of the outer product is then decomposed using SVD into

$$\overline{\boldsymbol{R}}_U \cdot \overline{\boldsymbol{R}}_V^H = \hat{\boldsymbol{U}} \cdot \hat{\boldsymbol{\Sigma}} \cdot \hat{\boldsymbol{V}}^H, \tag{3.19}$$

Algorithm 2 Recompression Algorithm

procedure RECOMPRESS($\overline{\boldsymbol{U}}, \overline{\boldsymbol{V}}, \epsilon$)
Compute a truncated QR factorization of $\overline{\boldsymbol{U}}$ and $\overline{\boldsymbol{V}}$:
$\overline{\boldsymbol{U}} = \overline{\boldsymbol{Q}}_U \cdot \overline{\boldsymbol{R}}_U$;
$\overline{\boldsymbol{V}}^H = \overline{\boldsymbol{Q}}_V \cdot \overline{\boldsymbol{R}}_V$;
$\hat{\boldsymbol{R}} = \overline{\boldsymbol{R}}_U \cdot \overline{\boldsymbol{R}}_V^H$;
Compute an SVD of $\hat{\boldsymbol{R}}$:
$\hat{\boldsymbol{R}} = \hat{\boldsymbol{U}} \cdot \hat{\boldsymbol{\Sigma}} \cdot \hat{\boldsymbol{V}}^H$;
$k' = 1$;
while $\sigma_{k'} > \epsilon\sigma_1$ **do**
$k' = k' + 1$
end while
$\overline{\boldsymbol{U}}^{\mathrm{RACA}}_{(m\times k')} = \overline{\boldsymbol{Q}}_U \cdot \hat{\boldsymbol{U}}$;
$\overline{\boldsymbol{V}}^{\mathrm{RACA},H}_{(n\times k')} = \overline{\boldsymbol{Q}}_V \cdot \hat{\boldsymbol{V}} \cdot \hat{\boldsymbol{\Sigma}}$;
end procedure

where the cost of the above operation is $22k^3$. Note that $\overline{\boldsymbol{Q}}_U \cdot \hat{\boldsymbol{U}}$ and $\overline{\boldsymbol{Q}}_V \cdot \hat{\boldsymbol{V}}$ both have orthonormal columns, and hence the resulting decomposition

$$\overline{\boldsymbol{A}} \approx \overline{\boldsymbol{U}} \cdot \overline{\boldsymbol{V}} = (\overline{\boldsymbol{Q}}_U \cdot \hat{\boldsymbol{U}}) \cdot \hat{\boldsymbol{\Sigma}} \cdot (\overline{\boldsymbol{Q}}_V \cdot \hat{\boldsymbol{V}})^H \tag{3.20}$$

is an SVD of $\overline{\boldsymbol{A}}$. Together with the multiplications of $\overline{\boldsymbol{Q}}_U \cdot \hat{\boldsymbol{U}}$ and $\overline{\boldsymbol{Q}}_V \cdot \hat{\boldsymbol{V}}$, which require $k(2k-1)(m+n)$ operations, the number of operations sum up to approximately $6k^2(m+n) + 20k^3$ operations. As a conclusion, the recompressed low-rank approximation of the matrix $\overline{\boldsymbol{A}}_{(m\times n)}$ can be computed within $\mathcal{O}(k^2(m+n) + k^3)$ operations; this is much lower than the cost of computing the SVD of $\overline{\boldsymbol{A}}_{(m\times n)}$ with $14mn^2 + 8n^3$ complex operations [15]. After all these computations, an RSVD will be applied to replace rank k with a smaller rank k'. After this recompression the final low-rank approximation will be in the form of

$$\overline{\boldsymbol{A}}_{(m\times n)} \approx \overline{\boldsymbol{U}}^{\mathrm{RACA}}_{(m\times k')} \cdot \overline{\boldsymbol{V}}^{\mathrm{RACA}}_{(k'\times n)}. \tag{3.21}$$

3.5 Application in Multiple Right-Hand Side Problems

As discussed earlier, SIEs can be converted into a matrix equation for multiple excitation problems using MOM. The matrix equation can be represented as

$$\overline{\boldsymbol{Z}}_{(N\times N)} \cdot \overline{\boldsymbol{X}}_{(N\times M)} = \overline{\boldsymbol{V}}_{(N\times M)}, \tag{3.22}$$

where $\overline{\boldsymbol{Z}} \in \mathbb{C}^{N\times N}$ is the matrix of interactions between N testing and basis functions, $\overline{\boldsymbol{X}} \in \mathbb{C}^{N\times M}$ contains unknown coefficient vectors corresponding to the excitation vectors in the RHS matrix $\overline{\boldsymbol{V}} \in \mathbb{C}^{N\times M}$, and M is the number of excitations. For the case of multiple 2-D MRCSs, the geometry is illuminated by M plane waves from different spherical directions (θ_i, ϕ_i), where $\theta_i \in [\theta_{\min}, \theta_{\max}]$ and $\phi_i \in [\phi_{\min}, \phi_{\max}]$. The solution of (3.22) for one RHS can be accelerated by MLFMA. Calculating multiple MRCSs requires multiple solutions. Despite the reduced complexity of MLFMA, the solutions for multiple RHSs is a time-consuming process [16].

As discussed in Section 2.8, in some applications we require a specific angular response resolution, which means a higher sampling rate of incident angles. Hence, we enforce high linear dependencies between columns of the RHS matrix in (3.22). Using this fact, we can reduce the number of solutions from M to k' that are defined as the effective rank of $\overline{\boldsymbol{V}}$.

Various techniques can be incorporated to obtain a low-rank approximation of the RHS matrix because of its rank-deficient nature. One method is to use SVD [7]. For a given threshold, SVD results in a low-rank approximation of a matrix with the minimum rank [15]. Another technique is to use rank-revealing QR decomposition [6]. Both methods have two main disadvantages: they have a high computational cost and require full knowledge of the RHS matrix elements.

An alternative technique for finding the low-rank approximation of a matrix is ACA [13], which only requires partial knowledge of the matrix. Furthermore, the computational cost of ACA is much less than the SV and QR decompositions. For the RHS matrix, $\overline{\boldsymbol{V}}_{(N\times M)}$, the memory usage complexities of QR factorization

and SVD scales as $\mathcal{O}(NM)$ because these methods require complete knowledge of $\overline{\boldsymbol{V}}$. However, ACA has a memory complexity of $\mathcal{O}(k(M+N))$, where k is the effective rank of $\overline{\boldsymbol{V}}$ obtained by ACA and $k < \min(N, M)$. The computational complexities of SV and QR decompositions are $\mathcal{O}(\max(N, M)M^2)$. The required operations for ACA scales as $\mathcal{O}(k^2(M+N))$.

Despite the fact that SVD leads to a minimum-rank approximation [15], it can be applied only on relatively small problems due to its high computational cost. Compressing the RHS matrix using ACA will take much less time but there will be some redundant columns in the resulting decomposed matrices. We can apply a recompression technique to reach a decomposition of a lower rank k' compared to k, obtained from ACA.

For a given threshold ϵ, we use ACA to compress and factorize the RHS matrix. Applying ACA on $\overline{\boldsymbol{V}}$ leads to

$$\overline{\boldsymbol{V}}_{(N\times M)} \approx \overline{\boldsymbol{A}}_{(N\times k)} \cdot \overline{\boldsymbol{B}}_{(k\times M)}, \tag{3.23}$$

where $\overline{\boldsymbol{A}}_{(N\times k)} \in \mathbb{C}^{N\times k}$, $\overline{\boldsymbol{B}}_{(M\times k)} \in \mathbb{C}^{M\times k}$, and k denotes the effective rank of $\overline{\boldsymbol{V}}$. To obtain a decomposition of $\overline{\boldsymbol{V}}$, with an effective rank k' less than k, we employ the recompression technique in Algorithm 2. The method uses the QR decomposition and calculates SVD in an efficient manner. We apply a QR factorization for $\overline{\boldsymbol{A}}$ to get

$$\overline{\boldsymbol{A}}_{(N\times k)} = \overline{\boldsymbol{Q}}_{A(N\times k)} \cdot \overline{\boldsymbol{R}}_{A(k\times k)}, \tag{3.24}$$

and for $\overline{\boldsymbol{B}}$ to obtain

$$\overline{\boldsymbol{B}}^{H}_{(M\times k)} = \overline{\boldsymbol{Q}}_{B(M\times k)} \cdot \overline{\boldsymbol{R}}_{B(k\times k)}, \tag{3.25}$$

where matrices $\overline{\boldsymbol{Q}}_A$ and $\overline{\boldsymbol{Q}}_B$ are unitary matrices. Substituting (3.24) and (3.25) in (3.23) leads to

$$\overline{\boldsymbol{V}}_{(N\times M)} \approx \overline{\boldsymbol{Q}}_{A(N\times k)} \cdot \overline{\boldsymbol{R}}_{A(k\times k)} \cdot \overline{\boldsymbol{R}}^{H}_{B(k\times k)} \cdot \overline{\boldsymbol{Q}}^{H}_{B(M\times k)}. \tag{3.26}$$

Then, we perform a truncated SVD with a given threshold ϵ on the product of $\overline{\boldsymbol{R}}_A$ and $\overline{\boldsymbol{R}}^{H}_B$ so that (3.23) becomes

$$\overline{\boldsymbol{V}} \approx \overline{\boldsymbol{A}} \cdot \overline{\boldsymbol{B}} \approx \overline{\boldsymbol{Q}}_A \cdot \hat{\boldsymbol{A}} \cdot \left(\overline{\boldsymbol{Q}}_B \cdot \hat{\boldsymbol{B}} \cdot \hat{\boldsymbol{\Sigma}}^H\right)^H, \tag{3.27}$$

and we have already accomplished the recompression process. By defining

$$\overline{\boldsymbol{A}}^{\mathrm{RACA}}_{(N\times k')} = \overline{\boldsymbol{Q}}_A \cdot \hat{\boldsymbol{A}} \tag{3.28}$$

and

$$\overline{\boldsymbol{B}}^{\mathrm{RACA},H}_{(k'\times M)} = \overline{\boldsymbol{Q}}_B \cdot \hat{\boldsymbol{B}} \cdot \hat{\boldsymbol{\Sigma}}^H, \tag{3.29}$$

the desired decomposition is achieved. After this recompression, the final low-rank approximation will be in the form of

$$\overline{\boldsymbol{V}}_{(N\times M)} \approx \overline{\boldsymbol{A}}^{\mathrm{RACA}}_{(N\times k')} \cdot \overline{\boldsymbol{B}}^{\mathrm{RACA}}_{(k'\times M)}, \tag{3.30}$$

where k' is the effective rank of the RHS matrix and is less than the rank k achieved by ACA.

The proposed recompression technique can be computed within $\mathcal{O}(k^2(N+M)+k^3)$ operations. This way, we have computed the low-rank approximation of $\overline{\boldsymbol{V}}$ with a lower complexity compared to SVD. Furthermore, the achieved effective rank, k', is very close to the rank that we could obtain from ordinary SVD.

We can utilize the low-rank approximation of the RHS matrix, which we obtained in (3.30), to accelerate the solution of the matrix equation in (3.22). An approximate solution to (3.22) can be achieved by substituting (3.30) in (3.22) and rewriting it as

$$\overline{\boldsymbol{X}} \approx \left(\overline{\boldsymbol{Z}}^{-1} \cdot \overline{\boldsymbol{A}}\right) \cdot \overline{\boldsymbol{B}}, \tag{3.31}$$

where $\overline{\boldsymbol{Z}}^{-1}$ denotes a standard solution, i.e., MLFMA in this work, but other fast methods can also be used to accelerate the solution [16, 17, 18, 19]. The main advantage of the formulation in (3.31) is that it reduces the solution time immensely, because the number of solutions decreases from M to k'. In other words, after solving

$$\hat{\boldsymbol{X}} = \overline{\boldsymbol{Z}}^{-1} \cdot \overline{\boldsymbol{A}}, \tag{3.32}$$

where $\hat{\boldsymbol{X}} \in \mathbb{C}^{N\times k'}$, we can construct $\overline{\boldsymbol{X}}$ from

$$\overline{\boldsymbol{X}} \approx \hat{\boldsymbol{X}} \cdot \overline{\boldsymbol{B}}. \tag{3.33}$$

3.6 Review of Other Methods

In this section, we will review other methods for the fast solution of multiple radar cross section problems. In a major classification, we can divide the solutions into three types: algebraic methods, function approximation methods, and interpolation techniques. The provided method in this thesis, i.e., RACA, is a purely algebraic method.

