MEMORY-EFFICIENT MULTILEVEL PHYSICAL OPTICS ALGORITHM FOR THE SOLUTION OF ELECTROMAGNETIC SCATTERING PROBLEMS

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND

ELECTRONICS ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCES

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Kaplan Alp Manyas

September 2007

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(Supervisor)

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. Adnan Köksal

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.

Assist. Prof. Dr. Vakur B. Ertürk

Approved for the Institute of Engineering and Sciences:

Prof. Dr. Mehmet Baray
Director of Institute of Engineering and Sciences

ABSTRACT

MEMORY-EFFICIENT MULTILEVEL PHYSICAL OPTICS ALGORITHM FOR THE SOLUTION OF ELECTROMAGNETIC SCATTERING PROBLEMS

Kaplan Alp Manyas
M.S. in Electrical and Electronics Engineering
Supervisor: Prof. Dr. Levent Gürel
September 2007

For the computation of electromagnetic scattering from electrically large targets, physical optics (PO) technique can provide approximate but very fast solutions. Moreover, higher order approximations, such as physical theory of diffraction (PTD) including the diffraction from the edges or sharp corners can also be added to the PO solution in order to enhance the accuracy of the PO. On the other hand, in real-life radar applications, where the computation of the scattering pattern over a range of frequencies and/or angles with sufficient number of samples is desired, further acceleration may be needed. Multilevel physical optics (MLPO) algorithm can be used for such applications, in which a remarkable speed-up can be achieved by evaluating the PO integral in a multilevel fashion. As the correction terms like PTD are evaluated independently just on the edges or sharp corners, whereas the PO integration is carried out on the entire target surface, PO integration is the dominant factor in the computational time of such higher order approximations. Therefore accelerating the PO integration will also reduce the computational time of such higher order approximations. In this thesis, we propose two different improvements on the MLPO algorithm.

First improvement is the modification of the algorithm that enables the solution of the scattering problems involving nonuniform triangulations, thus decreasing the CPU time. Second improvement is the memory-efficient version, in which the $O(N^3)$ memory requirement is decreased to $O(N^2 \log N)$. Efficiency of the two proposed improvements are demonstrated in numerical examples including a real-life scattering problem, with which the scattering pattern of a three-dimensional stealth target is evaluated as a function of elevation angle, azimuth angle, and frequency.

Keywords: Physical optics; scattering problems; multilevel physical optics algorithm.

ÖZET

SAÇILIM PROBLEMLERİNİN ÇÖZÜMÜ İÇİN BELLEĞİN VERİMLİ KULLANILDIĞI ÇOK SEVİYELİ FİZİKSEL OPTİK ALGORİTMASI

Kaplan Alp Manyas
Elektrik ve Elektronik Mühendisliği Bölümü Yüksek Lisans
Tez Yöneticisi: Prof. Dr. Levent Gürel
Eylül 2007

Fiziksel optik tekniği (FO), elektriksel olarak büyük hedeflerden elektromanyetik saçılımın hesaplanmasında oldukça hızlı çözümler sunabilmektedir. Dahası, gometrinin kenarlarından ve sivri köşelerinden kaynaklanan kırınım etkisini de kapsayan kırınımın fiziksel teorisi (KFT) gibi daha yüksek dereceli yaklaşıklamalarda kullanılarak FO tekniğinin hassasiyeti arttırılabilir. Öte yandan, saçılımın belli bir frekans ve/veya açı aralığında, yeterli sıklıkta örneklenerek hesaplanmasının istendiği durumlarda, daha fazla hızlanmaya gereksinim duyulabilmektedir. Bu çeşit uygulamalarda, FO integralinin çok seviyeli bir yaklaşım ile hesaplandığı çok seviyeli fiziksel optik algoritması (ÇSFO) ile büyük ölçüde hızlanma sağlanabilir. KFT gibi yüksek dereceli yaklaşıklamalarda, kırınım etkisi, sadece geometrinin kenarlarında ve sivri köşelerinde hesaplanarak FO sonucuna eklenir. Öte yandan, FO integrali geometrinin tüm yüzeyi üzerinde hesaplanır ve bu nedenle bu tür yaklaşıklamaların hesaplanma zamanında belirleyici unsurdur ve FO integralinin daha hızlı alınması bu tür yaklaşıklamalarda da kayda değer bir hızlanma sağlayacaktır. Bu tez çalışmasında, bu algoritma üzerinde iki farklı iyileştirme sunulmaktadır. İlk iyileştirme, algoritmanın

düzensiz üçgenlemeleri içeren saçılım problemlerinin de çözülebileceği şekilde değiştirilerek, CPU zamanının düşürülmesidir. İkinci iyileştirme ise algoritmanın $O(N^3)$ bellek gereksiniminin $O(N^2 \log N)$'e düşürüldüğü, belleğin verimli kullanıldığı uyarlamasıdır. Sunulan bu iki iyileştirmenin verimlilikleri ise, üç boyutlu bir hayalet uçağın saçılım örgüsünün yükseliş açısına, yanca açısına ve frekansa bağlı olarak hesaplandığı, gerçek hayatta karşılaşılan bir problemin de aralarında bulunduğu sayısal örnekler ile sunulmuştur.

Anahtar Kelimeler: Fiziksel optik, saçılım problemleri, çok seviyeli fiziksel optik algoritması

ACKNOWLEDGMENTS

I would like to thank my supervisor Dr. Levent Gürel for his supervision, guidance, and suggestions throughout the development of this thesis. I would also like to thank him for his support and trust. Without him, I would never had a chance to even start working towards a degree of master of science.

I also thank Prof. Dr. Adnan Köksal and Assist. Prof. Vakur B. Ertürk for reading and commenting on this thesis.

Contents

List of Figures

List of Tables

To my mother and sister...

Chapter 1

INTRODUCTION

Physical optics (PO) is a high-frequency technique that can offer quick but approximate solutions for the scattering problems involving smooth, convex, and electrically large targets. On the other hand, speed-up for even PO can be needed for problems, for which the solutions are desired in a range of frequencies and/or angles.

1.1 Historical Background

Current-based high-frequency techniques, such as PO [1] and physical theory of diffraction (PTD) [2], are widely used asymptotic techniques to approximate the solution of the electromagnetic scattering from electrically large, convex, and smooth targets. In those techniques, PO integration is the dominant factor in the computational time since the additional correction terms are usually in the form of line integrals along the edges or wedges. For scattering problems with complex targets, MLPO algorithm turns out to be a suitable approach to decrease the PO integration time [6].

1.2 Motivation

To find the PO scattering pattern of a target, the integral

$$\mathbf{E}^s(\mathbf{r}) = -jk\eta \frac{e^{-jkr}}{2\pi r} \int_S \hat{\mathbf{n}}(\mathbf{r}') \times \mathbf{H}_0^i e^{jk(\hat{\mathbf{r}}-\hat{\mathbf{r}}_i)\cdot\mathbf{r}'} \mathbf{d}s' \tag{1.1}$$

should be evaluated on the lit regions of the target. In this equation, $\mathbf{r}'$ is the source point on S, $\hat{\mathbf{n}}(\mathbf{r}')$ is the normal of the surface at $\mathbf{r}'$, $\mathbf{H}_0^i$ is the incident magnetic field vector, $\hat{\mathbf{r}}$ is the unit vector along the direction of observation, and $\hat{\mathbf{r}}_i$ is the unit vector along the direction of incidence. As the target surface can be arbitrarily complex, obtaining an explicit solution for this integral is usually impossible. Therefore, dividing the surface area into triangles and evaluating the PO integral analytically on each triangle is a common approach [8]. PO integral evaluated in this manner takes the form of

$$\mathbf{E}^s(\mathbf{r}) = -jk\eta \frac{-e^{jkr}}{2\pi r} \sum_{k=1}^{T} \hat{\mathbf{n}}_k \times \mathbf{H}_0^i \int_{\Delta_k} e^{jk(\hat{\mathbf{r}}-\hat{\mathbf{r}}_i)\cdot\mathbf{r}'} \mathbf{d}s', \tag{1.2}$$

where T is the total number of triangles. Let R be the radius of the smallest sphere that can contain the target and $N = kR$, where k is the wavenumber. Then, the total number of triangles will be proportional to N^2.

In order to be able to interpolate such a scattering pattern from its samples with a prescribed error, the sampling rate should be proportional to the electrical size of the target. The required number of samples in θ, ϕ, and frequency are $O(N)$ each.

Therefore, computational complexity of finding the scattering pattern over a range of frequencies and angles with sufficient number of samples turns out to be $O(N^5)$. In Chapter 3, it will be shown that this computational complexity can

be reduced to $O(N^3 \log N)$ without sacrificing the accuracy, but with a memory cost of $O(N^3 \log N)$.

1.3 Our Contributions

In this thesis, two-dimensional (2-D) MLPO algorithm proposed by Boag [6] is applied to 3-D problems. We also propose two improvements on the MLPO algorithm. First improvement is the modification of the algorithm that enables the solution of the scattering problems involving nonuniform triangulations, thus decreasing the CPU time. Second improvement is the memory-efficient version, in which the $O(N^3)$ memory requirement is decreased to $O(N^2 \log N)$.

1.4 Outline

In Chapter 2, we first introduce the PO approximation. In Chapter 2, we also mention the properties of the PO approximation that are basic tools for the MLPO algorithm presented in Chapter 3. The MLPO algorithm for nonuniform triangulations is proposed in Chapter 4, where the computational efficiency of the proposed algorithm is also demonstrated. We propose the memory-efficient MLPO algorithm in Chapter 5 and demonstrate its memory efficiency. We conclude and list future research areas in Chapter 6.

Chapter 2

PO APPROXIMATION

Integral-equation methods, such as the method of moments [3], fast multipole method [4], and multilevel fast multipole method [5], in which the current is solved to satisfy the boundary condition, are usually impractical for electrically large problems. This is because, in these methods, the required computational resources increase rapidly with the electrical size of the problem. On the other hand, in the PO approximation, the surface current $\mathbf{J}_s$ is assumed to be

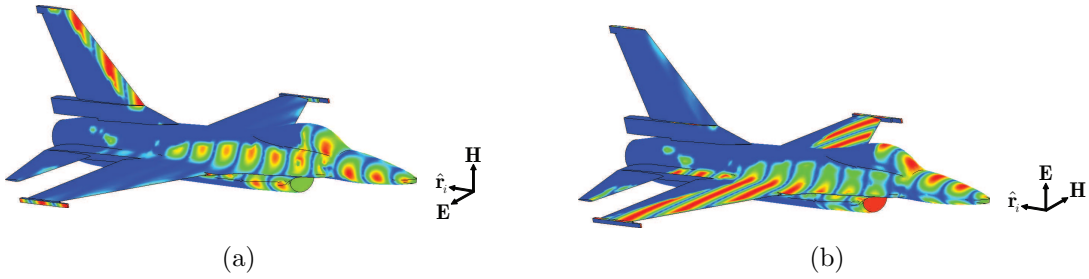
$$\mathbf{J}_s = 2\hat{\mathbf{n}} \times \mathbf{H}^{inc} \tag{2.1}$$

in the lit regions, instead of being solved. This assumption offers less accurate but faster solutions. Here $\hat{\mathbf{n}}$ is the surface normal and $\mathbf{H}^{inc}$ is the incident magnetic field. In Fig. 2.1, the induced PO currents on an F-16 plane is given as an illustration for horizontally and vertically polarized incidences. Under this assumption, for a plane-wave incidence $\mathbf{H}^{inc}(\mathbf{r}) = \mathbf{H}_0^{inc} e^{-jk\mathbf{r}\cdot\hat{\mathbf{r}}_i}$, the far-field radiation integral

$$\mathbf{E}^{scat}(\mathbf{r}) = -jk\eta \frac{e^{-jkr}}{4\pi r} \int \mathbf{J}_s(\mathbf{r}') e^{jk\hat{\mathbf{r}}\cdot\mathbf{r}'} \mathbf{dr}' \tag{2.2}$$

takes the form of

$$\mathbf{E}^{scat}(\mathbf{r}) = -jk\eta \frac{e^{-jkr}}{2\pi r} \int\limits_{\mathbf{r}' \in S^{lit}} \widehat{\mathbf{n}} \times \mathbf{H}_0^{inc} e^{jk\mathbf{r}'\cdot(\hat{\mathbf{r}}-\hat{\mathbf{r}}_i)} \mathbf{dr}'. \tag{2.3}$$



(a) (b)

Figure 2.1: PO surface current density: (a) For horizontally polarized electric field incidence. (b) For vertically polarized electric field incidence.

In operator notation,

$$\Psi S = -jk\eta \frac{e^{-jkr}}{2\pi r} \int\limits_{\mathbf{r}' \in S^{lit}} \hat{\mathbf{n}} \times \mathbf{H}_0^{inc} e^{jk\mathbf{r}' \cdot (\hat{\mathbf{r}} - \hat{\mathbf{r}}_i)} \mathbf{dr}', \tag{2.4}$$

where Ψ is the PO operator evaluating the scattering pattern of S. Note that the $\mathbf{r}'$ dependence of $\hat{\mathbf{n}}$ is suppressed for simplicity. Since it is difficult to evaluate the integral in Equation (2.3) analytically for an arbitrary complex 3-D target, triangulation or triangular meshing is a common approach for modeling 3-D targets with arbitrary complex geometries. Triangulations of various geometries are shown in Figs. 2.2 and 2.3.

In a uniform triangulation, the triangle size (or mesh size) is usually chosen approximately between $\lambda/10$ and $\lambda/5$ depending on the desired accuracy. In a nonuniform triangulation, the triangle size can be adjusted locally according to the surface curvature. More detailed description of nonuniform triangulation is presented in Section 4.1.

After the triangulation of the geometry, Gordon's well-known formula for the analytic evaluation of PO integral over planar patches [8] can be used to evaluate the PO radiation from each triangle. For a single triangle Δ shown in Fig. 2.4, this formula reduces to

$$\mathbf{E}_{\Delta}^{scat}(\mathbf{r}) = -jk \frac{-e^{jkr}}{2\pi r} \left(\hat{\mathbf{n}}_{\Delta} \times \mathbf{H}_0^{inc} \right) T_{\Delta}, \tag{2.5}$$

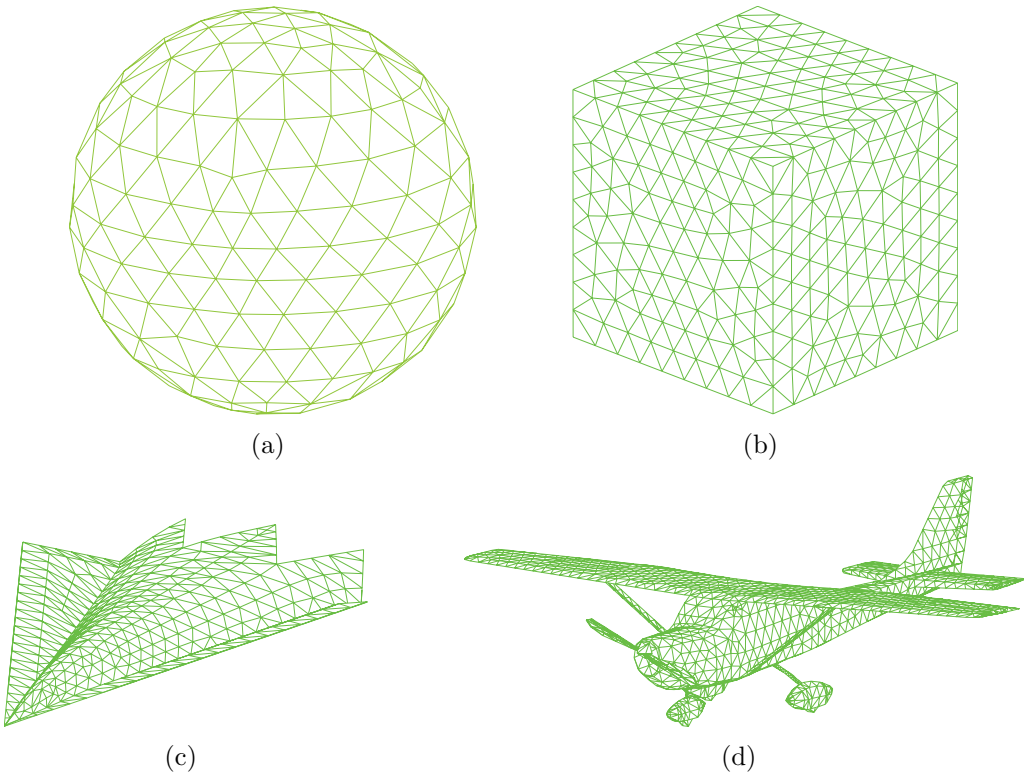


Figure 2.2: Uniform meshes of arbitrary targets: (a) Sphere, (b) cube, (c) a stealth target, and (d) a plane.

