EXTRAPOLATION WITH THE MATRIX PENCIL METHOD TO COMBINE LOW-FREQUENCY AND HIGH-FREQUENCY ELECTROMAGNETIC SCATTERING RESULTS

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND

ELECTRONICS ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Ahmet Ferhat Yıldırım

August 2005

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.

Assist. Prof. Dr. Vakur B. Ertürk

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. İbrahim Körpeoğlu

Approved for the Institute of Engineering and Science:

Prof. Dr. Mehmet Baray
Director of the Institute Engineering and Science

to my family, for their endless support;
to my friends, the only treasure I have ever had.

ABSTRACT

EXTRAPOLATION WITH THE MATRIX PENCIL METHOD TO COMBINE LOW-FREQUENCY AND HIGH-FREQUENCY ELECTROMAGNETIC SCATTERING RESULTS

Ahmet Ferhat Yıldırım
M.S. in Electrical and Electronics Engineering
Supervisor: Prof. Dr. Levent Gürel
August 2005

Accurate frequency domain solutions of electromagnetic scattering problems are known to require high computing resources as the solution frequency increases. On the other hand, high-frequency techniques provide us solutions with limited accuracy in the relatively high-frequency regions. In this thesis we aimed to fill the intermediate gap by using extrapolation techniques. Matrix pencil method (MPM) is presented to find the parameters of the model in our model-based extrapolation approach. In order to fully incorporate the two separate available data sources, i.e., accurate low-frequency solvers and asymptotic high-frequency solvers, we proposed two methods of coupling, namely coupled MPM and coupled deconvolution MPM. Results of proposed extrapolation methods are tested both on analytically generated backscattering solution of conducting sphere and numerical solutions of various three dimensional bodies.

Keywords: Extrapolation, Matrix pencil method, Coupled extrapolation, RCS, COMPM, CDMPM.

ÖZET

YÜKSEK VE ALÇAK ELEKTROMANYETİK SAÇINIM SONUÇLARININ MATRİS KALEM YÖNTEMİ İLE BİRLEŞTİRİLEREK DIŞDEĞERLEMESİ

Ahmet Ferhat Yıldırım
Elektrik ve Elektronik Mühendisliği, Yüksek Lisans
Tez Yöneticisi: Prof. Dr. Levent Gürel
Ağustos 2005

Elektromanyetik saçınım problemlerinin frekansa bağlı yüksek doğruluktaki çözümleri için frekans arttıkça çok fazla işlemci zamanı ve bellek tükettikleri bilinmektedir. Bunun yanı sıra, çok yüksek frekanslarda sonuşur yüksek frekans çözüm yöntemleri ise kısıtlı doğruluğa sahip çözümler sunmaktadır. Bu tezde, ara frekans bölgesindeki saçınım değerlerinin dışdeğerleme yöntemi ile yaklaşımsal olarak bulunması amaçlanmıştır. Matris kalem yöntemi (MKY), modele dayalı dışdeğerleme yaklaşımında model parametrelerinin bulunması için kullanılmıştır. Mevcut alçak ve yüksek frekans çözücülerinden elde edilen çözümleri tam olarak birleştirebilmek için iki bağlaşim yöntemi öneriyoruz: "bağlaşımlı MKY" ve "bağlaşımlı ters evrişim MKY." Önerilen bu dışdeğerleme yöntemleri hem iletken kürenin analitik olarak üretilmiş geri saçınım çözümlerinde, hem de çeşitli üç boyutlu nesnelerin sayısal olarak elde edilmiş çözümlerinde denenmiştir.

Anahtar sözcükler: Dışdeğerleme, Matris kalem yöntemi, Bağlaşımlı dışdeğerleme, RKA.

ACKNOWLEDGEMENTS

To begin with, I would like to express my sincere gratitude to my supervisor, Prof. Levent Gürel, for his guidance, motivation, understanding and support.

I also would like to thank to the members of my thesis committee for their valuable comments and inputs.

I also thank members of the Bilkent Computational Electromagnetics Laboratory for their cooperation and support; especially to Özgür and Alp for providing the computational data, and Ali Rıza for making the solutions possible.

I am also grateful to Anirudh for proof reading and commenting on this thesis.

Many thanks to Ayça for her support and help during the redaction of this thesis.

Special thanks to my family for their relentless encouragement and moral support during my endeavors.

Last but not least, it is an obligation for me to thank my friends for their support and presence.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Electromagnetic Problems and Their Solutions

Scattering solutions of arbitrarily shaped three-dimensional (3-D) conducting bodies have always been of interest in computational electromagnetics (CEM). There are various applications in which these solutions play crucial roles, such as target identification, military aircraft design, and antenna design. Often, it is important to have the full knowledge of the scatterer's spectral properties. Accurate solvers that calculates the scattering solutions for these problems exist, but these solutions demand more memory and CPU time as the solution frequency increases. These type of solvers perform the same solution procedure for every other solution frequency, but as the frequency increases, resource demands of the solution procedure increase. Therefore, it is usually impossible to obtain the scattering solution on the whole spectrum using these solvers.

On the other hand, as the solution frequency goes to relatively high frequencies, the electromagnetic waves start to act under the laws of optics. Using this approximation, it is possible to simplify the solution procedures involving complex interactions and dense matrix operations to algorithms requiring the evaluation of much simpler expressions. This provides us with a new class of solvers, namely, high-frequency (HF) solvers, which can produce scattering solutions very rapidly. The only drawback of these solvers is their reduced accuracy, which is related to the HF approximations of electromagnetics.

A common approach for accurately solving arbitrarily shaped 3-D bodies is using method of moments (MOM). In order to solve for the scattered field when the target body is illuminated with a plane wave, we first need to discretize the surface of the geometry using small triangles, which is called meshing. A large matrix is filled with the electromagnetic interaction values of these triangular elements. The excitation information is used to fill the right-hand-side vector. The solution of the ensuring matrix equation is the coefficients of the surface current induced on the scatterer. The surface current is then used to calculate the scattered field at any point in space. In order to solve accurately enough, dimensions of each triangle should not be larger than $\lambda/10$, where λ corresponds to the wavelength at the specific solution frequency. With a closer look at this procedure, it can be seen that, when the solution frequency increases, i.e., electromagnetic size of the scatterer increases or wavelength decreases, the size of the interaction matrix also increases; hence the solution of that system becomes prohibitively difficult.

Various approaches have been developed such as the fast multipole method (FMM) and its multi-level version, namely, the multi-level fast multiple algorithm (MLFMA), to increase the solution efficiency of the matrix system. A good overview and detailed algorithms for these methods can be found in [1]. A recent trend in CEM is to solve these high-resource demanding problems in parallel computing environments. These improvements to MOM are quite effective in increasing the maximum solvable frequency limit, nevertheless, still a limit exists, albeit higher.

On the other end of the spectrum, with the help of the HF approximations of the electromagnetics, HF solvers do not deal with much of the burden that the low-frequency (LF) solvers face. We use a popular HF solution algorithm, namely, physical optics (PO), to find the scattering solution at relatively high frequencies. Once again, as in the MOM based solvers, the surface of the scatterer is discretized using small triangles. Since the electromagnetic waves can be assumed to obey the laws of optics at these frequencies, each triangle on the surface of the scatterer is marked as lit or dark, depending on whether it is directly (or indirectly, by multiple reflections) illuminated from the source. Scattered field is then calculated by evaluating a "PO integral" on all lit triangles. To sum up, as opposed to MOM-based solvers, in PO, there is no interaction between the triangles, no dense matrix formation, and no solution of that dense system. Therefore, PO algorithm is extremely fast and can produce scattering solutions practically for every frequency. Because the accuracy of the HF approximations tends to decrease as the frequency decreases, the accuracy of the PO solution also decreases, and thus the solution results at low frequencies is nothing useful.

Even at relatively high frequencies, PO may not produce very accurate results. This is mainly due to the fundamental approximations employed in the HF electromagnetics. In order to correct the solution that is obtained by PO, various correction strategies have been developed. In this thesis, we will use edge/wedge current corrected PO data (POE/POW). In the classical PO theory, scattering and diffraction from wedges and edges are ignored. POE and POW correct this by taking into account the scattering from the induced currents on edges and wedges; however, the inaccuracy problem of the overall solution for low frequencies is still intact.

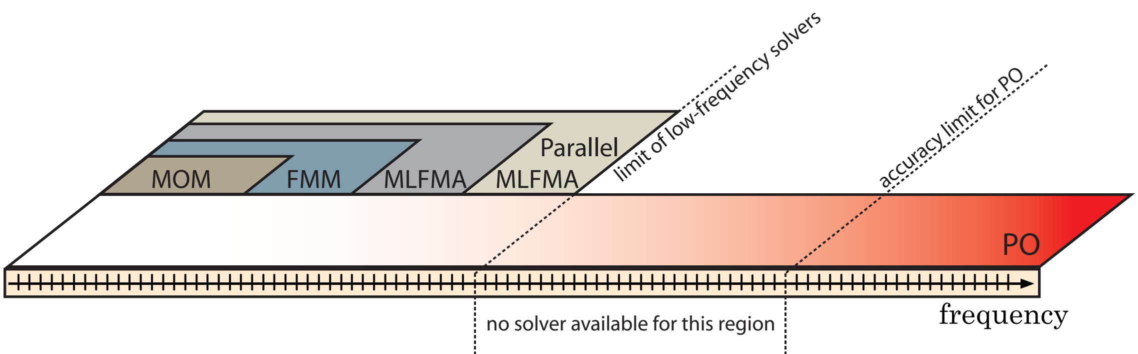


Figure 1.1: Comparison of numerical solvers over the solution spectrum.

Often, solution intervals of LF and HF solvers do not overlap. Figure 1.1

summarizes the general distribution of the available solvers over the solution spectrum and the gap between them that stimulates the research presented in this thesis. In order to obtain the full spectral behavior of a scattering body, we propose using extrapolation techniques to estimate the scattering spectrum for intermediate frequencies. In other words, the purpose of this research and this thesis is to bridge the gap between the LF and HF electromagnetic solvers.

1.2 Model-Based Extrapolation

In order to estimate the solution in the intermediate frequency band, we will use the knowledge of the data at LF and HF, which are obtained from corresponding electromagnetic solvers. Using the known data, an appropriate model will be constructed, and extrapolation will be performed by evaluating that model at the frequencies for which the scattering solution is not available. This approach is called model-based extrapolation [2]-[5]. Instead of the widely used rational-function based modeling, we will use complex-exponential based model as they are more consistent with the characteristics of the scattering signals. The model we will use is basically the sum of weighted complex exponentials,

$$y\left(f\right) = \sum_{i=1}^{M} R_i e^{s_i f}. \tag{1.1}$$

We assume that values of the solution signal $y(f)$ are specified at N equally spaced points, $f = 0, 1, 2, ..., N-1$ [6], and expand the summation in (1.1) to obtain N equations.

$$\begin{aligned}
y\left(0\right) &= R_1 + R_2 + \cdots + R_M, \\
y\left(1\right) &= R_1 e^{s_1} + R_2 e^{s_2} + \cdots + R_M e^{s_M}, \\
y\left(2\right) &= R_1 e^{2s_1} + R_2 e^{2s_2} + \cdots + R_M e^{2s_M}, \\
&\vdots \\
y\left(N-1\right) &= R_1 e^{(N-1)s_1} + R_2 e^{(N-1)s_2} + \cdots + R_M e^{(N-1)s_M}.
\end{aligned} \tag{1.2}$$

This system of N equations defines the projection of the solution signal $y(f)$ onto an M dimensional space that is spanned by complex exponential basis. If we happen to know the exponent set, $\{s_i\}$, the set of equations in (1.2) becomes a linear system, whose solution is straightforward. However, we also seek the complex exponential basis along with the residual set, $\{R_i\}$. Simultaneous solution of both parameters from the samples of the solution signal $y(f)$ is a nonlinear problem but in most cases linearizing the nonlinear problem gives an equivalent result. The nonlinear problem is discussed in [7]-[9]. There are two popular linear approaches for the solution of (1.1), namely, polynomial method and matrix pencil method (MPM) [10]. The basic difference between the two methods is that the former one is a two-step process, whereas, the latter one is a one-step process. In addition to this computational inefficiency of the polynomial method, it has low tolerance to noise. The comparison of these two methods, their historical perspectives, their computational and statistical efficiencies are discussed by various authors [10],[11]. In this work, due to its noise tolerance and computational efficiency, we use MPM to find the parameters of (1.1). MPM is explained in the next chapter in detail.

After finding the model parameters through either of the methods mentioned above by using the knowledge of y values in the interval $f = (0 : f_1)$, extrapolation is performed by evaluating (1.1) for values f greater than f_1. However, this approach will under utilize our knowledge of data, as we will be using only some fraction of the available data. Thus, a new extrapolation scheme will be introduced in the third chapter.

In order to test all the extrapolation schemes we used the analytical solution of a conducting sphere. As we have the analytical solution available, we can obtain the scattering data anywhere on the spectrum. This will enable us to compare the extrapolated signal with the actual signal, and thus make some modifications on the proposed extrapolation strategies. During the progress of this research analytical solver for conducting sphere was studied and implemented, but as it is beyond the scope of this thesis, those topics will be skipped. Interested readers are referred to [12] for the theory of the analytical solution of conducting sphere.

The actual scatterers to be solved are of course not limited to conducting sphere. Results using computationally obtained data are demonstrated in Chapter 5; but before that we will present problems and possible solutions that arise when working with these computational data in Chapter 4.

Chapter 2

Matrix Pencil Method (MPM)

2.1 MPM Theory

Due to the nature of the electromagnetic solvers, we have samples of the scattering solution signal at discrete frequencies. Therefore, we need to discretize the model (1.1) as

$$y\left[k\right] = \sum_{i=1}^{M} R_i z_i^k \qquad k = 0, \cdots, N-1, \tag{2.1}$$

where we used the discretization

$$y\left[k\right] = y\left(kF_s\right) \tag{2.2}$$

and

$$\begin{aligned} z_i^k &= e^{s_i F_s k} \\ &= e^{(\alpha_i + j\omega_i) F_s k}. \end{aligned} \tag{2.3}$$

In (2.2) and (2.3), F_s represents the sampling interval. In (2.1) the complex exponential set $\{z_i\}$ and their corresponding residual set $\{R_i\}$ are the unknowns to be determined along with the number of exponentials that will be used when constructing the model, i.e., the number M. This parameter M is very important when performing extrapolation using the model and will be discussed thoroughly in the next section. The method uses a mathematical tool called matrix pencil,

$$\bar{\mathbf{Y}}_{\mathbf{2}} - \lambda\bar{\mathbf{Y}}_{\mathbf{1}}, \tag{2.4}$$

where λ corresponds to generalized eigenvalues of (2.4) [13]. [14] constructs $[Y_1]$ and $[Y_2]$ using the N samples of $y[k]$ in such a way that λ corresponds to the complex exponentials in (2.1). Therefore, the solution of λ, which is indeed an eigenvalue problem, becomes equivalent to the solution of the complex exponential set of the model (2.1). $\bar{\mathbf{Y}}_{\mathbf{1}}$ and $\bar{\mathbf{Y}}_{\mathbf{2}}$ are defined in [14] as

$$\bar{\mathbf{Y}}_{\mathbf{1}} = \bar{\mathbf{Z}}_{\mathbf{1}} \cdot \bar{\mathbf{R}} \cdot \bar{\mathbf{Z}}_{\mathbf{2}}, \tag{2.5}$$

$$\bar{\mathbf{Y}}_{\mathbf{2}} = \bar{\mathbf{Z}}_{\mathbf{1}} \cdot \bar{\mathbf{R}} \cdot \bar{\mathbf{Z}}_{\mathbf{0}} \cdot \bar{\mathbf{Z}}_{\mathbf{2}}, \tag{2.6}$$

where

$$\bar{\mathbf{Z}}_{\mathbf{1}} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ z_1 & z_2 & \cdots & z_M \\ \vdots & \vdots & & \vdots \\ z_1^{(N-L-1)} & z_2^{(N-L-1)} & \cdots & z_M^{(N-L-1)} \end{bmatrix}_{(N-L)\times M}, \tag{2.7}$$

$$\bar{\mathbf{Z}}_{\mathbf{2}} = \begin{bmatrix} 1 & z_1 & \cdots & z_1^{(L-1)} \\ 1 & z_2 & \cdots & z_2^{(L-1)} \\ \vdots & \vdots & & \vdots \\ 1 & z_M & \cdots & z_M^{(L-1)} \end{bmatrix}_{M\times L}, \tag{2.8}$$

$$\bar{\mathbf{Z}}_{\mathbf{0}} = \mathrm{diag}\left[z_1, z_2, \cdots, z_M\right], \tag{2.9}$$

$$\bar{\mathbf{R}} = \mathrm{diag}\left[R_1, R_2, \cdots, R_M\right]. \tag{2.10}$$

diag in (2.9) and (2.10) represents a diagonal matrix of size $M \times M$. The parameter L in (2.7) and (2.8) is called pencil parameter and acts as an upper bound on the possible values on M. If we evaluate (2.5) and (2.6) by using (2.7)–(2.10), we would obtain

$$\bar{\mathbf{Y}}_{\mathbf{1}} = \begin{bmatrix} y[0] & y[1] & \cdots & y[L-1] \\ y[1] & y[2] & \cdots & y[L] \\ \vdots & \vdots & & \vdots \\ y[N-L-1] & y[N-L] & \cdots & y[N-2] \end{bmatrix}_{(N-L)\times L}, \tag{2.11}$$

$$\bar{\mathbf{Y}}_{\mathbf{2}} = \begin{bmatrix} y[1] & y[2] & \cdots & y[L] \\ y[2] & y[3] & \cdots & y[L+1] \\ \vdots & \vdots & & \vdots \\ y[N-L] & y[N-L+1] & \cdots & y[N-1] \end{bmatrix}_{(N-L)\times L} . \tag{2.12}$$

By using (2.5) and (2.6) in (2.4), we get

$$\bar{\mathbf{Y}}_{\mathbf{2}} - \lambda \bar{\mathbf{Y}}_{\mathbf{1}} = \bar{\mathbf{Z}}_{\mathbf{1}} \cdot \bar{\mathbf{R}} \cdot \{\bar{\mathbf{Z}}_{\mathbf{0}} - \lambda \bar{\mathbf{I}}\} \cdot \bar{\mathbf{Z}}_{\mathbf{2}}. \tag{2.13}$$

It is showed in [10] that, provided

$$M \leq L \leq N - M, \tag{2.14}$$

the rank of the matrix pencil defined in (2.4) will reduce by one if λ is equal to one of the diagonal entries of $\bar{\mathbf{Z}}_{\mathbf{0}}$. In other words, λ will be a rank-reducing number when it is equal to one of the $\{z_i\}$; which is also equivalent to saying $\{z_i\}$ are the generalized eigenvalues of the system defined in (2.4).

We construct our eigenvalue problem as

$$\left\{\bar{\mathbf{Y}}_{\mathbf{2}} - \lambda \bar{\mathbf{Y}}_{\mathbf{1}}\right\} \cdot \mathbf{v_i} = 0, \tag{2.15}$$

or equivalently

$$\left\{\bar{\mathbf{Y}}_{\mathbf{1}}^{\dagger} \bar{\mathbf{Y}}_{\mathbf{2}} - \lambda \bar{\mathbf{I}}\right\} \cdot \mathbf{v_i} = 0, \tag{2.16}$$

where $\{\mathbf{v_i}\}$ are the eigenvectors of the matrix pencil and $\bar{\mathbf{I}}$ is the identity matrix. $\bar{\mathbf{Y}}_{\mathbf{1}}^{\dagger}$ is defined as the Moore-Penrose pseudoinverse of $\bar{\mathbf{Y}}_{\mathbf{1}}$:

$$\bar{\mathbf{Y}}_{\mathbf{1}}^{\dagger} = \left\{\bar{\mathbf{Y}}_{\mathbf{1}}^{H} \cdot \bar{\mathbf{Y}}_{\mathbf{1}}\right\}^{-1} \cdot \bar{\mathbf{Y}}_{\mathbf{1}}^{H}, \tag{2.17}$$

where H denotes conjugate transpose. Often the $\bar{\mathbf{Y}}_{\mathbf{1}}^{\dagger} \cdot \bar{\mathbf{Y}}_{\mathbf{2}}$ product produces a bad conditioned matrix, especially when dealing with noisy data. Total least squares matrix pencil method (TLS–MPM) proposes a new approach to find λ that can be used for noisy data. In TLS–MPM, we combine (2.11) and (2.12) to form the single matrix

$$\bar{\mathbf{Y}} = \begin{bmatrix} y[0] & y[1] & \cdots & y[L] \\ y[1] & y[2] & \cdots & y[L+1] \\ \vdots & \vdots & & \vdots \\ y[N-L-1] & y[N-L] & \cdots & y[N-1] \end{bmatrix}_{(N-L)\times(L+1)} , \tag{2.18}$$

which can be defined in terms of (2.12) and (2.13) as

$$\begin{aligned}\bar{\mathbf{Y}} &= \begin{bmatrix} c_1 & \bar{\mathbf{Y}}_{\mathbf{2}} \end{bmatrix} \\ &= \begin{bmatrix} \bar{\mathbf{Y}}_{\mathbf{1}} & c_{L+1} \end{bmatrix},\end{aligned} \tag{2.19}$$

where c_i represents the i^{th} column of (2.18). We use singular value decomposition (SVD) approach on (2.18) to filter the singular values that cause bad conditioning.

$$\bar{\mathbf{Y}} = \bar{\mathbf{U}} \cdot \bar{\mathbf{\Sigma}} \cdot \bar{\mathbf{V}}^H. \tag{2.20}$$

In (2.20) we can modify $\bar{\mathbf{V}}$ such that

$$\bar{\mathbf{Y}}_{\mathbf{1}} = \bar{\mathbf{U}} \cdot \bar{\mathbf{\Sigma}} \cdot \bar{\mathbf{V}}_{\mathbf{1}}^H, \tag{2.21}$$

$$\bar{\mathbf{Y}}_{\mathbf{2}} = \bar{\mathbf{U}} \cdot \bar{\mathbf{\Sigma}} \cdot \bar{\mathbf{V}}_{\mathbf{2}}^H, \tag{2.22}$$

where, $\bar{\mathbf{V}}_{\mathbf{1}}$ is obtained by removing the last row of $\bar{\mathbf{V}}$ and $\bar{\mathbf{V}}_{\mathbf{2}}$ is obtained by removing the first row of $\bar{\mathbf{V}}$. We filter the singular values by keeping M of them, which ensures a better condition number. Likewise, we also keep or discard the corresponding columns and rows of $\bar{\mathbf{U}}$ and $\bar{\mathbf{\Sigma}}$. This can be represented in MATLAB notation as

$$\bar{\mathbf{U}}' = \bar{\mathbf{U}}(:, 1 : M), \tag{2.23}$$

$$\bar{\mathbf{V}}_{\mathbf{1}}' = \bar{\mathbf{V}}_{\mathbf{1}}(:, 1 : M), \tag{2.24}$$

$$\bar{\mathbf{V}}_{\mathbf{2}}' = \bar{\mathbf{V}}_{\mathbf{2}}(:, 1 : M), \tag{2.25}$$

$$\bar{\mathbf{\Sigma}}' = \bar{\mathbf{\Sigma}}(1 : M, 1 : M). \tag{2.26}$$

By using (2.23)-(2.26), we can rewrite (2.21) and (2.22) as

$$\bar{\mathbf{Y}}_{\mathbf{1}}' = \bar{\mathbf{U}}' \cdot \bar{\mathbf{\Sigma}}' \cdot \bar{\mathbf{V}}_{\mathbf{1}}'^H, \tag{2.27}$$

$$\bar{\mathbf{Y}}_{\mathbf{2}}' = \bar{\mathbf{U}}' \cdot \bar{\mathbf{\Sigma}}' \cdot \bar{\mathbf{V}}_{\mathbf{2}}'^H. \tag{2.28}$$

After substituting (2.27) and (2.28) back into (2.15), we obtain

$$\left\{\bar{\mathbf{U}}' \cdot \bar{\mathbf{\Sigma}}' \cdot \bar{\mathbf{V}}_{\mathbf{2}}'^H - \lambda \bar{\mathbf{U}}' \cdot \bar{\mathbf{\Sigma}}' \cdot \bar{\mathbf{V}}_{\mathbf{1}}'^H\right\} \cdot \mathbf{v_i} = 0. \tag{2.29}$$

Left multiplying (2.29) with $\bar{\mathbf{\Sigma}}'^{-1} \cdot \bar{\mathbf{U}}'^H$ will give us

$$\left\{\bar{\mathbf{V}}_{\mathbf{2}}'^H - \lambda \bar{\mathbf{V}}_{\mathbf{1}}'^H\right\} \cdot \mathbf{v_i} = 0, \tag{2.30}$$

which follows from the fact that $\bar{\mathbf{U}}'$ is a unitary matrix and $\{\bar{\mathbf{\Sigma}}'^{-1} \cdot \bar{\mathbf{U}}'^{H} \cdot \bar{\mathbf{U}}' \cdot \bar{\mathbf{\Sigma}}'\}$ is equal to the identity matrix. If we redefine the eigenvectors as

$$\mathbf{v_i} = \bar{\mathbf{V}}'_{\mathbf{1}} \cdot \mathbf{v}'_{\mathbf{i}}, \tag{2.31}$$

we will get

$$\left\{\bar{\mathbf{V}}'^{H}_{\mathbf{2}} \cdot \bar{\mathbf{V}}'_{\mathbf{1}} - \lambda \bar{\mathbf{V}}'^{H}_{\mathbf{1}} \cdot \bar{\mathbf{V}}'_{\mathbf{1}}\right\} \cdot \mathbf{v}'_{\mathbf{i}} = 0 \tag{2.32}$$

$$\left\{\left([\bar{\mathbf{V}}_{\mathbf{1}}]'^{H} \cdot \bar{\mathbf{V}}'_{\mathbf{1}}\right)^{-1} \cdot \bar{\mathbf{V}}'^{H}_{\mathbf{2}} \cdot \bar{\mathbf{V}}'_{\mathbf{1}} - \lambda \bar{\mathbf{I}}\right\} \cdot \mathbf{v}'_{\mathbf{i}} = 0 \tag{2.33}$$

$$\left\{\left([\bar{\mathbf{V}}_{\mathbf{1}}]'^{H} \cdot \bar{\mathbf{V}}'_{\mathbf{1}}\right)^{-1} \cdot \bar{\mathbf{V}}'^{H}_{\mathbf{2}} \cdot \bar{\mathbf{V}}'_{\mathbf{1}}\right\} \cdot \mathbf{v}'_{\mathbf{i}} = \lambda \mathbf{v}'_{\mathbf{i}}. \tag{2.34}$$

The solution of the eigenvalue problem in (2.34) will give the M complex exponentials. Once we determine the complex exponentials we can easily determine the residuals from (2.1) as

$$\begin{bmatrix} y[0] \\ y[1] \\ \vdots \\ y[N-1] \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ z_1 & z_2 & \cdots & z_M \\ \vdots & \vdots & & \vdots \\ z_1^{N-1} & z_2^{N-1} & \cdots & z_M^{N-1} \end{bmatrix} \cdot \begin{bmatrix} R_1 \\ R_2 \\ \vdots \\ R_M \end{bmatrix}. \tag{2.35}$$

2.2 Extrapolation Using MPM

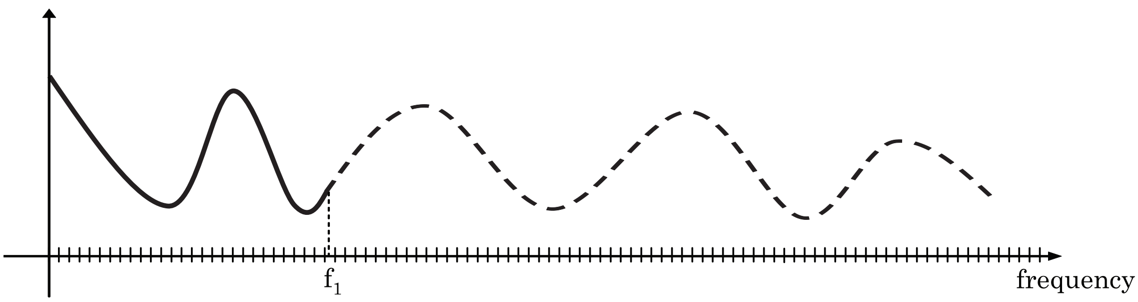


Figure 2.1: Low frequency part of the typical solution spectrum. Solid line represents the available data from LF solver and dashed line represents the unknown region.

Consider the situation that is illustrated in Figure 2.1, which represents a sample solution signal for the low frequency part of the solution spectrum. In this figure, solid line represents the known part of the signal, whereas, the dashed line represents the unknown part. Typically, the solid-line part is acquired from an LF solver such as FMM, where f_1 represents the maximum solvable frequency for this specific problem. The maximum solvable frequency depends on the method that is used to solve the problem (MOM, FMM, MLFMA or parallel MLFMA), electromagnetic size of the scatterer, mesh density and physical memory of the computer on which the problem is solved.

MPM, as described in the previous section, is used to find the parameters of the model that is to be constructed from the values of the scattering solution signal up to frequency f_1. Afterwards, this model, which is parametric in frequency, will be evaluated for the frequencies that are greater than f_1 to calculate the extrapolated scattering signal.

2.2.1 Forward and Backward Extrapolation

We perform all the computations in discrete frequency domain; hence, instead of using the actual frequency, the model is evaluated for frequency index k. For simplicity, we can assign $k = 0$ to the lowest frequency on the spectrum and $k = N-1$ to the maximum solvable frequency assuming that we have N samples in the solution region. Using these N samples MPM produces M complex exponentials and M residuals. The extrapolation is performed by evaluating (2.1) for values of k as we march in the extrapolation region in the increasing frequency direction. We call this procedure, i.e., modeling the LF solution and extrapolating in the increasing frequency direction, forward MPM extrapolation (FMPM).

Similarly, we can also model the HF solution and extrapolate in the decreasing frequency direction. We named this approach as backward MPM extrapolation (BMPM). There are two straightforward ways to perform this extrapolation: The first and simpler one is reversing the signal on the frequency axis so that the highest frequency becomes the lowest frequency with index $k = 0$ and the lowest frequency whose accuracy is acceptable becomes f_1 in Figure 2.1, with index $k = N - 1$, once again assuming we have N samples for the HF solution. After flipping the signal, FMPM can be applied, but the result of the extrapolation should also be reversed on the frequency axis. Second one is to start the frequency index of the model at $k = k_2$, where k_2 corresponds to the starting frequency of the HF solution, i.e., the lowest frequency whose accuracy is acceptable.

If we look closer to the theory of MPM described in the previous section, it can be observed that as long as the data samples that we construct the model from are same, we would end up with the same complex exponential set. In other words, changing the initial value of frequency index does not alter the complex exponentials. However, it does change the residuals that are associated with those complex exponentials. As a result, the extrapolated signal would be same regardless of the choice of the initial value for the frequency index.

Another remark for the frequency index k is that one should watch out for the behavior of the extrapolated signal at large values of k. As the extrapolation frequency increases, the value of k increases too. Recall that k is the power of the complex exponential in (2.1). If the magnitude of any complex exponential is significantly larger than unity, large values of k will blow the extrapolated signal up. There may also be some numerical errors for very large values of k, "very large" being dependent on the platform of implementation.

Up to this point we have never mentioned the parameter M. This parameter is nothing but the number of complex exponentials that is used to construct the model and is actually the key point of constructing an accurate model and performing a good extrapolation. In the next sections, we will investigate this parameter in detail and propose a M selection algorithm to obtain the best possible extrapolation.

2.2.2 The Parameter M

During the modeling and extrapolating process, we have left out mentioning about two very important parameters in MPM. They are pencil parameter L and number of exponentials M. Recall (2.14) that

$$M \leq L \leq N - M. \tag{2.36}$$

Although L is an intermediate parameter, it does affect the outcome of complex exponentials very much. When dealing with noisy data, it is always a good idea to consider changing L to seek for a better model. However, for simplicity, throughout this thesis, unless otherwise specified, we took

$$L = \frac{N}{2}, \tag{2.37}$$

which in turn gives us

$$M \leq \frac{N}{2}. \tag{2.38}$$

Equation (2.38) also gives the maximum number of exponentials that can be used by using a length-N data. It is obvious that if we have length-N data, i.e., N linear equations like (1.2), we could solve for N number of unknowns; $\frac{N}{2}$ of them being the complex exponentials, $\frac{N}{2}$ of them being corresponding residuals. As M represents the quantity of terms in the model, it should take integer values. Therefore, considering the case of N being odd, we should rewrite (2.38) as

$$M \leq \lfloor \frac{L}{2} \rfloor, \tag{2.39}$$

where $\lfloor\ \rfloor$ represents the floor operation.

Recall that, summation series like (2.1) tend to become more accurate as the number of terms in the series increases. Taylor series, Mie series and Fourier series are some infinite-length series that could be given as example to support this idea. As we cannot sum infinite terms, eventually we have to stop at some point, where the remaining not-summed terms would be the approximation error. In our work, the best model to represent the solution signal would be obtained by using the maximum possible number of exponentials which is $\frac{N}{2}$. This may seem correct when we consider the accuracy of the series summation idea just discussed, however, one should keep in mind that we will use the constructed model to extrapolate the signal, not to regenerate it in the interpolation region, i.e., in the region, where we already know the solution values. Using the maximum possible number of exponentials indeed models the interpolation region as perfectly as possible, but evaluating that model for frequencies in the deep extrapolation region will cause a blow-up on the extrapolated signal. The reason for this possible blow-up is obvious actually, by using maximum number of exponentials, we optimized our model with respect to the interpolation region and did not care about the extrapolation process. In the next section we will provide an optimization strategy to select a value of M that does "care" for the extrapolation process.

2.2.3 Brute-Force Optimization of M

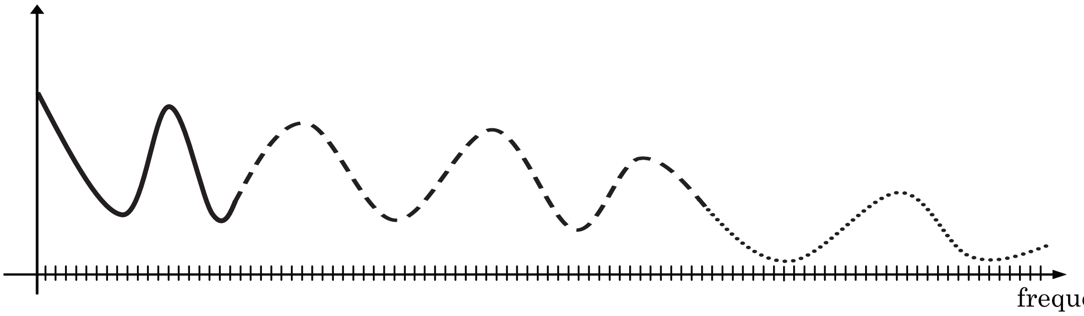


Figure 2.2: Brute-force M optimization for FMPM extrapolation. Solid line is modeled and extrapolation is performed in the increasing frequency direction. Extrapolation performance is evaluated by comparing extrapolated signal with the dotted line part.

Figure 2.2 illustrates the general situation for the case of FMPM. The LF available data is represented with a solid line, and the HF available data is represented with a dotted line; dashed line represents the unknown region for the solution signal. Usually what we perform in FMPM, is first modeling the part that is represented by solid line and evaluating that model for the rest of the spectrum. However, there are $\lfloor \frac{N}{2} \rfloor$ different models that we can build using different number of exponentials. Clearly, as discussed in the previous subsection, the approximation accuracy of the solid-line region, i.e., interpolation region (IR), increases as the number of exponentials used to create the model increases, but what happens to the extrapolated signal as we increase M?

In order to find a good value for M to use in extrapolation, we developed a simple algorithm that scans all possible values of M defined in (2.38). For each possible value of M, a model is formed and extrapolation is performed up to the highest frequency of the spectrum. In order to identify a good extrapolated signal, we compare the error in the dotted-line region, i.e., we find the error between the extrapolated signal and HF available data. The M value that gives the smallest error is selected and used for extrapolation. Since we scan all possible values of M during this optimization approach, we call this procedure brute-force M optimization.

Available HF data gives us a chance to obtain an extrapolation error value, which is in fact invaluable. By using this technique we are minimizing the error in the extrapolation region (ER). The only possible drawback for this approach is the execution speed of the algorithm. Although MPM and extrapolation algorithms require very little CPU time, increasing the sample number on the data, N, will increase the CPU time of optimization process due to (2.39).

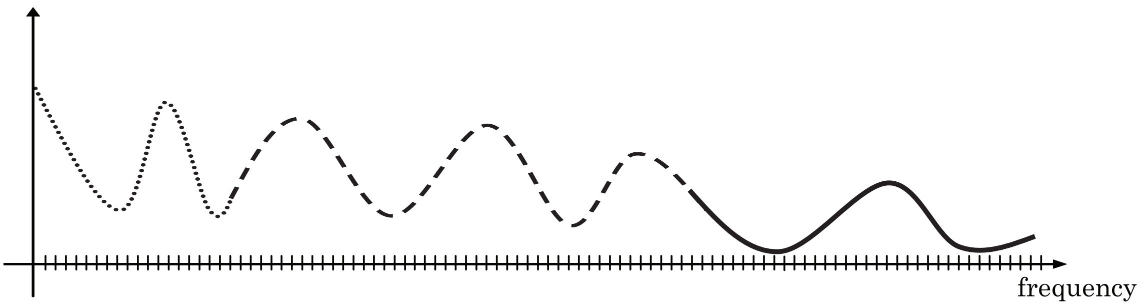


Figure 2.3: Brute-force M optimization for BMPM. Solid line is modeled and extrapolation is performed in the decreasing frequency direction. Extrapolation performance is evaluated by comparing extrapolated signal with the dotted line part.