3.6.1 Algebraic Methods

Algebraic methods benefit from redundancies in the RHS matrix. These methods are essentially based on the compression of the excitation matrix assembled over all plane wave incident angles using algebraic techniques. Removing the redundancies in the excitation matrix using QR decomposition [6], SVD [7], ID [8], and ACA [13] or its recompressed version, RACA, leads to fast RCS computations. As stated in (3.22), for problems with multiple excitations, the matrix equation can be written as

$$\overline{\boldsymbol{Z}}_{(N\times N)} \cdot \overline{\boldsymbol{X}}_{(N\times M)} = \overline{\boldsymbol{V}}_{(N\times M)}. \tag{3.34}$$

Because all algebraic methods compress the RHS matrix, we will provide a new formulation for (3.34) using low-rank approximations of $\overline{\boldsymbol{V}}$. Hence, an approximate solution to (3.34) can be achieved via

$$\overline{\boldsymbol{X}} \approx \left(\overline{\boldsymbol{Z}}^{-1} \cdot \overline{\boldsymbol{A}}\right) \cdot \overline{\boldsymbol{B}}^{H}, \tag{3.35}$$

where $\overline{\boldsymbol{Z}}^{-1}$ denotes a standard solution, i.e., MLFMA in this work. In (3.35), matrices $\overline{\boldsymbol{A}}$ and $\overline{\boldsymbol{B}}$ are obtained from applying a compression method.

3.6.1.1 QR Decomposition

As discussed above, the vectors that span the column space of $\overline{\boldsymbol{V}}$ are much lower than the columns in $\overline{\boldsymbol{V}}$ due to redundancies in $\overline{\boldsymbol{V}}$. Using this fact, the number of required matrix solutions reduces from M to r. The effective rank r depends on

the column rank of $\overline{\boldsymbol{V}}$. Consider a rank revealing QR factorization (RRQR) [35] of $\overline{\boldsymbol{V}}$ with a predefined tolerance, which can be written as

$$\overline{\boldsymbol{V}} \approx \overline{\boldsymbol{Q}} \cdot \overline{\boldsymbol{R}}. \tag{3.36}$$

The matrix equation in (3.34) reduces to (3.35), where $\overline{\boldsymbol{A}} = \overline{\boldsymbol{Q}}$ and $\overline{\boldsymbol{B}}^H = \overline{\boldsymbol{R}}$. The columns of $\overline{\boldsymbol{Q}}$ are orthonormal and approximately span the column space of $\overline{\boldsymbol{V}}$. The modified Gram-Schmit algorithm can be used to obtain the low-rank approximation in (3.36) [6]. Although the method reduces the number of required solutions, it has a high computational cost compared to RACA. Furthermore, all elements of $\overline{\boldsymbol{V}}$ must be available, which increases memory consumption.

3.6.1.2 Singular Value Decomposition

The approximate form in (3.35) can be achieved by applying SVD to the RHS matrix $\overline{\boldsymbol{V}}$. After employing SVD, the RHS matrix will be decomposed into

$$\overline{\boldsymbol{V}} = \overline{\boldsymbol{U}} \cdot (\overline{\boldsymbol{S}} \cdot \boldsymbol{F}), \tag{3.37}$$

where $\overline{\boldsymbol{U}}$ is an $N \times N$ unitary matrix, the matrix $\overline{\boldsymbol{S}}$ is an $N \times M$ orthogonal matrix, and $\boldsymbol{F}$ is an $M \times M$ unitary matrix. By using the Eckart-Young theorem, $\overline{\boldsymbol{V}}$ can be approximated by

$$\overline{\boldsymbol{V}} \approx \overline{\boldsymbol{A}} \cdot \overline{\boldsymbol{B}}^H = \overline{\boldsymbol{U}}_k \cdot (\overline{\boldsymbol{S}}_k \cdot \boldsymbol{F}_k), \tag{3.38}$$

where the matrix $\overline{\boldsymbol{S}}$ contains first k singular values [15]. Hence, the efficient rank of $\overline{\boldsymbol{V}}$, or the rank of matrices $\overline{\boldsymbol{A}}$ and $\overline{\boldsymbol{B}}$, is equal to k, which means only k solutions are required, instead of M solutions ($k < M$). The benefits of solving (3.35) instead of (3.34) depends mainly on the ratio of the effective rank k to the number of plane wave excitations M. For a given threshold ϵ, RSVD results in a very good low-rank approximation but at the cost of high computational complexity. Like RRQR, this method suffers from dependence on all the matrix elements, which increases the memory complexity [7].

3.6.1.3 Adaptive Cross Approximation

To reduce the computational effort of solving (3.34), we have to find the low-rank approximation of $\overline{\boldsymbol{V}}$. As discussed in this paper, one efficient method to obtain a low-rank decomposition of a matrix is ACA. We can bring ACA into play to achieve the following approximation for a given threshold ϵ:

$$\overline{\boldsymbol{V}}_{(N\times M)} \approx \overline{\boldsymbol{A}}_{(N\times k)} \cdot \overline{\boldsymbol{B}}^{H}_{(k\times M)}, \tag{3.39}$$

where $\overline{\boldsymbol{A}}_{(N\times k)} \in \mathbb{C}^{N\times k}$, $\overline{\boldsymbol{B}}_{(M\times k)} \in \mathbb{C}^{M\times k}$, and k denotes the effective rank of $\overline{\boldsymbol{V}}$. The complexity of this algorithm is $\mathcal{O}(k^2(N+M))$. The relation $k < M$ holds for rank-deficient matrices and makes the ACA algorithm efficient compared to the SVD of RRQR. Nevertheless, the effective rank obtained from ACA is less than that of SVD because ACA generates low-rank approximations using a limited number of rank-1 approximations and requires only partial knowledge of the elements of $\overline{\boldsymbol{V}}$ [13]. In this work, we present a recompression technique that leads to a low-rank approximation with an effective rank, which is comparable to that of SVD. The complexity of RACA and ACA are almost the same.

3.6.1.4 Interpolative Decomposition

A low-rank approximation of $\overline{\boldsymbol{V}}$ can be achieved by utilizing the ID and skeleton concepts. The accuracy of approximation obtained from the ID algorithm can be controlled by a threshold. In the ID algorithm, a matrix $\overline{\boldsymbol{A}}$ consists of a subset of columns of $\overline{\boldsymbol{V}}$. There also exists a complex matrix $\overline{\boldsymbol{B}}^{H}$ with the following properties:

- Some subset of the columns of $\overline{\boldsymbol{B}}^{H}$ creates the identity matrix.
- All elements of $\overline{\boldsymbol{B}}^{H}$ have an absolute value less than or equal to 1.
- The kth singular value of $\overline{\boldsymbol{B}}^{H}$ is at least one.
- $||\overline{\boldsymbol{V}} - \overline{\boldsymbol{A}} \cdot \overline{\boldsymbol{B}}^{H}||_2 \leq \sigma_{k+1}\sqrt{k(n-k)+1}$, where σ_{k+1} is the $(k+1)$st singular value of $\overline{\boldsymbol{V}}$.

The last statement provides a low-rank approximation of $\overline{\boldsymbol{V}}$ [36]. Once we compute the low-rank decomposition, the matrix equation in (3.34) will reduce to (3.35). The complexity of the ID algorithm is less than the complexity of SVD, but the former has a higher computational complexity compared to the ACA or RACA algorithms. ID skeletonization is a sub-optimal algorithm [8] and the resulting effective rank is higher than the ranks that can be obtained using the RSVD or RACA algorithms.

3.6.2 Rational Function Approximation

One efficient approach to the solution of RCS problems over a wide range of incident plane waves is the rational function approximation technique. There are three main approaches for creating approximated rational functions: asymptotic waveform evaluation (AWE) [9, 10], the Padé approximation, and model-based parameter estimation (MBPE) [11, 12]. We will briefly describe each method in the following sections.

3.6.2.1 Asymptotic Waveform Evaluation

This method was first developed for time-domain analysis. It has been applied to MOM for the fast solution of electromagnetics problems. The matrix equation for any frequency can be written as

$$\overline{\boldsymbol{Z}}(k) \cdot \boldsymbol{a}(k) = \boldsymbol{v}(k), \tag{3.40}$$

where k is the wavenumber. For a given frequency f_0 and its corresponding wavenumber k_0, the solution of (3.40), i.e., $\boldsymbol{a}(k_0)$, is obtained. The essential idea behind AWE is expanding $\boldsymbol{a}(k)$ into a Taylor series, such as

$$\boldsymbol{a}(k) = \boldsymbol{a}(k_0) + \boldsymbol{a}^{(1)}(k_0)(k-k_0) + \frac{\boldsymbol{a}^{(2)}(k_0)(k-k_0)^2}{2!} + \cdots, \tag{3.41}$$

where $\boldsymbol{a}^{(n)}(k_0)$ is the nth derivative of $\boldsymbol{a}(k)$, evaluated at k_0. (3.41) can be written as

$$\boldsymbol{a}(k) = \sum_{n=0}^{\infty} \boldsymbol{q}_n (k-k_0)^n. \tag{3.42}$$

In (3.42), $\boldsymbol{q}_n$ is the nth moment and can be computed via

$$\boldsymbol{q}_n = \frac{\boldsymbol{a}^{(n)}(k_0)}{n!}, \quad n = 0, 1, 2, \cdots, \tag{3.43}$$

with $\boldsymbol{a}^{(0)}(k_0) = \boldsymbol{a}(k_0)$. Combining (3.40), (3.42), and (3.43), a recursive relationship to compute the nth moment, $\boldsymbol{q}_n$, can be written as

$$\boldsymbol{q}_n = \overline{\boldsymbol{Z}}^{-1}(k_0) \left[\frac{\boldsymbol{v}^{(n)}(k_0)}{n!} - \sum_{m=0}^{n} \frac{(1-\delta_m)\overline{\boldsymbol{Z}}^{(m)}(k_0)\boldsymbol{q}_{n-m}}{m!} \right], \tag{3.44}$$

where the Kronecker delta is defined as

$$\delta_m = \begin{cases} 1, & m = 0 \\ 0, & m \neq 0. \end{cases} \tag{3.45}$$

In (3.44), $\overline{\boldsymbol{Z}}^{(n)}(k_0)$ and $\boldsymbol{v}^{(n)}(k_0)$ are the nth derivatives of $\overline{\boldsymbol{Z}}$ and $\boldsymbol{v}$ at the frequency f_0. To find the solution at any desired frequency (i.e., in an acceptable range), we can use the moments defined in (3.44) to find the solution vector [9].

3.6.2.2 Padé Approximation

In most cases, the Taylor series expansion provides good results. However, the accuracy of this expansion is limited to a radius of convergence. To improve the accuracy of the solution, we replace the Taylor series expansion with the Padé approximation by matching it with a rational function, which can be shown as

$$\boldsymbol{a}(k) = \sum_{n=0}^{L+M} \boldsymbol{q}_n (k-k_0)^n = \frac{\sum_{i=0}^{L} \boldsymbol{x}_i (k-k_0)^i}{\sum_{j=0}^{M} b_j (k-k_0)^j + 1}. \tag{3.46}$$

The details of computing the coefficients can be found in [10]. The rational function has a smaller error in comparison to polynomial approximation [10].

3.6.2.3 Model-Based Parameter Estimation

Alternatively, matching the derivatives of $\boldsymbol{a}(k)$ at the frequency f_0 results in the coefficients of the rational function at (3.46). A detailed description of MBPE

can be found in [11]. It can be shown that both the Padé approximation and MBPE are the same. The Padé approximation can be obtained using the Taylor series expansion and MBPE can be achieved by matching the the derivatives of the function [12].

3.6.2.4 Cubic-Spline Interpolation

The cubic-spline interpolation is a numerical approximation method. It can be used to approximate the MRCS curve on a set of control points. Unlike AWE, derivatives of the impedance matrix are not required. The cubic spline is constructed of third-order polynomials, which cross a set of nonuniform sampling nodes. Commonly, the second derivative is set to zero at the endpoints. In this way, we can generate a "natural" cubic spline. The main idea of this method is to interpolate the solution matrix [5], which is obtained from the solution of the matrix equation for selected excitations. One main drawback of this method is its sampling of the RHS matrix, which makes the method error uncontrollable.

Chapter 4

Numerical Results

We performed three numerical experiments to exhibit the efficiency and the accuracy of the proposed method. All simulations are performed on a server with two quad-core Intel Xeon X5355 processors with a clock rate of 2.66 GHz and 16 GB memory. The threshold for a bi-conjugate gradient stabilized (BiCGSTAB) solver [37] is set to 10^{-4}. The required MVMs in each iteration of BiCGSTAB iterative solver are accelerated by MLFMA. Brute-force (BF) multiple 2-D MRCS values are obtained from the solution of the matrix equation in (3.22), which is accelerated by MLFMA. The threshold for the ACA and truncated SVD algorithms is set to 10^{-4}.