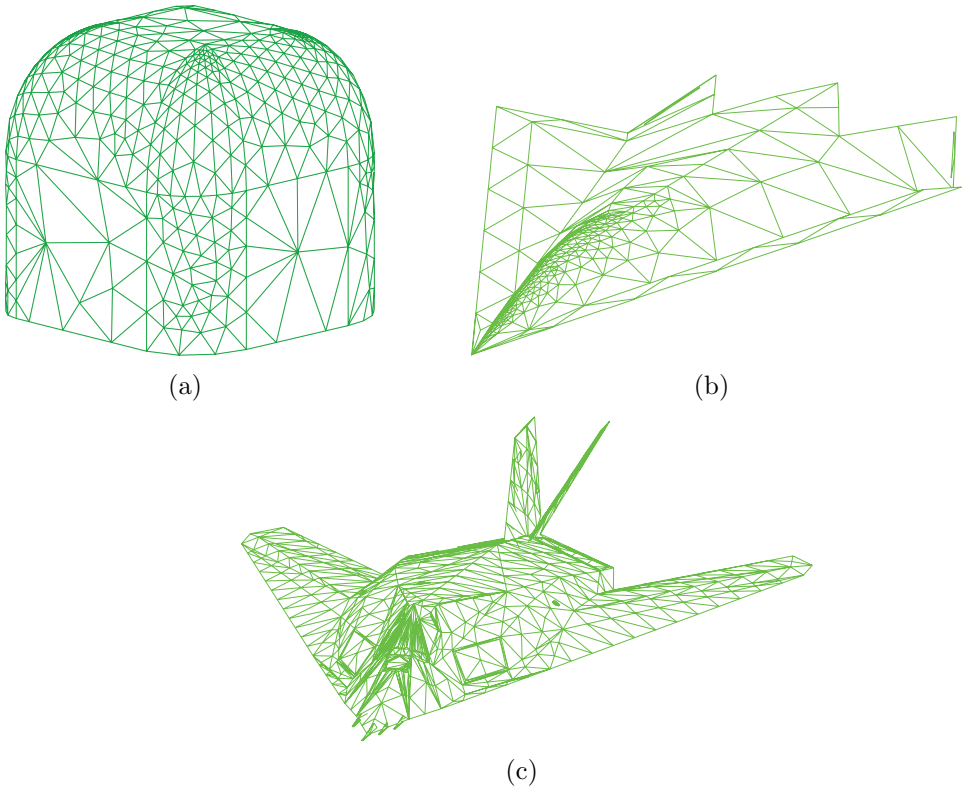
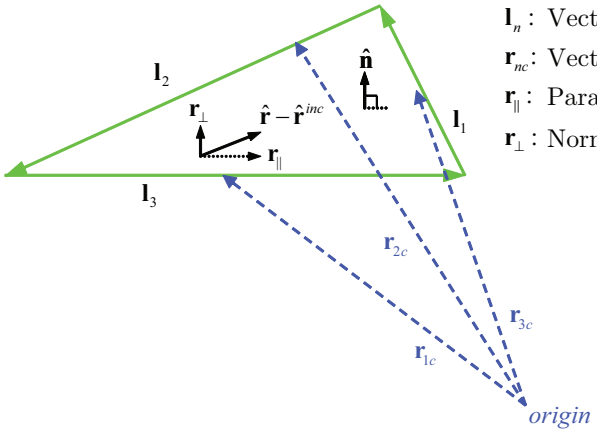


Figure 2.3: Nonuniform meshes of arbitrary targets: (a) Cube with smooth edges, (b) a stealth target, and (c) another stealth target.

where

$$T_{\Delta} = e^{j\mathbf{r}_{\perp}r} \sum_{n=1}^{3} \frac{1}{\left|\mathbf{r}_{\parallel}\right|^2} e^{j\mathbf{r}_{\parallel}\cdot\mathbf{r}_{nc}} \left[\left(\mathbf{l}_n \times \mathbf{r}_{\parallel}\right) \cdot \hat{\mathbf{n}}\right] \operatorname{sinc}\left(\frac{\mathbf{r}_{\parallel} \cdot \mathbf{l_n}}{2}\right). \tag{2.6}$$



$\mathbf{l}_n$: Vector showing the nth edge of the triangle.
$\mathbf{r}_{nc}$: Vector pointing to the center of the nth edge.
$\mathbf{r}_{\parallel}$: Parallel component of $\mathbf{r}-\mathbf{r}^{inc}$ to the triangle.
$\mathbf{r}_{\perp}$: Normal component of $\mathbf{r}-\mathbf{r}^{inc}$ to the triangle.

Figure 2.4: Triangle for the analytic evaluation of the PO integral.

Since Equation (2.6) is a closed-form expression with no integrals, it offers accurate and efficient evaluation of the PO integral over a single triangle. Afterwards, the radiation pattern from each triangle can be added to find the overall scattering pattern as

$$E^{scat} = \sum_{n=1}^{N_t} E^{scat}_{\Delta_n}. \tag{2.7}$$

2.1 Properties of the PO Approximation

The properties of the PO approximation listed below are the basis of the MLPO algorithm presented in Chapter 3.

2.1.1 Superposition

If the target surface S is split into non-overlapping subsurfaces, then the superposition of the radiation patterns of these subsurfaces will be equal to the

scattering pattern of S as

$$\Psi S = \sum_{q=1}^{Q} \Psi S_q. \tag{2.8}$$

In addition to allowing the triangulation of the target surface and evaluation of the PO integral separately on each triangle, this property will be used to group the triangles and their scattering patterns in the MLPO algorithm presented in Chapter 3.

2.1.2 Shift of Origin

Let $O[\mathbf{r}_0]$ denote an operator shifting the target surface by vector $\mathbf{r}_0$ as in Fig. 2.5. Then, direct substitution of $\mathbf{r}' = \mathbf{r}' - \mathbf{r}_0$ in Equation (2.4) yields

$$O[\mathbf{r}_0]\,\Psi S = e^{jk(\hat{\mathbf{r}}-\hat{\mathbf{r}}_i)\cdot\mathbf{r}_0}\Psi S. \tag{2.9}$$

Similarly, $E[\mathbf{r}_0]$, which is the inverse of $O[\mathbf{r}_0]$, can be defined as $O[-\mathbf{r}_0]$, where $O[\mathbf{r}_0]$ and $E[\mathbf{r}_0]$ denote multiplication by $e^{-jk\mathbf{r}_0\cdot(\hat{\mathbf{r}}-\hat{\mathbf{r}}_i)}$ and $e^{jk\mathbf{r}_0\cdot(\hat{\mathbf{r}}-\hat{\mathbf{r}}_i)}$, respectively. This property is essential for removing the extra phase oscillations from complex

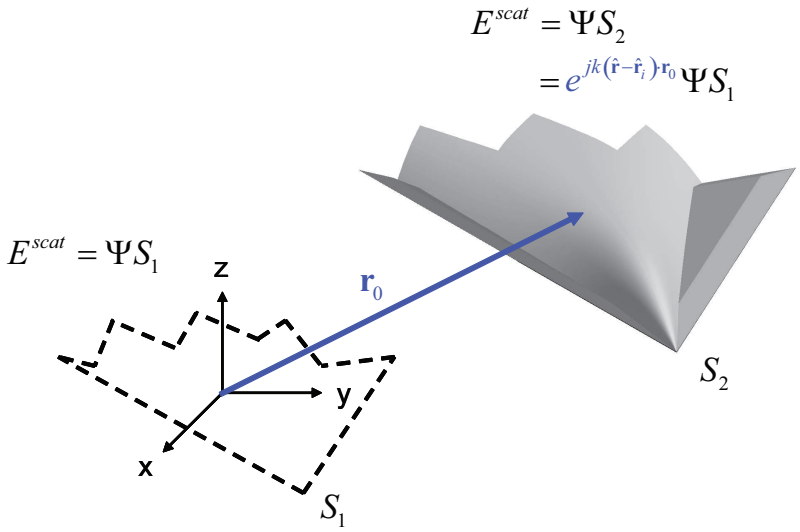


Figure 2.5: Shift of the surface S by a vector $\mathbf{r}_0$.

scattering patterns before the interpolation in the MLPO algorithm to be presented in Chapter 3.

2.1.3 Frequency Sampling

When considered as a function of frequency, the PO scattering pattern in Equation (2.4) can be written as

$$\Psi S = -jk\eta \frac{e^{-jkr}}{2\pi r} \int\limits_{\mathbf{r}' \in S^{lit}} \widehat{\mathbf{n}} \times \mathbf{H}_0^{inc} e^{j\frac{2\pi f}{c}\mathbf{r}'\cdot(\hat{\mathbf{r}}-\hat{\mathbf{r}}_i)} \mathbf{dr}'. \tag{2.10}$$

This expression is in fact the superposition of exponential terms $e^{j\frac{\mathbf{r}'\cdot(\hat{\mathbf{r}}-\hat{\mathbf{r}}_i)}{c}2\pi f}$. If R is the radius of the smallest sphere that can contain the target as in Fig. 2.6, then $\frac{\mathbf{r}'\cdot(\hat{\mathbf{r}}-\hat{\mathbf{r}}_i)}{c}$ can take a maximum value of $2R/c$. This occurs when $\hat{\mathbf{r}}-\hat{\mathbf{r}}_i$ (backscattering case) and $\mathbf{r}'$ is in the same direction with $\hat{\mathbf{r}} = -\hat{\mathbf{r}}_i$. According to the Nyquist

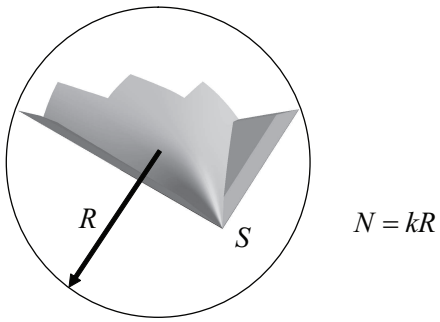


Figure 2.6: Illustration of the smallest sphere that can contain the target.

sampling theorem [10], f should be sampled with $\Delta f < c/4R$, where Δf is the sampling period of f. As a consequence, in order to allow interpolation, the required number of frequency samples is

$$N_f = \Omega_f \frac{4R\,(f_{\max} - f_{\min})}{c}, \tag{2.11}$$

where Ω_s is a real number greater than 1 indicating the ratio of the sampling rate to the theoretical one. As a result, the number of frequency samples is $O\,(N)$, where $N = kR$.

In Fig. 2.8, spectrum of the backscattering signal from a helicopter model shown in Fig. 2.7 is presented in order to verify the validity of Equation (2.11). In the same figure, the resolved band for different oversampling ratios are also shown.

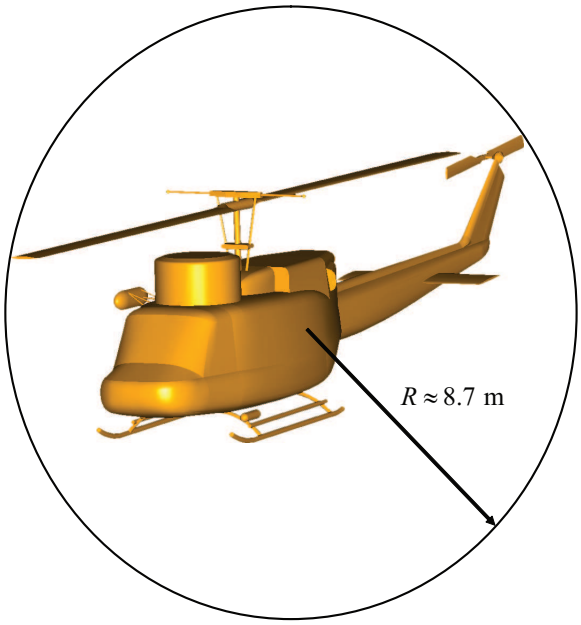


Figure 2.7: A helicopter model and the illustration of the smallest sphere that can contain it.

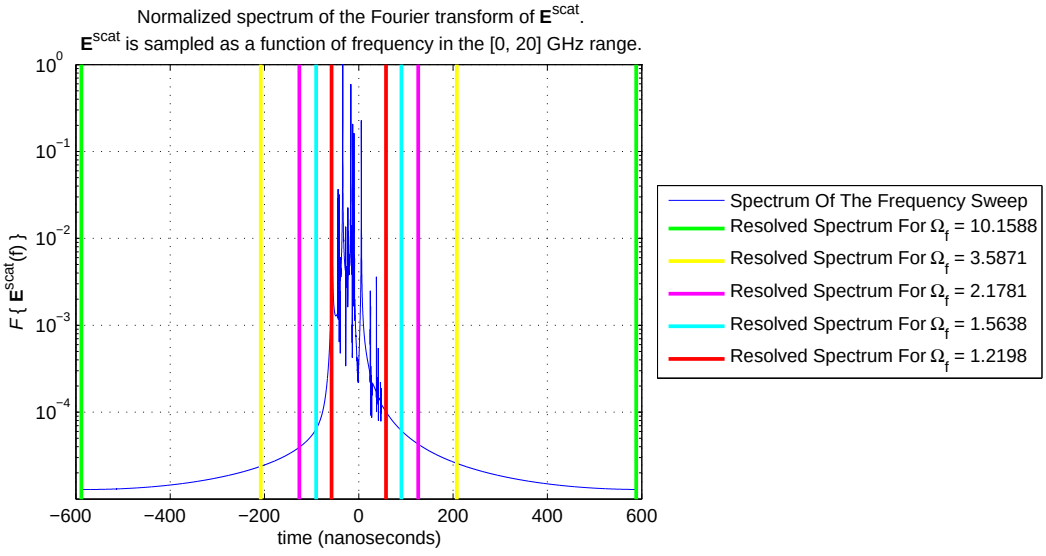


Figure 2.8: Spectrum of the $\mathbf{E}^{scat}$ signal as a function of frequency for the backscattering case. The resolved bands for different oversampling ratios are also shown in the same figure.

From Fig. 2.8, it is observed that $\Omega_f \approx 1$ is sufficient to resolve the harmonics having a normalized weight of 10^{-4}. On the other hand, in Fig 2.9, it is shown that in order to have a norm error

$$\|error\|_{normalized} = \frac{\left\|S^{ref} - S^{ip}\right\|}{\|S^{ip}\|} \tag{2.12}$$

below 10^{-3}, the oversampling ratio in frequency should be greater than 3. Note that S^{ref} is the reference signal and S^{ip} is the interpolated signal in Equation (2.12).

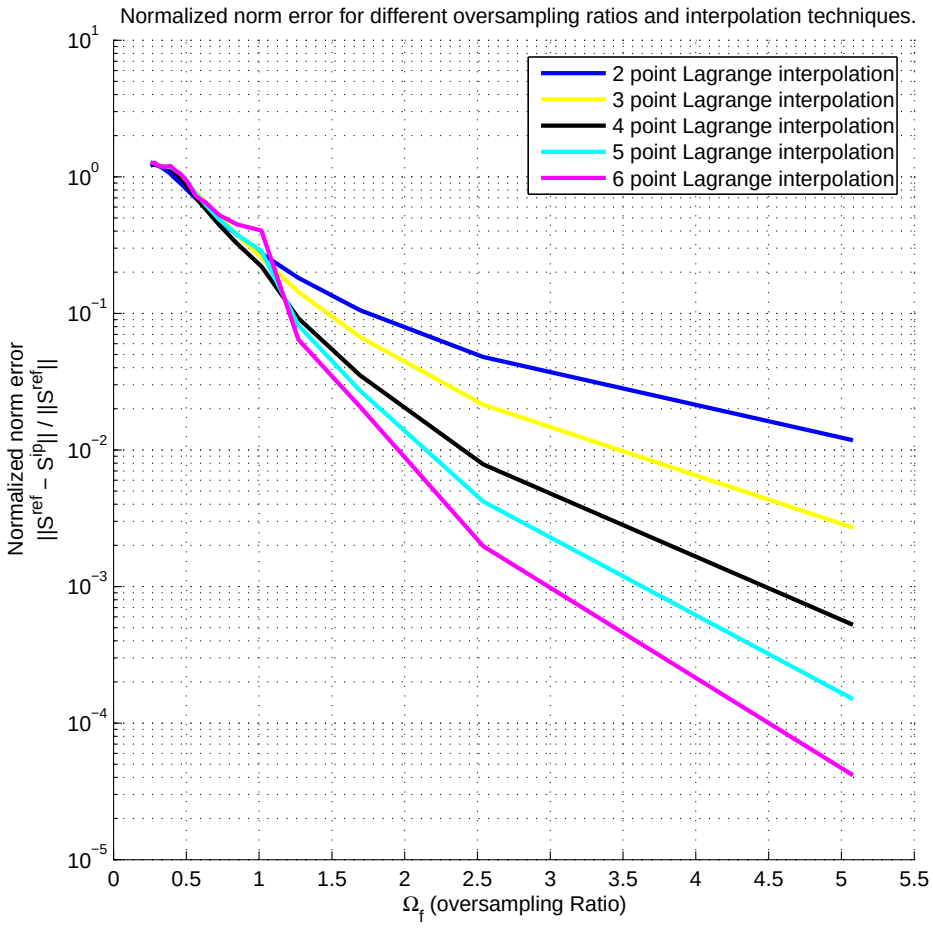


Figure 2.9: Increase of accuracy as the sampling rate increases.

.

Fig 2.9 also shows that by employing different number of points in the interpolation, the interpolation error can be decreased for a given oversampling ratio.

It should be noted since the Lagrange interpolation is a local interpolation, increasing the number of points does not increase the computational complexity for uniformly sampled data. More detailed discussion of the Lagrange interpolation can be found in Appendix.