In a similar sense we can apply this optimization to BMPM. Figure 2.3 illustrates the spectrum for the backward case. In this figure solid line represents the HF available data, dotted line represents the LF available data and dashed line again represents the unknown region. The optimization procedure is same but reversed.

The availability of two sources of information for the solution signal at the opposite ends of the spectrum is really a chance, but not absolutely necessary to perform an M optimization. In the case of single data source availability, e.g., only LF data, we can spare some solution data at the end of the solution region and model the rest. A similar M scan could be performed but this time instead of finding the error at the other end of the spectrum, we use the spared data to find the error at the beginning of the extrapolation region. When compared, using the error on the other end of the spectrum is clearly a better idea than the latter one.

2.3 Scattering from a Conducting Sphere

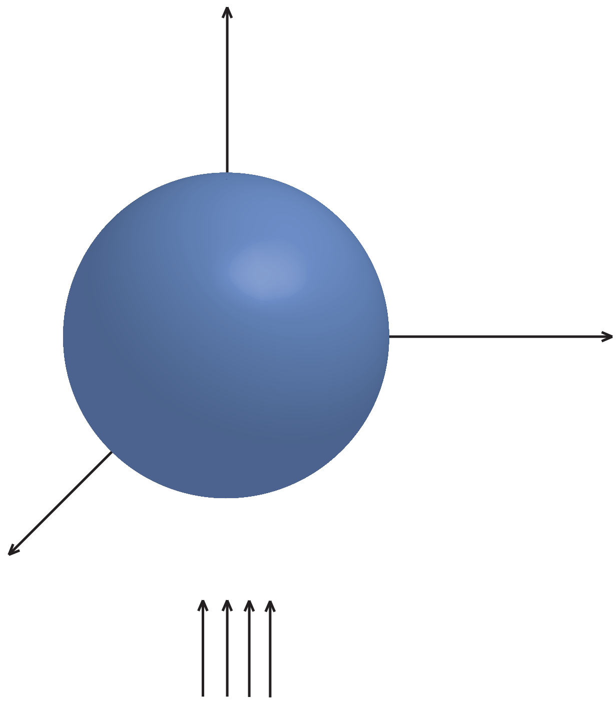


Figure 2.4: Conducting sphere illuminated from its south pole.

Up to this point we have given the theoretical background information for MPM, FMPM, BMPM, and brute-force M optimization. In this section, we will present some extrapolation examples that are end products of these methods. We will use backscattering radar cross section (RCS) of a conducting sphere that is illuminated by a plane wave from its south pole. Figure 2.4 illustrates this situation graphically. Backscattering RCS is the scattering from the south pole of the sphere, i.e., opposite to the direction of illumination.

We chose conducting sphere because we have implemented an analytical solver for this scattering problem in order to test the proposed extrapolation methods. Detailed theory of this solution is given in [12]. In the scattering solution of a sphere, all electromagnetic quantities are expressed as infinite summation of spherical harmonics, which is called Mie series. As we can not operate on infinite number of terms, an iterative technique is used to determine the number of terms to be used in the series for an externally given error criteria. Reflected field from the surface of the conducting sphere is derived analytically using the incident field and boundary conditions are enforced on the surface of the sphere. After finding the required number of spherical harmonics to be used in the expansion, the scattered far-field is calculated by simply evaluating the series. Thus, using this analytical solver we can obtain scattering value of conducting sphere at any frequency easily, in a way, before starting extrapolation we already know the signal on the whole spectrum.

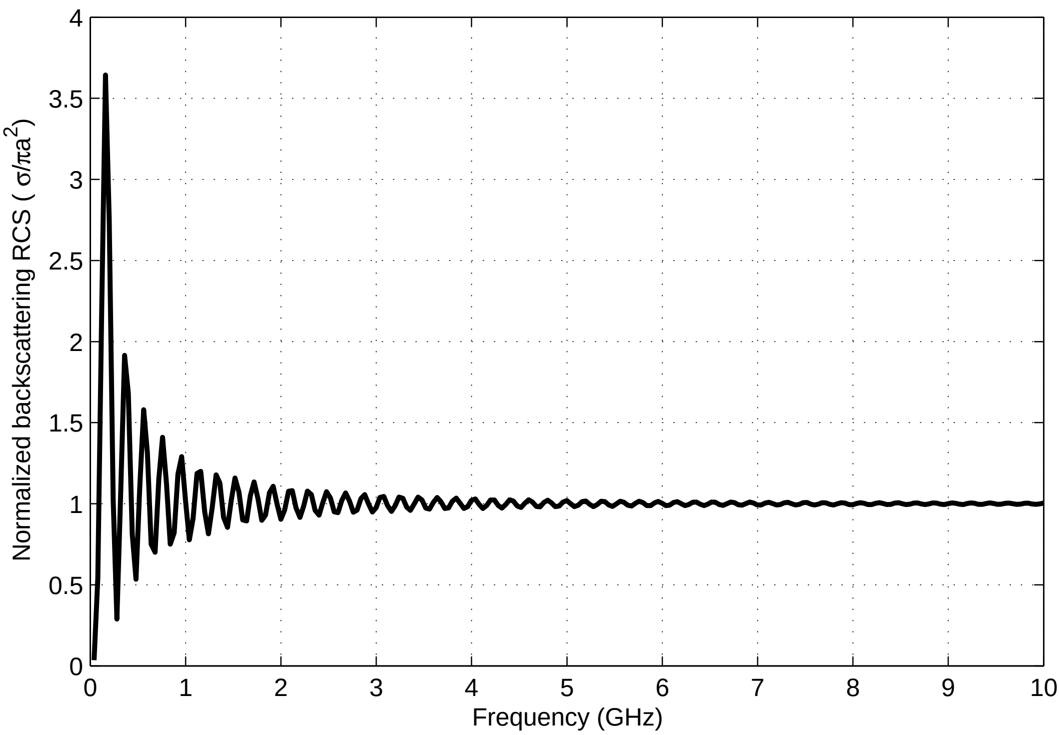


Figure 2.5: Normalized backscattering RCS of conducting sphere.

The knowledge of the scattering signal over the whole spectrum gives us the chance to test the accuracy of the proposed extrapolation methods. During the testing process, we will restrict our knowledge of the scattering signal to the relevant LF and HF region intervals, forming a situation similar to Figure 2.2 or 2.3. The performance of the extrapolation will be evaluated by comparing the extrapolated data and the previously obtained analytical data in the ER. The backscattering signal that will be used is illustrated in Figure 2.5. This signal is a well-known signature of the conducting sphere and is widely used to test the proposed methods and improvements in CEM. The RCS values, σ, are normalized with the geometric cross section of the sphere, πa^2, hence the signal goes to unity with oscillations of decreasing magnitude. This is actually expected because as we increase the frequency, the electromagnetic waves act more like light, and at very high frequencies RCS, i.e., what radar sees, becomes equivalent to what we see with our own eyes: the cross section of the sphere.

In Figure 2.5 we only represent the first 10 GHz of the signal, but during the extrapolation process, we used the interval [1:35] GHz. Because very-low frequency scattering solutions can have high oscillations that can degrade the performance of the modeling, we dismiss that region and start our solution signal from 1 GHz. Besides, for the purpose of extrapolation, generally scattering values whose frequency is smaller than 1 GHz are irrelevant. In the next section we will demonstrate the results of FMPM.

2.3.1 Sphere Results with FMPM

In order to test the extrapolation methods, we first select the LF and HF intervals. We make this selection as realistic as possible by keeping the limitations of actual numerical solvers in mind. For the backscattering RCS signal of the conducting sphere, we choose 1–5 GHz region as LF region and 25–35 GHz region as HF data. We used a 0.04 GHz frequency sampling when obtaining solution signal from analytical solver, therefore, we have 101 LF data points and 251 HF data points.

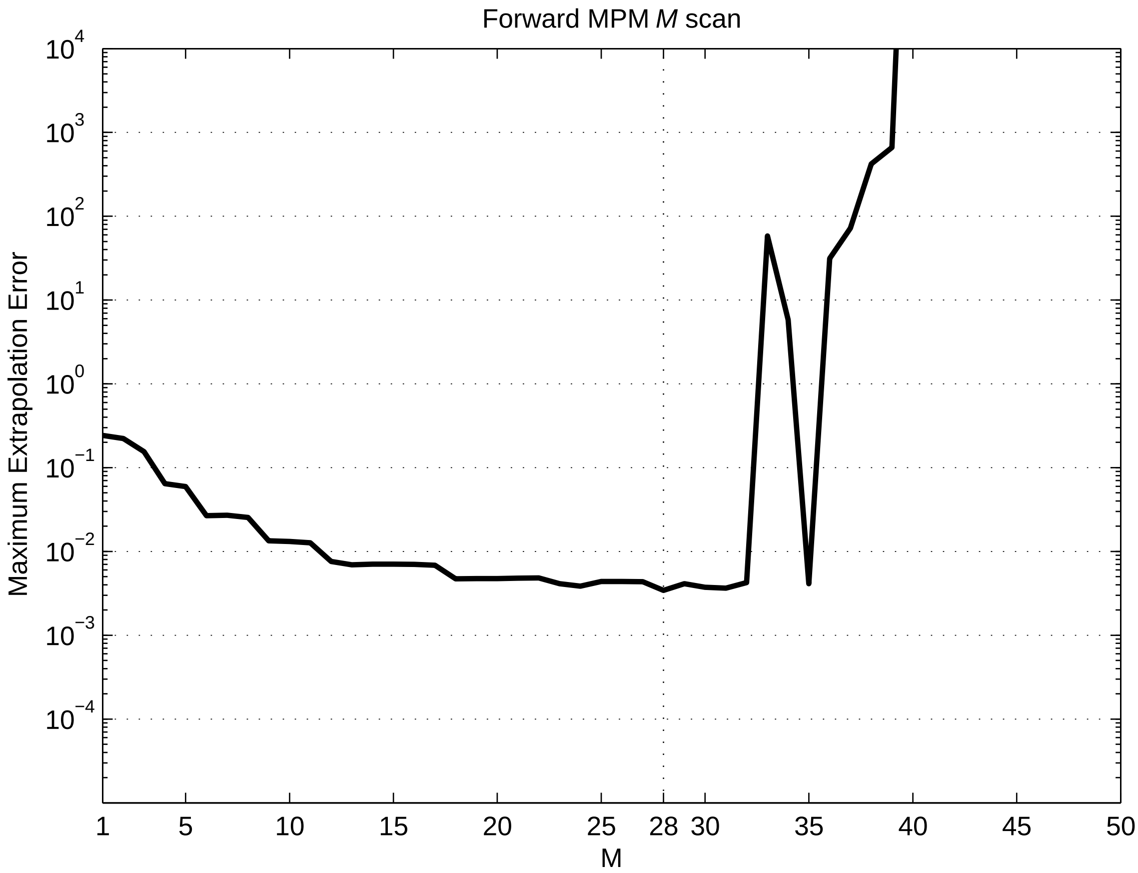


Figure 2.6: Results of brute-force M optimization for FMPM.

Figure 2.6 illustrates the results of the brute-force M optimization. Recall that in FMPM we model the LF data, in this case 1–5 GHz region, and perform extrapolation up to 35 GHz. The extrapolation error is the difference between the extrapolated signal and the HF data. Hence, the error levels in Figure 2.6 corresponds to the maximum error between the extrapolated data and HF data in the 25–35 GHz region for each value of M.

Notice that, as we discussed earlier, for very large values of M, maximum extrapolation error tends to increase; the error value is even greater than 10^4 after $M = 40$, where it can not fit into the plot. At those values of M, interpolation error is very low, but, it can be clearly seen why it is a bad idea to select those values of M for extrapolation. The minimum of maximum extrapolation errors is achieved at $M = 28$, which is indicated with a dotted vertical line in the figure.

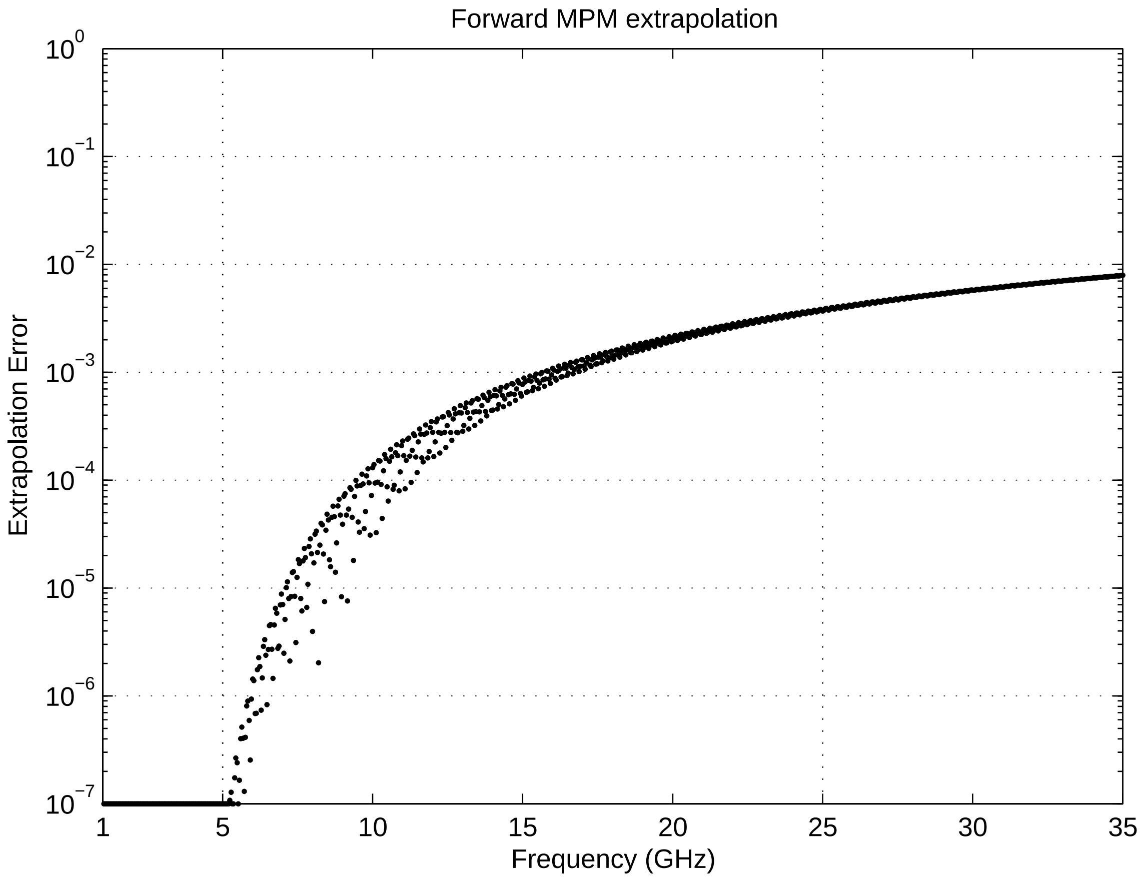


Figure 2.7: Extrapolation error for FMPM.

In Figure 2.7, we present the extrapolation result using this M value. We only present the error level for each frequency because at these levels of error two signals are almost undistinguishable on the same plot. In this figure, error levels that are below 10^{-7} are also shown as 10^{-7}, so interpolation error, as expected, is very low. As we march in the ER, error tends to increase but with a smaller slope as frequency increases. The maximum error at the end of the spectrum is below 10^{-2}, which can also be seen from Figure 2.6. As the original signal converges to unity as frequency increases, we can conclude that 10^{-2} corresponds to 1% maximum error.

2.3.2 Sphere Results with BMPM

In BMPM, we model the HF data, which is between 25–35 GHz, and extrapolate in the decreasing frequency direction. In the sphere problem HF data has 251 data points, which means maximum value of M is 125, according to (2.39).

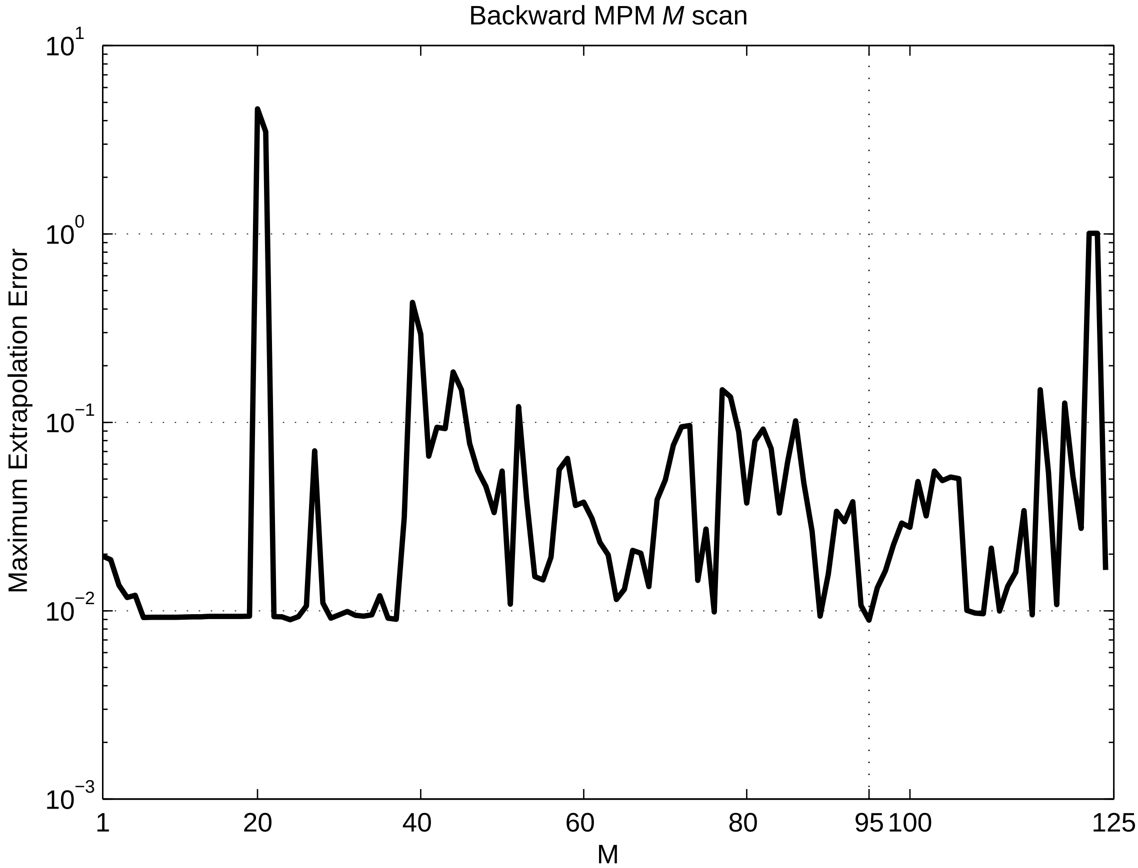


Figure 2.8: Results of brute-force M optimization for BMPM.

Figure 2.8 illustrates the brute-force M optimization results. Each error level in this figure is calculated as the maximum value of difference between the extrapolated signal and LF data. Minimum of maximum extrapolation errors occurs at $M = 95$, which is indicated by a dotted vertical line in the figure. When we compare Figures 2.6 and 2.8, it can be seen that, in BMPM case, maximum extrapolation error does not blow up for very large values M. The reason for this difference arises from the way we perform the BMPM. If we happen to reverse the signal, as discussed in the previous section, this figure will look similar to

Figure 2.6; however, we select the initial k index of the model without changing the original frequency-index assignment. Therefore, k values starts from 601 in the model (2.1) and in the system (2.35), where $k = 601$ corresponds to 25 GHz when spectrum starts from 1 GHz and increments with 0.04 GHz. By doing so, we, in a way, prevent the extrapolated signal to blow up.

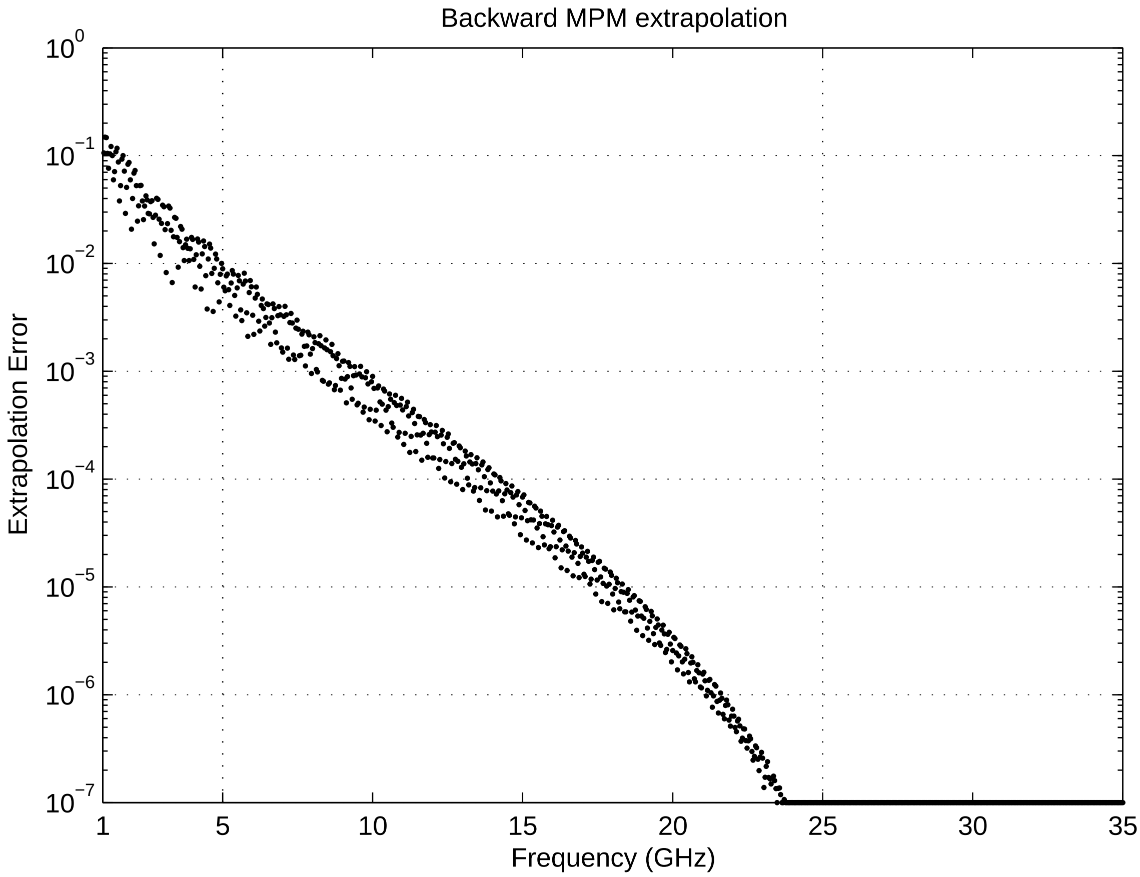


Figure 2.9: Extrapolation error for BMPM.

Using the result of brute-force M optimization, we construct the model with 95 exponentials and perform extrapolation in the decreasing frequency direction. Once again we only present the error values for each frequency in Figure 2.9 due to the same reasoning as in the FMPM case. Similarly, the error is very small in the IR and tends to increase as we march in the ER. Maximum error for BMPM turns out to be smaller than 2×10^{-1}, which corresponds to 20% maximum error.

When we look closely at Figures 2.6 and 2.8, it can be realized that, the automatic selection of minimum value for maximum extrapolation errors is indeed the absolute minimum. However, there are plenty of other possible values of M such that the maximum error is very close to the absolute minimum. For example, in the BMPM case there is no need to take 95 exponentials to construct the model, only 5–10 will be enough and will give very close extrapolation results. This concept will be revisited when we introduce the techniques that are used to model computational data in Chapter 4.

Chapter 3

Coupled Extrapolation

3.1 Reasons for Coupling

In the previous chapter we discussed the basic extrapolation strategies: FMPM and BMPM. The results of these extrapolation strategies were tested on the conducting sphere problem. These results seem promising, especially FMPM case, where maximum error is less than 1%. However, even the idea of under-utilizing the available data during modeling is enough to seek for some improvements on these extrapolation strategies.

At the end of the extrapolation results in the previous chapter, we have two separate extrapolated signals that are obtained from different models. Among those two extrapolated signals one can not simply identify one as the better one. In the regions where FMPM signal has high error levels, BMPM signal has very low error levels and vice versa. Therefore, after looking at this general picture, we start to seek some way to combine these two signals into one and came up with two different methods. In the next section, Coupled Matrix Pencil Method (COMPM), a modification to original MPM will be presented. Later, we will present the Coupled Deconvolution Matrix Pencil Method (CDMPM), a signal processing approach to combine the two signals. At the end of the chapter, we once again present the results of these two proposed methods on conducting sphere problem.

3.2 Coupling the Models - COMPM

Using either FMPM or BMPM approach causes an under utilization of the information we have about the scattering signal. In the FMPM case we only model the LF data without considering the behavior of the HF data. Although we use the HF data in the brute-force M optimization, it still does not affect the model and its parameters. As discussed in Section 2.2.3, using the HF data is not so crucial to find an appropriate M in FMPM; we can also spare some data from the LF data and use it as the reference data. Therefore, having an HF data region is not a necessity when performing FMPM. For BMPM we have a similar argument, where we only model the HF data and the LF data is used only to select an appropriate M value. However, as we do have extra data we should use it to improve the quality of the model and reduce the error levels on the extrapolated signal.

Our first proposal is to couple the two models. Instead of constructing two separate models to extrapolate the same signal, we form a single model and obtain a single extrapolated signal. We call this approach COMPM. First, we need to synchronize the k indexes of the two separate data so that, while constructing the single model for both data sources we have an absolute k index. Synchronizing means, the LF data starts with an index of $k = 0$ and ends with the index $k = k_1 = N_1$, the HF data starts with an index of $k = k_2 = N_2$ and ends with $k = N - 1$, assuming that whole spectrum is represented with N samples. k_1 and k_2 corresponds to the highest and lowest frequencies of LF and HF data respectively. The general picture would look as follows: we have a scattering signal with N samples but we only have the values of $N_L + N_H$ samples, rest will be estimated with extrapolation; here N_L and N_H are the LF and HF data point quantities respectively.

We obtain two complex exponential sets using the procedure described in Section 2.1 for two separate data sources, $\{z_{Li}\}$ and $\{z_{Hi}\}$. This process is performed independently. After finding these two sets of complex exponentials, we construct a matrix equation similar to (2.35), but this time we use both exponential sets;

$$\begin{bmatrix} y[0] \\ y[1] \\ \vdots \\ y[N_1] \\ y[N_2] \\ \vdots \\ y[N-1] \end{bmatrix} = \begin{bmatrix} 1 & 1 & \cdots & 1 & 1 & 1 & \cdots & 1 \\ z_{L1} & z_{L2} & \cdots & z_{LM_L} & z_{H1} & z_{H2} & \cdots & z_{HM_H} \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ z_{L1}^{N_1} & z_{L2}^{N_1} & \cdots & z_{LM_L}^{N_1} & z_{H1}^{N_1} & z_{H2}^{N_1} & \cdots & z_{HM_H}^{N_1} \\ z_{L1}^{N_2} & z_{L2}^{N_2} & \cdots & z_{LM_L}^{N_2} & z_{H1}^{N_2} & z_{H2}^{N_2} & \cdots & z_{HM_H}^{N_2} \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ z_{L1}^{N-1} & z_{L2}^{N-1} & \cdots & z_{LM_L}^{N-1} & z_{H1}^{N-1} & z_{H2}^{N-1} & \cdots & z_{HM_H}^{N-1} \end{bmatrix} \cdot \begin{bmatrix} R_1 \\ R_2 \\ \vdots \\ R_{M_L+M_H} \end{bmatrix}, \quad (3.1)$$

where M_L and M_H are the number of complex exponentials that are obtained from LF and HF data respectively. Notice that in (3.1), the two sets of complex exponentials are arranged side-by-side. This enforces the model to use both complex exponential sets for each point on the scattering signal, i.e., we are using both complex exponential sets from LF and HF regions to model the LF region, as well as HF region. The left hand side of (3.1) has a discontinuity on k index, where we simply cascaded all of the known data. Solution of (3.1) will result in $M_L + M_H$ coupled residuals, hence completing the coupled model.

Extrapolation is performed by evaluating (2.1) for coupled complex exponentials and coupled residuals. As a result of extrapolation, a single extrapolated signal is obtained. Equation (3.1) is also a least squares problem like (2.35) and solution of residuals is in a way fitting the model to the signal y as much as possible; but as we couple the two data sources in (3.1), the fitting treats both data sources equally and tries to estimate the signal y in its LF and HF regions as much as possible. The flowchart that summarizes COMPM is presented in Figure 3.1. Next, we will discuss the modifications that we should do on brute-force M optimization to use with COMPM.

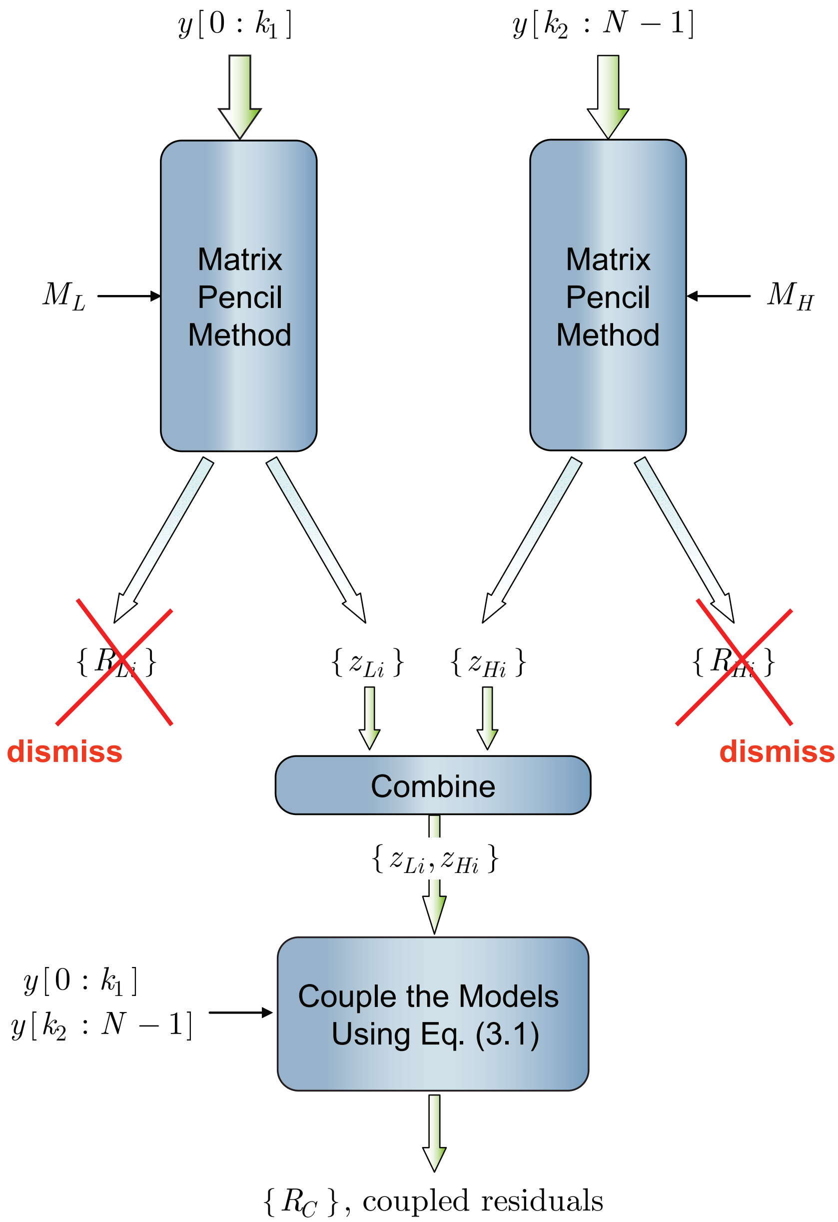


Figure 3.1: Flowchart of COMPM.

3.2.1 Brute-Force Optimization for COMPM

We discussed the brute-force optimization for FMPM and BMPM in Section 2.2.3. In COMPM, we once again need to select number of complex exponentials that will be used when constructing the model; however, this time we have two degrees of freedom, i.e., two M numbers to select, one for the LF data and one for the HF data. Although finding the complex exponential set for these two separate data sources is an independent process, they form a single model. Therefore, we need to perform a two-dimensional optimization and find an M pair that is good for extrapolation.

In brute-force optimization for COMPM, we scan all possible values of M_L and M_H under the constraints of (2.39) using N_L and N_H. For each pair of (M_L,M_H), coupled residuals are calculated and the coupled model is constructed. This model is then used to calculate the extrapolated signal over the entire spectrum. The maximum error in the interpolation region is stored in a matrix, whose location is associated with a (M_L,M_H) pair. We have to use the IR as the reference data because we have used all the data available to us. Similar to the one-dimensional case, we can spare some data from the end of the LF available data and the beginning of the HF data, however, this would decrease the amount of data we would use when constructing the model. Whatever the reference data is chosen as, we select the best M pair by locating the minimum value in the error matrix.

3.3 Coupling the Signals - CDMPM

In the previous section we perform extrapolation by coupling the models, i.e., we couple the model parameters. However, in this section we directly use the extrapolation results that we present in Section 2.3 and apply signal processing techniques on those two extrapolated signals to produce one, coupled extrapolated signal. We refer to this process as CDMPM as we make use of the deconvolution operation.

The main motivation for coupling the signals is the low error levels of FMPM and BMPM at the beginning of the extrapolation region. We will use weighting windows for each signal such that the influence of FMPM signal would be high relative to BMPM right after $k = k_1$; and relatively low right before $k = k_2$. This way, coupled extrapolated signal would be affected by FMPM or BMPM signal more than other at the regions where the error levels are lower than the other extrapolated signal.

Notice that, in CDMPM, we do not alter the complex exponentials or their residuals. These model parameters are directly taken from the model that is used to perform FMPM and BMPM. Likewise, number of exponentials that we should use when constructing the models are selected according to the procedure that is discussed in Section 2.2.3. Therefore there is no need to alter the brute-force optimization as we did in COMPM. Next, we will present the basics of deconvolution operation in discrete domain and carry on with the description of coupled deconvolution process.

3.3.1 Deconvolution in Discrete Domain

Deconvolution is simply defined as undoing what convolution does. It is a well known property in signal processing that convolution in time domain corresponds to multiplication in the Fourier domain [15]; considering two signals x and y and their convolution c in time domain, this property can be represented as

$$c\left(t\right) = \mathcal{F}^{-1}\left\{\mathcal{F}\left\{x\left(t\right)\right\}\ \mathcal{F}\left\{y\left(t\right)\right\}\right\}, \tag{3.2}$$

where, $\mathcal{F}\{\ \}$ represents the Fourier transform and convolution is defined as

$$c\left(t\right) = \int_{-\infty}^{\infty} x\left(\tau\right) y\left(t-\tau\right) d\tau. \tag{3.3}$$

When we work in discrete domain, we should change our domain transformation tool, i.e., we should use discrete Fourier transform (DFT) instead of continuous Fourier transform. We can rewrite (3.2) in discrete domain with $x[n]$ and

$y[n]$ sampled at N points as

$$c\,[n] = \mathrm{DFT}^{-1}\left\{\mathrm{DFT}\left\{x\,[n]\right\}\ \ \mathrm{DFT}\left\{y\,[n]\right\}\right\}. \tag{3.4}$$

In addition to changing the domain transformation tool, we should also change the convolution operation. In order for (3.4) to be valid, a new type of convolution, circular convolution, should be defined [15]:

$$c\,[n] = \sum_{m=0}^{N-1} x\,[\langle m\rangle_N]\, y\,[\langle n-m\rangle_N]. \tag{3.5}$$

Equation (3.5) defines an N point circular convolution, where $\langle\ \rangle_N$ defines a circular shift in a length N sequence. With this set of signals, finding $x[n]$ from known $c[n]$ and $y[n]$ is a circular deconvolution problem. If we rewrite (3.5) in matrix notation as

$$\bar{\mathbf{Y}} \cdot \mathbf{x} = \mathbf{c}, \tag{3.6}$$

we can define the circular deconvolution problem as

$$\mathbf{x} = \bar{\mathbf{Y}}^{-1} \cdot \mathbf{c}, \tag{3.7}$$

where $\bar{\mathbf{Y}}$ is $N \times N$ matrix that contains the values of $y[n]$. For a signal like

$$y\,[n] = \{a, b, c, d, e\} \tag{3.8}$$

the circular convolution matrix, $\bar{\mathbf{Y}}$, would be,

$$\bar{\mathbf{Y}} = \begin{bmatrix} a & e & d & c & b \\ b & a & e & d & c \\ c & b & a & e & d \\ d & c & b & a & e \\ e & d & c & b & a \end{bmatrix}, \tag{3.9}$$

circularly shifting every column by one. We will use this circular deconvolution operation as a basic building block of coupling in the next section.

3.3.2 Coupled Deconvolution

Assume that $Y[k]$ represents the complete scattering signal in frequency domain. At the beginning of the extrapolation problem, this signal is not available in complete form; we only have the knowledge of LF and HF regions. We model the LF data between $k = 0$ and $k = k_1$ and perform extrapolation up to $k = N - 1$ using FMPM; similarly we model the HF data between $k = k_2$ and $k = N - 1$ and perform extrapolation down to $k = 0$ using BMPM. Therefore, we have two estimates of $Y[k]$ in the interval between $k = 0$ and $k = N - 1$.