4.0.3 NASA Almond

The first experiment is performed on the NASA almond geometry [38] at a frequency of 1 GHz. The corresponding wavelength is 30 cm. The NASA almond has a length of 1.7 m and hence the electrical size of the largest dimension of the geometry is 5.6λ, as evident in Fig. 4.1. The surface of the NASA almond geometry is discretized with planar triangles of an average mesh size of 0.1λ, which leads to $8,673$ unknowns. Based on (2.41), the maximum angular resolution must

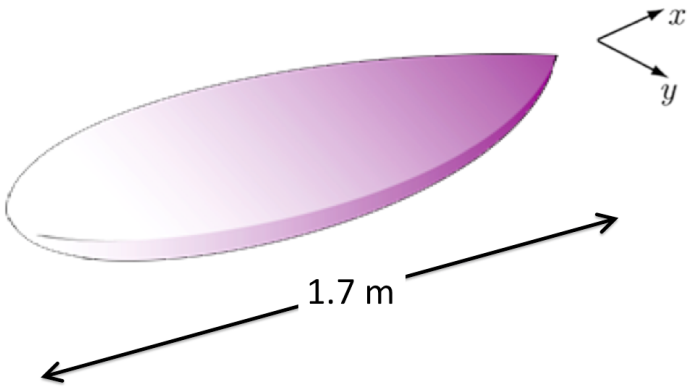

Figure 4.1: NASA almond geometry. The electrical size of the largest dimension of the geometry is 5.6λ at 1 GHz.

be

$$\Delta\theta_{\max} = \Delta\phi_{\max} = \frac{\lambda}{2d} = \frac{0.3}{2 \times 1.7} = 0.08 \text{ rad} = 5.05^{\circ} \tag{4.1}$$

to obtain a full representation of back-scattered fields. In this example, we illuminate the NASA almond with ϕ-polarized plane waves incident from a sector defined by $[\theta_{\min}, \theta_{\max}] = [30^{\circ}, 60^{\circ}]$ and $[\phi_{\min}, \phi_{\max}] = [45^{\circ}, 75^{\circ}]$. The angular resolution for the computation of a 2-D spatial response is set to 1°. As a consequence, obtaining the BF results require 961 runs.

Figure 4.2(a) presents the 2-D spatial MRCS values correspond to the BF runs and the results obtained from RACA are illustrated in Fig. 4.2(b). The third plot in Fig. 4.2 presents the percentage of the relative error between the proposed method and the BF runs. The relative error is calculated via

$$\text{Relative Error} = 100 \times \frac{|\sigma_{\text{BF}} - \sigma_{\text{RACA}}|}{|\sigma_{\text{BF}}|_{\max}}, \tag{4.2}$$

where σ_{BF} denotes the MRCS values obtained from direct calculation and σ_{RACA} implies the MRCS values computed via RACA. Based on (4.1), a 1° resolution of spherical incident angles causes linear dependencies in the RHS matrix. A major portion of the existing linear dependencies between the columns of $\overline{\boldsymbol{V}}$ is removed by ACA in the first step. Table 4.1 reports the statistics of this case, where one can see that ACA reduces the required solutions from 961 to 51.

As per the earlier discussion, the low-rank approximation obtained from ACA also contains linear dependencies. The proposed recompression technique omits

the dependencies in the matrices obtained from ordinary ACA and results in a better low-rank approximation, of rank 40, which is less than 51. In other words, the truncated SVD of the RHS matrix is obtained with a low computational cost.

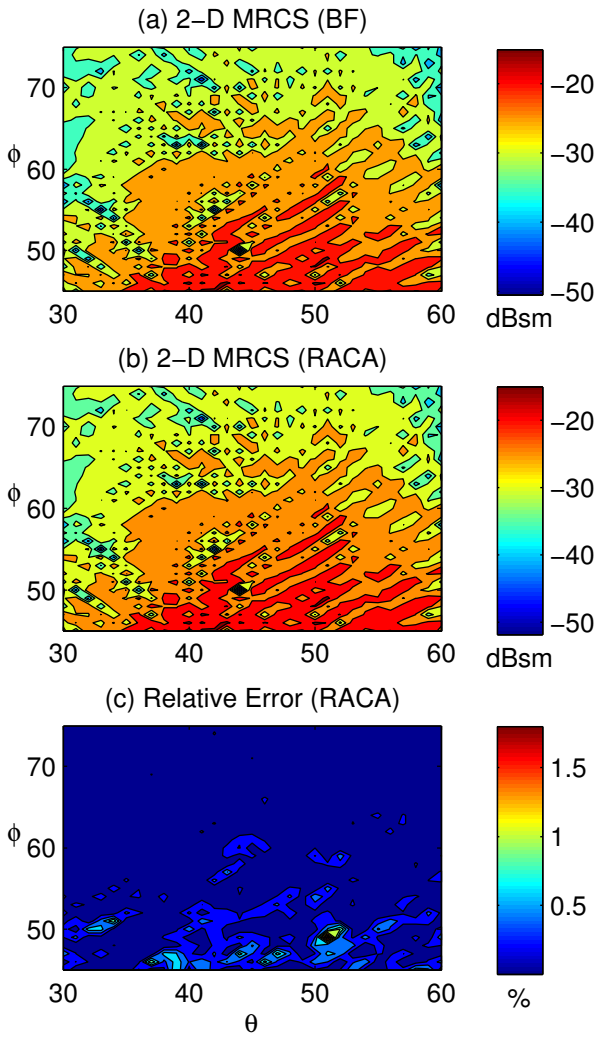

Figure 4.2: $\phi\phi$-polarized 2-D MRCS of the NASA almond. 2-D MRCS (a) obtained from the BF runs and (b) computed using the RACA. (c) The percentage of the relative error.

To analyze the efficiency of RACA, we use different interpolation techniques to obtain 1° resolution from the BF runs performed with various angular intervals. The first set of interpolations are performed using the samples with a resolution

of 2°. The required number of BF runs in this case is $16 \times 16 = 256$. The result of the CS interpolation is provided in Fig. 4.3(a) and the percentage of the relative error is illustrated in Fig. 4.3(b). We present the 2-D MRCS values obtained from linear and nearest-neighbor interpolation techniques in Fig. 4.3(c) and Fig. 4.3(e), respectively. The percentage of the relative error of these two interpolation methods are, respectively, provided in Fig. 4.3(d) and Fig. 4.3(f). Despite the large number of required simulations in the interpolation method, the error rate in the interpolation method is much higher than the error rate of the RACA.

Table 4.1: Statistics of the NASA Almond Simulations

Frequency	1 GHz
Number of Unknowns	8673
Number of Incident Plane Waves	961
ACA Threshold	10^{-4}
SVD Threshold	10^{-4}
k in ACA	51
Compression (%)	94.6
k' of RACA	40
Compression (%)	95.6
Time for BF Run (s)	$5,679.6$
Time for RACA Run (s)	376.2
Speed-up Factor	15.1

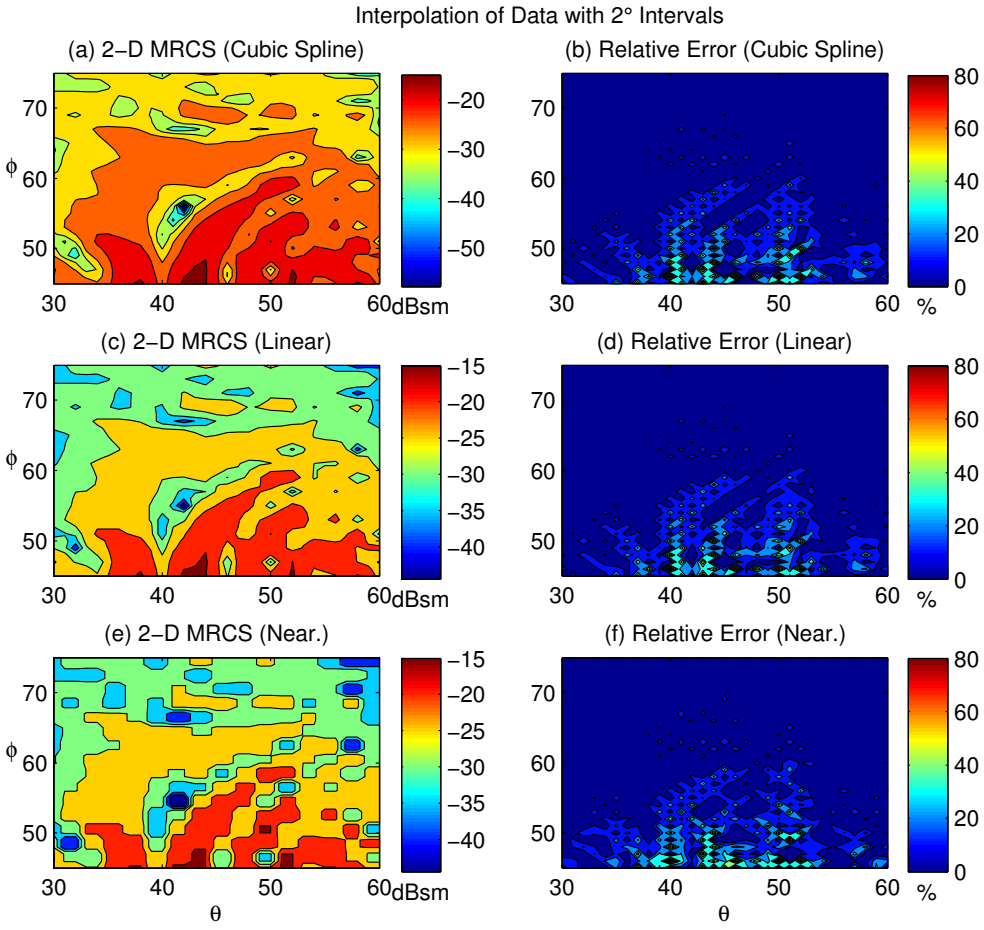

Figure 4.3: $\phi\phi$-polarized 2-D MRCS values of the NASA almond obtained using different interpolation methods. The plots have a resolution of 1° and the sampling interval is 2°. (a) 2-D MRCS values obtained from the CS interpolation method. (b) The relative error in percents for the CS interpolation method. (c) 2-D MRCS values resulting from linear interpolation method. (d) The relative error in percents corresponding to the linear interpolation method. (e) 2-D MRCS values computed using the nearest-neighbor interpolation technique. (f) The relative error in percents for the nearest-neighbor interpolation method.

Another set of simulations are performed using the samples with 3° angular resolution and the results provided in Fig. 4.4. Since we use less samples compared to the data with 2° resolution, the interpolation lead to higher error rate. The number of required BF runs for this case is 121.

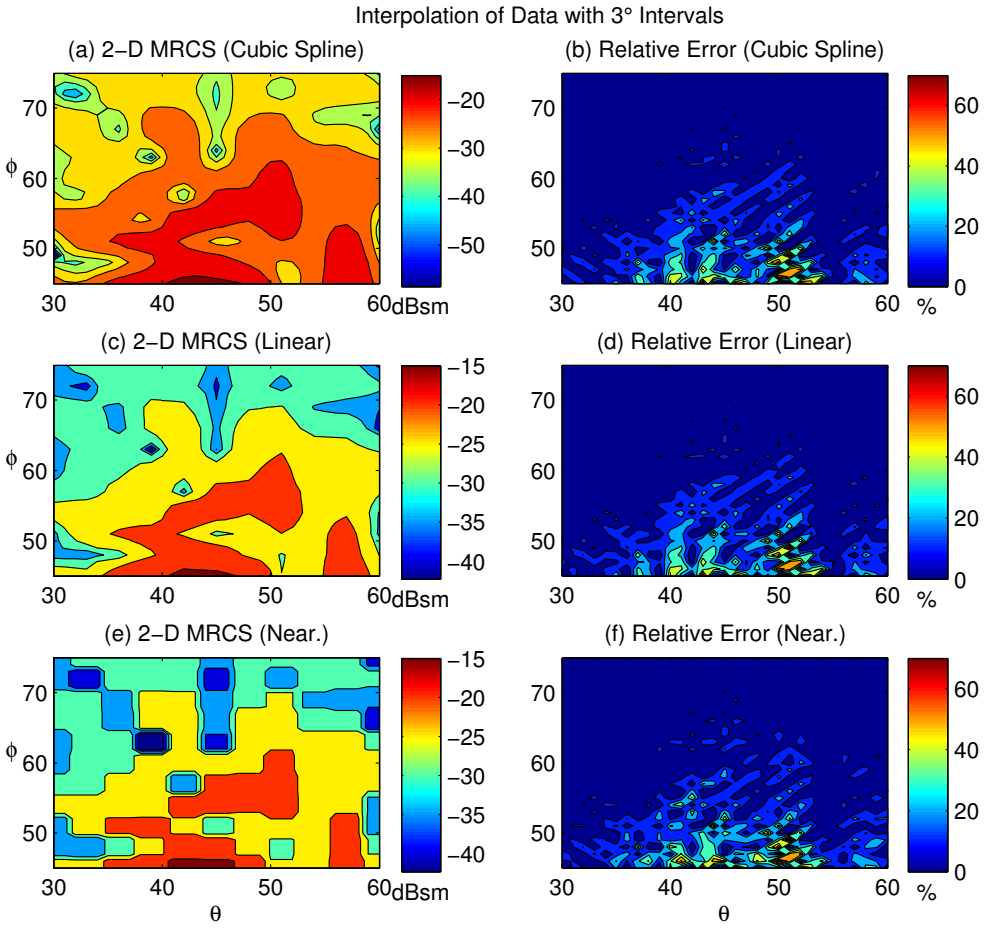

Figure 4.4: $\phi\phi$-polarized 2-D MRCS values of the NASA almond obtained using different interpolation methods. The plots have a resolution of 1° and the sampling interval is 3°. (a) 2-D MRCS values obtained from the CS interpolation method. (b) The relative error in percents for the CS interpolation method. (c) 2-D MRCS values resulting from linear interpolation method. (d) The relative error in percents corresponding to the linear interpolation method. (e) 2-D MRCS values computed using the nearest-neighbor interpolation technique. (f) The relative error in percents for the nearest-neighbor interpolation method.