2.1.4 Angular Sampling

As mentioned in Section 2.1.3, the PO scattering pattern has the largest oscillation rate for the backscattering case, where $\hat{\mathbf{r}}_i = -\hat{\mathbf{r}}$. For the sake of simplicity, the minimum angular sampling rate will be derived for the backscattering case, which is also applicable to the bistatic case. For the backscattering case, the PO scattering pattern in Equation (2.4) can be written as

$$\mathbf{E}^{scat}(\mathbf{r}) = -jk\eta \frac{e^{-jkr}}{2\pi r} \int\limits_{\mathbf{r}' \in S^{lit}} \widehat{\mathbf{n}} \times \mathbf{H}_0^{inc} e^{2jkr'[\sin(\theta)\sin(\theta')\cos(\phi-\phi')+\cos(\theta)\cos(\theta')]} \mathbf{dr}'. \tag{2.13}$$

In this expression, (θ, ϕ) and (θ', ϕ') are the elevation and azimuth angles of the direction of scattering and the direction of source point, respectively. For the backscattering case, as a function of ϕ, PO scattering pattern in Equation (2.4) can be written as

$$\mathbf{E}^{scat}(\mathbf{r}) = -jk\eta \frac{e^{-jkr}}{2\pi r} \int\limits_{\mathbf{r}' \in S^{lit}} \widehat{\mathbf{n}} \times \mathbf{H}_0^{inc} e^{j4\pi fr' \cos(\theta)\cos(\theta')} e^{j4\pi fr' \sin(\theta)\sin(\theta')\cos(\phi-\phi')} \mathbf{dr}'. \tag{2.14}$$

This expression can be thought of as the superposition of exponential terms of type $e^{j\beta\cos(\phi-\phi')}$, where $\beta = 4\pi fr' \sin(\theta)\sin(\theta')$. Fourier series expansion of these terms yields [9]

$$e^{j\beta\cos(\phi-\phi')} = J_0(\beta) + 2\sum_{n=1}^{\infty} j^n J_n(\beta)\cos[n(\phi-\phi')]. \tag{2.15}$$

Here $J_n(\cdot)$ represents the Bessel function of order n. Since Bessel functions decay faster than exponentially for orders higher than the argument (i.e., $n > \beta$) [9], the maximum oscillation rate of these harmonics in Equation (2.15) is bounded with

β. As the maximum value that β can take is $4\pi fR/c$, the maximum oscillation rate of the harmonics in Equation (2.15) is bounded with $4\pi fR/c$. From Nyquist sampling theorem, the sampling rate in ϕ must be twice this value. Therefore, $\Delta\phi$, which is the sampling interval in ϕ, should be at most $c/8\pi f_{\max}R$. As a consequence, for an angular range of $[0, 2\pi]$, the number of ϕ samples is

$$N_\phi = \Omega_\phi \frac{8\pi f_{\max} R}{c}. \tag{2.16}$$

Therefore, the number of ϕ samples that will allow interpolation grows with $O(N)$. Similarly, the number of θ samples is

$$N_\theta = \Omega_\theta \frac{4\pi f_{\max} R}{c} \tag{2.17}$$

since θ is to be sampled in the $[0, \pi]$ range instead of $[0, 2\pi]$. Here Ω_ϕ and Ω_θ are real numbers greater than 1 indicating the ratio of the actual sampling rates to the theoretical ones.

As in the frequency sampling section, in order to verify the validity of Equation (2.11), spectrum of the backscattering from a helicopter model (Fig. 2.7) is given in Fig. 2.8. In the same figure, the resolved band for different oversampling ratios are also shown.

From Fig. 2.8, it is observed that increasing Ω_ϕ beyond 3 does not decrease the norm error significantly. This can be explained by the illumination effect. By employing the visibility function

$$V(\mathbf{r}') = \left\{ \begin{array}{l} 1, \mathbf{r}' \in S_{lit} \\ 0, \text{otherwise} \end{array} \right\}, \tag{2.18}$$

PO operator given in Equation (2.4) can be written as

$$\Psi S = -jk\eta \frac{e^{-jkr}}{2\pi r} \int\limits_{\mathbf{r}' \in S} V(\mathbf{r}')\, \widehat{\mathbf{n}} \times \mathbf{H}_0^{inc} e^{jk\mathbf{r}'\cdot(\hat{\mathbf{r}}-\hat{\mathbf{r}}_i)} d\mathbf{r}'. \tag{2.19}$$

It is clear that since $\widehat{\mathbf{n}} \times \mathbf{H}_0^{inc}$ term in the Fourier transform integral is multiplied with the visibility function$V(\mathbf{r}')$, the spectrum of the scattered signal is also

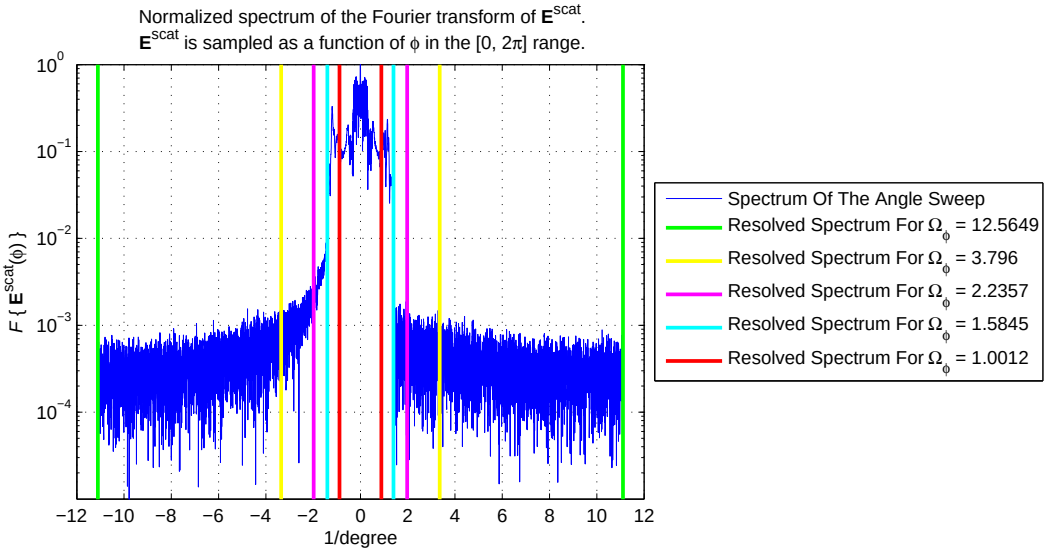


Figure 2.10: Spectrum of the $\mathbf{E}^{scat}$ signal as a function of ϕ for the backscattering case. The resolved bands for different oversampling ratios are also shown in the same figure.

broadened. This fact can be verified from Fig. 2.10, where the spectrum of the scattered signal decays slower compared to Fig. 2.8. In addition, as the PO operator given in Equation (2.19) is evaluated on a coarse grid at the bottom level, illumination is also computed for this coarse grid. Therefore at each level, illumination is also interpolated with scattering patterns of the subdomains.

.

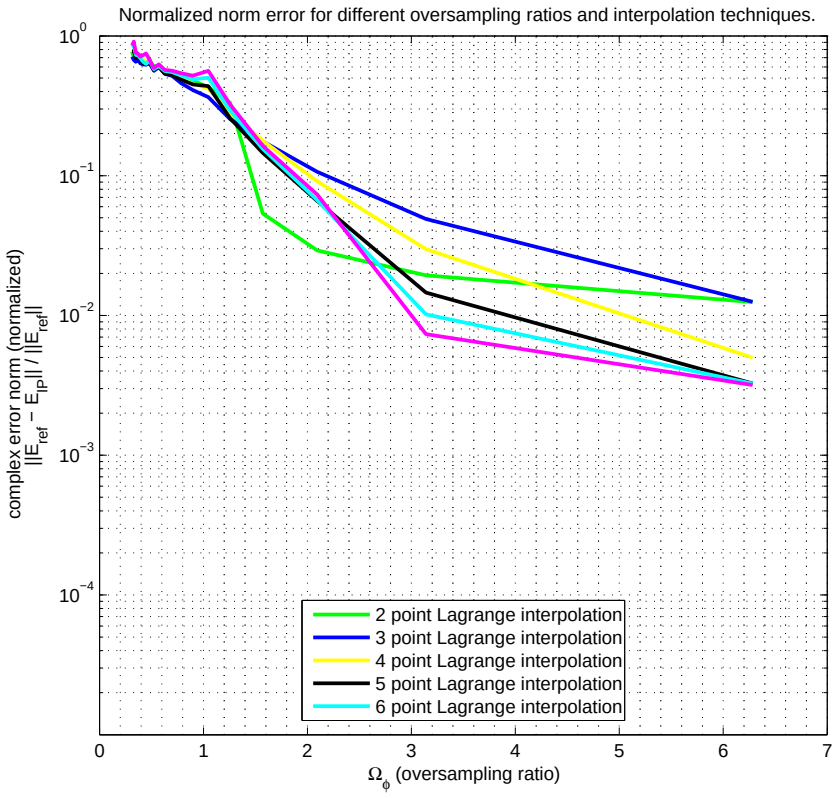


Figure 2.11: Increase of accuracy as the sampling rate increases.

Chapter 3

MLPO ALGORITHM

When computing the PO scattering pattern with sufficient number of samples as a function of θ, ϕ, and frequency, sampling rate in each dimension should be proportional to the electrical size of the target. Let R be the smallest radius of a sphere that can contain the target and $N = kR$, where k is the wavenumber. Then, the required number of samples in θ, ϕ, and frequency are $O(N)$ each and the total number of required samples is $O(N^3)$. If the target surface is modeled with a triangular mesh, there will be $O(N^2)$ triangles as the number of triangles will be proportional to the surface area. Hence, the computational complexity of evaluating the PO integral analytically [8] on the triangular mesh for each θ, ϕ, and frequency turns out to be $O(N^5)$. The MLPO algorithm aimed to decrease this complexity is based on the decomposition of the target surface S into nonoverlapping subdomains. Since each subdomain will have a smaller size, its scattering pattern can be sampled at a rate lower than the rate required for the scattering pattern of S as a whole. After the evaluation of the subdomain scattering patterns at lower sampling rates, these patterns can be aggregated to find the scattering pattern of the whole geometry. As the pattern evaluated by the aggregation of the smaller subdomain patterns will be larger, the subdomain

scattering patterns should be interpolated to a finer grid before aggregation. This scheme is illustrated in Fig. 3.1.

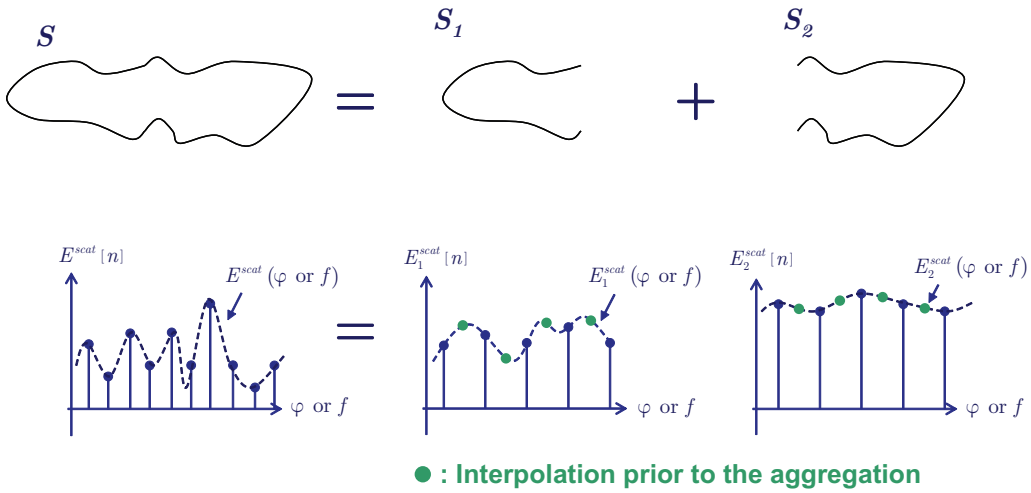


Figure 3.1: Calculating the scattering pattern of S as a sum of scattering patterns of smaller subsurfaces S_1 and S_2.

Since the origin of each subdomain will be different from the global origin, subdomain patterns will oscillate at higher rates because of the phase shift (Fig. 2.5). Therefore, prior to interpolation, subdomain patterns should be shifted to the origin and restored after interpolation. In this scheme, PO operator Ψ that computes the scattering pattern of any arbitrary surface S can be written as

$$\Psi S = \sum_{q=1}^{Q} E\left[\bar{\mathbf{r}}_q\right] \mathbf{I}_{\bar{N}_\theta \bar{N}_\phi \bar{N}_f}^{N_\theta N_\phi N_f} O\left[\bar{\mathbf{r}}_q\right] \Psi \bar{S}_q. \tag{3.1}$$

Here, $\bar{\mathbf{r}}_q$ is the center of the smallest sphere that can include the qth subdomain. $O\left[\bar{\mathbf{r}}_q\right]$ is the operator that shifts the origin of the qth subdomain to the global origin in order to remove the phase oscillations. The $\mathbf{I}_{\bar{N}_\theta \bar{N}_\phi \bar{N}_f}^{N_\theta N_\phi N_f}$ matrix is the interpolation matrix that increases the number of samples from $\bar{N}_\theta \times \bar{N}_\phi \times \bar{N}_f$ points to $N_\theta \times N_\phi \times N_f$ points, and $E\left[\bar{\mathbf{r}}_q\right]$ is the operator that shifts the origin of the qth subdomain back to its location after the interpolation. Efficient implementation of interpolation $\mathbf{I}_{\bar{N}_\theta \bar{N}_\phi \bar{N}_f}^{N_\theta N_\phi N_f}$ is given in Appendix A.

In the MLPO algorithm, each subdomain S_q is recursively subdivided into smaller subdomains and the scattering patterns of these subdomains are also calculated using the PO operator Ψ in Equation (3.1). When the subdomain size is in the

order of λ (wavelength), the subdivision process is stopped and the scattering patterns of the lowest-level subdomains are evaluated with the direct evaluation of the PO integral. Hence in the MLPO algorithm, the PO operator Ψ in Equation (3.1) takes the form of

$$\Psi S = \sum_{q=1}^{Q} E\left[\bar{\mathbf{r}}_q\right] \mathbf{I}_{\bar{N}_\theta \bar{N}_\phi \bar{N}_f}^{N_\theta N_\phi N_f} O\left[\bar{\mathbf{r}}_q\right] \sum_{q=1}^{Q} E\left[\bar{\mathbf{r}}_q\right] \mathbf{I}_{\bar{N}_\theta \bar{N}_\phi \bar{N}_f}^{N_\theta N_\phi N_f} O\left[\bar{\mathbf{r}}_q\right] \cdots \Psi \bar{S}_q. \tag{3.2}$$

3.1 Computation Time

As the electrical size of the bottom-level subdomains will be bounded, the required number of θ, ϕ, and frequency samples will be fixed for each subdomain at this level. There will be $O\left(N^2\right)$ filled subdomains at this level and therefore calculating the PO patterns of all subdomains analytically at this level will require $O\left(N^2\right)$ operations. At each aggregation step, local interpolations transforming the scattering patterns form a coarse grid of θ, ϕ, and frequency to a finer grid will require $O\left(N^3\right)$ operations. As there will be $O\left(\log N\right)$ levels, total computational cost of aggregations will be $O\left(N^3 \log N\right)$. Therefore, the overall computational complexity of the MLPO algorithm is $O\left(N^2\right) + O\left(N^3 \log N\right) = O\left(N^3 \log N\right)$. This complexity is far less than the $O\left(N^5\right)$ complexity of the conventional PO integration.

3.2 Memory Requirement

As mentioned before, the number of pattern samples in the bottom level is fixed and does not grow with N. Since the number of subdomains in the bottom level is $O\left(N^2\right)$, memory required to store the scattering patterns of the bottom-level subdomains is also $O\left(N^2\right)$. When aggregating from lower levels to the upper levels, the required memory to store the scattering patterns of the subdomains will grow by a factor of 8 as the subdomain sizes are doubled. Since PO current is

only on the surface of the target, we can assume that the number of filled clusters is reduced by a factor of 4 at each higher level. Therefore, memory required at each aggregation step will increase by a factor of $8/4 = 2$. Hence, at the uppermost level, memory requirement will be $O\left(N^2 2^{\log N}\right)$, which is $O\left(N^3\right)$. As will be shown in Chapter 5 , this memory requirement may prevent the solution of larger problems using the MLPO algorithm. In Chapter 5, we present a memory-efficient implementation that reduces the memory complexity from $O\left(N^3\right)$ to $O\left(N^2 \log N\right)$.

Chapter 4

MLPO ALGORITHM FOR NONUNIFORM TRIANGULATIONS

4.1 Nonuniform Triangulation

In electromagnetic scattering problems, triangulation or triangular meshing is a common approach for modelling targets with complex geometries. In a uniform triangulation, the triangle size (or mesh size) is usually chosen approximately between $\lambda/10$ and $\lambda/5$ depending on the desired accuracy. In this scheme, the number of triangles is directly proportional to the surface area of the target and the complexity of the number of triangles grows with $O(N^2)$.

