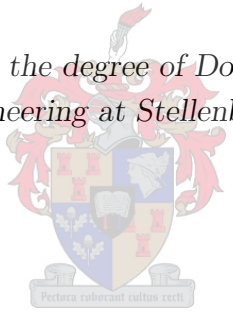
# Efficient Numerical Analysis of Finite Antenna Arrays using Domain Decomposition Methods

by

Danie Ludick

*Dissertation presented for the degree of Doctor of Engineering in the Faculty of Engineering at Stellenbosch University*



Promoters:

Prof. D.B. Davidson
Department of Electrical and Electronic Engineering
Stellenbosch University

Dr. U. Jakobus
Altair Development S.A. (Pty) Ltd

December 2014

# Declaration

By submitting this dissertation electronically, I declare that the entirety of the work contained therein is my own, original work, that the copyright thereof is governed by the Annexure B of the Agreement between EM Software & Systems S.A. (Pty) Ltd and Stellenbosch University duly signed on 11 August 2009 and the related Study Agreement Contract between EM Software & Systems S.A. (Pty) Ltd and myself, and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Signature: .....
D. J. Ludick

Date: November 13, 2014 .....

# Abstract

This work considers the efficient numerical analysis of large, aperiodic finite antenna arrays. A Method of Moments (MoM) based domain decomposition technique called the Domain Green's Function Method (DGFM) is formulated to address a wide range of array problems in a memory and runtime efficient manner. The DGFM is a perturbation approach that builds on work initially conducted by Skrivervik and Mosig for disjoint arrays on multi-layered substrates, a detailed review of which will be provided in this thesis.

Novel extensions considered for the DGFM are as follows: a formulation on a higher block matrix factorisation level that allows for the treatment of a wider range of applications, and is essentially independent of the elemental basis functions used for the MoM matrix formulation of the problem. As an example of this, both conventional Rao-Wilton-Glisson elements and also hierarchical higher order basis functions were used to model large array structures. Acceleration techniques have been developed for calculating the impedance matrix for large arrays including one based on using the Adaptive Cross Approximation (ACA) algorithm. Accuracy improvements that extend the initial perturbation assumption on which the method is based have also been formulated. Finally, the DGFM is applied to array geometries in complex environments, such as that in the presence of finite ground planes, by using the Numerical Green's Function (NGF) method in the hybrid NGF-DGFM formulation.

In addition to the above, the DGFM is combined with the existing domain decomposition method, viz., the Characteristic Basis Function Method (CBFM), to be used for the analysis of very large arrays consisting of sub-array tiles, such as the Low-Frequency Array (LOFAR) for radio astronomy.

Finally, interesting numerical applications for the DGFM are presented, in particular their usefulness for the electromagnetic analysis of large, aperiodic sparse arrays. For this part, the accuracy improvements of the DGFM are used to calculate quantities such as embedded element patterns, which is a major extension from its original formulation.

The DGFM has been integrated as part of an efficient array analysis tool in the commercial computational electromagnetics software package, FEKO.

# Opsomming

In hierdie werkstuk word die doeltreffende analise van eindige, aperiodiese antenna samestellings behandel. Eindige gebied benaderings wat op die Moment Metode (MoM) berus, word as vetrekpunt gebruik. 'n Tegniek genaamd die Gebied Green's Funksie Metode (GGFM) word voorgestel en is geskik vir die analise van 'n verskeidenheid van ontkoppelde samestellings. Die effektiewe gebruik van rekenaargeheue en looptyd is onderliggend in die implementasie daarvan. Die GGFM is 'n perturbasie metode wat op die oorspronklike werk van Skrivervik en Mosig berus. Laasgenoemde is hoofsaaklik ontwikkel vir die analise van ontkoppelde antenna samestellings op multilaag diëlektrikums. 'n Deeglike oorsig van voorafgaande word in die tesis verskaf.

In hierdie tesis is die bogenoemde werk op 'n unieke wyse uitgebrei: 'n hoër blok matriks vlak formulering is ontwikkel wat dit moontlik maak vir die analise van 'n verskeidenheid strukture en wat onafhanklik is van die onderliggende basis funksies. Beide lae-vlak Rao-Wilton-Glisson (RWG) basis funksies, asook hoër orde hierargiese basis funksies word gebruik vir die modellering van groot antenna samestellings. Die oorspronklike perturbasie aanname is uitgebrei deur akkuraatheidsverbeteringe vir die tegniek voor te stel. Die Aanpasbare Kruis Benaderings (AKB) tegniek is onder andere gebruik om spoed verbeteringe vir die GGFM te bewerkstellig. Die GGFM is verder uitgebrei vir die analise van antenna samestellings in 'n komplekse omgewing, bv. 'n antenna samestelling bo 'n eindige grondplaat. Die Numeriese Green's Funksie (NGF) metode is hiervoor ingespan en die hibriede NGF-GGFM is ontwikkel.

Die GGFM is verder met die Karakteristieke Basis Funksie Metode (KBFM) gekombineer. Die analise van groot skikkings wat bestaan uit sub-skikkings, soos die wat tans by die "Low-Frequency Array (LOFAR) " vir radio astronomie in Nederland gebruik word, kan hiermee gedoen word.

In die werkstuk word die GGFM ook toegepas op 'n reeks interessante numeriese voorbeelde, veral die toepaslike EM analise van groot aperiodiese samestellings. Die akkuraatheidsverbeteringe vir die GGFM maak die berekening van elementpatrone vir skikkings moontlik.

Die GGFM is by the sagteware pakket FEKO geintegreer.

# Acknowledgements

I would like to thank the following people, who all played a role in making this research possible:

**My wife Sunel, son Benjamin, and baby girl Amélie** for their love and support.

**My Dad** for introducing me to science at a very early age.

**My Family** for everything that they have done for me.

**Prof. David B. Davidson and Dr. Ulrich Jakobus** for their guidance throughout the years.

**Dr. Rob Maaskant** for valuable discussions, motivation and collaboration that formed a key part of my research and understanding of the field of domain decomposition.

**Prof. Raj Mittra** for collaboration and useful discussions.

**Prof. Matthys Botha** for valuable discussions.

**Dr. Dirk de Villiers** for collaboration on the analysis of the Zig-Zag antenna.

**Dr. Mariana Ivashina and Carlo Bincevenni** for collaboration on sparse antenna design.

**Dr. Evan Lezar** for his support and motivation.

**My co-workers** at EMSS/Altair for their support and input.

**Lucy Weggler** for discussions around the use of the ACA algorithm.

If I forgot anyone, please excuse me. I was probably typing this late in the evening on the day before handing in the thesis - but thank you too!

# Contents

# List of Figures

# List of Tables

# Nomenclature

### Notation

| | |
|---|---|
| $\|\cdot\|_F$ | the Frobenius norm |
| $\mathbf{X}$ | is a matrix or a column vector |
| $X_{ij}$ | the entry at row $i$ and column $j$ of the matrix $\mathbf{X}$ |
| $X_i$ | the $i^{\text{th}}$ element in the vector $\mathbf{X}$ |
| $\vec{x}$ | a spatial vector |
| $\hat{x}$ | a spatial vector with unit norm |

### Constants

| | |
|---|---|
| $c_0 \approx$ | 299792458 $\text{m} \cdot \text{s}^{-1}$ (the speed of light in vacuum) |
| $\pi \approx$ | 3.14159265 |

### Symbols

| | |
|---|---|
| $\mathbf{E}$ | electric field |
| $\mathbf{E}^i$ | incident electric field |
| $\mathbf{E}^s$ | scattered electric field |
| $\epsilon_r$ | relative permittivity of a medium |
| $f$ | frequency |
| $\vec{f}_n$ | the RWG basis function associated with the $n^{\text{th}}$ edge |
| $\mathbf{J}$ | surface current density |
| $\vec{k}$ | propagation vector |
| $\mu_r$ | relative permeability of a medium |
| $\nabla$ | the gradient operator |
| $\Omega_n$ | the $n^{\text{th}}$ domain |
| $\vec{r}$ | the position vector |
| $x$, $y$, $z$ | coordinates in the Cartesian coordinate system |
| $\hat{x}$, $\hat{y}$, $\hat{z}$ | the unit coordinate vectors in the Cartesian coordinate system |

**Abbreviations and acronyms**

| | |
|---|---|
| ACA | Adaptive Cross Approximation |
| ASP | Antenna Signal Processing |
| AIM | Adaptive Integral Method |
| CBFM | Characteristic Basis Function Method |
| CBF | Characteristic Basis Function |
| CEM | Computational Electromagnetics |
| CFIE | Combined Field Integral Equation |
| CPU | Central Processing Unit |
| DDM | Domain decomposition method |
| DGFM | Domain Green's Function Method |
| DGF | Domain Green's Function |
| EEP | Embedded element pattern |
| EFIE | Electric Field Integral Equation |
| EM | Electromagnetic |
| FDTD | Finite-Difference Time-Domain |
| FEM | Finite Element Method |
| FFT | Fast Fourier Transform |
| FMM | Fast Multipole Method |
| FSS | Frequency Selective Surface |
| GO | Geometrical Optics |
| HOBF | Higher Order Basis Function |
| IE | Integral Equation |
| LOFAR | Low-Frequency Array |
| MBF | Macro Basis Function |
| MFIE | Magnetic Field Integral Equation |
| MLFMM | Multilevel Fast Multipole Method |
| MoM | Method of Moments |
| NGF | Numerical Green's Function |
| PBC | Periodic Boundary Condition |
| PEC | Perfect Electric Conductor |
| PO | Physical Optics |
| RCS | Radar cross-section |
| RWG | Rao-Wilton-Glisson |
| SEP | Surface Equivalence Principle |

| | |
|---|---|
| SFX | Synthetic Function Expansion |
| SKA | Square Kilometre Array |
| SP | Speedup |
| VEP | Volume Equivalence Principle |
| UTD | Uniform Theory of Diffraction |

# 1

# Introduction

The numerical analyses of large and complex electromagnetic (EM) structures are of great interest to various scientific and engineering research groups in both the academic and commercial sectors. A class of problems that typically forms part of these studies, is that of electrically large and finite antenna arrays. Applications for antenna arrays are vast and range from radar systems embedded in complex environments [3], for satellite communication [4] and radio astronomy undertakings such as the Square Kilometre Array (SKA) and the Karoo Array Telescope (KAT) [5], to name a few.

Depending on the array configuration and the array elements used, the design and development of these structures can become complex and costly. Numerical simulations are therefore typically used to reduce this design cost and associated risk and equip the antenna designer with a collection of computational electromagnetics (CEM) techniques that may be used for the analysis. To this end, commercial CEM software packages such as FEKO [6] incorporate various methods to analyse EM structures efficiently. These techniques include full-wave methods such as the Method of Moments (MoM) [7], that is suitable for the analysis of small to moderate sized EM problems. For electrically larger problems the designer can resort to using high frequency asymptotic methods such as Physical Optics (PO) [8], Geometrical Optics (GO) [9], the Uniform Theory of Diffraction (UTD) [10] and fast solution techniques[11] such as the Multilevel Fast Multipole Method (MLFMM) [12, 13] and the Adaptive Cross Approximation (ACA) algorithm [14–18]. Furthermore, the use of parallel programming models, in the form of distributed and shared memory parallelisation [19–21] and also GPU acceleration [22], have played a significant role in reducing both the run-time and memory overheads of large simulations.

The typical antenna design processes, an overview of which is presented in Figure 1.1, comprise of various systematic steps, starting with the formulation of the array parameters. These requirements are a quantification of various goals regarding the array performance, e.g. side-lobe levels, beamwidth, gain, polarisation, axial ratio, scattering parameters and radar cross section. For a detailed description of the aforementioned quantities, the reader is directed to well known texts on antenna design, such as [23].

Following the finalisation of the requirements, the numerical analysis begins with the design of the array element (step (a) in Figure 1.1). Depending on the type of array, this step can quickly become complex and time consuming as the designer iterates simulations to obtain an optimal

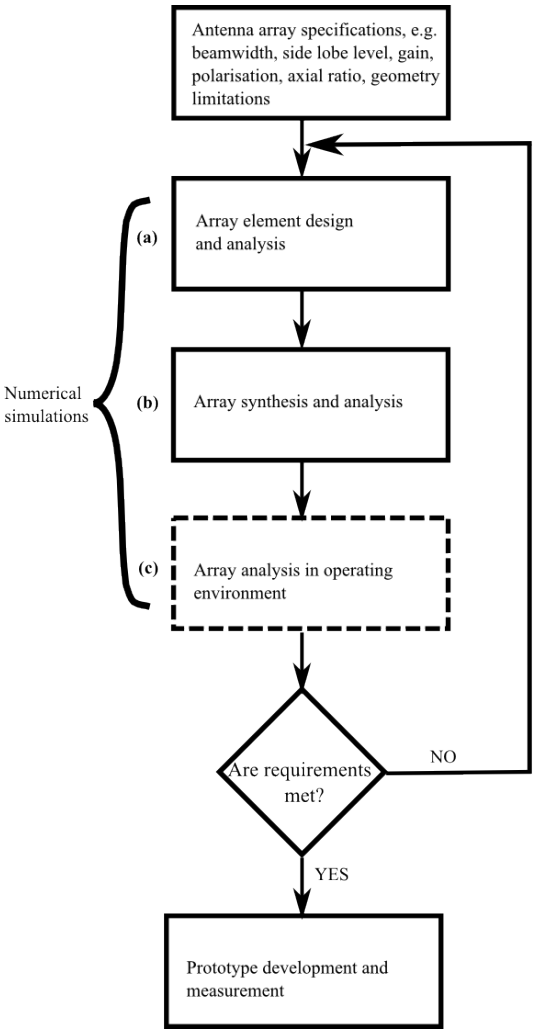



**Figure 1.1:** Array synthesis and analysis.

design. As the number of elements are increased to form the array (step (b) in Figure 1.1), the computational complexity increases significantly and may exceed the available computing resources, resulting in very long simulations. The next step of the design process may be to evaluate the manner in which the array will function in its operating environment. An example of this would be to analyse the efficiency of a phased antenna array that is used as feed-structure for an electrically large parabolic dish reflector antenna or the presence of a finite ground plane. The above mentioned design steps are iterated until the simulated array parameters correspond to the specifications, after which the prototype is manufactured and measured. When using CEM as part of Array Signal Processing (ASP) methods, e.g. that of array thinning, then the number of electromagnetic simulations that must be performed can become quite large [24, 25].

Central to steps (a) to (c) in Figure 1.1, i.e. the design and analysis of the array with CEM techniques, is that the simulations can result in very long run-times and significant memory usage. Depending on the available computing resources, this might introduce significant delays in the design phase, especially when complex array geometries for sensitive applications are being developed.

A typical design strategy followed in engineering to mitigate large complex analyses is the so-called *divide-and-conquer* approach [26], where a complex problem is subdivided into a number of smaller, more manageable sub-problems. Each sub-problem is then solved independently and the results are combined to yield the final solution to the original problem. In the field of CEM, a number of interesting advances have been initiated that conform to this type of *divide-and-conquer* strategy, one of which is that of domain decomposition. In summary, domain decomposition methods can be used to simplify the analysis of large EM problems by solving smaller sub-domains while accounting for mutual coupling between them. This allows for a more efficient solution in terms of memory usage and computational run-time.

Various domain decomposition techniques have been developed, such as the Iterative Field Bouncing (IFB) technique [27], the Characteristic Basis Function Method (CBFM) [28–31] and the Synthetic Function Expansion approach (SFX) [32, 33]. A promising domain decomposition approach is also undertaken by Wang *et al.* in the form of a multisolver domain decomposition method (MS-DDM) [34–36]. The approach followed with the aforementioned, is to decompose the entire computational domain into many subregions, based on local material properties and geometrical features. The most efficient CEM technique is then applied in each domain. Mutual coupling between well-seperated subregions are modelled in the MS-DDM through Stratton-Chu representation formulas [36].

The proposed research was focused on obtaining a MoM-based domain decomposition approach that is suited to a selection of antenna array designs, including irregulary spaced elements with arbitrary excitations. The research was carried out in the framework of the commercial CEM software package FEKO, which includes various solution techniques such as the MoM, Finite Element Method (FEM), PO, GO, UTD and fast solution methods such as the MLFMM and ACA. The research was also focussed on incorporating parallelisation methods in the development as far as possible, using both distributed and shared programming models. A detailed description of the research objectives and the methodology that was followed is presented in the following subsections.

## 1.1 Research objectives

The overarching aim of this research has been to develop efficient numerical analysis techniques for antenna designers to assist them in the array design process as outlined in Figure 1.1. Domain decomposition was utilised in the implementation to improve computational efficiency while preserving a high degree of numerical accuracy.

The main objectives are as follows: For finite array analysis, the objective is that the simulation technique developed meet the following criteria: it should be runtime efficient and memory economical, it should account for boundary effects and the effects of mutual coupling between array elements, and it should determine the antenna performance parameters accurately. Apart from the aforementioned, it is also imperative that a variety of general array problems can be analysed with the method. To this end, the objective has been to include not only perfectly electric conducting (PEC) structures, but also general structures modelled with arbitrary basis

functions. The array geometries investigated should not only be comprised of periodically and equally spaced elements, but should also include irregularly spaced elements with arbitrary excitations. Finally, a further objective has been to develop a way in which to use the fast domain decomposition based array solver in a multi-domain or complex environment, e.g. an antenna array with a finite ground plane, or a focal plane array illuminating a parabolic dish reflector.

Secondary objectives include the integration of the techniques developed into the existing commercial CEM software package FEKO, to allow for a broader scope of problems to be addressed. The methods developed throughout the research project have also been verified and validated continually with the various solution methods already available in the FEKO framework [37–39]. Of particular interest for the finite array solution technique, i.e. the DGFM is to compare the numerical accuracy and computational efficiency against that of the infinite periodic boundary condition (PBC) solution implemented in FEKO [40, 41]. Parallelisation methods were also deemed important in the development of the DGFM by using both distributed and shared programming models, following similar approaches to work done by Ludick *et al.* [42] for the CBFM.

## 1.2 Original contributions

This work presents a number of contributions that address the original research objectives discussed in Section 1.1. The development of a fast domain decomposition based array solution method, viz., the Domain Green's Function Method (DGFM) forms the key contribution. As will be discussed in more detail in Chapter 4, the DGFM in its original form is a perturbation method that is based on work done by Skrivervik and Mosig [43–46]. A number of novel extensions to the above mentioned have been added in this thesis to address the objectives as set out in the previous subsection. These constributions, with references to the present author's published work, are as follows:

1. The DGFM is formulated on a higher block matrix factorisation level to allow for the treatment of a wider range of problems. It can be applied to both conventional Rao-Wilton-Glisson (RWG) basis functions [47] and also hierarchical higher order basis functions (HOBFs) [48, 49]. A list of publications on the DGFM, its formulation and applications can be found in [50–52].

2. A more accurate version of the DGFM, viz. the improved DGFM or i-DGFM [53, 54] has been formulated, to allow also for the calculation of embedded element patterns in a passive array environment. This presents a major extension to the original work done by Skrivervik and Mosig in [43–46].

3. The DGFM is hybridised with an existing domain decomposition method, the Characteristic Basis Function Method (CBFM), for the analysis of very large arrays consisting of sub-array tiles, such as those used at LOFAR [55].

4. The hybrid NGF-DGFM allows for the analysis of array geometries in complex environments, such as those in the presence of finite sized ground planes [56].

5. Different acceleration strategies for the DGFM (and i-DGFM) have been developed, that include a manual approach whereby the active impedance matrix summation is truncated [53] or approximated through the use of the ACA algorithm [57].

6. An efficient parallelisation strategy using a hybrid distributed and shared memory programming paradigm has also been applied to accelerate the DGFM on multi-core architectures [53].

Finally, interesting numerical applications for the DGFM are also presented, in particular its usefulness for the electromagnetic analysis of sparse, irregular antenna arrays. The DGFM has also been integrated as part of an efficient array analysis tool in the commercial computational electromagnetics software package FEKO.

## 1.3 Chapter summary

A brief summary of the thesis is as follows: In Chapter 2 the MoM will be reviewed as it forms the basis of various discussions and derivations in later Chapters. In Chapter 3 an overview of various existing array analysis techniques will be presented. The DGFM formulation is discussed in Chapter 4 where a rigorous comparison is also made with the CBFM to distinguish it from the aforementioned. Furthermore, acceleration strategies for the method are also explained in this Chapter. Simulation results are provided that illustrate the numerical accuracy and computational complexity of the DGFM. In Chapter 5 the accuracy improved version of the DGFM, the i-DGFM, is presented. The i-DGFM allows for the analysis of phased array simulations and the calculation of embedded element patterns.

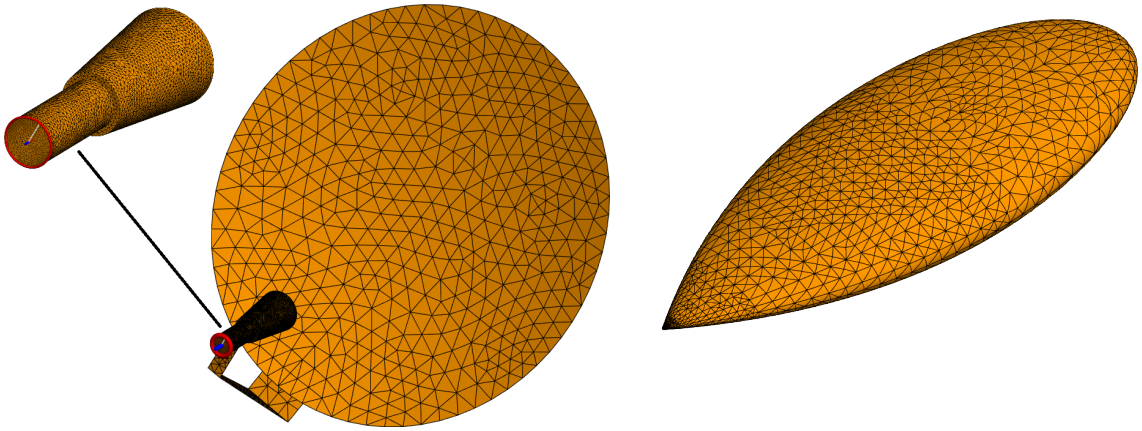
In Chapter 6 the application of the DGFM and i-DGFM is extended through the hybridisation with other domain decomposition (DD) techniques, viz. the CBFM and NGF, respectively. The hybrid CBFM-DGFM allows for the analysis of large disjoint arrays consisting of subarray tiles. The analysis of finite arrays in the presence of other structures is also presented in Chapter 6 with the hybrid NGF-DGFM approach. A summary and general conclusions are presented in Chapter 7 that includes recommendations for future research. In addition, Appendix A summarises the key steps of the partially pivoted ACA algorithm, that accelerates the calculation of the active impedance matrix equations for the DGFM, as discussed in Chapter 4.

# 2

# The Method of Moments

This Chapter presents a detailed overview of the Method of Moments (MoM) that forms the basis for the various techniques presented in the following Chapters. The focus is directed towards three dimensional perfectly conducting radiators and scatterers and the underlying theory is based on that presented by Rao, Wilton and Glisson [47]. The use of higher order hierarchical basis functions (HOBFs) [48, 49] on curvilinear triangles will also be discussed as means of extending the capabilities of the conventional subsectional RWG basis functions.

The underlying goal of the frequency domain MoM approach is to obtain a mathematical expression for the discretised current distribution on an arbitrary electromagnetic scatterer or radiator. In Figure 2.1 examples of open and closed structures are presented. The structures are modelled with triangular patch elements, which are capable of accurately conforming to nearly any geometrical surface. The NASA almond [58] shown in Figure 2.1(b) is discretised with curvilinear triangles that allows one to model curved structures more accurately.



(a) (b)

**Figure 2.1:** Examples of (a) open and (b) closed structures modelled with triangular patches.

Integral equations can be formulated for the currents flowing on the surface of the structure. The type of equation used is dependent on whether the structure is open or closed. For closed structures, the magnetic field integral equation (MFIE) can be used which in general leads to more well-posed equations. In this context, a well-posed problem is one for which the solution is not strongly dependent on the physics of the geometry of the problem [59]. The other type of integral equation, i.e., the electric field integral equation (EFIE) has the advantage that it can be used for both open and closed problems. A combination between the MFIE and EFIE, termed the combined field integral equation (CFIE), can also be used for closed structures. In domain decomposition applications, the problem is typically subdivided into a number of smaller open sub-problems. For this reason the focus in this Chapter is limited to a discussion based on the EFIE.

This Chapter is structured as follows: In the following Section, the EFIE is presented, that relates the induced surface current distribution on a conducting body to the scattered electric field by applying certain boundary conditions. In Section 2.2 the RWG basis function will be introduced as a means of solving the EFIE through the application of the MoM. A testing/weighting procedure that is applied to generate a set of linearly independent equations is discussed in Section 2.3. The latter is used to formulate the MoM matrix equation as explained in Section 2.4. In Section 2.5 higher order hierarchical basis functions (HOBFs) are introduced as a means of improving the computational complexity associated with solving the MoM matrix equation. The Chapter is then concluded in Section 2.6.

## 2.1 The Electric Field Integral Equation

The EFIE defines a relationship between the electric field, $\mathbf{E}$, and the electric current distribution, $\mathbf{J}$, and is derived from Maxwell's equations [7]. To explain its use, consider an electric field, $\mathbf{E}^i$, incident on any of the PEC structures depicted in Figure 2.1 consisting of surfaces only. The scattered electric field, $\mathbf{E}^s$, can be computed from the induced surface current as follows,

$$\mathbf{E}^s = -j\omega\mathbf{A} - \nabla\Phi, \tag{2.1}$$

with $\nabla$ the gradient operator, that is defined as,

$$\nabla = \frac{\partial}{\partial x}\hat{x} + \frac{\partial}{\partial y}\hat{y} + \frac{\partial}{\partial z}\hat{z}, \tag{2.2}$$

with $\hat{x}$, $\hat{y}$, $\hat{z}$ the unit coordinate vectors in the Cartesian coordinate system, respectively. The magnetic vector potential, $\mathbf{A}$, and scalar potential, $\Phi$, in (2.1) can be defined as follows,

$$\mathbf{A}(\vec{r}) = \frac{\mu}{4\pi}\int_S \mathbf{J}(\vec{r}\,')G(\vec{r},\vec{r}\,')dS', \tag{2.3}$$

$$\Phi(\vec{r}) = \frac{1}{4\pi\epsilon}\int_S \sigma G(\vec{r},\vec{r}\,')dS'. \tag{2.4}$$

The quantity $G(\vec{r},\vec{r}\,')$ is the scalar free-space Green's function and is defined as

$$G(\vec{r},\vec{r}\,') = \frac{e^{-jk_0R}}{R}, \tag{2.5}$$

where the term $R = |\vec{r} - \vec{r}\,'|$ is the distance between an arbitrarily located source point $\vec{r}\,'$ and observation point $\vec{r}$ on the structure and $k_0 = 2\pi/\lambda$ is the free space wavenumber. The scalar quantity, $\sigma$, is the surface charge density and is related to the divergence of the surface current density, $\mathbf{J}$, through the continuity condition as follows,

$$\nabla_S \cdot \mathbf{J} = -j\omega\sigma. \tag{2.6}$$

In the above, $\nabla_S$ denotes the gradient operator that is applied to the surface $S$. To relate the scattered and electric fields, we use the boundary condition for the tangential component of the electric field on the PEC structure that can be written as follows,

$$\hat{n} \times \mathbf{E} = \hat{n} \times (\mathbf{E}^i + \mathbf{E}^s) = 0, \tag{2.7}$$

where $\hat{n}$ is the unit normal for the surface, $S$. By using (2.7) we can now enforce the following integro-differential equation,

$$-\mathbf{E}^i_{\tan} = (-j\omega\mathbf{A} - \nabla\Phi)_{\tan}, \text{ with } \vec{r} \text{ on } S \tag{2.8}$$

Equations (2.8), with (2.3) to (2.6), forms the EFIE, that relates the *unknown* surface current distribution, $\mathbf{J}$, to the *specified* incident electric field, $\mathbf{E}^i$. The first step in obtaining a solution for (2.8), is to represent the unknown surface current distribution with a set of basis functions, as discussed in the following subsection.

## 2.2 The RWG basis function

The set of basis functions used to model the surface current distribution, $\mathbf{J}$, needs to be suited to both the triangular-patch elements used to represent the geometry, as well as the EFIE used for the underlying formulation. The basis function selected for this is known as the RWG basis function [47] and is mentioned throughout the text. In this subsection, the RWG basis function will be investigated in more detail, which will also form the foundation for the higher order basis functions that will be discussed in Section 2.5.

The final goal is to use the basis functions to obtain a suitable approximation for the current distribution, $\mathbf{J}$, on the surface $S$, as follows,

$$\mathbf{J} = \sum_{n=1}^{N_{\text{RWG}}} I_n \vec{f}_n(\vec{r}), \tag{2.9}$$

where $I_n$ is an unknown complex expansion coefficient for the basis function, $\vec{f}_n(\vec{r})$. The meaning and properties of the aforementioned will be discussed in the remainder of this subsection.

Consider Figure 2.2(a) and (b) that illustrates a triangular pair in two and three dimensions,

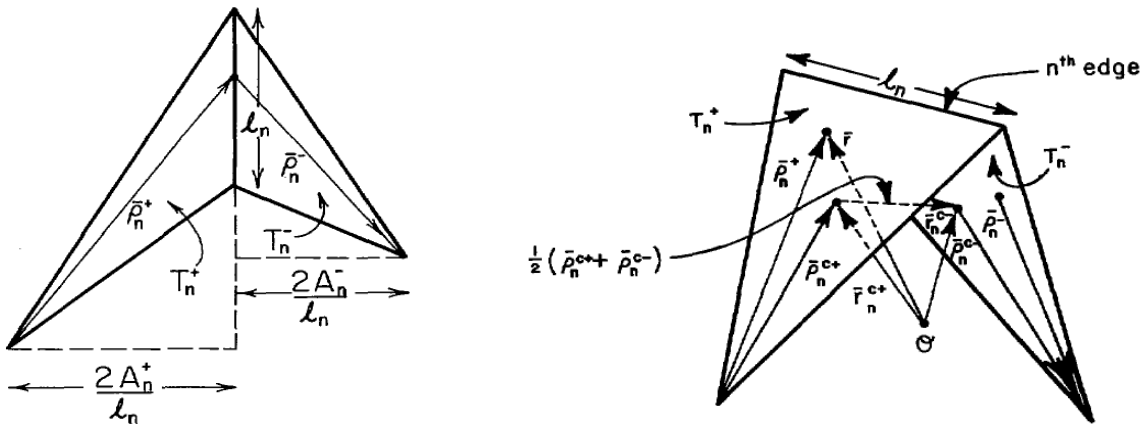



**Figure 2.2:** The Rao-Wilton-Glisson (RWG) basis function (*cf.* Figure 2 and Figure 3 in [47]). The diagram illustrates a triangular pair forming a surface that shares an internal (i.e. non-boundary) edge in (a) two dimensions and (b) three dimensions.

respectively. The first and foremost observation is that the basis function $\vec{f}_n(\vec{r})$ is associated with an *interior* edge (i.e. a non-boundary edge). The basis function associated with the $n$th edge is zero on every other triangle, except for the triangles bordering that edge, i.e., $T_n^+$ and $T_n^-$. The points in any of the triangles may be designated either by the position vector, $\vec{r}$, with respect to a global coordinate origin, or by the local position vector, $\vec{\rho}^{\pm}$ which is defined with respect to the free-vertex of $T_n^{\pm}$, respectively, as shown in Figure 2.2. The $\pm$ sign that is associated with each of the triangles is used to define the reference direction of the RWG element associated with the $n$th edge. The aforementioned is assumed to be from triangle $T_n^+$ to triangle $T_n^-$.

The vector basis function $\vec{f}_n(\vec{r})$ can now be defined as follows,

$$\vec{f}_n(\vec{r}) = \begin{cases} l_n/(2A_n^+)\vec{\rho}_n^+ & \text{if } \vec{r} \text{ in } T_n^+ \\ l_n/(2A_n^-)\vec{\rho}_n^- & \text{if } \vec{r} \text{ in } T_n^- \\ 0 & \text{otherwise} \end{cases} \tag{2.10}$$

where $l_n$ is the length of the $n$th edge, $A_n^{\pm}$ are the areas of triangles $T_n^{\pm}$, respectively, and $\vec{\rho}_n^{\pm}$ is defined as in Figure 2.2. The RWG basis function proves to be a useful choice for representing the current flow between a triangular pair due to the following reasons [47]:

1. The current component normal to the boundary formed by triangles $T_n^{\pm}$ is zero, which means that no fictitious line charge exists along this boundary.