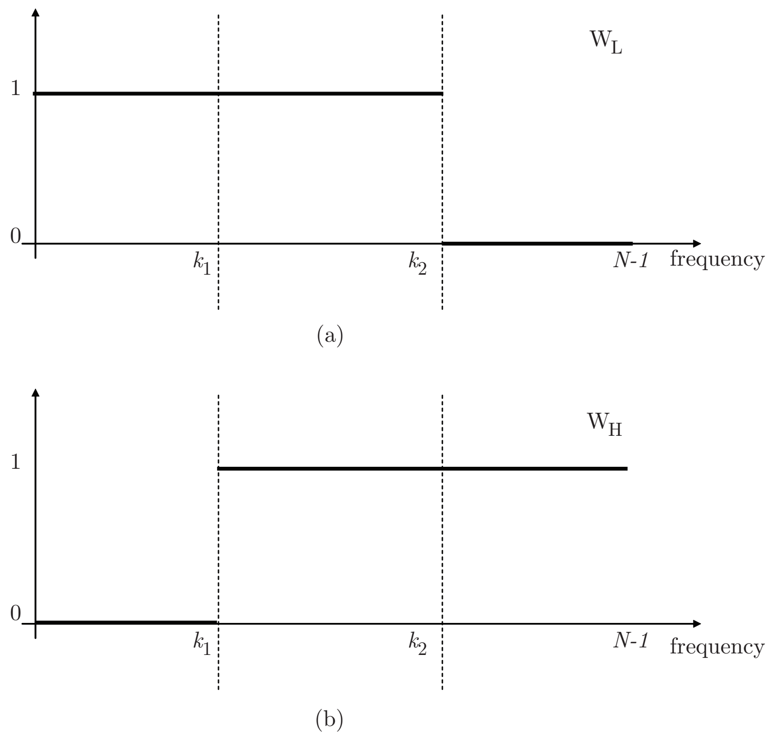


Figure 3.2: Truncation windows for (a) FMPM signal (b) BMPM signal.

Consider the two windows presented in Figure 3.2; dotted vertical lines separate LF region, ER and HF region. In the interval $k = [0 : k_1]$, i.e., in the LF region, we keep the value of the window that will be used with FMPM signal unity and value of the window that will be used for BMPM signal zero. This is due to the fact that in this region we already have a solution data and do not want the BMPM signal interfere with it, i.e., we keep the FMPM signal and discard the BMPM one. A similar approach is carried out for the region $k = [k_2 : N-1]$. This way we ensure that coupling only occurs in the ER, i.e., $k = [k_1 : k_2]$. Next we define two signals in frequency domain, $X_L[k]$ and $X_H[k]$ as

$$W_L[k]\ Y_1[k] = X_L[k], \tag{3.10}$$

$$W_H[k]\ Y_2[k] = X_H[k]. \tag{3.11}$$

$X_L[k]$ and $X_H[k]$ are the truncated versions of $Y_1[k]$ and $Y_2[k]$ respectively. $Y_1[k]$ and $Y_2[k]$ are estimates of $Y[k]$ obtained from FMPM and BMPM. Therefore, with the knowledge of $X_L[k]$ and $X_H[k]$ we can write eqs. (3.10) and (3.11) as

$$W_L[k]\ Y[k] = X_L[k], \tag{3.12}$$

$$W_H[k]\ Y[k] = X_H[k]. \tag{3.13}$$

Separate solution of either (3.12) or (3.13) will yield $Y_1[k]$ or $Y_2[k]$, depending on the equation solved. However, simultaneous solution of these two equations will lead to a coupled $Y[k]$, a single signal in frequency domain. By using property (3.4), in time domain we can write

$$x_L[n] = w_L[n] \otimes y[n], \tag{3.14}$$

$$x_H[n] = w_H[n] \otimes y[n], \tag{3.15}$$

where $\otimes$ is the N point circular convolution operator, N being the number of samples of $y[n]$. We can rewrite (3.14) and (3.15) in matrix form as in (3.6):

$$\bar{\mathbf{W}}_{\mathbf{L}} \cdot \mathbf{y} = \mathbf{x}_{\mathbf{L}}, \tag{3.16}$$

$$\bar{\mathbf{W}}_{\mathbf{H}} \cdot \mathbf{y} = \mathbf{x}_{\mathbf{H}}, \tag{3.17}$$

where $\bar{\mathbf{W}}_{\mathbf{L}}$ and $\bar{\mathbf{W}}_{\mathbf{H}}$ are circular convolution matrices of $w_L[n]$ and $w_H[n]$ respectively. If we combine (3.16) and (3.17) into a single matrix equation, we would obtain

$$\begin{bmatrix} \bar{\mathbf{W}}_{\mathbf{L}} \\ \bar{\mathbf{W}}_{\mathbf{H}} \end{bmatrix} \cdot [\mathbf{y}] = \begin{bmatrix} \mathbf{x}_{\mathbf{L}} \\ \mathbf{x}_{\mathbf{L}} \end{bmatrix}. \tag{3.18}$$

Solution of (3.18) is a total least squares problem, which will give the time domain counterpart of the coupled extrapolated signal. In order to obtain the frequency domain coupled signal, we need to apply DFT on the solution of (3.18). Various approaches can be used to solve (3.18), such as singular value decomposition or pseudoinverse [16].

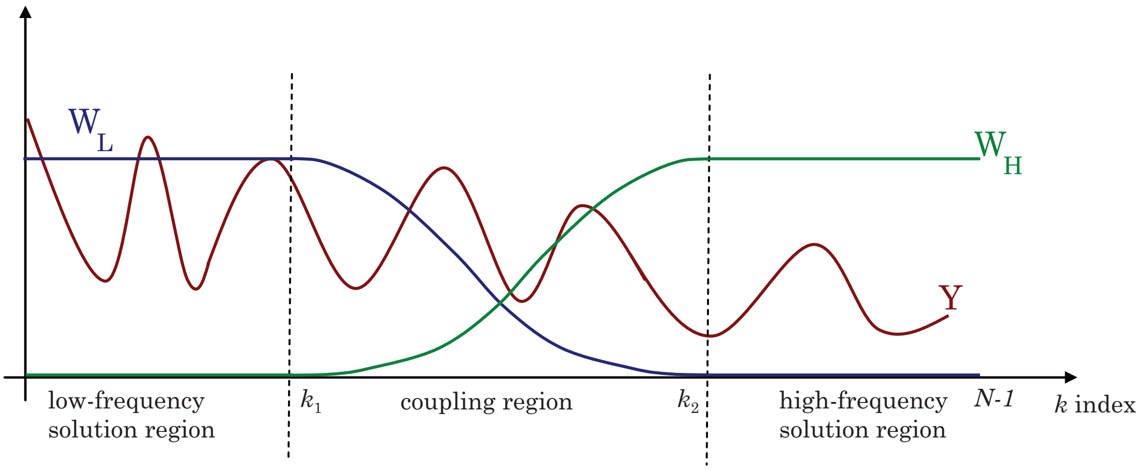


Figure 3.3: Weighting windows used in CDMPM.

Using windows in Figure 3.2 will introduce coupling in the region where both window has value one, i.e., in the ER. However, when the results of Section 2.3 are considered, using rectangular window may not be the best way to perform the coupling operation for MPM-based extrapolation. From the results of Section 2.3 we observe that extrapolation error is low at the beginning of the ER and it increases in the direction of extrapolation. Therefore, it is better to use smooth-transition windows as illustrated in Figure 3.3.

$W_L[k]$ and $W_H[k]$ in Figure 3.3 become the weighting windows for FMPM and BMPM signals respectively. We once again use unity and zero weights in the IR. However, in the coupling region, where index is between $k = k_1$ and $k = k_2$, the weights are adjusted such that at indexes closer to k_1, $W_L[k]$ is significantly greater than $W_H[k]$, and at the indexes that are closer to k_2, $W_H[k]$ is significantly greater than $W_L[k]$; around the center of the coupling region, both weights are similar.

In Figure 3.4 we illustrate reasons of weight values in each region. Note that, these weights are not weights in the regular sense, but they can still be treated as weights in the sense that we are weighting the coupling process by looking at the extrapolation errors. They are just some windowing functions in frequency domain and they do not have to add-up to unity for each index k. The same coupling procedure will be carried out with these new windows too. The

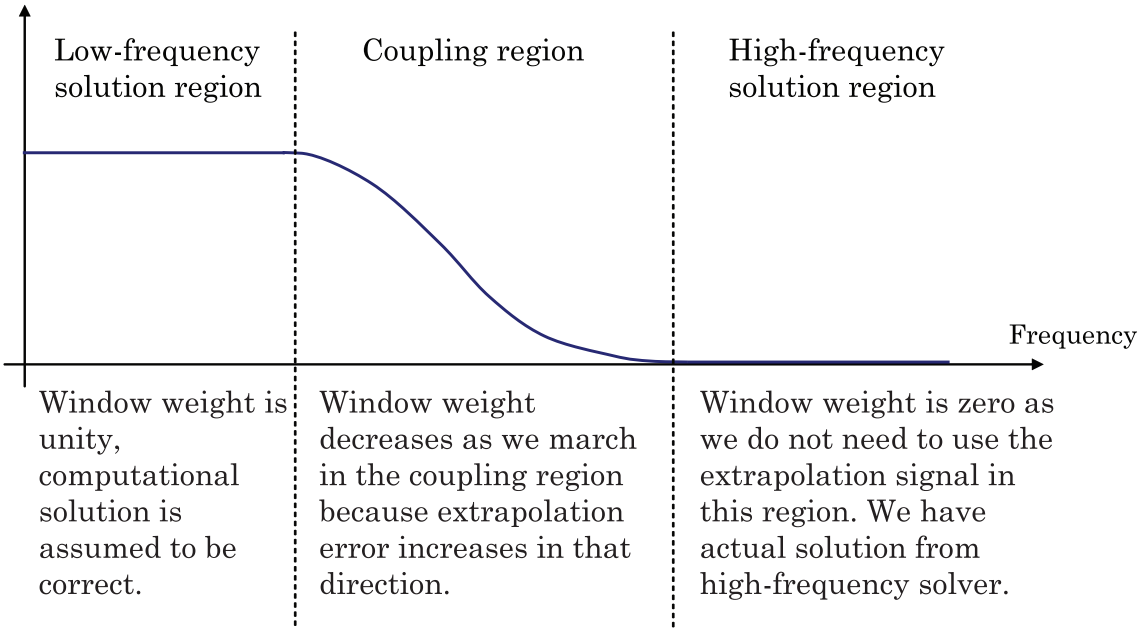


Figure 3.4: Reasoning for weights of FMPM.

flowchart in Figure 3.5 summarizes the CDMPM procedure.

Selection of window type in Figure 3.3 is a study on its own; however, we observed that the widely known window templates such as Blackman–Harris, Hamming, Triangle, Hanning, Chebyshev do not alter the result very much. We choose to use Hanning window throughout this study, without retrying every window for each scattering problem we would like to solve.

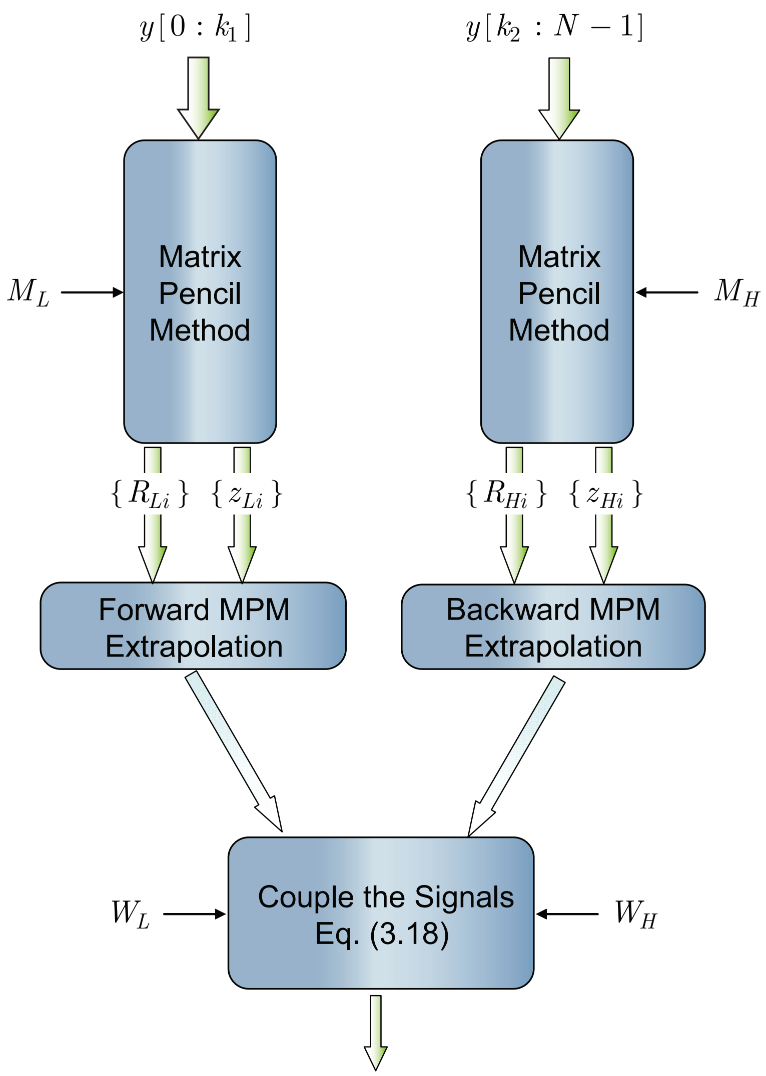


Figure 3.5: Flowchart of CDMPM.

3.4 Scattering from a Conducting Sphere

In this section, the results of coupled extrapolation performed on the conducting sphere problem are presented. In order to make a logical comparison with the results presented in Section 2.3, same LF and HF regions are used. First we present the results of COMPM and continue with CDMPM.

3.4.1 Sphere Results with COMPM Extrapolation

A similar choice of regions as in Section 2.3 will lead us to designate 1–5 GHz region as LF region and 25–35 GHz region as HF region. Using a frequency increment of 0.04 GHz we have 101 LF data sample points and 251 HF data sample points. This limits the M_L and M_H value to 50 and 125 according to (2.39) respectively.

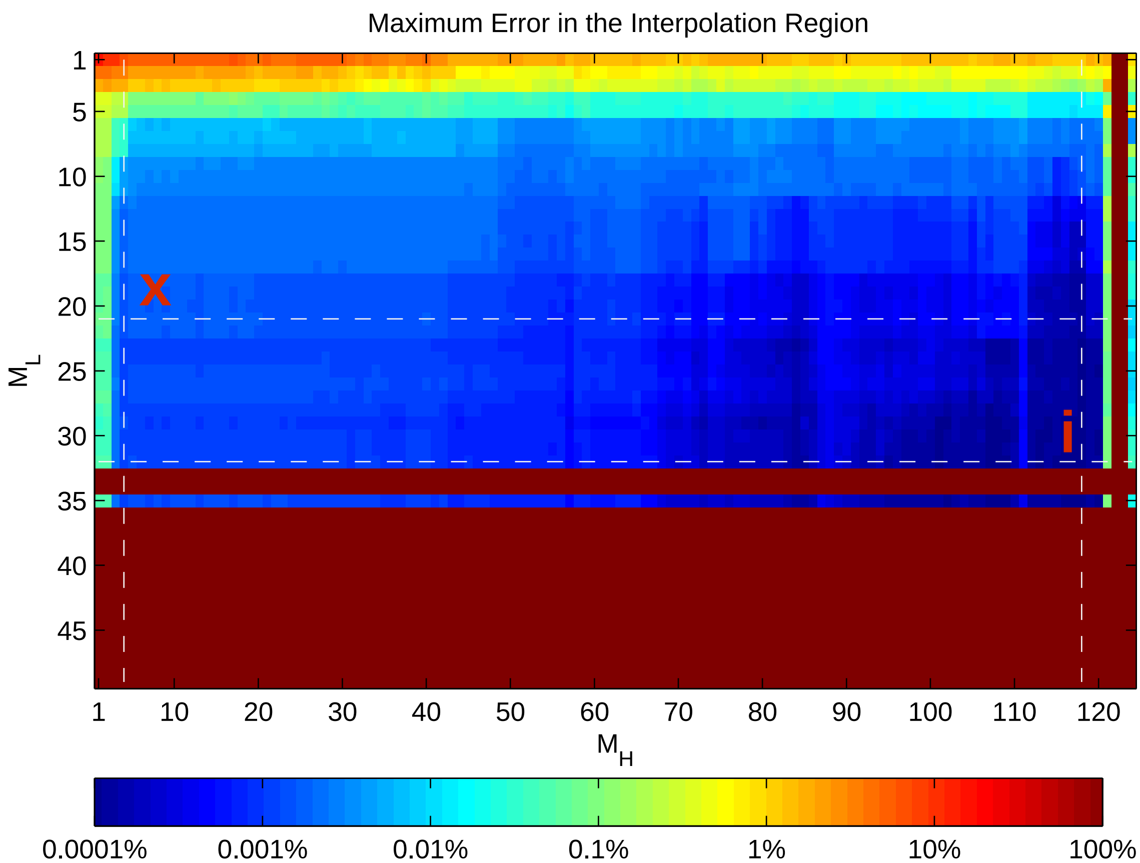


Figure 3.6: Maximum error in the IR.

We conduct (M_L, M_H) optimization using brute-force M optimization discussed in Section 3.2.1. In order to test the effectiveness of this optimization algorithm, we select minimum of maximum errors by looking at both IR error and ER error. In IR error, we fill an $M_L \times M_H$ matrix with maximum error value between extrapolated signal and the original analytical solution in the IR, i.e., the combination of intervals $k = [0 : k_1]$ and $k = [k_2 : N-1]$. In addition, we also fill a same size matrix with the maximum error value between extrapolated signal and the original analytical solution in the ER, i.e., in the interval $k = [k_1, k_2]$. In real-life cases, where we do not have the solution signal over the entire spectrum, we only have the IR error matrix, and decide the best (M_L, M_H) pair accordingly.

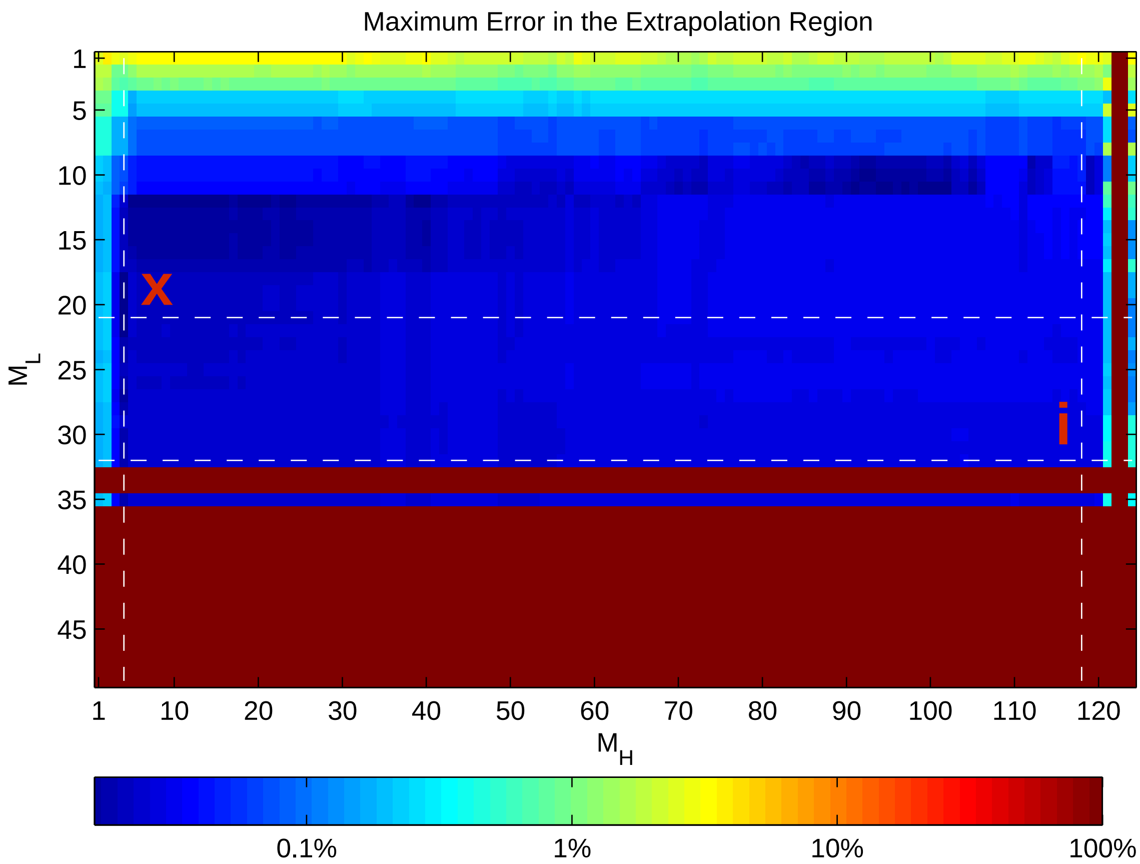


Figure 3.7: Maximum error in the ER.

IR and ER error matrices are presented in Figures 3.6 and 3.7 respectively. In these figures **x** represents the minimum of maximum errors with respect to ER error matrix and **i** represents the minimum of maximum errors with respect to IR error matrix. They indicate same locations in the two figures; **i** is the minimum for Figure 3.6 and **x** is the minimum for Figure 3.7. Notice that IR maximum error values are smaller than those of ER errors in general, which is actually an expected phenomenon. Table 3.1 summarizes the M_L and M_H values, the minimum of maximum extrapolation error values for both cases.

Although (M_L, M_H) pair is very different for both cases, what one should

	min(max(IR error)) (minimum in Figure 3.6)	min(max(ER error)) (minimum in Figure 3.7)
M_L	32	21
M_H	118	4
minimum IR error	8.86×10^{-7}	1.35×10^{-5}
minimum ER error	2.83×10^{-4}	1.27×10^{-4}

Table 3.1: Brute-force optimization results for COMPM

compare is the last row of Table 3.1. The last row implies that, we would obtain a maximum ER error of 2.83×10^{-4} by using the $(32, 118)$ pair instead of the absolute minimum of maximum ER errors, 1.27×10^{-4}, which could be obtained by using $(21, 4)$ pair. The difference between the two maximum error values is not very large, infact it is small enough to state that selecting (M_L, M_H) pair by finding the minimum of maximum ER error matrix that is constructed with respect to IR is good enough for optimizing the M pair.

Continuing as if we are solving a real-life problem, we choose $(M_L, M_H) = (32, 118)$ and construct the model as described in Section 3.2. The error on the extrapolated signal at each frequency is shown in Figure 3.8. When compared with the extrapolation results in Section 2.3, COMPM clearly reflects a great improvement on the accuracy of the extrapolation process. The maximum error value is below 10^{-3}, i.e., below 0.1% maximum error with the reasoning that signal converges to unity.

In order to increase the accuracy of the extrapolation relative to the results of FMPM and BMPM we trade the execution time and accuracy in the IR. The brute-force M optimization for COMPM takes significantly more time, when compared with the one described in Section 2.2.3 due to having two degrees of freedom,. The increase in the error values in interpolation region is simply because we start to use extra exponentials in the model and when calculating the coupled residuals, least squares approach tries to fit the model to both IRs at the same time.

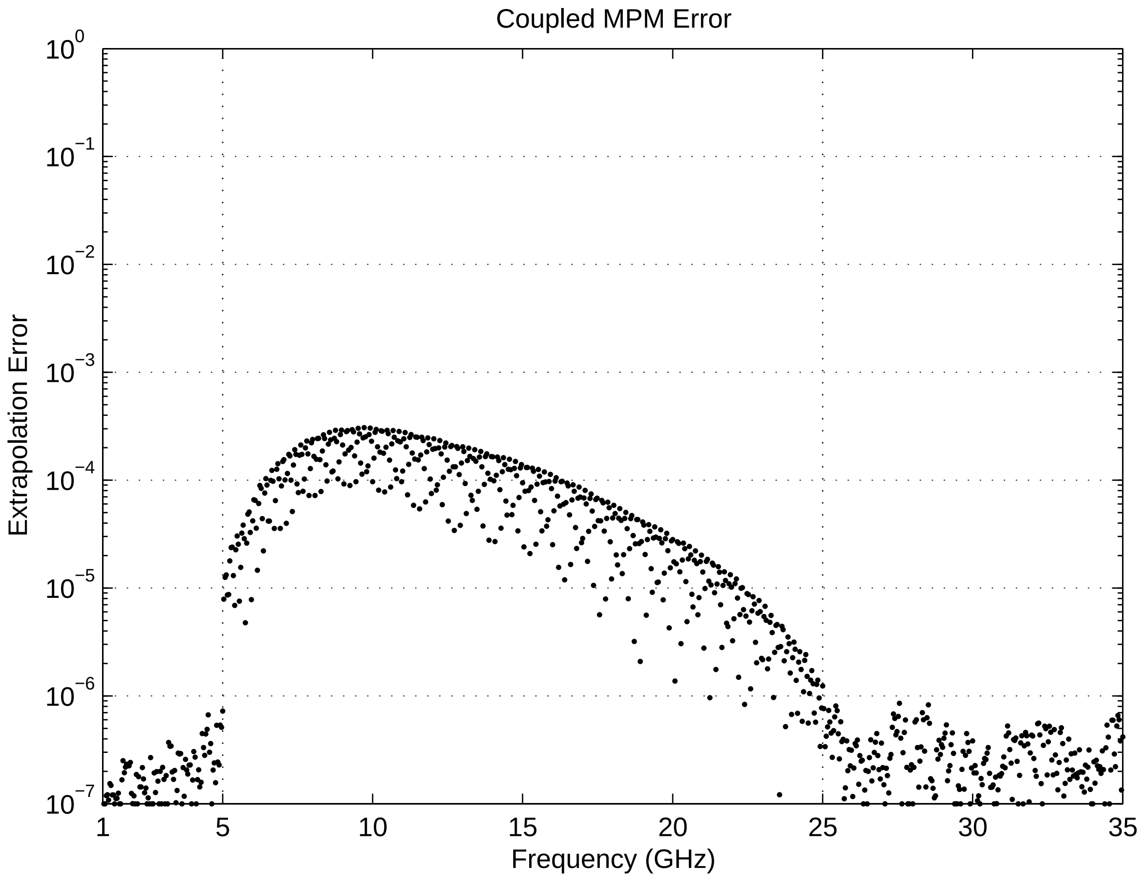


Figure 3.8: Extrapolation error for COMPM.

As stated for one-dimensional optimization in Section 2.3, the result of brute-force optimization for COMPM provides us the absolute minimum of maximum IR error values. However, it is clear in Figure 3.6 that, the point indicated with **i** can indeed be the absolute minimum but there are other points, i.e., M pairs, whose corresponding maximum IR error value is similar or very close to the absolute minimum; notice the large blue and dark blue regions in the figure. In the next chapter we will discuss a better way to perform this optimization in detail; this will also decrease the execution time on the side.

3.4.2 Sphere Results with CDMPM Extrapolation

In this section, we will present the results of applying CDMPM on the extrapolated signals of Figures 2.7 and 2.9. We follow the procedure that is described in Section 3.3. Figure 3.9 illustrates Hanning windows that are used in the process.

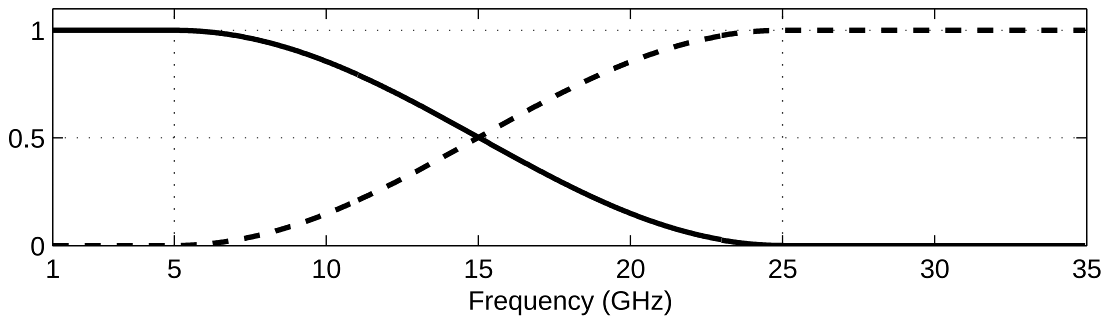


Figure 3.9: Hanning windows used in CDMPM.

The result of coupling the two signals is shown in Figure 3.10. When compared with the extrapolation result of COMPM that is presented in Figure 3.8, CDMPM has slightly higher error levels, but still lower than 10^{-3}, i.e., lower than 0.1%. However, as the process of coupled deconvolution does not include any new model construction or M optimization, this approach is extremely fast when compared with the COMPM approach. On the other hand, the result we obtain by CDMPM is almost similar to COMPM.

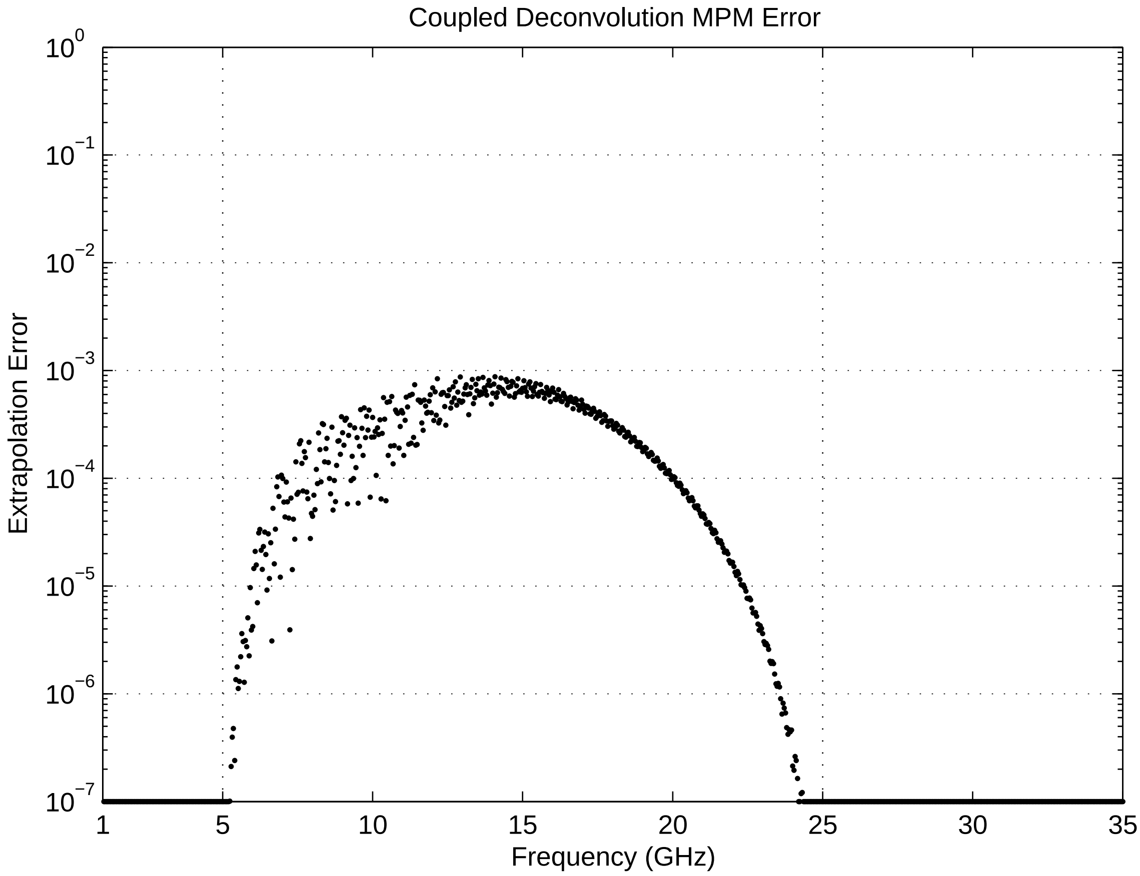


Figure 3.10: Extrapolation error for CDMPM.

Chapter 4

Working with Computational Data

Until this point we have worked on synthetically generated sphere data. This data is obtained by evaluating closed-form equations and solution is available for every desired frequency. This sphere data is used to test the proposed extrapolation methods, but eventually we have to move on to the real-life data cases. The major difference between computational and analytical data is the frequency limitation of the computational solution. Recall from Chapter 1, that we could obtain LF accurate solution data up to a frequency, which is related to the solution technique that is used and the computer resources available to the solver. Similarly, we could obtain HF solution data in a limited high frequency band, whose width is related with the desired accuracy of the solution. The aim of this research and thesis is to estimate the solution data in the region where no appropriate solution technique is present. It has been shown that methods that are proposed in Chapters 2 and 3 have overcome this problem on analytically generated data. However, when we move to the computational data cases, we observe that there are further differences between computational and analytical generated data, which make it more difficult to construct a good enough model and perform extrapolation. In this chapter, we will discuss the problems that we encountered when working with computational data and possible solution approaches to this problems.

4.1 Computational Noise and Error

The next logical step after working with analytically generated sphere data is the computational sphere data. For this purpose we acquired a LF solution data from FMM and HF data from PO. Figures 4.1 and 4.2 compare the analytically-generated data and computational data. In these figures, we illustrate the difference in the magnitude of the backscattered field, i.e., θ component of the electric field. During the modeling and extrapolation process we use complex field signals due to the fact that the model we are constructing is designed to work with complex signals. Of course there is no restriction on modeling real signals, but when we take the magnitude of a complex signal and convert it to a real signal we are almost disregarding half the information that complex signal possesses, i.e., we are disregarding the phase information.

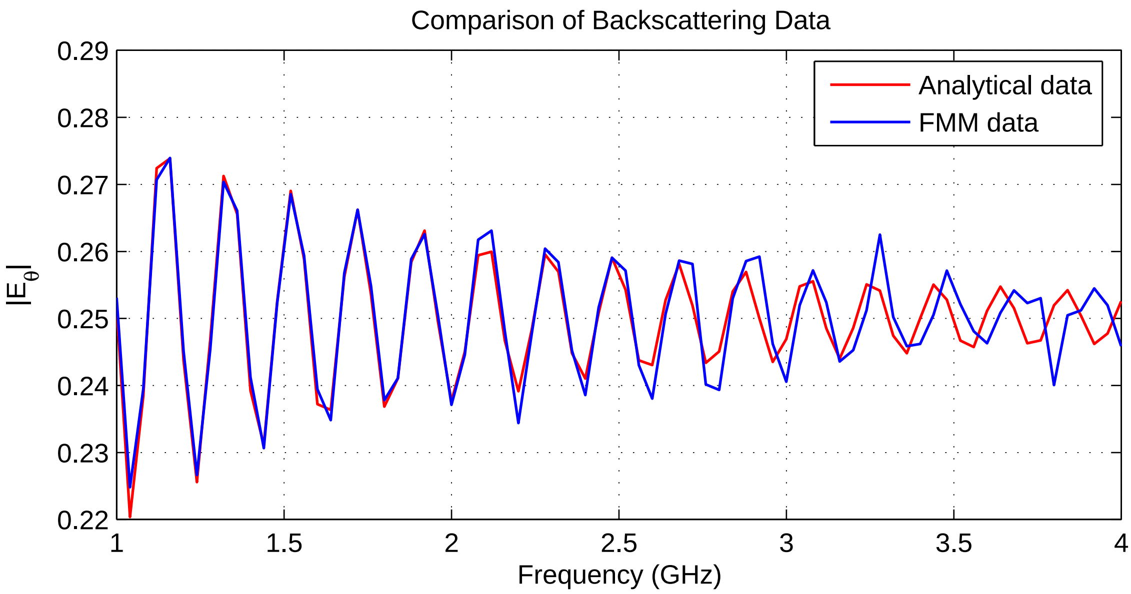


Figure 4.1: Comparison of analytical and FMM data.

It can be clearly seen from both of the figures that analytical and computational data do not match. We concluded there are two quantities that contribute to this deviation:

1) *Computational error* is a quantity related with the assumptions and approximations made at the beginning of the solution process. They may or may

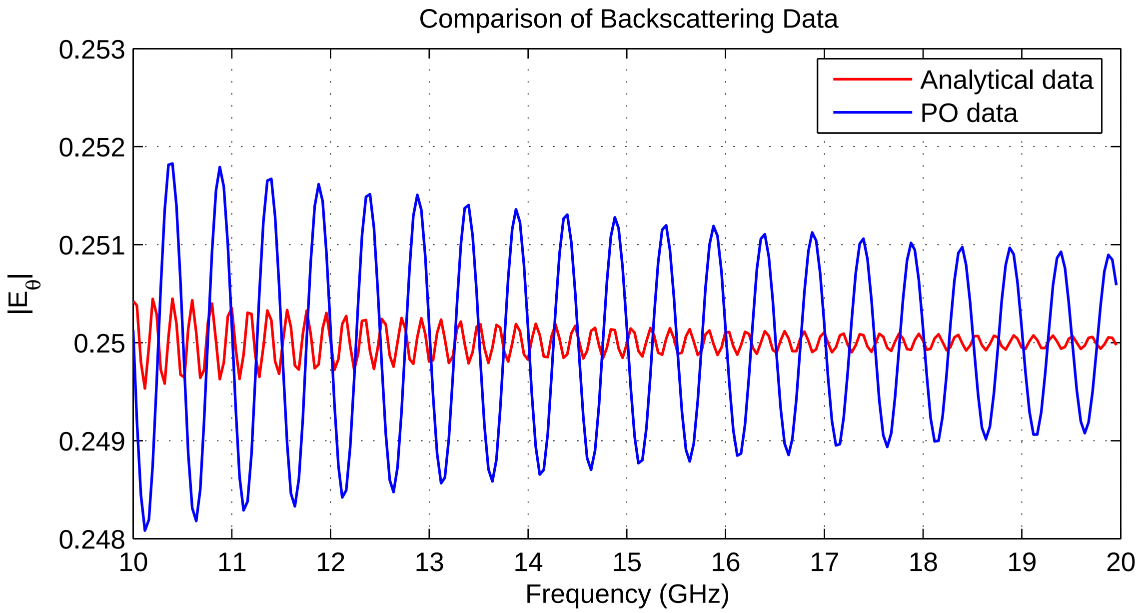


Figure 4.2: Comparison of analytical and PO data.

not be controllable during the solution process. For example in PO, HF approximations of electromagnetics forms the basis of the solution technique, however on the side it introduces an uncontrollable error to the solution. On the other hand, discretization of the geometry (meshing) or number of terms used in the series expansions introduce controllable error in the solution signal.