On the other hand, if a nonuniform triangulation is employed, instead of adjusting the size of each triangle approximately equal to some fraction of λ, the triangle size can be adjusted according to the surface curvature of the target, i.e., the triangle size gets smaller where the curvature is high and larger where the curvature is low. In such a triangulation, the size of each triangle can be

iteratively modified until the scattering pattern of each subdomain, evaluated via PO, converges [7]. Alternatively, the size of each triangle can be adjusted until the distance of each triangle from the target surface is less than a predefined deviation. Second alternative is chosen for simplicity in the example problems that demonstrate the application of the MLPO algorithm on nonuniform triangulations.

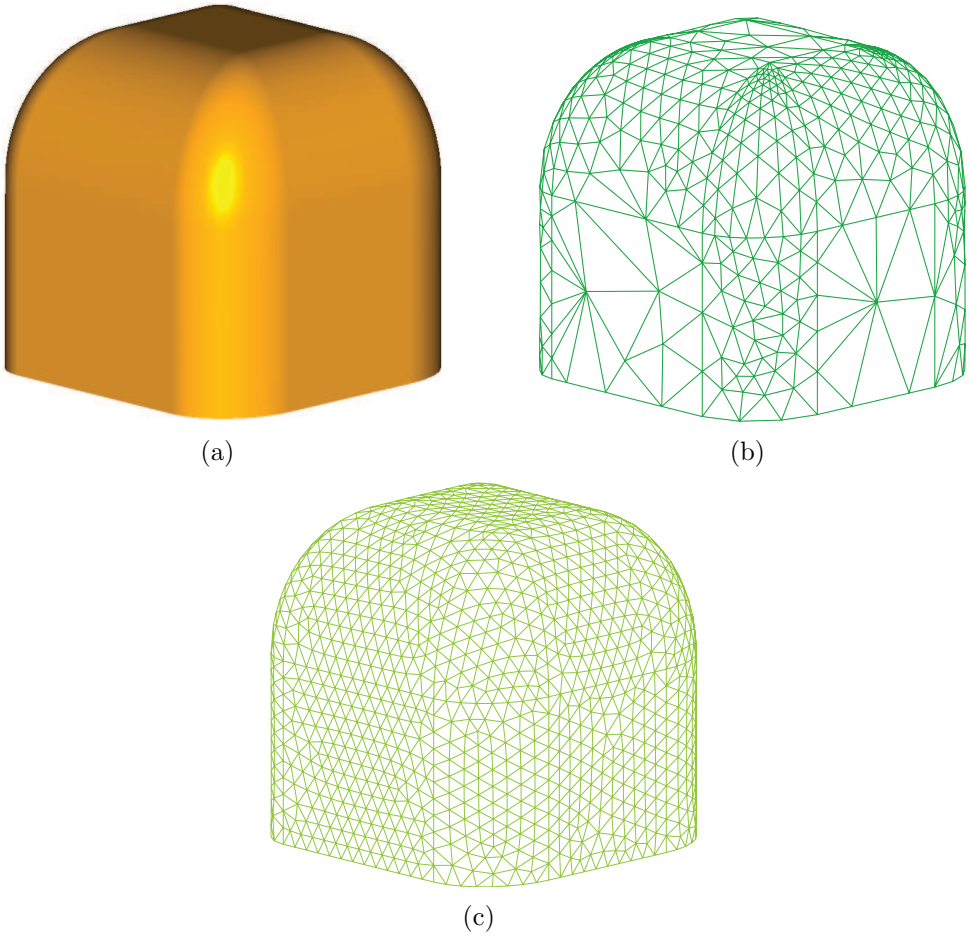
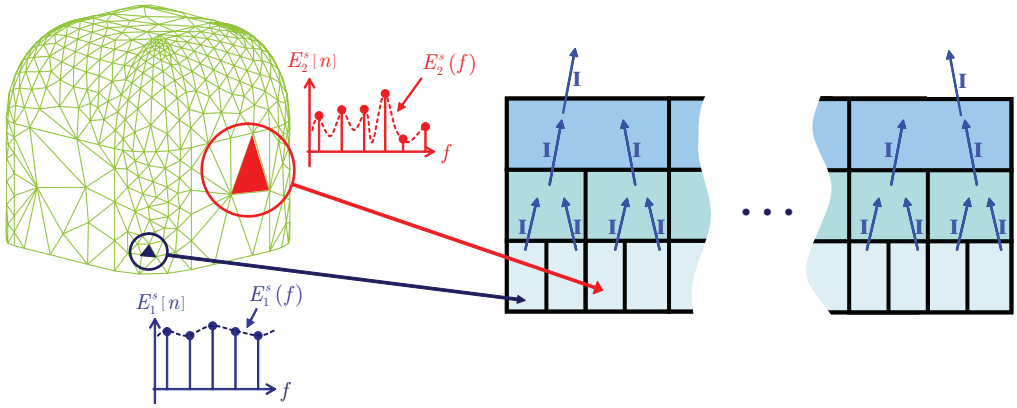


Figure 4.1: An example geometry with both smooth and curved regions: (a) Shaded view. (b) A uniform triangulation example. (c) A nonuniform triangulation example.

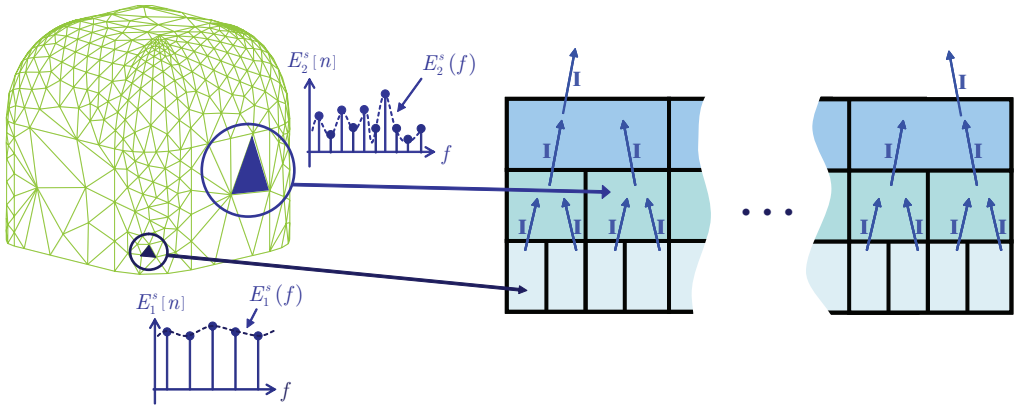
4.2 Modified MLPO Algorithm

As mentioned in Chapter 3, in the MLPO algorithm, only the scattering patterns of the smallest subdomains are computed via PO and those patterns are

sampled at lower rates according to their dimensions. In a nonuniform triangulation, there may be triangles that are too large to fit in the smallest subdomains. Since the required sampling rate for the patterns of these triangles will be higher, sampling and aggregation of these triangles in the lowest levels will result in an interpolation error that grows at each aggregation step as illustrated in Fig. 4.2(a). Therefore the MLPO algorithm is modified in such a way that the radiation patterns of the larger triangles are sampled at a rate proportional to their dimensions and aggregated at the appropriate levels as in Fig. 4.2(b).



(a)



(b)

Figure 4.2: Aggregation of larger triangles. (a) Aggregating at the bottom level with a lower sampling rate, which will cause an interpolation error. (b) Aggregating at higher levels with the correct sampling rate, which will prevent the interpolation error.

Table 4.1: Computation of the generic helicopter model's RCS pattern: Number of triangles and CPU times for direct PO evaluation, MLPO algorithm, and modified MLPO algorithm with nonuniform triangulation.

	Direct PO evaluation	MLPO algorithm	Modified MLPO algorithm with nonuniform triangulation
Number of triangles	88,000	88,000	24,000
CPU time	5.7 min	0.26 min	0.075 min

4.3 Results

To illustrate the efficiency of the improved MLPO algorithm for nonuniform triangulations, the geometry shown in Fig. 4.3 is illuminated from the $(\theta^i, \phi^i) = (135°, 90°)$ direction. The RCS results computed with PO and MLPO algorithm are shown in Fig. 4.4

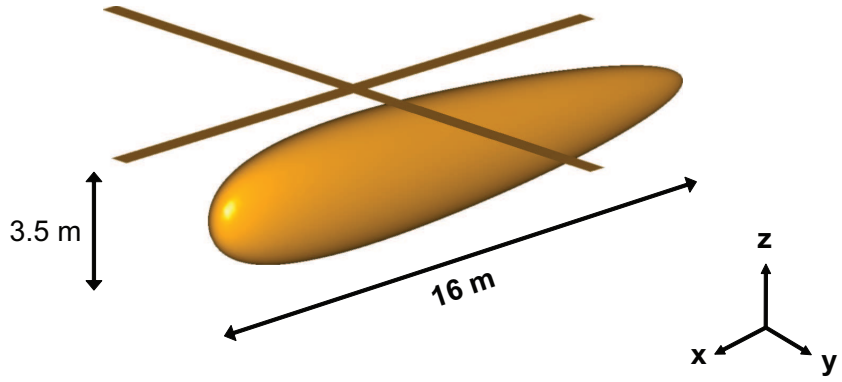
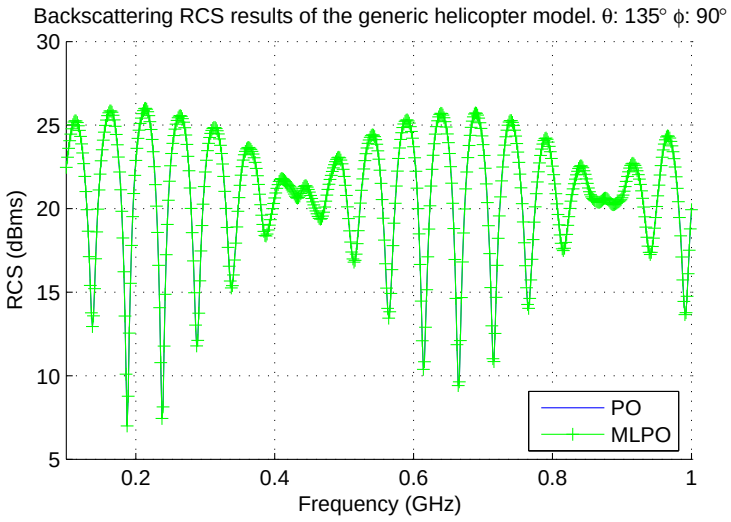


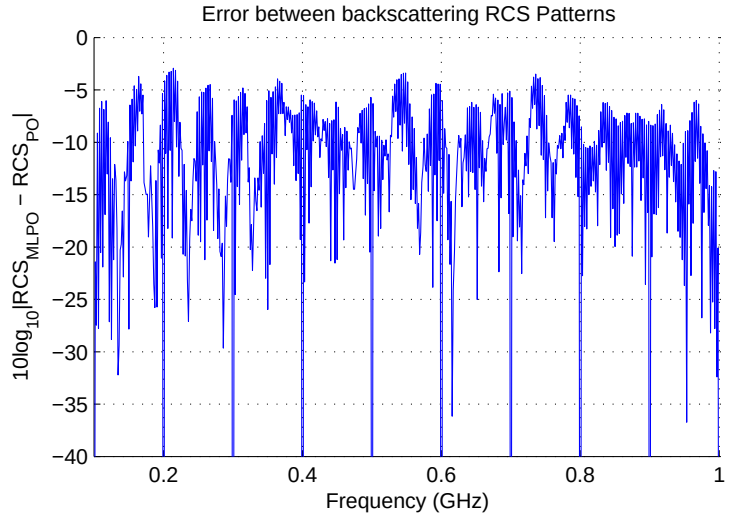
Figure 4.3: Generic helicopter geometry

In Fig. 4.4, RCS results and the error between the RCS patterns calculated with PO integration and MLPO algorithm is presented. It is seen that the error in RCS is below 1%.

The scattering pattern is evaluated for 577 points in the frequency range of $[0.1, 1]$ GHz via direct PO evaluation, MLPO algorithm, and the modified MLPO algorithm for nonuniform triangulations. The number of triangles and CPU times are given in Table 4.1.



(a)



(b)

Figure 4.4: Backscattering RCS pattern of the generic helicopter geometry, computed via direct PO integration and MLPO algorithm: (a) RCS results. (b) Normalized error between the complex scattered fields.

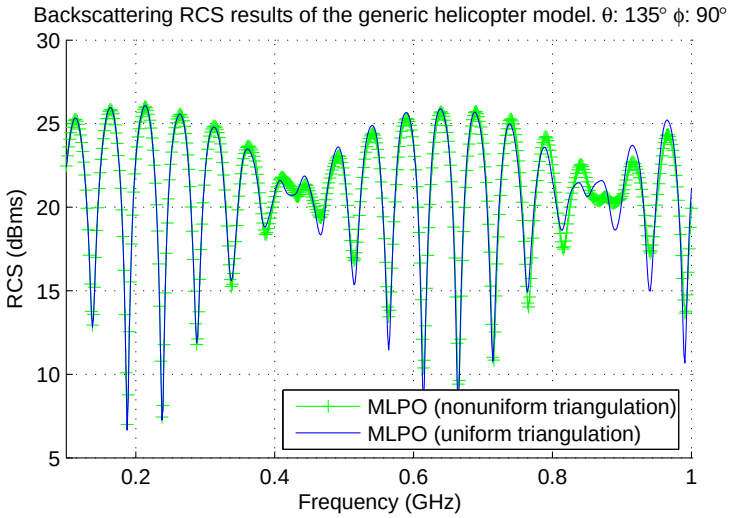


Figure 4.5: Backscattering RCS pattern of the generic helicopter geometry, computed via MLPO algorithm for uniform and nonuniform triangulations.

The nonuniform triangulation consists of approximately 24,000 triangles. The CPU time of the MLPO with uniform triangulation of 6 cm is 0.26 min. On the other hand, CPU time of MLPO with the nonuniform triangulation is 0.075 min. Although a speed-up of nearly 3.5 may seem insufficient, a greater speed-up can be achieved by increasing the deviation of the nonuniform mesh from the geometry and thereby decreasing the accuracy. Moreover, uniform triangulations may require millions of triangles in real-life applications with target sizes up to thousands of λ. Therefore, even generating such triangulations for complex geometries may become impossible and a nonuniform triangulation may become a good choice.

Chapter 5

MEMORY-EFFICIENT MLPO ALGORITHM

The proposed memory-efficient implementation of the MLPO algorithm is based on the idea that, the patterns of the clusters need not be stored for the entire range of θ, ϕ, or frequency values at the same time. By a careful implementation, the θ, ϕ, or frequency ranges can be divided into smaller ranges so that interpolations and aggregations can be performed on those smaller ranges.

For instance, when aggregating the bottom level to the upper level, θ can be sampled in the $\left[0, \frac{\pi}{2}\right]$ range instead of $[0, \pi]$. This way, the memory required for each cluster will grow by a factor of 4 instead of 8. In the next level, the number of clusters will be reduced by a factor of 4, and the total required memory will be constant. Then, θ can be sampled in the $\left[0, \frac{\pi}{4}\right]$ range instead of $\left[0, \frac{\pi}{2}\right]$ at the next level, and this procedure can be applied till the uppermost level is reached. This scheme is illustrated in Fig. 5.1.

If L is the number of levels, after the aggregations are performed, $1/2^{L-1}$ portion of whole scattering pattern will be available at the Lth level. This portion can be output to a file and the remaining portion can be obtained by aggregating the

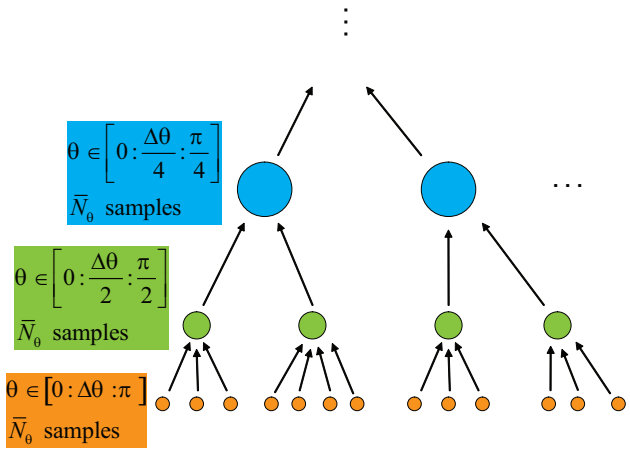


Figure 5.1: Aggregation steps in the proposed memory-efficient algorithm. Note that $\theta \in [0 : \Delta\theta : 180]$ means that the patterns at that level is sampled from 0 to 180, with an increment of $\Delta\theta$

second half to Lth level. Then the 3rd portion of the pattern will be required. This portion will not be available at the $(L-1)$st level. Therefore, the pattern available at the $(L-2)$nd level should be aggregated to the $(L-1)$st level. The 3rd portion of the whole scattering pattern at the Lth level can be obtained by aggregation from the $(L-1)$st level. In this scheme, after 2^{L-1} passes, the whole scattering pattern will be obtained. As an example, aggregations of pattern portions for a 6 -level problem is depicted in Fig. 5.2.