2. The current component normal to the $n$th edge is constant and continuous across this edge. This is enforced by the normal components of the vectors, $\vec{\rho}_n^{\pm}$, being equal to the inverse of the coefficients of $\vec{f}_n(\vec{r})$, i.e., $2A_n^{\pm}/l_n$, respectively. The latter is also illustrated in Figure 2.2(a). The reason for doing so, is to ensure that the $n$th edge is also free of fictitious line-charge densities. The normal current density component across the $n$th edge

is unity.

3. The surface charge density that is proportional to the surface divergence of the vector function, $\vec{f}_n(\vec{r})$, can be calculated as follows,

$$\nabla \cdot \vec{f}_n(\vec{r}) = \begin{cases} l_n/A_n^+ & \text{if } \vec{r} \text{ in } T_n^+ \\ -l_n/A_n^- & \text{if } \vec{r} \text{ in } T_n^- \\ 0 & \text{otherwise ,} \end{cases} \tag{2.11}$$

which shows that the charge density is constant in each of the triangles, and that the *total* charge associated with triangles $T_n^{\pm}$ is zero.

By directing focus back to (2.8) it is clear that a vector basis function is associated with each of the $N_{\text{RWG}}$ non-boundary edges. Furthermore, the coefficients $I_n$ can be interpreted as the normal component of surface current density flowing over the $n$th edge. The reason for this is that for any given non-boundary edge $n$, only the basis function $\vec{f}_n(\vec{r})$ has a normal component of current flowing over that edge and, as specified in 2), that component is unity.

As illustrated in this Section, the RWG vector basis functions are essentially independent of one another. This is an important observation when considering the manner in which each of the unknown expansion coefficients, $I_n$ in (2.8) will be calculated, which forms the focus of the following Section.

## 2.3 A weighting/testing procedure

At this stage it is clear that by substituting the expression for the surface current $\mathbf{J}$ in (2.9) into the EFIE, i.e., (2.8), with (2.3) to (2.6), we are left with a single equation containing $N_{\text{RWG}}$ unknowns. Solving this equation with the MoM entails that one incorporates the use of testing functions. The purpose of the testing functions is to generate $N_{\text{RWG}}$ linearly independent equations from which the $N_{\text{RWG}}$ unknown expansion coefficients can be determined. We choose as testing function the RWG basis function, $\vec{f}_n$, discussed in the previous Section (note that for brevity we now omit the position vector, $\vec{r}$). By doing so we are applying Galerkin's method [7][1] that will be explained in the remainder of this Section.

We begin by defining the symmetric inner-product between two vector functions, $\mathbf{f}$ and $\mathbf{g}$, as

$$\langle \mathbf{f}, \mathbf{g} \rangle = \int_S \mathbf{f} \cdot \mathbf{g} \, dS. \tag{2.12}$$

The expression for testing the EFIE equation, i.e., (2.8) with $\vec{f}_m$ where $m = 1, \ldots, N_{\text{RWG}}$ can be written as follows,

$$\langle \mathbf{E}^i, \vec{f}_m \rangle = j\omega \langle \mathbf{A}, \vec{f}_m \rangle + \langle \nabla \Phi, \vec{f}_m \rangle. \tag{2.13}$$

[1] As will be noted at the end of the Section, the method explained here is not a true Galerkin approach, due to approximations that are made for the integration.

From [47] it follows that the last term in (2.13) can be expressed as,

$$\langle \nabla\Phi, \vec{f}_m \rangle = -\int_S \Phi \nabla \cdot \vec{f}_m \; dS. \tag{2.14}$$

We can then further utilise the expression for the surface divergence of $\vec{f}_m$, as shown in (2.11), to rewrite (2.14) as follows,

$$\int_S \Phi \nabla \cdot \vec{f}_m \; dS = l_m \left(\frac{1}{A_m^+} \int_{T_m^+} \Phi \; dS - \frac{1}{A_m^-} \int_{T_m^-} \Phi \; dS\right). \tag{2.15}$$

The term $\int_{T_m^\pm} \Phi \; dS$ represents a double integration that involves both the basis function $\vec{f}_n$ and testing function $\vec{f}_m$, respectively, and may be written using (2.4) as follows,

$$\int_{T_m^\pm} \Phi \; dS = \frac{1}{4\pi\epsilon} \int_{T_m^\pm} \left[ \int_{T_n^\pm} \sigma G(\vec{r}, \vec{r}\,') dS' \right] \; dS. \tag{2.16}$$

In [47], the double integration in (2.16) is replaced by a single integration by approximating the average of the scalar potential over each of the triangles $T_n^\pm$ as the value of $\Phi$ at the triangle-centroids as follows,

$$\int_{T_m^\pm} \Phi \; dS \simeq A_m^\pm \Phi(\vec{r}_m^{\,c\pm}) \tag{2.17}$$

By using (2.17), we can simplify (2.15) as follows,

$$\int_S \Phi \nabla \cdot \vec{f}_m \; dS \simeq l_m \left[ \Phi(\vec{r}_m^{\,c+}) - \Phi(\vec{r}_m^{\,c-}) \right]. \tag{2.18}$$

Similarly, the first term in (2.13) can be written as follows,

$$\langle \mathbf{E}^i, \vec{f}_m \rangle \simeq \frac{l_m}{2} \left[ \mathbf{E}^i(\vec{r}_m^{\,c+}) - \mathbf{E}^i(\vec{r}_m^{\,c-}) \right], \tag{2.19}$$

and the second term in (2.13) that involves the magnetic vector potential can be written as,

$$\langle \mathbf{A}, \vec{f}_m \rangle \simeq \frac{l_m}{2} \left[ \mathbf{A}(\vec{r}_m^{\,c+}) - \mathbf{A}(\vec{r}_m^{\,c-}) \right]. \tag{2.20}$$

The approximations made in (2.18) to (2.20) are justified in [47] by noting that the potentials are locally smooth within the triangular subdomain spanned by $T_n^\pm$. This follows from their integral definitions, as well as the locally smooth nature of the source representation in terms of the RWG basis function.

It should be noted that the approximations in (2.18) to (2.20) do not lead to a true Galerkin approach as explained in [7]. Instead, the methodology outlined in this Section can be seen as a point-matching scheme [59]. A Galerkin method involves a double integration over both the source and observation domains and leads to symmetric entries[2] of the MoM matrix equation. For this integration, an adaptive scheme can be used that uses more integration samples when the source and observation regions are closely spaced, and fewer when the regions are well

[2]This depends on the numerical integration over the source and observation regions.

seperated.

## 2.4 The MoM matrix equation

By using the discretisation of the surface current density in (2.9) with the weighted EFIE equation in (2.13) and applying the approximations made in (2.18) to (2.20) we obtain an $N_{\text{RWG}} \times N_{\text{RWG}}$ matrix equation that may be written as follows,

$$\mathbf{ZJ} = \mathbf{V} \tag{2.21}$$

where $\mathbf{J}$ is a vector containing the unknown expansion coefficients, $I_n$, associated with each of the RWG basis functions defined in (2.9). The MoM impedance matrix, $\mathbf{Z}$, can be written as follows,

$$\mathbf{Z}_{mn} = l_m \left[ j\omega \left( \mathbf{A}^{+}_{mn} \cdot \frac{\vec{\rho}^{\,c+}_m}{2} + \mathbf{A}^{-}_{mn} \cdot \frac{\vec{\rho}^{\,c-}_m}{2} \right) + \Phi^{-}_{mn} - \Phi^{+}_{mn} \right], \tag{2.22}$$

with the magnetic vector potentials defined as,

$$\mathbf{A}^{\pm}_{mn} = \frac{\mu}{4\pi} \int_S \vec{f}_n(\vec{r}\,') G(\vec{r}^{\,c+}_m, \vec{r}\,') dS', \tag{2.23}$$

and the electric scalar potential written as,

$$\Phi^{\pm}_{mn} = -\frac{1}{4\pi \; j\omega \; \epsilon} \int_S \nabla' \cdot \vec{f}_n(\vec{r}\,') G(\vec{r}^{\,c+}_m, \vec{r}\,') dS'. \tag{2.24}$$

In both (2.23) and (2.24), the homogeneous free-space Green's function, $G(\vec{r}, \vec{r}\,')$, are defined according to (2.5). The excitation vector, $\mathbf{V}$, can be calculated accordingly as follows,

$$\mathbf{V}_m = l_m \left( \mathbf{E}^i(\vec{r}^{\,c+}_m) \cdot \frac{\vec{\rho}^{\,c+}_m}{2} + \mathbf{E}^i(\vec{r}^{\,c-}_m) \cdot \frac{\vec{\rho}^{\,c-}_m}{2} \right). \tag{2.25}$$

A typical model that is used for the incident field, $\mathbf{E}^i$, is the so-called *feeding-edge model* [60] that is used to apply an excitation at an edge between two triangles. When only wires are used, then the *magnetic frill source* [59] may be applied. The aforementioned, as well other excitation models that are included in the commercial CEM package FEKO [6], e.g. microstrip ports, are used in later Chapters.

One drawback to the approximations made in the previous Section, i.e. in (2.18) to (2.20), is that the symmetry of the impedance matrix deteriorates such that $\mathbf{Z}_{mn} \neq \mathbf{Z}_{nm}$. This would typically not be the case should a double integration over both the source and testing region be performed with an equal number of sampling points. This desirable symmetry property of the impedance matrix leads to a saving in both memory and runtime when solving the MoM matrix equation in (2.21).

The numerical evaluation and storage of the MoM matrix elements is dominated by calculating the impedance matrix, that scales as $\mathcal{O}(N^2_{\text{RWG}})$. A number of numerical integrations are carried out for each of the matrix elements, which can be accomplished by using numerical

quadrature schemes, specifically developed for triangular subdomains [61, 62]. One particular case that should be treated carefully arises when integrating the Green's function, or its divergence, i.e., (2.23) and (2.24), for singular or near singular terms. This singularity for the Green's function occurs when the source and observation points coincide, i.e., when $\vec{r} \simeq \vec{r}'$. Two popular methods by which the Green's function singularity can be treated are by means of subtraction [63], or cancellation [64–66].

In the singularity subtraction approach, terms having the same asymptotic behaviour as the integrand at the singularity are first subtracted from the integrand, leaving a bounded difference integrand that may be integrated numerically. The subtracted singular term is then analytically integrated and the result added back to the numerically integrated terms to complete the potential evaluation. This method, although widely used, has a number of drawbacks [64], that include increased complexity associated with the analytical expression for complex bases, geometry and Green's functions involved.

To extend the capabilities, accuracy, and maintainability of general-purpose CEM codes, the subtraction method can be replaced by a purely numerical quadrature scheme. This scheme employs singularity cancellation methods in which a change of variables is chosen such that the Jacobian of the transformation cancels the singularity [64]. In contrast to the singularity subtraction method, the resulting integrand is analytic in the transformed variables on the element geometry, and hence is amenable to numerical integration [64–66].

To conclude this chapter our focus now shifts to the solution of (2.21), the simplest being a so-called direct approach that may be written as follows,

$$\mathbf{J} = \mathbf{Z}^{-1}\mathbf{V}. \tag{2.26}$$

As explained in [59] this approach is seldom followed due to the high cost associated with calculating the inverse, $\mathbf{Z}^{-1}$. Instead, the impedance matrix, $\mathbf{Z}$, is factored into the product of a lower and upper triangular matrix such that $\mathbf{Z} = \mathbf{LU}$ where $\mathbf{L}$ is the lower triangular matrix and $\mathbf{U}$ is the upper triangular matrix, respectively. This is termed LU-factorisation or LU-decomposition and can be substituted into (2.21), followed by a number of backward substitutions to obtain the unknown current expansion vector, $\mathbf{J}$. The computational complexity associated with this matrix factorisation is $\mathcal{O}(N_{\text{RWG}}^3)$ and dominates the solution process to a prohibitive degree as the problem becomes electrically large. This can readily be the case, as a typical guideline used for the discretisation is $\lambda/10$. In the next section, the use of higher order basis functions will be discussed as a means of allowing a coarser discretisation, while still maintaining a sufficient degree of accuracy.

## 2.5 Higher order basis functions

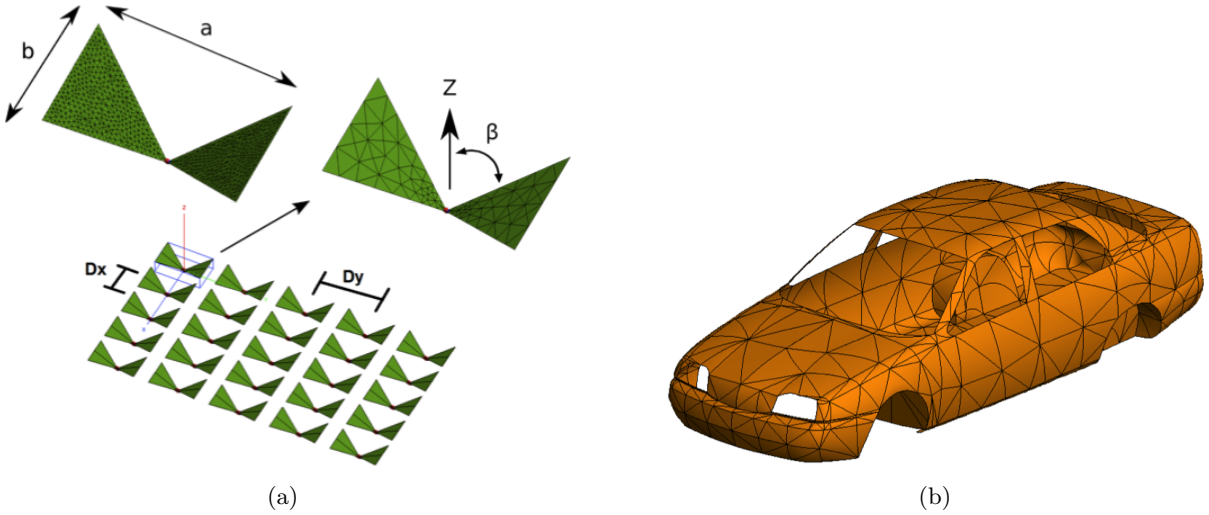
Consider the geometries shown in Figure 2.3(a) and (b), where larger planar triangles are shown for the elements of a finite antenna array and curvilinear triangles that model a vehicle. The discretisation that is used is much coarser than that allowed for conventional RWG basis functions.

To accurately capture the current distribution on the structure, higher order basis functions (HOBFs) [48, 49, 67, 68] are used. The HOBFs are applied to larger triangles, which reduces the number of unknowns and also the computational complexity associated with storing and solving the MoM matrix equation. HOBFs also converge faster to the true solution compared to RWGs when increasing the number of unknowns. An additional benefit is that different orders of current approximation can be used on different elements, making it particularly suitable for error estimation by selectively increasing the polynomial order of certain elements. In addition, curvilinear triangular elements can also be used, to allow for more accurate modelling of arbitrary curved structures (as shown in Figure 2.3(b)).

Examples of edge-based HOBFs of order 0.5 and 3.5 are illustrated in Figure 2.4(a) and (b), respectively. Face-based HOBFs also exist [48], but are not shown here. Edge-based HOBFs of order 0.5 correspond to the conventional RWG element introduced in Section 2.2.

To quantify the benefit of using HOBFs, consider Table 2.1 that gives the HOBF order versus the degrees of freedom (DOF) per triangle, the average triangular patch size as well as the number of Gaussian integration samples as done in [1].

To quantify the benefit of HOBFs, we repeat here the results shown in [1] where the mono-static RCS of a metre long missile was calculated at 3 GHz. The RCS is computed over 180°

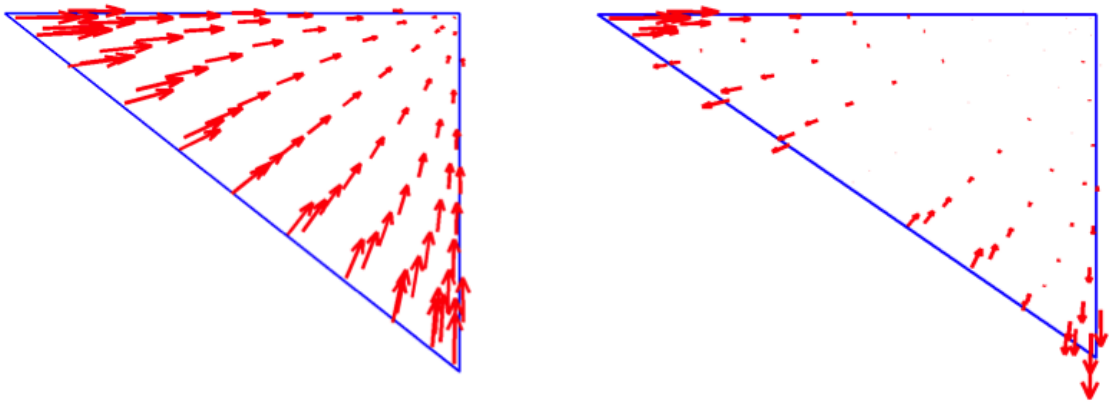



(a) (b)

**Figure 2.3:** (a) Finite antenna array consisting of bow-tie elements that is meshed with planar triangles of side lengths in the order of $\lambda/10$ (top left) and $\lambda$ (top right). (b) Vehicle meshed with curvilinear elements of the second order.

**Table 2.1:** Parameters showing the number of unknowns and triangle size that correspond to a given HOBF order (adapted from [1]).

| Order | DOFs/triangle | Average patch size ($\lambda$) |
|---|---|---|
| 0.5 | 3 | 0.1 |
| 1.5 | 8 | 0.3 |
| 2.5 | 15 | 0.5 |
| 3.5 | 24 | 0.8 |



(a) (b)

**Figure 2.4:** Edge-based HOBFs of order (a) 0.5 and (b) 3.5

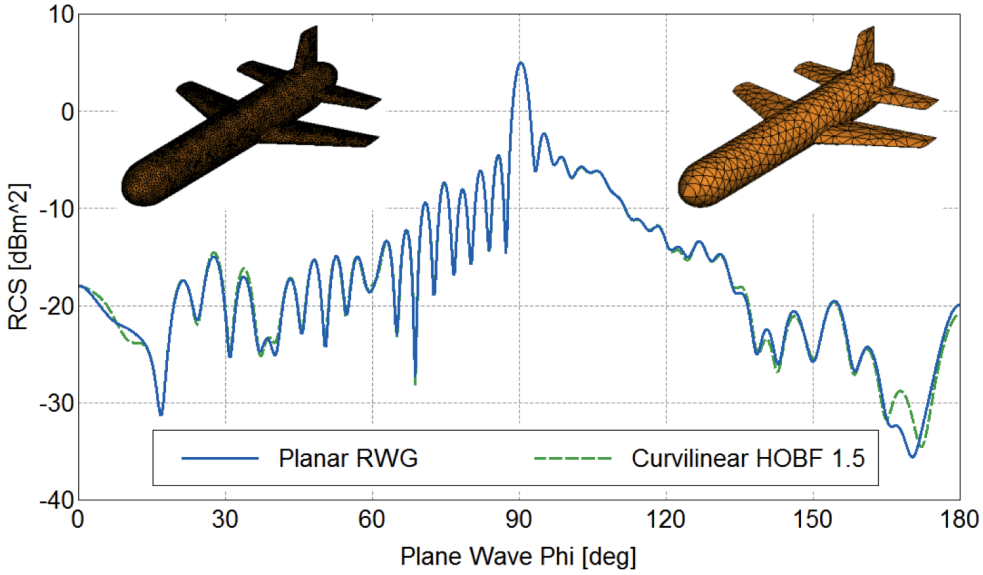



**Figure 2.5:** Monostatic RCS of a missile at 3 GHz calculated using HOBFs and conventional RWG basis functions (as presented in [1]).

for every 0.25° at 721 points. Curvilinear HOBFs of order 1.5 were applied to the structure and the results are shown in Figure 2.5. Excellent agreement between the planar RWG and curvilinear HOBFs can be seen. The number of planar triangles is 19,272 resulting in 28,908 RWG unknowns. For the HOBFs only 1,174 curvilinear triangles were used, reducing the number of unknowns by a factor of 4.9 to only 5,870. With HOBFs the overall run-time is 6 times faster, and the total memory usage is $4.9^2 = 24$ times smaller.

In Section 4.5.3 a bow-tie antenna array is modelled using HOBFs (and also RWG basis functions) and solved using the standard MoM and DGFM, respectively.

## 2.6 Conclusion

The purpose of this Chapter was to provide a sufficient introduction of the MoM as well as its underlying formulation in terms of basis functions, the weighting procedure and the numerical evaluation of the integrals to derive the MoM matrix equation. This will form the underlying basis for the discussions in the remainder of the thesis, especially the MoM matrix equation discussed in Section 2.4, that is used as starting point for the domain decomposition approaches discussed throughout the thesis. The use of HOBFs was also presented as a means of reducing the computational overhead associated with modelling electrically large structures, by using higher order polynomial basis functions. Applying HOBFs in connection to the DGFM will also form part of an example in Chapter 4.

The discussions in this Chapter were limited to simple PEC cases, although in practice various enhancements for the MoM exist, e.g. the Surface Equivalence Principle (SEP) and Volume Equivalence Principle (VEP) [69, 70] for the treatment of dielectric and magnetic bodies; the multi-layered planar Green's function for treating stratified media [71]; and iterative based solution methods such as the MLFMM [12, 13] for analysing electrically large problems.

In the following Chapter, we will consider the MoM-based solution methods of a particular class of problem, viz., that of array antennas.

# 3

# An overview of array analysis techniques

*"Why consider finite arrays?" The short answer to this question is, "Because they are the only ones that really exist."*

– Ben A Munk, *Finite Antenna Arrays and FSS [72]*

Antenna array analysis has been a topic of interest for various decades. The shape and size of these structures can be periodic, such as the type of antenna arrays used in RADAR applications [3], Frequency Selective Surfaces (FSS) [72] and recently in Electromagnetic Band Gap (EBG) materials [73] and Metamaterials [74]. Likewise, various applications of aperiodic, i.e. irregular arrays, have also been of interest, especially in applications such as the SKA [5] where aperiodicity may mitigate the effect of mutual coupling between array elements [75]. In addition finite antenna array elements can be connected, disconnected or any combination thereof. Only disjoint antenna array elements will be considered in this work, which nonetheless cover a wide variety of antenna arrays.

Various successful computational methods have been developed over the years for the analysis of finite (periodic and aperiodic) and infinite antenna array structures as noted in Section 1. A discussion of those methods will be presented in this Chapter. For electrically small array configurations, full-wave methods, such as the MoM can be used. Even for moderately sized arrays, one can consider accelerated methods such as the MLFMM [12, 13] or the ACA [18]. For larger arrays, the aforementioned may become inefficient, as the array spacing and multi-scale discretisation may lead to ill-conditioned problems for which convergence of the underlying iterative Krylov subspace [76] might not be achieved. In this case, computationally expensive preconditioning has to be applied in order to mitigate this effect. For this reason, the focus in this thesis was directed towards direct solution techniques. Special attention is given to the computational complexity associated with the aforementioned, as memory and runtime can be a limiting factor for using direct methods to analyse large structures.

A brief overview of the Chapter is as follows: In Section 3.1 the classical array element pattern methods are discussed as a means of approximating large finite array characteristics. In Section 3.2, the infinite array approach that applies periodic boundary conditions is explained.

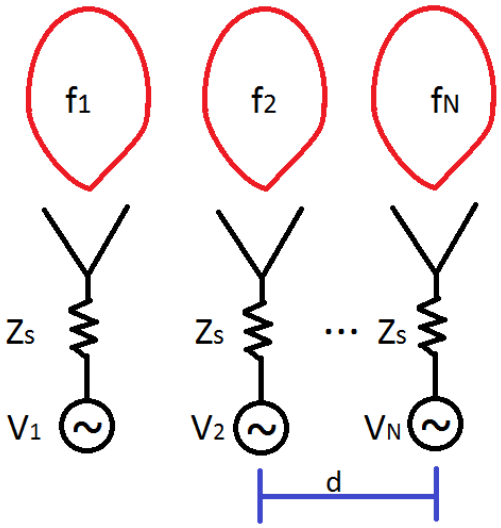



**Figure 3.1:** A uniform $N$ element linear array.

A perturbation method is also reviewed that allows for modelling finite arrays by modifying the infinite array approach with a windowing function. In Section 3.3 the focus is shifted towards the analysis of finite arrays, specifically irregular geometries. Special attention is given to macro basis function (MBF) methods, that is used as part the formulation for the improved Domain Green's Function Method (i-DGFM) presented in Chapter 5.

## 3.1 Array pattern multiplication methods

### 3.1.1 Classical element pattern multiplication

A classic and computationally inexpensive approach to obtain the radiation characteristics of a finite antenna array is by means of array pattern synthesis [23, 77], whereby the results of the elementary cell are multiplied with the array factor. To explain the method, consider the array geometry shown in Figure 3.1 that consists of $N$ identical elements, spaced a distance $d$ apart[1]. The classical array pattern multiplication approach is based on the premise that each of the elements has the same element pattern, which is taken as that of an isolated element in free space. Superposition may then be used to calculate the total field of the fully excited array, $\mathbf{E}(\theta, \phi)$, as follows,

$$\mathbf{E}(\theta, \phi) = \mathbf{f}_{\text{isol}}(\theta, \phi) \sum_{n=1}^{N} I_n e^{-jk\vec{r}\cdot\vec{r}_n}, \tag{3.1}$$

where $\mathbf{f}_{\text{isol}}(\theta, \phi)$ is the isolated element pattern, $I_n$ is the current at the input port of the antenna, $\vec{r}$ is the beam steering position in $(\theta, \phi)$, and $\vec{r}_n$ is the array element position relative to the coordinate reference. The term $e^{-jk\vec{r}\cdot\vec{r}_n}$ represents the spatial phase term that shows the dependence of the pattern on the array geometry. The terminal current $I_n$ is related to the

[1]For the sake of simplicity we consider only linear array geometry here, but the methods can also be applied to planar or 3D arrays.

applied array excitation, $V_n$ and source impedance, $Zs$, as follows,

$$I_n = \frac{V_n}{Zs} \tag{3.2}$$

For the classical approach it is clear from (3.2) that mutual impedance between the array elements are neglected. The method assumes that the element current distributions have the same relative spatial variation along each element. As noted in [77], this assumption is not valid in many cases, such as for arrays of electrically large or different elements. Nevertheless, patterns computed using classical analysis are useful for certain arrays encountered in practice, particularly single mode antennas, that are not too closely spaced.

### 3.1.2 Embedded element pattern method

From [78] the classical element pattern method can be improved by accounting for mutual coupling by using the embedded element pattern (EEP) method. For this technique, the element pattern is calculated by exciting only a specific element in the array while all the others are terminated in matched loads. Equation (3.1) is therefore modified as follows,

$$\mathbf{E}(\theta,\phi) = \sum_{n=1}^{N} \mathbf{f}_n(\theta,\phi) V_n e^{-jk\vec{r}\cdot\vec{r}_n}, \tag{3.3}$$

where $\mathbf{f}_n(\theta,\phi)$ is the embedded element pattern and (3.3) is referred to as the *phase-adjusted unit-excitation active element pattern method*. In [77] an approximation for very large periodic arrays is also discussed, whereby the embedded element patterns for elements near the centre of the array are considered to be similar and replaced with an average embedded element pattern. In Chapter 5, the embedded element patterns of a large irregular array will be calculated as part of an example that illustrates the accuracy improvements obtained by the i-DGFM.

## 3.2 Infinite array simulations

A convenient and computationally efficient aproach to solving the active current distribution on electrically large periodic structures, is the infinite array method [79–81]. In Figure 3.2 an example of a two-dimensional infinite array lattice consisting of identical array elements is shown. The array is created by translating a unit cell, i.e. the array element, to an infinite number of equally spaced positions seperated by distances $d_x$ and $d_y$ in the $xy$-plane, respectively. The position for an element at position $(m,n)$ in the lattice may be written as follows,

$$\vec{\rho}_{mn} = (md_x)\hat{x} + (nd_y)\hat{y}, \text{ with } (n,m) \in \mathbb{N}. \tag{3.4}$$

The excitations that are local to the unit cell are also copied and therefore limits this analysis to the active array environment. The phase of the elements can however differ. This active current distributions are calculated through the use of a periodic Green's function as discussed in the following subsection.

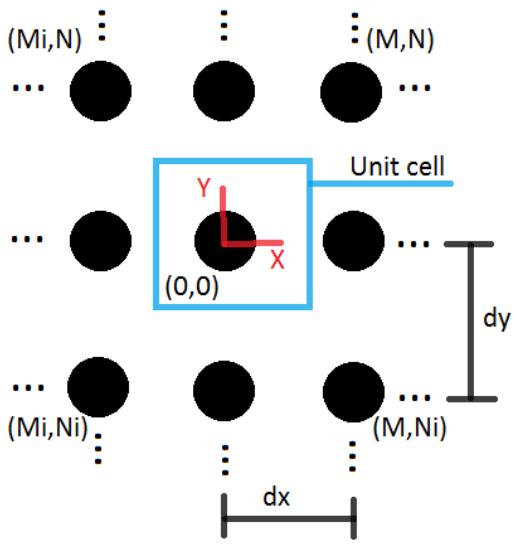



**Figure 3.2:** A 2-dimensional planar periodic array geometry where the unit cell is shown.

### 3.2.1 The periodic Green's function

As shown in [79] the free-space periodic Green's function for the unit cell in Figure 3.2 may be written in the spatial domain as follows,

$$G_p(\vec{r},\vec{r}\,') = \sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} e^{-j\vec{k}_{t00}\cdot\vec{\rho}_{mn}}\frac{e^{-jk_0R_{mn}}}{R_{mn}}, \tag{3.5}$$

where the term $R_{mn} = |\vec{r}-\vec{r}\,'-\vec{\rho}_{mn}|$ is the distance between an arbitrarily located source point $\vec{r}\,'$ and observation point $\vec{r}$ on the structure. The aforementioned is also modified with the term $\vec{\rho}_{mn}$ to account for the periodicity of the structure. The free space wavenumber, $k_0$, was already defined in Section 2.1. The vector wavenumber, $\vec{k}_{t00}$, can be expressed as follows,

$$\vec{k}_{t00} = k_0(\sin\theta_0\cos\phi_0\hat{x} + \sin\theta_0\sin\phi_0\hat{y}), \tag{3.6}$$

where $(\theta_0, \phi_0)$ are the spherical coordinates corresponding to the scan angles of a phased array, or the arrival angles of an incident plane wave [79].

The same expression for the periodic Green's function in (3.5) can be rewritten in the spectral domain as noted in [79]. Equation (3.5) (as well as its spectral representation) is an infinite summation, the convergence of which can be accelerated significantly by using the Ewald transform [82, 83]. The periodic Green's function in (3.5) can also be adapted for layered dielectric media as explained in [84].

An important aspect of (3.5), from a computational viewpoint, is that the analysis domain is limited to that of the unit cell only, i.e., only the unit cell is evaluated. A matrix equation of dimension $N_i \times N_i$ (with $N_i$ the number of unknowns in the unit cell) is solved with the MoM in a memory and runtime efficient manner. Mutual coupling from the surrounding array environment is accounted for through (3.5).