We also carry out a set of interpolation simulations with the sampling interval of 5°. As evident in Fig. 4.5, the 2-D MRCS pattern do not agree with the pattern obtained from BF runs (Fig. 4.2). The number of required solutions for this interpolation is 25 that is comparable with that of the RACA (i.e., 40). However,

the error rate in the interpolation method is not acceptable that testifies the accuracy and the efficiency of RACA.

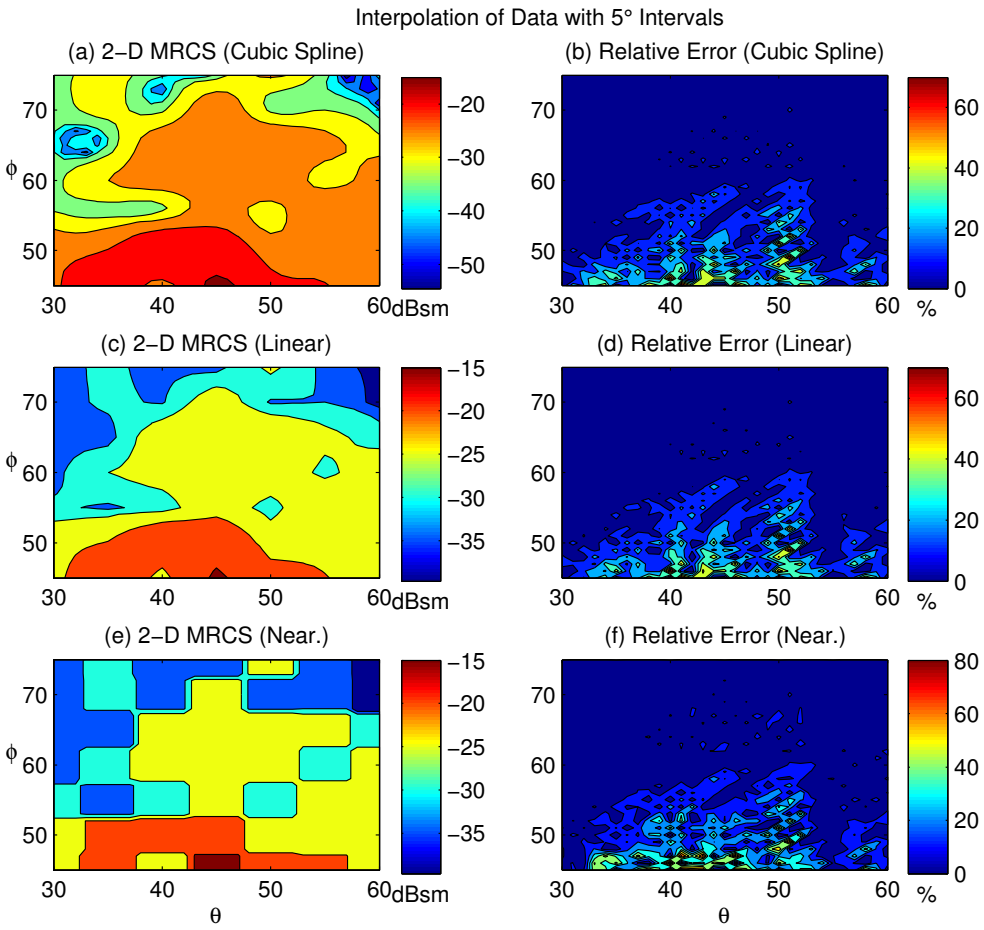

Figure 4.5: $\phi\phi$-polarized 2-D MRCS values of the NASA almond obtained using different interpolation methods. The plots have a resolution of 1° and the sampling interval is 5°. 2-D MRCS values obtained from (a) the CS interpolation method, (c) the linear interpolation method, (e) the nearest-neighbor interpolation technique and corresponding relative percentage of errors (b), (d) and (f), respectively.

All the mentioned interpolations are performed with an angular resolution less than the maximum angular resolution provided in (4.1). Figure 4.6 illustrates the results of three different interpolations with a 10° sampling resolution. As

we would expect, the results do not agree with the 2-D MRCS values in Fig. 4.2, since the Nyquist rate is not satisfied for sampling.

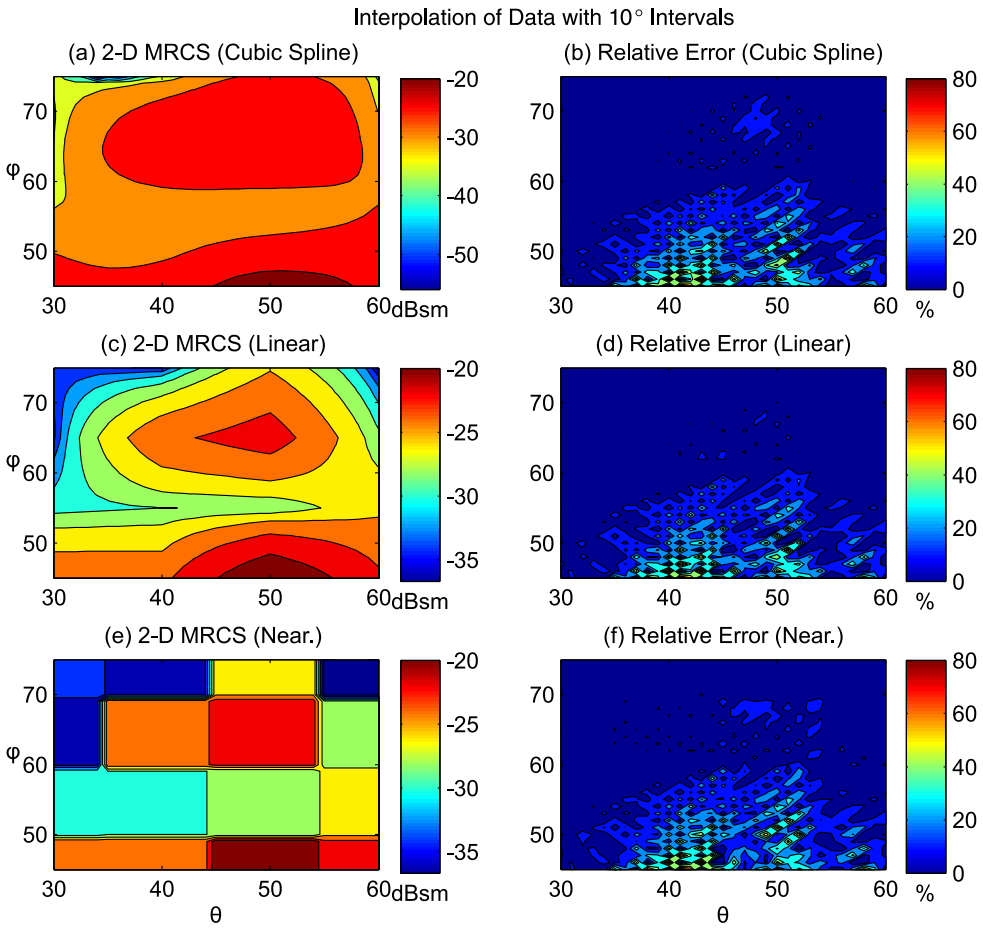

Figure 4.6: $\phi\phi$-polarized 2-D MRCS values of the NASA almond obtained using different interpolation methods. The plots have a resolution of 1° and the sampling interval is 10°. 2-D MRCS values obtained from (a) the CS interpolation method and (b) corresponding relative percentage of error, (c) the linear interpolation method and (d) corresponding relative percentage of error, (e) the nearest-neighbor interpolation technique and (f) corresponding relative percentage of error.

4.1 Flamme

For the second numerical experiment, we investigated the computation of the multiple 2-D MRCSs the Flamme geometry at a frequency of 4 GHz, where the corresponding wavelength is 7.5 cm. The Flamme was designed and built by l'Office National d'Etudes et de Recherche Aerospatiales for the 1993 Paris Air Show exhibition, for an unclassified demonstration of a possible future airborne stealth technology [4]. As shown in Fig. 4.7, the Flamme has a maximum length of 0.6 m, and hence the electrical size of the largest dimension of the geometry is 8λ. The surface of the geometry is discretized with planar triangles with an average mesh size of 0.1λ, which leads to 13,386 unknowns.

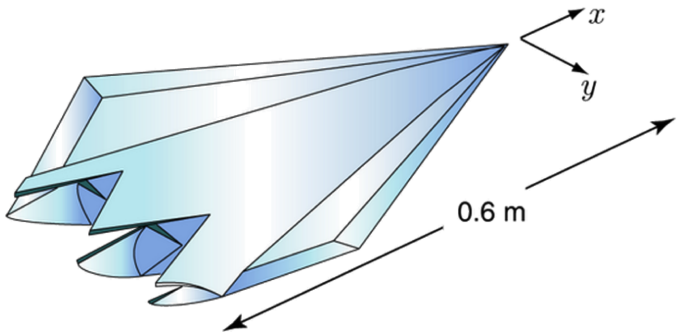

Figure 4.7: The Flamme geometry. The electrical size of the largest dimension of the geometry is 8λ at 4 GHz.

The maximum angular resolution that satisfies the Nyquist rate is

$$\Delta\theta_{\max} = \Delta\phi_{\max} = \frac{\lambda}{2d} = \frac{0.075}{2 \times 0.6} = 0.06 \text{ rad} = 3.5^{\circ}. \tag{4.3}$$

The Flamme geometry is illuminated with θ-polarized plane waves incident from a sector defined by $[\theta_{\min}, \theta_{\max}] = [30^{\circ}, 50^{\circ}]$ and $[\phi_{\min}, \phi_{\max}] = [45^{\circ}, 65^{\circ}]$, with an angular resolution of 0.5°. The BF simulation requires 1681 runs. Figure 4.8 illustrates the multiple 2-D MRCS values. The first plot in the figure shows the BF results, and the results obtained from RACA are shown in the second plot (Fig. 4.8(b)). The percentage of relative error between the proposed method and the BF runs is presented in Fig. 4.8(c). The relative error is calculated using the formula in (4.2).

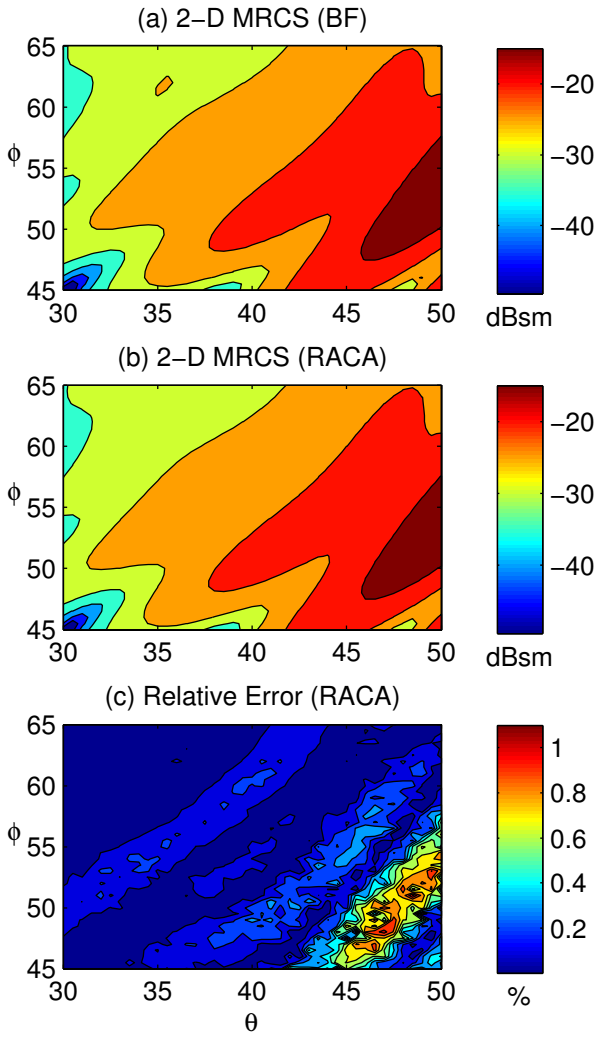

Figure 4.8: $\theta\theta$-polarized 2-D MRCS values of the Flamme. (a) 2-D MRCS obtained from the BF runs. (b) 2-D MRCS computed using the RACA. (c) The percentage of the relative error.