The following pseudo-code describes the partial aggregation of the clusters at each pass:

{n is an array of size $L-1$, indicating which half of the available pattern portion should be aggregated at each pass.}

{In the first pass, first halves of the available pattern portions should be aggregated.}

$n(1:L) \leftarrow 1$

for $t = 1$ to 2^{L-1} **do**

  for $l = 1$ to $L-1$ **do** {all levels 1 to $L-1$}

    $m \leftarrow L - l + 1$

    if mod $(t, 2^{m-2}) = 1$ or $m = 2$ **then**

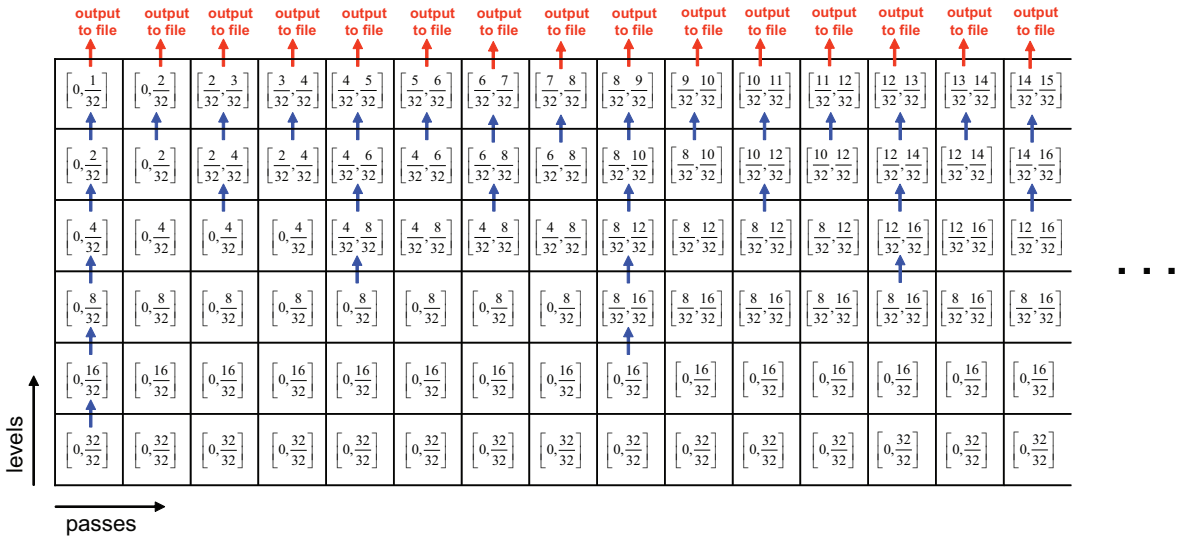


Figure 5.2: Aggregations of pattern portions for a 6-level problem. Aggregations are shown as blue arrows and output to files are shown as red arrows.

for all $cli \in l^{th}$ *level* **do** {all clusters in lth level}

{Aggregate the $n(l)$th half of the cluster.}

$aggregate\_cluster(cli, n(l))$

end for

if $n(l) = 1$ **then**

$n(l) \leftarrow 2$

else

$n(l) \leftarrow 1$

end if

end if

end for{Write the available portion of the whole targets scattering pattern.}

$writeToFile$

end for

It should be noted that higher-order interpolation schemes may be desired in order to prevent the interpolation error in aggregations [11]. In this case, additional sample points at the end points in the partial patterns should be included in the interpolations as illustrated in Fig. 5.3.

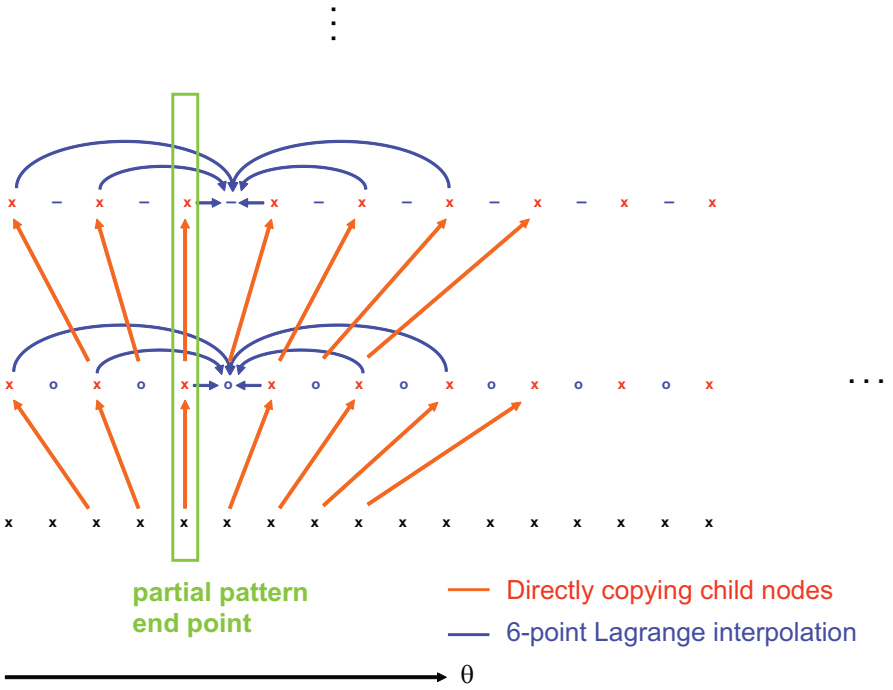


Figure 5.3: Interpolating cluster patterns near the end points.

For the end points corresponding to $\theta = 0$ or $\theta = \pi$, the samples corresponding to neighboring nodes on the unit sphere can be used since $(0 - \alpha, \phi) = (\alpha, \phi + \pi)$ and $(\pi + \alpha, \phi) = (\pi - \alpha, \phi + \pi)$ on a unit sphere.

Dividing the ranges of other dimensions will reduce the required memory at each aggregation step but will not significantly improve the memory efficiency. This is because the patterns of the bottom-level clusters dominate the memory requirement. On the other hand, aggregating the bottom-level clusters directly to the upper level without storing their patterns will reduce the required memory.

5.1 Results

5.1.1 Bistatic RCS of the Flamme Geometry

To demonstrate the efficiency and accuracy of the MLPO algorithm, bistatic RCS pattern of the scaled Flamme geometry shown in Figure 5.4 is computed for all

Table 5.1: Computation of the Flamme's bistatic RCS pattern: Growth of the number of triangles, θ samples, ϕ samples, and frequency samples as N increases

$N/2\pi$ (Target Size/λ in Frequency Range)	Number of Triangles	N_θ	N_ϕ	N_f
$[0, 1.5]$	628	49	101	17
$[0, 3]$	1604	97	201	33
$[0, 6]$	5200	193	401	65
$[0, 12]$	19288	385	801	129
$[0, 24]$	75634	769	1601	257
$[0, 48]$	300020	1537	3201	513

directions on the unit sphere. RCS values are evaluated for the frequency ranges given in Table 5.1, and CPU times are compared in Figure 5.8. For the sake of simplicity, first column of Table 5.1 is given as $N/2\pi$, which is the electrical size of the target in λ.

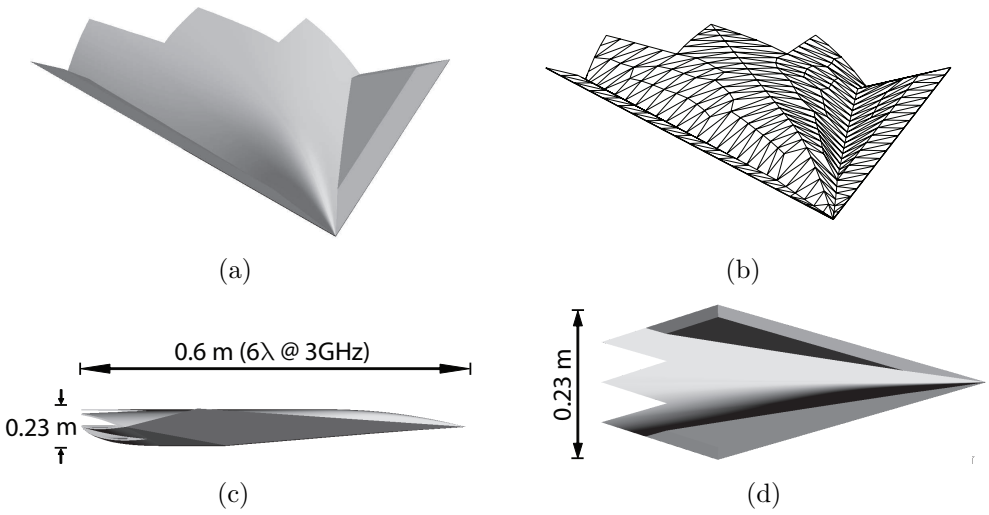


Figure 5.4: Geometry of the stealth Flamme target: (a) Front view. (b) A uniform mesh example. (c) Rear view. (d) Top view.

From the bistatic RCS results shown in Figs. 5.6 and 5.5, it is observed that, on the x-y, x-z, and y-z planes and in the $[0, 3]$ GHz frequency range, the MLPO algorithm and direct PO evaluation results are in excellent agreement. Fig. 5.7 presents the RCS error, $10 \log |RCS_{MLPO} - RCS_{PO}|$, which is lower than 1%.

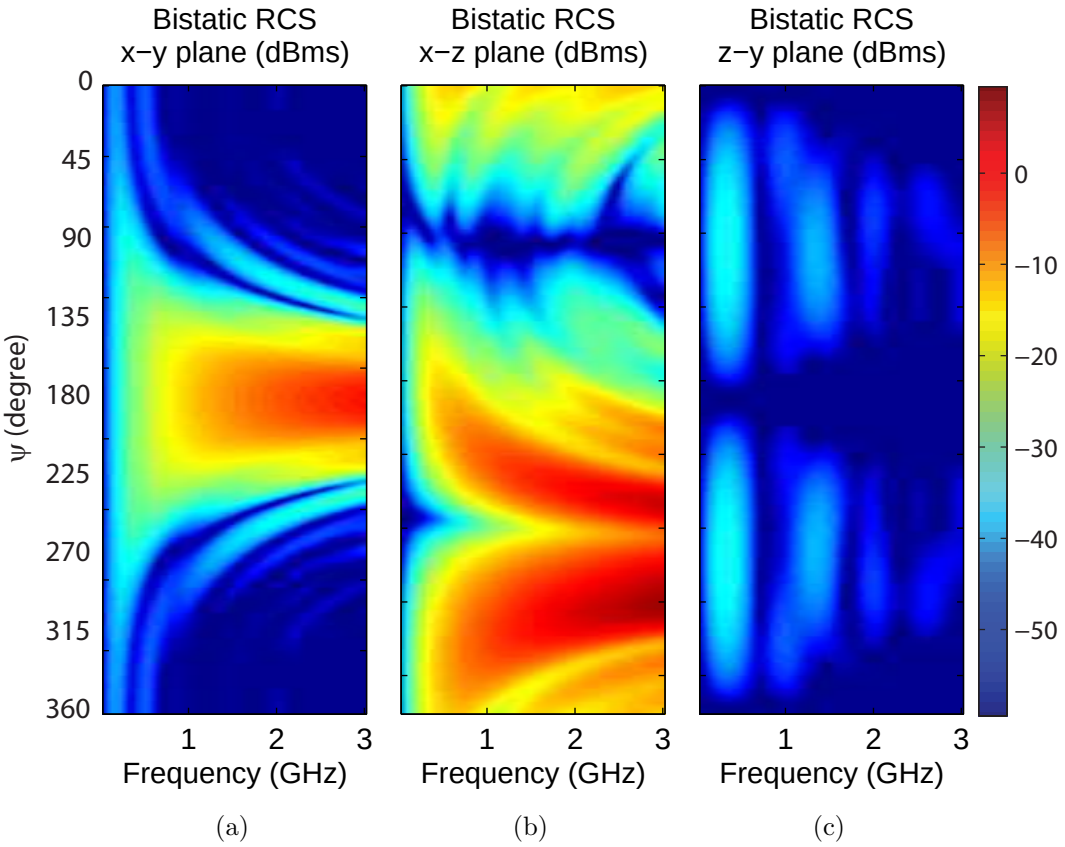


Figure 5.5: Bistatic RCS pattern of the Flamme geometry computed with direct PO evaluation: (a) x-y, (b) x-z, and (c) z-y planes.

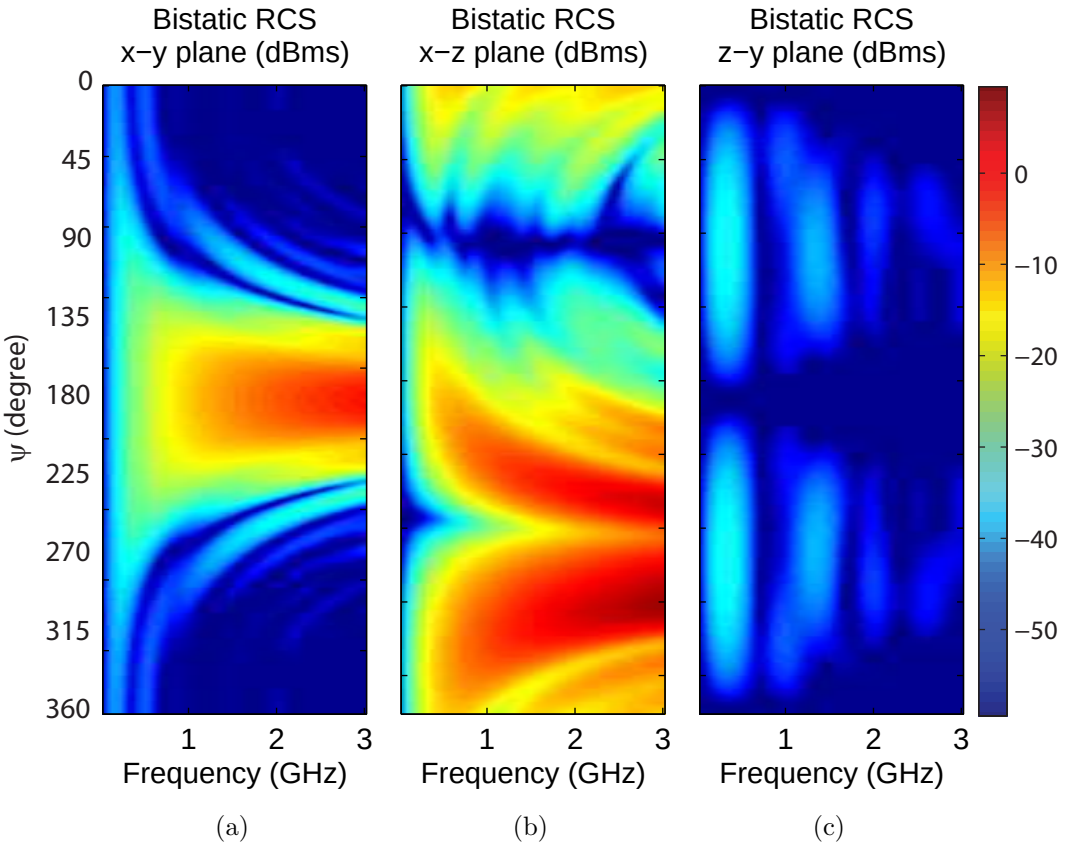


Figure 5.6: Bistatic RCS pattern of the Flamme geometry computed with MLPO algorithm: (a) x-y, (b) x-z, and (c) z-y planes.