The inherent limitation with the infinite array solution using (3.5), is that the currents will

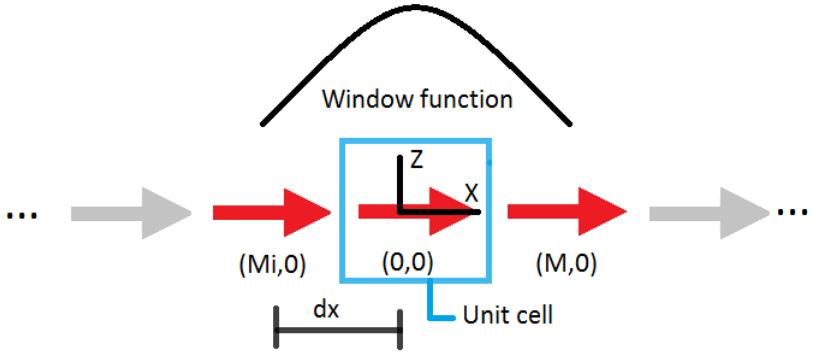



**Figure 3.3:** A 1-dimensional periodic array geometry where the unit cell, as well as a windowing function is shown.

have the same magnitude distribution, but the phase may differ. Edge effects are therefore not accounted for and can become non-negligible, especially for moderate sized arrays. This may prove a significant problem, especially for sensitive applications. In Chapter 4, the infinite array method implemented in FEKO [40, 41], the MoM, and the DGFM, will be compared in terms of computational complexity as well as numerical accuracy at the hand of an example. In the following subsection we consider a windowing approach to introduce a first order correction of the infinite array solution to account for edge effects.

### 3.2.2 Windowing approach for the infinite array method

As noted in the previous subsection, the infinite array approach does not accurately predict the behaviour of radiating elements near the edge of the array. An interesting method to include the edge effects in the infinite array solution was introduced by Ishimaru *et al.* [85] where the infinite array solution is convolved with an appropriate window function. The derivation is based on the use of Poisson's sum formula in the case of infinite sums. In [44], this work was extended by Skrivervik *et al.* that presented a similar approach for the analysis of phased arrays of microstrip patches. In [44] field quantities are used in the convolution process to obtain the active current distribution on each element, as opposed to circuit quantities as is done in [85].

To explain the approach followed in [44], consider the infinite periodic array of elementary $\hat{x}$-directed current sources shown in Figure 3.3. The array lattice corresponds to the planar periodic array in Figure 3.2 for which only the $ZX$-plane is shown in Figure 3.3 for simplicity. The unit cell is illustrated as well as a so-called *windowing function* that tapers the current distribution of a subset of the elements. Each of the cells shown in Figure 3.3 contains only a single source, and following the convention in [44], the cells are numbered starting from the centre of the array. The elementary $\hat{x}$-directed current source in cell $(0,0)$ can be written as follows,

$$\mathbf{J}_{00} = \delta(\xi, \nu)\hat{x}, \tag{3.7}$$

where $(\xi, \nu)$ is the location of the source in cell $(0,0)$. By using (3.7), the excitation for the

array can then be written as follows,

$$\mathbf{J}_{mn} = \mathbf{J}_{00} w(md_x, nd_y) e^{-j\vec{k}_{t00}\cdot\vec{\rho}_{mn}} \tag{3.8}$$

where $\vec{k}_{t00}$ is already defined in (3.6) and accounts for both the periodic array lattice as well as the scan angle $(\theta_0, \phi_0)$. The windowing function, $w(md_x, nd_y)$, can be defined as follows,

$$w(md_x, nd_y) = \begin{cases} 1 & \text{if } m = n = 0 \\ \text{real} & \text{if } M_i \leq m \leq M \text{ and } N_i \leq n \leq N \\ 0 & \text{otherwise .} \end{cases} \tag{3.9}$$

The amplitude of the cell located at $(0, 0)$ is normalised to unity. The electric field due to these elementary sources at a point located inside array element $(k, l)$ can then be written as follows,

$$\mathbf{E}_n(x_k, y_l) = \sum_{m=M_i}^{M} \sum_{n=N_i}^{N} \overline{\overline{\mathbf{G}}}(x_k, y_l \mid \xi_i + d_x(m-k), \nu_l + d_y(n-l))\hat{x} \\ \frac{w(md_x, nd_y)}{w(kd_x, ld_y)} e^{-jk_0(m-k)d_x Tx} e^{-jk_0(n-l)d_y Ty}, \tag{3.10}$$

where $Tx = \sin\theta_0 \cos\phi_0$ and $Ty = \sin\theta_0 \sin\phi_0$ and $\overline{\overline{\mathbf{G}}}(x, y \mid \xi, \nu)$ is the dyadic Green's function [59] for the electric field at a point $(x, y)$ due to a source located at $(\xi, \nu)$. In (3.10) the electric field is evaluated at point $(x, y)$ in cell $(k, l)$, which is written as $(x_k, y_l)$. As done in [44, 85], Poisson's sum formula for finite summations can now be used to transform this finite summation into an infinite summation by using the following identity [85],

$$\sum_{n=N_i}^{N} f(n) = \sum_{m=-\infty}^{\infty} \int_{N_i}^{N} f(\nu) e^{-j2m\pi\nu} d\nu. \tag{3.11}$$

By applying (3.11) to (3.10) the electric field at point $(x_k, y_l)$ can now be rewritten as the convolution of the infinite array solution and the Fourier transform of the amplitude taper on the array (normalised for the cell $(k, l)$ being evaluated), as follows,

$$\mathbf{E}_n(x_k, y_l) = \Big\{ \frac{2\pi}{d_x d_y} \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \tilde{\overline{\overline{\mathbf{G}}}}(k_x, k_y)\hat{x} \\ \cdot e^{-jk_x(x_k - \xi_k)} e^{-jk_y(y_l - \nu_l)} \Big\} * W_{MN}^{kl}(Tx, Ty), \tag{3.12}$$

where $k_x = -k_0 Tx + \frac{2\pi m}{d_x}$ and $k_y = -k_0 Ty + \frac{2\pi n}{d_y}$ . The tilde $\tilde{()}$ represents the Fourier transform and the asterisk $(*)$ the convolution product. The spectral window function, $W_{MN}^{kl}$, is calculated

as follows,

$$W_{MN}^{kl}(Tx,Ty) = \frac{1}{4\pi^2}\int_{(M_i-k)d_x}^{(M-k)d_x}\int_{(N_i-1)d_y}^{(N-1)d_y}\frac{w(m+k,n+1)}{w(k,l)}e^{-jmk_0Tx}e^{-jnk_0Ty}d_m d_n$$
$$= \frac{1}{2\pi}\tilde{w}_{MN}^{kl}(Tx,Ty) \tag{3.13}$$

From (3.12) and (3.13) a representation for the *finite array Green's function* [44] can be written as follows,

$$\overline{\overline{\mathbf{\Gamma}}}_{MN}^{kl}(x_k,y_l \mid \xi_k,\nu_l) = \Big\{\frac{2\pi}{d_x d_y}\sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty}\tilde{\overline{\overline{\mathbf{G}}}}(k_x,k_y)$$
$$\cdot\, e^{-jk_x(x_k-\xi_k)}e^{-jk_y(y_l-\nu_l)}\Big\} * W_{MN}^{kl}(Tx,Ty), \tag{3.14}$$

The first term in brackets is the spectral representation of the infinite array Green's function, and the second term, $W_{MN}^{kl}$, is the window function that accounts for the array excitation law. Equation (3.14) can be used to calculate the active current distribution on cell $(k,l)$. The computational benefit of this approach is that the evaluation of the infinite periodic Green's function is done only once and then convolved with the window function associated with each of the elements considered part of the finite array. If an excitation taper is applied to the array, then the window functions for the elements will differ, i.e. $W_{MN}^{kl} \neq W_{MN}^{mn}$. This in turn leads to a unique Green's function for each of the array elements, i.e., $\overline{\overline{\mathbf{\Gamma}}}_{MN}^{kl} \neq \overline{\overline{\mathbf{\Gamma}}}_{MN}^{mn}$ and in turn to different current distributions on the elements. In Section 3.3.1 this approach is applied in the spatial domain to allow also for general, irregular array layouts and the aforementioned will be explained in more detail.

## 3.3 Finite array simulations

For moderate sized arrays, and for those that require a high degree of flexibility in terms of array lattice geometry (such as aperiodicity) it is preferable to use alternative methods to the infinite array approach. It is to be noted that a number of well-known methods have also been developed for finite periodic array geometries exist, such as the Adaptive Integral Method (AIM) [86] and the pre-corrected Fast Fourier Transform (pFFT) method [87]. The methods that will be discussed in the next two subsections are suited to the analysis of arbitrary spaced array elements and forms the basis of work that follows in the next two Chapters.

### 3.3.1 Finite summation of the Green's function

In this subsection we will consider a space domain formulation of the infinite array plus windowing method discussed in Section 3.2.2, as introduced by Skrivervik *et al.* [43, 45, 46, 88]. The main drawback with the spectral domain formulation for the finite array Green's function in (3.14) is that it applies only the periodic array lattices. By formulating the same windowing technique in the spatial domain, irregular array geometries such as that shown in Figure 3.4 can

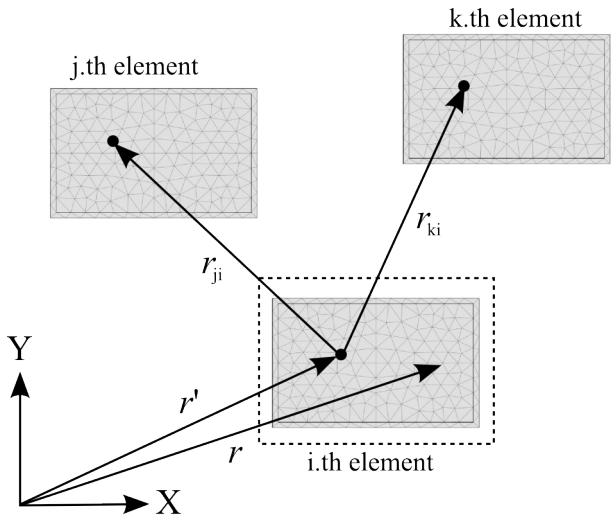



**Figure 3.4:** Antenna array consisting of $M$ elements with $N_i$ basis functions each

be analysed.

The basic premise behind this method, is to begin with the *infinite array assumption*, as done in [45], i.e. to assume that the unknown surface currents on all the array elements are identical to the surface current on the element that we wish to analyse, scaled by the complex array excitation coefficients. The aforementioned hypotheses may be written as follows,

$$\mathbf{J}_k(\vec{r}+\vec{r}_{ki}) = \frac{C_k}{C_i}\mathbf{J}_i(\vec{r}) = \alpha_{ki}\mathbf{J}_i(\vec{r}), \tag{3.15}$$

where $\mathbf{J}_{k,i}$ and $C_{k,i}$ are the unknown surface currents and defined complex excitation coefficients of array elements $k$ and $i$, respectively. The term $\alpha_{ki}$ is the complex weighting constant that is used to express the relationship between the currents and will be referred to as the $\alpha$ weight coefficients. The vectors $\vec{r}$ and $\vec{r}_{ki}$ are shown in Figure 3.4.

By using (3.15) and following the same approach as explained in Section 3.2.2, an integral equation can now be written for array element $i$ as follows,

$$-\hat{n}\times\mathbf{E}_i^i(\vec{r}) \;=\; \hat{n}\times\int_{\text{el. }i}\Big\{\sum_{k=1}^{M}\alpha_{ki}\overline{\overline{\mathbf{G}}}(\vec{r}\mid\vec{r}\,'+\vec{r_{ki}})\Big\}\cdot\mathbf{J}_i(\vec{r}\,')ds', \tag{3.16}$$

where $\mathbf{E}_i^i(\vec{r})$ is the electric field incident on array element $i$. Equation (3.16) can be solved with the MoM in which case the analysis is limited to array element $i$ only. The resulting MoM matrix equation that is to be solved is of dimension $N_i \times N_i$, where $N_i$ is the number of basis functions associated with element $i$. The term in brackets in (3.16) is the spatial representation of the spectral finite array Green's function defined in (3.14) and can be written as follows,

$$\overline{\overline{\mathbf{\Gamma}}}_i(\vec{r}\mid\vec{r}\,') = \sum_{k=1}^{M}\alpha_{ki}\overline{\overline{\mathbf{G}}}(\vec{r}\mid\vec{r}\,'+\vec{r_{ki}}) \tag{3.17}$$

Likewise, a *different* finite array Green's function can also be setup for array elemnent $j$ as

follows,

$$\overline{\overline{\mathbf{\Gamma}}}_j(\vec{r} \mid \vec{r}') = \sum_{k=1}^{M} \alpha_{kj} \overline{\overline{\mathbf{G}}}(\vec{r} \mid \vec{r}' + r\vec{_{kj}}). \tag{3.18}$$

It is clear by comparing (3.17) and (3.18) that $\overline{\overline{\mathbf{\Gamma}}}_i \neq \overline{\overline{\mathbf{\Gamma}}}_j$ (as also noted in Section 3.2.2). An integral equation with a *different* Green's function, $\overline{\overline{\mathbf{\Gamma}}}_m$, has to be formulated and solved for each of the array elements $m = 1, \ldots, M$, yielding as a result a different current distribution on each.

The method discussed here is a *perturbation* approach, i.e. where the infinite array approximation in (3.15) is perturbed through the use of the finite array Green's function defined in (3.17). The perturbation is valid if the final current distribution on the array is not too different from the initial approximation made in (3.15). The approach outlined here is therefore valid for cases where the current distributions between array elements are slowly varying. Typical examples for which this method will work well are finite antenna arrays consisting of elements that exhibit low mutual coupling, or for many printed array applications where the type of element used is close to resonance [45]. In Chapter 5 a method will be presented through which this limitation can be mitigated by applying Macro Basis Functions (MBFs) - a topic that is discussed in the following subsection.

### 3.3.2 Macro Basis Function methods

A range of macro basis function approaches exist, viz., the Characteristic Basis Function Methods (CBFM) [28], the Synthetic Function Expansion (SFx) [32] and the subdomain multilevel approach (SMA) [89]. In addition, macro basis functions derived from generalised eigenvalue solutions for the array elements have also been formulated [90–92]. In this Section, the CBFM will be reviewed in more detail. The discussion is primarily based on that presented in [28].

Consider again the finite array in Figure 3.4 consisting of $M$ identical, disjoint elements each of which is discretised with $N_i$ subsectional basis functions. In accordance with the domain decomposition paradigms the MoM matrix equation defined in Section 2.4, i.e. $\mathbf{ZJ} = \mathbf{V}$, can be block-partitioned as,

$$\begin{bmatrix} \mathbf{Z}_{11} & \mathbf{Z}_{12} & \cdots & \mathbf{Z}_{1M} \\ \mathbf{Z}_{21} & \mathbf{Z}_{22} & \cdots & \mathbf{Z}_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{Z}_{M1} & \cdots & \cdots & \mathbf{Z}_{MM} \end{bmatrix} \begin{bmatrix} \mathbf{J}_1 \\ \mathbf{J}_2 \\ \vdots \\ \mathbf{J}_M \end{bmatrix} = \begin{bmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \\ \vdots \\ \mathbf{V}_M \end{bmatrix}, \tag{3.19}$$

where $\mathbf{Z}_{pp}$ is the self-interaction matrix associated with array element $p$ and $\mathbf{Z}_{pq}$ represents the mutual coupling matrix between subdomains[2] $p$ and $q$ respectively. The purpose of MoM-based domain decomposition schemes such as the CBFM is to avoid the high computational costs associated with solving (3.19), which scale as $\mathcal{O}((M \times N_i)^3)$ for the solution and $\mathcal{O}((M \times N_i)^2)$ for memory usage should a direct solution approach be used.

[2]In the remainder of the thesis, a subdomain is synonymous for an array element.

The core idea of the CBFM is that the current on the $p$th subdomain be expanded in terms of primary and secondary characteristic basis functions (CBFs), each weighted with an unknown complex constant, $\beta_n$, as follows,

$$\mathbf{J}_p \simeq \beta_{0p} \underbrace{\mathbf{J}_{0p}}_{\text{Prim. CBF}} - \sum_{m=1, m\neq p}^{M} \beta_{pm} \underbrace{\mathbf{Z}_{pp}^{-1}\mathbf{Z}_{pm}\mathbf{J}_{0m}}_{\text{Sec. CBFs}}, \tag{3.20}$$

where $\mathbf{J}_{0p} = (\mathbf{Z}_{pp})^{-1}\mathbf{V}_p$ is the primary CBF, i.e., the induced current on the $p$th subdomain in isolation. The other terms in (3.20) represent the secondary CBFs. Strictly speaking, the vector $\mathbf{J}_p$ is the expansion coefficients for the basis functions on element $p$ that represent the discretised surface current distribution, $\mathbf{J}_p(\vec{r})$, as discussed in Section 2.4. Nonetheless, the vector $\mathbf{J}_p$ will be referred to as the current distribution on domain $p$ in this context.

By substituting (3.20) into (3.19), i.e. the partitioned MoM matrix equation, we are left with an $N_i \times M^2$ matrix equation. This equation can be transformed to a $M^2 \times M^2$ matrix equation by weighting each of the entries with a CBF (i.e. similar to the weighting method explained for the RWG basis functions in Section 2.3). This *reduced* impedance matrix equation for the CBFM can be written as follows,

$$\mathbf{Z}^{\text{CBFM}}\mathbf{I}^{\text{CBFM}} = \mathbf{V}^{\text{CBFM}}, \tag{3.21}$$

where $\mathbf{Z}^{\text{CBFM}}$ contains the submatrices, $\mathbf{Z}_{pq}^{\text{CBF}}$, that quantifies the reactions between the CBFs on subdomains $p$ and $q$, respectively. These reaction terms may be written as follows,

$$\mathbf{Z}_{pq}^{\text{CBF}} = \langle \mathbf{J}_p^T, \mathbf{Z}_{pq}^{\text{RWG}}\mathbf{J}_q \rangle, \tag{3.22}$$

where $T$ denotes the transposition operator and $\langle \cdot, \cdot \rangle$ the symmetric product between vectors and/or matrices. Both $\mathbf{J}_p$ and $\mathbf{J}_q$ are column-augmented matrices containing the CBFs of domains $p$ and $q$, respectively. In (3.20), the vector, $\mathbf{I}^{\text{CBFM}}$, contains the unknown $\beta_n$ coefficients introduced in (3.20) and $\mathbf{V}^{\text{CBFM}}$ is the excitation vector that is defined as follows,

$$\mathbf{V}_p^{\text{CBF}} = \langle \mathbf{J}_p^T, \mathbf{V}_p^{\text{RWG}} \rangle, \tag{3.23}$$

where $\mathbf{V}_p^{\text{RWG}}$ is the excitation associated with subdomain $p$ in (3.18).

Equation (3.21) can be solved using LU-factorisation. This is beneficial when considering multiple excitations, in that the factorisation is only done once, followed by a number of backward substitutions for each excitation, e.g. for phased array simulations.

Various extensions for the CBFM have also been introduced since its basic formulation in [28], that includes an approach to deal with interconnected elements [29], the use of tertiary basis functions to improve the accuracy of the method [93] and the application of acceleration strategies, such as the ACA algorithm[30], to speed up the matrix-vector product in (3.22). The CBFM is also highly parallelisable using distributed programming paradigms as shown by Ludick and Davidson in [42].

In addition it should be pointed out that the number of CBFs on a domain can be reduced

by performing an orthonormalisation, e.g. using a Gram-Schmidt scheme as noted in [28]. This reduces the dimension of (3.21) significantly and in turn also the memory usage and runtime associated with solving the CBFM reduced matrix equation. This orthonormalisation was not applied to the CBFM implementations that are used in Section 5.4 and Section 6.1.2.

## 3.4 Conclusion

An overview of array analysis techniques was presented in this Chapter. The methods that were considered include basic pattern multiplication strategies as well as infinite array solution approaches and methods suitable for the analysis of finite irregular (and regular) arrays. A first order accuracy improvement of the infinite array solution, obtained by introducing a window function, was also discussed. Other approaches suitable for finite array analysis were also considered, including a spatial domain formulation of the infinite array plus windowing method. This method is a perturbation approach that is applicable to the analysis of structures where the current distribution varies slowly between the array elements. This method also forms the basis for the DGFM approach presented in the following Chapter. In addition, Macro Basis Functions were also reviewed at the hand of the CBFM and forms part of various discussions to follow in the next Chapter, as well as the formulation of the i-DGFM in Chapter 5.

# 4

# The Domain Green's Function Method (DGFM)

The technique presented in this Chapter for the analysis of disjoint finite antenna arrays is the Domain Green's Function Method (DGFM) [50–53]. The DGFM is a perturbation approach that is fundamentally based on the work done by Skrivervik and Mosig which was developed for printed array antennas using the multi-layered Green's function [44, 45] as presented in Section 3.2.2 and Section 3.3.1. In the current work, the method is formulated on a higher block-matrix factorisation level. From an implementation point of view the DGFM can therefore be applied to a variety of MoM problems, using e.g. conventional RWG or HOBF basis functions, or different Green's functions, such as the free-space Green's function or the multi-layered planar Green's function for stratified media.

Mutual coupling between array elements is accounted for in the DGFM with the formulation of an active impedance matrix for each of the domains/array elements. The calculation of the active impedance matrix dominates the total solution time for very large array configurations. To overcome this limitation various acceleration strategies is proposed to reduce this computational cost. One of the strategies include sorting the coupling terms in the active impedance summation according to descending mutual coupling strength in the active array environment. The sorting criteria that is used, accounts for both the complex excitation and the distance between the domains. The aforementioned is then used to truncate the summation of the active impedance matrix calculation to include only a certain number of terms [53]. In addition, the DGFM has also been combined with the ACA algorithm to accelerate the calculation of the coupling matrix terms by approximating them as the product of low-rank rectangular matrices [57]. A hybrid distributed/shared memory parallelisation approach has also been applied to the DGFM implementation to benefit from multi-core computing environments. The DGFM is also integrated as part of an efficient array analysis tool in the commercial computational electromagnetics software package, FEKO [6].

The Chapter is structured as follows: in Section 4.1 the formulation for the DGFM is presented followed by a comparison between the DGFM and the CBFM in Section 4.2. Various acceleration strategies for calculating the active impedance matrices associated with the DGFM

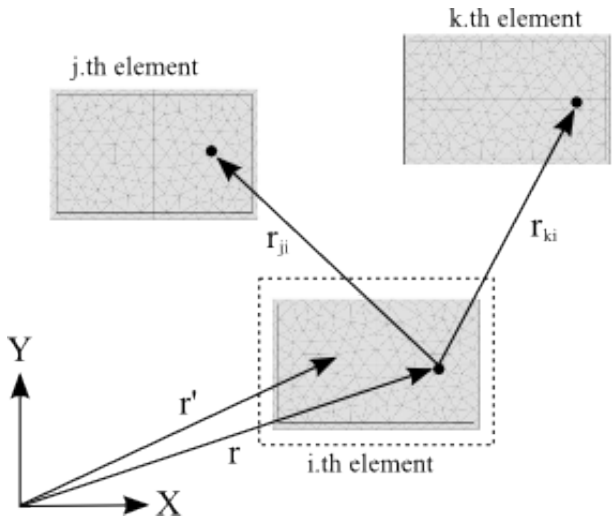



**Figure 4.1:** Antenna array consisting of $M$ elements with $N_i$ basis functions each, showing the various observation points as used by the DGFM

is discussed in Section 4.3. The parallelisation of the DGFM is explained in Section 4.4 and in Section 4.5 various test cases are presented to illustrate the computational performance and numerical accuracy of the method. The Chapter is then concluded in Section 4.6.

## 4.1 Formulation

Consider the array configuration shown in Figure 4.1 that consists of $M$ identical array elements. This figure is identical to Figure 3.4 illustrated in Section 3.3.1, except that the observation points are varied as opposed to the source locations. In Figure 4.1 the observation positions are located at $\vec{r}$, $\vec{r}+\vec{r}_{ji}$ and $\vec{r}+\vec{r}_{ki}$, with $\vec{r}_{ji}$ and $\vec{r}_{ki}$ the position vectors of array elements $j$ and $k$ relative to element $i$, respectively. Similar to the formulation for the finite array Green's function method discussed in Section 3.3.1, or infinite array plus windowing approach presented in Section 3.2.2, the first step in the DGFM is to apply the *infinite array assumption*, i.e.,

$$\mathbf{J}_k(\vec{r}+\vec{r}_{ki}) = \frac{C_k}{C_i}\mathbf{J}_i(\vec{r}) = \alpha_{ki}\mathbf{J}_i(\vec{r}). \tag{4.1}$$

In (4.1) the unknown surface currents on all the array elements are accepted as identical to the surface current on the element that we wish to analyse (element $i$ in this case), scaled by the $\alpha$ weighting coefficients defined in Section 3.3.1. The physical implication of (4.1) is that, initially, the current on each of the elements is assumed to be of equal spatial distribution. This assumption is then perturbed through the addition of mutual coupling from the surrounding finite array environment. As discussed in Section 3.3.1 and [45], for the final result to be considered valid, the current distributions that were calculated need to be smoothly varying. Otherwise stated, the final answer cannot deviate significantly from the original assumption made in (4.1). A method will be presented in Chapter 5 through which this limitation can be mitigated by applying MBFs.

Furthermore, consider Green's functions for which the following holds,

$$\overline{\overline{\mathbf{G}}}(\vec{r},\vec{r}\,'+\vec{r}_{ki})=\overline{\overline{\mathbf{G}}}(\vec{r}-\vec{r}_{ki},\vec{r}\,'). \tag{4.2}$$

Examples of Green's functions that exhibit the above property, include the free-space Green's function (defined in (2.5)) as well as the multi-layered Green's function [94]. Equation (4.2) implies that the domains are identically discretised and allows for an implementation based on working with matrix elements and not with a Green's function summation as was done for the method discussed in Section 3.3.1. This provides for a simpler implementation, the details of which will be explained next.

Using the assumption made in (4.1) with (4.2) we can write an EFIE for array element $i$ as follows,

$$-\hat{n}\times\mathbf{E}_i^i(\vec{r}) \;=\; \hat{n}\times\sum_{k=1}^{M}\Big\{\alpha_{ki}\int_{\text{el. }i}\overline{\overline{\mathbf{G}}}(\vec{r}-\vec{r_{ki}}\mid\vec{r}\,')\cdot\mathbf{J}_i(\vec{r}\,')ds'\Big\}, \tag{4.3}$$

which is equivalent to (3.16) with the order of summation and integration interchanged by applying (4.2).

Solving (4.3) with the MoM is localised to the $N_i$ unknowns associated with each of the $M$ elements being analysed. The runtime therefore scales as $\mathcal{O}(M\times N_i^3)$ as opposed to $\mathcal{O}((M\times N_i)^3)$ should the MoM be used for the entire structure. Likewise, the memory usage scales according to $\mathcal{O}(N_i^2)$ and not $\mathcal{O}((M\times N_i)^2)$. Equation (4.3) is solved for each array element in the active array environment, i.e. $M$ times.

The formulation explained above can alternatively be expressed in terms of a block matrix factorisation of the MoM matrix equation. As discussed in Section 3.3.2, the MoM matrix equation, $\mathbf{ZJ}=\mathbf{V}$, for the geometry shown in Figure 4.1 can be block-partitioned as,

$$\begin{bmatrix}\mathbf{Z}_{11}&\mathbf{Z}_{12}&\cdots&\mathbf{Z}_{1M}\\ \mathbf{Z}_{21}&\mathbf{Z}_{22}&\cdots&\mathbf{Z}_{2M}\\ \vdots&\ddots&\cdots&\vdots\\ \mathbf{Z}_{M1}&\cdots&\cdots&\mathbf{Z}_{MM}\end{bmatrix}\begin{bmatrix}\mathbf{J}_1\\ \mathbf{J}_2\\ \vdots\\ \mathbf{J}_M\end{bmatrix}=\begin{bmatrix}\mathbf{V}_1\\ \mathbf{V}_2\\ \vdots\\ \mathbf{V}_M\end{bmatrix}, \tag{4.4}$$

where $\mathbf{Z}_{pp}$ is the self interaction matrix associated with domain $p$ that is formulated for the $N_i$ basis functions. The matrix $\mathbf{Z}_{pq}$ represents the mutual coupling between domains $p$ and $q$. The column vectors $\mathbf{J}_p$ and $\mathbf{V}_p$ are the unknown expansion coefficients to be calculated and the known excitation vector associated with the $p$th domain.

By using (4.1), we can calculate the total active current distribution, i.e., the solution to (4.4), by solving smaller block matrices. To illustrate this point, consider that by applying (4.1), the currents on domains $2,\ldots,M$ may be expressed in terms of the current on domain $p$ as,

$$\mathbf{J}_2=\alpha_{2p}\mathbf{J}_p;\quad \mathbf{J}_q=\alpha_{qp}\mathbf{J}_p;\quad \mathbf{J}_M=\alpha_{Mp}\mathbf{J}_p. \tag{4.5}$$

By substituting (4.5) in (4.4), we can then solve for the active current distribution on the $p$th

domain, i.e.,

$$\begin{aligned}\mathbf{V}_p &= [\mathbf{Z}_{pp} + (\alpha_{qp}\mathbf{Z}_{pq} + \ldots + \alpha_{Mp}\mathbf{Z}_{pM})]\mathbf{J}_p \\ &= \left[\sum_{m=1}^{M} \alpha_{mp}\mathbf{Z}_{pm}\right]\mathbf{J}_p = \mathbf{Z}_p^{\text{act}}\mathbf{J}_p,\end{aligned} \tag{4.6}$$

where the excitation-dependent $\mathbf{Z}_p^{\text{act}}$ is called the *active impedance matrix* for array element $p$, that accounts for both the self-coupling as well as the mutual coupling from the surrounding active array environment.

On a similar basis, we can calculate the active impedance matrices for each of the other array elements. In general, the active impedance matrices of the domains are different, i.e., $\mathbf{Z}_p^{\text{act}} \neq \mathbf{Z}_q^{\text{act}}$ for $p \neq q$. The resulting current distribution on the array elements $p$ and $q$ are therefore perturbed in the sense that $\mathbf{J}_q \neq \alpha_{qp}\mathbf{J}_p$ for $p \neq q$ as per our initial assumption in (4.5). Formulating the DGFM based on the addition of weighted submatrices, and not on the summation of the Green's function as done in [44, 45], makes the approach essentially independent of the subsectional basis functions or Green's function used in the MoM formulation (provided the requirement in (4.2) is satisfied).

It should be noted that the DGFM technique presented here, i.e. by using $\alpha$ weight coefficient, does not lead to accurate results when applied to a passive array, e.g. for the calculation of embedded element patterns (EEPs) as will be shown in Chapter 5. The reason for this, is that passive elements are not excited, i.e., $C_k = 0$ in (4.1), that leads to zero $\alpha$ weighting coefficients. Hence, and in contrast to an all-excited array, the effect of mutual coupling due to a local excitation is not accounted for. The current on passive elements are assumed to be zero. The same limitation applies to calculating the scattering matrix for the array. Chapter 5 discusses a method for correcting this limitation so that first-order coupling effects are also accounted for whenever spatially localised solutions for the current are to be expected. The aforementioned ensures that $C_k \neq 0$ even for the case where only one element is excited. Furthermore, Chapter 5 discusses a method where individual subsectional basis functions are scaled by using an $\boldsymbol{\alpha}$ weight matrix, as an alternative to the $\alpha$ weight coefficients defined in (4.1). In the following section, a comparison between the DGFM and CBFM, as done in [53, 54], will explain how the DGFM accounts for mutual coupling in the active array environment, and how this is different from the approach followed by the CBFM.

## 4.2 Comparing the DGFM with the CBFM

As noted in Section 3.3.2, the basic premise behind the CBFM is that the total unknown current distribution on a domain can be expressed as a linear combination of primary and secondary CBFs[1], each weighted with an unknown complex constant, $\beta_n$ [28]. To illustrate this concept, consider again the current on domain $p$ (for a disjoint array such as that shown in Figure 4.1),

[1] In [93] tertiary CBFs were also used to model a connected patch array.

which was expressed in (3.20) as follows,

$$\mathbf{J}_p \simeq \beta_{0p} \underbrace{\mathbf{J}_{0p}}_{\text{Prim. CBF}} - \sum_{m=1,m\neq p}^{M} \beta_{pm} \underbrace{\mathbf{Z}_{pp}^{-1}\mathbf{Z}_{pm}\mathbf{J}_{0m}}_{\text{Sec. CBFs}}. \tag{4.7}$$

The first term in (4.7) is the primary CBF and is calculated as the current on the subdomain in isolation, i.e. $\mathbf{J}_{0p} = (\mathbf{Z}_{pp})^{-1}\mathbf{V}_p$. The secondary CBFs, i.e., the terms $m = 2, \ldots, M$, represents the current induced on subdomain $p$, by using as excitation the primary CBFs of the surrounding elements. The $\alpha_{qp}$ weight coefficients for the DGFM can be expressed in terms of the primary CBFs, as well as the ratio of the port excitations of domains $p$ and $q$, as follows,

$$\alpha_{qp} = \frac{V_q}{V_p} = \frac{\langle \mathbf{\Lambda}, \mathbf{J}_{0q} \rangle}{\langle \mathbf{\Lambda}, \mathbf{J}_{0p} \rangle} \tag{4.8}$$

In (4.8), $\mathbf{\Lambda}$ is a testing vector (e.g., a vector filled by ones) and $V_p$ and $V_q$ is the applied voltage excitation coefficients for domains $p$ and $q$, respectively[2]. Furthermore, it can be noted that (4.8) is valid since $V_p\mathbf{J}_{0p} = V_q\mathbf{J}_{0q}$ on account of (4.1). The ratio between the primary CBFs is therefore identical to the ratio of the applied voltage excitation coefficients of the subdomains $p$ and $q$. The first difference between the methods is that the DGFM does not require the initial calculation of a set of primary CBFs, which is of $\mathcal{O}(M \times N_i^3)$ and increases to $\mathcal{O}(M \times N_i^3 + (M \times N_i)^2)$ when adding also secondary CBFs, as discussed in [95].