The selected resolution of the spherical incident angles, i.e., $\Delta\theta = \Delta\phi = 0.5°$, is less than the resolution provided in (4.3). As a consequence, the RHS matrix will have linear dependencies. Using the same procedure as in the first example, we utilize the ACA algorithm to reduce the required solutions from 1681 to 43; the number of solutions can be further reduced by incorporating the proposed recompression method to 32. In other words, we have a compression rate of

almost 98%. Table 4.2 summarizes the statistics for this example. As reported in the table, the simulation time is reduced from almost 10.5 hours to about 18 minutes.

Table 4.2: Statistics of the Flamme Simulations

Frequency	4 GHz
Number of Unknowns	13386
Number of Incident Plane Waves	1681
ACA Threshold	10^{-4}
SVD Threshold	10^{-4}
k in ACA	43
Compression (%)	97.4
k' of RACA	32
Compression (%)	98.1
Time for BF Run (s)	$38,489.3$
Time for RACA Run (s)	$1,058.7$
Speed-up Factor	36.3

Similar to the NASA almond case, to analyze the efficiency of RACA, we use different interpolation methods to obtain 0.5° resolution from the BF runs performed with different angular resolutions. The first set of interpolations are performed using the samples with a resolution of 1°. Hence, the required number of BF runs in this case is $21 \times 21 = 441$. The results of the CS interpolation are provided in Fig. 4.9(a) and the percentage of the relative error is illustrated in Fig. 4.9(b). We present the 2-D MRCS values achieved from linear and nearest-neighbor interpolation techniques in Fig. 4.9(c) and Fig. 4.9(e), respectively. Also, the percentage of the relative error of these two interpolation methods are provided in Fig. 4.9(d) and Fig. 4.9(f). Despite the fact that employing interpolation requires more number of solutions, the error rate in the interpolation method is much higher than the error rate of the RACA.

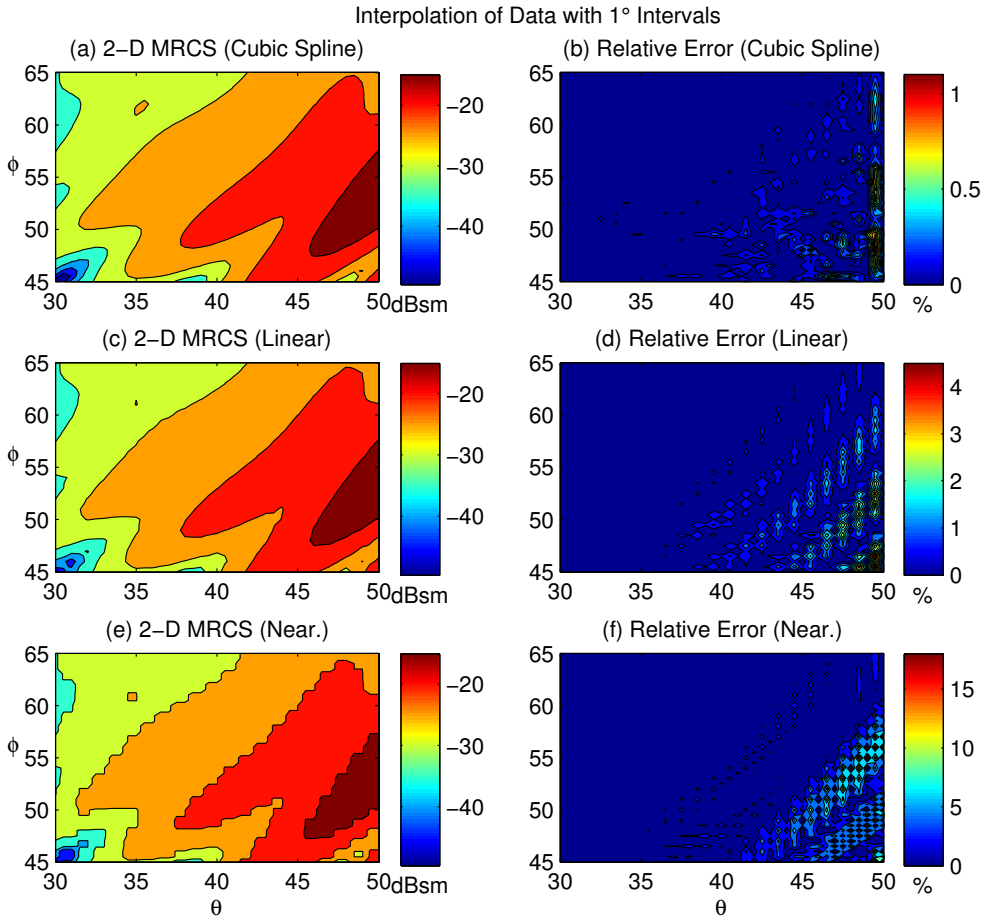

Figure 4.9: $\theta\theta$-polarized 2-D MRCS values of the Flamme obtained using different interpolation techniques. The plots have a resolution of 0.5° and the sampling interval is 1°. (a) 2-D MRCS values obtained from the CS interpolation method. (b) The relative error in percents for the CS interpolation method. (c) 2-D MRCS values resulting from linear interpolation method. (d) The relative error in percents corresponding to the linear interpolation method. (e) 2-D MRCS values computed using the nearest-neighbor interpolation technique. (f) The relative error in percents for the nearest-neighbor interpolation method.

Another set of interpolations is performed using the samples with 2° resolution and the results provided in Fig. 4.10. Since we use less samples compared to the data with 1° resolution, the interpolation lead to higher error rate. The number of required BF runs for this case is 121.

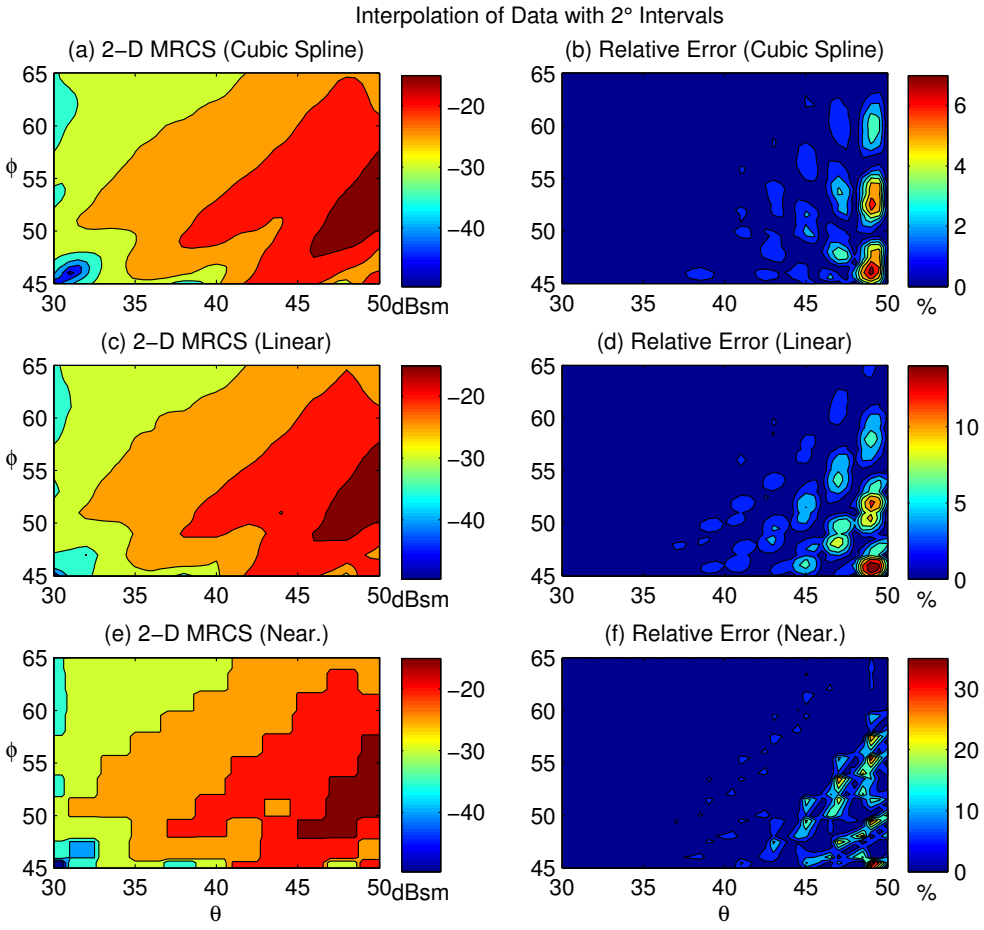

Figure 4.10: $\theta\theta$-polarized 2-D MRCS values of the Flamme obtained using different interpolation techniques. The plots have a resolution of 0.5° and the sampling interval is 2°. (a) 2-D MRCS values obtained from the CS interpolation method. (b) The relative error in percents for the CS interpolation method. (c) 2-D MRCS values resulting from linear interpolation method. (d) The relative error in percents corresponding to the linear interpolation method. (e) 2-D MRCS values computed using the nearest-neighbor interpolation technique. (f) The relative error in percents for the nearest-neighbor interpolation method.

We also carry out a set of interpolation simulations with the sampling interval of 2.5°. The results of these simulations are provided in Fig. 4.11. The number of required solutions for this interpolation is 81 that is almost two times more than that of the RACA. However, the error rate in the interpolation method is

not acceptable that testifies the accuracy and efficiency of RACA.

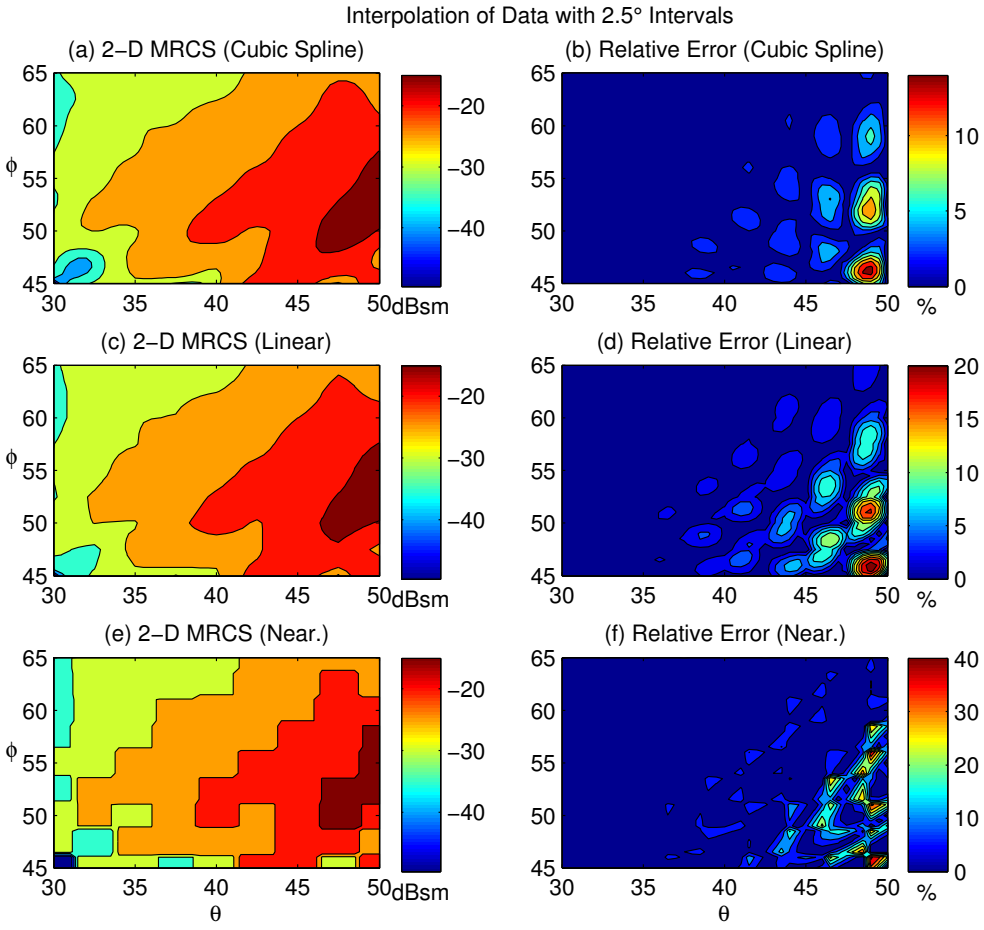

Figure 4.11: $\theta\theta$-polarized 2-D MRCS values of the Flamme obtained using different interpolation techniques. The plots have a resolution of 0.5° and the sampling interval is 2.5°. 2-D MRCS values obtained from (a) the CS interpolation method, (c) the linear interpolation method, (e) the nearest-neighbor interpolation technique and corresponding relative percentage of errors (b), (d) and (f), respectively.