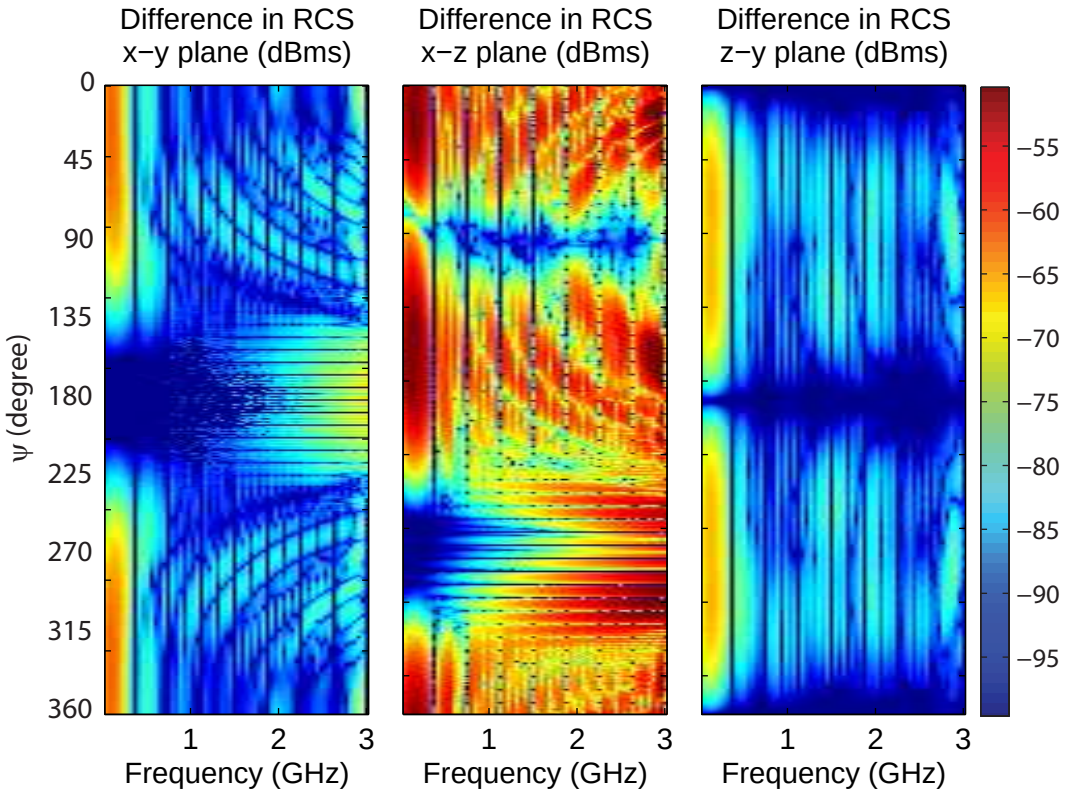
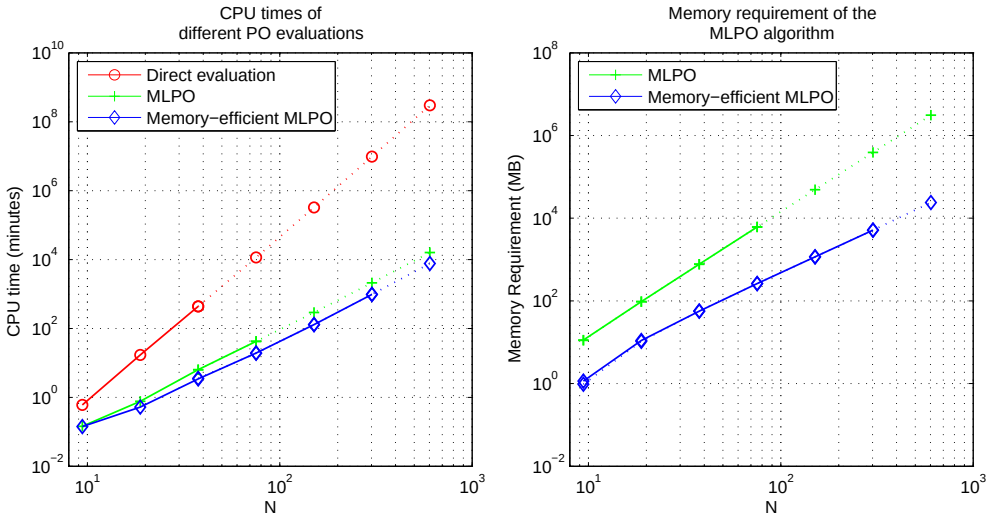


Figure 5.7: Absolute error of the MLPO algorithm: (a) x-y, (b) x-z, and (c) z-y planes.

Since the direct PO evaluation and MLPO algorithm results are very close to each other and therefore it is difficult to see the error by comparison, the absolute RCS difference, $\left|RCS^{PO} - RCS^{MLPO}\right|$ is also provided in Fig. 5.7.

From Table 5.1, it is observed that, as N increases, the number of triangles grows with $O\left(N^2\right)$ for large N. Numbers of θ, ϕ, and frequency samples (N_θ, N_ϕ, and N_f, respectively) increase with $O\left(N\right)$. Therefore, the total complexity of the direct PO evaluation turns out to be $O\left(N^5\right)$. This can be verified from the computation times presented in Fig. 5.8. Since both axes are in log scale, slopes of the curves indicate the complexity. For instance, $\log\left(N^5\right) = 5\log\left(N\right)$ and when plotted versus $\log\left(N\right)$, the curve is a straight line with slope 5. Similarly, $\log\left(N^3\log\left(N\right)\right) = 3\log\left(N\right) + \log\left(\log N\right) \approx 3\log\left(N\right)$ and when plotted versus $\log\left(N\right)$, the curve is approximately a straight line with slope 3.



(a) (b)

Figure 5.8: Efficiency of the MLPO algorithm: (a) CPU time. (b) Memory requirement. Dashed curves represent estimated values.

For the 48λ problem presented in Table 5.1 and Fig. 5.8, computing the scattering pattern with the memory-efficient MLPO algorithm takes approximately 53 hours. Computation time of the same problem with the direct PO integration is estimated to be 180,000 hours. Thus, a speed-up of nearly 3000 can be achieved with the MLPO algorithm. On the other hand, conventional MLPO algorithm would require 388 GB of memory, whereas the memory-efficient MLPO algorithm requires only 8 GB of memory, which is 77 times more efficient.

5.1.2 Backscattering RCS of the Flamme Geometry

MLPO algorithm can also be used to evaluate the backscattering RCS pattern in addition to the bistatic RCS pattern. To demonstrate the efficiency and accuracy of the MLPO algorithm, backscattering RCS pattern of the scaled Flamme geometry shown in Figure 5.4 is computed for all directions on the unit sphere for the frequency ranges given in Table 5.1. Note that as the backscattering and the bistatic RCS patterns of a target will have approximately similar spectral

content according to the target size, the number of θ, ϕ, and frequency samples are the same for both bistatic and backscattering cases.

From the backscattering RCS results shown in Fig. 5.9 and Fig. 5.10, as in the bistatic case, on the x-y, x-z, and y-z planes and in the $[0, 3]$ GHz frequency range, the MLPO algorithm and direct PO evaluation results are in excellent agreement. As in the bistatic case, the absolute error in the RCS is also presented in Fig. 5.11.

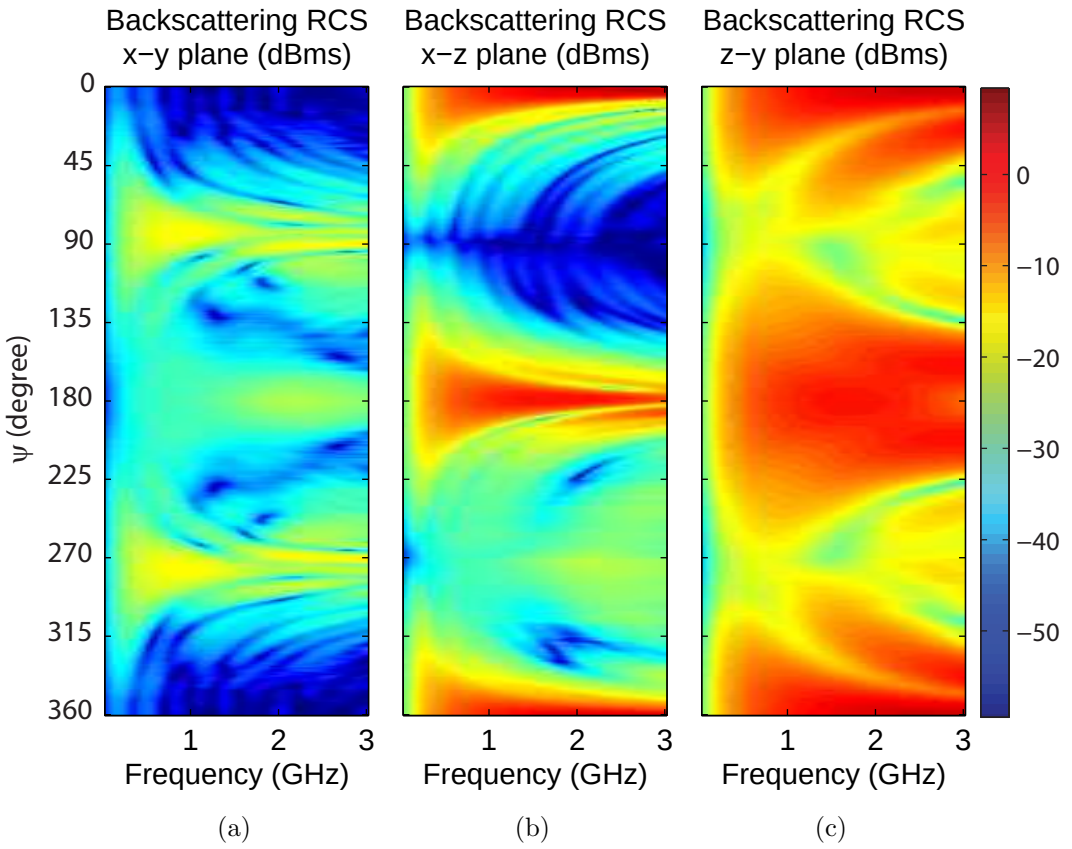


Figure 5.9: Backscattering RCS pattern of the Flamme geometry computed with direct PO evaluation: (a) x-y, (b) x-z, and (c) z-y planes.

From Fig. 5.12 and Fig. 5.8, it is observed that the memory requirement of the memory-efficient MLPO algorithm is exactly the same for the backscattering and bistatic cases. As the spectral contents of the scattered fields of the subdomains are very similar for the bistatic and backscattering cases, the sampling rates in θ, ϕ, and frequency are all the same for these two cases. As a result, since the

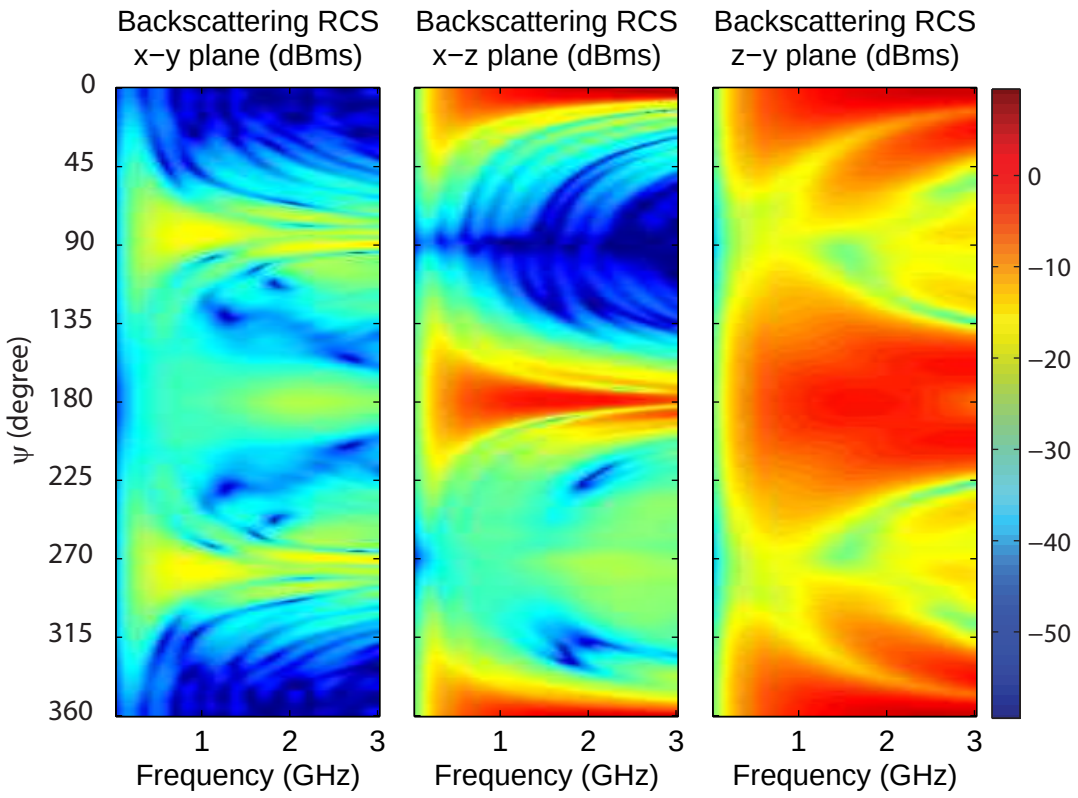


Figure 5.10: Backscattering RCS pattern of the Flamme geometry computed with MLPO algorithm: (a) x-y, (b) x-z, and (c) z-y planes.

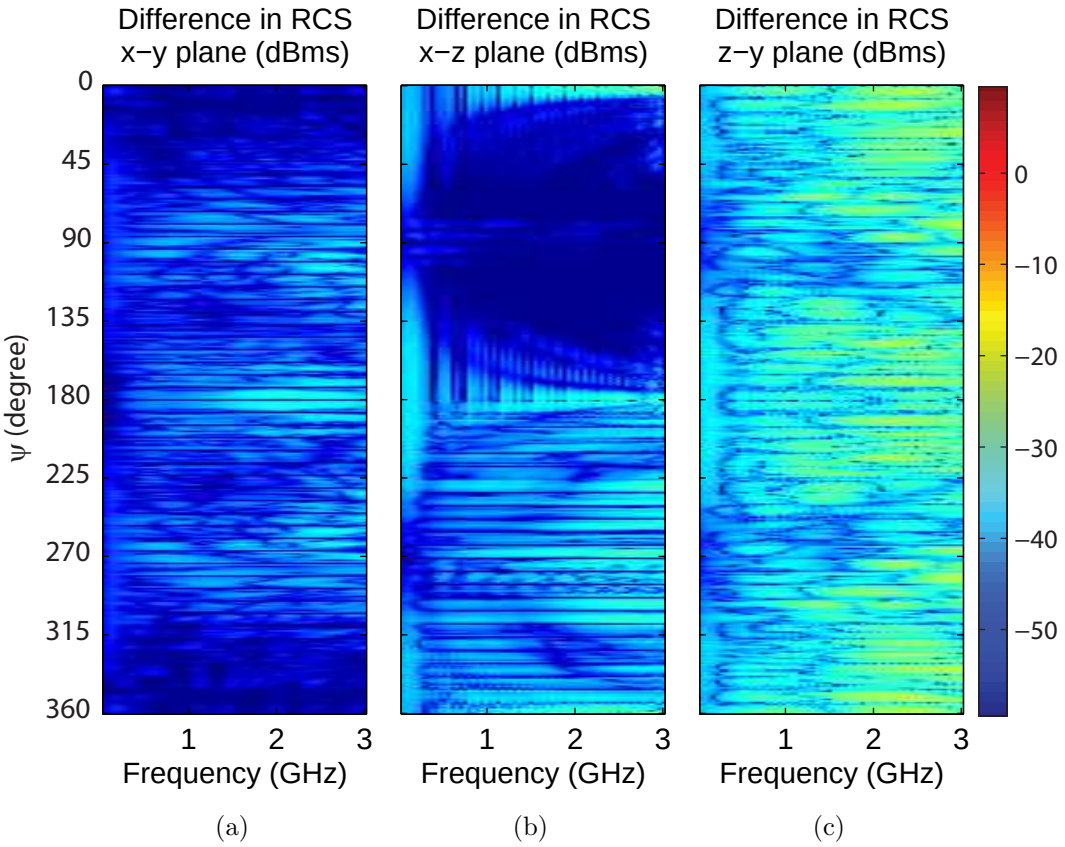
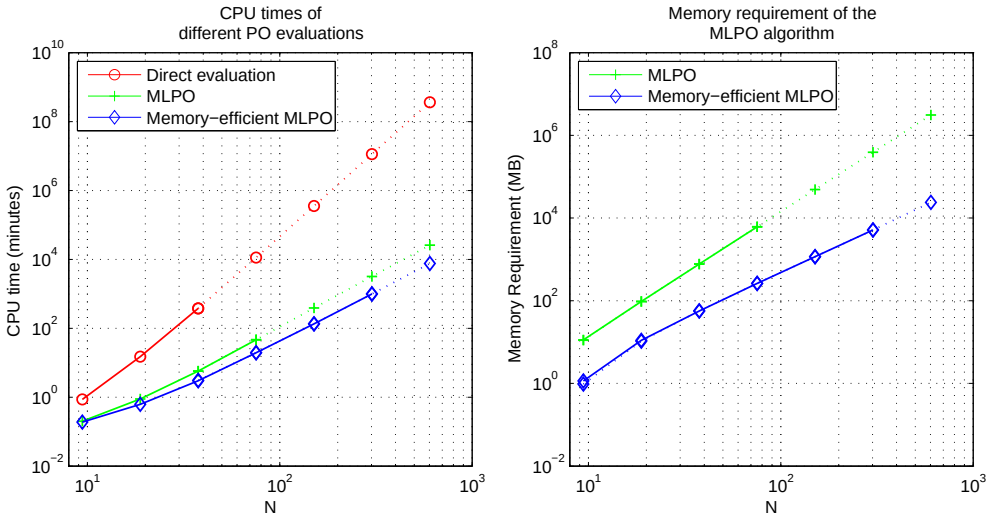


Figure 5.11: Absolute error of the MLPO algorithm: (a) x-y, (b) x-z, and (c) z-y planes.

domain decomposition is also the same for the bistatic and backscattering cases, memory required to store subdomain radiation patterns is also the same. In addition, it is also seen that the computation times of the bistatic and backscattering cases are also very close. This is because the only difference between the backscattering and bistatic cases is the computation of the illumination on a coarse grid of directions at the bottom level for the backscattering case. As the interpolations in the aggregations are the dominant factor determining the computational time, the bistatic and backscattering cases take the same amount of time as the number of aggregations is same in each case. As a result, the speed-up of 3000 in computational time and a gain of 77 in the required memory is also achievable for the backscattering case.