The next difference between the CBFM and the DGFM is clear when comparing the $M^2 \times M^2$ reduced impedance matrix calculation for the CBFM as discussed in Section 3.3.2. Each of the CBFM reduced impedance matrix entries in (3.21) represents the interaction between the CBFs generated on each of the $M$ domains through (3.22) as follows,

$$\mathbf{Z}_{pq}^{\text{CBF}} = \langle \mathbf{J}_{pm}^T, \mathbf{Z}_{pq}^{\text{RWG}}\mathbf{J}_{qn} \rangle. \tag{4.9}$$

Mutual coupling is modelled accurately for the CBFM with (4.9), as the interaction between CBFs are calculated using the full-wave coupling matrix, $\mathbf{Z}_{pq}^{\text{RWG}}$. For the DGFM, mutual coupling between the domains is approximated by weighting $\mathbf{Z}_{pq}^{\text{RWG}}$ with the known $\alpha_{qp}$ coefficient. In terms of accuracy, the DGFM does therefore not account for spatially localised first-order coupling effects, as in the CBFM; however, it still accounts for spatially extended global coupling effects on account of the infinite array assumption, unlike the CBFM. Another difference that can be noted, is that the CBFM allows for better treatment of multiple excitations, since the active impedance matrices in (4.6) have to be recalculated for every array excitation.

The benefit to using the DGFM is clear when comparing the computational costs, in terms of runtime and memory usage, with that of the CBFM. As already mentioned, the DGFM does not require the initial calculation of a set of primary and secondary CBFs. Another difference is the cost associated with calculating the active impedance matrices compared to that of setting up the reduced matrix equation. For the CBFM, calculating (4.9) for each of the domains $p = 1, \ldots, M$ and $q = 1, \ldots, M$ scales as $\mathcal{O}(M^4 \times N_i^2)$. The cost of calculating the active

[2]The applied excitation coefficient, $V_p$ or $V_q$, is equivalent to $C_k$ or $C_i$ in (4.1).

**Table 4.1:** Order of CPU-time and memory usage for certain phases of the CBFM and DGFM when applied to the array geometry of Figure 4.1, consisting of $M$ disjoint elements each with $N_i$ subsectional basis functions.

| **Method:** | CBFM | DGFM |
|---|---|---|
| MoM matrix setup time | $\mathcal{O}((M \times N_i)^2)$ | $\mathcal{O}((M \times N_i)^2)$ |
| Calculation of Primary basis | $\mathcal{O}(M \times N_i^3)$ | $NA$ |
| Calculation of Secondary basis | $\mathcal{O}((M \times N_i)^2)$ | $NA$ |
| Calculation of reduced matrix | $\mathcal{O}(M^4 \times N_i^2)$ | $NA$ |
| Calculation of active impedance matrices | $NA$ | $\mathcal{O}(M^2 \times N_i^2)$ |
| Memory usage (whole solution) | $\mathcal{O}(2N_i^2 + M^4)$ | $\mathcal{O}(N_i^2)$ |

impedance matrix equations in (4.6), scales as $\mathcal{O}(M^2 \times N_i^2)$, i.e., a saving of $\mathcal{O}(M^2)$.

It is however to be noted that the effect of applying techniques such as the ACA [30], that accelerates the matrix vector product in (4.9), by approximating $\mathbf{Z}_{pq}^{\mathrm{RWG}}$ as a low-rank decomposition, $\tilde{\mathbf{Z}}_{pq}^{\mathrm{RWG}}$, has not been considered. In [96], the cost of calculating the reduced impedance matrix has also been addressed by using a low-order harmonic-polynomial function as a compact representation for the reduced matrix entries in (4.9). This harmonic polynomial expression is obtained by precomputing the reaction integrals in a limited set of relative positions between domains.

In addition, $M \times$ CBFs per domain are typically not used, as the number of CBFs per domain can be reduced by applying a Gram-Schmidt orthogonalisation [28], or a Singular Value Decomposition (SVD) [30] to retain only a limited number of orthonormal CBFs. If we consider on average $L \times$ CBFs per domain, with $L < M$, then the cost for calculating the reduced impedance matrix in (4.9) reduces from $\mathcal{O}(M^4 \times N_i^2)$ to $\mathcal{O}(M^2 \times L^2 \times N_i^2)$. It should however be noted that the cost of applying the Gram-Schmidt orthogonalisation, or the SVD, to reduce the number of CBFs per domain adds additional complexity to the CBFM algorithm as explained in [95].

The CBFM and the DGFM also differ in terms of memory usage. The memory requirement for calculating the primary CBFs scales as $\mathcal{O}(N_i^2)$, as we need to store the LU factorisation of the self-interaction matrix of each domain that is of size $N_i \times N_i$. Another $\mathcal{O}(N_i^2)$ is required for storing the coupling matrices, $\mathbf{Z}_{pq}^{\mathrm{RWG}}$, used when generating the secondary CBFs in (4.7) and also when calculating the reduced matrix entries in (4.9). In addition to the aforementioned, storing the CBFM reduced impedance matrix scales as $\mathcal{O}(M^4)$. The memory requirement for the DGFM scales only as $\mathcal{O}(N_i^2)$ for storing the active impedance matrix of each domain. The computational complexity of the DGFM and CBFM formulation, as presented in [53, 54], is summarised in Table 4.1.

The computational complexities of the CBFM and DGFM algorithms have been qualitatively

compared in this section, and their relative accuracies have been examined. It has been shown that although the DGFM is less accurate than MBF techniques – such as the CBFM for cases where higher order coupling is required – one gains a saving in both memory usage and runtime for cases where it can be used. Finally, it is worth noting that the DGFM can be enhanced by the CBFM for the analysis of large disjoint arrays consisting of subarray tiles, as will be explained in Chapter 6.

## 4.3 Accelerating the active impedance matrix calculation

Calculating the MoM impedance matrix for the $M$ element array in Figure 4.1 scales as $\mathcal{O}((M \times N_i)^2)$ as noted in Section 4.2, and can rapidly become the dominant part in the solution phase for the DGFM, especially for large arrays. In the following subsections, three methods will be considered that accelerate the calculation of these submatrices that are used in the active impedance matrix calculation. These methods include the use of translational symmetry, a manual sorting and truncation approach for the summation involved in the calculation of $\mathbf{Z}_p^{\text{act}}$, as well as the use of the ACA algorithm to calculate low-rank representations for the coupling submatrices.

### 4.3.1 Applying translational symmetry

If an array exhibits a large degree of periodicity, then translational symmetry can be used during the active impedance matrix calculation. To illustrate this concept, consider the periodic $5 \times 1$ array with element numbering depicted as in Figure 4.2. In this case, the active impedance matrices for elements $1, \ldots, 5$ can be written as[3],

$$\begin{aligned}
\mathbf{Z}_1^{\text{act}} &= \mathbf{Z}_{11} + \alpha_{21}\mathbf{Z}_{12} + \alpha_{31}\mathbf{Z}_{13} + \alpha_{41}\mathbf{Z}_{14} + \alpha_{51}\mathbf{Z}_{15} \\
\mathbf{Z}_2^{\text{act}} &= \mathbf{Z}_{22} + \alpha_{12}\mathbf{Z}_{21} + \alpha_{32}\mathbf{Z}_{23} + \alpha_{42}\mathbf{Z}_{24} + \alpha_{52}\mathbf{Z}_{25} \\
\mathbf{Z}_3^{\text{act}} &= \mathbf{Z}_{33} + \alpha_{13}\mathbf{Z}_{31} + \alpha_{23}\mathbf{Z}_{32} + \alpha_{43}\mathbf{Z}_{34} + \alpha_{53}\mathbf{Z}_{35} \\
\mathbf{Z}_4^{\text{act}} &= \mathbf{Z}_{44} + \alpha_{14}\mathbf{Z}_{41} + \alpha_{24}\mathbf{Z}_{42} + \alpha_{34}\mathbf{Z}_{43} + \alpha_{54}\mathbf{Z}_{45} \\
\mathbf{Z}_5^{\text{act}} &= \mathbf{Z}_{55} + \alpha_{15}\mathbf{Z}_{51} + \alpha_{25}\mathbf{Z}_{52} + \alpha_{35}\mathbf{Z}_{53} + \alpha_{45}\mathbf{Z}_{54}
\end{aligned} \tag{4.10}$$

[3]Here the $\alpha$ weighting coefficients have been used. The derivation is also valid for the $\boldsymbol{\alpha}$ weight matrix that will be presented in Chapter 5.

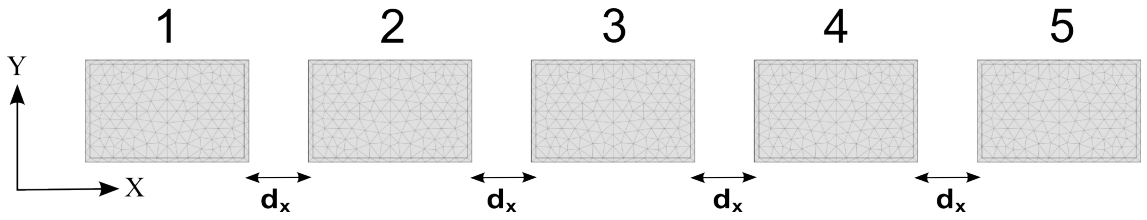



**Figure 4.2:** Linear array consisting of 5 elements

The following translational symmetry relations are valid for the array in Figure 4.2 with the active impedance matrices in (4.10),

$$\begin{aligned} \mathbf{Z}_{11} &= \mathbf{Z}_{22} = \mathbf{Z}_{33} = \mathbf{Z}_{44} = \mathbf{Z}_{55} \\ \mathbf{Z}_{12} &= \mathbf{Z}_{23} = \mathbf{Z}_{34} = \mathbf{Z}_{45} \\ \mathbf{Z}_{13} &= \mathbf{Z}_{24} = \mathbf{Z}_{35} \\ \mathbf{Z}_{14} &= \mathbf{Z}_{25} \end{aligned} \tag{4.11}$$

The first symmetry relation in (4.11) stems from the fact that the domains are identically discretised. The second may be written more generally as $\mathbf{Z}_{pq} = \mathbf{Z}_{p+1,q+1}$.

From (4.11) it is evident that we only need to calculate and store the first row of the partitioned MoM matrix in (4.4). When the coupling terms are required in the setup of the active impedances for the subsequent array elements, then these submatrices can be read from memory. Not considered here is the effect of rotational symmetry, i.e., where we have $\mathbf{Z}_{pq} = \mathbf{Z}_{qp}$. Similar relations may be derived for this case as well. Also, the above has been illustrated for a simple linear array case, but we can also extract the translational symmetry relationships for two-dimensional and circular arrays.

### 4.3.2 Truncating the active impedance matrix summation

For aperiodic arrays, such as that depicted in Figure 4.1, translational symmetry cannot be used. An approach that may be followed for this case, is to truncate the summation that is present in the calculation of the active impedance matrices. This requires that the active impedance matrix terms in (4.6) be sorted beforehand.

Following (4.6), the active impedance matrix of the $i$th domain may be expressed as follows,

$$\begin{aligned} \mathbf{Z}_i^{\text{act}} &= \mathbf{Z}_{ii} + \alpha_{1i}\mathbf{Z}_{i1} + \alpha_{ki}\mathbf{Z}_{ik} + \ldots + \alpha_{Mi}\mathbf{Z}_{iM} \\ &= \sum_{k=1}^{M} \alpha_{ki}\mathbf{Z}_{ik}, \end{aligned} \tag{4.12}$$

with $\alpha_{ki}$ as defined in (4.5). In order to sort the terms in (4.12), the following scalar quantity is defined for each element

$$\zeta^i = |\alpha_{ki}|/|\vec{r}_{ki}|. \tag{4.13}$$

Equation (4.13) accounts for both the relative excitation magnitude as well as the physical separation between the elements $k$ and $i$, respectively. The distance between the elements $\vec{r}_{ki}$, illustrated in Figure 4.1, assumes that the elements are not rotated arbitrarily with respect to each other. This sorting is done prior to the analysis of each element and, for optimal performance, the Quicksort algorithm is used [97].

If (4.12) is sorted according to (4.13), then this would represent a monotonic decreasing function and the number of terms used in the summation can be truncated to some value, $M_c$, that defines the radius of convergence for each of the elements, i.e., $M_c = M \times R_c$, with $R_c$ a

factor between 0 and 1. The active impedance matrix is then approximated as follows,

$$\begin{aligned}\mathbf{Z}_i^{\text{act}} &= \underbrace{\mathbf{Z}_{ii} + \alpha_{1i}\mathbf{Z}_{i1} + \alpha_{ki}\mathbf{Z}_{ik}}_{M_c \text{ terms}} + \ldots + \alpha_{Mi}\mathbf{Z}_{iM} \\ &\simeq \sum_{k=1}^{M_c} \alpha_{ki}\mathbf{Z}_{ik}. \end{aligned} \tag{4.14}$$

The complexity of the matrix-fill phase can therefore be reduced to $\mathcal{O}((R_c \times M \times N_i)^2 + M \times S)$, where $R_c$ is the convergence radius between 0 and 1, and $S$ is a small factor related to the sorting of the DGF terms, using (4.13).

Numerical results will be illustrated in Section 4.5, that will discuss the effect of truncating the active impedance matrix summation.

### 4.3.3 Applying the ACA algorithm

A drawback with the method discussed in Section 4.3.2, is that the value of $M_c$ is not known beforehand, and typically has to be estimated numerically, i.e. by simulating a few examples with values of $M_c$ staring from 0 to 100%. Another drawback is that in some cases the mutual coupling in dense regular array layouts can potentially combine in such a way so that (4.12) increase as more terms are added. For these cases, the truncation method will not produce accurate results.

A better and more generic approach would be to obtain a fast way to calculate the whole summation in 4.12. To this end, the focus is directed towards the ACA algorithm [14–18] that has advantageously been combined with MBF techniques such as the CBFM [30] to accelerate the matrix-vector product in 4.9. This is done by approximating $\mathbf{Z}_{ik}$ as a low-rank decomposition, $\tilde{\mathbf{Z}}_{ik}$. Likewise, the ACA algorithm can be used to accelerate the DGFM (or i-DGFM presented in Chapter 5 ) by applying it to (4.12), as follows,

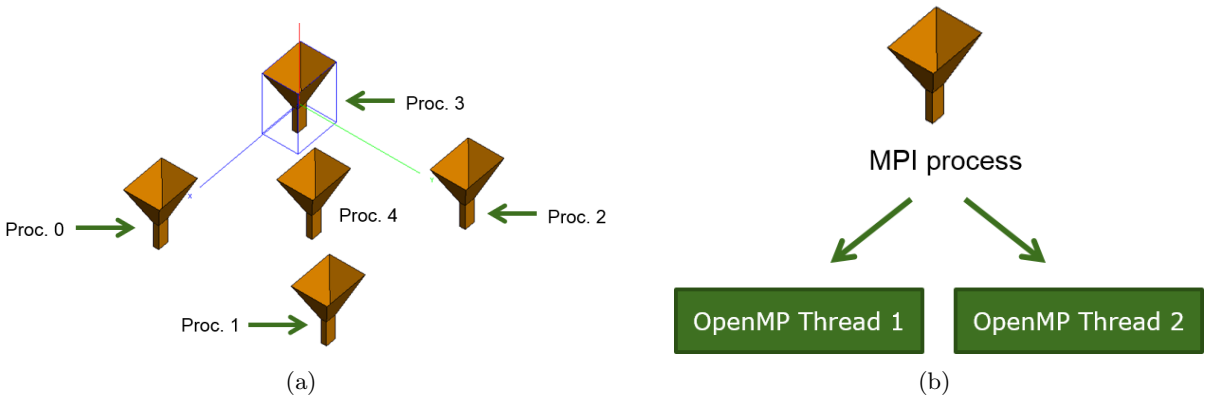
$$\begin{aligned}\mathbf{Z}_i^{\text{act}} &\simeq \mathbf{Z}_{ii} + \underbrace{\alpha_{1i}\tilde{\mathbf{Z}}_{i1} + \alpha_{ki}\tilde{\mathbf{Z}}_{ik} + \ldots + \alpha_{Mi}\tilde{\mathbf{Z}}_{iM}}_{\text{ACA}} \\ &\simeq \sum_{k=1}^{M} \alpha_{ki}\tilde{\mathbf{Z}}_{ik}. \end{aligned} \tag{4.15}$$

The fist term in (4.15), i.e. the self coupling term, is excluded from the approximation as these submatrices are diagonally dominant and seldom rank deficient [29]. For these terms, the application of the ACA algorithm will therefore introduce an unnecessary computational overhead.

The low-rank matrices, $\tilde{\mathbf{Z}}_{ik}$ are written as the product of two rectangular matrices as follows,

$$\tilde{\mathbf{Z}}_{ik} = \mathbf{U}_i^{N_i \times r}\mathbf{V}_k^{r \times N_i}, \tag{4.16}$$

where r is the effective rank of matrix $\tilde{\mathbf{Z}}_{ik}$ and $\mathbf{U}_i^{N_i \times r}$ is a column-augmented matrix of size $N_i \times r$ and $\mathbf{V}_k^{r \times N_i}$ is a row-augmented matrix of size $r \times N_i$. Furthermore, it is assumed that each of




(a) (b)

**Figure 4.3:** An illustration of how hybrid distributed/shared memory programming paradigm is applied for the DGFM with (a) an example of how array elements are distributed amongst processes for MPI; and (b) OpenMP threading applied to an array element.

the domains/array elements has $N_i$ basis functions. This low-rank decomposition in (4.16) can be calculated using the partially pivoted ACA algorithm [14] by following the implementation explained in [17] that is summarised in Appendix A.

Adding the weighted terms $2, \ldots, M$ in (4.15) can be done by concatenating the weighted columns of the $\mathbf{U}$ matrices, and the rows of the $\mathbf{V}$ matrices, as follows,

$$\alpha_{1i}\tilde{\mathbf{Z}}_{i1}+\alpha_{ki}\tilde{\mathbf{Z}}_{ik}+\ldots+\alpha_{Mi}\tilde{\mathbf{Z}}_{iM} = \left[\alpha_{1i}\mathbf{U}_i \quad \alpha_{ki}\mathbf{U}_k \quad \ldots \quad \alpha_{Mi}\mathbf{U}_M\right]\left[\mathbf{V}_i \quad \mathbf{V}_k \quad \ldots \quad \mathbf{V}_M\right]^T. \tag{4.17}$$

The active impedance matrix equation is then solved after multiplying the concatenated $\mathbf{U}$ and $\mathbf{V}$ matrix in (4.17) to form a dense $N_i \times N_i$ matrix equation and then adding this to the self-interaction matrix of domain $i$, i.e. $\mathbf{Z}_{ii}$. The cost of calculating $\tilde{\mathbf{Z}}_{ik}$ as discussed here scales as $\mathcal{O}(2N_i \times r)$ as opposed to a cost of $\mathcal{O}(N_i \times N_i)$ should the terms $\mathbf{Z}_{ik}$ be used in (4.15).

The effective rank ($r$) of $\tilde{\mathbf{Z}}_{ik}$ also reduces quite significantly as the mutual coupling between the elements reduces. In [30] this is illustrated for the CBFM by plotting $r$ as a function of the separation distance between array elements. In Section 4.5.9 the benefit of using the ACA will be illustrated through an example.

## 4.4 A hybrid MPI/OpenMP parallelisation strategy

Parallel programming can readily be applied to domain decomposition methods such as the CBFM and DGFM. The DGFM is particularly suited to a distributed parallelisation strategy using routines such as the message passing interface (MPI) [98]. Each parallel process that forms part of the computational domain is allocated a certain number of array elements to analyse as shown in Figure 4.3(a). The final result, i.e., the active current distribution for the entire array, is then combined at the end, when each process has completed the analysis of its subset of the problem.

Inherent to such a distributed programming model is the fact that each process is allocated

its own section of memory, which in turn leads to unnecessary overhead in the event that the processes are located on the same machine, operating in a shared memory environment. Another parallel programming paradigm, OpenMP [99], has emerged in recent years, which allows one to run multiple threads of one process in parallel on such an architecture. What makes OpenMP attractive is the avoidance of memory duplication if multiple parallel processes/threads need to access the same data.

The DGFM uses a hybrid OpenMP/MPI parallelisation scheme, whereby the MPI library is used for distributed parallelisation and OpenMP threading for intra-node communication as shown in Figure 4.3(b). The aforementioned is currently applied to the LU-decomposition of the active impedance matrices. An examples of applying parallelisation for the DGFM is presented in Section 4.5.4.

## 4.5 Test cases

A number of examples are considered for the DGFM in this Section. The examples illustrate the numerical accuracy as well as the computational complexity of the method for regular and irregular arrays, different basis functions (illustrated at the hand of conventional RWG as well as HOBF basis functions), and different Green's functions including the free-space and planar multi-layered Green's function. In addition, the speed improvements for the calculation of the active impedance matrix equations, as discussed in Section 4.3 and Section 4.4, are also shown. Furthermore, in Section 4.5.7 the DGFM will be applied to calculate the RCS of a FSS, which requires a particular modification for the $\alpha_{qp}$ weighting coefficients and in Section 4.5.8 the DGFM will be compared with the PBC implementation in FEKO [40, 41].

A relative error norm percentage ($\epsilon_{\%}$), defined as follows,

$$\epsilon_{\%} = \sqrt{\sum_{n}^{N} |I_n^{\text{ref}} - I_n|^2 / \sum_{n}^{N} |I_n^{\text{ref}}|^2} \times 100\%, \tag{4.18}$$

is used as a figure of merit for rigorously comparing the accuracy between various results in the following subsections. The vector, $\{I_n\}$, contains the values being evaluated, e.g. the far field calculated with the DGFM, and $\{I_n^{\text{ref}}\}$ is the reference solution, e.g. the results associated with the full-wave MoM. The quantities used in (4.18) will clearly be defined for each particular example where it is used.

### 4.5.1 Finite strip dipole antenna array

The strip dipole array illustrated in Figure 4.4 with $M_x = M_y = 50$, $\Delta_x = \Delta_y = \lambda$ and $L = \lambda/2$ was simulated using both the DGFM and a full-wave MoM solution. The analysis frequency is set to $f = 2$ GHz. The entire structure is discretised using 37,500 RWG basis-functions [47], with $N_i = 15$ basis functions per dipole. Each array element is excited with the same amplitude and phase.

The far field gain pattern of the array was calculated using both the DGFM and MoM

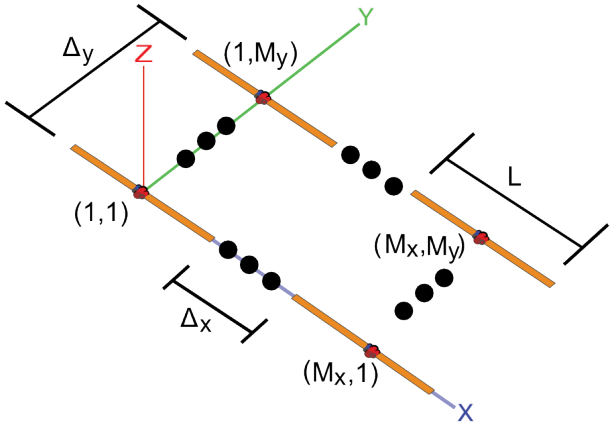



**Figure 4.4:** Strip dipole antenna array consisting of $N$ elements

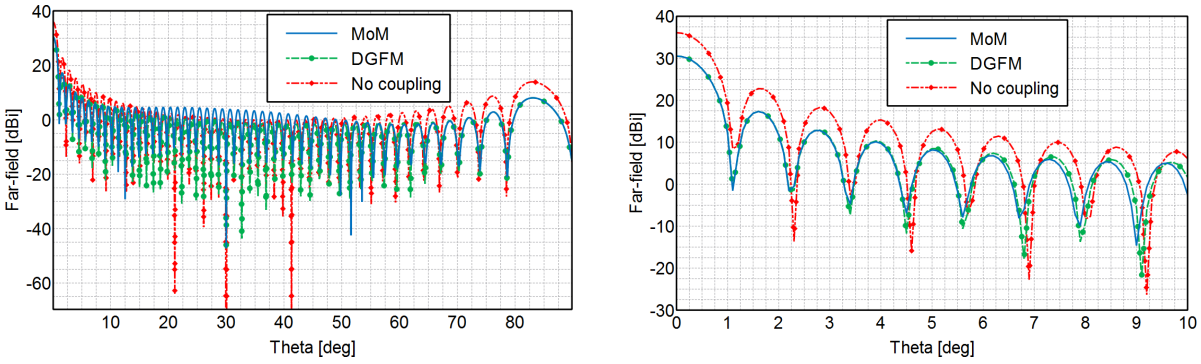
respectively and is illustrated in Figure 4.5(a) and (b). The result for the case where inter-element coupling is neglected has also been added, which is equivalent to that calculated with an active element pattern method discussed in Section 3.1.1. The computational complexity of the DGFM compared to the MoM is summarised in Table 4.2. The timing results include the calculation of the far field pattern, and the memory usage include that of storing the finite array geometry.

The relative error norm percentage ($\epsilon_{\%}$) defined in (4.18) has also been evaluated for the DGFM as well as for the case where mutual coupling is neglected. The expansion coefficients associated with the active surface current distribution is used for the calculation, with the MoM used as reference. The results have been added to Table 4.2. The relative error has significantly been improved with the DGFM through the addition of mutual coupling effects as explained in Section 4.1. Compared to the MoM this value of $\epsilon_{\%} = 20.51\%$ for the DGFM can contribute to the observable differences in the side lobe levels, particularly between $15° \leq \theta \leq 45°$.

The far field results calculated with the DGFM compares well to that of the MoM, with the peak value differing by 0.014% relative to the MoM result. For the case where mutual coupling is neglected, this deviation increases to 18.25%. Furthermore, a significant saving is gained in both runtime and memory usage when using the DGFM as compared to the MoM, as we analyse only a single array element at a time.

**Table 4.2:** Runtime and memory requirement for the analysis of the strip dipole array of Figure 4.5(a) on an Intel Xeon CPU with 2.67 GHz clock rate using the DGFM and MoM respectively.

| **Method** | **Total runtime** | **Total memory usage** | $\epsilon_{\%}$ |
|---|---|---|---|
| MoM | 5.6 hours | 21.0 GByte | – |
| DGFM | 2.3 hours | 168.8 MByte | 20.51% |
| No coupling | 39.430 seconds | 168.8 MByte | 355.37% |



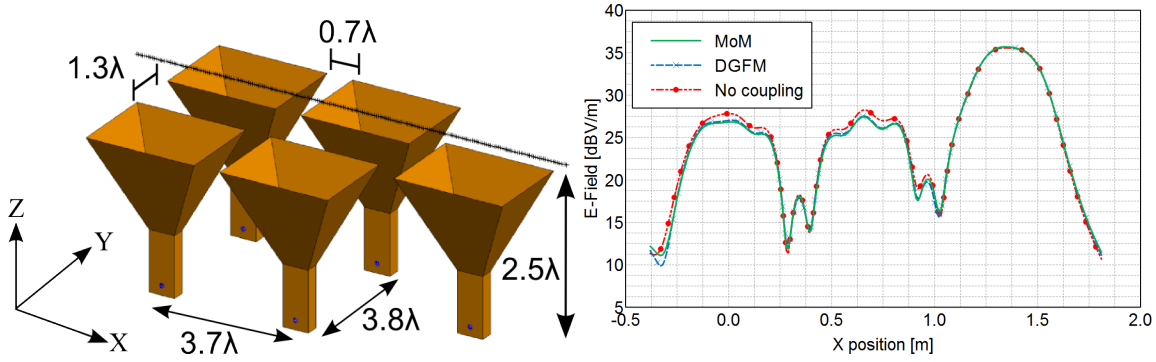

(a) (b)

**Figure 4.5:** (a) Comparison of the far field gain patterns obtained with the MoM and DGFM, respectively, for the strip dipole array shown in Figure 4.4. The result for the case where mutual coupling between elements are excluded has been added for comparison. (b) The same as (a) but for $0° \leq \theta \leq 10°$.

### 4.5.2 Aperiodic horn cluster antennas

To quantify the numerical accuracy and computational complexity of the DGFM, a non-uniform spaced pyramidal horn antenna array illustrated in Figure 4.6(a) was simulated using both the DGFM and a full-wave MoM solution. The analysis frequency is set to $f = 1.645$ GHz. The entire structure is discretised using 31,790 RWG type basis-functions [47], with $N_i = 6{,}358$ basis functions per horn element. Each array element is excited with the same amplitude and phase using a pin-feed placed inside the waveguide. The element spacing and horn geometry is shown in Figure 4.6(a).

The near field patterns of the array were calculated using both the DGFM and MoM respectively and are illustrated in Figure 4.6(b). To illustrate the manner in which the DGFM accounts for the mutual coupling between the elements, inter-element coupling was also excluded from the simulation and is added to the near field results for reference. By applying the relative error norm calculation to the near field results in Figure 4.6(b), with the MoM used as reference, values of $\epsilon_{\%} = 1.15\%$ for the DGFM and $\epsilon_{\%} = 2.78\%$ for the no coupling case are calculated. Again it is clear that the DGFM improves the manner in which coupling is accounted for in the

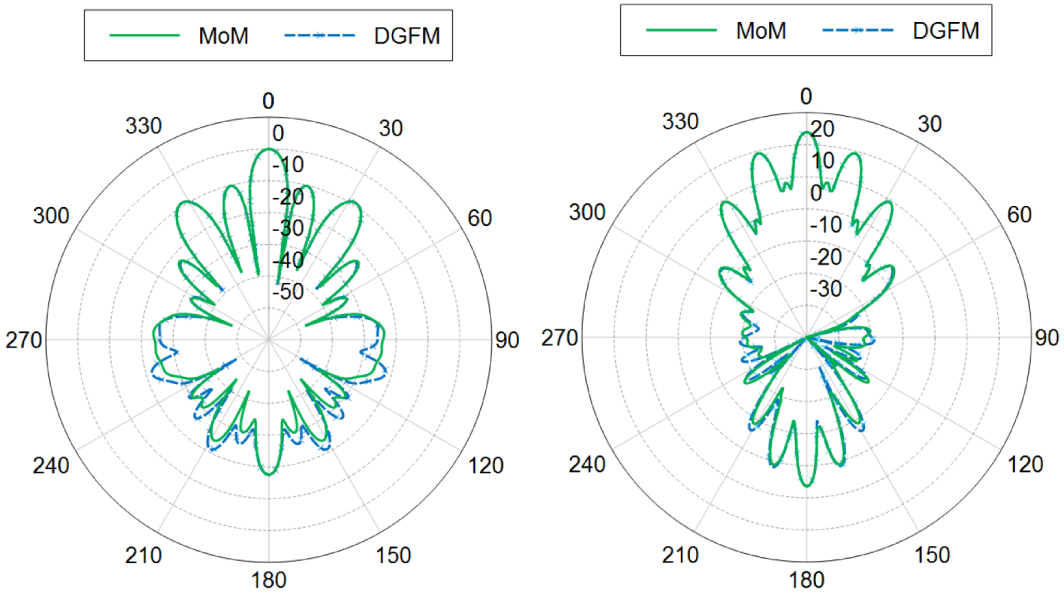



(a) Horn anntenna array (b) Total E-Field along the X-axis (dBV/m)

**Figure 4.6:** Applying the DGFM to a horn antenna configuration displayed in (a), and (b) the near field pattern measured $0.7\lambda$ above the horn aperture along the x-axis.

array environment when compared to the no coupling case.