2) *Computational noise* is the quantity that arises from numerical computation of integrals, solution of dense and large matrix systems and working with very large or very small numbers etc. This noise may depend on the implementation of the solver, compiler being used or platform on which the solution is performed.

Although their sources are different, when introduced on a clean signal it is often very hard to discriminate computational noise and error.

Analytical data in Figures 4.1 and 4.2 contains a computational error of 10^{-6}. This error is introduced by the truncation of spherical harmonic series expansion at a certain level. The level of this error is controllable and in fact is an input parameter to analytical solver. On the other hand, computational noise on analytical data is so small that it is almost insignificant when compared with a computational error of level 10^{-6}. This is due to the fact that computational operations that are performed within the analytical solver are simply evaluating basic functions for some specific arguments.

It is clear from Figure 4.1 that FMM data do not agree with the analytical solution. Eventhough we deliberately introduce a 10^{-6} level computational error to the analytical solution, it is still assumed to be exact up to the sixth decimal digit. The difference between FMM and analytical data in Figure 4.1 is far worse than the error on analytical data; it has in fact difference in third decimal digit. Although there should be some significant computational noise on the FMM data due to numerical computation of integrals and huge and dense matrix manipulations and solutions, computational error dominates it. Computational error on FMM for this problem probably arises from the fundamental assumptions that are made at the beginning of the solution procedure. This leads to almost a completely different solution signal as solution frequency increases. Recall that FMM produces scattering solution for a given frequency, where solution process is independent of the solution at other frequencies. Therefore, it is easy to conclude from Figure 4.1 that the assumptions made at the beginning of the solution process holds pretty good for very low frequencies, such as the frequency band of 1–2 GHz; however as the solution frequency increases FMM tends to produce a completely different solution signal.

For PO, the last of the computational signals, the situation is not getting any better. Figure 4.2 shows a clear difference between HF approximate solution of PO and the analytical data. Although magnitude span of these two solution signals are similar, their oscillatory behavior is completely different. The scattering solution in PO is basically obtained by evaluating closed-form expressions, so we do not expect significant computational noise on this data. However, the computational error is present and clearly visible. Once again, as in FMM case, this error is the result of assumptions and approximations that are made at the beginning of the solution process.

After these observations on Figures 4.1 and 4.2 it is better not to continue the extrapolation process with the numerical solution of scattering from conducting sphere. At this point, one should note that these observations solely depend on the backscattering solution of the conducting sphere. For any other geometry, computational error and noise levels may not be as discussed above, but this does not change the fact that they exist. In addition, there is no other geometry that has an analytical solution (except 3-D cylinder, which has a very similar solution to sphere and has been skipped to avoid redundancy) so that we can make such a comparison. In the next sections we will continue to use numerical solution of conducting sphere but eventually this data will not be used to perform extrapolation.

4.2 Distribution of Complex Exponentials

In the previous section we compared the signals of computational and analytical data. In this section we take one step forward and compare the models of computational and analytical data that are constructed by using MPM. In order to analyze the differences between the models we will compare the complex exponentials, $\{z_i\}$, which will be used to construct the model. According to (2.35), residuals are calculated according to complex exponentials and the data values, therefore, we can conclude that complex exponentials contain most of the information about the model and the extrapolation process.

Figure 4.3 illustrates the complex exponential distribution of analytical data between 1–6 GHz. We illustrate the $M = 40$ case, where 40 exponentials will be used to construct the model. Furthermore in Figure 4.4 the $M = 30$ case for 1–4 GHz analytical data is presented. Notice that the exponential groups that are indicated by red ellipses have very similar locations in both figures. From this we can conclude that the intrinsic characteristics that the model acquires from the solution signal does not depend on the length of the signal or number of exponentials used in the model. This statement has been tested and validated for other intervals of the analytical backscattering solution signal of sphere and other values of M; these results are skipped to avoid redundancy.

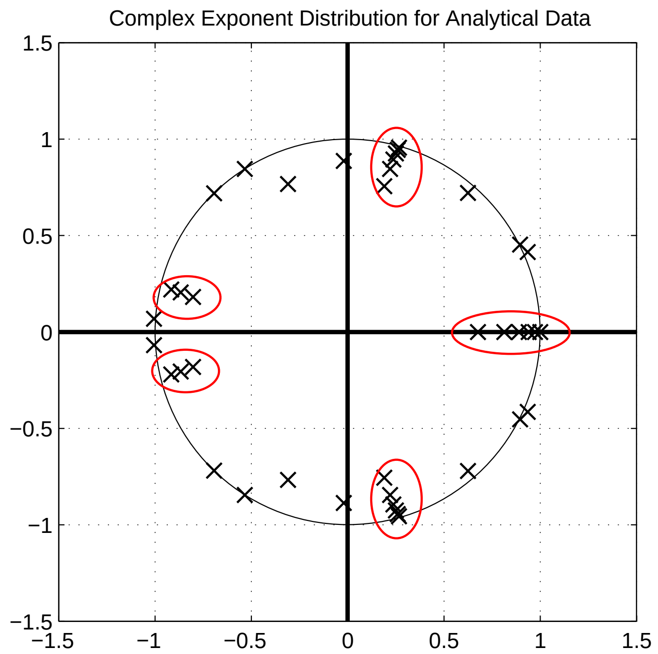


Figure 4.3: Distribution of complex exponentials for 1–6 GHz analytical data, $M = 40$.

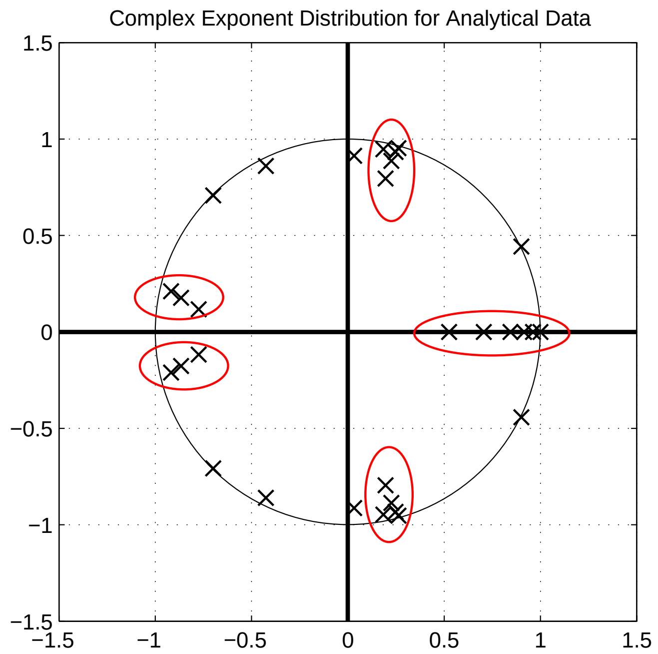


Figure 4.4: Distribution of complex exponentials for 1–4 GHz analytical data, $M = 30$.

After this conclusion, one should expect to observe a similar behavior for the computational data cases. However, by considering the results from the previous section, an educated guess would be the opposite. As expected, distribution of complex exponentials for computational data appears to be completely different. Figure 4.5 illustrates the complex exponential distribution for the FMM data between 1–6 GHz with $M = 40$, i.e., FMM counterpart of Figure 4.3. The first thing one should notice is that, there is no such grouping of complex exponentials as in the case of analytical data. Furthermore, the complex exponentials appear to localize around but mostly inside the unit circle. Some complex exponentials slightly appear outside the unit circle, i.e., have a magnitude that is greater than one. This type of exponentials might degrade the performance of the modelling and extrapolation since, while performing extrapolation, k index is used as the power of these exponentials and for increasing values of k, z^k would blow up. Complex exponential distribution of other data intervals and M values for FMM data have similar outcomes, i.e., scattered complex exponentials around the unit circle and no sign of grouping.

When we move on to the complex exponential distribution of PO data, which is illustrated in Figure 4.6, we see a similar situation as in the FMM case. The complex exponentials are scattered in the vicinity of the unit circle and there is no resemblance of the grouping that is present in the analytical case. These results are expected after the conclusion drawn in the previous section. The models that we construct for FMM and PO data have no relation with the model we construct for the analytical data in the same problem. Apart from this conclusion, we also observed a very important fact that, when dealing with computational data, there is a possibility that magnitudes of some complex exponentials can exceed unity. In order to avoid such instances, we can simply constrain the magnitude of the exponential; however, changing the value of the exponential without referring to the data may not be the best solution. Later in this chapter, we propose a modified modeling scheme for MPM that eliminates these blow-up situations.

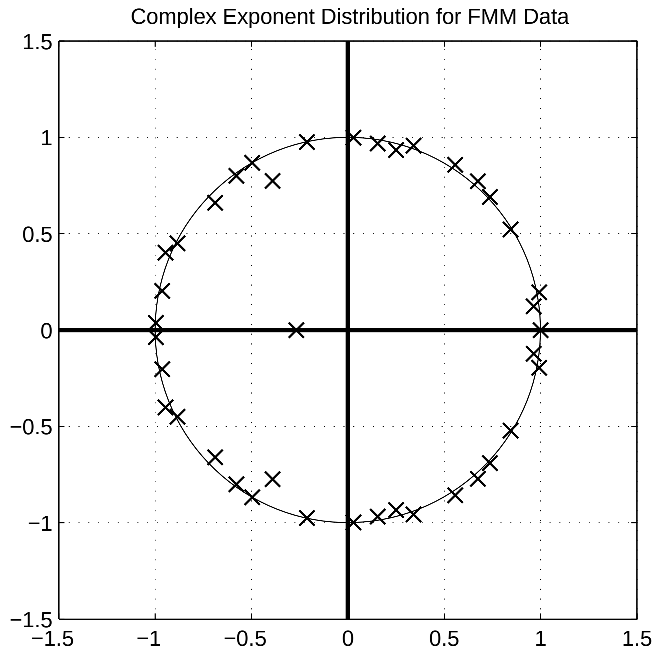


Figure 4.5: Distribution of complex exponentials for 1–6 GHz FMM data, $M = 40$.

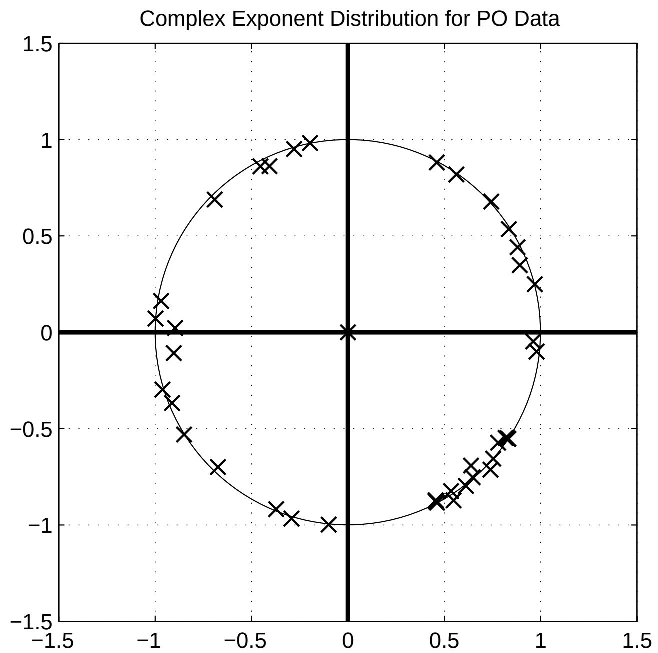


Figure 4.6: Distribution of complex exponentials for 10–15 GHz PO data, $M = 40$.

4.3 Observing Noise on Data

In the beginning of this chapter, we defined two quantities, computational error and noise that are possibly added to the solution signal and degrade its quality. We also discussed that once these quantities are added to the signal it is very hard to discriminate them in the noisy signal. Therefore, from this point on, we do not try to discriminate and identify those two quantities but simply refer to them as noise on the signal. However, one should keep in mind that the noise we are trying to estimate and hopefully isolate is mostly computational noise. It should be noted that by using signal processing and modeling techniques, it is almost impossible to isolate and remove computational error like the one on FMM and PO as seen in Figures 4.1 and 4.2 respectively. Such a correction should be sought inside the solution methodology and solution theory.

In this section, we will use two approaches to observe the noise on data. First we look at the distribution of singular values, one-step before the calculation of complex exponentials in MPM. Singular values, although remotely related with complex exponentials, carries as much information about the model as complex exponentials. Furthermore, for a given interval, singular values form a unique set as opposed to the complex exponentials set that change with number M. The second approach we use is, maximum interpolation error for every possible value of M. This plot will tell us how well the model estimates the known part of the signal.

Figure 4.7 illustrates the singular value distribution of FMM and analytical data. Notice that for the first few singular values, both data have the same distribution, but after some point, singular values of analytical data decreases with a much higher slope than the singular values of FMM data. After approximately 17 singular values, the magnitude of the singular values from analytical data also almost stops decreasing and tends to stay around some magnitude value. This plot, when combined with Figure 4.8, provides a valuable information about noise characteristics of analytical and FMM data. Figure 4.8 illustrates the maximum interpolation error for each possible value of M. Notice that in Figure 4.8 the maximum interpolation error does not decrease very much after $M = 17$,

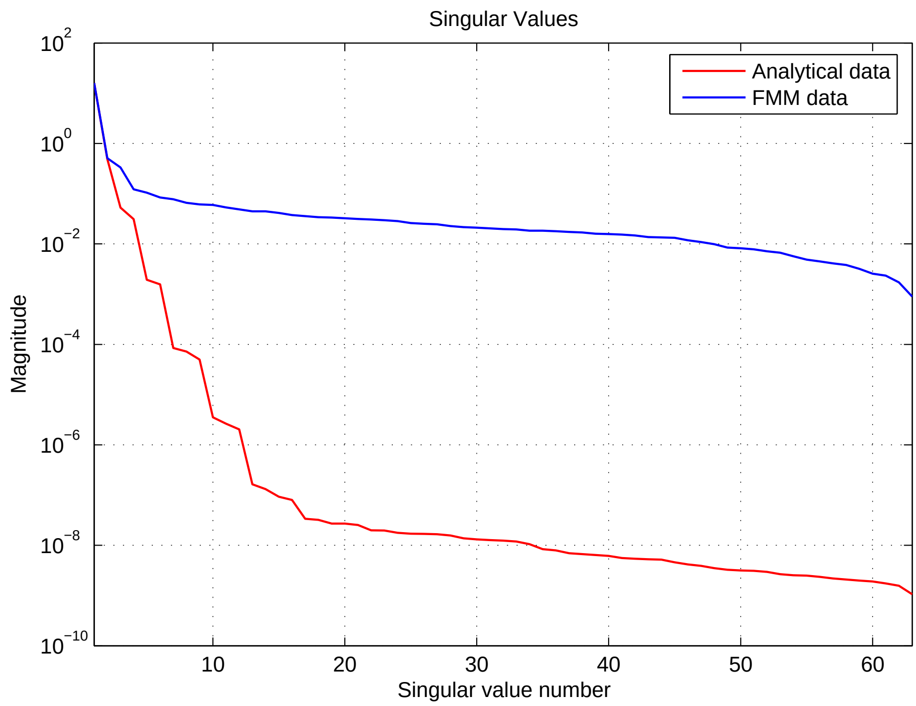


Figure 4.7: Singular value distribution for 1–6 GHz data, FMM and analytical data.

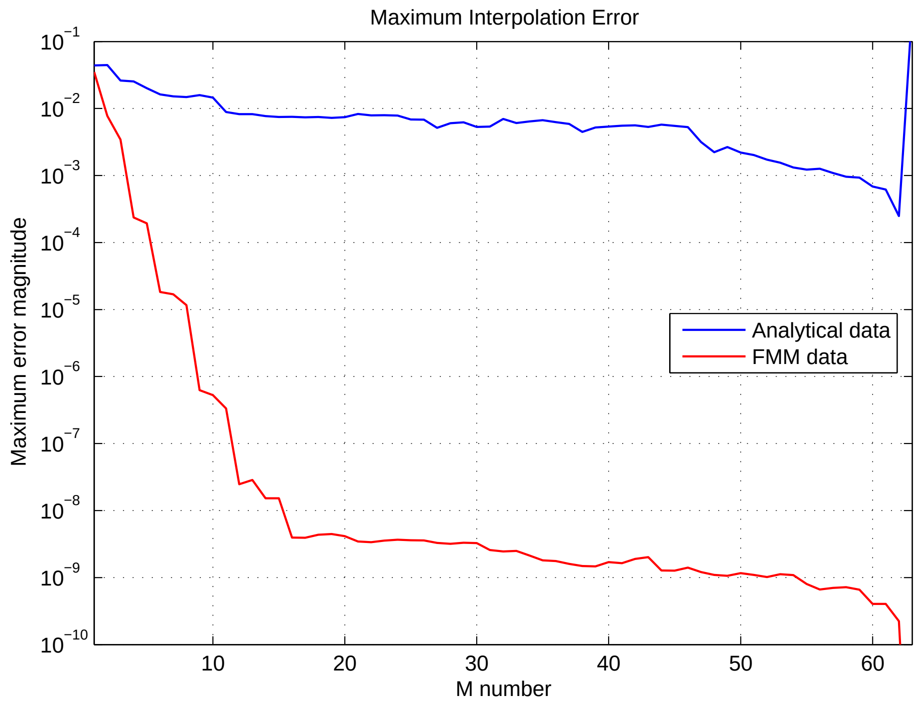


Figure 4.8: Maximum interpolation error for 1–6 GHz data, FMM and analytical data.

the same location of the break-point as in Figure 4.7. At this point, one must recall from MPM theory in Chapter 2 that truncation of singular values in $\bar{\mathbf{\Sigma}}$ in (2.20) is nothing but assigning a value to M. Therefore, it is expected that these two breakpoints on the red curve appear at the same point in the both figures. The thing that we should focus on is, why we can not decrease the interpolation error as we increase the number of exponentials used in the model. Actually we do observe a slight decrease in Figure 4.8 but according to the common sense of series expansion that is discussed in Chapter 2, we should observe a much steeper decrease on the error. The reason is simple; there exists some noise an analytical data (recall that it is around 10^{-6}), and after the 17th exponential, or simply 17th singular value, the model tries to estimate the noise. Because it is extremely difficult to model a noise, which in most cases is random, we observe a flat region on the interpolation error curve around the level of noise. A similar argument can be carried out for singular values too.

The same phenomenon is observed for the blue FMM line but as the range of the magnitudes of singular values and maximum error magnitude is small, it can not be seen clearly. For the FMM data with much less error, i.e., FMM data for other problems, this could be observed clearly, which will be illustrated in the next chapter.

In order to test this hypothesis about the relation between the value of the flat region in the maximum interpolation error plot and the noise on data, we add synthetic noise on a relatively clean analytical data. Figures 4.9 and 4.10 show the singular value distribution and maximum interpolation error of such tests respectively. Here we added a 10^{-3} level complex noise to the analytical signal and look at deviations in singular value distribution and maximum interpolation error.

Singular values of noise added data behave very similar to FMM data and we achieve a flat region in the maximum interpolation error plot at exactly 10^{-3}. The results of this test proved the correctness of our hypothesis and actually provide us with a method to estimate the noise level on any signal.

Before moving on to the methodology of estimating noise on computational

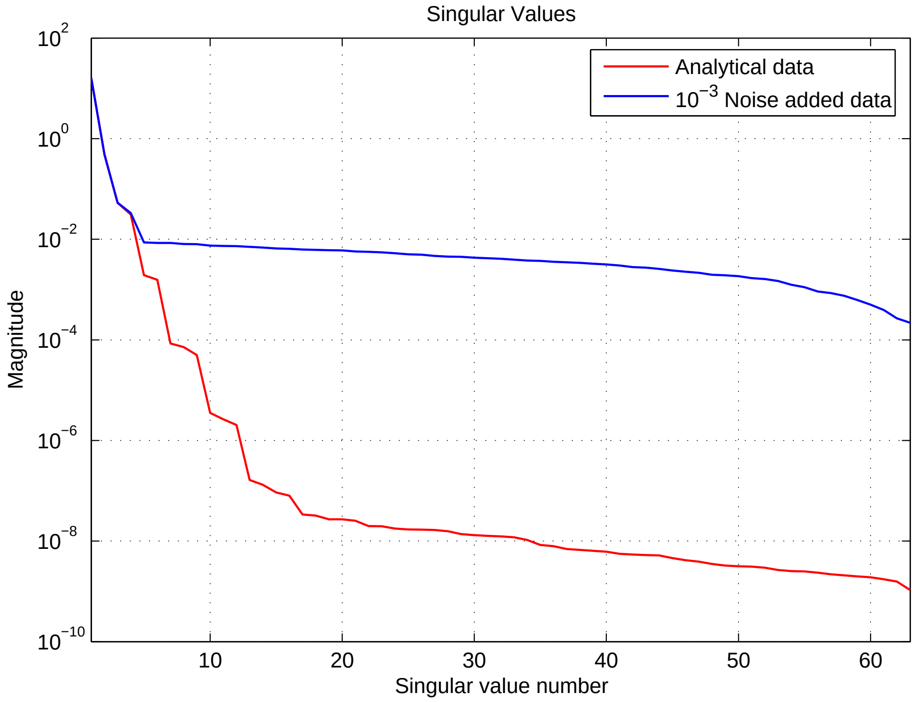


Figure 4.9: Singular value distribution for 1–6 GHz data, analytical and noise added data.

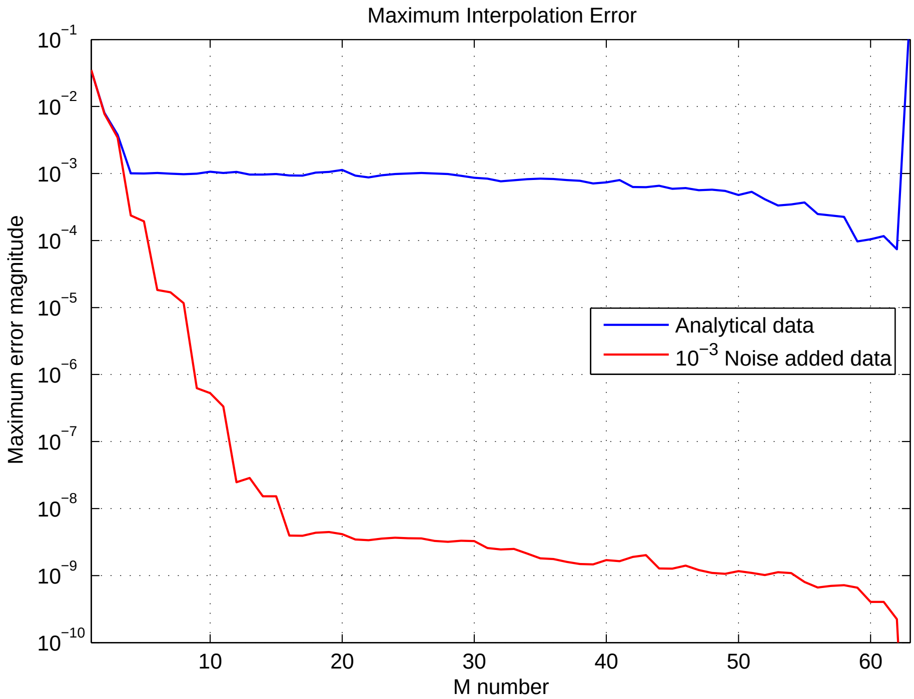


Figure 4.10: Maximum interpolation error for 1–6 GHz data, analytical and noise added data.

data, we should also observe the noise behavior of PO data. Figures 4.11 and 4.12 illustrate the singular value distribution and maximum interpolation error of PO data respectively. Notice that the magnitude of singular values in Figure 4.11 decreases very rapidly and a flat region occurs at a very low value. A similar behavior is also observed for maximum interpolation error. From these two figures we can conclude that the noise on PO data is very small and we can model the data with great accuracy. This of course does not mean that we are producing good results for the sphere problem. What we are modeling with great accuracy is the erroneous signal in Figure 4.2. However, as we will demonstrate for different PO solutions in the next chapter, the noise behavior of PO that we observe in Figures 4.11 and 4.12 is in fact a general behavior. The very rapid decrease of interpolation error, or equivalently singular values, will enable us to further constrain the M value which will dramatically decrease the execution time of the M-optimization in coupled extrapolation cases. This we will discuss in the next sections.

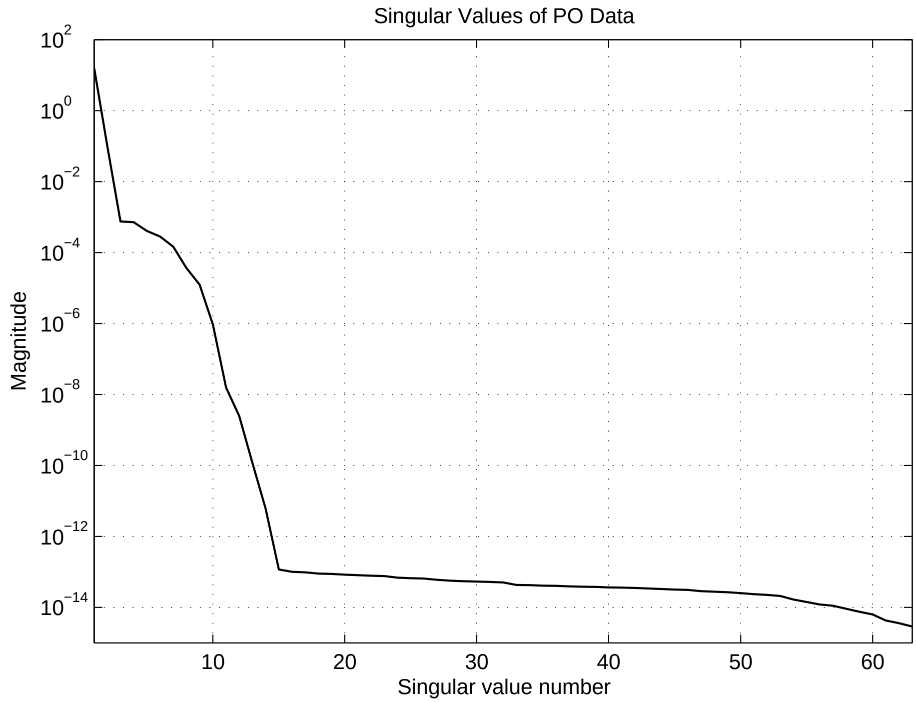


Figure 4.11: Singular value distribution for 10–15 GHz data, PO data.

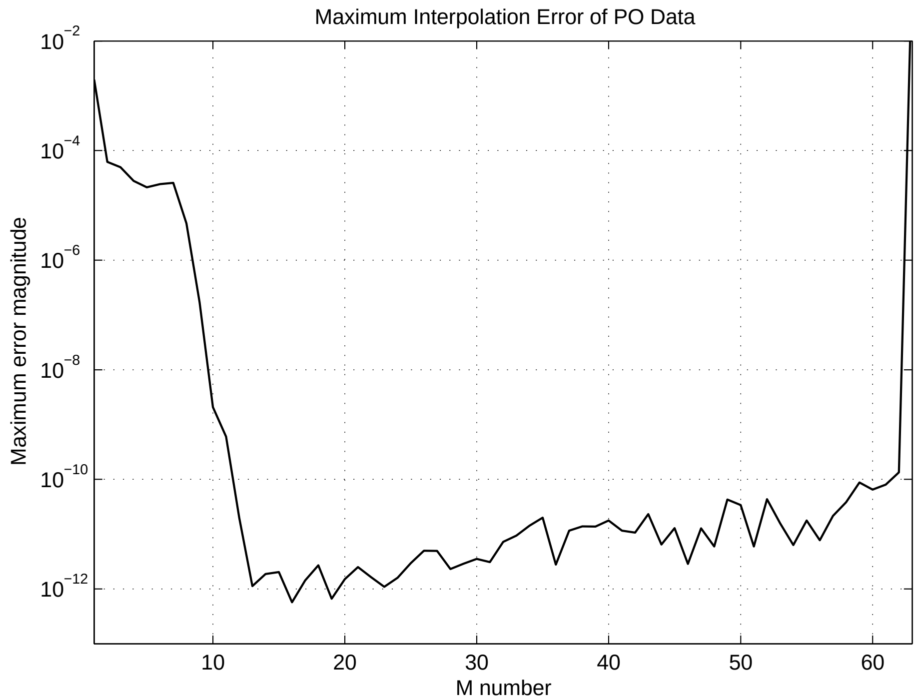


Figure 4.12: Maximum interpolation error for 10–15 GHz data, PO data.

4.4 Estimating Noise on Data

In order to demonstrate the noise estimation procedure, we should move on from the numerical solution data of conducting sphere. In this section, we use the backscattering solution of patch-over-ring geometry. Details of this geometry can be found in the next chapter, where we perform extrapolation tasks on the solution signal of this geometry. This should not give the reader any confusion, since this signal is just another computational data, albeit better than the sphere data. The maximum interpolation error plot of this signal is illustrated in Figure 4.13. By looking at this plot, it is difficult to determine a flat region that should indicate the noise level. In fact this is often the case for FMM data. Therefore, it is better to use a more generic approach to estimate the noise on computational data.

In order to achieve this goal, we systematically add synthetically generated random noise on the original data. The random noise we create can be written as,

$$N(k) = \frac{N_e}{\sqrt{2}}\{\mathrm{rand}(1).\mathrm{sign}(\mathrm{rand}(1) - 0.5) + i.\mathrm{rand}(1).\mathrm{sign}(\mathrm{rand}(1) - 0.5)\}, \quad (4.1)$$

where N_e is the envelope magnitude, rand(1) is the random number generator which produces a random value in the interval $[0:1]$ and sign( ) is the signum function, which simply outputs a 1, -1 or 0 according to the sign of the argument. The noise function (4.1) is designed to be completely random and independent for each value of frequency index k, similar to the actual computational noise case. The noise is created as a complex signal over frequency due to the fact that we are modeling complex electromagnetic field signals. By changing the envelope magnitude, such as $N_e = 10^{-1}, 10^{-2}, 10^{-3}, \ldots$ we create noisy signals for different levels of noise. The noise on the computational data is then estimated by observing the behavior of the maximum interpolation error plots of these noisy signals.

Figures 4.14, 4.15, 4.16 and 4.17 illustrate the maximum interpolation error plots of noise added patch-over-ring solution data for $N_e = 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}$ respectively. From the results of the previous section, it is expected that some interpolation error plots have flat regions and that the level of the flat region correspond to the added noise level. When we encounter a flat region as in Figures 4.14 and 4.15, it can surely be stated that the noise level on the computational data is smaller than the smallest level of noise added to the data in that figures. In our example it is certain that the noise on patch-over-ring data is smaller than 10^{-2}, and greater or equal to 10^{-4}. One should note that it is not so crucial to have the exact level of noise on the computational data. An estimate, such as around 10^{-3} in the case of patch-over-ring, would be enough. In the next section, we will use these results to find a new upper limit for M, which will both prevent us to model the noise and also decrease the execution time of M optimization in coupled extrapolation.

Figure 4.13: Maximum interpolation error, original data.

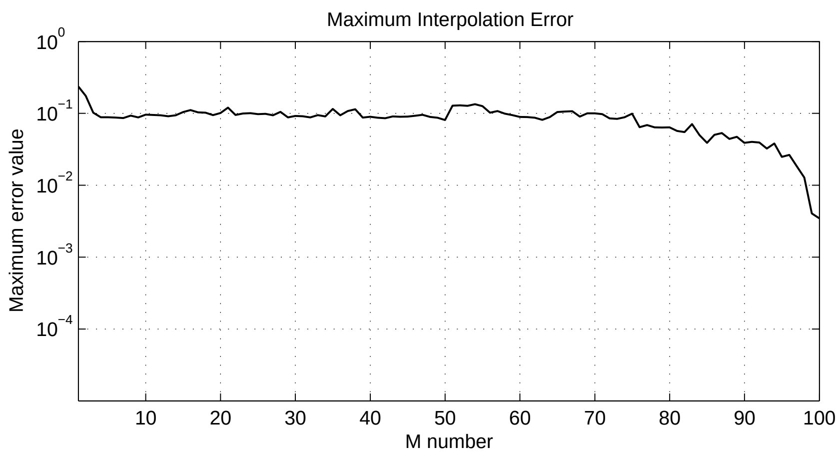


Figure 4.14: Maximum interpolation error, 10^{-1} noise added data.

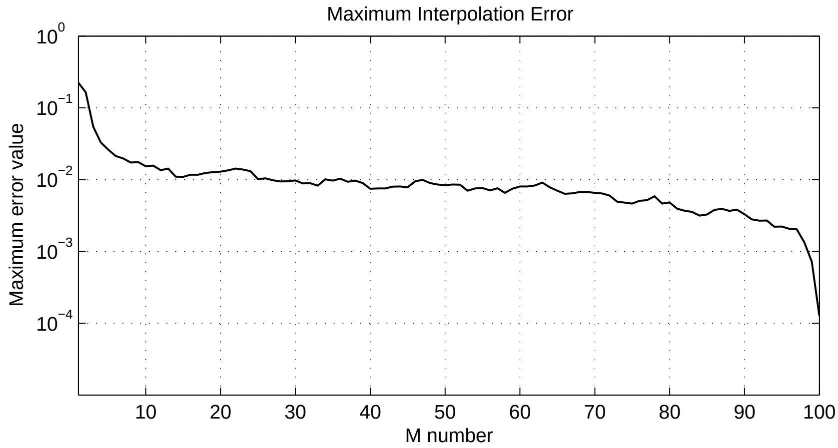


Figure 4.15: Maximum interpolation error, 10^{-2} noise added data.

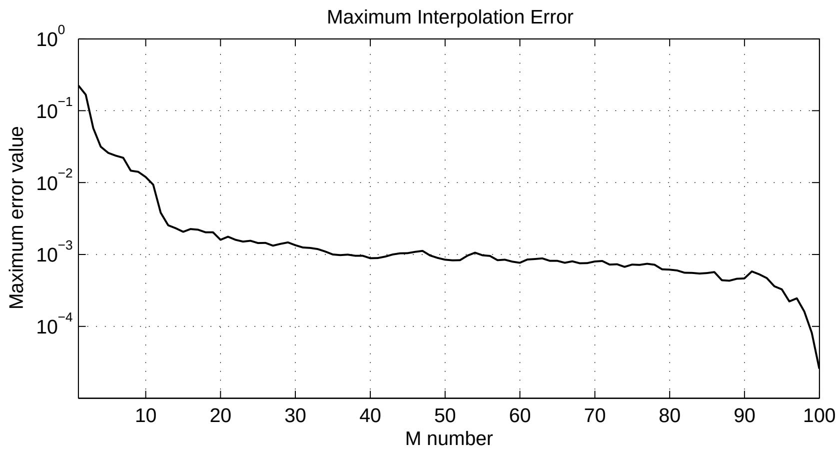


Figure 4.16: Maximum interpolation error, 10^{-3} noise added data.

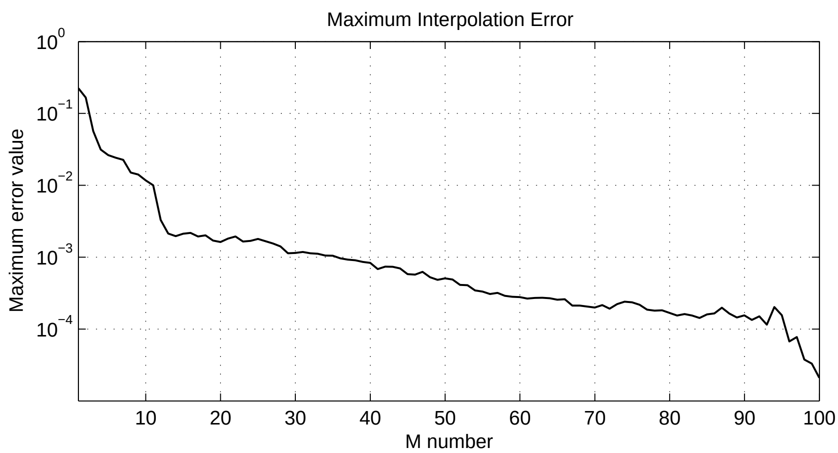


Figure 4.17: Maximum interpolation error, 10^{-4} noise added data.

4.5 A New Approach for the Selection of M Parameter

In the previous section we found an estimate for the noise level on the computational data by using the maximum interpolation error for different levels of noise added signals. In this section, we will combine these previous knowledge and observations to find a limit for M. This new limit should of course be smaller than the absolute theoretical limit of (2.39), i.e., $\lfloor \frac{N}{2} \rfloor$.