All the mentioned interpolations are performed with an angular resolution less than the Nyquist rate provided in (4.3). Figure 4.6 illustrates the results of three different interpolations with a 5° sampling resolution. As shown in Fig. 4.12, the 2-D MRCS pattern do not agree with the pattern obtained from BF runs

(Fig. 4.8)

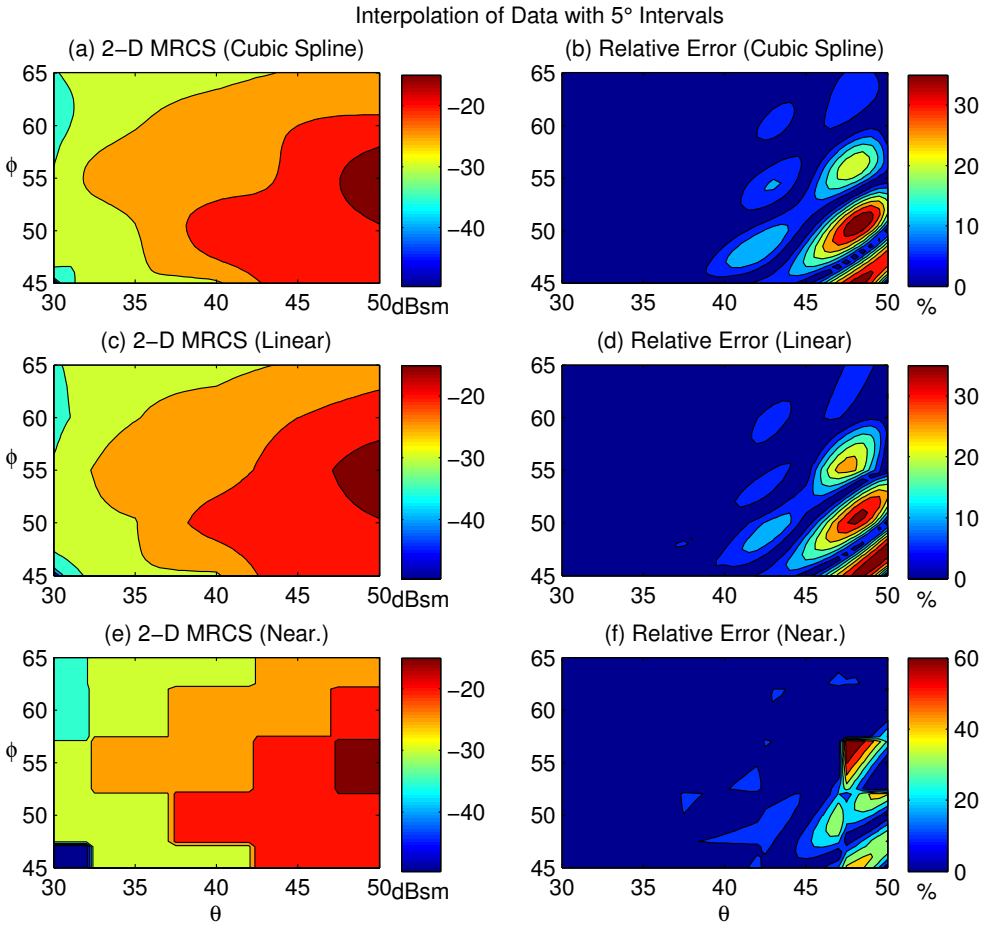

Figure 4.12: $\theta\theta$-polarized 2-D MRCS values of the Flamme obtained using different interpolation techniques. The plots have a resolution of 0.5° and the sampling interval is 5°. 2-D MRCS values obtained from (a) the CS interpolation method and (b) corresponding relative percentage of error, (c) the linear interpolation method and (d) corresponding relative percentage of error, (e) the nearest-neighbor interpolation technique and (f) corresponding relative percentage of error.

4.2 Generic Helicopter

The third experiment is performed on a generic helicopter geometry, shown in Fig. 4.13, at a frequency of 125 MHz.

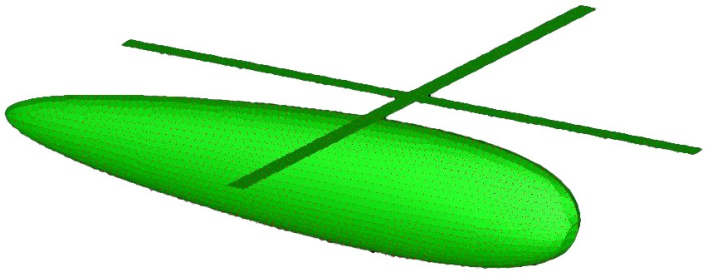

Figure 4.13: Generic helicopter geometry. The electrical size of the largest dimension of the geometry is 3.4λ at 125 MHz.

The surface of the geometry is discretized with 9,732 unknowns using planar triangles. Figure 4.14 presents a one-dimensional (1-D) MRCS values and the corresponding relative error. We illuminate the helicopter with ϕ-polarized plane waves incident from an angular sector defined by $[\phi_{\min}, \phi_{\max}] = [0°, 180°]$ on the $\theta = 90°$ plane and with an angular resolution of 1°. The detailed statistics are presented in Table 4.3, where it shows that employing RACA reduces the number of required solutions from 181 to 11.

Table 4.3: Statistics of the Helicopter Simulations

Number of Incident Plane Waves	181
k in ACA	18
Compression (%)	90.1
k' of RACA	11
Compression (%)	93.9
Time for BF Run (s)	$67,946.1$
Time for RACA Run (s)	$4,213.9$
Speed-up Factor	16.1

Figure 4.14 illustrates the $\phi\phi$-polarized MRCS values of the generic helicopter. As evident in Fig. 4.14, the patterns of the results obtained from BF runs and RACA agree well. The green curve in Fig. 4.14 illustrates the percentage of the relative error calculated via the formula in (4.2).

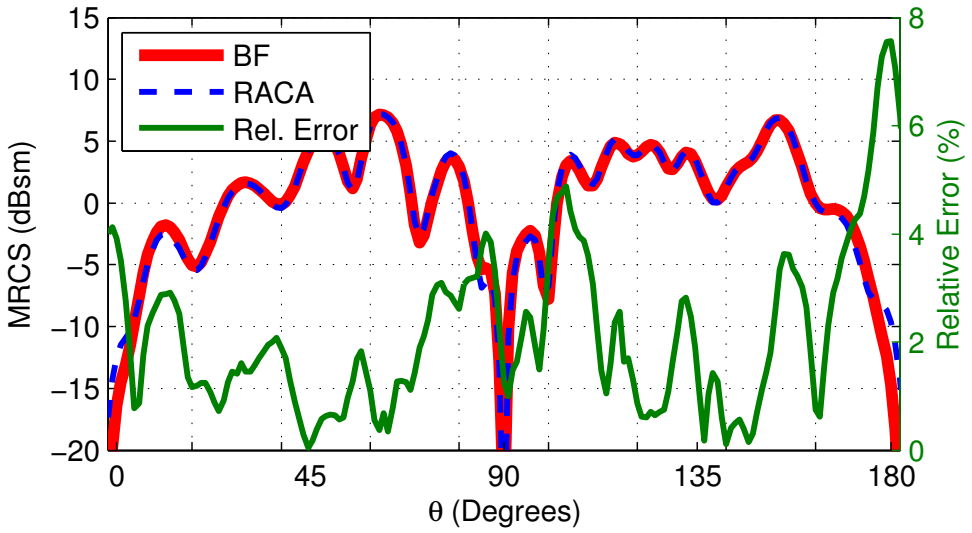

Figure 4.14: $\phi\phi$-polarized MRCS values of the generic helicopter in the angular sector defined $0 < \phi < \pi$ and $\theta = \pi/2$ with an angular resolution of 1°. The solid red curve indicates the MRCS values that are computed via BF runs. The dashed blue curve represents the MRCS values obtained from RACA. The green curve illustrates the percentage of the relative error.

To validate the efficiency and accuracy of RACA in the computation of 1-D MRCS values, we employed five different 1-D interpolation techniques including the cubic-spline, linear, spline, nearest-neighbor, and piecewise cubic hermite interpolations. Figure 4.15 illustrates the MRCS values achieved by the cubic-spline interpolation technique. The results of linear interpolation and the spline interpolation are provided in Fig. 4.16 and Fig. 4.17, respectively. Furthermore, we present the MRCS values obtained from nearest-neighbor and piecewise cubic hermite interpolation techniques in Fig. 4.18 and Fig. 4.19.

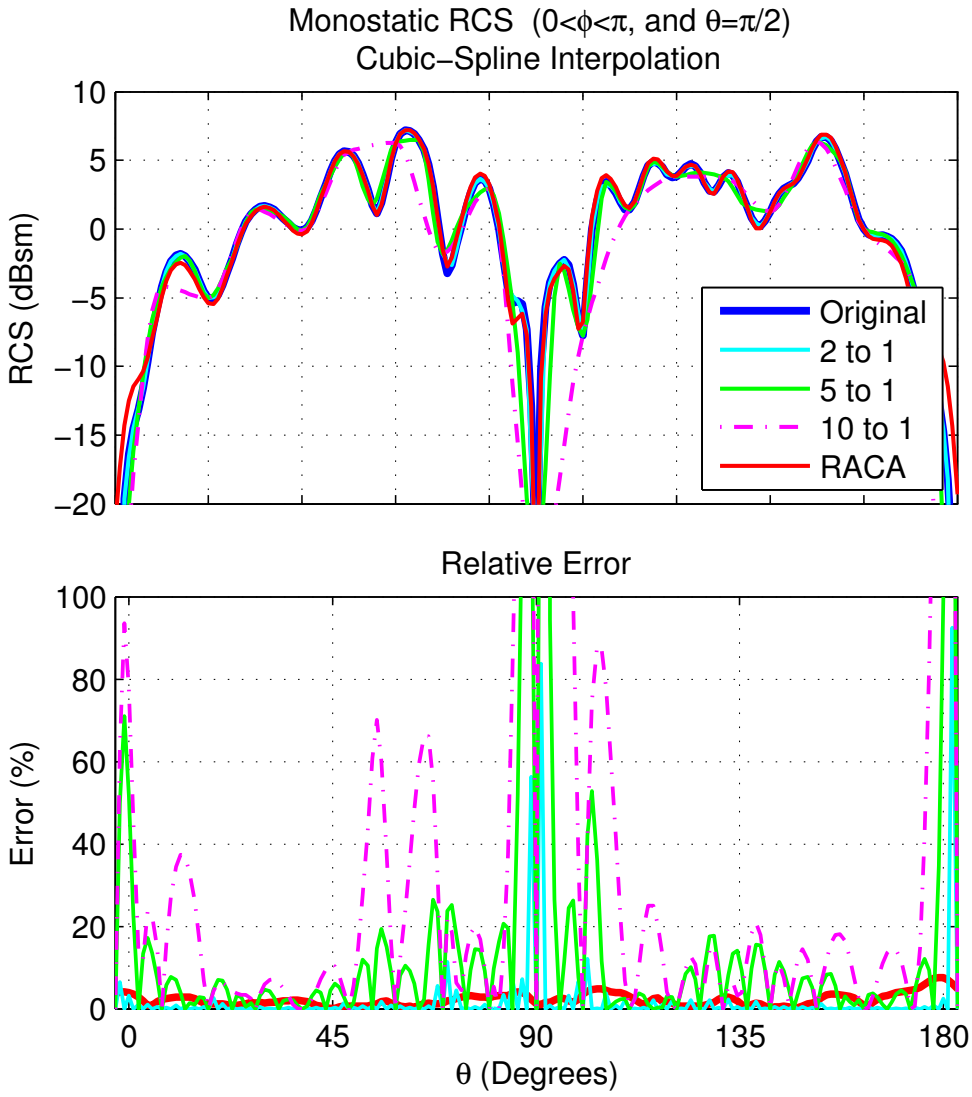

Figure 4.15: $\phi\phi$-polarized MRCS values of the generic helicopter in the angular sector defined by $0 < \phi < \pi$ and $\theta = \pi/2$ with an angular resolution of 1°. The solid blue curve represents the MRCS values obtained from BF runs. The solid red curve indicates the MRCS values that are computed via RACA. The cyan, green, and magenta curves illustrate the MRCS values achieved by the cubic-spline interpolation technique and correspond to 2°, 5°, and 10° sampling intervals, respectively. The relative errors are provided in the second plot with the same legend.