The far field gain pattern of the array in the E and H-planes[4] was also calculated using both the DGFM and MoM respectively and is illustrated in Figure 4.7(a) and (b). The computational complexity of the DGFM compared to the MoM is summarised in Table 4.3. The relative error norms associated with the active current distribution on the array calculated with the DGFM and no coupling case, as compared to the MoM, is also added to Table 4.3. The numerical results compare well and a significant saving is gained in both runtime and memory usage when using the DGFM.




(a) Total E-plane gain pattern (dB)

(b) Total H-plane gain pattern (dB)

**Figure 4.7:** Applying the DGFM to a horn antenna configuration displayed in Figure 4.6(a) with (a) and (b) the E- and H-plane gain patterns respectively. All results are obtained for the active array environment where all the elements are excited equally and simultaneously.

**Table 4.3:** Runtime and memory requirement for the analysis of the horn array configuration of Figure 4.6(a) on an Intel Xeon CPU X5550 at 2.67 GHz clock rate using the DGFM and MoM respectively.

| **Method** | **Total runtime** | **Total memory usage** | $\epsilon$% |
|---|---|---|---|
| MoM | 1.47 hours | 7.54 GByte | – |
| DGFM | 16.78 min | 323.4 MByte | 3.80% |
| No coupling | 3.36 minutes | 323.4 MByte | 5.02% |

[4]For the array configuration of Figure 4.6(a), the E-plane corresponds to the ZY-plane and th H-plane to the ZX-plane respectively

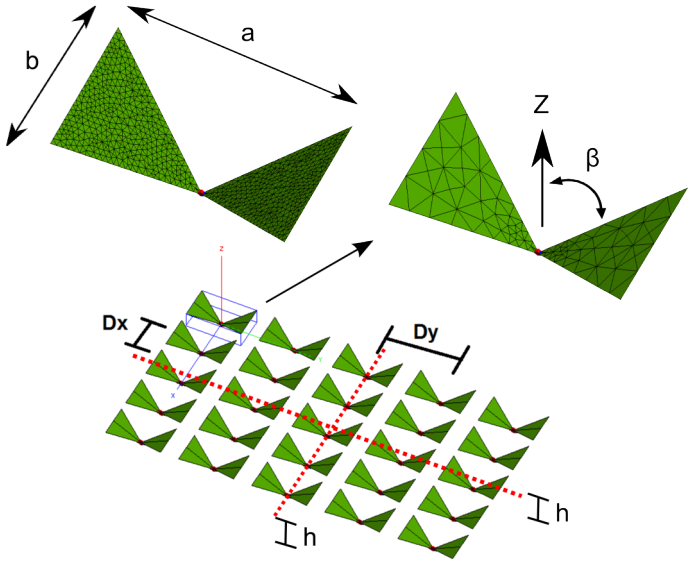



**Figure 4.8:** Finite array geometry, consisting of bow-tie elements meshed with planar triangles of side lengths in the order of $\lambda/10$ (top left) and $\lambda$ (top right). The near field request points are shown in red.

### 4.5.3 Bow-tie antenna array modelled with HOBFs

The efficient numerical analysis of large and complex geometry using the MoM solution technique, in connection with low-level RWG basis functions [47], is prohibitive in terms of runtime and memory usage. The reason for this is that conventional RWG elements require a fine mesh (typically $\lambda/10$) to approximate the current distribution on the structure accurately. Electrically large problems will therefore result in a large number of unknowns ($N$). When considering the computational complexity of the MoM, i.e. a scaling as $\mathcal{O}(N^2)$ in terms of memory usage and $\mathcal{O}(N^3)$ for solving the set of linear equations, the full-wave direct solution of large and complex structures places a significant burden on computational resources. To mitigate this limitation, hierarchical higher order basis functions (HOBFs) can be used that allow for larger triangular patches for the discretisation, and therefore fewer unknowns, as discussed in Section 2.5. In [52], the DGFM was applied to a bow-tie array modelled with the HOBFs applied on large, flat triangular patches. The geometry is presented in Figure 4.8.

The $5 \times 5$ bow-tie array geometry illustrated in Figure 4.8 was analysed using the MoM, with both conventional RWGs and HOBFs, respectively, as well as the DGFM with HOBFs. The simulation results were obtained for the case where all the elements are excited with unity amplitude and zero phase. The number of RWGs and HOBFs associated with each of the array elements is, 2,073 and 479, respectively. In total this amounts to 51,825 RWGs and 11,975 HOBFs for the 25 element array. The aforementioned includes HOBFs of polynomial order up to $p = 2.5$. The array elements are spaced along the $x$-axis with $D_x = 1.06\lambda$, and along the $y$-axis with $D_y = 1.92\lambda$. The bow-tie element dimensions are $a = 1.44\lambda$, $b = 0.83\lambda$ and $\beta = 60°$, respectively.

The near field patterns were calculated along a line in the $\hat{x}$ and $\hat{y}$ directions at a height of $h = 0.2\lambda$ (as illustrated in Figure 4.8) and is shown in Figure 4.9(a) and (b), respectively. The case where mutual coupling is neglected in the HOBF simlutation, has also been added

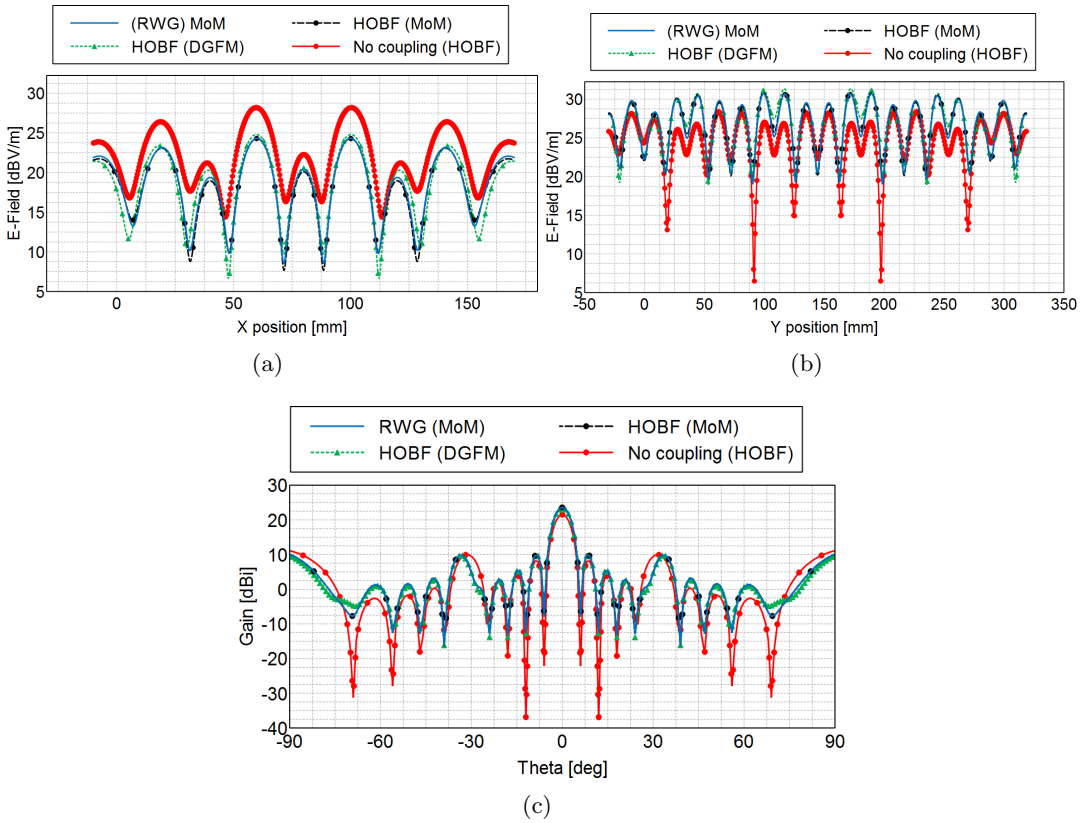



**Figure 4.9:** (a) Near field in the $\hat{x}$ direction, (b) $\hat{y}$ direction, and (c) far field gain pattern calculated for the $5 \times 5$ bow-tie array illustrated in Figure 4.8.

**Table 4.4:** Relative error norms for the near field and far field quantites shown in Figure 4.9.

| **Method** | $\epsilon_{\%}$ (near field: $\hat{x}$ direction) | $\epsilon_{\%}$ (near field: $\hat{y}$ direction) | $\epsilon_{\%}$ (far field) |
|---|---|---|---|
| RWG | – | – | – |
| HOBF | 2.176% | 1.028% | 0% |
| HOBF-DGFM | 7.45% | 2.76% | 16.59% |
| (HOBF) no coupling | 21.16% | 11.70% | 69.60% |

for comparison. The far field gain pattern was calculated along the $E$-plane (the $ZY$-plane). The results are illustrated in Figure 4.9(c) and also compared in terms of the relative error norm percentage defined in (4.18), with the references taken as the results calculated with the conventional RWG discretisation.

The runtime and memory usage are summarised in Table 4.5, with the relative error norm percentage associated with the active current distribution on the structures also included. For the latter, the HOBF results were used as reference. The results calculated with the HOBF-DGFM agrees well with that of the HOBF (and RWG cases) and a significant improvement in memory usage is also observed for the HOBF-DGFM combination.

The runtime of the HOBF-DGFM is comparable to that of the HOBF case, because the calculation of the matrix elements dominates the total solution time for this example.

**Table 4.5:** Runtime and memory requirement for the analysis of the $5 \times 5$ bow-tie array geometry illustrated in Figure 4.8 on an AMD Opteron CPU at 1.4 GHz clock rate, using the MoM with RWGs and HOBFs and also the DGFM with HOBFs, respectively.

| Method | Total runtime | Total memory usage | $\epsilon_{\%}$ |
|---|---|---|---|
| RWG | 6.3 hours | 20.0 GByte | – |
| HOBF | 1.0 hours | 1.1 GByte | – |
| HOBF-DGFM | 1.14 hours | 5.5 MByte | 9.45% |
| (HOBF) no coupling | 2.66 minutes | 5.5 MByte | 20.11% |

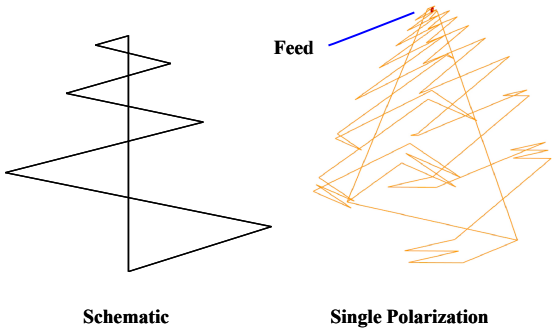
### 4.5.4 A 26 element array of dual-polarised Zig-Zag antennas

To illustrate the effect of limiting the mutual coupling during the calculation of the active impedance matrix to the $N_c$ dominating elements, as explained in Section 4.3.2, a 26 element array consisting of dual-polarised Zig-Zag antennas were considered, as presented in [53].

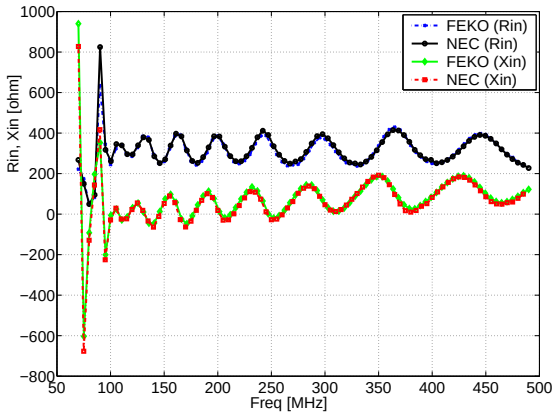
The Zig-Zag geometry is derived from that of a log-periodic antenna, as introduced by DuHamel [100]. Here, the geometry is simplified to a triangular tooth or Zig-Zag structure, as proposed by DuHamel in [101]. The Zig-Zag geometry is presented in Figure 4.10(a) where the trapezoidal tooth structure is illustrated (left) as well as the manner in which it is folded to obtain a single-polarised antenna. The Zig-Zag element is folded to limit the footprint of the antenna to 1 $m^2$ [102] in accordance with the SKA1-low specifications [103].

The real and imaginary components of the input impedance of the single-polarised (folded) Zig-Zag element is presented in Figure 4.10(b). The results were obtained with FEKO [6] and the Numerical Electromagnetics Code (NEC) [104] respectively.

The input impedance oscillates around 300 Ω over a fairly wide frequency band, ranging from 70 MHz to 450 MHz, in line with the requirements of the SKA1-low frequency range. Two single-polarised elements can also be rotated 90° with respect to each other, in order to achieve




(a) Schematic representation of the Zig-Zag geometry and a single-polarised antenna with the arms folded.




(b) Real and imaginary parts of the input impedance for the single-polarised Zig-Zag antenna, simulated with FEKO and NEC

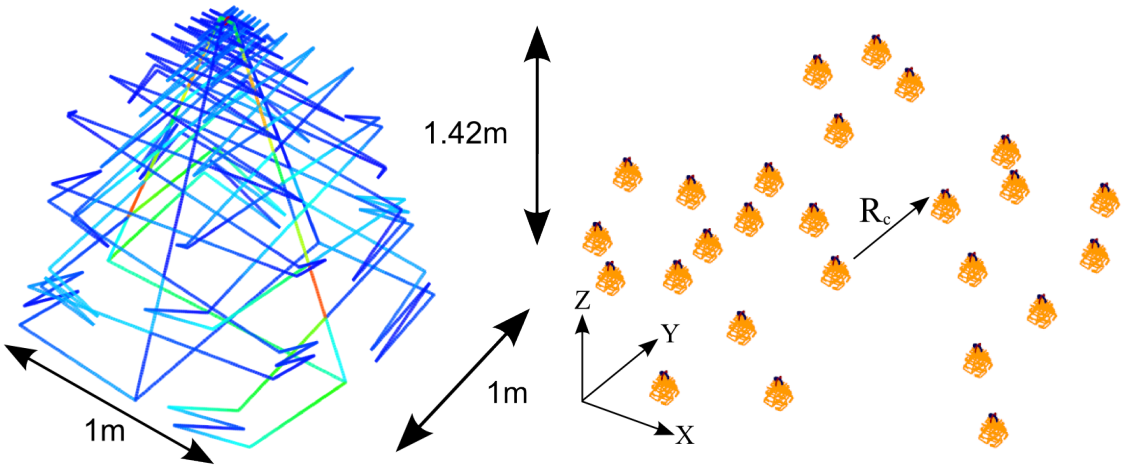
**Figure 4.10:** (a) Schematics representation and single-polarisation of the Zig-Zag antenna; and (b) the real and imaginary part of the input impedance of the single-polarised element simulated with FEKO and NEC, respectively.

dual-polarisation. In the remainder of this Section, an array of such dual-polarised Zig-Zag elements will be analysed with the DGFM.

The 26 element Zig-Zag array is illustrated in Figure 4.11(a) and is analysed using the MoM and DGFM with $R_c$ values of 100%, 50% and 0%, respectively. For $R_c = 0\%$, only self-coupling for the element is assumed. The element spacings range from between $\lambda/2$ to $3\lambda$ at 70 Mhz. Although spacings of $\lambda/2$ are clearly denser than normally needed in an irregularly-spaced array, it is chosen as such to illustrate the effect of various $R_c$ values on the accuracy of the DGFM. The element is discretised by using 325 elementary basis functions, leading to a total of 8,450 unknowns.

The far field gain pattern is analysed along the ZX-plane for scan-angles of $\theta = 0°$ and $\theta = 60°$, and is illustrated in Figure 4.12(a) and (b). The DGFM results for $R_c = 100\%$ compare well with those obtained by the MoM solution. Included in the results are those obtained for $R_c = 0\%$ and $R_c = 50\%$ values. Figure 4.11(c) and (d) illustrate the errors in dB for $R_c = 0\%$ and $R_c = 50\%$ compared to the DGFM with $R_c = 100\%$. As shown, the error is slightly larger than 1 dB for certain off-broadside angles. The reason for the aforementioned, is as follows. For a slowly varying excitation law, the *initial* condition that the currents be identical on all subdomains, except for a complex scaling factor, is particularly true for near-broadside scan angles and for antenna elements that support only a low number of antenna modal currents. These limitations were also observed in [105], where a similar perturbation approach has been described for the CBFM.

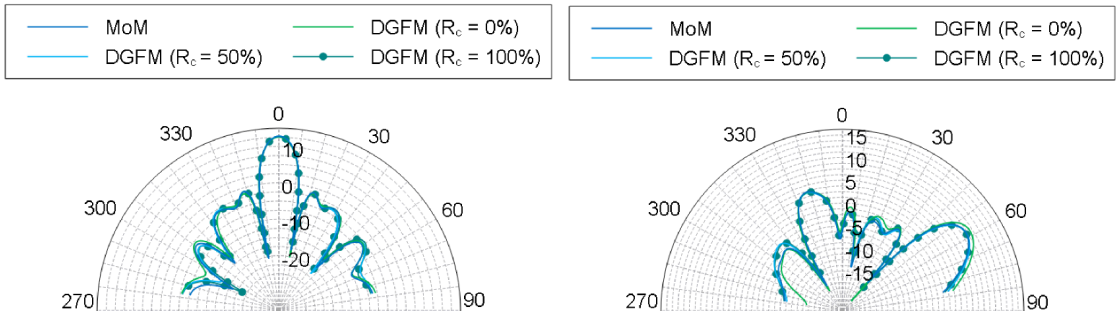
The computational complexity of the DGFM compared to the MoM is summarised in Table 4.6. The matrix fill-time for the conventional DGFM (i.e., with $R_c = 100\%$) is similar to that of the MoM; however significant savings can be observed in the matrix solution phase (which is dominated by the LU-decomposition) as well as in the memory usage.




(a) Dual-polarised Zig-Zag element geometry

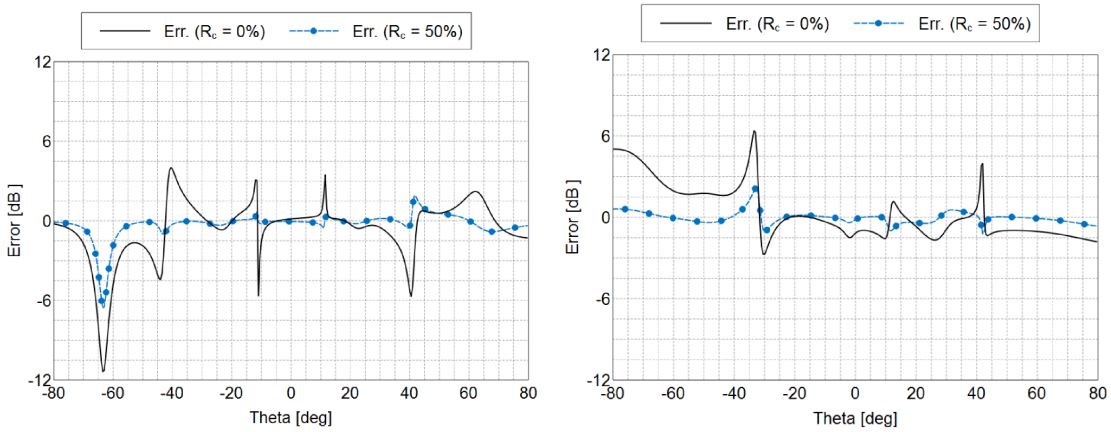
(b) Array configuration containing 26 irregularly spaced Zig-Zag elements. The element spacings range from $\lambda/2$ to $3\lambda$, at an operating frequency of 70 MHz

**Figure 4.11:** A Zig-Zag antenna displayed in (a), in a 26 element irregular array configuration displayed in (b)




(a) Total directivity pattern (dBi) for a scan-angle of $\theta = 0°$, $\phi = 0°$

(b) Total directivity pattern (dBi) for a scan-angle of $\theta = 60°$, $\phi = 0°$




(c) Error (dB) in the calculated directivity for $R_c = 0\%$ and $R_c = 50\%$ compared to $R_c = 100\%$. Considered scan angle: $(\theta = 0°, \phi = 0°)$

(d) Error (dB) in calculated directivity for $R_c = 0\%$ and $R_c = 50\%$ compared to $R_c = 100\%$. Considered scan angle: $(\theta = 60°, \phi = 0°)$

**Figure 4.12:** The directivity patterns for scan-angles of $(\theta = 0°, \phi = 0°)$ and $(\theta = 60°, \phi = 0°)$ are presented in (c) and (d), respectively. The errors in the directivity for different $R_c$ values are presented in (e) and (f), where $R_c = 100\%$ is used as reference. All results are obtained for the active array environment where all the elements are excited equally and simultaneously.

**Table 4.6:** Runtime and memory requirement for simulating a 26 element Zig-Zag array using the MoM and DGFM.

| **Method:** | **MoM** | **DGFM** ($R_c = 100\%$) |
|---|---|---|
| Time for matrix setup | 326.24 sec | 379.69 sec |
| Time for matrix solution | 188.616 sec | 0.83 sec |
| Memory usage (whole solution) | 1.072 GByte | 7.56 MByte |

Table 4.7 summarizes the runtimes for the matrix fill phase, associated with the various $R_c$ values. From the results it can be concluded that the accuracy of the DGFM can be controlled by changing the value of $R_c$. To illustrate the efficiency of the parallelisation of the DGFM as discussed in Section 4.4, the 26 element array was simulated using 5 compute nodes. The measured runtime speedup[5] is shown in Figure 4.13 and compares well to the ideal case.

[5]In [106] speedup was defined as *The ratio of time taken by an equivalent serial algorithm running on one*

**Table 4.7:** Runtime and memory requirement for simulating a 26 element Zig-Zag array using different $R_c$ values.

| **Method:** | **DGFM** ($R_c = 100\%$) | **DGFM** ($R_c = 50\%$) | **DGFM** ($R_c = 0\%$) |
|---|---|---|---|
| Number of coupling terms for DGF ($N_c$) | 29 | 15 | 1 |
| Time for matrix setup | 379.69 sec | 199.43 sec | 1.22 sec |

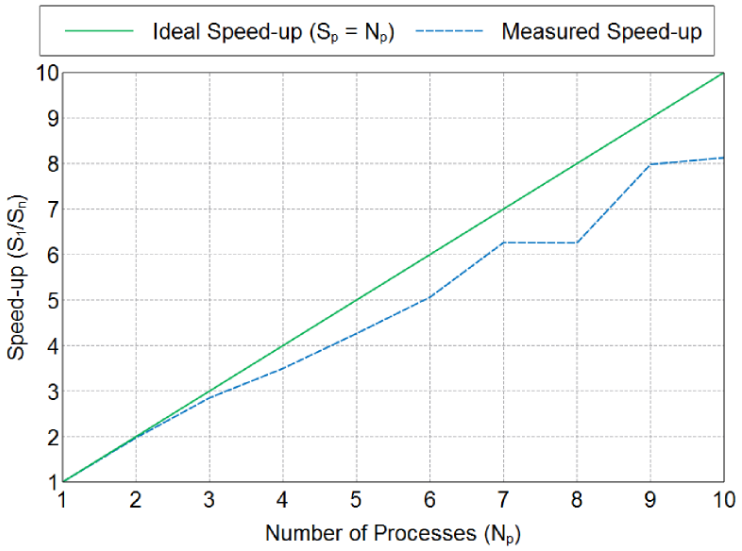
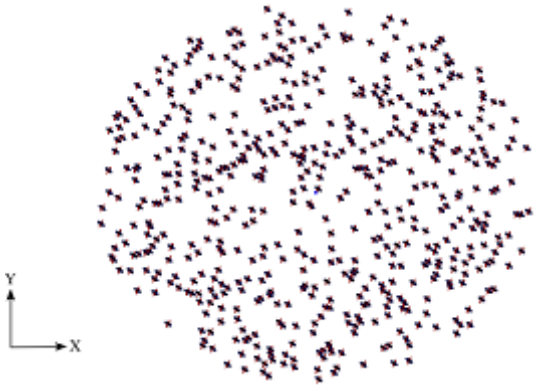



**Figure 4.13:** Runtime speedup measured for the 26 element Zig-Zag element array.

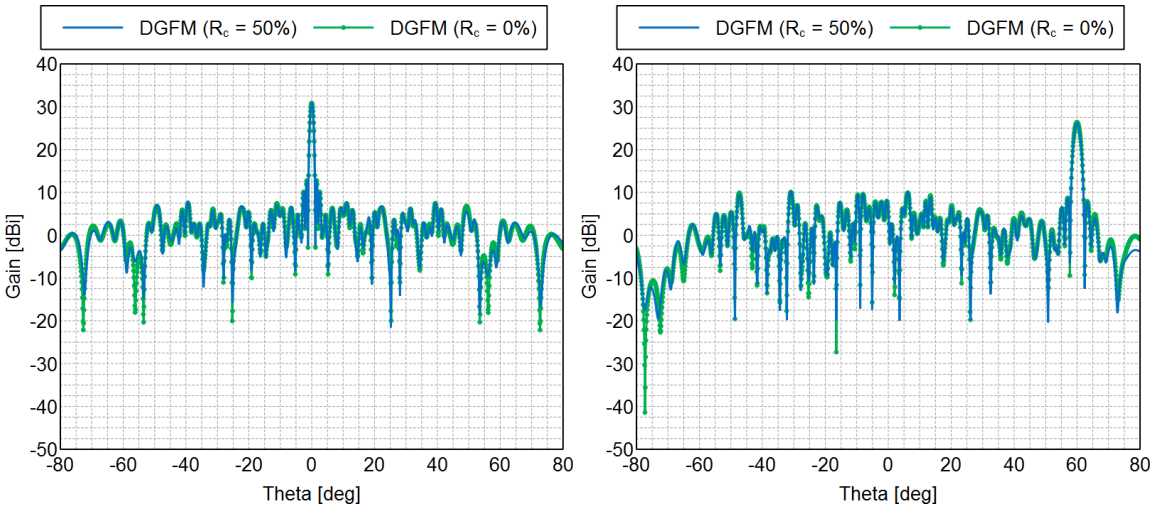
### 4.5.5 A 529 element irregularly spaced array of Zig-Zag antennas

Following the methodologies explained in Section 4.5.4, a larger 529 element dual-polarised Zig-Zag array [53] was analysed by using 8 distributed parallel processes on a Linux EM64T platform with 2 x quad core processors. The array geometry is illustrated in Figure 4.14(a). The array consists of 171,925 elementary basis functions with inter-element spacings again ranging from between $\lambda/2$ and $3\lambda$, at an operating frequency of 70 MHz. Convergence threshold values of $R_c = 0\%$ and $R_c = 50\%$ were used for the analysis. The computational complexity of the simulations is summarised in Table 4.8. The far field gain patterns for $\theta = 0°$ and $\theta = 60°$ are illustrated in Figure 4.14(b) and (c). From the results it can be seen that the errors in the modeled mutual coupling only has a minor effect for $-80° \leq \theta \leq -50°$ and $50° \leq \theta \leq 80°$, and even less for $\pm 45°$ off-vertical for SKA1-low. The accelleration strategy for the DGFM introduced in Section 4.3.2 has permitted a computationally efficient investigation to be undertaken. This can be applied to other candidate antennas, such as the log periodic proposed in [107].

*processor, $T_s$, to the time taken by the parallel algorithm using $N$ processors, $T_p$*




(a) Array configuration containing 529 irregularly spaced Zig-Zag elements. The element spacings range from $\lambda/2$ to $3\lambda$, at an operating frequency of 70 MHz



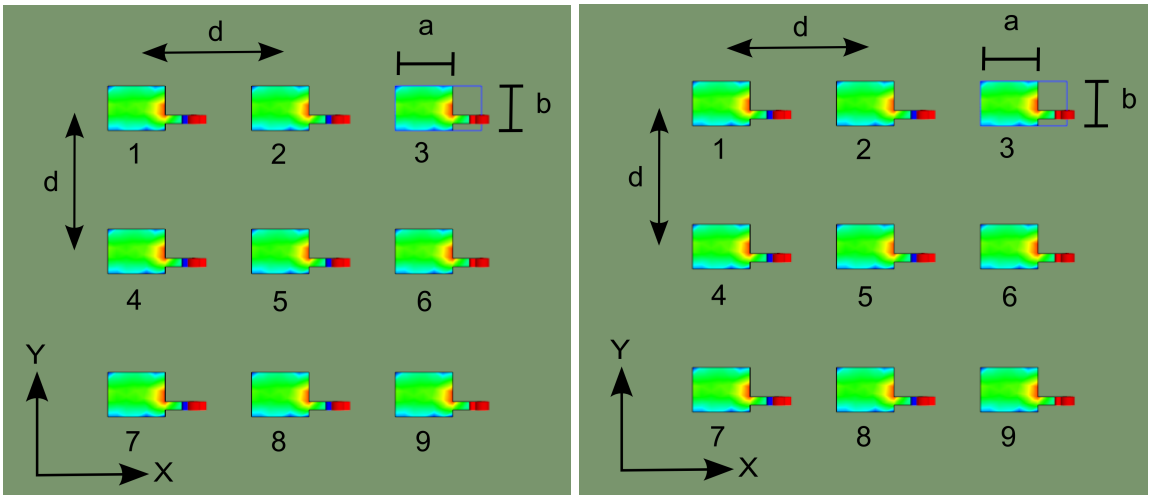

(b) Total gain pattern (dB) for a scan-angle of $\theta = 0°$, $\phi = 0°$

(c) Total gain pattern (dB) for a scan-angle of $\theta = 60°$, $\phi = 0°$

**Figure 4.14:** Applying the DGFM to the Zig-Zag antenna displayed in Figure 4.10(a) in a 529 element array configuration. The gain patterns for scan-angles of ($\theta = 0°$, $\phi = 0°$) and ($\theta = 60°$, $\phi = 0°$) are presented in (b) and (c) respectively. All results are obtained for the active array environment where all the elements are excited equally and simultaneously.

**Table 4.8:** Runtimes and memory requirements for the analysis of the Zig-Zag array configuration of Figure 4.14(a) on a Linux EM64T platform with 2 x Intel Xeon quad core processors at 2.67 GHz clock rate each. The DGFM with $R_c = 0\%$ and $R_c = 50\%$ was used, with 8 parallel processes.

| **Method:** | **DGFM** ($R_c = 0\%$) | **DGFM** ($R_c = 50\%$) |
|---|---|---|
| Total solution time | 14.23 min | 2.27 hours |
| Memory usage (per process) | 143.1 MByte | 143.1 MByte |




(a) (b)

**Figure 4.15:** Calculating the surface current on a printed patch antenna using the multi-layered Green's function and solved with (a) the MoM and (b) the DGFM, respectively

**Table 4.9:** Runtime and memory requirement for the analysis of the patch antenna array of Figure 4.15(a) and (b) on an Intel EM64T CPU with 2.67 GHz clock rate using the DGFM and MoM respectively.

| Method | Total runtime | Total memory usage |
|---|---|---|
| MoM | 6.64 min | 5.124 MByte |
| DGFM | 6.11 min | 568.03 kByte |

### 4.5.6 Patch antenna array on a multi-layered substrate

As explained in Section 4.1, the DGFM can be expressed in terms of a block matrix factorisation, i.e., equivalent of a summation of the Green's function as done for printed antennas array using the multi-layered Green's function [44, 45]. In this example, a 3-by-3 patch antenna array was simulated with the MoM and DGFM. The active current distribution calculated with each method is shown in Figure 4.15(a) and (b). The array element dimension is $a = 12.45$ mm and $b = 16$ mm and the elements are spaced $d = 40$ mm in both the $Y$ and $X$ direction. The elements are placed on a infinite grounded subtrate with $\epsilon_r = 2.213$, with a substrate thickness of 0.794 mm and a loss tangent of $\tan\delta = 0.0032$. The analysis frequency range is between 7.4 GHz and 8.0 GHz and the number of RWG basis functions is $N_i = 86$ for the array element, which results in a total of $N = 774$ basis functions for the array. The active reflection coefficient was calculated for elements 1 and 5, i.e. $S_{11}$ and $S_{55}$, with the MoM and DGFM and is shown in Figure 4.16. It is clear that the DGFM takes edge-effects into account by considering the difference between $S_{11}$ and $S_{55}$, respectively. The computational complexity for the methods is summarised in Table 4.9. For this relatively small example, the runtime benefit of using the DGFM is not that significant, although when looking at the memory usage, the DGFM still performs much better compared to the MoM.