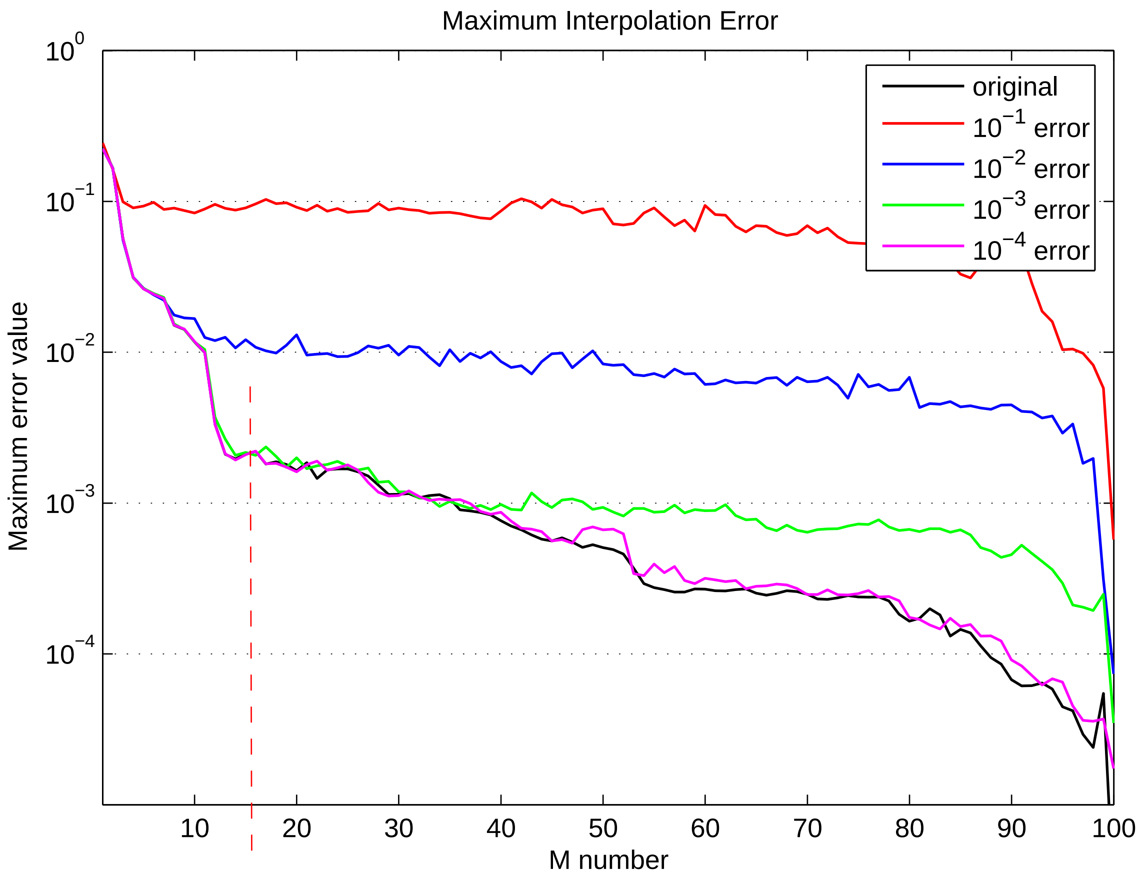


Figure 4.18: Maximum interpolation error for different levels of noise added data.

We would like to limit M, because after some point in the M value range, we start to model the noise that is present on the data. If we are able to keep modeling the noise as minimum as possible, we would in a way isolate the noise and obtain a model that better describes the clean data. Furthermore, if we able to find a lower limit value for M that is lower than (2.39), the brute-force optimization of coupled extrapolations would take much less time.

Figure 4.18 illustrates the maximum interpolation error plots for different levels of noise added signals. We have used the backscattering solution signal of the patch-over-ring geometry between 7–15 GHz. Figure 4.18 basically combines Figures 4.13–4.17 in a single plot, so that the deviations can be visualized more easily. The red vertical dashed line represents the proposed limit for M for this specific problem, which is approximately 15. This limit is selected such that for the M values that are smaller than the proposed limit the maximum interpolation error does not change much for 10^{-3} error level (green) and the original (black) curves; recall that we estimated that the error on the data is in the proximity of 10^{-3}. We should note that the selection of the limit is not an absolute pick, i.e., it is hand-picked without using any computational algorithm. Therefore, the limit could even be 14, 17 or even 30. However, as the limit value increases, the probability of modeling the noise increases; in addition, the small deviations in the limit value do not affect the extrapolation very much. Note that we are only redefining the limit, not selecting the actual M value to be used in the extrapolation. The actual selection of M value is still performed by the brute-force optimization of M that is discussed in chapter 2 and 3, albeit with lower maximum limits.

A similar approach can be followed to limit the M values for the PO data. However, this time we do not need to estimate the noise level on the data, which is determined to be very low in the previous sections. Observing the singular value distribution of PO data is sufficient for selecting a limit and even for selecting the actual number of exponentials that is needed to model the PO data in most cases. Figure 4.19 illustrates the singular value distribution of the patch-over-ring PO data. The first four singular values are significantly greater than the rest, in fact the ratio of fourth and fifth singular value is in the order of 10^{11}. This picture leads us to select the M limit and the actual value of M as 4. When the model, which consists of four complex exponentials, is compared with the original PO solution the modeling error turns out to be in the order of 10^{-5}.

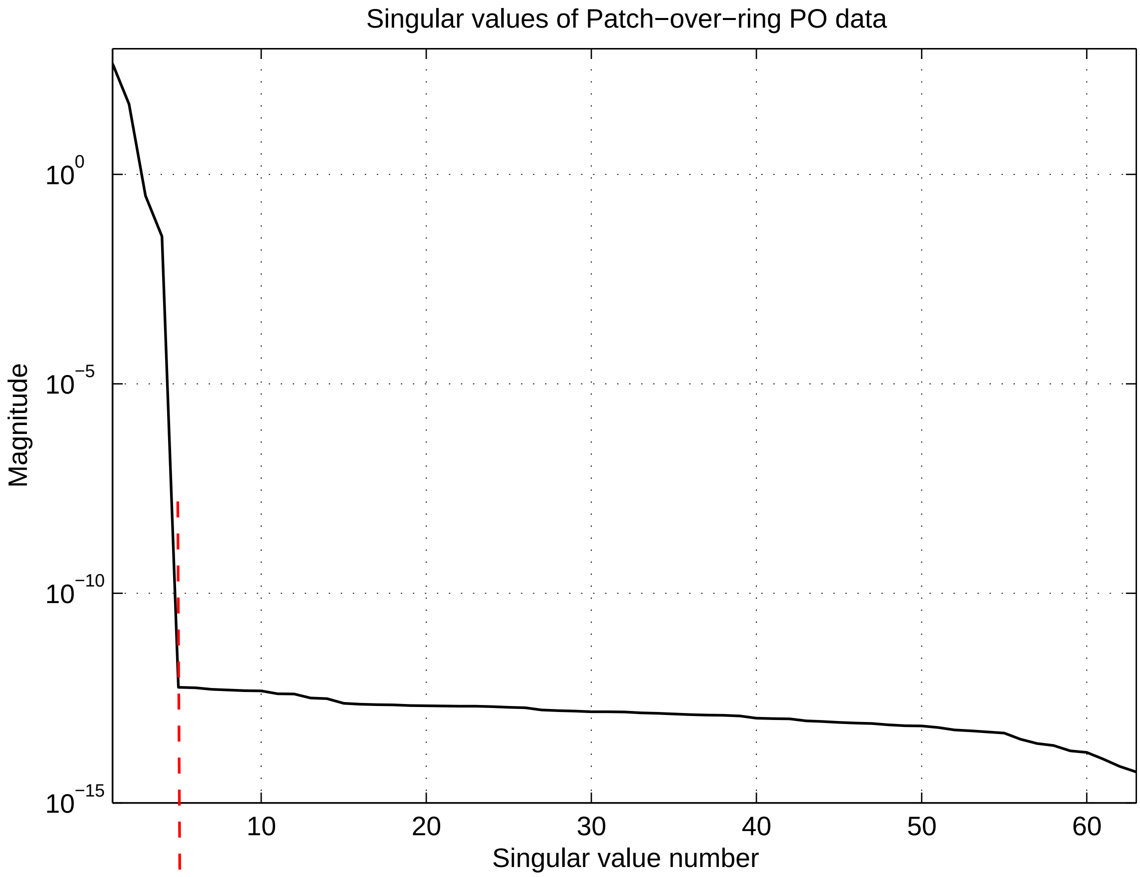


Figure 4.19: Singular value distribution of patch-over-ring PO data.

4.6 Coupled Extrapolation with Weighting

Up to this point, we compared the computational data and analytical data and observed the computational noise and error on the computational data. We proposed methods to estimate and isolate the noise, but working with computational data still needs some attention. When we are comparing the computational data with the analytical data, we skip the comparison of FMM and PO data. At an initial consideration, this may seem illogical and useless as FMM and PO data does not coincide in the solution spectrum and what we are aiming with this thesis is to the fill the intermediate gap. However, one should recall that although there is a hardware and software limitation on the maximum solvable frequency on FMM, the limitation on PO data is imposed externally, i.e., by manually defining an accuracy limit. Therefore we can compare the two computational data in the interval in which we have the FMM solution. One should not expect much from this comparison though; at that interval we already know that PO data is inaccurate. However, this inaccuracy should not change the signal completely. For example, Figure 4.20 shows the FMM and PO data in the interval 13–18 GHz for backscattering solution of patch-over-ring geometry. In patch-over-ring geometry, we have FMM solution up to 16 GHz; thus we compare the FMM and PO data in 13–16 GHz region in the figure. By looking at this figure one can easily see the inaccuracy problem on PO data. We should add a remark at this point that between FMM and PO, we refer FMM as the accurate, i.e., exact, data due to the fact that it contains far less approximations than PO data. The signals in Figure 4.20 should not cause any problem while performing extrapolation but this is not always the case. Figure 4.21 shows the backscattering FMM and PO solution signals for rectangular prism geometry. It can be easily recognized that the difference between FMM and PO data is far more than an accuracy issue on PO. These two different behaving signals at the opposite ends of the spectrum will definitely degrade the performance of the coupled extrapolations. In order to combat with this situation, a manipulation on COMPM extrapolation is proposed next, where we increase the influence of FMM on the extrapolated signal by introducing weighting.

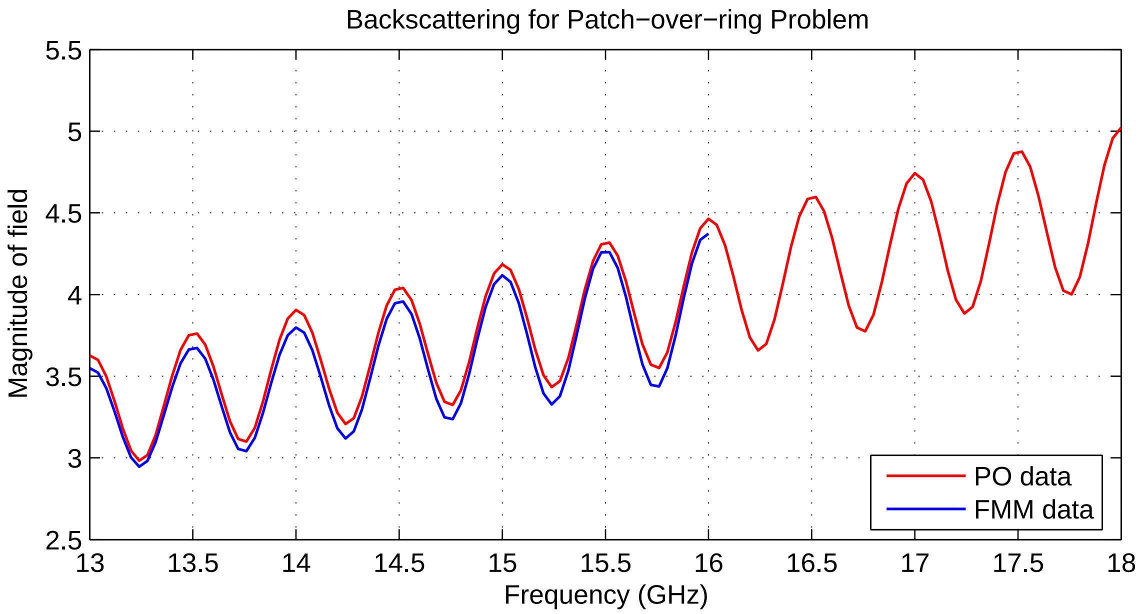


Figure 4.20: Backscattering solution signals of patch-over-ring geometry.

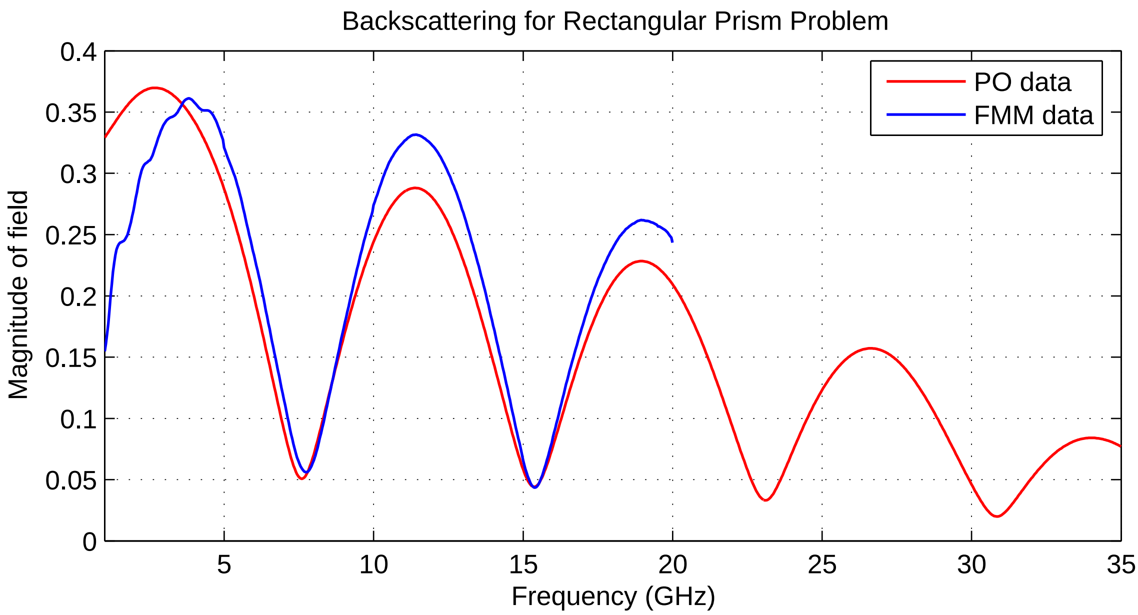


Figure 4.21: Backscattering solution signals of rectangular prism geometry.

4.6.1 Weighted COMPM (WCOMPM)

Although the weighting idea came up from the incompatibility between FMM and PO data, we can also use this phenomenon for a general computational data case because the accuracy advantage of FMM over PO is always valid.

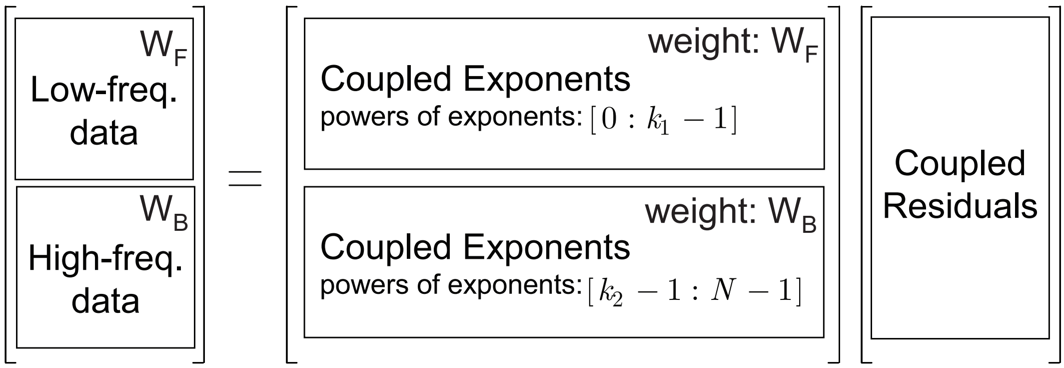


Figure 4.22: Schematic representation of WCOMPM.

In the regular COMPM extrapolation, the weights for each data source is assumed to be unity. This will give equal emphasis to LF and HF data, which is a logical choice for analytical data when mismatch is not present. The weighting is imposed into the equation (3.1) when the coupled residuals are being calculated. The weighting process is illustrated in Figure 4.22.

Assigning weights to the parts of such a matrix system should not change the individual equations of the system since both sides of the equation are multiplied by the same weighting factor. However, the overall system is an overdetermined system, which means that we should use a solution method such as least squares. Hence, weighting the individual parts of the matrix equation will give emphasis to a particular data according to the individual weights.

We define the weighting parameter as (W_F, W_B); W_F being the weight of FMM and W_B being the weight of PO. As the matrix equation has only two parts, what really matters is the ratio of the individual weights, i.e,

$$W_R = \frac{W_F}{W_B}, \tag{4.2}$$

W_R being the relative weight. This suggests that the weighting parameters (20,1) and (40,2) would produce the same least squares solution. Although weighting makes the extrapolation more accurate with respect to FMM it introduces a new parameter, W_R, to the process. This requires a whole new dimension added to the optimization procedure, but this will also add a huge computational burden. Therefore, in this study we simply used some major, predefined weights such as $W_R = 1, 2, 3, 4, 5, 10, 50, 100$ and handpick the best one. The results of WCOMPM will be shown in the next chapter along with other computational data results.

4.7 Weighted FMPM (WFMPM)

In the previous sections, we discussed that the complex exponentials whose magnitudes are greater than unity would cause problems in extrapolated signals. In this section we will try to resolve this issue by introducing weights to FMPM. We noticed that when working with computational data the results of FMPM often blows up. Figure 4.23 illustrates such a case where we forward extrapolated the backscattered FMM solution signal of the patch-over-ring geometry.

When we closely look at the model constructed and the complex exponential distribution of this data, we can confirm that this blow up occurs because of the complex exponentials whose magnitude is greater than unity. Such exponentials often do not occur in PO data; thus we will only discuss the weighting on FMPM; however the idea can easily be adapted to BMPM when necessary.

In the FMPM, we obtain the complex exponentials and their corresponding residuals from the LF data. The HF data is only used in brute-force M optimization. In most cases, this approach is not enough to avoid or eliminate the blow ups on the extrapolated signal when some complex exponentials have magnitudes greater than unity. Therefore we would like to incorporate the HF data into the model-construction stage and force the calculation of residuals to consider also the HF data when fitting the model to the signal. Figure 4.24 illustrates the

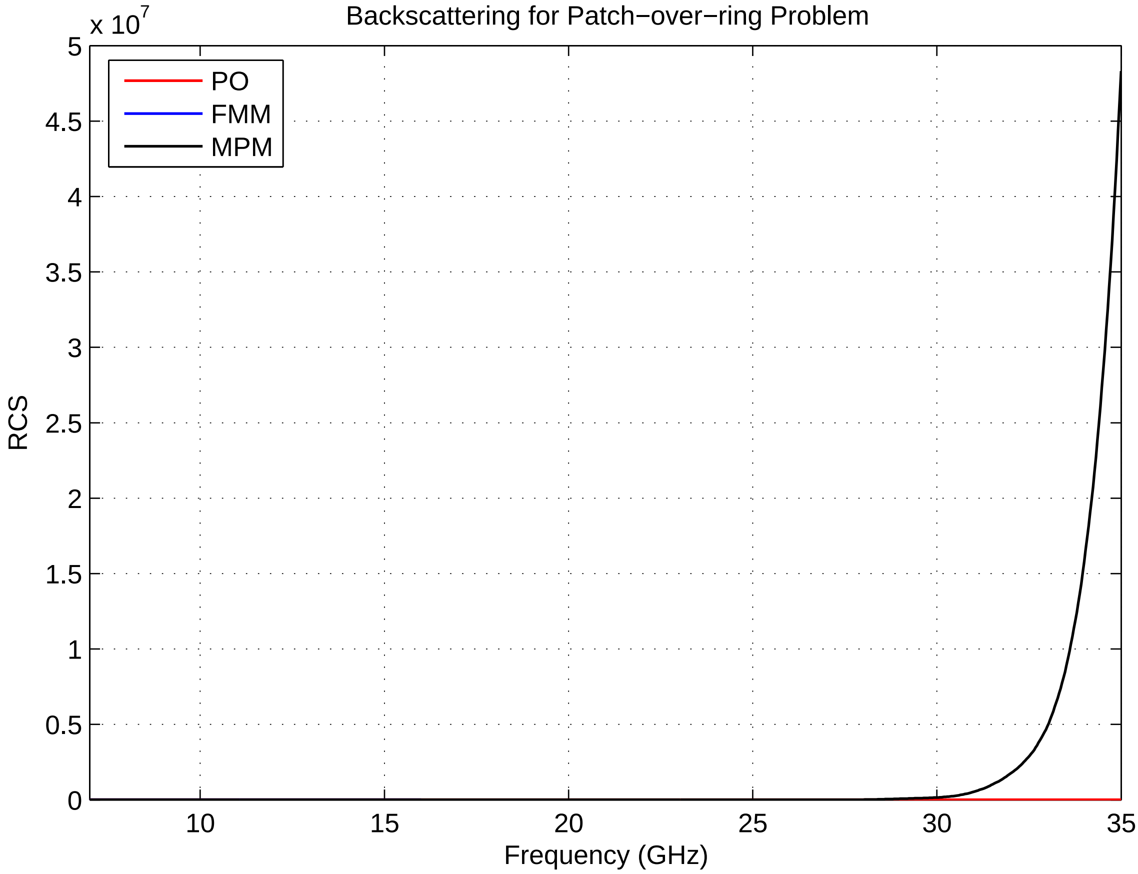


Figure 4.23: Blow up in the FMPM in the FMM solution signal.

modification done on the residual calculation stage in FMPM, (2.35).

Similarly, as in WCOMPM, W_F represents the FMM weight and W_B represents the PO weight. The matrix and the left-hand-side of the equation is divided into two parts as in ordinary COMPM, but the difference is only that the exponentials from the FMM data is used in the model as opposed to using both exponential sets from FMM and PO data in COMPM. The neutral weighting parameter is $(W_F, W_B) = (1, 0)$, whose result would be equal to FMPM. We should at least set $W_B = 1$ to introduce the influence of PO data in to the model. Increasing weight of FMM relative to weight of PO will give more emphasis to FMM, which should be the case. Note that a very high weight ratio such as (100000,1) will also converge to FMPM.

Figures 4.25 and 4.26 show the "before" and "after" plots of weighting process

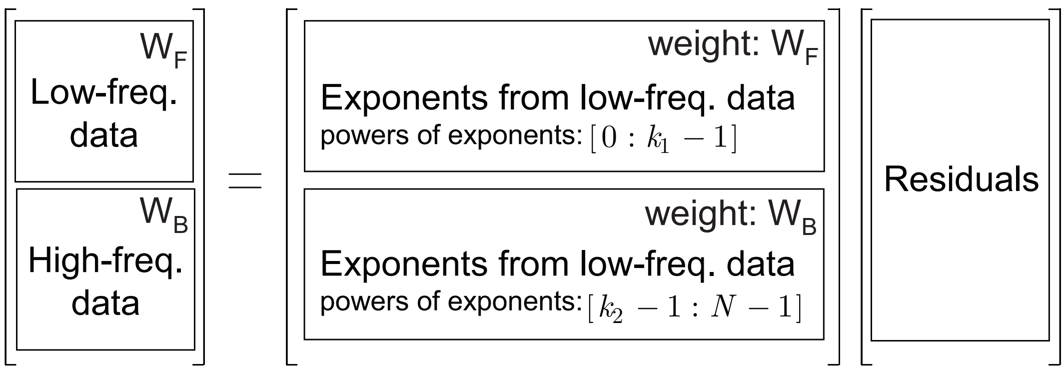


Figure 4.24: Residual calculation stage for WFMPM.

of FMPM. The signal that is shown in Figure 4.25 is actually the zoomed version of Figure 4.23. The weight parameter (100,1) is used in Figure 4.26 which clearly eliminates the blow up in the previous figure. The HF region of the extrapolated signal still does not agree with the PO data; however, this result will be coupled with the BMPM of the PO data using CDMPM extrapolation which will produce a better extrapolated signal. These results will be shown in the next chapter along with the other computational data results.

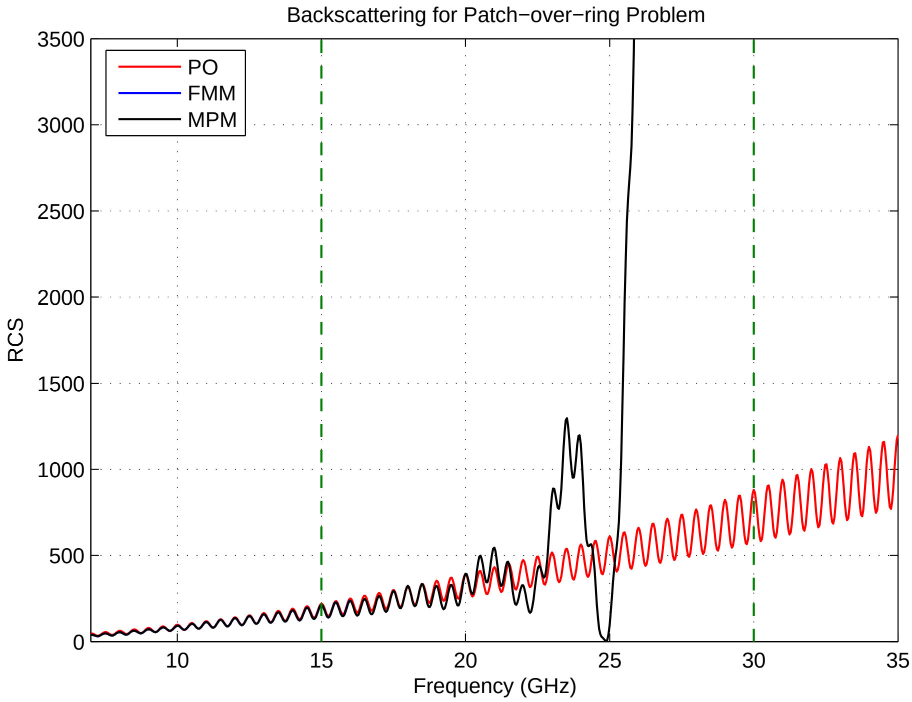


Figure 4.25: WFMPM with weighting parameter (1,0).

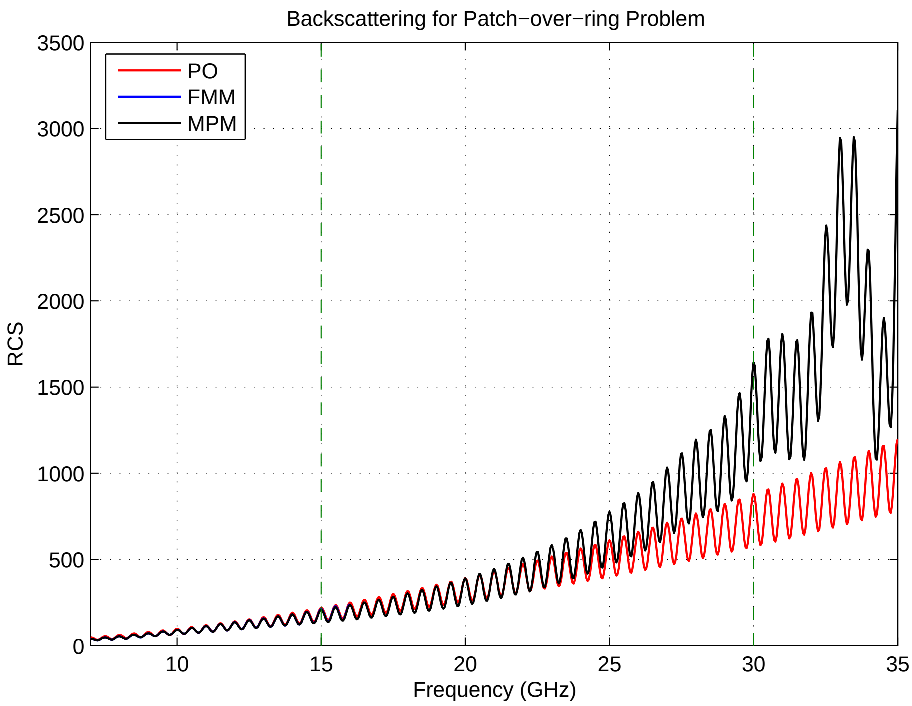


Figure 4.26: WFMPM with weighting parameter (100,1).

Chapter 5

Results with Computational Data

In this chapter we present extrapolation results achieved by using computationally generated scattering data. We present results on three distinct geometries: square patch, patch-over-ring and rectangular prism. In order to obtain the LF solution we used MLFMA, and PO for the HF solutions. In square patch and patch-over-ring geometries, we also introduced the edge current contributions to the PO solution. Similarly, in rectangular prism, in addition to the effects of edge currents, we also used the wedge contributions to further correct the approximate PO results. Throughout this chapter we will simply refer to LF and HF solutions as FMM and PO regardless of the method and correction used while performing the solution. LF and HF intervals that are used for modeling are selected by comparing the noise level and the general behavior of the signal in the solution region for each geometry individually.

5.1 Square Patch

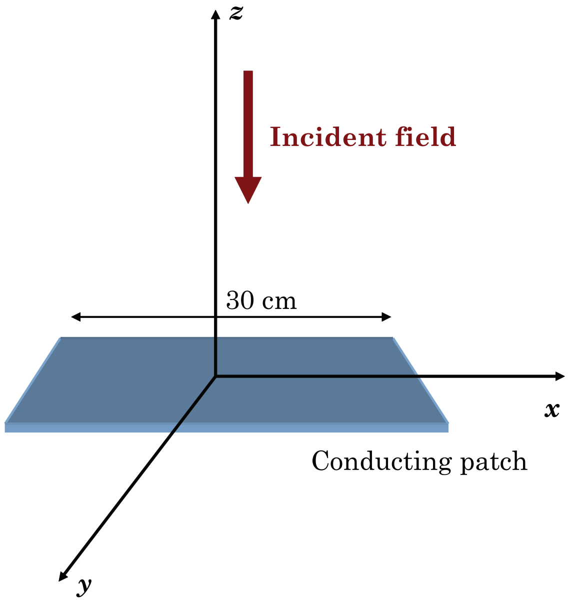


Figure 5.1: Square patch problem.

Our first result set is the extrapolation of backscattering solution of a square patch geometry. Figure 5.1 illustrates the scattering problem geometry; a thin square plate is positioned on the x–y plane and illuminated by a plane wave from -z direction. The origin is selected to be the center of the patch. This problem is selected due to its high backscattering characteristics. After facing high numerical errors in the conducting sphere due to weak backscattering, it was an intentional choice to move on the square patch problem.

Figure 5.2 illustrates the backscattering solution obtained from FMM and PO. We obtained FMM solution between 1–22 GHz and PO solution between 1–35 GHz. In this problem, PO solution data contains edge current contributions as well. We have selected 1–8 GHz interval as the LF available data interval and 30–35 GHz as the HF available data interval. The rest of the computational data will be used to show the accuracy of the extrapolated signal. Of course, PO data in 1–30 GHz interval can not be used as a reference data to observe the accuracy of the extrapolated signal, but still using it as a reference gives us some opinion on the general behavior of the extrapolation.

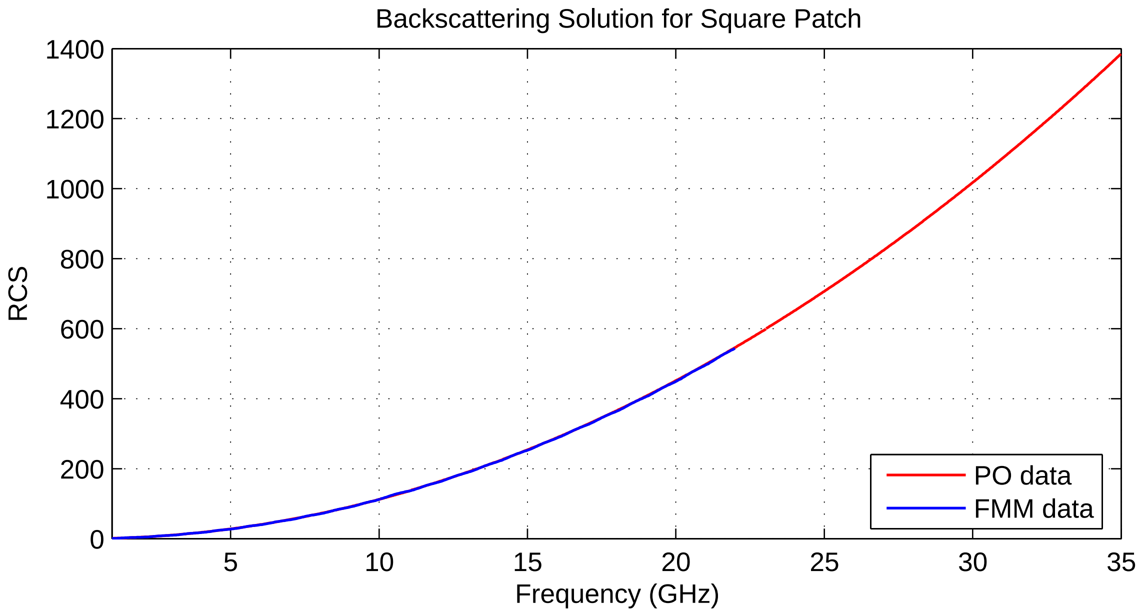


Figure 5.2: FMM and PO solution data for backscattering square patch problem.

As the next step, we try to estimate the noise on both FMM and PO data, and eventually select a value for M. Figure 5.3 illustrates the process of adding synthetically generated noise on the FMM data in the interval 1–8 GHz. It turns out that the computational noise on FMM data is greater then 10^{-3} but definitely smaller than 10^{-2}. We limit the M value for FMM data at nine. On the other hand, M limit for PO is determined more easily; singular values' magnitude drop significantly after two as shown in Figure 5.4. Therefore, for the PO data we not only limit M at two, but we also select it as two.

In Figure 5.5, we present the results of FMPM. The brute-force M optimization algorithm selects $M = 9$ as the best model to perform extrapolation. This is expected since in Figure 5.3 $M = 9$ has the smallest interpolation error. In this problem we select not to use weighting during FMPM, because from Figure 5.2, it can be seen that FMM and PO data agrees. In fact their agreement is so good that performing extrapolation for this problem is unnecessary. However, such an agreement can only be realized with advanced algorithms such as MLFMA and high computational resources like parallel PC clusters. Therefore, extrapolation is cheaper than actually solving the problem. In Figure 5.5 FMPM extrapolated signal seems to grow in error with increasing frequency but it will be shown that coupling will overcome this error.

In the BMPM we use two exponentials to create the model. The results are shown in Figure 5.6. As the computational noise on PO data is small, the BMPM is performed very accurately.

Next, we performed COMPM, whose results are shown in Figure 5.7. Brute-force optimization for COMPM selects the modeling parameters as $(M_L, M_H) = (9, 2)$. Coupling clearly removes the noise present in the FMPM. The maximum error appears around the level of noise on the FMM data. This is expected since we limit the M value in order not to model the noise. Once again, we select not to use weighting in coupling, i.e., FMM and PO influences are equal during coupling.

The other coupling approach, i.e., CDMPM, produces slightly higher error levels when compared with COMPM. These results are shown in Figure 5.8. Although the error levels on the extrapolation region is slightly higher, the overall result is still acceptable. Moreover, one should remember that, because no optimization is performed in CDMPM, the execution time is much better than COMPM.

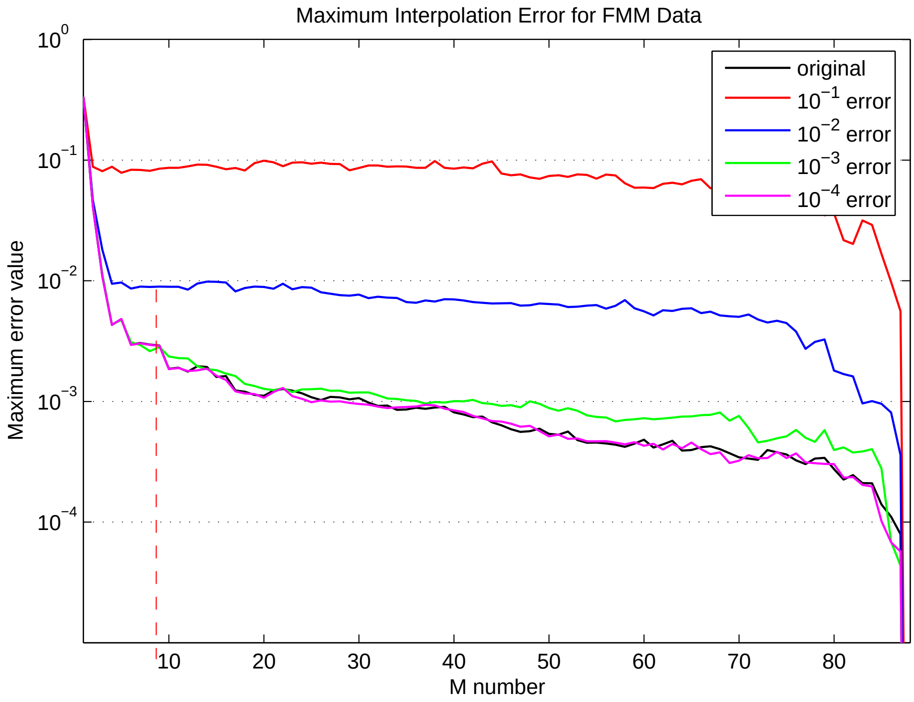


Figure 5.3: Estimation of noise on FMM and limiting M value for the square patch geometry. M limit is indicated by the vertical dashed line.