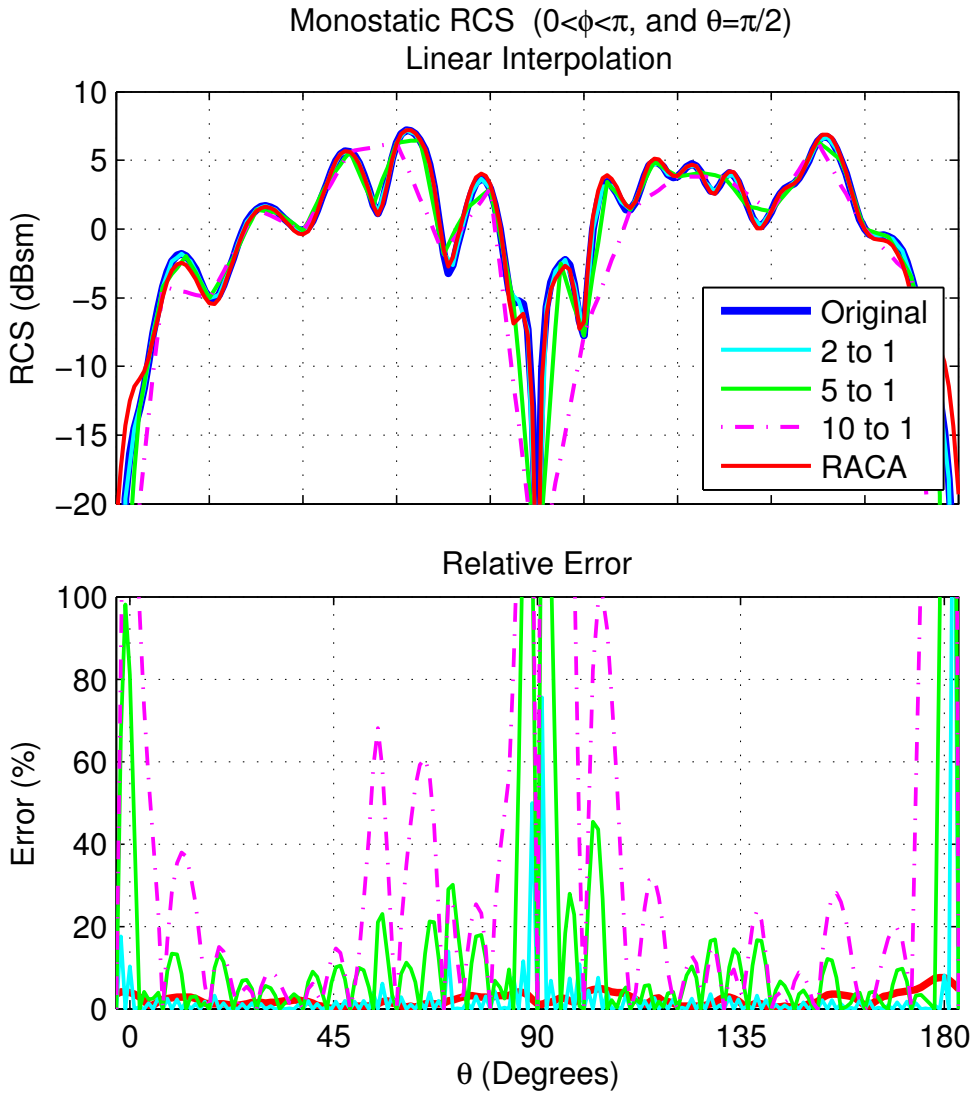

Figure 4.16: $\phi\phi$-polarized MRCS values of the generic helicopter in the angular sector defined by $0 < \phi < \pi$ and $\theta = \pi/2$ with an angular resolution of 1°. The solid blue curve represents the MRCS values obtained from BF runs. The solid red curve indicates the MRCS values that are computed via RACA. The cyan, green, and magenta curves illustrate the MRCS values achieved by the linear interpolation technique and correspond to 2°, 5°, and 10° sampling intervals, respectively. The relative errors are provided in the second plot with the same legend.

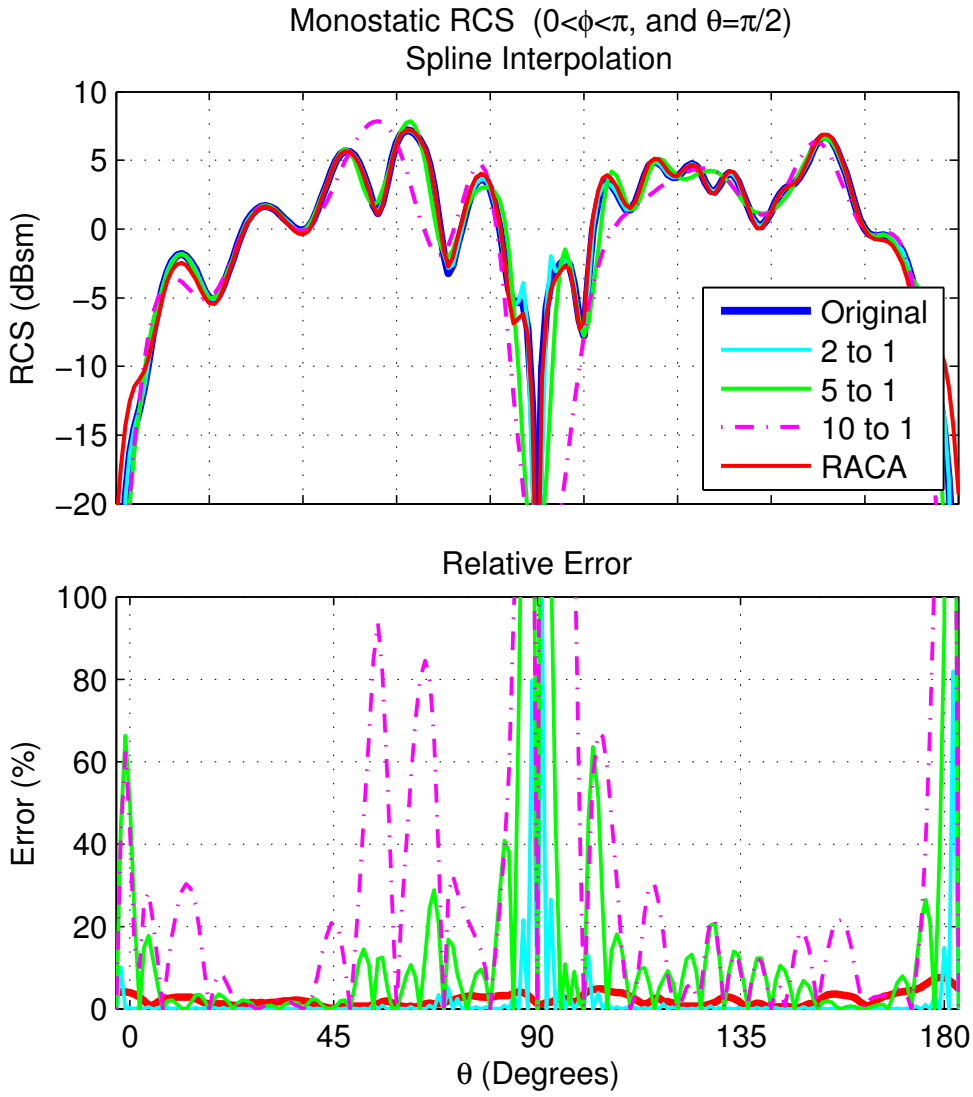

Figure 4.17: $\phi\phi$-polarized MRCS values of the generic helicopter in the angular sector defined by $0 < \phi < \pi$ and $\theta = \pi/2$ with an angular resolution of 1°. The solid blue curve represents the MRCS values obtained from BF runs. The solid red curve indicates the MRCS values that are computed via RACA. The cyan, green, and magenta curves illustrate the MRCS values achieved by the spline interpolation technique and correspond to 2°, 5°, and 10° sampling intervals, respectively. The relative errors are provided in the second plot with the same legend.

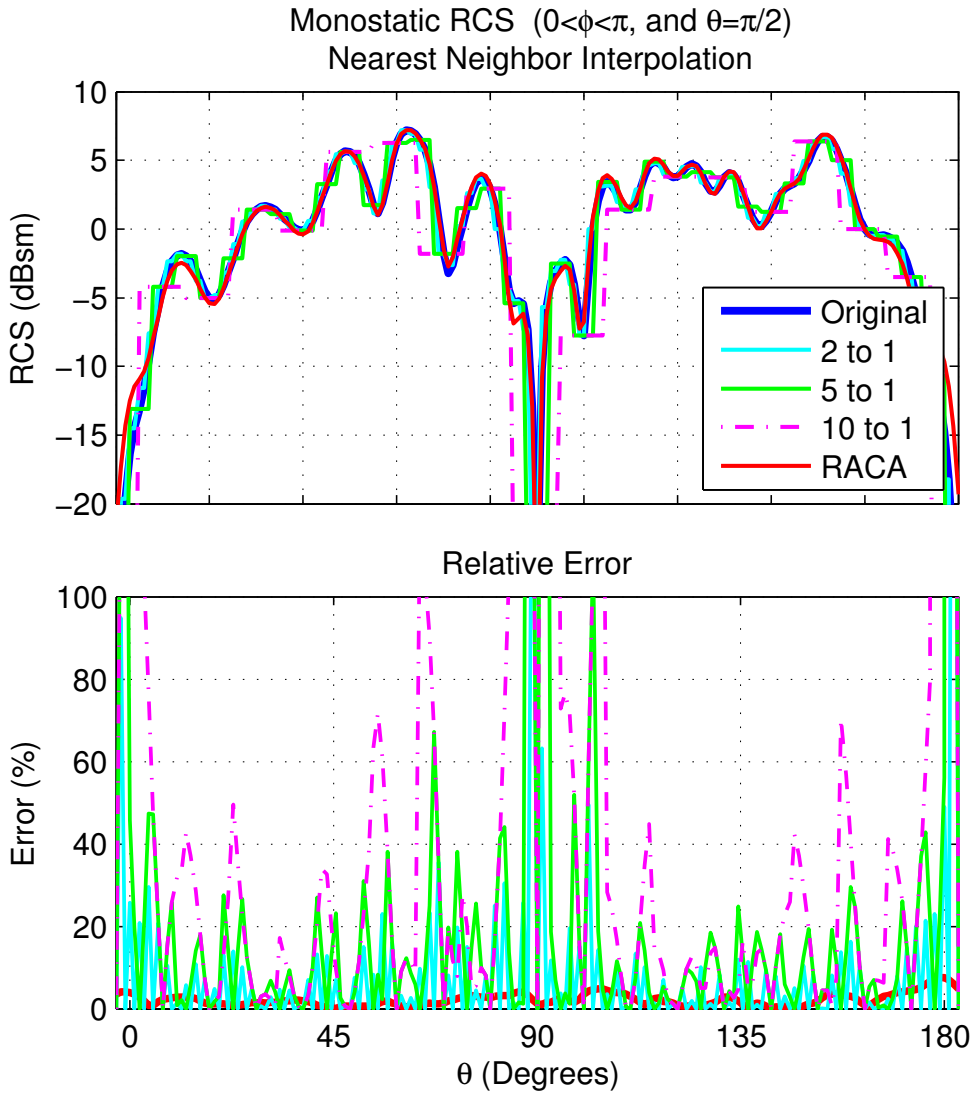

Figure 4.18: $\phi\phi$-polarized MRCS values of the generic helicopter in the angular sector defined by $0 < \phi < \pi$ and $\theta = \pi/2$ with an angular resolution of 1°. The solid blue curve represents the MRCS values obtained from BF runs. The solid red curve indicates the MRCS values that are computed via RACA. The cyan, green, and magenta curves illustrate the MRCS values achieved by the nearest neighbor interpolation technique and correspond to 2°, 5°, and 10° sampling intervals, respectively. The relative errors are provided in the second plot with the same legend.