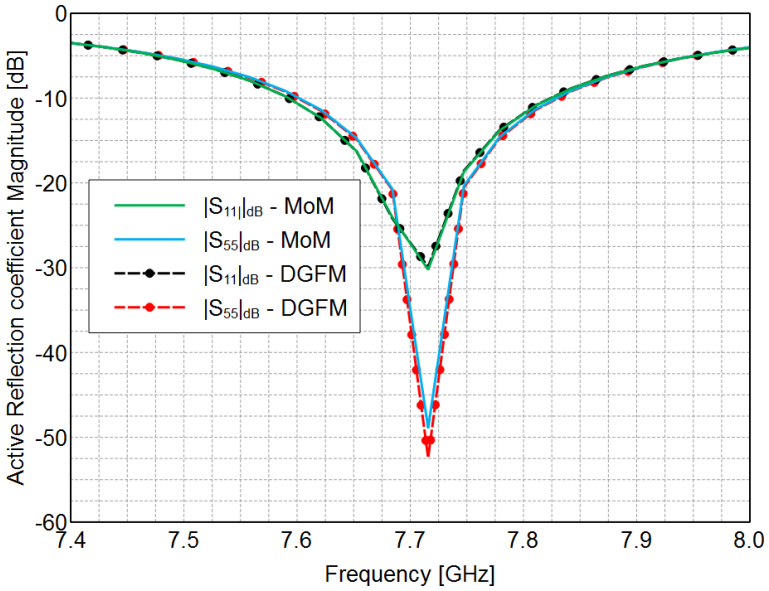



**Figure 4.16:** Active input reflection coefficients for elements 1 and 5 for the patch antenna array configuration that is shown in Figure 4.15(a) and (b).

### 4.5.7 RCS of a Jerusalem Cross FSS

In this example, a single-layered 11-by-11 Jerusalem Cross FSS [108] was simulated with the MoM and DGFM, respectively, and the RCS results were calculated along the $ZX$-plane for $0° \leq \theta \leq 60°$. The geometry is shown in Figure 4.17(a) and the RCS results in Figure 4.17(b). The array element dimension is 10 mm by 10 mm and they are spaced $d = 15.2$ mm apart in both the $Y$ and $X$ direction. The analysis frequency is set to 2.0 GHz and the number of RWG basis functions is $N_i = 41$ for the array element, which results in a total of $N = 4,961$ basis functions for the array.

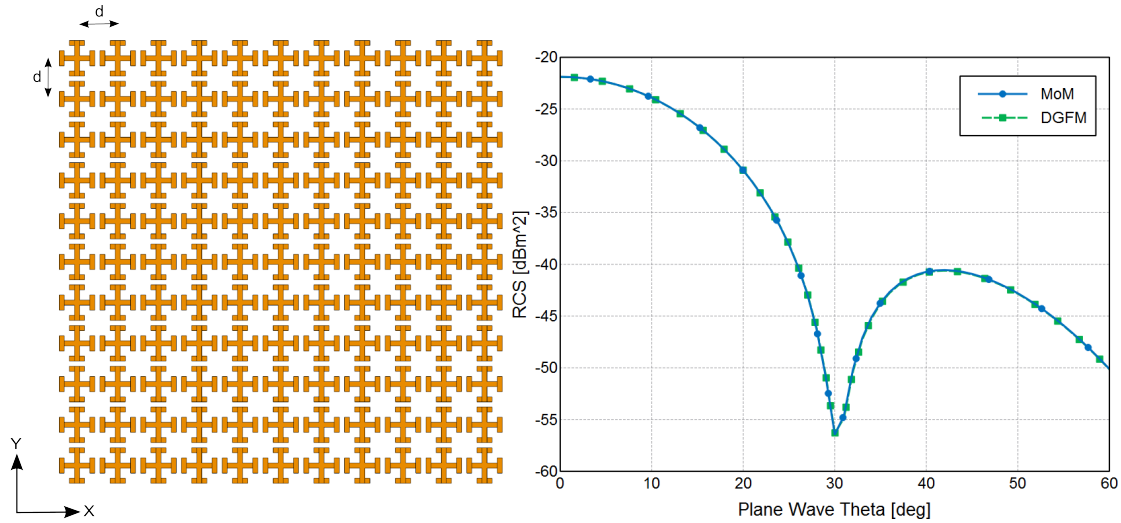
For this example, the $\alpha$ weight coefficients defined in Section 4.1 should account for the incident plane wave propagation direction, as well as the array element positions, as follows,

$$\alpha_{pq} = e^{-j\vec{k}\cdot\vec{r}_{pq}}. \tag{4.19}$$

The vector $\vec{k}$ is the propagation vector for the incident plane wave and is defined as follows,

$$\vec{k} = k(\sin\theta_\circ \cos\phi_\circ \hat{x} + \sin\theta_\circ \sin\phi_\circ \hat{y} + \cos\theta_\circ \hat{z}), \tag{4.20}$$

where $k = \frac{2\pi}{\lambda}$ and $(\hat{\theta}_\circ, \hat{\phi}_\circ)$ defines the angle of arrival of the plane wave in terms of usual spherical coordinate conventions. The vector $\vec{r}_{pq}$ defines the relative position between array elements $p$ and $q$. The computational complexity for the methods is summarised in Table 4.10. From the results it can be seen that the runtime associated with the MoM is much faster than that of the DGFM for this particular case. The reason for this is that the active impedance matrices are excitation dependent, i.e., they need to be recalculated for each incident plane wave direction. For the MoM, the impedance matrix pertaining to the whole array is calculated and factorised only for the first iteration, after which each following incident direction only results in a fast




(a) (b)

**Figure 4.17:** (a) A Jerusalem Cross FSS geometry in free-space and (b) The RCS results calculated with the MoM and DGFM along the $ZX$-plane.

**Table 4.10:** Runtime and memory requirement for the analysis of the Jerusalem Cross FSS of Figure 4.17(a) on an Intel EM64T CPU with 2.67 GHz clock rate using the DGFM and MoM respectively.

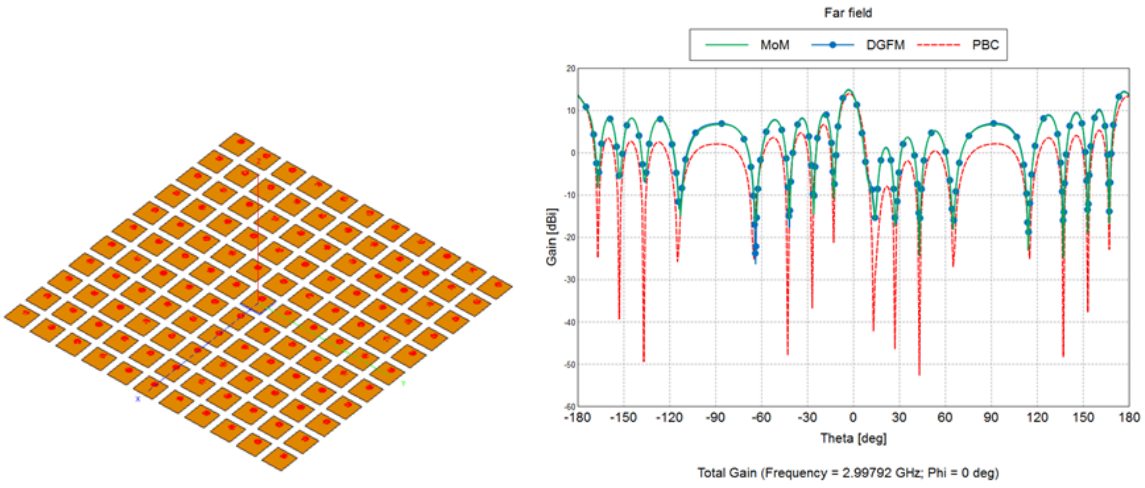
| Method | Total runtime | Total memory usage |
|---|---|---|
| MoM | 1.08 min | 131.484 MByte |
| DGFM | 47.52 min | 2.859 kByte |

backwards substitution to solve for the current distribution. The memory usage of the DGFM, however, still performs much better than that of the MoM, as can be seen from Table 4.10.

### 4.5.8 Comparing the PBC with the DGFM for a patch antenna array

In this example, the DGFM is compared to the MoM, as well as the infinite periodic Green's function, i.e., using the periodic boundary conditions (PBC) implemented in FEKO [40, 41], for an 11-by-11 pin-fed PEC patch array shown in Figure 4.18(a). The analysis frequency is set to 3 GHz and the total number of unknowns is 5,300 and the far field is calculated along the $ZX$-plane for $-180° \leq \theta \leq 180°$. The computational complexity associated with each of the methods is shown Table 4.11.

From the results shown in Figure 4.18(b) it is clear that the edge effects are not taken into account for the PBC, as expected, although this approximation leads to a significant saving in both runtime and memory usage, as shown in Table 4.11. The MoM and DGFM takes the edge effects into account and in doing so the computational complexity also increases. It is important to note at this stage that the DGFM is not designed to be a replacement for the PBC or MoM, but instead should be viewed as an intermediary analysis tool when considering finite antenna arrays.




(a) (b)

**Figure 4.18:** (a) An 11-by-11 pin fed patch antenna array, and (b) the far field calculated using the DGFM, MoM and PBC, respectively.

**Table 4.11:** Runtime and memory requirement for the 11-by-11 pin fed patch array shown in Figure 4.18(a) on an Intel Xeon CPU with 2.67 GHz clock rate using the MoM, DGFM and PBC, respectively.

| Method | Total runtime | Total memory usage |
|---|---|---|
| MoM | 3.35 min | 219.9 MByte |
| DGFM | 2.2 min | 3.34 MByte |
| PBC | 7 sec | 66.2 kByte |

### 4.5.9 Applying the DGFM and ACA algorithm for the analysis of an irregular bow-tie antenna array

To quantify the numerical accuracy and computational benefit of applying the ACA algorithm to the DGFM as discussed in Section 4.3.3, the bow-tie array illustrated in Figure 4.19 was analysed. The analysis frequency is set to $f = 2.7$ GHz, and the total number of RWG basis-functions is 7,429. A convergence threshold of $\kappa = 10^{-2}$ was used for the partially pivoted ACA algorithm that is explained in more detail in Appendix A.

In Figure 4.20(a) and (b) the $E$ and $H$-plane gain patterns calculated at broadside are shown for the various methods. The computational complexity of the techniques are summarised in Table 4.12. The total relative error norm percentage ($\epsilon_{\%}$) for the active current distribution, calculated for the DGFM with and without the ACA, has also been added to Table 4.12. The MoM solution is used as reference. To clearly show the error introduced by the ACA algorithm, the results associated with a convergence threshold of $\kappa = 0.5$ have been added. The runtime speedup is defined as that of the DGFM (with and without the ACA) relative to the MoM. The results show a significant runtime speedup when using the ACA with the DGFM, without sacrificing accuracy. By using a relaxed convergence criteria for the ACA algorithm (i.e. with $\kappa = 0.5$), the speedup improves, although, at the cost of an increasing error introduced by the

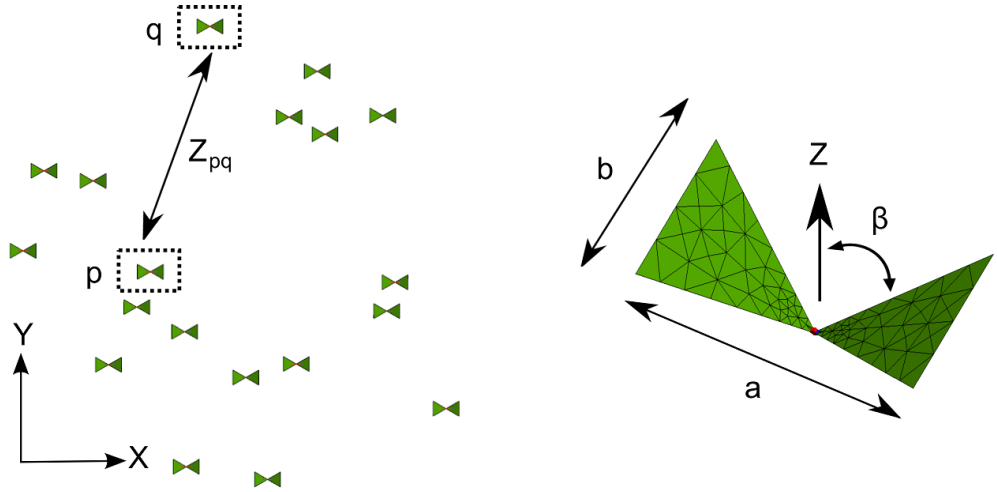
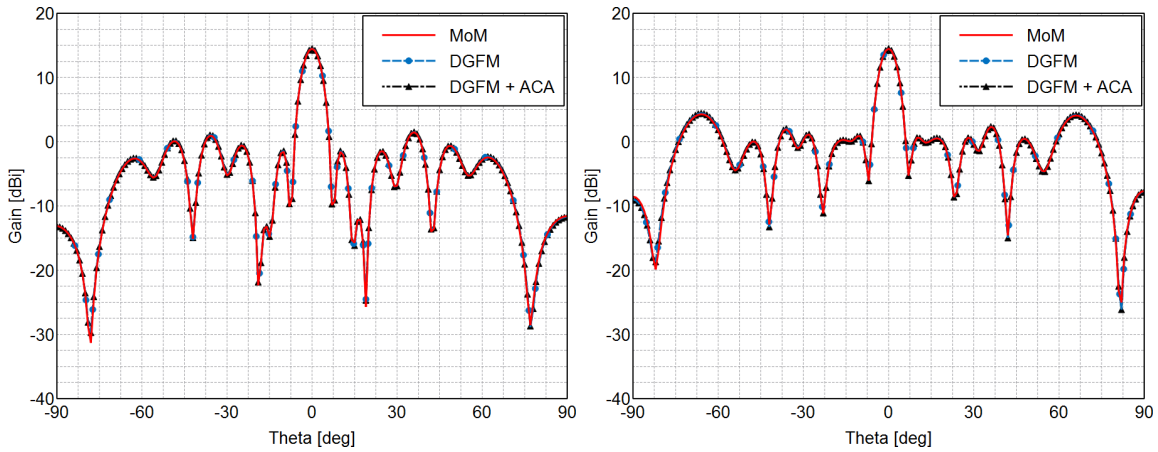



**Figure 4.19:** Aperiodic antenna array consisting of 19 arbitrarily spaced bow-tie elements.

approximation. Not shown in Table 4.12 is memory usage, which for the MoM is 843 MByte and for the DGFM is 2.33 MByte, respectively.




(a) Total $E$-plane gain patterns calculated with the MoM and DGFM (with and without the ACA).

(b) Total $H$-plane gain patterns calculated with the MoM and DGFM (with and without the ACA).

**Figure 4.20:** Applying the DGFM with ACA acceleration to a 19 element bow-tie array configuration displayed in Figure 4.19. (a) The $E$-plane gain pattern calculated for the array at $\theta = 0°$,$\phi = 0°$ using the MoM and DGFM with and without the ACA. (b) The $H$-plane gain pattern calculated for the array at $\theta = 0°$,$\phi = 90°$ using the MoM and DGFM with and without the ACA. All results are obtained for the active array environment where all the elements are excited equally and simultaneously.

## 4.6 Conclusion

The DGFM, presented in this Chapter, provides an easy and efficient approach for analysing disjoint finite antenna arrays. The approach is analogous to the perturbation method for the infinite array solution discussed in Section 3.3.1, and is formulated in Section 4.1 on a higher block-matrix factorisation level. This simplifies the underlying implementation, as a summation

**Table 4.12:** Runtime speedup and accuracy for the analysis of the bow-tie array configuration of Figure 4.19 using the MoM and DGFM, with and without the ACA, respectively.

| **Method:** | **Runtime Speedup** | Total-$\epsilon_{\%}$ | ACA-$\epsilon_{\%}$ |
|---|---|---|---|
| MoM | – | – | – |
| DGFM | 2.04 | 2.1988% | – |
| DGFM (with ACA - $\kappa = 0.01$) | 12.2 | 2.1999% | 0.001% |
| DGFM (with ACA - $\kappa = 0.5$) | 15.4 | 5.8379% | 3.639% |

of the Green's function as done in [44, 45], is not required. Instead, mutual coupling between the array elements is quantified for the DGFM with the formulation of an active impedance matrix equation for each of the domains.

In Section 4.2 various differences and similarities between the DGFM and CBFM was highlighted as well a discussion on the computational complexity of each method. Different acceleration techniques for the calculation of the active impedance matrix equation was presented in Section 4.3. Of these methods, the ACA algorithm is especially attractive, as it can be used for virtually any array configuration. In addition, a hybrid MPI/OpenMP parallelisation strategy presented in Section 4.4 also allows the DGFM to benefit from parallel computing environments. Finally, various numerical examples, as presented in Section 4.5, quantifies the accuracy and computational complexity of the DGFM.

In the following Chapter, an accuracy improved version of the DGFM, viz., the i-DGFM will be presented.

# 5

# The improved DGFM (i-DGFM)

As discussed in Chapter 4, a perturbation approach such as the DGFM requires that the current distribution between the elements be slowly varying. For this very reason, the accuracy of the DGFM deteriorates when investigating e.g. off-broadside scan angles for phased arrays, or the calculation of embedded element patterns (EEPs). In this Chapter, a novel technique is proposed for improving the accuracy of the DGFM. The technique is based on applying an approximate solution obtained iteratively with a Jacobi decomposition of the MoM block matrix factorisation [53]. The improvements also include the derivation of a more accurate weighting coefficient matrix, viz., the $\boldsymbol{\alpha}$ weighting matrix, that is used during the active impedance matrix calculation for each array element [54]. The weighting matrix entries express the ratio between the individual basis functions associated with the identically discretised domains. The improved version of the DGFM is called the i-DGFM.

A brief overview of the Chapter is as follows: In Section 5.1 the $\boldsymbol{\alpha}$ weighting matrix is derived by concentrating on the ratio of the individual basis functions between array elements, as opposed to the applied voltage coefficients. In Section 5.2, a more accurate expression for the current on the individual array elements is obtained through the use of Jacobi generated CBFs as done in [53]. In Section 5.3, the computational complexity of the method is considered, and compared with that of the DGFM and CBFM. In Section 5.4 the accuracy of the i-DGFM is presented at the hand of different test cases, that include phased array simulations, as well as the calculation of EEPs. The Chapter is then concluded in Section 5.5.

## 5.1 Deriving a weight matrix

The $\alpha_{qp}$ weighting coefficients defined in Section 4.1 represents the ratio between the currents on two domains, viz. the domain being analysed (domain $p$), and the domain from which coupling is considered (domain $q$). In Section 4.2 this ratio was expressed in terms of the applied excitations, or the primary CBFs, as follows,

$$\alpha_{qp} = \frac{V_q}{V_p} = \frac{\langle \boldsymbol{\Lambda}, \mathbf{J}_{0q} \rangle}{\langle \boldsymbol{\Lambda}, \mathbf{J}_{0p} \rangle}. \tag{5.1}$$

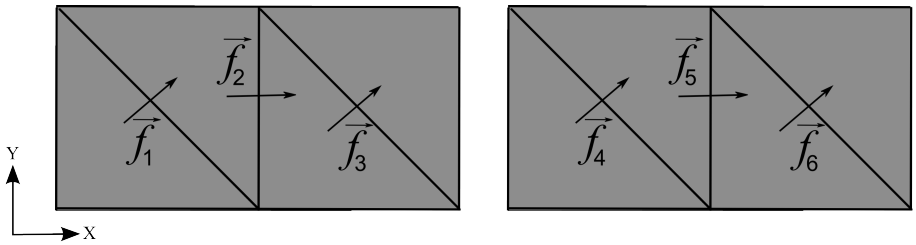



**Figure 5.1:** Geometry consisting of 2 domains, $\Omega_1$ (left) and $\Omega_2$ (right), with 3 basis functions each.

If we consider that $\mathbf{J}_{0p}$ and $\mathbf{J}_{0q}$ are the full-wave MoM currents on domains $p$ and $q$, respectively, then the limitation of (5.1) is such that the resulting currents on these domains will have similar spatial distributions after applying the DGFM. A better approach would be to derive an $\boldsymbol{\alpha}$ weight matrix to express the ratio between the *individual basis functions* associated with the discretised array elements.

To derive such a matrix, consider the simple geometry shown in Figure 5.1 with two identically discretised domains, $\Omega_1$ and $\Omega_2$. Each domain supports three subsectional basis functions, $(\vec{f}_1, \vec{f}_2, \vec{f}_3)$ and $(\vec{f}_4, \vec{f}_5, \vec{f}_6)$, respectively. The MoM matrix equation, $\mathbf{ZJ} = \mathbf{V}$, for this particular problem can be written as,

$$\begin{bmatrix} Z_{11} & Z_{12} & Z_{13} & Z_{14} & Z_{15} & Z_{16} \\ Z_{21} & Z_{22} & Z_{23} & Z_{24} & Z_{25} & Z_{26} \\ Z_{31} & Z_{32} & Z_{33} & Z_{34} & Z_{35} & Z_{36} \\ Z_{41} & Z_{42} & Z_{43} & Z_{44} & Z_{45} & Z_{46} \\ Z_{51} & Z_{52} & Z_{53} & Z_{54} & Z_{55} & Z_{56} \\ Z_{61} & Z_{62} & Z_{63} & Z_{64} & Z_{65} & Z_{66} \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_5 \\ I_6 \end{bmatrix} = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \end{bmatrix}, \tag{5.2}$$

where $I_1, \ldots, I_6$ are the unknown complex expansion coefficients associated with the basis functions $\vec{f}_1, \ldots, \vec{f}_6$. The current on $\Omega_1$ is defined by the expansion coefficients, $I_1, \ldots, I_3$, that is associated with the first three rows of (5.2). This may be written as a set of linear equations as follows,

$$\begin{aligned} Z_{11}I_1 + Z_{12}I_2 + Z_{13}I_3 + Z_{14}I_4 + Z_{15}I_5 + Z_{16}I_6 &= V_1 \\ Z_{21}I_1 + Z_{22}I_2 + Z_{23}I_3 + Z_{24}I_4 + Z_{25}I_5 + Z_{26}I_6 &= V_2 \\ \underbrace{Z_{31}I_1 + Z_{32}I_2 + Z_{33}I_3}_{\text{Coupling internal to } \Omega_1} + \underbrace{Z_{34}I_4 + Z_{35}I_5 + Z_{36}I_6}_{\text{Coupling from } \Omega_2} &= V_3 \end{aligned} \tag{5.3}$$

The coupling from $\Omega_2$ is represented by the last three terms in each of the equations in (5.3). The goal, as with domain decomposition schemes in general, is to solve (5.3), by considering only the current on $\Omega_1$. As we already have the exact current ratio between $\Omega_1$ and $\Omega_2$, respectively, we can achieve the aforementioned by expressing the expansion coefficients on $\Omega_2$, i.e., $I_4, \ldots, I_6$,

in terms of the *corresponding* expansion coefficients on $\Omega_1$, as follows,

$$\begin{aligned} I_4 &= \alpha_{41} I_1 \\ I_5 &= \alpha_{52} I_2 \\ I_6 &= \alpha_{63} I_3 \end{aligned} \tag{5.4}$$

Equation 5.3 can now be rewritten in terms of only $I_1, \ldots, I_3$ using (5.4), as follows,

$$\begin{aligned} (Z_{11} + \alpha_{41} Z_{14}) I_1 + (Z_{12} + \alpha_{52} Z_{15}) I_2 + (Z_{13} + \alpha_{63} Z_{16}) I_3 &= V_1 \\ (Z_{21} + \alpha_{41} Z_{24}) I_1 + (Z_{22} + \alpha_{52} Z_{25}) I_2 + (Z_{23} + \alpha_{63} Z_{26}) I_3 &= V_2 \\ (Z_{31} + \alpha_{41} Z_{34}) I_1 + (Z_{32} + \alpha_{52} Z_{35}) I_2 + (Z_{33} + \alpha_{63} Z_{36}) I_3 &= V_3 \end{aligned} \tag{5.5}$$

Equation 5.5, can be rewritten in matrix format as,

$$\begin{bmatrix} (Z_{11} + \alpha_{41} Z_{14}) & (Z_{12} + \alpha_{52} Z_{15}) & (Z_{13} + \alpha_{63} Z_{16}) \\ (Z_{21} + \alpha_{41} Z_{24}) & (Z_{22} + \alpha_{52} Z_{25}) & (Z_{23} + \alpha_{63} Z_{26}) \\ (Z_{31} + \alpha_{41} Z_{34}) & (Z_{32} + \alpha_{52} Z_{35}) & (Z_{33} + \alpha_{63} Z_{36}) \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix}. \tag{5.6}$$

We can rewrite the impedance matrx in (5.6) using the Hadamard product[1] as follows,

$$\begin{bmatrix} Z_{11} & Z_{12} & Z_{13} \\ Z_{21} & Z_{22} & Z_{23} \\ Z_{31} & Z_{35} & Z_{36} \end{bmatrix} + \begin{bmatrix} \alpha_{41} & \alpha_{52} & \alpha_{63} \\ \alpha_{41} & \alpha_{52} & \alpha_{63} \\ \alpha_{41} & \alpha_{52} & \alpha_{63} \end{bmatrix} \circ \begin{bmatrix} Z_{14} & Z_{15} & Z_{16} \\ Z_{24} & Z_{25} & Z_{26} \\ Z_{34}) & Z_{35} & Z_{36} \end{bmatrix}. \tag{5.7}$$

Equation 5.7 represents the active impedance matrix for $\Omega_1$ that, analogous to (4.6), can be written more generally for domain $p$ as,

$$\mathbf{Z}_p^{\text{act}} = \sum_{m=1}^{M} \boldsymbol{\alpha}_{mp} \circ \mathbf{Z}_{pm}, \tag{5.8}$$

where $M$ is the number of array elements. The $\boldsymbol{\alpha}$ factors will be termed the improved $\boldsymbol{\alpha}$ weight matrix that supports more rapid current variation between the domains, as compared to the $\alpha$ weighting coefficients defined in (5.1). The reason for this is that each individual basis function is now associated with a complex scaling factor.

An alternative derivation for the above is done in [54], where the $\boldsymbol{\alpha}$ weights are expressed as a diagonal matrix as follows,

$$\boldsymbol{\alpha}_{qp} = \text{diag}(I_1^q / I_1^p, I_2^q / I_2^p, ..., I_N^q / I_N^p), \tag{5.9}$$

where $I_n^p$ is the $n$th expansion coefficient associated with the $p$th domain. Equation (5.5) can then be rewritten such that $\mathbf{J}_q = \boldsymbol{\alpha}_{qp} \mathbf{J}_p$. Accordingly, the active impedance matrix for domain

[1] The Hadamard product, otherwise known as the Schur or entrywise product, between two $m \times n$ matrices, $\mathbf{A}$ and $\mathbf{B}$, is a matrix of dimension $m \times n$ with elements given by $(\mathbf{A})_{i,j}) \cdot (\mathbf{B})_{i,j})$.

$p$ in (5.8) can then be expressed as follows, using (5.9),

$$\mathbf{Z}_p^{\text{act}} = \sum_{m=1}^{M} \mathbf{Z}_{pm}\boldsymbol{\alpha}_{mp}. \tag{5.10}$$

It should be noted that from an implementation point of view (5.8) is more efficient. A review of the computational complexities comparing the DGFM and i-DGFM will be given in Section 5.3. In the following subsection, a method is proposed to calculate an initial approximation for the currents on domains $p$ and $q$ from which the improved $\boldsymbol{\alpha}_{qp}$ weight matrix can be calculated.

## 5.2 Improving the initial current approximation

The key to deriving the $\boldsymbol{\alpha}_{qp}$ weight matrix, is to calculate an accurate representation for the current distributions domains $p$ and $q$, in a computationally efficient manner, i.e. without calculating the full-wave MoM current on each array element. In [53], a method is proposed that applies the iterative Jacobi method [109, 110] with the calculation of primary and secondary CBFs to approximate the current on a disjoint finite antenna array.

To explain the method consider again a finite array geometry consisting of $M$ domains, as shown in Figure 4.1. As noted in Section 4.1, the MoM impedance matrix can be block-partitioned as follows,

$$\mathbf{Z} = \begin{bmatrix} \mathbf{Z}_{11} & \mathbf{Z}_{12} & \cdots & \mathbf{Z}_{1M} \\ \mathbf{Z}_{21} & \mathbf{Z}_{22} & \cdots & \mathbf{Z}_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{Z}_{M1} & \cdots & \cdots & \mathbf{Z}_{MM} \end{bmatrix}. \tag{5.11}$$

Equation (5.11) can also be written as the combination of an on and off-diagonal matrix, as follows,

$$\mathbf{Z} = \mathbf{Z}_{\text{on}} + \mathbf{Z}_{\text{off}} = \begin{bmatrix} \mathbf{Z}_{11} & & & \\ & \mathbf{Z}_{22} & & \\ & & \ddots & \\ & & & \mathbf{Z}_{MM} \end{bmatrix} + \begin{bmatrix} & \mathbf{Z}_{12} & \cdots & \mathbf{Z}_{1M} \\ \mathbf{Z}_{21} & & \cdots & \mathbf{Z}_{2M} \\ \vdots & \vdots & & \vdots \\ \mathbf{Z}_{M1} & \mathbf{Z}_{M2} & \cdots & \end{bmatrix}. \tag{5.12}$$

Consider now solving the MoM matrix equation, $\mathbf{ZJ} = \mathbf{V}$, using the expression for the impedance matrix written in (5.12) and multiplying each side of the equation with $\mathbf{Z}_{\text{on}}^{-1}$, i.e.,

$$\begin{aligned} \mathbf{ZJ} &= \mathbf{V} \\ (\mathbf{Z}_{\text{on}}^{-1}\mathbf{Z}_{\text{on}} + \mathbf{Z}_{\text{on}}^{-1}\mathbf{Z}_{\text{off}})\mathbf{J} &= \mathbf{Z}_{\text{on}}^{-1}\mathbf{V} \\ (\mathbf{I} + \mathbf{Z}_{\text{on}}^{-1}\mathbf{Z}_{\text{off}})\mathbf{J} &= \mathbf{J}_0 \\ \mathbf{J} &= (\mathbf{I} + \mathbf{Z}_{\text{on}}^{-1}\mathbf{Z}_{\text{off}})^{-1}\mathbf{J}_0 \end{aligned} \tag{5.13}$$

where the term $\mathbf{J}_0 = \mathbf{Z}_{\text{on}}^{-1}\mathbf{V}$ is the current on the array structure in the absence of subdomain

coupling and $\mathbf{I}$ is the identity matrix. The term, $(\mathbf{I} + \mathbf{Z}_{\text{on}}^{-1}\mathbf{Z}_{\text{off}})^{-1}$, can be rewritten with the aid of the infinite geometric series $(1-r)^{-1} = \sum_{n=0}^{\infty} r^n$. The solution for the array current in (5.13) can now be obtained as follows,

$$\begin{aligned} \mathbf{J} &= (\mathbf{I} + \mathbf{Z}_{\text{on}}^{-1}\mathbf{Z}_{\text{off}})^{-1}\mathbf{J}_0 \\ \mathbf{J} &= \sum_{n=0}^{\infty}(-\mathbf{Z}_{\text{on}}^{-1}\mathbf{Z}_{\text{off}})^n\mathbf{J}_0. \end{aligned} \tag{5.14}$$

Convergence of (5.14) is achieved if the magnitude of the eigenvalue of the principal eigenvector of $\mathbf{Z}_{\text{on}}^{-1}\mathbf{Z}_{\text{off}}$ is less than unity [111]. By applying (5.14) to the block partitioning of the impedance matrix in (5.11), the full-wave MoM solution on each of the $M$ array elements can be written as follows,

$$\begin{bmatrix} \mathbf{J}_1 \\ \mathbf{J}_2 \\ \vdots \\ \mathbf{J}_M \end{bmatrix} = \begin{bmatrix} \mathbf{J}_{01} - \sum_{m=1,m\neq 1}^{M} \mathbf{Z}_{11}^{-1}\mathbf{Z}_{1m}\mathbf{J}_{0m} + \sum_{m=1,m\neq 1}^{M} \left[\mathbf{Z}_{11}^{-1}\mathbf{Z}_{1m}\right]^2 \mathbf{J}_{0m} - \dots \\ \mathbf{J}_{02} - \sum_{m=1,m\neq 2}^{M} \mathbf{Z}_{22}^{-1}\mathbf{Z}_{2m}\mathbf{J}_{0m} + \sum_{m=1,m\neq 2}^{M} \left[\mathbf{Z}_{22}^{-1}\mathbf{Z}_{2m}\right]^2 \mathbf{J}_{0m} - \dots \\ \vdots \\ \mathbf{J}_{0M} - \sum_{m=1,m\neq M}^{M} \mathbf{Z}_{MM}^{-1}\mathbf{Z}_{Mm}\mathbf{J}_{0m} + \sum_{m=1,m\neq M}^{M} \left[\mathbf{Z}_{MM}^{-1}\mathbf{Z}_{Mm}\right]^2 \mathbf{J}_{0m} - \dots \end{bmatrix}, \tag{5.15}$$

where $\mathbf{J}_{0p} = (\mathbf{Z}_{pp})^{-1}\mathbf{V}_p$ is the induced current on the $p$th subdomain in isolation. Note that $\mathbf{J}_{0p}$ corresponds to the primary CBF as generated by the CBFM for subdomain $p$. Furthermore, $\sum_{m=1,m\neq p}^{M} \mathbf{Z}_{pp}^{-1}\mathbf{Z}_{pm}\mathbf{J}_{0m}$ is the contribution of all the secondary CBFs to the domain $p$ [see (3.20) in Section 3.3.2]. Hence, the final MoM solution for the current on each subdomain is the sum of the primary, secondary, tertiary CBFs, and so on, with known excitation coefficients in accordance with the known voltage excitation vector $\mathbf{V}$ and the Jacobi-iterative method in (5.13) and (5.14).