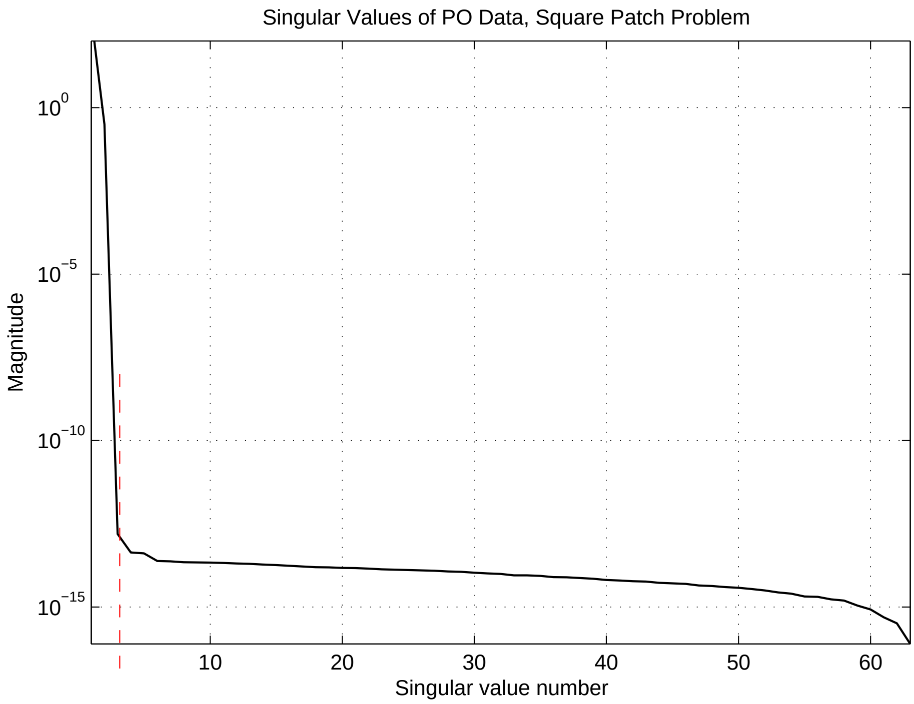
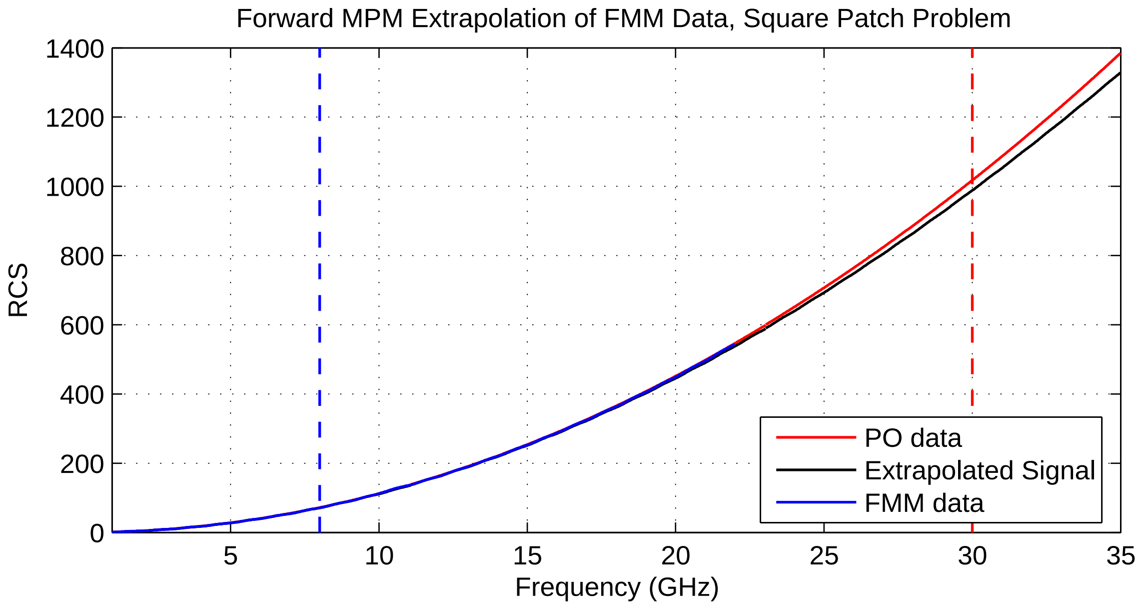
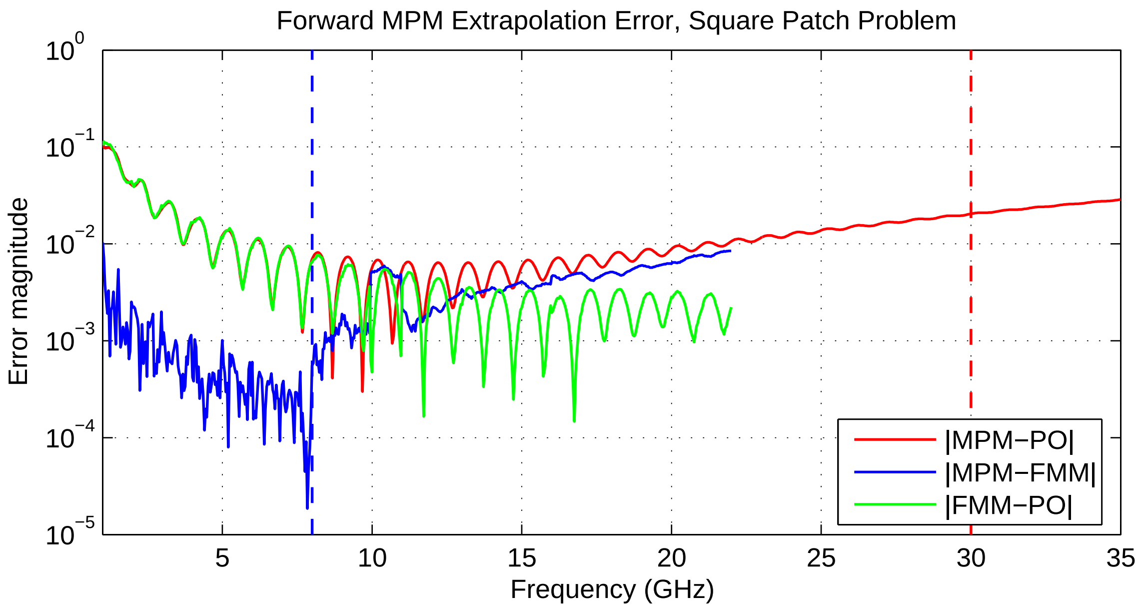


Figure 5.4: Singular values of PO data and limit for M value.

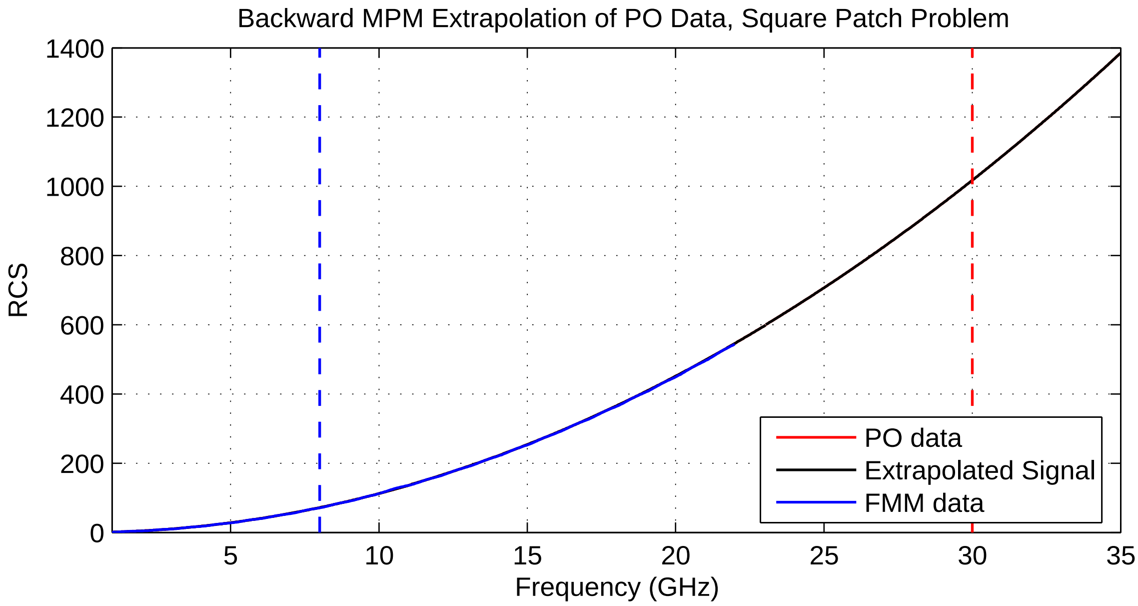


(a)

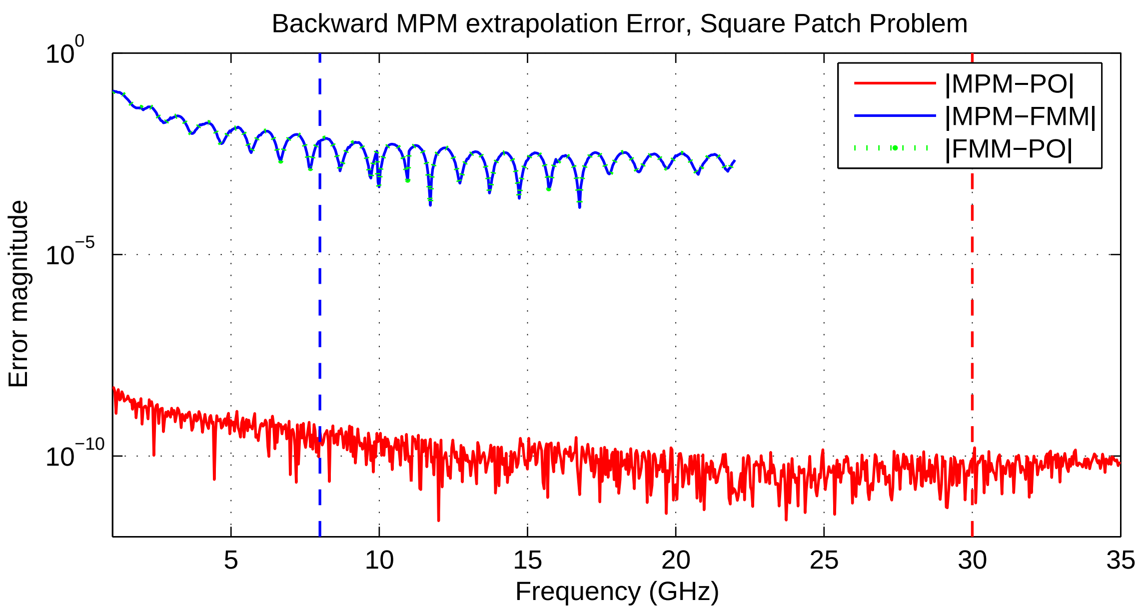


(b)

Figure 5.5: FMPM extrapolation of 1–8 GHz FMM data, $M = 9$, (a) signals, (b) errors.

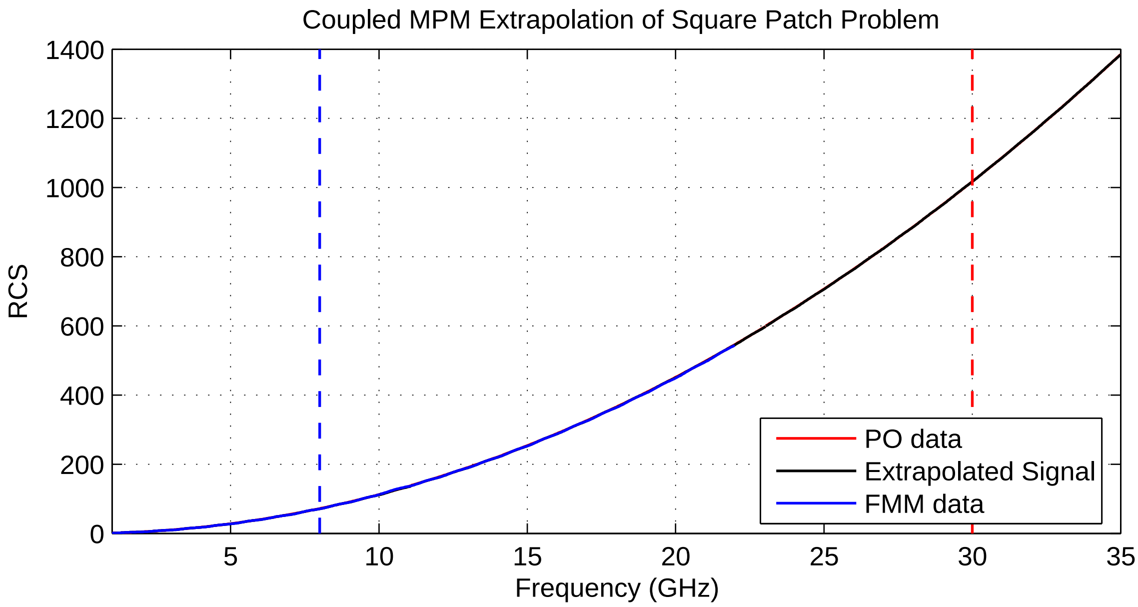


(a)

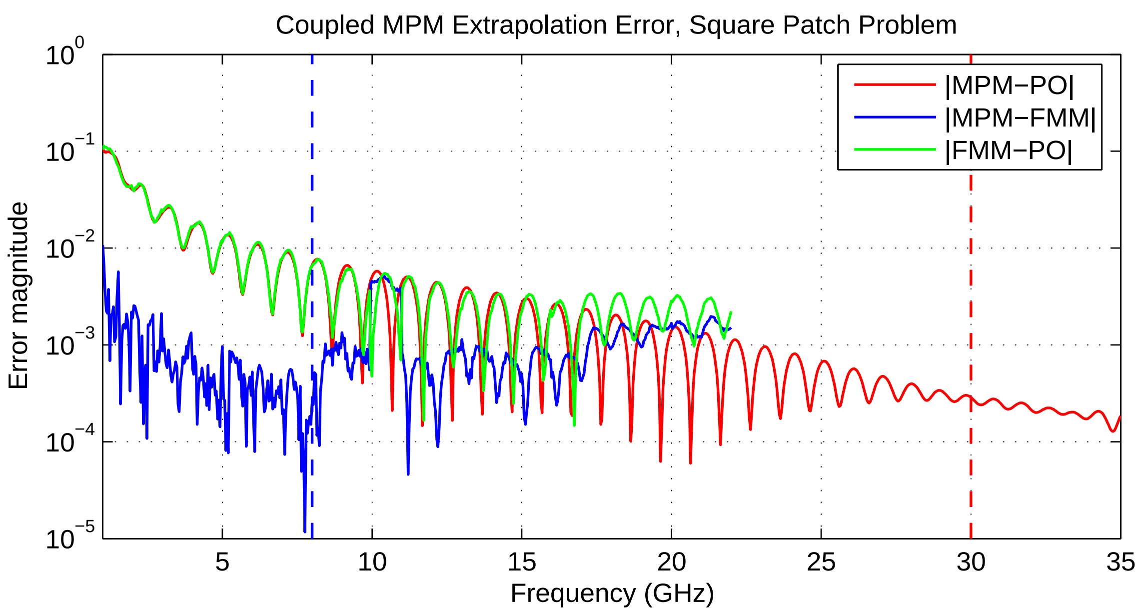


(b)

Figure 5.6: BMPM extrapolation of 30–35 GHz PO data, $M = 2$, (a) signals, (b) errors.

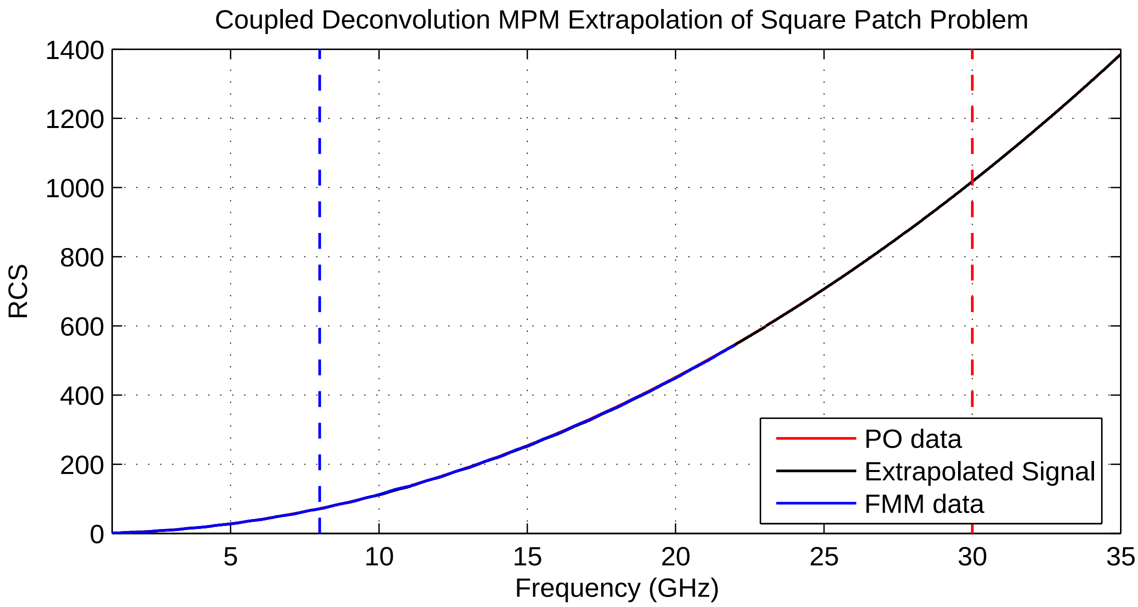


(a)

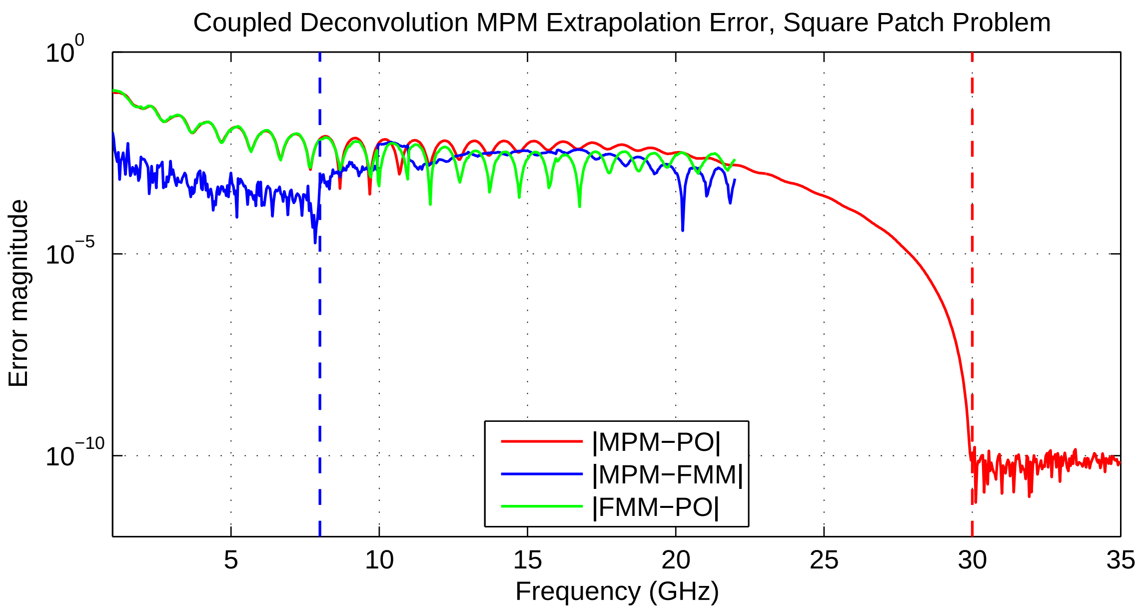


(b)

Figure 5.7: COMPM extrapolation of 1–8 GHz FMM, 30–35 GHz PO data, $(M_L, M_H) = (9, 2)$, (a) signals, (b) errors.



(a)



(b)

Figure 5.8: CDMPM extrapolation of 1–8 GHz FMM, 30–35 GHz PO data, $(M_L, M_H) = (9, 2)$, (a) signals, (b) errors.

5.2 Patch Over Ring (POR)

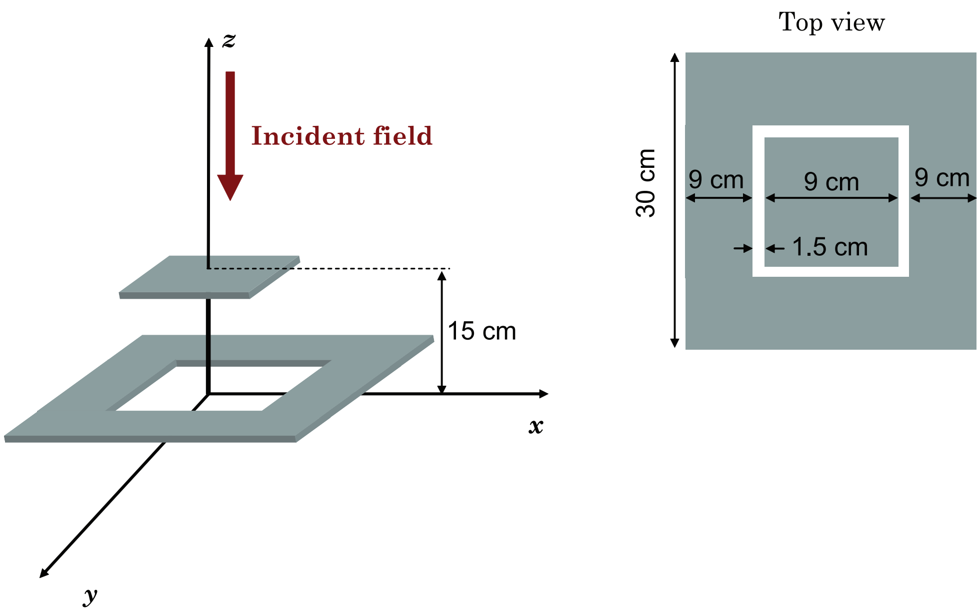


Figure 5.9: Patch-over-ring problem.

In the square patch problem, the backscattering signal is too easy to model. Besides, computational solutions agree, which may not be always the case. Therefore, in order to perform extrapolation on more generic situations we alter the solution geometry. We introduce a new, but smaller square patch on top of the original one and cut a hole in the middle of the bigger patch. This conducting body is illuminated by a plane wave from the top, in the -z direction. Problem geometry is illustrated in Figure 5.9. Our aim is to have a strong backscattering as in the previous example, but in addition, we would like to have some more oscillations to make the modelling relatively harder.

Figure 5.10 shows the backscattering solution of patch-over-ring geometry for FMM and PO data. It can be seen that backscattering is strong and has oscillations. Moreover, FMM and PO data do not agree much for LFs. For this geometry we obtain FMM data in the interval 1–16 GHz and PO data in the interval 1–35 GHz. Only 7–15 GHz and 30–35 GHz of this FMM and PO data will be used for modeling respectively.

In Figure 5.11, we show the results of the noise estimation on FMM data.

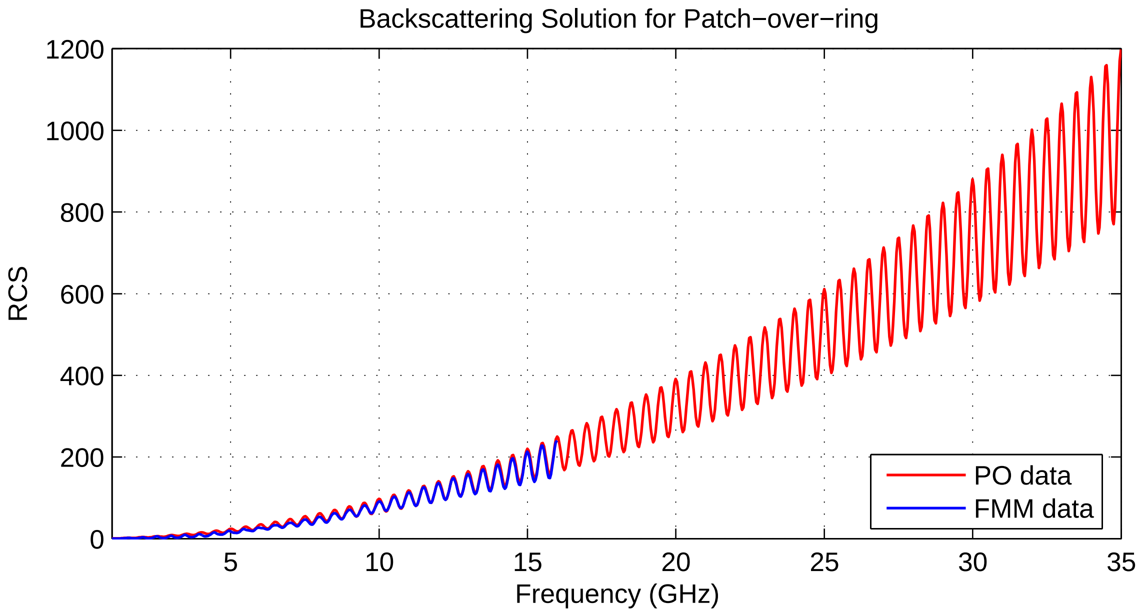


Figure 5.10: FMM and PO solution data for backscattering patch-over-ring problem.

Noise is estimated as between 10^{-2} and 10^{-3} and M limit for FMM data is selected as 12. For the PO data, we once again look at the magnitude of singular values. We observe a sharp fall in the magnitudes at $M = 4$ in Figure 5.12. Therefore, with a similar reasoning as in the previous section, we select $M = 4$ for PO data.

Figure 5.13 shows the results of WFMPM. Brute-force M optimization selects 12 as the best M value, and we choose $W_L = 1000$, the weight of FMM. The extrapolated signal seems very erroneous, but recall from Section 4.7 that if we have not used the weighting, the results would be worse; in fact it would blow up. On the other hand, BMPM works very accurately as in the square patch problem. Figure 5.14 shows the results of the BMPM. It is easy to conclude that the complex exponentials with magnitudes greater than unity do not occur in PO data models; however they do occur in FMM data models and we have to use weighting to combat them.

In Figure 5.15, we illustrate the WCOMPM results. The brute-force M optimization selects $(M_L, M_H) = (9, 2)$ as the best M pair and we choose W_R, the relative weight, to be four. It can be seen that the extrapolated signal agrees with the general behavior of the computational data. The red error in Figure 5.15(b) is slightly higher than the errors of previous extrapolation examples in the ER, but this can be misleading. Red error in Figure 5.15(b) is the relative error of the extrapolated signal with respect to the PO data. However, it is not logical to compare the extrapolated signal with the PO data in the ER, because we do not trust PO data in that region. If the PO data was correct and accurate in that region, we would not be doing extrapolation but using the PO data instead. On the other hand, there is nothing else to compare with in the ER, therefore, we have to use red error but approach carefully when discussing the performance of the extrapolation by looking at this error level. To sum up, extrapolation result seems good in the sense that the general behavior of the extrapolated signal agrees with that of the computational data.

A similar situation is observed for the results of CDMPM in Figure 5.16. Notice how coupling with CDMPM eliminates the error in FMPM. The results of COMPM is very much similar to COMPM results, so it is hard to say that one method is better than the other for this specific problem.

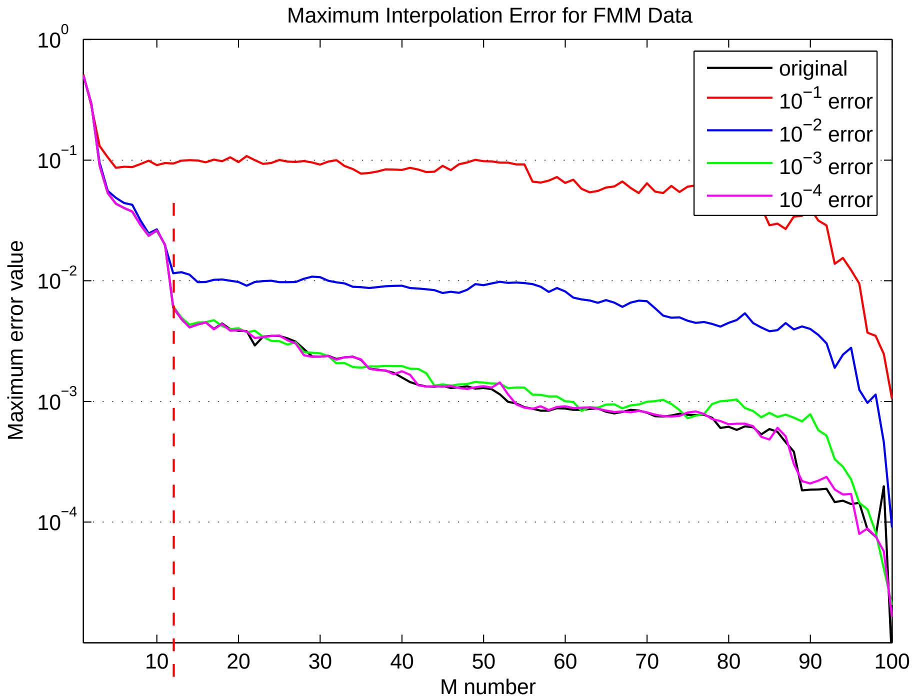


Figure 5.11: Estimation of noise on FMM and limiting M value for the patch-over-ring geometry. M limit is indicated by the vertical dashed line.

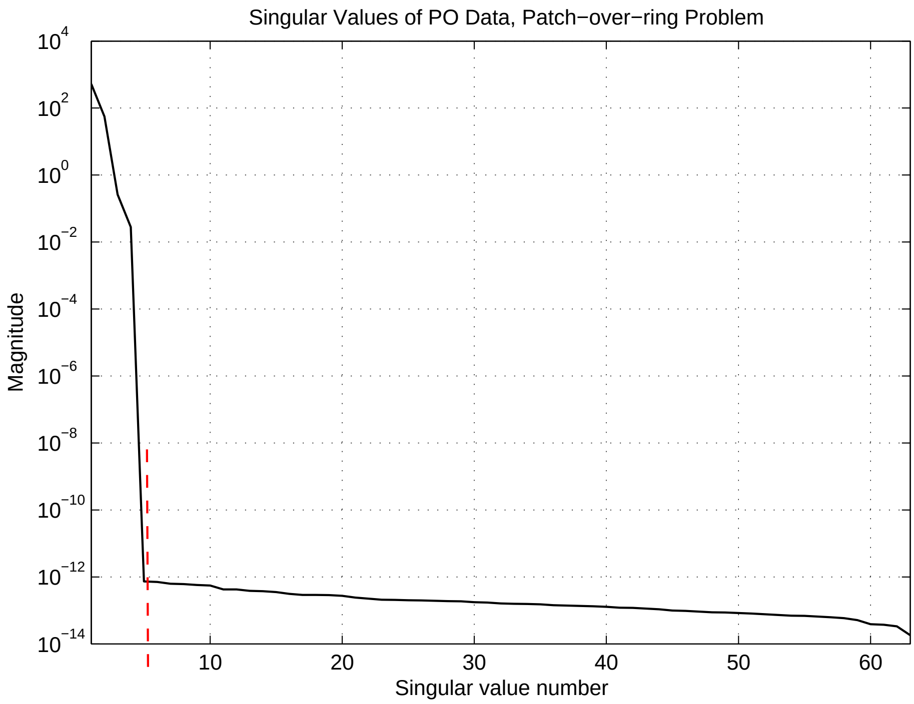
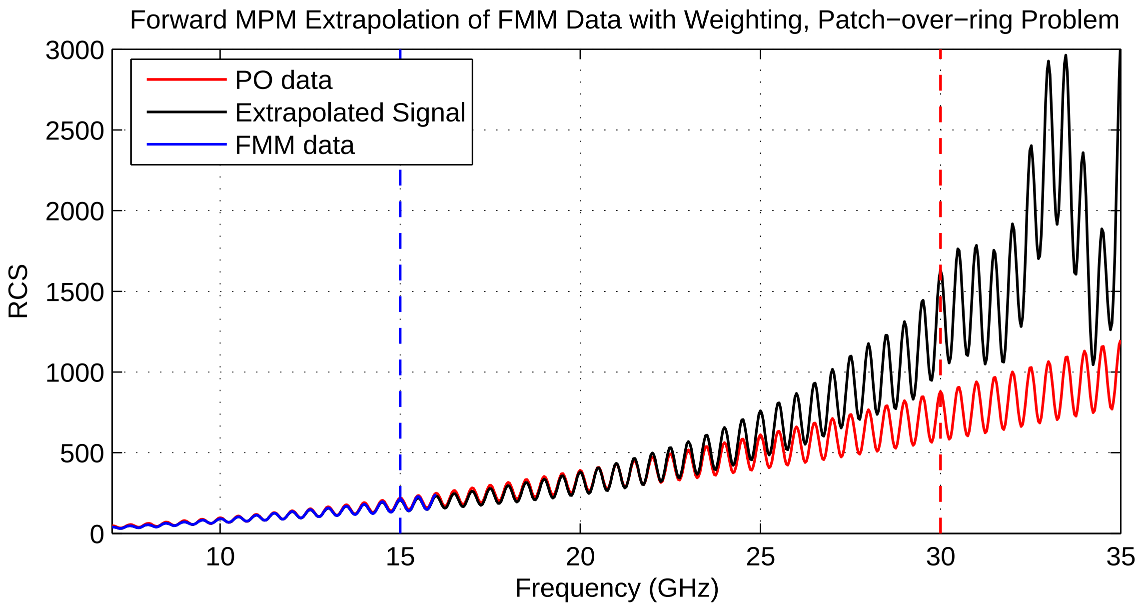
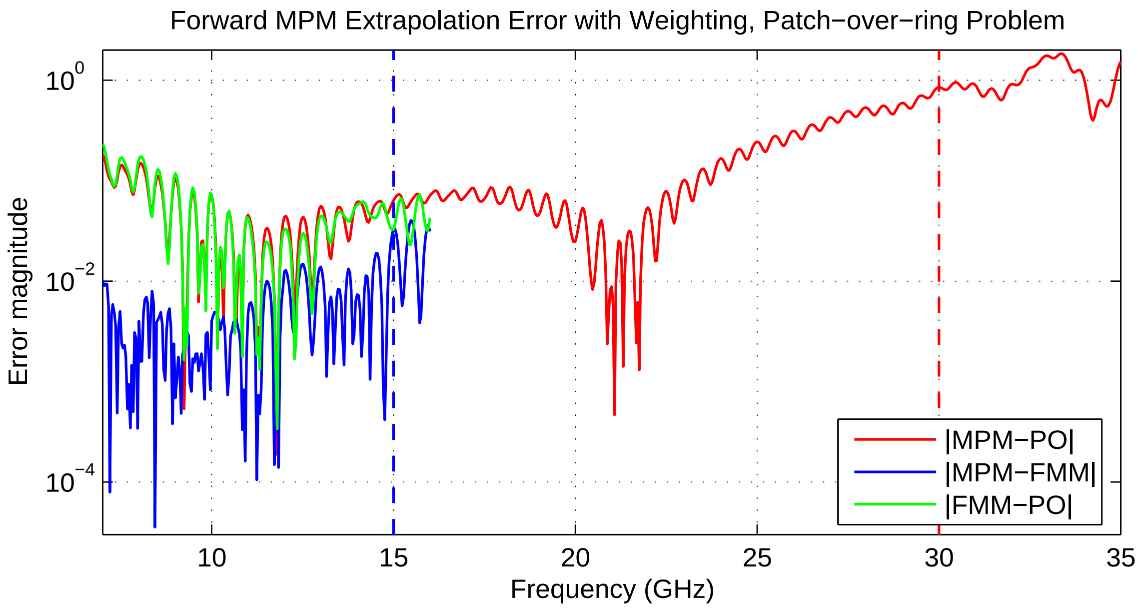


Figure 5.12: Singular values of PO data and limit for M value.