(a) (b)

Figure 5.12: Efficiency of the MLPO algorithm: (a) CPU time. (b) Memory requirement. Dashed curves represent estimated values.

As the operations done in the backscattering and the bistatic cases are very similar, the CPU times and memory requirements are also very similar in these two cases. On the other hand, for the backscattering case, the error is slightly higher compared to the bistatic case. This is because the illumination is computed only on the coarse grid of bottom level of the MLPO algorithm, when the scattering

patterns are computed. As the scattering patterns are aggregated, the illumination effect is also interpolated at each aggregation step. Whereas in the direct evaluation of the PO, all the illuminations are directly computed in the finer grid of illumination directions. Note that for the bistatic case, illumination is in one direction and it is not a function of θ or ϕ.

Chapter 6

CONCLUSIONS

In this thesis, we extend the 2-D MLPO algorithm proposed by Boag [6] to 3-D problems. We also propose two improvements on the MLPO algorithm. First improvement is the modification of the algorithm that enables the solution of the scattering problems involving nonuniform triangulations, thus decreasing the CPU time. The algorithm is modified so that the radiation patterns of the larger triangles in the nonuniform mesh are sampled according to their sizes and aggregated at the appropriate levels. Therefore, in addition to the speed-up of the MLPO algorithm, a speed-up of nearly $N_t/N_t^{'}$ can be achieved, while preventing the interpolation error. Here, N_t is the number of triangles in the uniform mesh and $N_t^{'}$ is the number of triangles in the nonuniform mesh. It is shown that for a generic helicopter model, a speed-up of 3.5 can be achieved. Although this speed-up may seem low, a greater speed-up can be achieved by increasing the deviation of the nonuniform mesh from the geometry and thereby decreasing the accuracy. Besides, as the target size may grow beyond thousands of wavelengths in real-life radar applications, uniform triangulations may require hundreds of millions of triangles. MLPO algorithm with nonuniform triangulation may become a good choice for such cases since even generating such triangulations for complex geometries may become impossible.

We also propose, develop, and demonstrate a novel memory-efficient version of the MLPO algorithm, in which the $O(N^3)$ memory requirement of the MLPO algorithm is decreased to $O(N^2 \log N)$. It is shown that, for a 48λ scattering problem, computation time of the scattering pattern as a function of θ, ϕ, and frequency with MLPO algorithm is only 53 hours, whereas the computation time of the same scattering pattern with direct PO evaluation is estimated to be 180,000 hours. Thus, a speed-up of 3000 can be achieved via the MLPO algorithm. In addition to this remarkable speed-up, it is shown that the maximum error in the RCS is less than 1%. On the other hand, this speed-up comes with a memory cost of 388 GB of memory. With the memory-efficient algorithm, this memory requirement can be reduced to 8 GB. Another advantage of the proposed memory-efficient algorithm is that the operations performed in the conventional and the memory-efficient algorithms are exactly the same. Therefore, the memory requirement is decreased while the accuracy and the computational efficiency are kept unchanged.

APPENDIX A

Lagrange Interpolation

A.1 One-Dimensional (1-D) Lagrange Interpolation

Lagrange interpolation polynomial, $L(\theta)$, is a polynomial of order $(n-1)$ that passes through a set of n data samples $(\theta_1, y_1), (\theta_2, y_2), ..., (\theta_n, y_n)$ [12]. The polynomial is constructed by a weighted sum of Lagrange basis polynomials $P_j(\theta)$ as

$$L(\theta) = \sum_{j=1}^{N_\theta} P_j(\theta), \tag{A.1}$$

where

$$P_j(\theta) = y_j \prod_{\substack{k=1 \\ k \neq j}}^{n} \frac{\theta - \theta_k}{\theta_j - \theta_k}. \tag{A.2}$$

Substituting Equation (A.2) and

$$w_j(\theta) = \prod_{\substack{k=1 \\ k \neq j}}^{n} \frac{\theta - \theta_k}{\theta_j - \theta_k} \tag{A.3}$$

in Equation (A.1) yields

$$L(\theta) = \sum_{j=1}^{N} w_j(\theta) y_j. \tag{A.4}$$

Thus, Lagrange interpolation can be viewed as the weighted sum of the data points, in which the weight of each sample is determined according to its distance from the interpolated value. If the distance between the samples is equal and the interpolated values are on a uniform grid with equal distances, then the weights w_j in Equation (A.4) can be calculated once and can be used for the entire interpolation. This property is useful in decreasing the CPU time of the MLPO algorithm presented in Chapter 3, since the interpolations are the dominant factor in the computation time. As a result, computational complexity of the lagrange interpolation for a data set of N_θ samples turns out to be $O(N_\theta)$.

A.2 Three-Dimensional (3-D) Lagrange Interpolation

For a 2-D function $y(\theta, \phi)$, Lagrange interpolation can be written as

$$L(\theta, \phi) = \sum_{j=1}^{n} w_j(\theta) \sum_{i=1}^{n} v_i(\phi)\, y(\theta_j, \phi_i), \tag{A.5}$$

where the weights w_j and v_i are defined as

$$w_j(\theta) = \prod_{\substack{k=1 \\ k \neq j}}^{n} \frac{\theta - \theta_k}{\theta_j - \theta_k}, \qquad v_i(\phi) = \prod_{\substack{l=1 \\ l \neq i}}^{n} \frac{\phi - \phi_l}{\phi_i - \phi_l}. \tag{A.6}$$

Equation (A.5) can be interpreted as "interpolation in one dimension, then interpolation in the other dimension" as illustrated in Fig. A.1.

By using this property, redundant calculation of the interpolated values can be omitted. For instance in Fig. A.1, V_2 can be calculated via 1-D interpolation using V_0 and V_1, with a weighted sum of two terms. On the other hand, 2-D interpolation would require the addition of four terms. As a result, the number of operations can be reduced by a factor 2 if the interpolated values are stored. In the MLPO algorithm, the interpolated values are already stored. Therefore,

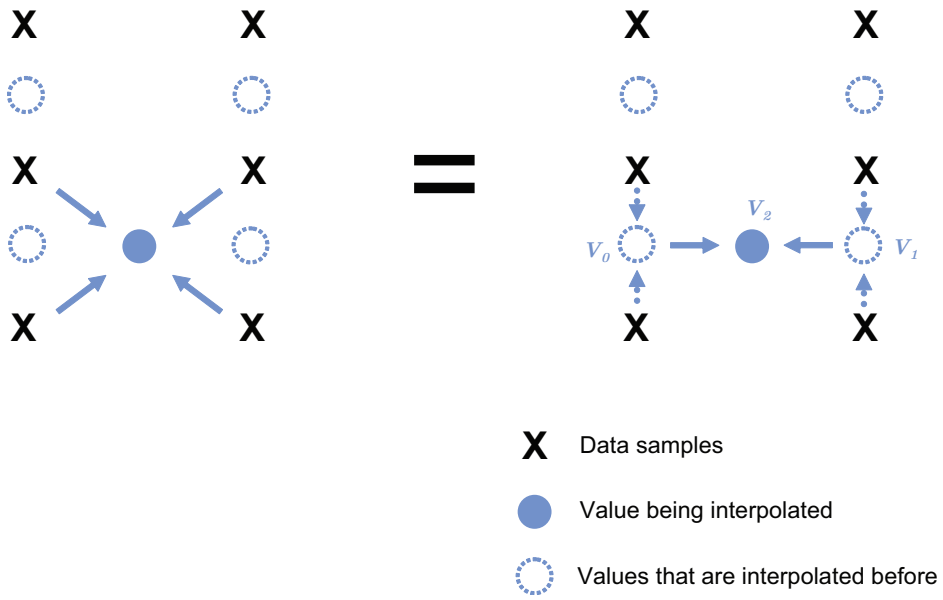


Figure A.1: Implementation of 2-D Lagrange interpolation with 1-D Lagrange interpolations. Since V_0 and V_1 are calculated previously, calculating V_2 requires summation of only two terms, whereas 2-D interpolation would require summation of four terms.

interpolation scheme illustrated in Fig. A.1 should be employed in the implementation.

3-D Lagrange interpolation is the straight-forward extension of Equation (A.5). For instance, for a 3-D function $y(\theta, \phi, f)$, Lagrange interpolation can be written as

$$L(\theta, \phi, f) = \sum_{j=1}^{n} w_j(\theta) \sum_{i=1}^{n} v_i(\phi) \sum_{h=1}^{n} z_h(f)\, y(\theta_j, \phi_i, f_h), \tag{A.7}$$

where the weights w_j and v_i are defined as

$$w_j(\theta) = \prod_{\substack{k=1 \\ k \neq j}}^{n} \frac{\theta - \theta_k}{\theta_j - \theta_k}, \qquad v_i(\phi) = \prod_{\substack{l=1 \\ l \neq i}}^{n} \frac{\phi - \phi_l}{\phi_i - \phi_l}, \qquad z_h(\phi) = \prod_{\substack{m=1 \\ m \neq h}}^{n} \frac{f - f_m}{f_h - f_m}. \tag{A.8}$$

The following pseudo-code describes an efficient way of implementing a 3-D Lagrange interpolation for a 3-D function $y(\theta, \phi, f)$ that is a function of θ, ϕ, and, frequency. Note that 1-D Lagrange interpolation is the building block of the algorithm:

for $j = 1,\ j \leq N_\theta,\ j = j + 2$ **do**

  for $i = 1,\ i \leq N_\phi,\ i = i + 2$ **do**

    1-D interpolation along frequency($y\,(\theta_j, \phi_i, :)$)

  end for

  for $h = 1,\ h \leq N_f,\ h = h + 1$ **do**

    1-D interpolation along phi($y\,(\theta_i, :, f_h)$)

  end for

end for

Now the 3-D space consists of filled ϕ-f planes. However, these planes have one empty space between them along the θ dimension. The following loop fills those empty planes.

for $i = 1,\ i \leq N_\phi,\ i = i + 1$ **do**

  for $h = 1,\ h \leq N_f,\ h = h + 1$ **do**

    1-D interpolation along theta($y\,(:, \phi_i, f_h)$)

  end for

end for

Here in the statement $j = 1,\ j \leq N_\theta,\ j = j + 2$, the term $j = 1$ initializes the running index j to 1. $j \leq N_\theta$ is the control condition that terminates the **for** loop when satisfied and $j = j+2$ term stands for incrementing the running index j by 2 at each step of the loop.

A.3 Lagrange Interpolation Near the End Points

In Chapter 2, it is shown that the accuracy of the interpolation can be enhanced by increasing the order of the interpolating polynomial, thus increasing the number of samples used in the interpolation. For the end points corresponding to

$\theta = 0$ or $\theta = \pi$, the samples corresponding to neighboring nodes on the unit sphere can be used since, $(0 - \alpha, \phi) = (\alpha, \phi + \pi)$ and $(\pi + \alpha, \phi) = (\pi - \alpha, \phi + \pi)$ on a unit sphere.

Unfortunately for a function of frequency, this approach is not possible. On the other, the hand first and second derivative information at the neighboring sample can be used, as illustrated in Fig. A.2. Since the PO integral is evaluated analytically on the triangles [8], evaluation of its first two derivatives at the end points has nearly no additional computational cost.

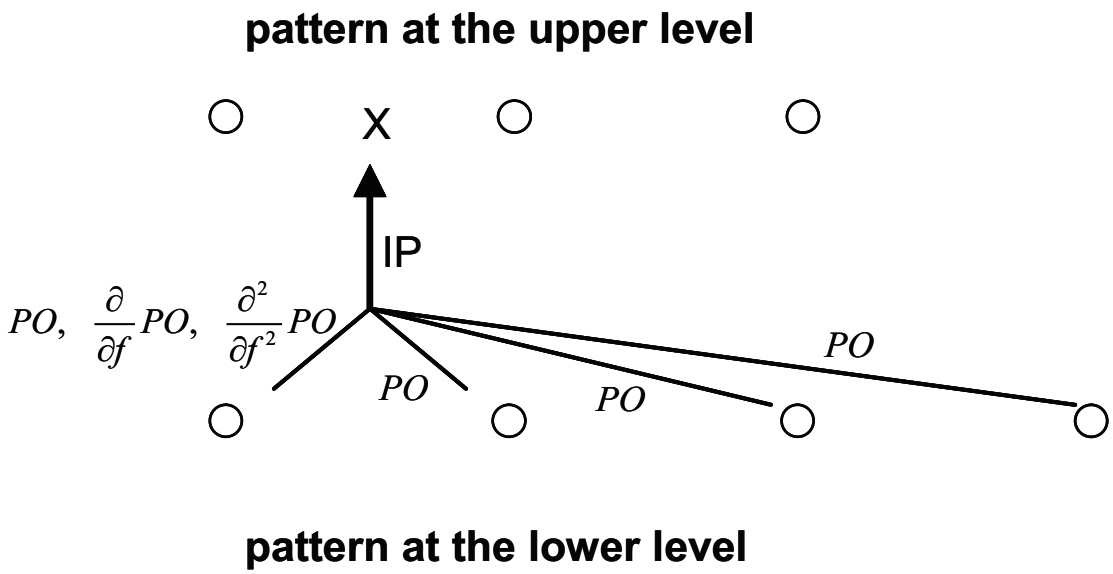


Figure A.2: Using the derivative information at the points that are close to the interpolated sample near the end points.

An interpolation using the derivative information can be implemented in a variety of ways. A modified polynomial interpolation can be used for simplicity. In this approach, some of the rows of the Vandermonde matrix that correspond to the unavailable nodes are replaced in order to satisfy the first two derivatives at the end points.

To interpolate the data points given by (f_i, y_i), $i = 1, 2, ..., n + 1$, where all f_i are different, an interpolating polynomial of degree n, expressed as $P(f) = C_n f^n + C_{n-1} f^{n-1} + ... + C_2 f^2 + C_1 f + C_0$, can be used. As $P(f)$ should satisfy

$P(f_i) = y_i$ for all data points, the linear system

$$\begin{bmatrix} f_1^n & f_1^{n-1} & \dots & f_1 & 1 \\ f_2^n & f_2^{n-1} & \dots & f_2 & 1 \\ \vdots & \vdots & & \vdots & \vdots \\ f_n^n & f_n^{n-1} & \dots & f_n & 1 \\ f_{n+1}^n & f_{n+1}^{n-1} & \dots & f_{n+1} & 1 \end{bmatrix} \begin{bmatrix} C_n \\ C_{n-1} \\ \vdots \\ C_1 \\ C_0 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \\ y_{n+1} \end{bmatrix} \tag{A.9}$$

should be solved for the coefficients C_i in order to construct the polynomial $P(f)$ [12]. Suppose that f_1 is the end point and the data is unavailable to the left of f_1. Imposing the derivative constraint at f_1 can be achieved by replacing the last two equations by the equations

$$\left.\frac{d}{df}P(f)\right|_{f=f_1} = \dot{y}_1 \tag{A.10}$$

and

$$\left.\frac{d^2}{df^2}P(f)\right|_{f=f_1} = \ddot{y}_1, \tag{A.11}$$

where $\dot{y}_1$ and $\ddot{y}_1$ are the first and second derivatives of the data at f_1. Then, the linear system of equations in Equation (A.9) becomes

$$\begin{bmatrix} f_1^n & f_1^{n-1} & \dots & f_1 & 1 \\ f_2^n & f_2^{n-1} & \dots & f_2 & 1 \\ \vdots & \vdots & & \vdots & \vdots \\ f_{n-1}^n & f_{n-1}^{n-1} & \dots & f_{n-1} & 1 \\ nf_1^{n-1} & (n-1)f_1^{n-2} & \dots & 1 & 0 \\ n(n-1)f_1^{n-2} & (n-1)(n-2)f_1^{n-3} & \dots & 0 & 0 \end{bmatrix} \begin{bmatrix} C_n \\ C_{n-1} \\ \vdots \\ C_2 \\ C_1 \\ C_0 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n-1} \\ \dot{y}_1 \\ \ddot{y}_1 \end{bmatrix} \tag{A.12}$$

Once Equation (A.12) is solved for the coefficients C_i, interpolated values at any point can be found by evaluating $P(f)$ at that point.

When applying this approach in the MLPO algorithm, first of all before the interpolation, the pattern of each subdomain is shifted to the origin at each aggregation step. As this shift is a multiplication by a factor of $e^{-jk\mathbf{r}_0\cdot(\hat{\mathbf{r}}-\hat{\mathbf{r}}_i)}$,

derivative of the each subdomain pattern should be modified according to

$$\frac{\partial}{\partial f}\left(Ue^{Af}\right) = \left(e^{Af}\frac{\partial}{\partial f} + Ae^{Af}\right)U, \tag{A.13}$$

where U is the subdomain pattern with respect to the global origin and $A = -jk\mathbf{r}_0 \cdot (\hat{\mathbf{r}} - \hat{\mathbf{r}}_i)$. Secondly, the matrix equation in Equation (A.12) should be solved in order to find the coefficients of the interpolating polynomial. Since the pattern of each subdomain is assumed to be sampled uniformly, coefficients C_i can be evaluated only once per aggregation step and can be used for all interpolations in that step. Lastly, data grid should be scaled in order to prevent the matrix entries in Equation (A.12) from growing too large. For instance, if the frequency range of interest is in the order of GHz, the Vandermonde matrix given in Equation (A.12) will be too difficult to invert since its entries will contain powers of 10^9 in addition to 1's 0's. The pattern derivatives should also be modified according to the chain rule after the scaling of the grid.