Secondary coupling effects can now be considered for the i-DGFM through (5.15), by approximating the currents on the subdomains $p$ and $q$ after two Jacobi iterations, as follows,

$$\mathbf{J}_p \simeq \mathbf{J}_{0p} - \sum_{m=1,m\neq p}^{M} \mathbf{Z}_{pp}^{-1}\mathbf{Z}_{pm}\mathbf{J}_{0m} \tag{5.16a}$$

$$\mathbf{J}_q \simeq \mathbf{J}_{0q} - \sum_{m=1,m\neq q}^{M} \mathbf{Z}_{qq}^{-1}\mathbf{Z}_{qm}\mathbf{J}_{0m}, \tag{5.16b}$$

which is identical to using primary and secondary CBFs on each of the domains, as done in (3.20), with unity $\beta_n$ weighting coefficients.

By taking the ratio of the expansion coefficients associated with the currents on domains $p$ and $q$ as calculated with (5.16), i.e. $I_n^p$, $I_n^q$, with $n = 1, \dots, N_i$, the $\boldsymbol{\alpha}_{qp}$-coefficient matrix can

be constructed as discussed in Section 5.1, i.e.,

$$\boldsymbol{\alpha}_{qp} = \begin{bmatrix} I_1^q/I_1^p & I_2^q/I_2^p & \cdots & I_N^q/I_N^p \\ I_1^q/I_1^p & I_2^q/I_2^p & \cdots & I_N^q/I_N^p \\ \vdots & \vdots & \ddots & \vdots \\ I_1^q/I_1^p & I_2^q/I_2^p & \cdots & I_N^q/I_N^p \end{bmatrix}. \tag{5.17}$$

Equation (5.17) can be used to calculate the active impedance matrix (and finally the current distribution) for the $p$th domain.

As discussed in [53], (5.16) provides a means to compute the $\boldsymbol{\alpha}_{pq}$ weight matrix by including first-order coupling effects used to model spatially concentrated solutions more accurately in addition to the spatially extended solutions already incorporated in the infinite-array-type of assumption used by the DGFM. In (5.16), two relatively inexpensive Jacobi iterations need to be performed – possibly even limiting the inclusion of only the near-coupling effects – which is then followed by the DGFM for solving the system rapidly through an infinite-array-type assumption. The two Jacobi iterations in (5.16) are analogous to adding primary and secondary CBF components to the currents that are approximated on domains $p$ and $q$ respectively. This leads to a block-diagonal system of linear equations, as opposed to taking a large number of Jacobi iterations or building and solving a reduced matrix equation as in the CBFM. In Section 5.4 different test cases will be presented to illustrate the accuracy improvement achieved by the i-DGFM.

By increasing the number of terms that are used for the current approximation in (5.16), the initial guess for the current on the domains can be improved. This however comes at the cost of increased computational complexity and was not investigated in more detail.

## 5.3 Computational complexity

The use of the $\boldsymbol{\alpha}_{qp}$ weight matrix, as well as the calculation of the primary and secondary CBFs for the Jacobi iterations in (5.16), increases the calculation time of the the i-DGFM when compared to the DGFM. The various acceleration strategies presented in Section 4.3 can however be used to alleviate this cost. Specifically, the ACA algorithm can also be used in the calculation of the secondary CBF terms for domain $p$ in (5.16) for the matrix-vector product, as follows,

$$\sum_{m=1,m\neq p}^{M} \mathbf{Z}_{pp}^{-1}\mathbf{Z}_{pm}\mathbf{J}_{0m} \simeq \sum_{m=1,m\neq p}^{M} \mathbf{Z}_{pp}^{-1}\tilde{\mathbf{Z}}_{pm}\mathbf{J}_{0m}. \tag{5.18}$$

In (5.18) $\tilde{\mathbf{Z}}_{pm}$ represents the low-rank approximation of the dense coupling matrix $\mathbf{Z}_{pm}$ calculated with the ACA algorithm, as explained in Section 4.3.3.

In addition, the memory usage for the i-DGFM now increases from $\mathcal{O}(N_i^2)$ to $\mathcal{O}(2N_i^2)$ for storing the $\boldsymbol{\alpha}_{qp}$ weight matrix. Strictly speaking this is not required, as we perform an element wise matrix multiplication where the coefficient, $\alpha_{qp} = I_n^q/I_n^p$, can be calculated as needed. The computational complexity of each of the primary steps in the i-DGFM, DGFM and CBFM

(*cf.* [54]) is summarised in Table 5.1.

**Table 5.1:** Order of CPU-time and memory usage of the CBFM, DGFM and i-DGFM when applied to the array geometry of Figure 4.1, consisting of $M$ disjoint subarrays each with $N_i$ subsectional basis-functions.

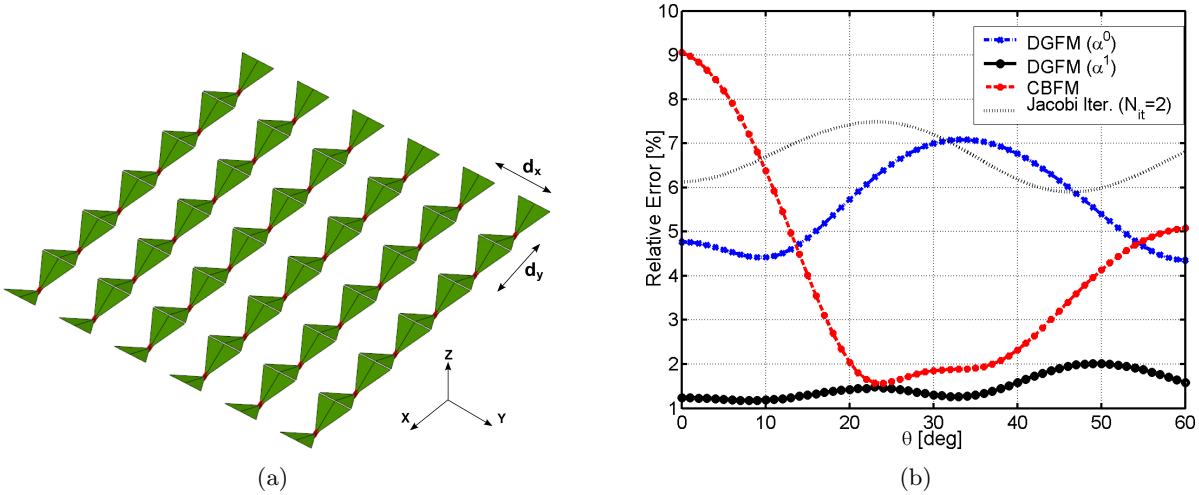
| **Method:** | CBFM | DGFM | i-DGFM |
|---|---|---|---|
| MoM matrix setup time | $\mathcal{O}((M \times N_i)^2)$ | $\mathcal{O}((M \times N_i)^2)$ | $\mathcal{O}((M \times N_i)^2)$ |
| Calculation of Primary basis | $\mathcal{O}(M \times N_i^3)$ | $NA$ | $\mathcal{O}(M \times N_i^3)$ |
| Calculation of Secondary basis | $\mathcal{O}((M \times N_i)^2)$ | $NA$ | $\mathcal{O}((M \times N_i)^2)$ |
| Calculation of the reduced matrix | $\mathcal{O}(M^4 \times N_i^2)$ | $NA$ | $NA$ |
| Calculation of active impedance matrices | $NA$ | $\mathcal{O}(M^2 \times N_i^2)$ | $\mathcal{O}(M^2 \times N_i^2)$ |
| Memory usage (whole solution) | $\mathcal{O}(2N_i^2 + M^4)$ | $\mathcal{O}(N_i^2)$ | $\mathcal{O}(2N_i^2)$ |

## 5.4 Test cases

To illustrate the numerical accuracy and computational efficiency of the i-DGFM, two test cases will be considered next, viz. phased array examples and the calculation of embedded element patterns (EEPs). The examples clearly illustrate the accuracy improvement that can be obtained by the improved $\boldsymbol{\alpha}_{qp}$ weight matrix derived in Section 5.1. In addition, numerical results pertaining to the CBFM and iterative Jacobi method are also included for comparison. In the results, the $\boldsymbol{\alpha}_{qp}$ weight matrix will also be referred to as the $\boldsymbol{\alpha}^{\mathbf{1}}$-weighting coefficients, and the constant $\alpha_{qp}$ weighting coefficients associated with the DGFM, as $\alpha^0$ coefficients.

### 5.4.1 Phased array simulations

Consider the array geometry illustrated in Figure 5.2 (a) that illustrate a $6 \times 6$ bow-tie element array, simulated at $f = 2.7$ GHz. A total of 2844 RWG basis functions are used. The elements are spaced $0.5\lambda$ in both the E and H-planes, i.e., the ZX and ZY planes, respectively. The array was scanned along the E-plane from $\theta = 0°$ to $\theta = 60°$. The relative error norm percentage ($\epsilon_{\%}$) defined in (4.18) was used as a figure of merit for rigorously comparing the accuracy between the results. The expansion coefficients used in (4.18) for this example, i.e., $\{I_n^{\text{ref}}\}$, is calculated with the full-wave MoM and $\{I_n\}$ is the set of coefficients calculated with the DGFM, CBFM or iterative Jacobi method, respectively. The error norm percentage, as a function of scan angle, is presented in Figure 5.2 (b). The results clearly show that the improved $\boldsymbol{\alpha}^1$-weighting coefficients yields better accuracy for the i-DGFM when compared to the DGFM over the whole range of scan angles.




(a) (b)

**Figure 5.2:** (a) A $6 \times 6$ bow-tie antenna array; and (b) the relative error norm percentage for the CBFM, DGFM, i-DGFM and the iterative Jacobi method (using 2 iterations) compared to the MoM.

The computational complexity of the methods are shown in Table 5.2 and was measured on a Windows XP, Pentium 4 with CPU speed at 3 GHz and with 2 GByte of RAM. The solution methods were implemented in MATLAB 6.5. The MoM matrix as exported by the CEM software package, FEKO, was used for the analysis. The simulation runtimes and memory usage is for one scan angle only.

In total, one primary CBF is supported per domain and 35 secondary CBFs are induced by the surrounding array environment. The primary CBFs are used for the zeroth term of the Jacobi iterations, and the secondary CBFs are used for the $n = 1, \ldots, 35$ terms, yielding two iterations of the method. The DGFM with the improved $\boldsymbol{\alpha}^1$-weighting scheme, also makes use of these primary and secondary CBFs to capture a more accurate representation of the current distribution on the array environment through the formulation of the active impedance matrices as discussed in Section 5.1. The CBFM memory is primarily dominated by storing the $1260 \times 1260$ reduced impedance matrix, $\mathbf{Z}^{\text{CBFM}}$, defined in (3.21) in Section 3.3.2.

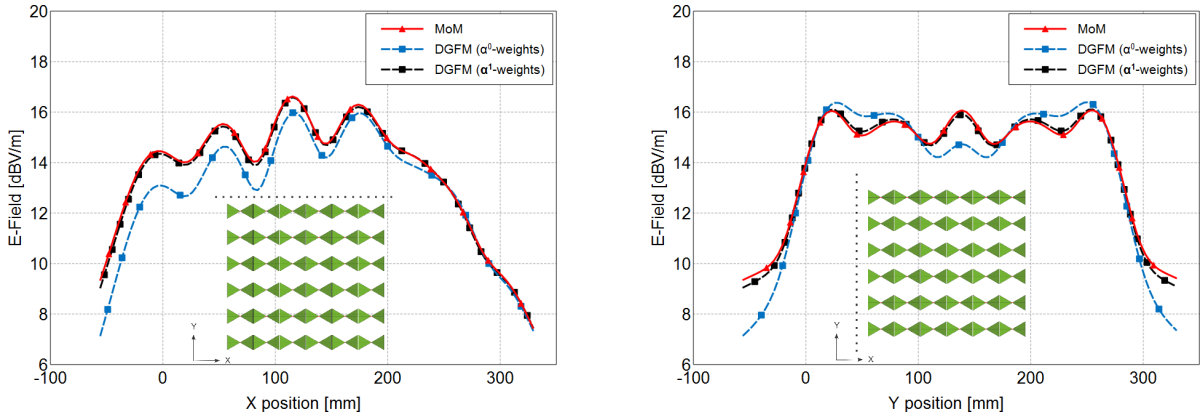
From the results it is clear that the i-DGFM is efficient in terms of both runtime and memory usage, with the time and memory usage associated with the $\boldsymbol{\alpha}^1$-weighting scheme increasing moderately from that of the $\alpha^0$-weighting scheme.

In Figure 5.3 (a) and (b), the near field was measured along the edges of the 36 element bow-tie array, as illustrated. The results clearly show the accuracy improvement as a result of using the $\boldsymbol{\alpha}^1$-weighting coefficients for the DGFM.

The aforementioned case study was repeated with the bow-tie antennas replaced by smaller half wave strip dipoles in a larger 64 element array, as shown in Figure 5.4 (a). The elements are spaced $0.5\lambda$ in both the E and the H plane. In Figure 5.4 (b) the relative error norm percentage is presented as a function of E-plane scan angle, with $\theta$ ranging from $0^\circ$ to $60^\circ$. The MoM solution is used as reference. The DGFM with the $\boldsymbol{\alpha}^1$-coefficients also performs satisfactorily over the whole range of scan angles for this test case.

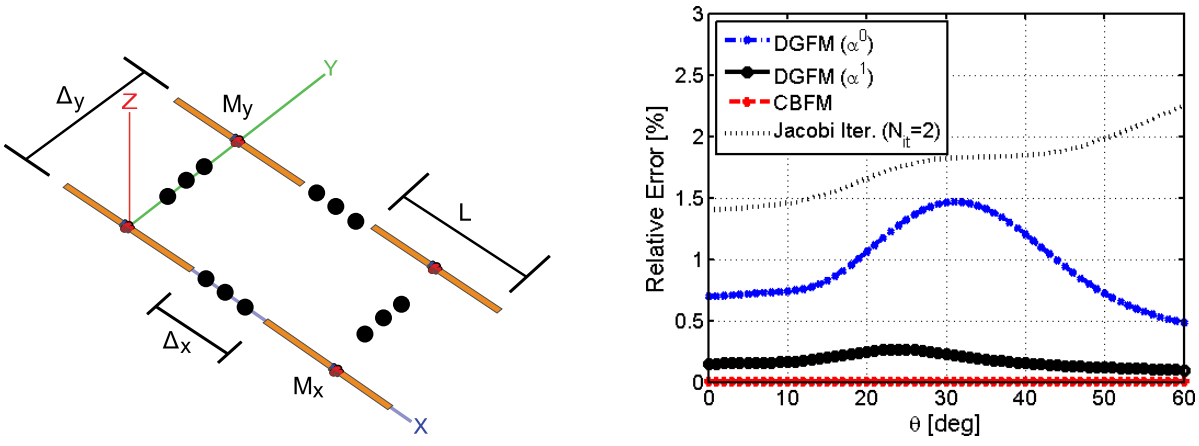
**Table 5.2:** Run-time and memory requirement for the analysis of the bow-tie array configuration of Figure 5.3(a) using the MoM, DGFM, i-DGFM, CBFM and iterative Jacobi methods with 2 iterations, respectively. The results are for one scan angle.

| Method | Solution time | Memory usage |
|---|---|---|
| MoM | 33 sec | 123.42 MByte |
| DGFM ($\alpha^0$-weights) | 0.703 sec | 97.52 kByte |
| i-DGFM ($\boldsymbol{\alpha}^{\mathbf{1}}$-weights) | 4.281 sec | 195.4 kByte |
| CBFM | 7.187 sec | 25.63 MByte |
| Jacobi method ($N_{it} = 2$) | 0.062 sec | 195.4 kByte |
| Calculating prim. + second. CBFs | (1.122 + 74.654) sec | 44.44 kByte + 1.52 MByte |




(a) (b)

**Figure 5.3:** (a) The near field measured along the E-plane, and (b) the E-plane, as illustrated, for the $6 \times 6$ bow-tie antenna array of Figure 5.2 (a).




(a) (b)

**Figure 5.4:** (a) Strip dipole antenna array consisting of 64 elements; and (b) the relative error norm percentage for the CBFM, DGFM (using both $\alpha^0$ and $\boldsymbol{\alpha}^1$-weighting coefficients) and the iterative Jacobi method (using 2 iterations).

### 5.4.2 Embedded element pattern calculation

To compare the i-DGFM with the DGFM, CBFM and Jacobi iterations in a *passive* array environment, the embedded element patterns (EEPs) for an irregular array of strip dipole antennas were calculated. The array elements are the same as that shown in Figure 5.4 (a). The minimum distance between the elements are $0.4\lambda$ and the array elements are contained in a radius of $5\lambda$ as shown in Figure 5.5(a). The array is discretised using 2,109 RWG basis functions. The EEP method discussed in Section 3.1.2 are used here to synthesise the far field of the array.

In Figure 5.5 (b), the relative error norm, $\epsilon_{\%}$, for the i-DGFM, DGFM, CBFM and also the iterative Jacobi method (with 2 iterations) are illustrated for each solution configuration. The independent axis numbering in Figure 5.5 (b) corresponds to the EEP calculation (i.e. solution configuration) for the particular array element with the same index. Solution configuration 20 represents the all excited array case. The MoM solution is again used as reference.

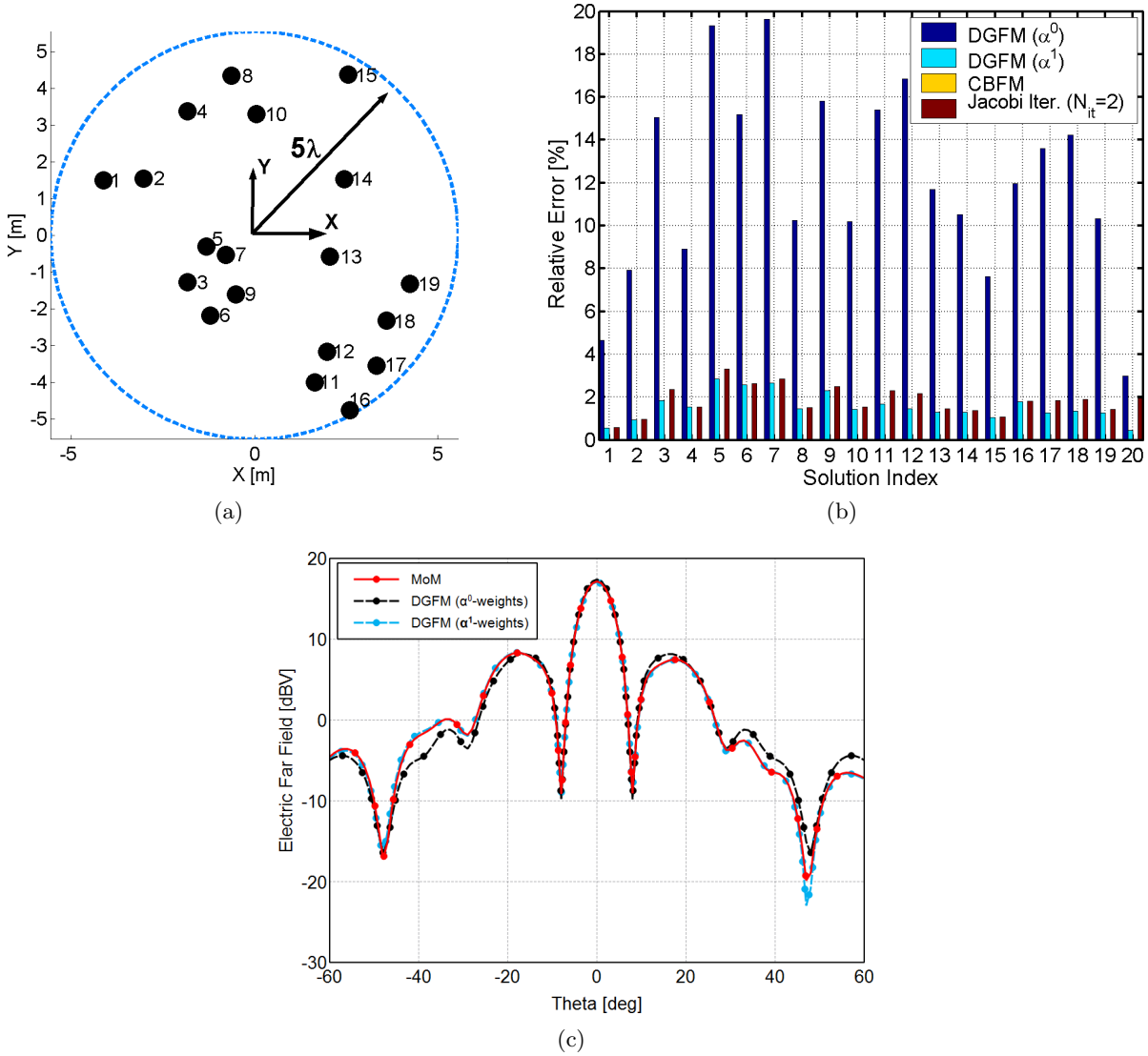
When calculating the contribution from passive array elements in the DGFM, then $\alpha_{pq} = 0$ as $Vq = 0$. The resulting far field will therefore be the same as that of the classical array pattern multiplication method discussed in Section 3.1.1. For the DGFM with $\boldsymbol{\alpha}^1$-weights, the results from the Jacobi iterations are used for the passive elements. The accuracy improvement for the DGFM with the $\boldsymbol{\alpha}^1$ in comparison to the iterative Jacobi method for solution configurations $1, \ldots, 19$ is therefore not as significant as for the all excited array case. For this test case, the CBFM results in an error of nearly 0%, as the primary and secondary CBFs are coupled efficiently through the formulation of the reduced impedance matrix equation, as explained in Section 4.2.

The far field synthesised with the DGFM and i-DGFM is shown in Figure 5.5. The results for the full-wave MoM are included for references. From the results it is clear that the i-DGFM results in more accurate results for this example. In Table 5.3 the computational complexity of each of the methods is presented in terms of memory usage and runtime. The same computing environment as used in Section 5.4.1 was used in this case as well. The runtime is associated with solving all of the solution configurations, i.e. $1, \ldots, 19$ for the EEPs and solution 20, for the active array case. It can be seen that the runtime for the CBFM increases here quite significantly. This is expected, as for this relatively small array the time for constructing the CBFM reduced excitation vector, $\mathbf{V}^{\text{CBFM}}$, dominates the solution time. This will become less significant for larger arrays configurations where the time is dominated by calculating the reduced impedance matrix, $\mathbf{Z}^{\text{CBFM}}$, that is done only once, after which the factorisation thereof is used to solve the reduced matrix equation in (3.21) for multiple excitations.

The runtimes of the DGFM and i-DGFM scale quite well for the multiple excitation case here, as the primary and secondary CBFs that are calculated, are simply extracted for computing the $\alpha^0$ or $\boldsymbol{\alpha}^1$ weights - or summated as in the Jacobi method and used for the passive elements.

## 5.5 Conclusion

In this Chapter, the i-DGFM has been presented as a means of improving the accuracy of the DGFM. The accuracy improvement is achieved by incorporating spatially concentrated coupling




(a) (b)

(c)

**Figure 5.5:** (a) An aperiodic 19 element strip dipole antenna array; and (b) the relative error norm percentage for the i-DGFM, CBFM, DGFM and the iterative Jacobi method (using 2 iterations) for each solution configuration, $1, \ldots, 20$. (c) The array pattern that is synthesised from the EEPs calculated with the i-DGFM, DGFM and the MoM.

**Table 5.3:** Run-time and memory requirement for the analysis of the irregular strip dipole array configuration of Figure 5.5 (a) using the MoM, DGFM, i-DGFM, CBFM and iterative Jacobi methods with 2 iterations, respectively, for 20 solution configurations.

| Method | Solution time | Memory usage |
|---|---|---|
| MoM | 18.64 sec | 67.87 MByte |
| DGFM ($\alpha^0$-weights) | 0.953 sec | 192.52 kByte |
| i-DGFM ($\boldsymbol{\alpha^1}$-weights) | 3.734 sec | 385.04 kByte |
| CBFM | 23.156 sec | 1.99 MByte |
| Jacobi method ($N_{it} = 2$) | 0.079 sec | 385.04 kByte |
| Calculating prim. + second. CBFs | (0.03 + 1.236) sec | 32.95 kByte + 593.16 kByte |

effects in addition to the global coupling solutions that are inherent to the infinite-array-type of assumption made in the DGFM. The improvement is based on the Jacobi decomposition of the block factorised MoM matrix which not only allows for the analysis of more concentrated localised excitation schemes – or more rapidly varying currents on the array elements – but also the calculation of embedded element patterns (EEPs) and off-broadside scan angles for phased array analysis. The derivation of an improved $\boldsymbol{\alpha}$ weight matrix also allows for arbitrary current distributions on array elements, which is a major improvement over the constant $\alpha$ coefficients used in the DGFM.

# 6

# Advanced applications of the DGFM and i-DGFM : hybrid extensions

In this Chapter, the DGFM will be hybridised with two existing domain decomposition methods, viz. the CBFM and the Numerical Green's Function (NGF), to formulate the so-called hybrid CBFM-DGFM and NGF-DGFM methods, respectively. By combining the DGFM and i-DGFM techniques with the CBFM and NGF methods, specific type of electrically large problems can be solved in a runtime efficient and memory economical manner.

With the hybrid CBFM-DGFM [55], large arrays consisting of multiple disjoint subarrays can be analysed. Active impedance matrices[1] are constructed for each of the *subarrays* from the block-partitioned CBFM reduced impedance matrix, introduced in Section 3.3.2, which account for the mutual coupling in the array environment. The weighting coefficients applied in the calculation of the active impedance matrices are deduced in a similar manner as those obtained in Section 4.5.7 where scattering from an FSS is considered. Runtime and memory usage scale efficiently for the hybrid CBFM-DGFM method as computational complexity is limited to that required for the analysis of a single subarray only.

For the hybrid NGF-DGFM [56], finite arrays will be analysed in the presence of arbitrary geometry, e.g. finite sized ground planes. The hybrid approach is based on the partitioned MoM scheme, viz. the NGF solution technique. The NGF solution is equal to the full-wave MoM solution, i.e. no approximation is made, as will be discussed in more detail in Section 6.2.1. The accuracy of the hybrid NGF-DGFM method is therefore only limited by the accuracy of applying the DGFM to the finite array, which can be enhanced significantly through the i-DGFM, as explained in Chapter 5.

The Chapter outline is as follows: in Section 6.1.1 the hybrid CBFM-DGFM will be formulated, followed by a test case in Section 6.1.2. In Section 6.2.1 the hybrid NGF-DGFM will be considered, as well as a discussion on the computational complexity of the method in Section 6.2.2. Various test cases will be considered in Section 6.2.3 for the NGF-DGFM, consisting of finite arrays in the presence of arbitrary structures. The Chapter is then concluded in Section 6.3.

[1] In [55], the active impedance matrices are referred to as scan impedance matrices and the hybrid CBFM-DGFM method is referred to as the CBFM-enhanced DGFM technique.

**Figure 6.1:** Example of the LOFAR [2] radio telescope. ©Top-Foto, Assen. as obtained on [2].

## 6.1 The hybrid CBFM-DGFM

The hybrid CBFM-DGFM is particularly suited to the analysis of large finite arrays configurations, consisting of subarrays. Examples of such subarrays are those used at LOFAR [2, 112] which is shown in Figure 6.1. LOFAR is the Low Frequency Array built for radio astronomy by ASTRON - the Netherlands Institute for Radio Astronomy. LOFAR was developed by ASTRON, together with other projects such as APERTIF and EMBRACE, to examine both aperture and focal plane array concepts. EMBRACE also forms part of SKA pathfinding activities [29, 113].

In [29] this class of antenna array consisting of subarray tiles was analysed using the CBFM, for which the reduced matrix equation (refer to (3.21) in Section 3.3.2) can become quite large. The approach followed here is to apply the DGFM, using $\alpha_{qp}$ weighting coefficients on the reduced matrix equation to efficiently analyse the interaction between the actively phase-steered subarrays.

### 6.1.1 Formulation

Consider the array configuration illustrated in Figure 6.2, that consists of $M$ identical subarrays, each of which consists of $K$ antenna elements, such as that used at LOFAR [2]. Applying the CBFM to the array of Figure 6.2 leads to the formulation of the reduced matrix equation, i.e. $\mathbf{Z}^{\mathrm{CBFM}}\mathbf{J}^{\mathrm{CBFM}} = \mathbf{V}^{\mathrm{CBFM}}$ as defined in (3.21), that can be expressed as follows,

$$\underbrace{\begin{bmatrix} \mathbf{Z}_{11} & \mathbf{Z}_{12} & \cdots & \mathbf{Z}_{1M} \\ \mathbf{Z}_{21} & \mathbf{Z}_{22} & \cdots & \mathbf{Z}_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{Z}_{M1} & \mathbf{Z}_{M2} & \cdots & \mathbf{Z}_{MM} \end{bmatrix}}_{\mathbf{Z}^{\mathrm{CBFM}}} \underbrace{\begin{bmatrix} \mathbf{J}_1 \\ \mathbf{J}_2 \\ \vdots \\ \mathbf{J}_M \end{bmatrix}}_{\mathbf{J}^{\mathrm{CBFM}}} = \underbrace{\begin{bmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \\ \vdots \\ \mathbf{V}_M \end{bmatrix}}_{\mathbf{V}^{\mathrm{CBFM}}}. \tag{6.1}$$

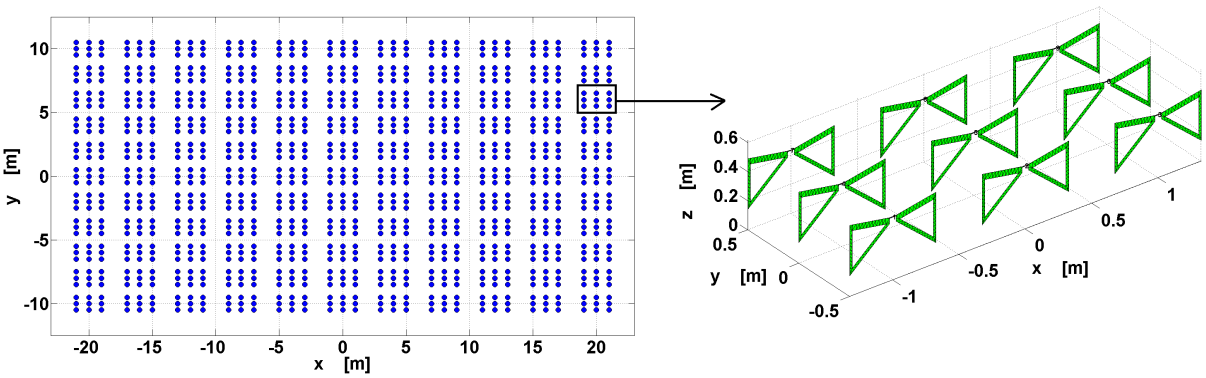



**Figure 6.2:** Antenna array consisting of $M$ disjoint subarrays, each with $K$ antenna elements.

In (6.1), a block partitioning scheme has been followed such that each of the matrix entries, $\mathbf{Z}_{pq}$, is a $(KP)^2 \times (KP)^2$ submatrix that accounts for the interaction between the $KP\times$CBFs on the subarrays $p$ and $q$, respectively ($P$ is the number of CBFs per antenna element).