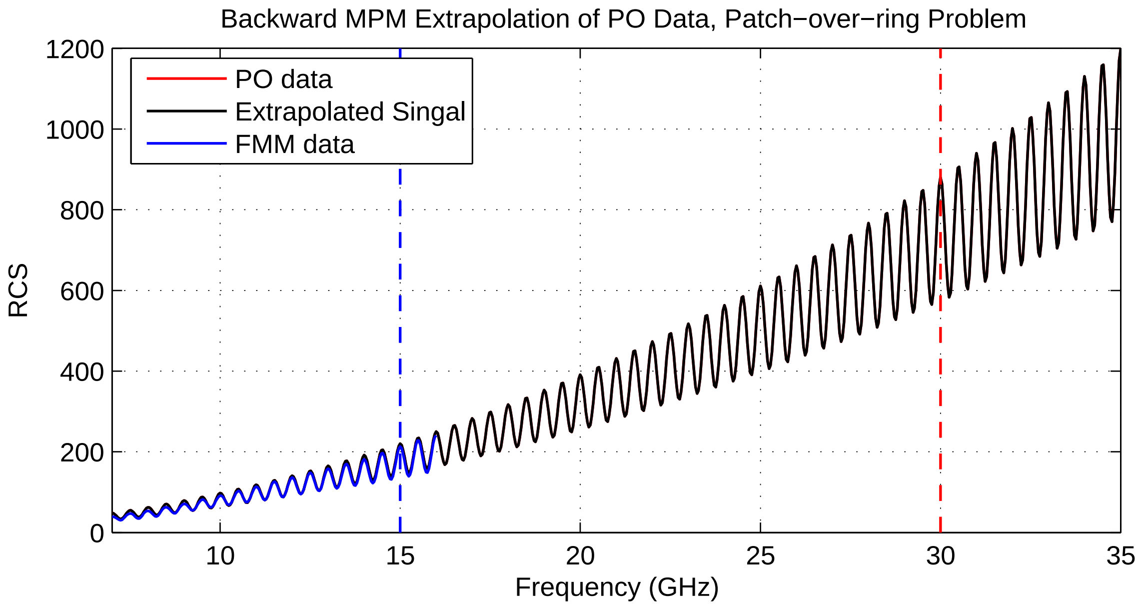


(a)

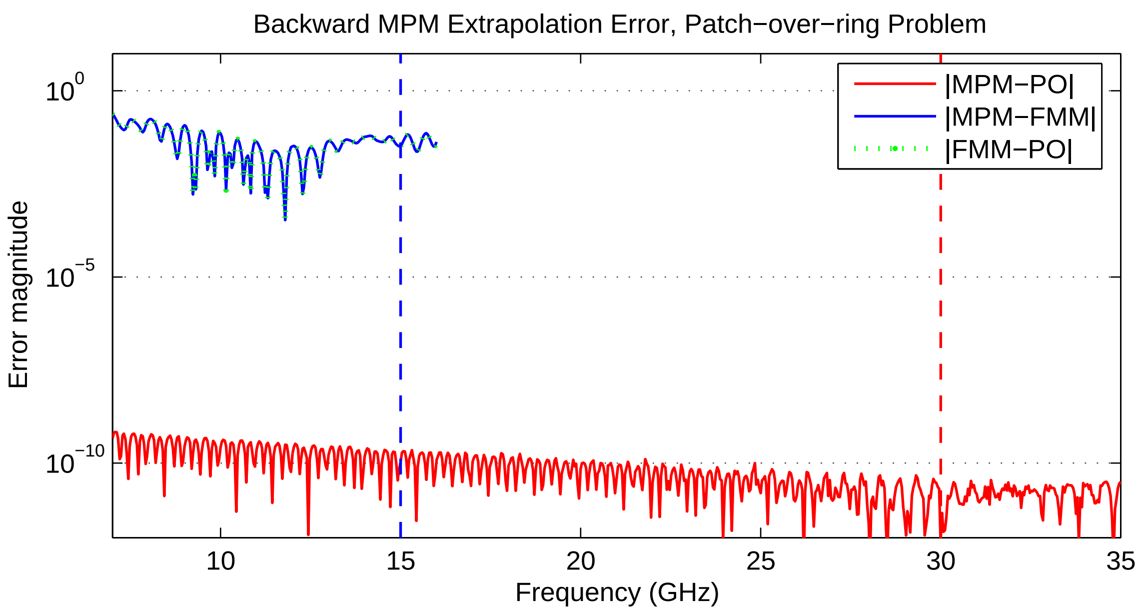


(b)

Figure 5.13: WFMPM extrapolation of 7–15 GHz FMM data, $M = 12$ and $W_L = 1000$, (a) signals, (b) errors.

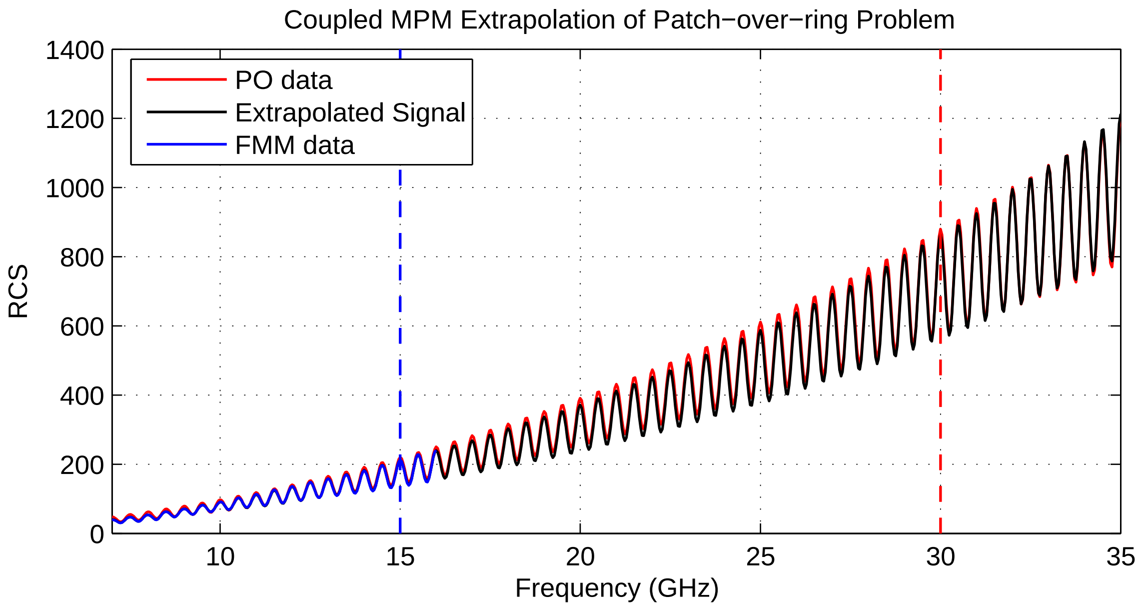


(a)

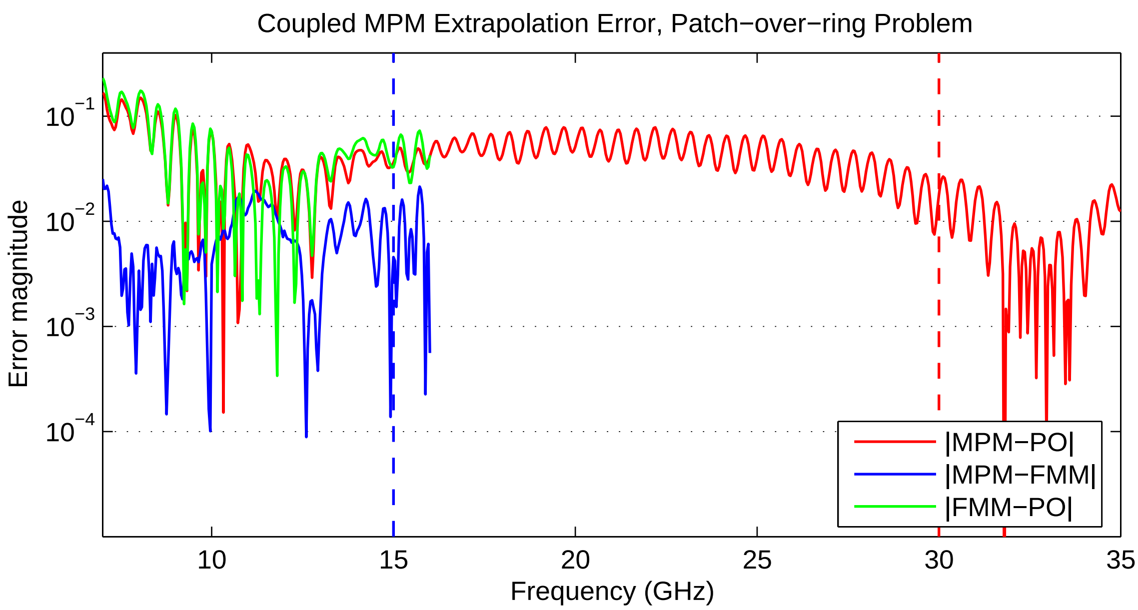


(b)

Figure 5.14: BMPM extrapolation of 30–35 GHz PO data, $M = 4$, (a) signals, (b) errors.

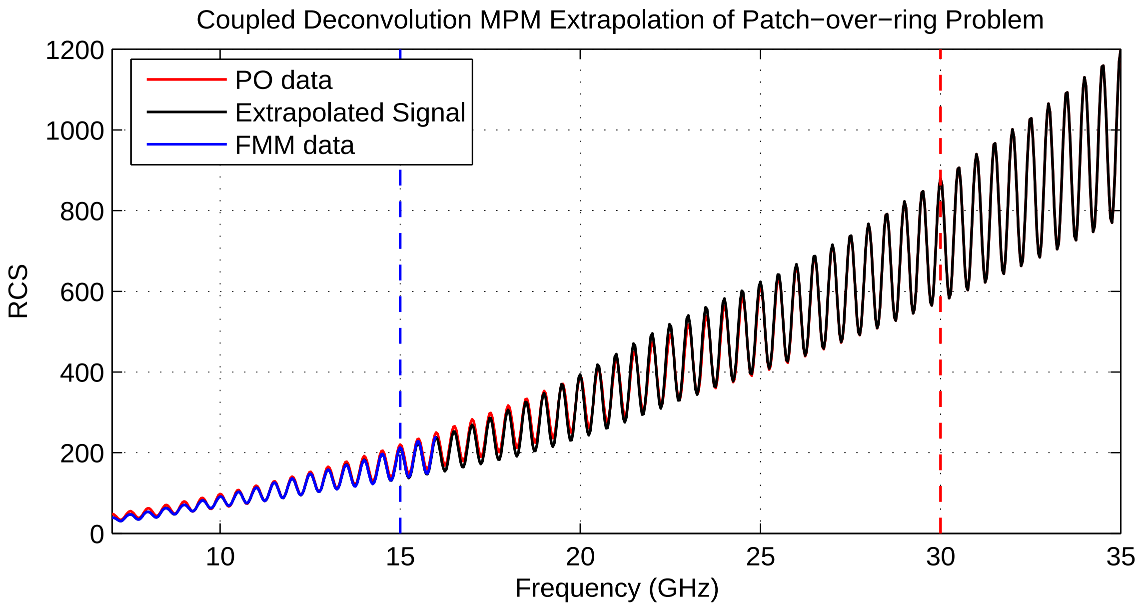


(a)

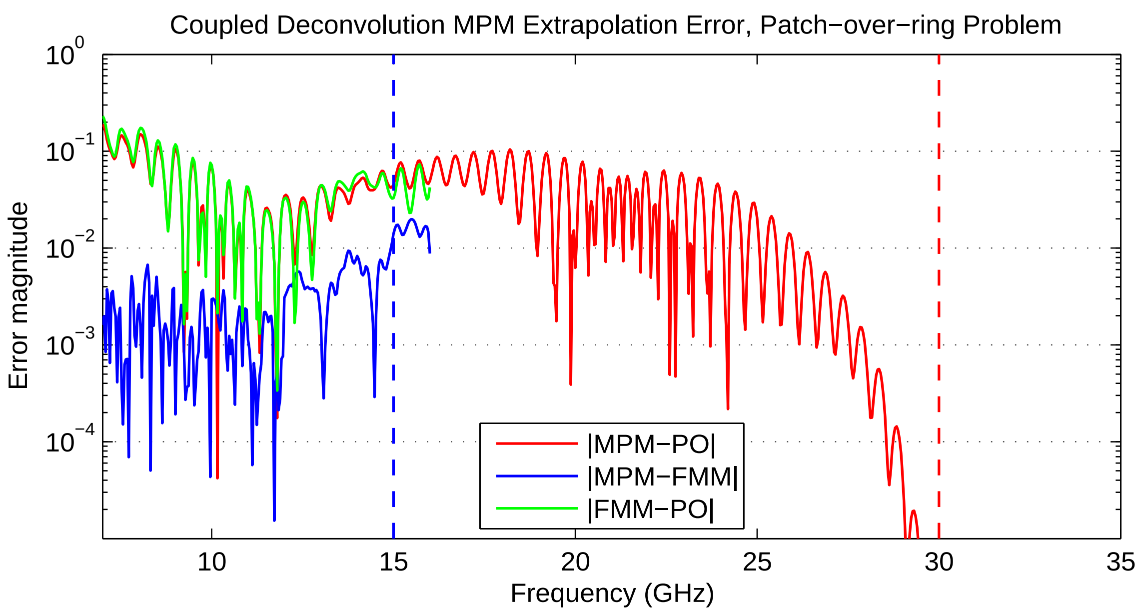


(b)

Figure 5.15: WCOMPM extrapolation of 7–15 GHz FMM, 30–35 GHz PO data, $(M_L, M_H) = (9, 2)$ and $W_R = 4$, (a) signals, (b) errors.



(a)



(b)

Figure 5.16: CDMPM extrapolation of 7–15 GHz FMM, 30–35 GHz PO data, $(M_L, M_H) = (9, 2)$, $W_L = 1000$, (a) signals, (b) errors.

5.3 Rectangular Prism

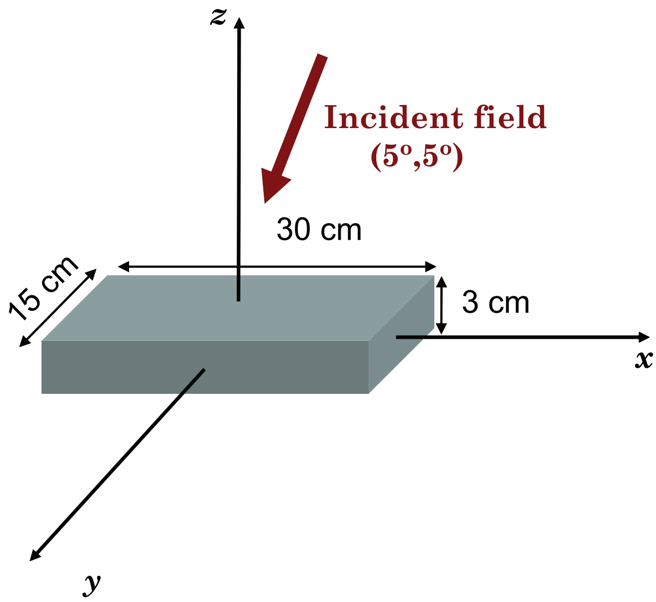


Figure 5.17: Rectangular prism problem.

Our last geometry for extrapolation examples is the rectangular prism. The geometry of the problem is illustrated in Figure 5.17. In order to take the wedge current contributions into account, we used an oblique plane wave incidence at $\phi = 5°$, $\theta = 5°$.

Figure 5.18 presents the FMM and PO solutions for the backscattering with respect to the incident angle. We have FMM solution in the 1–20 GHz interval and PO solution in the 1–40 GHz interval. An experienced eye at extrapolation could easily tell by looking at Figure 5.18 that extrapolation of these signals would be very hard. The very first problem that catches the eye is the mismatch of the FMM and PO data. However, this is not the main reason for the degradation of the performance of the extrapolation; we have proposed weighting for such situations. What really would affect the modelling and extrapolation is the slow oscillation of the data over the 1–40 GHz interval. For the FMM data, we only have two and a half peaks to estimate the total behavior of the complete data. On the other hand, very low frequency solutions of FMM data will be disregarded since they seem to contain LF oscillations; this leaves one and a half peak. In the

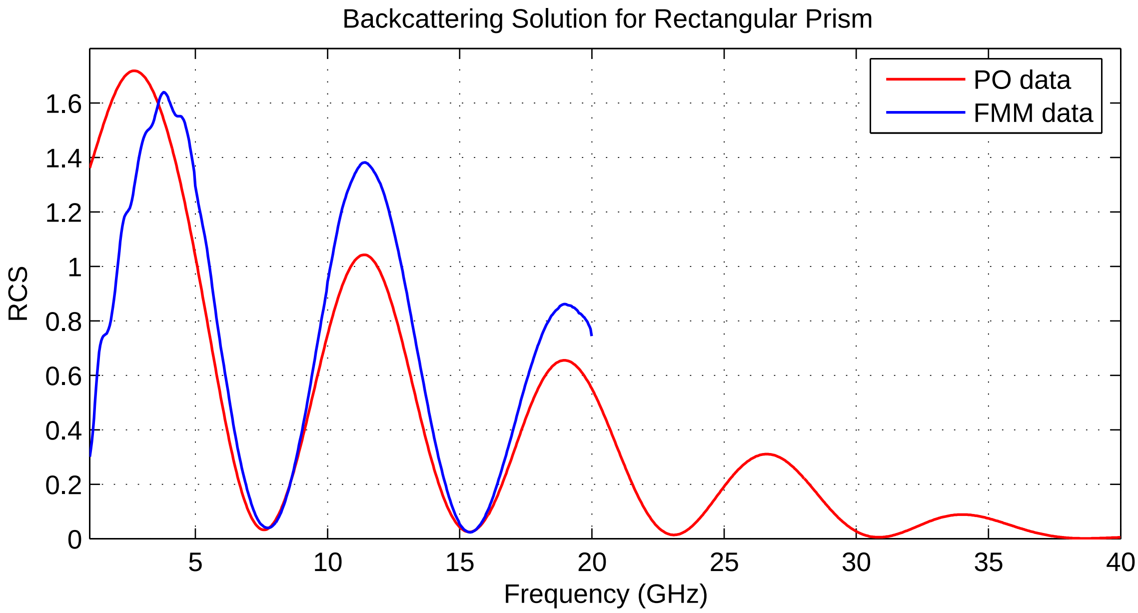


Figure 5.18: FMM and PO solution data for backscattering rectangular prism problem.

next figures, we will see that these presumptions are indeed true.

In order not to take the LF oscillations, we use the 4–16 GHz interval as LF region and 30–40 GHz interval as the HF region. Figure 5.19 shows the noise estimation procedure for the FMM data We estimated the noise as between 10^{-2} and 10^{-3}, and the M limit for FMM is selected as 14. The magnitude of singular values for PO data shows a sharp fall at $M = 8$ as shown in Figure 5.20, which is selected as the PO limit for M.

The results of WFMPM extrapolation in Figure 5.21 supports the fact that it is very hard to estimate the whole data by using almost one period of it. The weighting by 1000 only made the extrapolated signal not to blow up, but caused irregular behavior at HF.

A similar result can be observed for the BMPM of the PO data in Figure 5.22. 30–40 GHz data turns out to be enough to estimate the PO data in the extrapolation region, but not in the LF, where the maximum error occurs. Even so, BMPM of PO data is much better than the FMPM of FMM data.

What comes as a surprise is the results of WCOMPM in Figure 5.23. The extrapolated signal agrees with the FMM data in the ER and also agrees with the PO data in the HF region to some extent. For this figure, it might be misleading to judge the performance of the extrapolation by just looking at the error levels. Especially in the 30–40 GHz region, where data is very small. In this interval the magnitude of the data even goes to zero at two frequencies, which infact causes the error to produce its peak values. Therefore, by looking at the signal comparison in Figure 5.23(a), it could be said that, WCOMPM is successful for this geometry. The success of COMPM is truly proven by this example.

On the other hand things do not work out that well for CDMPM. Figure 5.24 clearly shows that the error of FMPM at HF vanishes by the use of coupling, but the error in the ER is still intact. This problem shows us that, proposing and using two separate methods for coupling is far from being redundant. There may also be situations where CDMPM performs much better than COMPM.

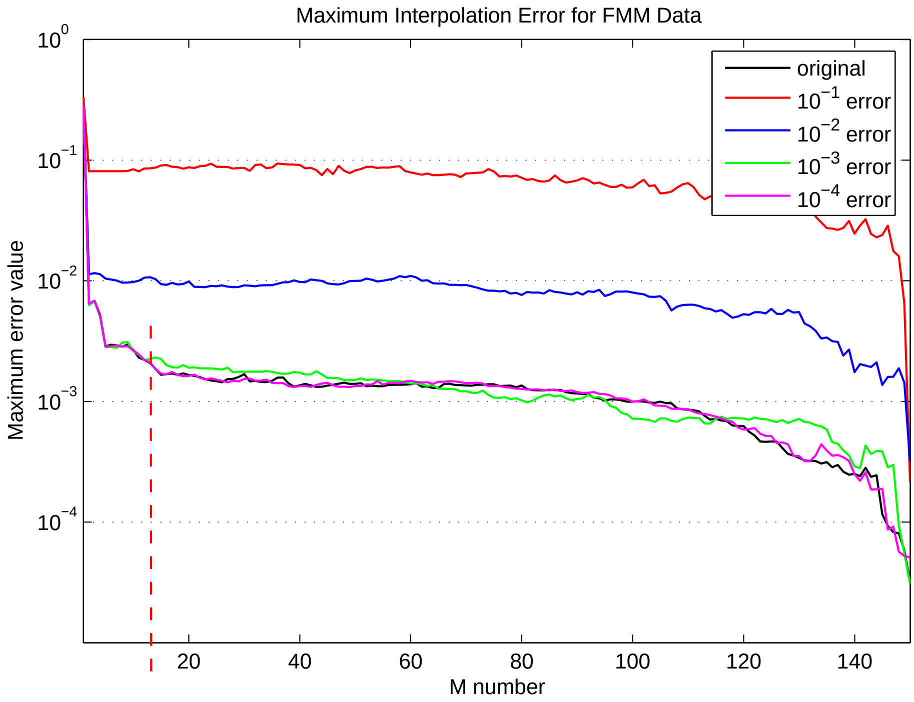


Figure 5.19: Estimation of noise on FMM and limiting M value for the rectangular prism geometry. M limit is indicated by the vertical dashed line.

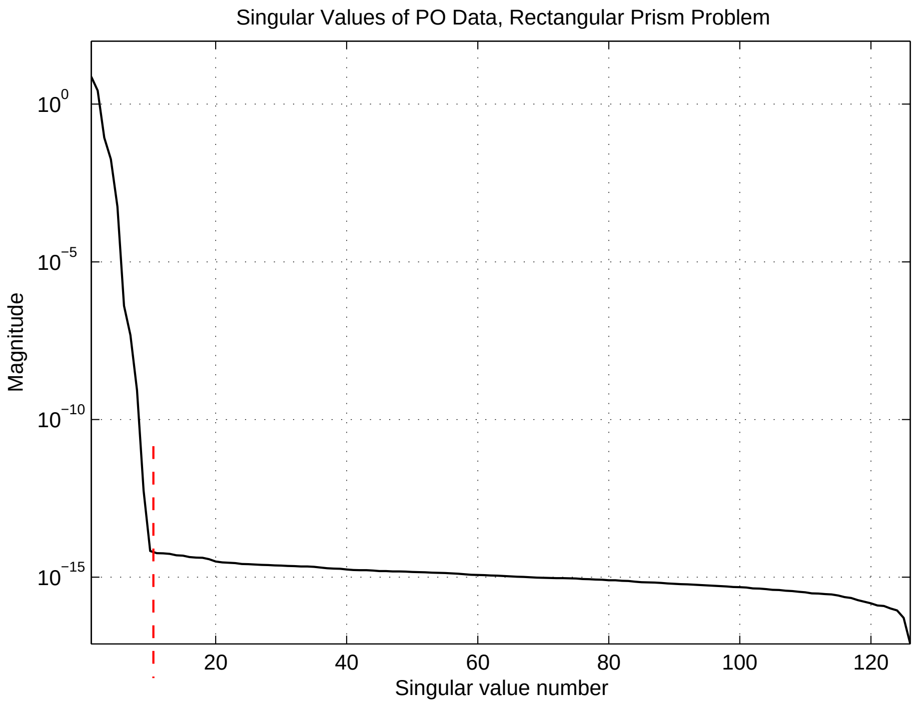
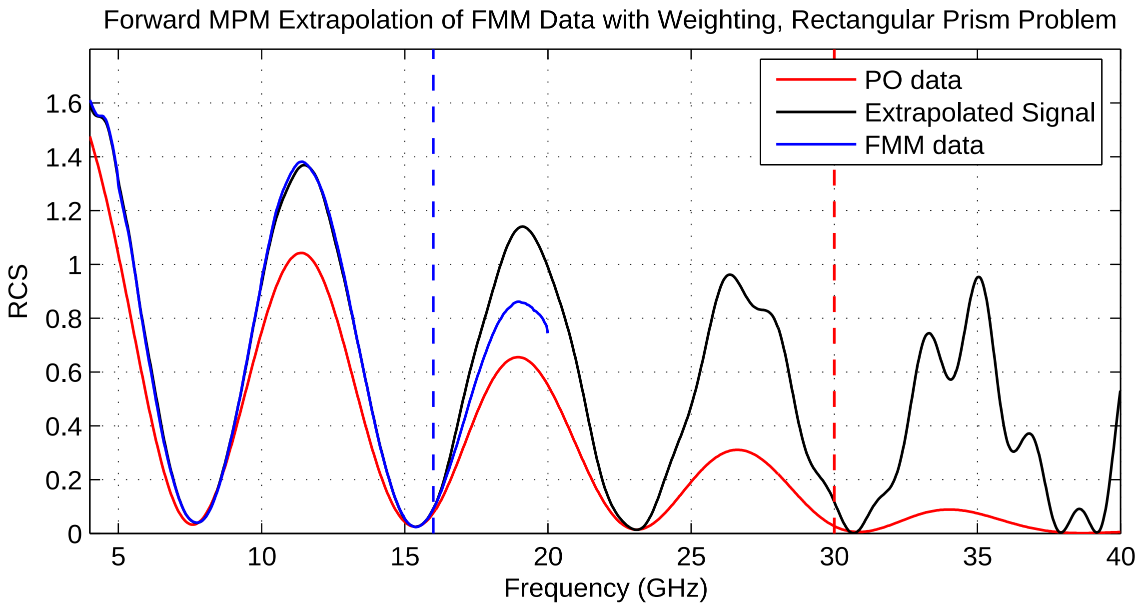
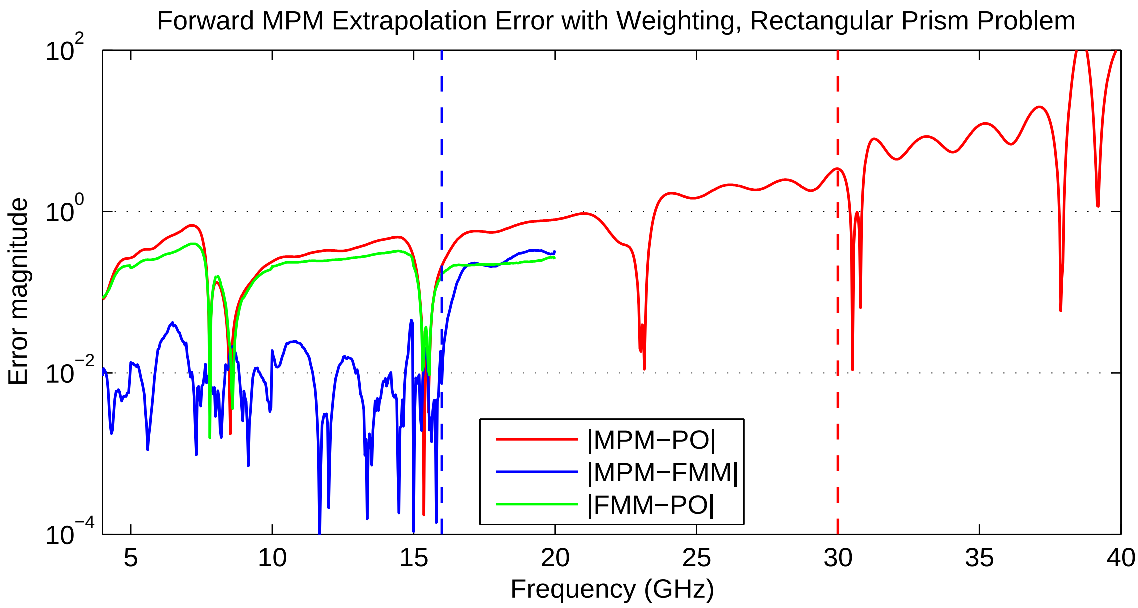


Figure 5.20: Singular values of PO data and limit for M value.

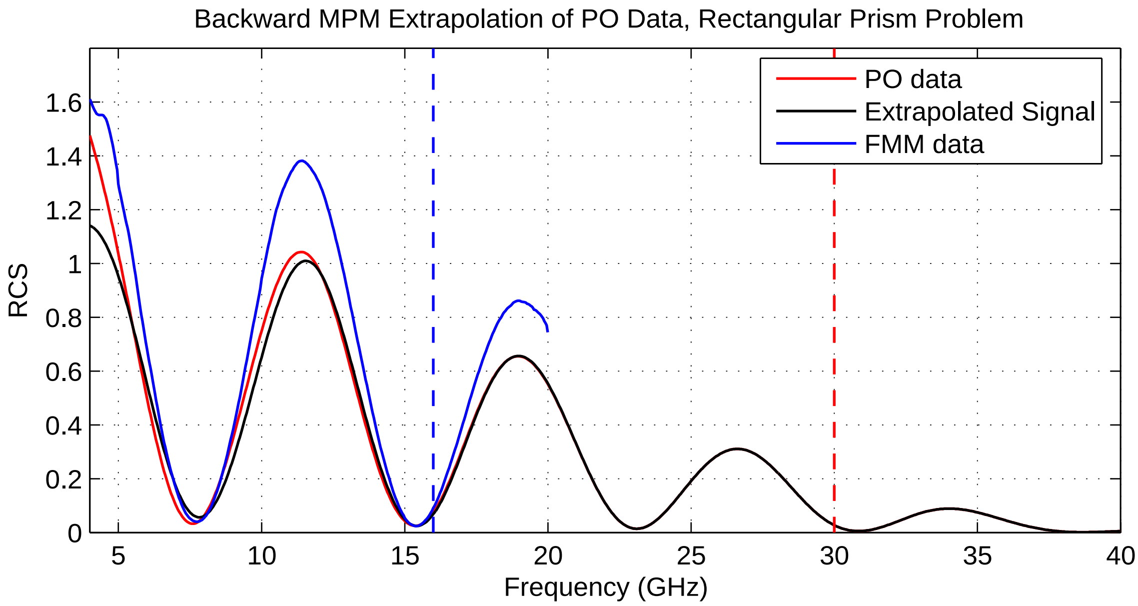


(a)

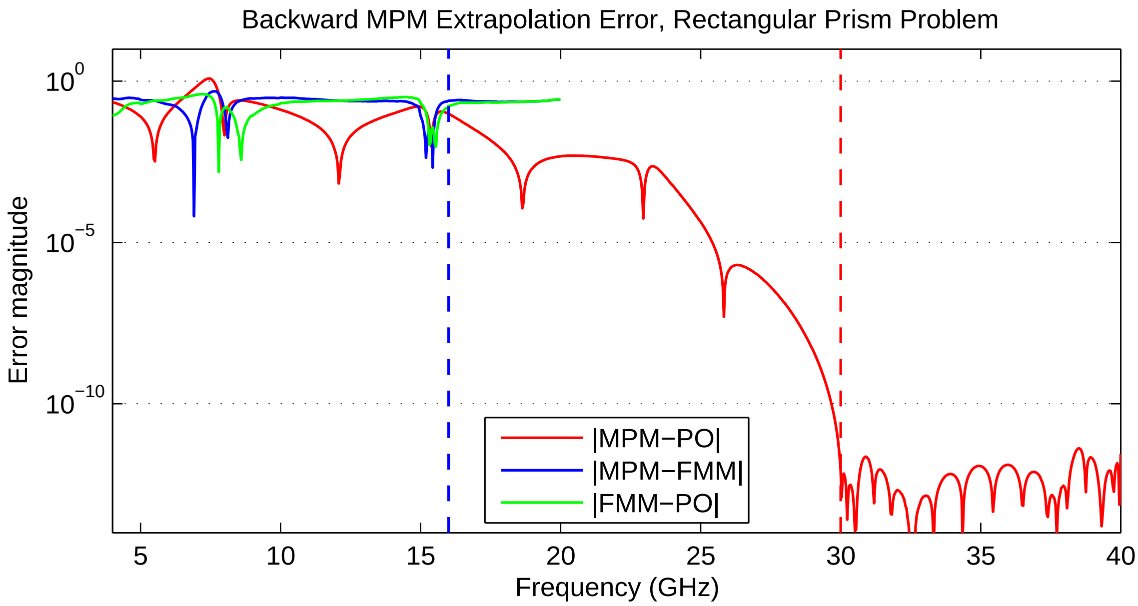


(b)

Figure 5.21: WFMPM extrapolation of 4–16 GHz FMM data, $M = 14$ and $W_L = 1000$, (a) signals, (b) errors.

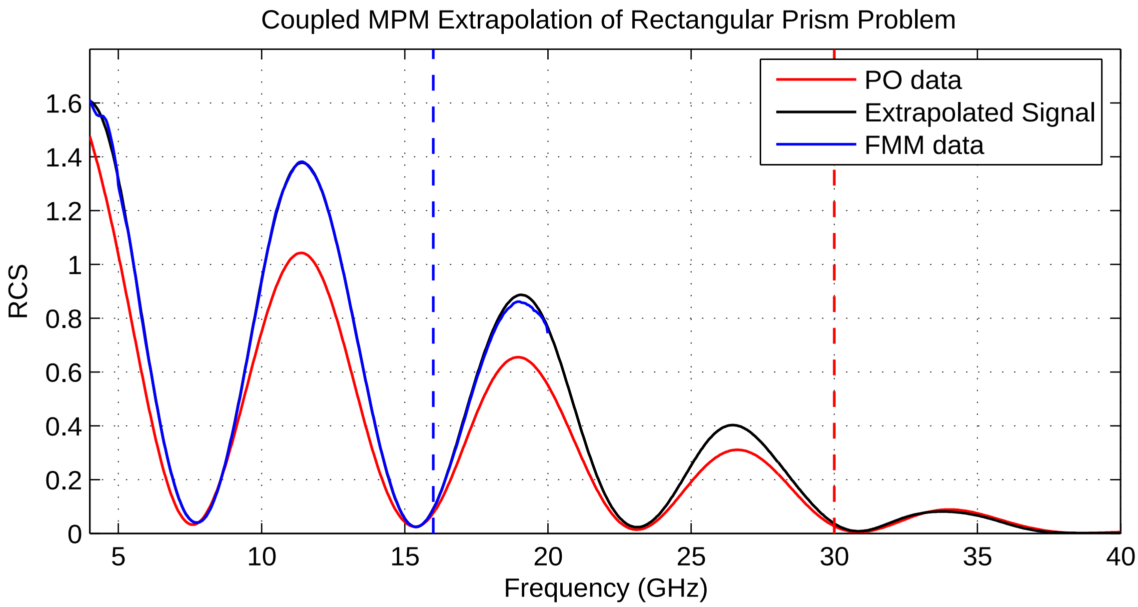


(a)

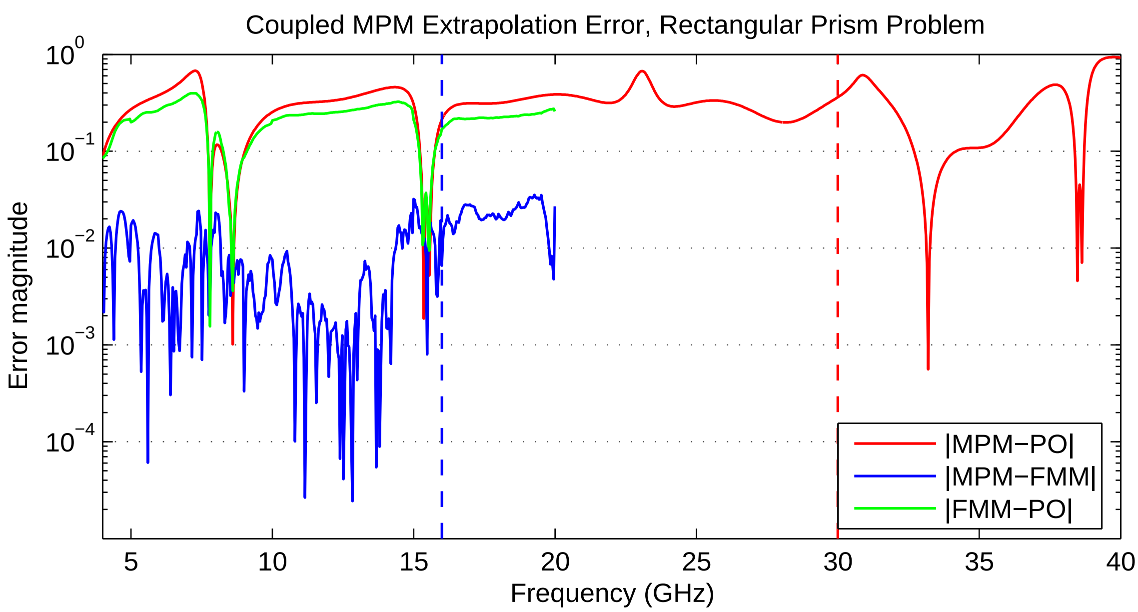


(b)

Figure 5.22: BMPM extrapolation of 30–40 GHz PO data, $M = 6$, (a) signals, (b) errors.