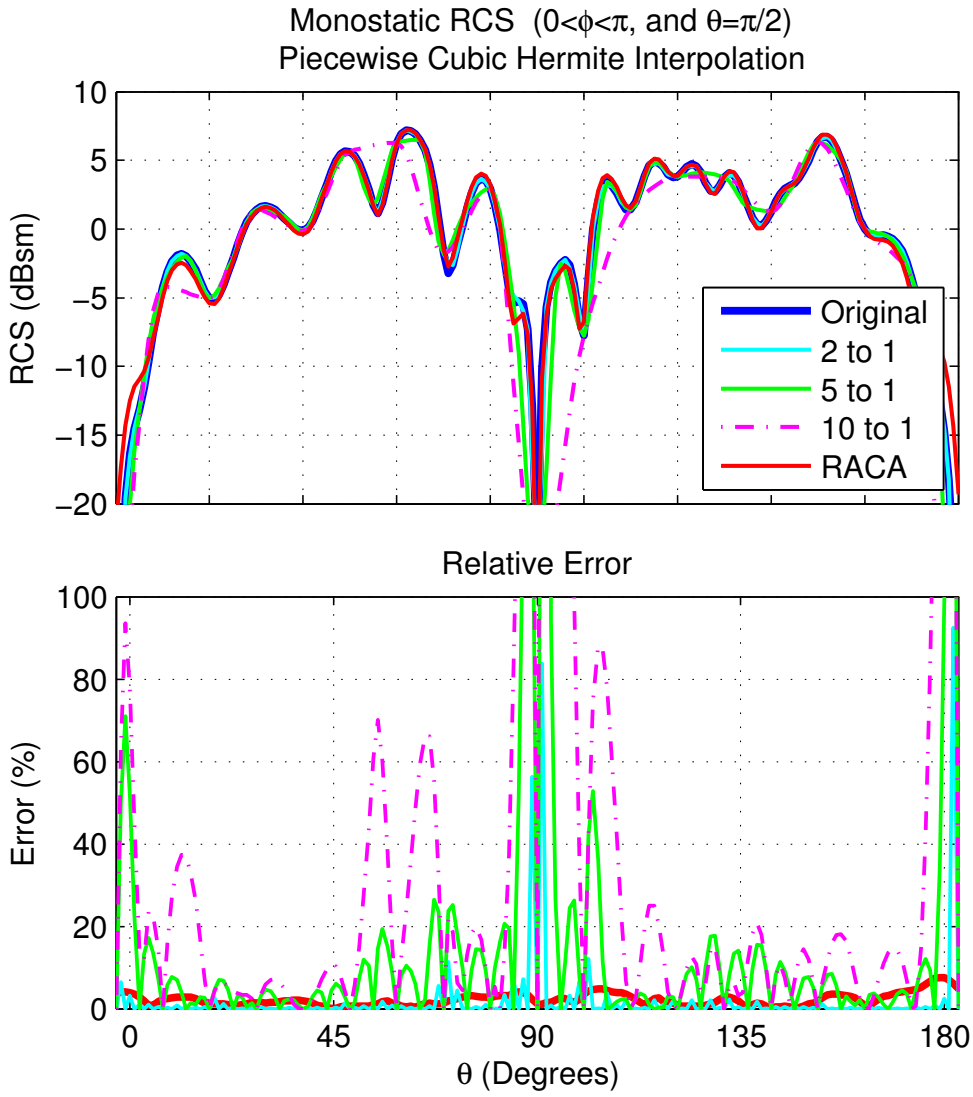

Figure 4.19: $\phi\phi$-polarized MRCS values of the generic helicopter in the angular sector defined by $0 < \phi < \pi$ and $\theta = \pi/2$ with an angular resolution of 1°. The solid blue curve represents the MRCS values obtained from BF runs. The solid red curve indicates the MRCS values that are computed via RACA. The cyan, green, and magenta curves illustrate the MRCS values achieved by the piecewise cubic hermite interpolation technique and correspond to 2°, 5°, and 10° sampling intervals, respectively. The relative errors are provided in the second plot with the same legend.

Chapter 5

Conclusion

We propose an efficient and low-complexity algorithm based on RACA to accelerate the computation of MRCS. The results show that RACA is more accurate and more efficient than interpolation techniques. RACA also outperforms the ordinary ACA. The method incorporates the QR decomposition and computes SVD in an efficient and fast manner. Due to the purely algebraic nature of the proposed method, it can be combined with any fast solver. As a future work, a parallel version of this technique could be developed that enables the solutions of very large-scale electromagnetic scattering problems.

Bibliography

[1] E. F. Knott, J. F. Shaeffer, and M. T. Tuley, *Radar Cross Section*, Norwood, MA: Artech House, Inc, 1993.

[2] M. Skolnik, *Radar Handbook*, NewYork: McGraw-Hill, 1990.

[3] T. A. Winchester, "Pulsed radar return from a chaff cloud," *IEE Proc., Part F*, vol. 139, no. 4, pp. 315–320, 1992.

[4] L. Gürel, H. Bağcı, J. C. Castelli, A. Cheraly, and F. Tardivel, "Validation through comparison: Measurement and calculation of the bistatic radar cross section of a stealth target," *Radio Sci.*, vol. 38, no. 3, pp. 1046–1058, 2003.

[5] Z. W. Liu, R. S. Chen, and J. Q. Chen, "Adaptive sampling cubic-spline interpolation method for efficient calculation of monostatic RCS," *Microw. Opt. Technol. Lett.*, vol. 50, no. 3, pp. 751–755, 2008.

[6] N. Venkatarayalu, Y.-B. Gan, K. Zhao, and J.-F. Lee, "Fast monostatic RCS computation in FEM based solvers using QR decomposition," in *Proc. 1st IEEE Eur. Conf. Antennas Propag.*, Nov. 2006, pp. 1-5.

[7] Z. Peng, M. Stephanson, and J.-F. Lee, "Fast computation of angular responses of large-scale three-dimensional electromagnetic wave scattering," *IEEE Trans. Antennas Propag.*, vol. 58, no. 9, pp. 3004–3012, 2010.

[8] X.-M. Pan and X.-Q. Sheng, "Fast Computation of Two-Dimensional Spatial Electromagnetic Scattering from Large-Scale Targets," in *Proc. Computational Electromagnetics Workshop, CEM'13*, Izmir, Turkey, Aug. 2013.

[9] Y. E. Erdemli, J. Gong, C. J. Reddy, and J. L. Volakis, "Fast RCS pattern fill using AWE technique," *IEEE Trans. Antennas Propag.*, vol. 46, no. 11, pp. 1752-1753, Nov. 1998.

[10] X. C. Wei, Y. J. Zhang, and E. P. Li, "The hybridization of fast multiple method with asymptotic waveform evaluation for the fast monstatic RCS computation," *IEEE Trans. Antennas Propag.*, vol. 52, no. 2, pp. 605-607, Feb. 2003.

[11] E. K. Miller, "Model-based parameter estimation in electromagnetic III: Applications to EM integral equations," *IEEE Antennas Propag. Mag.*, vol. 40, no. 3, pp. 49–66, Jun. 1998.

[12] Z. Peng and X.-Q. Sheng, "Application of rational function approximation technique to hybrid FE-BI-MLFMA for 3D scattering," *Int. J. RF Microw. Comput.-Aided Eng.*, vol. 17, no. 6, pp. 521-532, Nov. 2007.

[13] A. Schroder, H.-D. Brüns, and C. Schuster, "A hybrid approach for rapid computation of two-dimensional monostatic radar cross section problems with the multilevel fast multipole algorithm," *IEEE Trans. Antennas Propag.*, vol. 60, no. 12, pp. 6058–6061, Dec. 2012.

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

[15] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., Baltimore, MD: John Hopkins University Press, 1996.

[16] Ö. Ergül and L. Gürel, "A hierarchical partitioning strategy for an efcient parallelization of the multilevel fast multipole algorithm," *IEEE Trans. Antennas Propag.*, vol. 57, no. 6, pp. 1740–1750, Jun. 2009.

[17] J. M. Taboada, M. G. Araujo, J. M. Bertolo, L. Landesa, F. Obelleiro, and J. L. Rodriguez, "MLFMA-FFT parallel algorithm for the solution of large-scale problems in electromagnetics," *Prog. Electromagn. Res.*, vol. 105, pp. 15–30, 2010.

[18] M. Kazempour and L. Gürel, "Solution of low-frequency electromagnetics problems using hierarchical matrices," in *Proc. Computational Electromagnetics Workshop, CEM'13*, pp. 9–10, Izmir, Turkey, Aug. 2013. vol. 105, pp. 15–30, 2010.

[19] K. Zhao , M. Vouvakis, and J.-F. Lee, "The adaptive cross approximation algorithm for accelerated method of moment computations of EMC problems," *IEEE Trans. Electromagn. Compat.*, vol. 47, pp. 763–773, 2005.

[20] N. Morita, N. Kumagai, and J. R. Mautz, *Integral Equation Methods for Electromagnetics*, Boston: Artech House, 1990.

[21] X.-Q. Sheng, J.-M. Jin, J. Song, W. C. Chew, and C.-C. Lu, "Solution of combined-field integral equation using multilevel fast multipole algorithm for scattering by homogeneous bodies," *IEEE Trans. Antennas Propag.*, vol. 46, no. 11, pp. 1718–1726, Nov. 1998.

[22] A. J. Poggio and E. K. Miller, "Integral equation solutions of three-dimensional scattering problems," in *Computer Techniques for Electromagnetics*, R. Mittra, Ed. Oxford: Pergamon Press, 1973, Chap. 4.

[23] J. R. Mautz and R. F. Harrington, "H-field, E-field, and combined field solutions for conducting bodies of revolution," *AEÜ*, vol. 32, no. 4, pp. 157–164, Apr. 1978.

[24] Ö. Ergül and L. Gürel, "Discretization error due to the identity operator in surface integral equations," *Comput. Phys. Comm.*, vol. 180, no. 10, pp. 1746–1752, Oct. 2009.

[25] Ö. Ergül and L. Gürel, "Improving the accuracy of the magnetic field integral equation with the linear-linear basis functions," *Radio Sci.*, vol. 41, RS4004, Jul. 2006.

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

[27] R. E. Hodges and Y. Rahmat-Samii, "The evaluation of MFIE integrals with the use of vector triangle basis functions," *Microw. Opt. Technol. Lett.*, vol. 14, no. 1, pp. 9–14, Jan. 1997.

[28] W. C. Chew, J.-M. Jin, E. Michielssen, and J. Song, *Fast and Efficient Algorithms in Computational Electromagnetics.* Boston: Artech House, 2001.

[29] C. A. Balanis, *Advanced Engineering Electromagnetics.* New York, NY, USA: Wiley, 1989.

[30] H. Buddendick and T. Eibert, "Acceleration of ray-based radar cross section predictions using monostatic-bistatic equivalence," *IEEE Trans. Antennas Propag.*, vol. 58, no. 2, pp. 531–539, Feb. 2010.

[31] T. Vaupel and T. F. Eibert, "Comparison and application of near-field ISAR imaging techniques for far-field radar cross section determination," *IEEE Trans. Antennas Propag.*, vol. 54, no. 1, 2006.

[32] M. Bebendorf and S. Rjasanow, "Adaptive low-rank approximation of collocation matrices," *J. Computing*, vol. 70, no. 1, pp. 1–24, 2003.

[33] M. Bebendorf, *Hierarchical Matrices: A Means to Efficiently Solve Elliptic Boundary Value Problems*, Berlin: Springer-Verlag, 2008.

[34] S. Borm, *Efficient Numerical Methods for Non-Local Operators: $\mathcal{H}^2-$Matrix Compression, Algorithms and Analysis*, Switzerland: European Mathematical Society, 2010.

[35] M. Gu and S. C. Eisenstat, "Efcient algorithms for computing a strong rank-revealing QR factorization," *SIAM J. Sci. Comput.*, vol. 17, no. 4, pp. 848-869, 1996.

[36] E. Liberty, F. Woolfe, P. G. Martinsson, V. Rokhlin, and M. Tygert, "Randomized algorithms for the low-rank approximation of matrices,", *Proc. Natl. Acad. Sci.*, vol. 104, pp. 20167–20172, 2007.

[37] R. Barrett, M. Berry, T. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. van der Vorst, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, Philadelphia: SIAM, 1994.

[38] A. C. Wool, H. T. G. Wan, M. J. Schuh, and M. L. Sanders, "Benchmark radar targets for the validation of computational electromagnetics programs," *IEEE Antennas Propag. Mag.*, vol. 35, no. 1, pp. 84–89, Feb. 1993.