APPENDIX B

Accuracy of the PO Approximation

It is known that PO approximation is usually suitable for the computation of scattered fields from smooth, convex, and electrically-large targets. On the other hand, the error of the PO approximation cannot be determined precisely for a given problem. Therefore, most common way of verifying the PO results is the comparison with the analytical solutions of canonical geometries, such as sphere [13]. In Fig. B.1, the PO and analytical solutions for the backscattering RCS of a sphere with radius a are given. Note that, as the electrical dimensions of the sphere increase, PO approximation becomes more accurate. . In real-life scattering problems with complex geometries, the analytical solution is usually unavailable. Therefore, comparison with the measurements [4] or comparison with the solutions of the integral-equation methods [3] for low frequencies are common ways of verifying the PO results [14]. Since the integral-equation methods usually require more computational resources as the electrical size of the target increases, the solutions are usually compared for low and moderately high frequencies. Since the PO is expected to give better results as the frequency increases, such comparisons help predicting the error bound for higher frequencies.

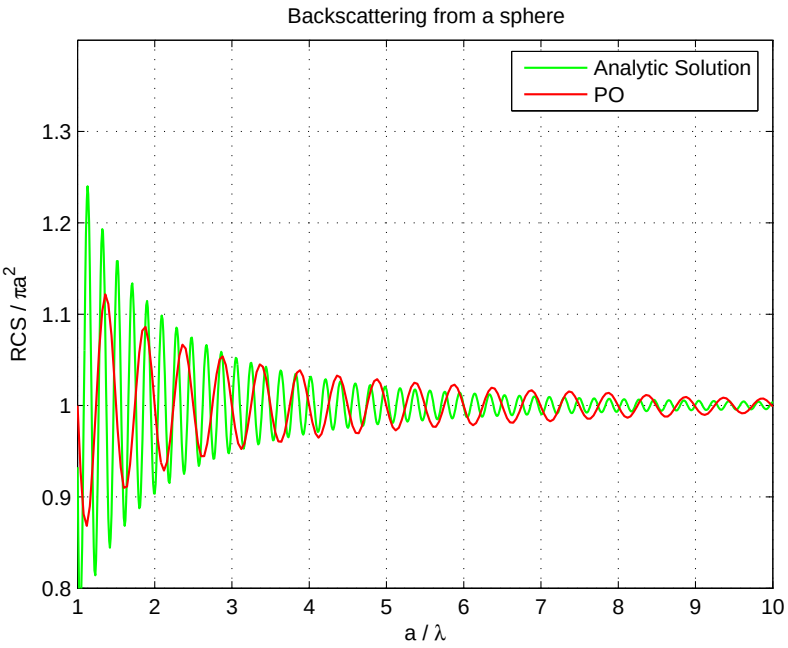


Figure B.1: PO and analytical solutions for the backscattering RCS of a sphere with radius a.

In an other example, PO results for the bistatic scattered field from the rectangular cube shown in Fig. B.2 are compared with multilevel fast multipole algorithm (MLFMA) [5] results in Figs. B.3 and B.4. It is observed the PO and MLFMA results are in good agrement in the specular reflection directions $\psi = 90°$ and $\psi = 90°$. In the forward-scattering direction $\psi = 180°$, the results are also in good agrement. On the other hand, for the backscattering case $\psi = 0°$, the results are further apart. This can be explained by the fact that total scattering is dominated by the scattering from the illuminated edge of the prism and the PO currents are known to be less accurate on the edges.

.

.

.

In order to verify the accuracy of the PO approximation for a more complex geometry, PO and MLFMA results for the bistatic scattering from the helicopter

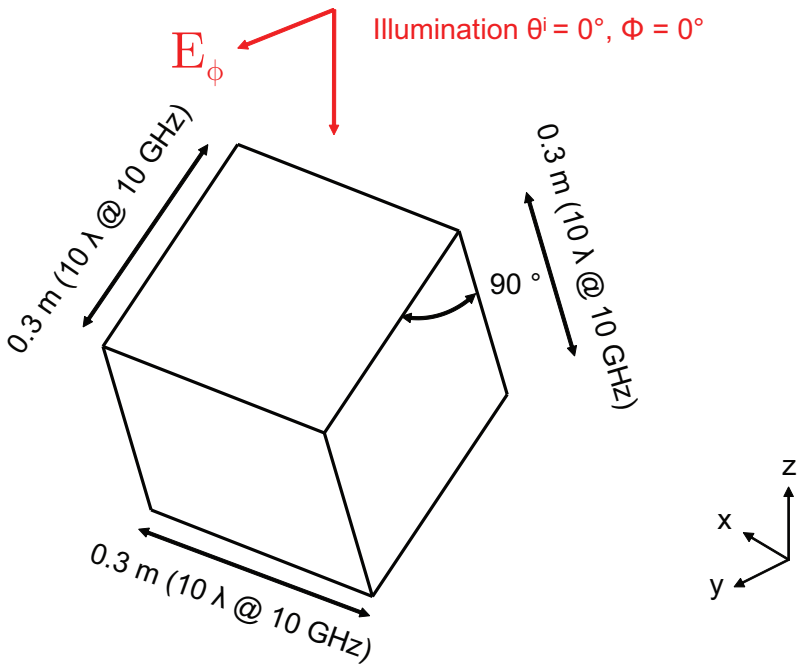


Figure B.2: Rectangular prism.

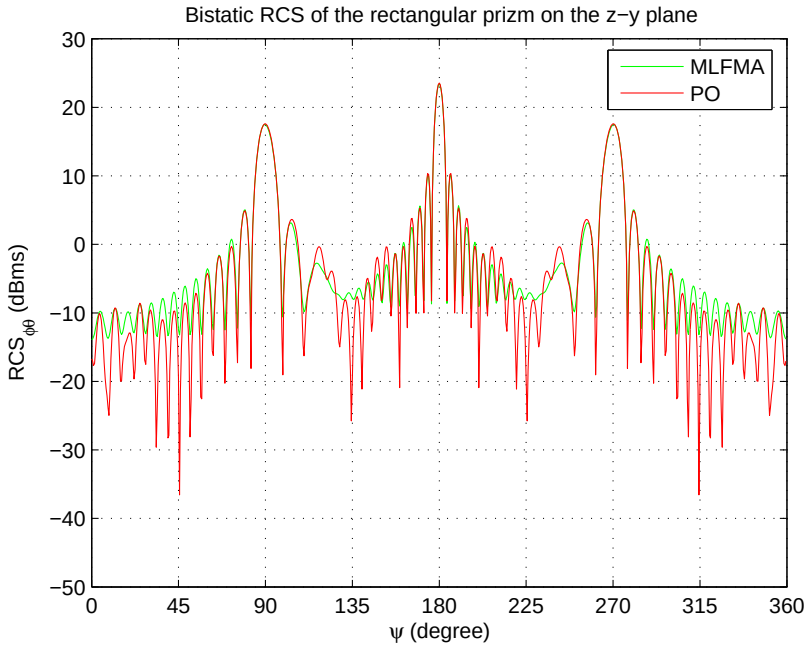


Figure B.3: Bistatic scattering from the rectangular prism on the z-y plane.

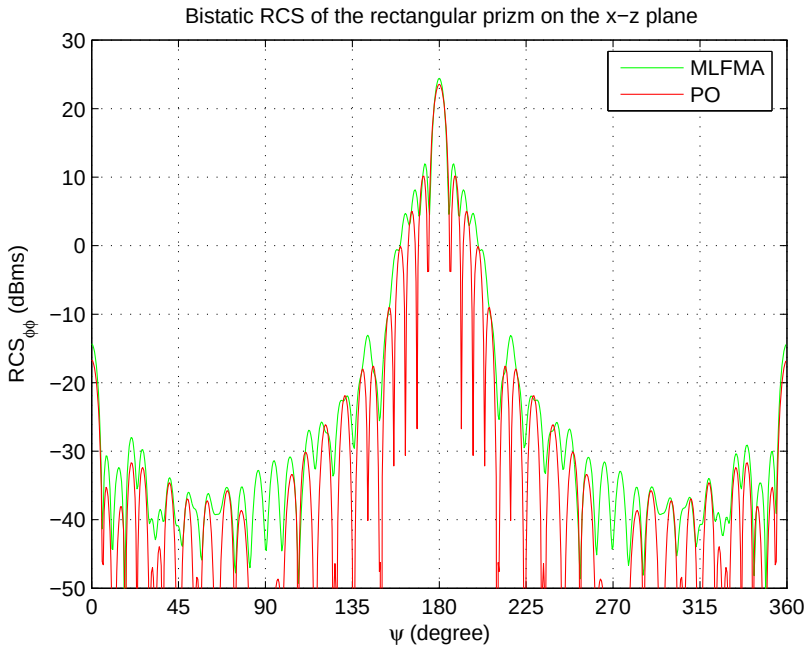


Figure B.4: Bistatic scattering from the rectangular prism on the x-z plane.

model shown in Fig. B.5 are compared in Fig. B.6. It is seen that PO is more accurate for the backscattering $\psi = 270°$ and forward-scattering $\psi = 90°$ directions and less accurate in the other directions.

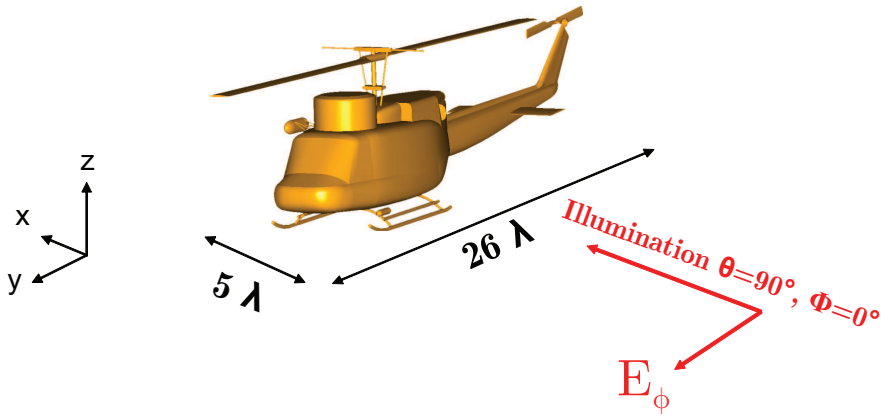


Figure B.5: Helicopter model.

.

.

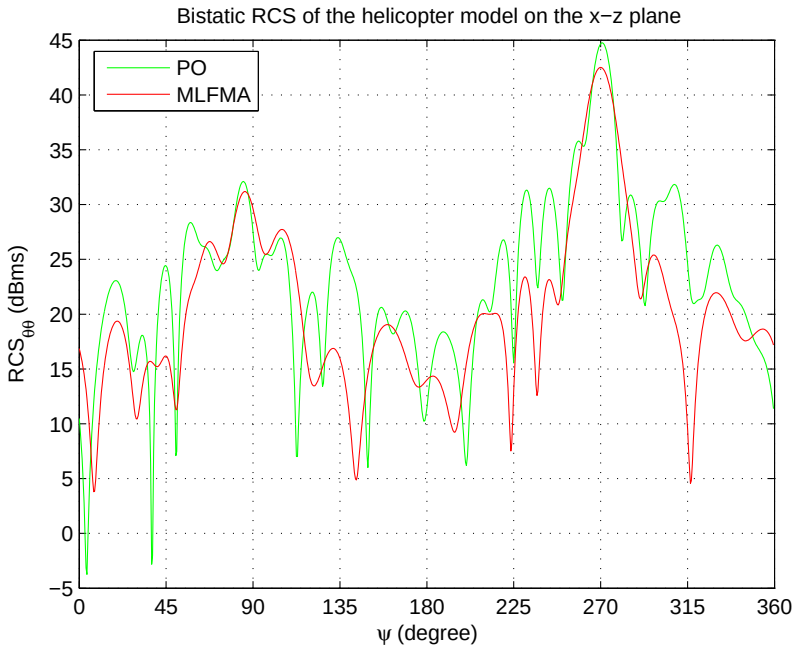


Figure B.6: Bistatic scattering from the helicopter model on the x-z plane.

Bibliography

[1] R. F. Harrington, *Time Harmonic Electromagnetic Fields*, McGraw-Hill, New York, 1961.

[2] P. Ya Ufimtsev, "Method of Edge Waves in the Physical Theory of Diffraction", (from the Russion "Method Krayevykh Voin V Fizichenskoy Theorii Diffraktsii", *Izd-Vo Sov. Radio, pp.1-243, 1962.)*, Translation prepared by the U.S. Air Force Foreign Technology Division, Wright-patterson AFB, OH, Sept. 1971.

[3] R. F. Harrington, *Field Computation by Moment Methods*. IEEE Press, 1993.

[4] R. Coifman, V. Rokhlin, and S. Wandzura, "The fast multipole method for the wave equation: a pedestrian prescription," *IEEE Ant. Propag. Mag.*, vol. 35, no. 3, pp. 7-12, June 1993.

[5] C.-C. Lu and W. C. Chew, "Multilevel fast multipole algorithm for electromagnetic scattering by large complex objects," *IEEE Trans. Antennas Propagat.*, vol. 45, no. 10, pp. 1488-1493, Oct. 1997.

[6] A. Boag, "A fast physical optics algorithm for high frequency scattering," *IEEE Trans. Antennas Propagat.*, vol. 52, pp. 197-204, Jan. 2004.

[7] S. Sefi and J. Oppelstrup, "Physical optics and NURBS for RCS calculations," *Proceedings of EMB04 Computational Electromagnetics*, pp. 90-97, Chalmers Univ. of Tech., Sweden, Oct. 2004.

[8] W. B. Gordon, "Far-field approximations to the Kirchhoff-Helmholtz representation of scattered fields," *IEEE Trans. Antennas Propagat.*, vol. 23, pp. 590-592, July 1975.

[9] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions*. New York: Dover, 1965.

[10] Alan V. Oppenheim, Alan S. Willsky, and S. Hamid Nawab, *Signals and Systems*. Prentice Hall, 1996.

[11] A. Manyas and L. Gürel, "Multilevel PO algorithm for non-uniform triangulations," *European Conference on Antennas and Propagation (EuCAP 2006)*, Nice, France, Nov. 2006.

[12] Micheal T. Heath, *Scientific Computing An Introductory Survey*, McGraw-Hill, January, 1996.

[13] Constantine A. Balanis, *Advanced Engineering Electromagnetics*, Wiley, May, 1998.

[14] L. Gürel, H. Bağcı, J. C. Castelli, A. Cheraly, and F. Tardivel, "Validation through comparison: measurement and calculation of the bistatic radar cross section (BRCS) of a stealth target," *Radio Science*, vol. 38, no. 3, pp. 12-1–12-10, June 2003.

Glossary

dbms : "Decibel meter square.

$\mathbf{E}(\cdot)$: Electric field intensity.

$\mathbf{E}^{inc}(\cdot)$: Incident electric field intensity.

$\mathbf{E}^{scat}(\cdot)$: Scattered electric field intensity.

$E[\mathbf{r}_0]$: Operator that shifts the target by $-\mathbf{r}_0$.

$F\{\cdot\}$: Fourier transform.

$\mathbf{H}(\cdot)$: Magnetic field intensity.

$\mathbf{H}^{inc}(\cdot)$: Incident magnetic field intensity.

$\mathbf{H}^{scat}(\cdot)$: Scattered magnetic field intensity.

$\mathbf{I}^{N_\theta N_\phi N_f}_{\bar{N}_\theta \bar{N}_\phi \bar{N}_f}$: Interpolation matrix that increases the number of samples from $\bar{N}_\theta \times \bar{N}_\phi \times \bar{N}_f$ points to $N_\theta \times N_\phi \times N_f$ points.

$J_n(\cdot)$: Bessel function of order n.

$\mathbf{J}_s(\cdot)$: Electric surface current density.

k : Wavenumber.

$L(\cdot)$: Lagrange interpolation polynomial.

λ : Wavelength.

N : kR.

η : Intrinsic impedance.

N_f : Number of frequency samples.

N_ϕ : Number of ϕ samples.

N_t : Number of triangles in the uniform mesh.

N_t' : Number of triangles in the nonuniform mesh.

N_θ : Number of θ samples.

$O\left[\mathbf{r}_0\right]$: Operator that shifts the target by $\mathbf{r}_0$.

Ω_f : Oversampling ratio in frequency.

Ω_ϕ : Oversampling ratio in ϕ.

Ω_θ : Oversampling ratio in θ.

$P\left(\cdot\right)$: Lagrange basis polynomial.

Ψ : PO operator.

$\hat{\mathbf{r}}$: Observation unit vector.

$\mathbf{r}'$: Source point.

$\mathbf{r}_0$: Shift vector.

RCS : Radar cross section defined as $4\pi \left| \frac{\mathbf{E}^{scat}}{\mathbf{E}^{inc}} \right|^2$.

$\hat{\mathbf{r}}_i$: Incidence unit vector.

$\mathbf{r}_q$: Center of the smallest sphere that can contain the qth subsurface.

S : Surface of the object.

S_{lit} : Lit regions of S.