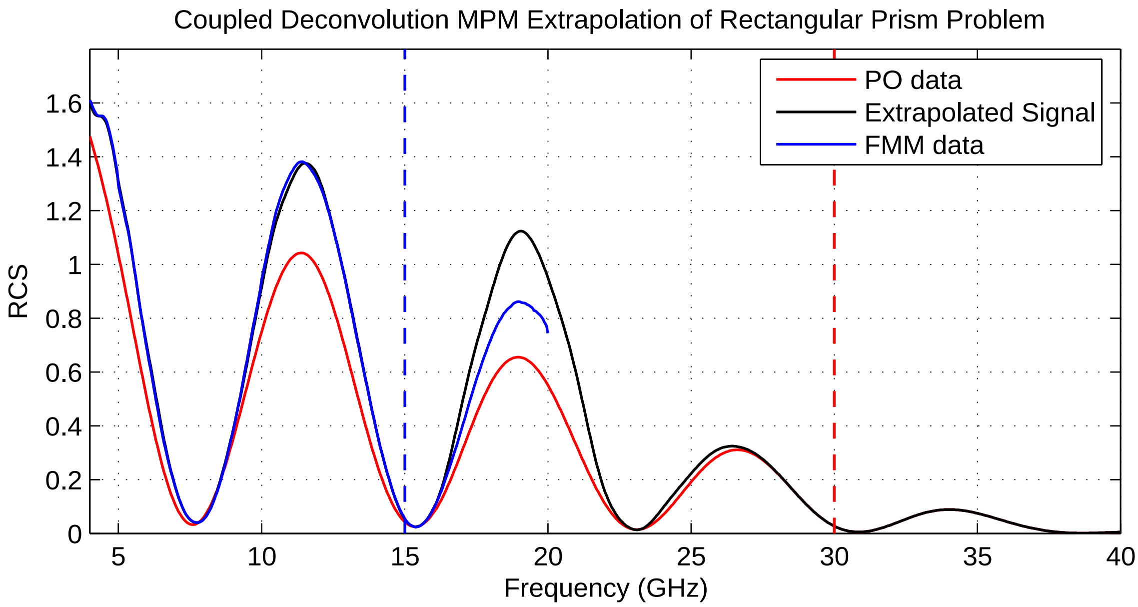


(a)

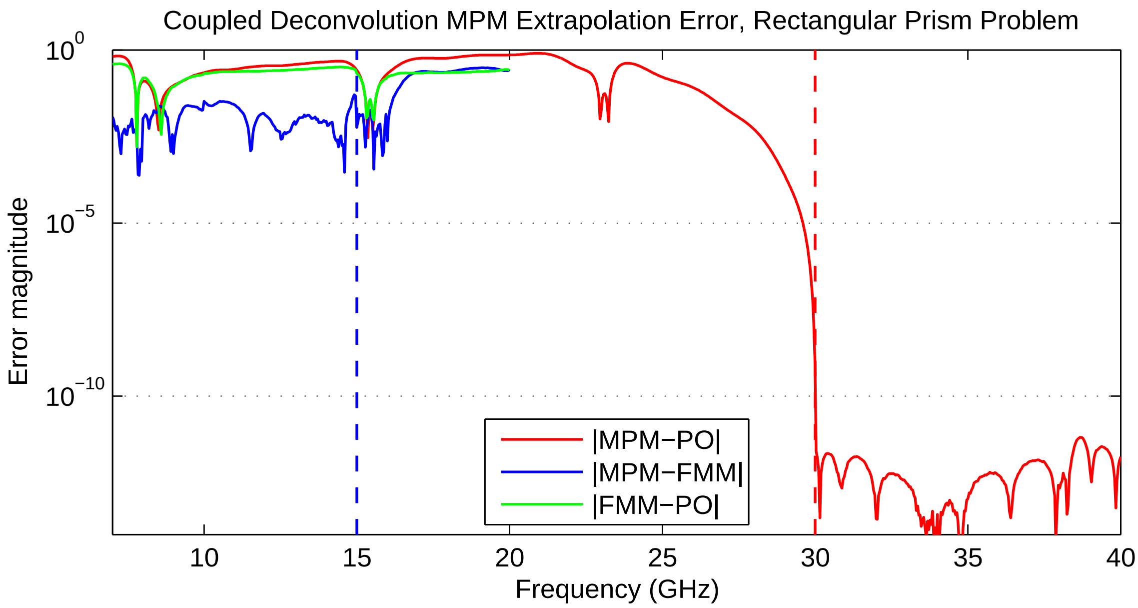


(b)

Figure 5.23: WCOMPM extrapolation of 4–16 GHz FMM, 30–40 GHz PO data, $(M_L, M_H) = (14, 4)$ and $W_R = 100$, (a) signals, (b) errors.



(a)



(b)

Figure 5.24: CDMPM extrapolation of 4–16 GHz FMM, 30–40 GHz PO data, $(M_L, M_H) = (14, 6)$, $W_L = 1000$, (a) signals, (b) errors.

5.4 Extrapolation of Bistatic RCS of Square Patch

In this section, we will replicate the results of Section 5.1, but this time, instead of using only backscattering data, we will extrapolate scattering solutions for all observation angles on the x–z plane. This will form a two dimensional image, one dimension for observation angle of the bistatic RCS and other for the solution frequency. This format is the one that is generally used to plot bistatic RCS in real-life applications. The RCS values in these plots are decibel (dB) values, i.e., in logarithmic scale. Therefore, small errors like 10^{-2} would not change the result very much.

Figures 5.25 and 5.26 illustrate the FMM and PO solutions for the problem described in Section 5.1 respectively. Notice the strong backscattering of this geometry around 0° observation. The modeling intervals are selected similarly, i.e., 1–8 GHz interval for the LF available data and 30–35 GHz for the HF available data. Figure 5.27 shows the result of COMPM. For each horizontal line, i.e., each observation angle, an independent COMPM procedure is carried out; this includes the brute-force M optimization for COMPM. The result seems to agree with both computational solutions, whose difference in dB scale is presented in Figure 5.28. The error of COMPM signal with respect to FMM and PO are shown in 5.29 and 5.30 respectively. From these error plots, it can be seen that most of the error is represented by dark blue, which means error is less than 0.6 dB. This is clearly the best visual representation of what COMPM and extrapolation can achieve. We estimate a gap of 24 Ghz, i.e., 600 solution samples for each observation angle, by only using 14 GHz of data.

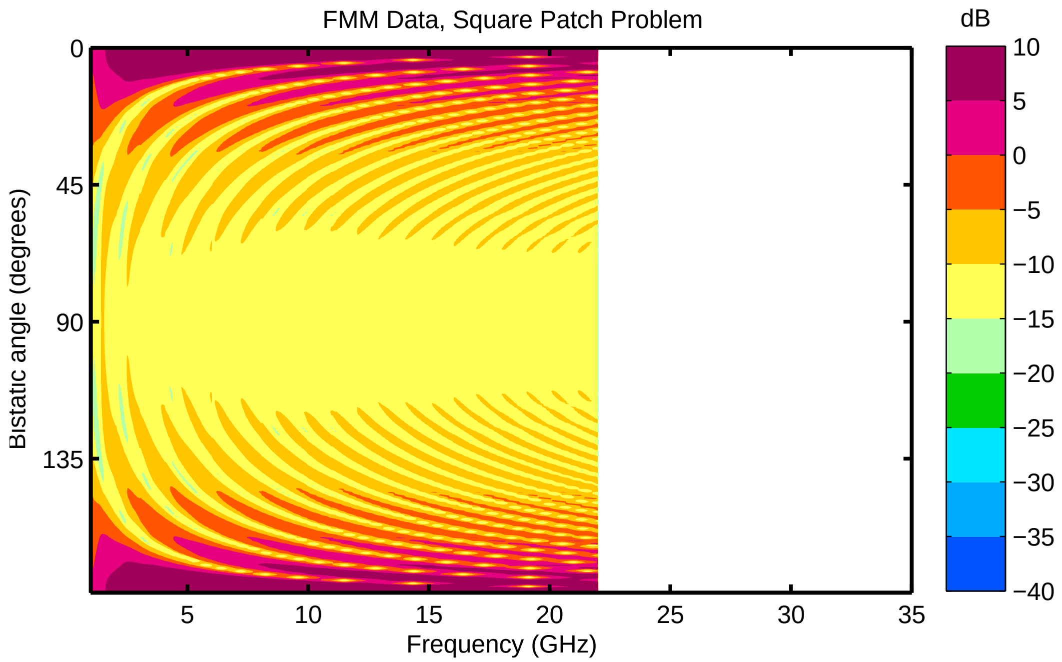


Figure 5.25: Bistatic RCS plot of FMM solution, square patch problem.

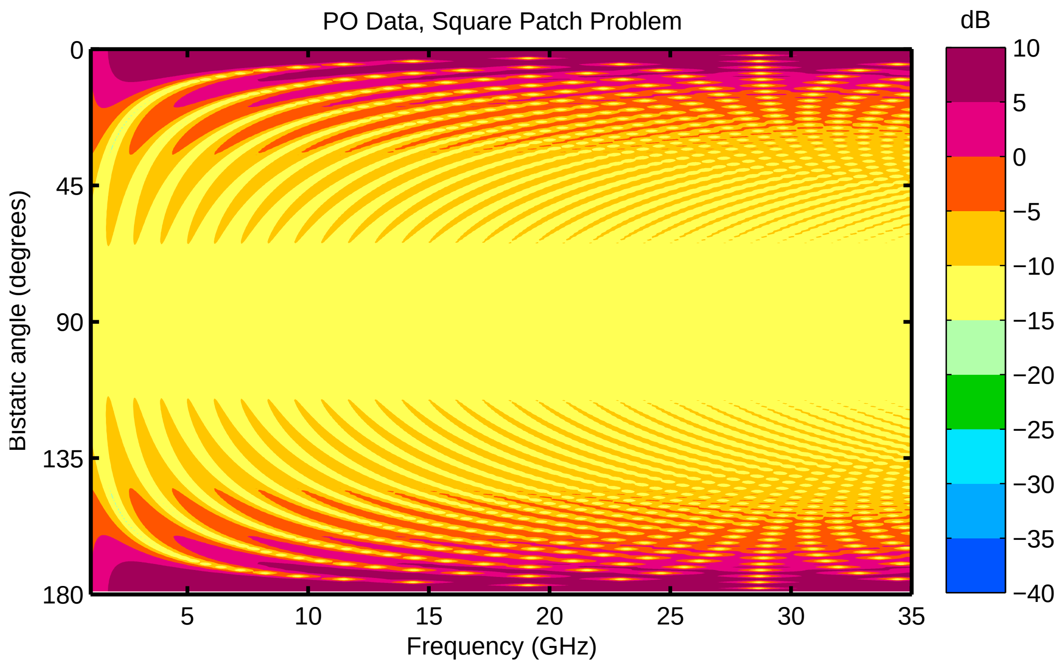


Figure 5.26: Bistatic RCS plot of PO solution, square patch problem.

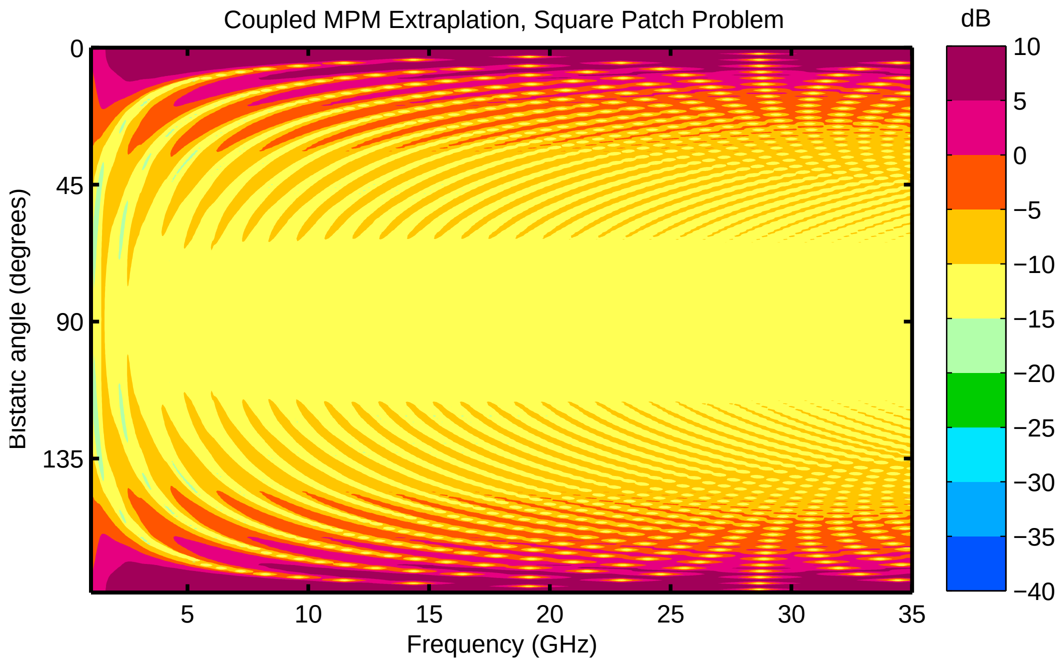


Figure 5.27: Bistatic RCS plot of COMPM extrapolation, square patch problem.

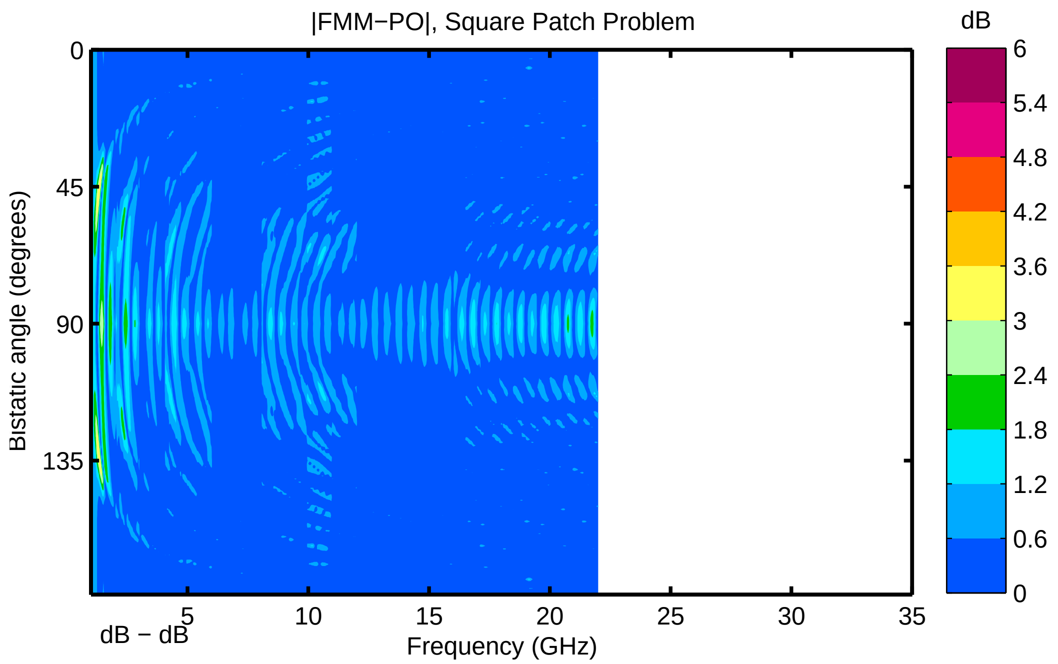


Figure 5.28: Error between FMM and PO data for every observation angle and frequency, square patch problem.

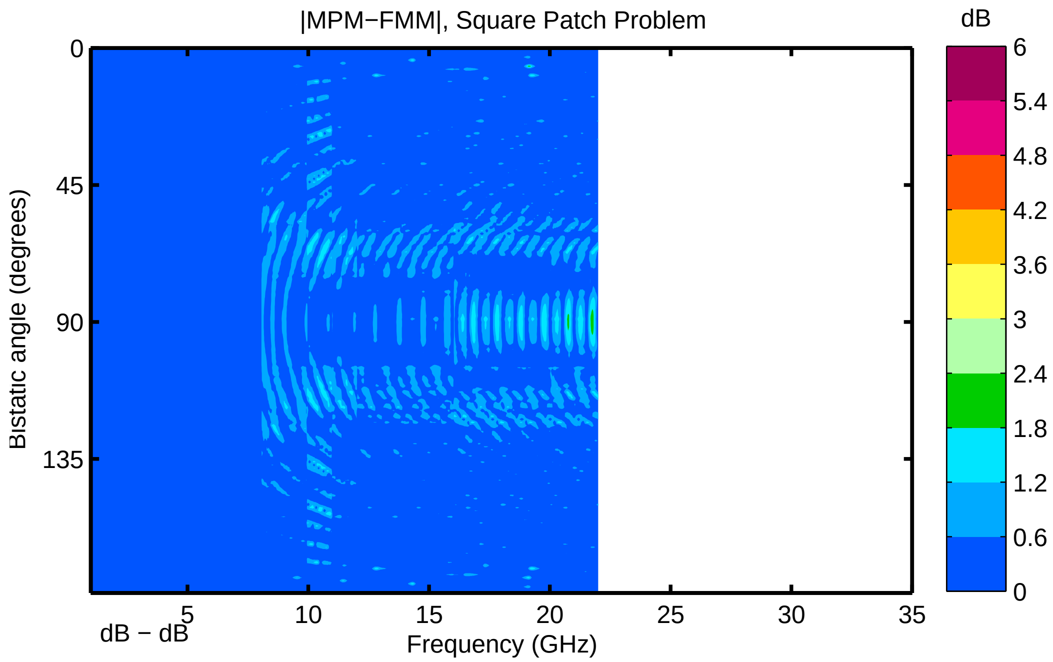


Figure 5.29: Error between FMM data and extrapolated signal for every observation angle and frequency, square patch problem.

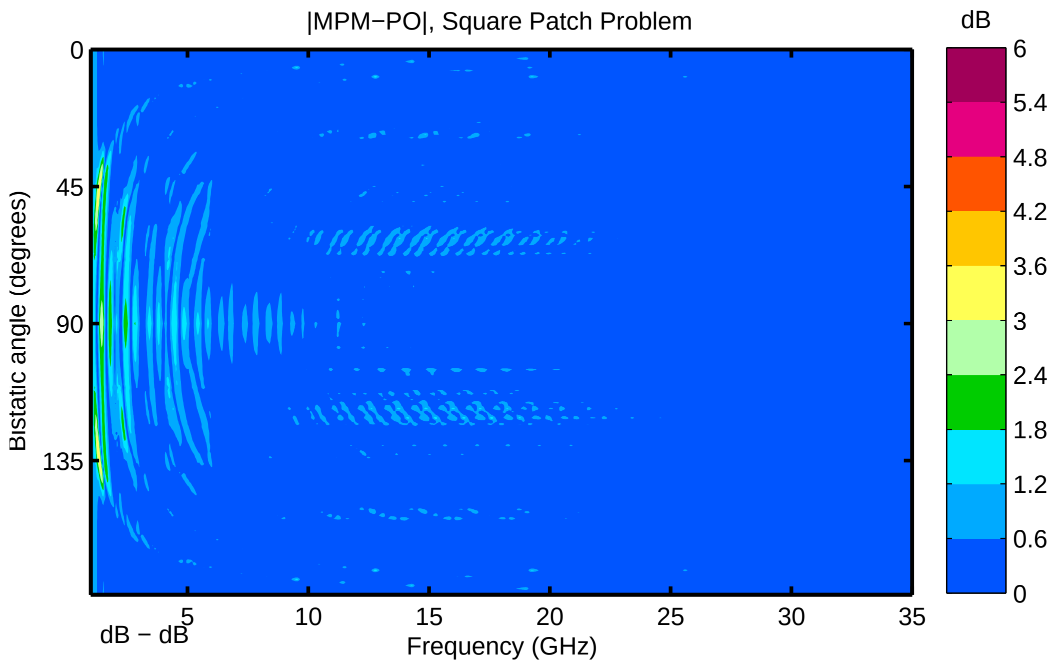


Figure 5.30: Error between PO data and extrapolated signal for every observation angle and frequency, square patch problem.

Chapter 6

Conclusion and Future Work

In this thesis we presented numerous methods to perform extrapolation from available LF and HF solutions of electromagnetic scattering problems. We presented the theory of MPM in Chapter 2, the method which we use to model the available scattering solution of the electromagnetic problem. During the derivation of the theory of MPM, we skipped the theoretical background of pencil parameter L and fixed its value throughout the thesis. The importance of L in the theory of MPM, its effect on the extrapolated signal should be further researched.

In Chapter 3, we proposed two approaches to couple the two independent MPM extrapolations, i.e., FMPM and BMPM, and presented the improvement on the analytically generated backscattering signal of the conducting sphere. In the first coupling approach, namely, COMPM, we couple the models in the parameter level. It requires a new and harder optimization to be performed to select the best possible method for extrapolation. The second coupling approach is CDMPM, in which we couple the extrapolated signals obtained from FMPM and BMPM. When insightful knowledge about pencil parameter L has been acquired, one should also consider optimizing L using the already proposed brute-force optimization. However, with the introduction of L parameter to the optimization algorithm, the optimization space expands to four dimensions. This will require higher computational resources and execution time. A new optimization approach or algorithm should be implemented to avoid this situation. One can also extend this study by proposing a robust algorithm for the selection of windows in CDMPM. In this thesis, although such results are not presented, we tried several different windows and select the best one to use it throughout the study. However, this selected window is selected for one specific problem. The type, length and weight of the windows can vary for each problem; the CDMPM theory is flexible in that way.

In Chapter 4, we discuss the problems that we face when working with computational data. We identified computational noise and error that may be present on the computational data. We identified noise as the possible cause of the degraded modeling performance. On the other hand, computational error is identified as the intrinsic error that is due to the approximations that are done in the beginning of the solution process; this error we can not isolate and combat. We proposed limiting M value in order not to model the noise on the data, so that we would obtain a more clean extrapolated signal. This causes an increase in the extrapolation error, but as long as the maximum error of extrapolation is around the level of estimated noise, this extrapolation is assumed to be successful. In Chapter 4, we also discussed how we should trust FMM data when compared with the PO data. For the cases where FMM data and PO data do not agree, we proposed using weighting in COMPM to increase the influence of the FMM data over PO data. We also introduce weighting to FMPM, and possibly to backward case too, in order to combat with the complex exponents that have magnitudes greater than unity. We observed that these complex exponents cause the extrapolated signal to blow up. Further methods and approaches should be researched to combat such complex exponentials; weighting might not be solely enough to combat them.

In the last chapter, we present examples of extrapolation using computational data on three distinct scatterers. The problems that we present span a wide range of situations. In square patch problem, we have strong backscattering but the scattering signal is monotonous, which causes relatively easy modeling. Moreover, FMM and PO solution data for square patch problem holds. This leads to good extrapolation performance even without using weighting. As this problem has such nicely behaving computational solution, we moved to extrapolate other bistatic angles. 2D plots of RCS are generally used in real-life situations for commenting the behavior of the scatterer. We then present a more complex geometry called patch-over-ring, whose backscattering is still strong but contains more oscillations. This causes the modeling become harder, which is in fact what we intentionally wanted. As the last example we presented rectangular prism geometry. This example represents the situation where the backscattering is weak and slowly oscillating and moreover there is considerable disagreement between FMM and PO data. This last example clearly showed us the success of COMPM.

To conclude, in this thesis we have presented theoretical basis for extrapolation and coupled extrapolation. We present some numerical results that show successful extrapolation results. There are some possible extension points to further improve this research. This thesis should bear enough information to expand the extrapolation research for more complex scatterers such as airborne, military and stealth targets.

Bibliography

[1] O. S. Ergul, *Fast multipole method for the solution of electromagnetic scattering problems*, M. Eng. Thesis, Bilkent University, Ankara, Turkey, June 2003.

[2] C. J. Reddy, M. D. Deshpande, C. R. Cockrell, and F. B. Beck, "Fast RCS computation over a frequency band using method of moments in conjuction with asymptotic waveform evaluation technique" *IEEE Trans. Antennas Propagat.*, vol. 46, no. 1, pp. 1229–1233, Aug. 1998.

[3] Y. Wang, H. Ling, J. Song, and W. C. Chew, "A frequency extrapolation algorithm for FISC" *IEEE Trans. Antennas Propagat.*, vol. 45, pp. 1891–1893, Dec. 1997.

[4] Z. Altman and R. Mittra, "Combining an extrapolation technique with the method of moments for solving large scattering problems involving bodies of revolution" *IEEE Trans. Antennas Propagat.*, vol. 44, no. 1, pp. 548–553, Apr. 1996.

[5] G. J. Burke, E. K. Miller, S. Chakrabarti, and K. Demarest, "Using model-based parameter estimation to increase the efficiency of computing electromagnetic transfer function" *IEEE Trans. Magn.*, vol. 25, pp. 2807–2809, July 1989.

[6] F. B. Hildebrand, *Introduction to Numerical Analysis* New York City, NY: McGraw Hill Press., 1974.

[7] A. G. Evans, "Least square systems parameter identification and time domain appoximation," Ph. D. thesis, Drexel University, 1972.

[8] R. Stieglitz, "On simultaneous estimation of poles and zeros in speech analysis," *IEEE Trans. Acoustic, Speech and Signal Proc.* ASSP-25, pp. 229–234, Jun. 1972.

[9] D. M. Goodman and D. G. Dudley, "An output error model and algorithm for electromagnetic system identification," *Circuits, Systems and Signal Processing*, vol. 6, no. 4, pp. 471–505, 1987.

[10] Tapan K. Sarkar, Odilon Pereira, "Using the matrix pencil method to estimate the parameters of a sum of complex exponentials," *IEEE Antennas and Propagation Mag.*, vol. 37, no. 1, pp. 48–55, Feb. 1995.

[11] Y. Hua and T. K. Sarkar, "Matrix pencil method for estimating parameters of exponentially damped/undamped sinusoids in noise," *IEEE Trans. on Acoustic, Speech and Signal Proc.*, ASSP-38, 5, pp. 81–824, May 1990.

[12] Balanis C. A., *Advanced Engineering Electromagnetics*, John Wiley & Sons, New York, 1989.

[13] Gene H. Golulo, Charles F. Van Loan, *Matrix Computations* Baltimore, MD: Johns Hopkins University Press., 1989.

[14] Raviraj S. Adve, T. K. Sarkar, Odilon Maroja C. Pereira-Filko, Sadasiva M. Rao, "Extrapolation of time domain responses from three-dimensional conducting objects utilizing the matrix pencil technique," *IEEE Trans. on Antennas and Propagation*, vol. 45, no. 1, pp. 147–156, Jan. 1997.

[15] Alan V. Oppenheim, Ronald W. Schafer, John R. Buck, *Discrete-Time Signal Processing* Upper Saddle River, NJ: Prentice Hall, Inc., 1999.

[16] Strang G., *Linear Algebra and its Aplications*, third edition, Harcourt Brace Jovanovich, New York, 1988.

Appendix A

Fourier Domain Analysis

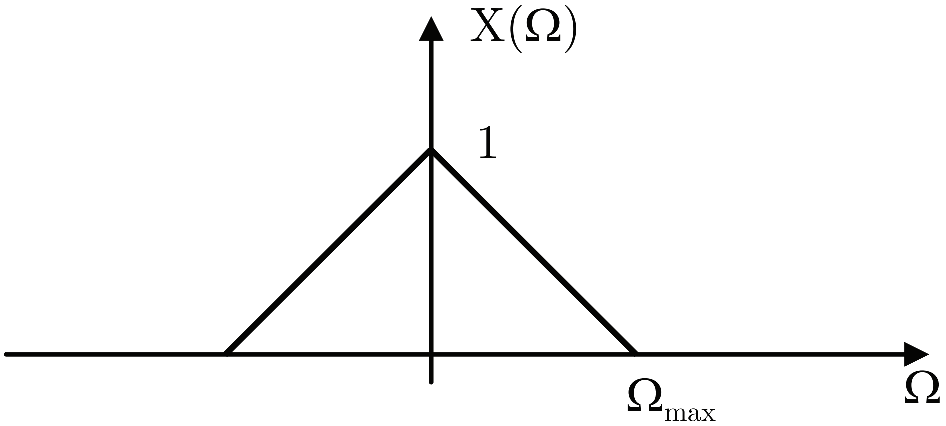


Figure A.1: Sample frequency domain signal.

Let's assume that $x(t)$ is a continuous time signal. We can write this signal in frequency domain by using continuous Fourier transform (CFT) as

$$X(\Omega) = \int_{-\infty}^{\infty} x(t)\, e^{-j\Omega t} dt, \tag{A.1}$$

where Ω is the continuous frequency in radians. Figure A.1 represents the frequency domain response of $x(t)$. The result of (A.1) is continuous. Now assume that $x[k]$ is the sampled version of $x(t)$ with a sampling period of Δt, i.e.,

$$f_s = \frac{1}{\Delta t}, \tag{A.2}$$

$$\Omega_s = \frac{2\pi}{\Delta t}, \tag{A.3}$$

where f_s and Ω_s are sampling frequencies in Hertz and radians, respectively. Sampling equation can be written as

$$x[k] = \sum_{n=-\infty}^{\infty} x\left(n\Delta t\right) \delta\left(t - n\Delta t\right), \tag{A.4}$$

where $\delta(\,)$ represents the impulse function. If we apply CFT to $x[k]$ we will obtain

$$X_d(\Omega) = \int_{-\infty}^{\infty} x[k] e^{-j\Omega t} dt, \tag{A.5}$$

$$= \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x\left(n\Delta t\right) \delta\left(t - n\Delta t\right) e^{-j\Omega t} dt, \tag{A.6}$$

$$= \sum_{n=-\infty}^{\infty} x\left(n\Delta t\right) \int_{-\infty}^{\infty} \delta\left(t - n\Delta t\right) e^{-j\Omega t} dt, \tag{A.7}$$

$$= \sum_{n=-\infty}^{\infty} x\left(n\Delta t\right) e^{-j\Omega n\Delta t}, \tag{A.8}$$

$$X\left(e^{j\omega}\right) = \sum_{n=-\infty}^{\infty} x\left[n\right] e^{-j\omega n}, \tag{A.9}$$

where the discrete frequency ω is defined as,

$$\omega = \Omega\Delta t. \tag{A.10}$$

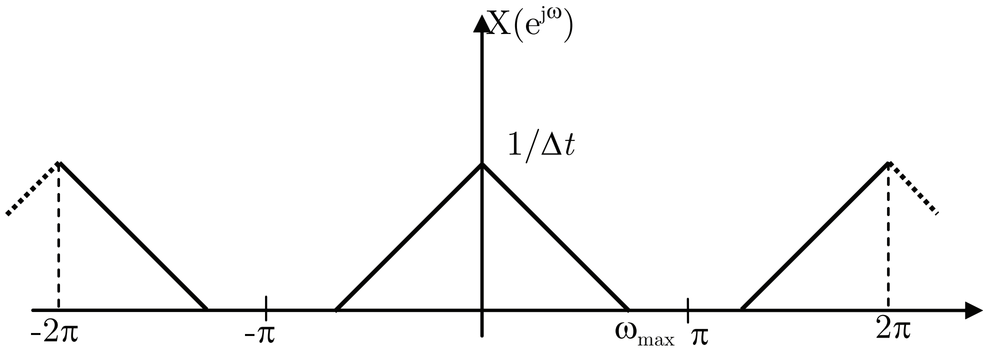


Figure A.2: DTFT representation of the sampled signal.

Equation (A.9) is the definition of discrete time Fourier transform (DTFT). If we assume that the CFT of signal $x(t)$ is the one illustrated in Figure A.1, then the DTFT of the sampled version of $x(t)$ would be like the one in Figure A.2.

Periodicity in DTFT representation is related with the sampling in time domain. Although the time domain signal is discrete, the result of DTFT is still continuous. If the sampling of the signal satisfies the Nyquist criterion, the relation between DTFT and CFT can be written as

`Frequency response of DTFT` $[-\pi : \pi]$ (magnitude $\times \Delta t$; frequency $\times \frac{1}{\Delta t}$) $=$ `Frequency response of CFT`

The problem with DTFT is, we can not process it in a computer. We still need to discretize the frequency domain in order to use it in a computer environment. Therefore we sample ω in the $[-\pi : \pi]$ interval with a frequency sampling of

$$\Delta\omega = \frac{2\pi}{N}, \tag{A.11}$$

where N is the number of frequency samples in the $[-\pi : \pi]$ interval. As DTFT is periodic with 2π, we can also take the interval $[0 : 2\pi]$ to avoid negative index values in the new definition of Fourier transform. Assuming that $x[k]$ is limited in time, i.e., x has non-zero values only for $x[0 : N-1]$, we define the discrete Fourier transform (DFT):

$$X\left[k\right] = \sum_{n=0}^{N-1} x\left[n\right] e^{-j\frac{2\pi}{N}kn}. \tag{A.12}$$

DFT takes discrete valued signal and produces discrete valued spectrum. Fast Fourier transform (FFT) is a very efficient algorithm that implements DFT. Spectrum obtained by DFT can be converted to CFT spectrum with a similar transformation that is performed for DTFT.

A.1 Linear and Circular Convolution

Linear convolution of two time signals, x and y, is defined as

$$c(t) = \int_{-\infty}^{\infty} x(\tau)\, y(t-\tau)\, d\tau. \tag{A.13}$$

If x and y are discrete-time functions, this equation can be written as

$$c[n] = \sum_{m=-\infty}^{\infty} x[m]\, y[n-m]. \tag{A.14}$$

In (A.14), the sampling frequencies of x and y are assumed to be the same. Recall that, convolution in time domain corresponds to multiplication in frequency domain; we can write this identity as

$$\mathcal{F}^{-1}\{\mathcal{F}\{x(t)\}\,\mathcal{F}\{y(t)\}\} = c(t), \tag{A.15}$$

where $\mathcal{F}\{\ \}$ represents the CFT. However, relation (A.15) does not hold for DFT and linear convolution defined in (A.14). We need to define a new type of convolution to incorporate the periodicity of the time domain signal that is introduced with the sampling of the frequency; recall that sampling in time domain also introduces periodicity in frequency domain. This new convolution is circular convolution, which is defined as

$$c[n] = \sum_{m=0}^{N-1} x[\langle m\rangle_N]\, y[\langle n-m\rangle_N], \tag{A.16}$$

where $c[n]$ is a length N time-domain signal. Recall that in linear convolution, $c(t)$ is always longer than either of the convolved signals, however in circular convolution $c[n]$ is the same size with either of the convolved signals. Using circular convolution we can define a relation similar to (A.15) as

$$c[n] = DFT^{-1}\{DFT\{x[n]\}.DFT\{y[n]\}\}. \tag{A.17}$$

To sum up, in continuous case the linear convolution of two signals is the inverse CFT of the product of the CFT of each signal. In discrete case, the circular convolution of two signals is the inverse DFT of the product of the DFT of each signal.

Proof of (A.17):

We can rewrite $c[n]$ as

$$c\left[n\right] = \frac{1}{N}\sum_{k=0}^{N-1}\left(X\left[k\right]Y\left[k\right]\right)e^{j\frac{2\pi}{N}kn}, \tag{A.18}$$

where

$$X\left[k\right] = \sum_{m=0}^{N-1}x\left[m\right]e^{-j\frac{2\pi}{N}km}. \tag{A.19}$$

Thus, we can rewrite (A.18) as,

$$\begin{aligned} c\left[n\right] &= \frac{1}{N}\sum_{k=0}^{N-1}\left\{\sum_{m=0}^{N-1}x\left[m\right]e^{-j\frac{2\pi}{N}km}\right\}Y\left[k\right]e^{j\frac{2\pi}{N}kn} && \text{(A.20)}\\ &= \sum_{m=0}^{N-1}x\left[m\right]\left\{\frac{1}{N}\sum_{k=0}^{N-1}Y\left[k\right]e^{j\frac{2\pi}{N}(n-m)k}\right\}. && \text{(A.21)}\end{aligned}$$

Using the circular shift identity we define

$$h\left[\langle n-m\rangle_N\right] = \frac{1}{N}\sum_{k=0}^{N-1}Y\left[k\right]e^{j\frac{2\pi}{N}(n-m)k}. \tag{A.22}$$

Substituting (A.22) back will result in circular convolution equation (A.16):

$$c\left[n\right] = \sum_{m=0}^{N-1}x\left[m\right]y\left[\langle n-m\rangle_N\right]. \tag{A.23}$$

Example:

Suppose that we have two signals of length 5:

$$x[m] = \{1,2,3,4,5\}$$
$$y[m] = \{5,4,3,2,1\}$$

We shall compare the linear and circular convolution of these two signals and try to understand the concept of time aliasing:

`Linear convolution` : $\{5,14,26,30,27,18,10,4,1\}$

`Circular convolution` : $\{23,24,30,31,27\}$

Linear convolution in matrix form

$$\begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_8 \\ c_9 \end{bmatrix} = \begin{bmatrix} 5 & 0 & 0 & 0 & 0 \\ 4 & 5 & 0 & 0 & 0 \\ 3 & 4 & 5 & 0 & 0 \\ 2 & 3 & 4 & 5 & 0 \\ 1 & 2 & 3 & 4 & 5 \\ 0 & 1 & 2 & 3 & 4 \\ 0 & 0 & 1 & 2 & 3 \\ 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 2 \\ 3 \\ 2 \\ 1 \end{bmatrix}$$

Circular convolution in matrix form

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} 5 & 1 & 2 & 3 & 4 \\ 4 & 5 & 1 & 2 & 3 \\ 3 & 4 & 5 & 1 & 2 \\ 2 & 3 & 4 & 5 & 1 \\ 1 & 2 & 3 & 4 & 5 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 2 \\ 3 \\ 2 \\ 1 \end{bmatrix}$$

Appendix B

SVD on Circular Convolution Matrix

We define convolution as

$$\bar{\mathbf{A}} \cdot \mathbf{x} = \mathbf{c}, \tag{B.1}$$

and deconvolution as

$$\mathbf{x} = \bar{\mathbf{A}}^{-1} \cdot \mathbf{c}. \tag{B.2}$$

Matrix $\bar{\mathbf{A}}$ in (B.1) and (B.2) can be linear or circular convolution matrix as defined in Appendix A. In this thesis, we use circular convolution matrix, hence for a length N window in frequency domain, size of $\bar{\mathbf{A}}$ would be $N \times N$. If there are L nonzero elements in the frequency domain window, the rank of $\bar{\mathbf{A}}$ will be L, too. Therefore, there are $N - L$ very small singular values relative to the largest singular value. This causes ill-conditioning in matrix $\bar{\mathbf{A}}$ and the solution of (B.1) or equivalently calculation of (B.2) will be severely affected. In order to solve such a problem we should eliminate those small singular values by using SVD:

$$\mathbf{x} = \left(\bar{\mathbf{U}} \cdot \bar{\mathbf{\Sigma}} \cdot \bar{\mathbf{V}}^{H}\right)^{-1} \cdot \mathbf{c}. \tag{B.3}$$

Matrices $\bar{\mathbf{U}}$ and $\bar{\mathbf{V}}$ are unitary, hence

$$\mathbf{x} = \bar{\mathbf{V}} \cdot \bar{\mathbf{\Sigma}}^{-1} \cdot \bar{\mathbf{U}}^{H} \cdot \mathbf{c}. \tag{B.4}$$

$\bar{\mathbf{\Sigma}}$ is a diagonal matrix, whose diagonal entries are the singular values ordered in descending order. Inverting such a matrix is performed by simply taking the inverse of each singular value on the diagonal. This will result in very large numbers to appear in the diagonal as we invert very small numbers. In order to obtain a meaningful solution to (B.4), we need to discard either those very large number in $\bar{\mathbf{\Sigma}}^{-1}$ or very small numbers in $\bar{\mathbf{\Sigma}}$. This will result in a meaningful but an approximate solution of $\mathbf{x}$.