Storing the matrix equation in (6.1) scales as $\mathcal{O}((M \times (K \times P)^2)^2)$ and solving the set of linear equations using LU decomposition scales as $\mathcal{O}((M \times (K \times P)^2)^3)$. The aforementioned can become prohibitive when the number of subarrays and/or the number of elements per subdomain increase.

As done for the DGFM in Section 4.1, assume that the CBFs between the subarrays $p$ and $q$ are identical except for a complex scaling factor, $\alpha_{pq}$, i.e.,

$$\begin{bmatrix}\mathbf{J}_{0q} & \mathbf{J}_{1q} & \dots & \mathbf{J}_{K'q}\end{bmatrix} = \alpha_{qp}\begin{bmatrix}\mathbf{J}_{0p} & \mathbf{J}_{1p} & \dots & \mathbf{J}_{K'p}\end{bmatrix}, \tag{6.2}$$

where each of the columns represent a CBF, with $K' = KP$, i.e. denoting the $(KP)$th CBF on each of the subarrays, respectively. The scaling coefficients, $\alpha_{pq}$, are calculated as follows,

$$\alpha_{pq} = \exp(-jk[x_{pq}\cos\phi_0 \sin\theta_0 + y_{pq}\sin\phi_0\sin\theta_0]), \tag{6.3}$$

where $(\theta_0, \phi_0)$ is the scan angle, and $x_{pq}$ and $y_{pq}$ are the $x$ and $y$ offset positions between the centre of the subarray tiles $p$ and $q$, respectively. This expression is similar to that derived for the FSS example in Section 4.5.7.

By substituting (6.2) in (6.1), and applying the $\alpha_{qp}$ weights defined in (6.3), the active CBF current distribution on subarray $p$ can be calculated as follows,

$$\begin{aligned}\mathbf{V}_p &= [\mathbf{Z}_{pp} + (\alpha_{qp}\mathbf{Z}_{pq} + \ldots + \alpha_{Mp}\mathbf{Z}_{pM})]\mathbf{J}_p \\ &= \left[\sum_{m=1}^{M}\alpha_{mp}\mathbf{Z}_{pm}\right]\mathbf{J}_p \;=\; \mathbf{Z}_p^{\text{scan}}(\theta_0,\phi_0)\mathbf{J}_1.\end{aligned} \tag{6.4}$$

where the term $\mathbf{Z}_p^{\text{scan}}$ is called the scan impedance matrix that accounts for both the self-coupling internal to the $p$th subarray, as well as the mutual coupling from the surrounding subarray tiles through the phase steering vector in (6.3). As noted for the DGFM in Section 4.1, the scan impedance matrices of the domains are different, i.e. $\mathbf{Z}_p^{\text{scan}} \neq \mathbf{Z}_q^{\text{scan}}$ for $p \neq q$. The resulting

current distribution on the subarrays $p$ and $q$ are therefore perturbed from our initial assumption in (6.2).

Solving each of the scan impedance matrix equations using LU decomposition is localised to the $(K \times P)^2$ unknowns associated with each of the $M$ subdomains being analysed. The run-time therefore scales as $\mathcal{O}(((K \times P)^2)^3)$. Likewise, the memory usage scales according to $\mathcal{O}(((K \times P)^2)^2)$.

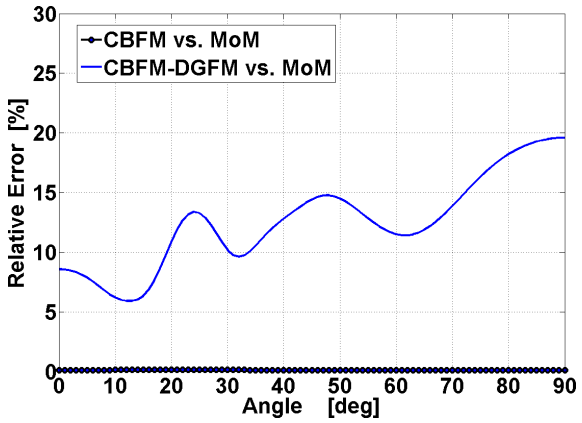
### 6.1.2 Test case

**Phased array of multiple disjoint subarrays**

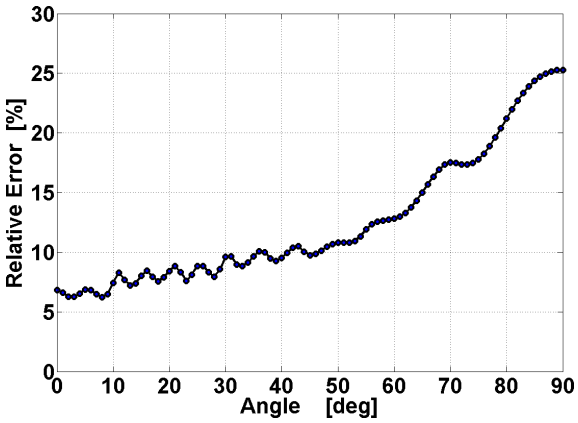
To quantify the numerical accuracy and computational complexity of the CBFM-enhanced DGFM, the array configuration illustrated in Figure 6.2 was analysed using the MoM, CBFM and hybrid CBFM-DGFM, respectively. The lowest level element that was used is a single polarised LOFAR bow-tie element [2] and the analysis frequency was set to $f = 170$ MHz. Two structures were considered, viz. an $9 \times 9$ and an $11 \times 11$ array configuration with the element in each case consisting of a $3 \times 3$ subarray tile of bow-tie antennas. The total number of elementary RWG type basis functions used to discretise the individual array elements are 14,013 for the $9 \times 9$ array and 188,397 for the $11 \times 11$ array configurations, respectively. A total of 9 CBFs are used per subarray element (i.e. the individual bow-tie antenna), resulting in 81 CBFs for each of the $3 \times 3$ subarrays. The aforementioned is for the case where both primary and secondary CBFs are used. The number of unknowns for the reduced matrix equation of the $9 \times 9$ array is 6,561 and for the $11 \times 11$ array increases to 9,801. For each case, the number of unknowns associated with the scan impedance matrix equations formulated with the DGFM is 81 (equal to the number of CBFs per subarray). The element and subarray spacing is shown in Figure 6.2.

The relative error norm percentage, $\epsilon_{\%}$, defined in (4.18) was used as a figure of merit for rigorously comparing the accuracy between the results. For this particular test case, $\{I_n\}$ is the set of expansion coefficients calculated with the CBFM or hybrid CBFM-DGFM, respectively, and $\{I_n^{\text{ref}}\}$ is the reference solution. For the $9 \times 9$ array $\{I_n^{\text{ref}}\}$ is the full-wave MoM solution, that verifies the CBFM results that are used as reference for the larger $11 \times 11$ array. The results are illustrated in Figure 6.3(a) and (b), respectively. In Figure 6.3(c) and (d) the $H$-plane ($ZY$-plane) gain patterns calculated at broadside and $\theta_0 = 60°$, respectively, for the $11 \times 11$ array are shown. The computational complexity of the hybrid CBFM-DGFM compared to the CBFM for the $11 \times 11$ array is summarised in Table 6.1. The relative error norm percentage ($\epsilon_{\%}$) obtained with the hybrid CBFM-DGFM remains below 15% when compared with the full wave MoM for the smaller $9 \times 9$ array for scan angles up to $\theta_0 = 70°$. For the larger array, the same applies for scan angles up to $\theta_0 = 65°$.

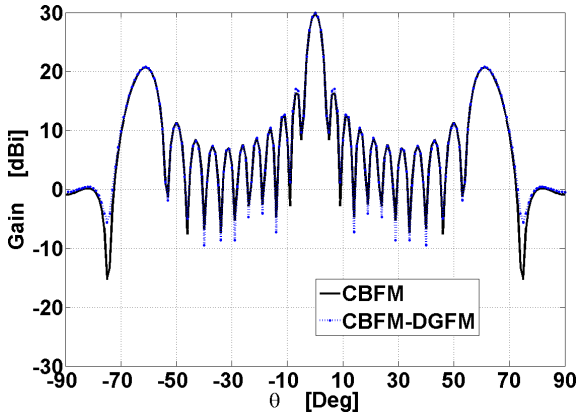
For both arrays the relative error norm percentage ($\epsilon_{\%}$) increases for larger scan angles. A possible reason for this, is that for larger scan angles the infinite array type assumption made in the DGFM, i.e. that the currents on the domains are equal, is no longer valid. By considering $\boldsymbol{\alpha}_{qp}$ weighting coefficients as done for the i-DGFM in Chapter 5, with the coefficients derived from solving the reduced impedance matrix equations of the CBFM, the accuracy of the hybrid CBFM-DGFM can possibly be extended to support also more rapidly varying current



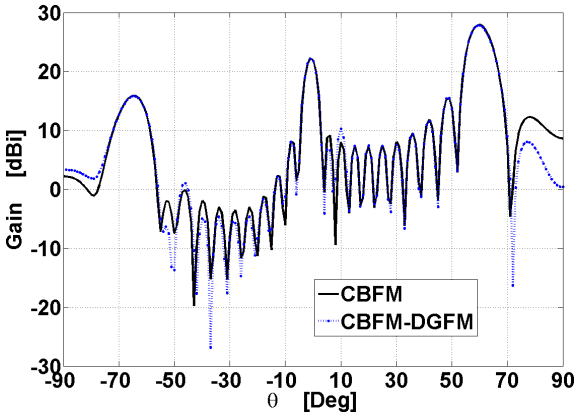

(a) Relative error norm percentage $\epsilon_{\%}$ of the $9 \times 9$ array comparing the CBFM and CBFM-enhanced DGFM solutions to the full-wave MoM solution.




(b) Relative error norm percentage $\epsilon_{\%}$ of the $11 \times 11$ array comparing the CBFM and CBFM-enhanced DGFM solutions.




(c) Total $H$-plane gain patterns calculated with the CBFM and CBFM-enhanced DGFM (dB) for $\theta_0 = 0°$.




(d) Total $H$-plane gain patterns calculated with the CBFM and CBFM-enhanced DGFM (dB) for $\theta_0 = 60°$.

**Figure 6.3:** Applying the CBFM-enhanced DGFM to a large array configuration displayed in Figure 6.2

**Table 6.1:** Run-time and memory requirement for the analysis of the $11 \times 11$ array configuration of Figure 6.2 on an Intel Xeon CPU E5640 at 2.67 GHz clock rate using the CBFM and CBFM-enchanced DGFM respectively. All results are obtained for the active array environment, i.e. where all the elements are excited equally and simultaneously

| **Method:** | CBFM | CBFM-DGFM |
|---|---|---|
| Time for matrix solution | 77.68 sec | 2.19 sec |
| Memory usage (whole solution) | 1.78 GByte | 126.56 kByte |

distributions on electrically larger, multi-mode subarray tiles. The far field results compare well for scan angles of $\theta_0 = 0°$ and $\theta_0 = 60°$, respectively. A significant saving is gained in both run-time and memory usage when using the hybrid CBFM-DGFM, as we analyse only a single subarray at a time.

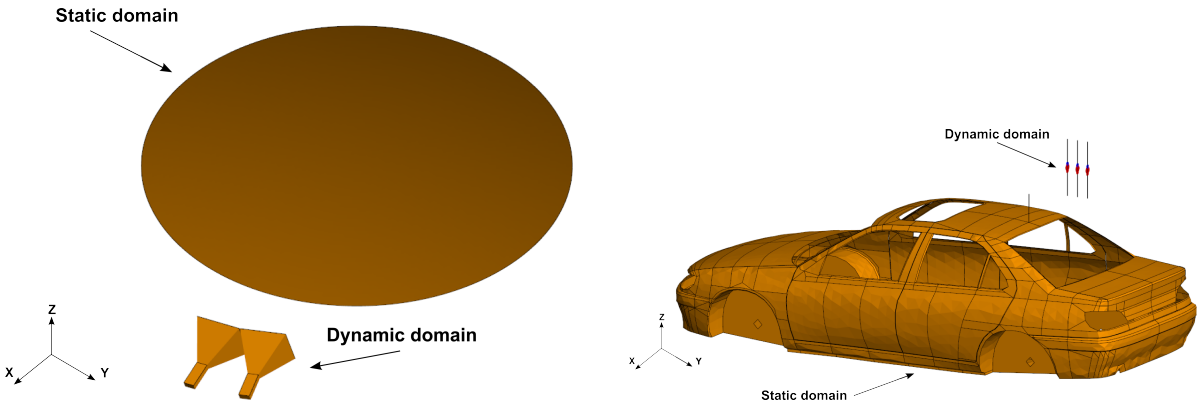



**Figure 6.4:** (a) A two element horn cluster illuminating a parabolic dish reflector antenna, and (b) a dipole array near the side of a vehicle with a monopole antenna mounted on the roof.

## 6.2 The hybrid NGF-DGFM

In this Section, finite arrays will be analysed in the presence of general structures, e.g. finite ground planes. A hybrid approach between a partitioned MoM scheme called the Numerical Green's Function (NGF) solution technique and the DGFM is presented, viz., the hybrid NGF-DGFM[2].

The NGF is well suited to electromagnetic problems for which there is a predominant static[3] part, the formulation of which is based on that implemented in the Numerical Electromagnetics Code (NEC) [104]. The NGF exploits the fact that the LU factorisation of the static interaction matrix, i.e., the impedance matrix for the static domain, can be calculated once, stored to an external file and re-used for various dynamic sub-domain configurations. The NGF technique has been extended in this Section to support as a dynamic sub-domain, the active impedance matrices that are derived for the finite array using the DGFM or i-DGFM, as formulated in Chapter 4 and Chapter 5.

The NGF method is not an approximation, and yields the same solution as obtained by the full-wave MoM. The accuracy of the hybrid NGF-DGFM is therefore primarily determined by the accuracy of the solution of the finite array calculated with the DGFM, or i-DGFM.

Consider the geometry that is illustrated in Figure 6.4 (a) and (b), that shows finite arrays in the presence of other "static" geometry. The analysis and design of the array geometry typically entails that the array changes between simulations, e.g. different array spacing, element type, feed-networks, etc., while the remainder of the problem remains unchanged. The focus of this Section is to efficiently recalculate the solution while ensuring that mutual coupling between the "dynamic" and the "static" domains is accounted for as accurately as possible.

[2]In [56] the hybrid NGF-DGFM is referred to as the NGF-enhanced DGFM.
[3]In this context, *static* refers to unchanging geometry, and not *static fields* as encountered in electrostatics.

### 6.2.1 Formulation

To analyse finite array problems such as that illustrated in Figure 6.4 (a) and (b) with the hybrid NGF-DGFM, we begin by block partitioning the MoM matrix equation, $\mathbf{ZJ} = \mathbf{V}$, as,

$$\begin{bmatrix} \mathbf{Z}_{\text{ss}} & \mathbf{Z}_{\text{sd}} \\ \mathbf{Z}_{\text{ds}} & \mathbf{Z}_{\text{dd}} \end{bmatrix} \begin{bmatrix} \mathbf{J}_{\text{s}} \\ \mathbf{J}_{\text{d}} \end{bmatrix} = \begin{bmatrix} \mathbf{V}_{\text{s}} \\ \mathbf{V}_{\text{d}} \end{bmatrix}, \tag{6.5}$$

where $\mathbf{Z}_{\text{ss}}$ and $\mathbf{Z}_{\text{dd}}$ are the self-interaction matrices of the static and dynamic subdomains, respectively, and $\mathbf{Z}_{\text{ds}}$ and $\mathbf{Z}_{\text{sd}}$ are the mutual coupling matrices between the domains. The terms $\mathbf{J}_{\text{s}}$ and $\mathbf{J}_{\text{d}}$ are the unknown expansion coefficients associated with the static and dynamic domains. Likewise, the partitioned excitation vectors associated with the static and dynamic domains are $\mathbf{V}_{\text{s}}$ and $\mathbf{V}_{\text{d}}$, respectively. Typically, the number of unknowns in the static domain is significantly larger than that of the dynamic domain, i.e. $N_s >> N_d$.

The matrix equation in (6.5) can also be written as the following linear equations,

$$\mathbf{Z}_{\text{ss}}\mathbf{J}_{\text{s}} + \mathbf{Z}_{\text{sd}}\mathbf{J}_{\text{d}} = \mathbf{V}_{\text{s}} \tag{6.6}$$

$$\mathbf{Z}_{\text{ds}}\mathbf{J}_{\text{s}} + \mathbf{Z}_{\text{dd}}\mathbf{J}_{\text{d}} = \mathbf{V}_{\text{d}}. \tag{6.7}$$

The key to solving (6.6) and (6.7) is to rewrite $\mathbf{J}_{\text{s}}$ in (6.6) as follows,

$$\mathbf{J}_{\text{s}} = \mathbf{Z}_{\text{ss}}^{-1}\mathbf{V}_{\text{s}} - \mathbf{Z}_{\text{ss}}^{-1}(\mathbf{Z}_{\text{sd}}\mathbf{J}_{\text{d}}). \tag{6.8}$$

By substituting (6.8) in (6.7), we can derive an expression for the current on the dynamic domain, i.e. $\mathbf{J}_{\text{d}}$, as follows,

$$\mathbf{J}_{\text{d}} = (\mathbf{Z}_{\text{dd}} - \underbrace{\mathbf{Z}_{\text{ds}}\mathbf{Z}_{\text{ss}}^{-1}\mathbf{Z}_{\text{sd}}}_{\text{correction / coupling term}})^{-1}(\mathbf{V}_{\text{d}} - \mathbf{Z}_{\text{ds}}\mathbf{Z}_{\text{ss}}^{-1}\mathbf{V}_{\text{s}}). \tag{6.9}$$

In (6.8) it is clear that the current on the static domain, that is induced by the primary excitation, $\mathbf{V}_{\text{s}}$, is the same as the primary CBF as discussed in Section 3.3.2. The contribution from the dynamic domain current, i.e. $\mathbf{J}_{\text{d}}$, is then added via the coupling matrix, i.e., $\mathbf{Z}_{\text{sd}}$, which is again analogous to a secondary induced CBF.

The same can be reasoned for the current on the dynamic domain, $\mathbf{J}_{\text{d}}$, in (6.9), that is separated into a primary component induced by $\mathbf{V}_{\text{d}}$, and a secondary component that is the result of the primary current induced on the static domain, i.e., $\mathbf{Z}_{\text{ss}}^{-1}\mathbf{V}_{\text{s}}$ in (6.8). In (6.9) the self-interaction matrix of the dynamic domain, $\mathbf{Z}_{\text{dd}}$, is modified to account also for the mutual coupling from the static domain, with the correction/coupling term, $\mathbf{Z}_{\text{ds}}\mathbf{Z}_{\text{ss}}^{-1}\mathbf{Z}_{\text{sd}}$. For the DGFM (and i-DGFM), a similar correction/coupling term will be added to the active impedance matrix calculations as explained later in this subsection.

It should be noted, that if the full-wave MoM is used to model *both* the static and the dynamic domains, then calculating $\mathbf{J}_{\text{s}}$ and $\mathbf{J}_{\text{d}}$ according to (6.8) and (6.9) yields exactly the same answer as the full-wave MoM solution, i.e., $\mathbf{J} = \mathbf{Z}^{-1}\mathbf{V}$.

Based on the derivation of (6.8) and (6.9), consider modifying the impedance matrix terms in (6.5) as follows,

$$\begin{aligned}
\mathbf{Z}_{\mathrm{ss}} &\rightarrow \mathbf{Z}'_{\mathrm{ss}} = (\mathbf{Z}_{\mathrm{ss}})_l(\mathbf{Z}_{\mathrm{ss}})_u, \\
\mathbf{Z}_{\mathrm{sd}} &\rightarrow \mathbf{Z}'_{\mathrm{sd}} = \mathbf{Z}_{\mathrm{ss}}^{-1}\mathbf{Z}_{\mathrm{sd}}, \\
\mathbf{Z}_{\mathrm{ds}} &\rightarrow \mathbf{Z}'_{\mathrm{ds}} = \mathbf{Z}_{\mathrm{ds}}, \\
\mathbf{Z}_{\mathrm{dd}} &\rightarrow \mathbf{Z}'_{\mathrm{dd}} = \mathbf{Z}_{\mathrm{dd}} - \mathbf{Z}_{\mathrm{dd}}\mathbf{Z}'_{\mathrm{sd}}.
\end{aligned} \tag{6.10}$$

The current on the static and dynamic domains, i.e. $\mathbf{J}_{\mathrm{s}}$ and $\mathbf{J}_{\mathrm{d}}$, can then be expressed in terms of (6.10) as follows,

$$\mathbf{J} = \begin{Bmatrix} \mathbf{J}_{\mathrm{s}} \\ \\ \mathbf{J}_{\mathrm{d}} \end{Bmatrix} = \begin{Bmatrix} \mathbf{Z}_{\mathrm{ss}}^{-1}\mathbf{V}_{\mathrm{s}} - \mathbf{Z}'_{\mathrm{sd}}\mathbf{J}_{\mathrm{d}} \\ \\ (\mathbf{Z}'_{\mathrm{dd}})^{-1}(\mathbf{V}_{\mathrm{d}} - \mathbf{Z}_{\mathrm{ds}}\mathbf{Z}_{\mathrm{ss}}^{-1}\mathbf{V}_{\mathrm{s}}) \end{Bmatrix} \tag{6.11}$$

In (6.11), the inverse $\mathbf{Z}_{\mathrm{ss}}^{-1}$ is not calculated explicitly each time the dynamic domain geometry changes. Instead, the LU factored form of the static interaction matrix, i.e. $\mathbf{Z}'_{\mathrm{ss}} = (\mathbf{Z}_{\mathrm{ss}})_l(\mathbf{Z}_{\mathrm{ss}})_u$ is used. The efficiency of (6.11) is that the LU factorisation of the static interaction matrix, $\mathbf{Z}'_{\mathrm{ss}}$, is calculated once, stored to a *.ngf file and reused for any dynamic domain. When a solution is required for any dynamic sub-domain configuration, the matrices $\mathbf{Z}_{\mathrm{sd}}$, $\mathbf{Z}_{\mathrm{ds}}$ and $\mathbf{Z}_{\mathrm{dd}}$ in (6.5) are evaluated and factored according to (6.10).

Vendor optimised implementations of the Basic Linear Algebra Subprograms (BLAS) [114] and the Linear Algebra Package (LAPACK) [115] are used for the operations required in the above mentioned matrix operations. The BLAS provides a number of basic linear algebra routines, including vector and matrix multiplication which is used as the basis for a number of more complex linear algebra functions such as matrix factorisations implemented in LAPACK. The benefit of using these routines, is that the factorisations in (6.10) can be done *in-place*, i.e. without allocating additional memory. The memory requirement for the NGF therefore scales according to $\mathcal{O}((N_s + N_d)^2)$, which is identical to that of the full-wave MoM solution, should the latter be used for *both* the static and dynamic domains. The NGF method presented here for the full-wave MoM is also useful when considering array element design, e.g. when investigating only minor modifications to a certain part of the geometry while the rest remains unchanged.

When the array geometry that is considered becomes complex in terms of the type or number of elements, then the MoM may resort in very long run-times and high memory usage when it is applied to the dynamic domain. The approach followed for the hybrid NGF-DGFM, is to use the DGFM method to obtain an *approximate* matrix equation for the dynamic domain, in the form of a *modified* active impedance matrix equation for each element. To explain the approach, consider a general finite array geometry, such as that in Figure 6.4 (a) or (b) with $M$ arbitrarily spaced elements. A matrix equation, $\mathbf{Z}_{\mathrm{dd}}\mathbf{J}_{\mathrm{d}} = \mathbf{V}_{\mathrm{d}}$, can be formulated for the antenna array

using the DGFM, as follows,

$$\begin{bmatrix} \mathbf{Z}_1^{\text{act}} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{Z}_2^{\text{act}} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{Z}_M^{\text{act}} \end{bmatrix} \begin{bmatrix} \mathbf{J}_1 \\ \mathbf{J}_2 \\ \vdots \\ \mathbf{J}_M \end{bmatrix} = \begin{bmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \\ \vdots \\ \mathbf{V}_M \end{bmatrix}, \tag{6.12}$$

which is a block matrix containing the active impedance matrices on the diagonal, i.e. $\mathbf{Z}_n^{\text{act}} = \sum_{m=1}^{M} \alpha_{mn} \mathbf{Z}_{nm}$ when using the scalar $\alpha$ weighting coefficients introduced in Chapter 4, or $\mathbf{Z}_n^{\text{act}} = \sum_{m=1}^{M} \boldsymbol{\alpha}_{mn} \circ \mathbf{Z}_{nm}$ when using the improved $\boldsymbol{\alpha}$ weight matrix as derived in Chapter 5. It should be noted that the goal is not to store the whole $\mathbf{Z}_{\text{dd}}$ in (6.12), but rather to incorporate a correction factor for each of the active impedance matrices, that represents the coupling from the static domain, as done in (6.9). The memory requirement for the dynamic domain, will therefore still be limited to that of a single array element.

Following the block-partitioning of the NGF, one can replace the dynamic domain matrix, $\mathbf{Z}_{\text{dd}}$ in (6.5) with the sparse active impedance matrix for the array in (6.12) as follows,

$$\left[\begin{array}{c|ccc} \mathbf{Z}_{\text{ss}} & \mathbf{Z}_{\text{s}1} & \mathbf{Z}_{\text{s}2} & \cdots & \mathbf{Z}_{\text{s}M} \\ \hline \mathbf{Z}_{1\text{s}} & \mathbf{Z}_1^{\text{act}} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{Z}_{2\text{s}} & \mathbf{0} & \mathbf{Z}_2^{\text{act}} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{Z}_{M\text{s}} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{Z}_M^{\text{act}} \end{array}\right], \tag{6.13}$$

where the submatrix, $[\mathbf{Z}_{\text{s}1} \quad \mathbf{Z}_{\text{s}2} \quad \cdots \quad \mathbf{Z}_{\text{s}M}]$, is $\mathbf{Z}_{\text{sd}}$ in (6.5) and represents the coupling from the array environment on the static domain. Likewise, the submatrix $[\mathbf{Z}_{1\text{s}} \quad \mathbf{Z}_{2\text{s}} \quad \cdots \quad \mathbf{Z}_{M\text{s}}]^T$ is $\mathbf{Z}_{\text{ds}}$ in (6.5) and represents coupling from the static domain on the array elements.

The current on the array elements, $\mathbf{J}_{\text{d}}$, can now be written for each of the array elements $1, \ldots, M$ as,

$$\mathbf{J}_d = \begin{Bmatrix} \mathbf{J}_1 \\ \mathbf{J}_2 \\ \vdots \\ \mathbf{J}_M \end{Bmatrix} = \begin{Bmatrix} \left[\sum_{m=1}^{M} \alpha_{m1}(\mathbf{Z}_{1m} - \mathbf{Z}_{1\text{s}}\mathbf{Z}'_{\text{s}1})\right]^{-1} (\mathbf{V}_1 - \mathbf{Z}_{1\text{s}}\mathbf{Z}_{\text{ss}}^{-1}\mathbf{V}_{\text{s}}) \\ \left[\sum_{m=1}^{M} \alpha_{m2}(\mathbf{Z}_{2m} - \mathbf{Z}_{2\text{s}}\mathbf{Z}'_{\text{s}2})\right]^{-1} (\mathbf{V}_2 - \mathbf{Z}_{2\text{s}}\mathbf{Z}_{\text{ss}}^{-1}\mathbf{V}_{\text{s}}) \\ \vdots \\ \left[\sum_{m=1}^{M} \alpha_{mM}(\mathbf{Z}_{Mm} - \mathbf{Z}_{M\text{s}}\mathbf{Z}'_{\text{s}M})\right]^{-1} (\mathbf{V}_M - \mathbf{Z}_{M\text{s}}\mathbf{Z}_{\text{ss}}^{-1}\mathbf{V}_{\text{s}}) \end{Bmatrix}, \tag{6.14}$$

with $\mathbf{Z}'_{sn} = \mathbf{Z}_{\text{ss}}^{-1}\mathbf{Z}_{sn}$ for $n = 1, \ldots, M$ a correction term for the active impedance matrix equation of each array element, that represents mutual coupling between the dynamic and static

environment. This may be written more generally for the $n$th element, as follows,

$$\mathbf{Z}_n^{\text{act}} = \sum_{m=1}^{M} \alpha_{mn}(\mathbf{Z}_{1m} - \mathbf{Z}_{ns}\mathbf{Z}'_{sn}) \tag{6.15}$$

by using the $\alpha$ weighting coefficient, or as follows when using the improved $\boldsymbol{\alpha}$ coefficient matrix introduced in Chapter 5.

$$\mathbf{Z}_n^{\text{act}} = \sum_{m=1}^{M} \boldsymbol{\alpha}_{\mathbf{mn}} \circ (\mathbf{Z}_{1m} - \mathbf{Z}_{ns}\mathbf{Z}'_{sn}) \tag{6.16}$$

In (6.14) we also note that the exicitation vectors for solving each of the localised active matrix equations are modified with the terms $\mathbf{Z}_{ns}\mathbf{Z}_{\text{ss}}^{-1}\mathbf{V}_{\text{s}}$ for $n = 1, \ldots, M$, that represents the secondary induced current on the array element, as a result of the primary excitation on the static domain (if $\mathbf{V} \neq \mathbf{0}$).

The current on the static domain, $\mathbf{J}_{\text{s}}$, can then be calculated from (6.8) as follows,

$$\mathbf{J}_{\text{s}} = \mathbf{Z}_{\text{ss}}^{-1}\mathbf{V}_{\text{s}} - \mathbf{Z}'_{sn}\mathbf{J}_n \text{ for } n = 1, \ldots, M \tag{6.17}$$

where $\mathbf{Z}'_{sn} = \mathbf{Z}_{\text{ss}}^{-1}\mathbf{Z}_{sn}$ can be seen as an operator that couples the current calculated for the $n$th array element, using (6.14), onto the static domain. In the following Section, the computational complexity of the hybrid NGF-DGFM will be reviewed.

### 6.2.2 Computional complexity

In each case where $\mathbf{Z}_{\text{ss}}^{-1}$ is applied in Section 6.2.1, the factorisation, $\mathbf{Z}'_{\text{ss}}$, can be used. In addition, this factorisation can also be stored and read as needed from a file. This saves a lot of computational effort when constructing the MoM matrix, that reduces from $\mathcal{O}((N_s + N_d)^2)$ to $\mathcal{O}(2N_sN_d + N_d^2)$, where $N_s$ and $N_d$ are the number of basis functions associated with the static and dynamic domains, respectively. The cost of factoring $\mathbf{Z}_{\text{ss}}$ to obtain $\mathbf{Z}'_{\text{ss}}$ scales as $\mathcal{O}(N_s^3)$ and the memory usage for storing this factorisation scales as $\mathcal{O}(N_s^2)$.

The calculation of the current on the array, i.e., $\mathbf{J}_{\text{d}}$, is dominated by the calculations of the active impedance matrices for the domains in (6.14). This cost increases from $\mathcal{O}(M^2 \times N_i^2)$, as noted in Section 5.3, to $\mathcal{O}(M^2 \times N_i^2 + M^2 \times N_i^2 \times N_{\text{s}})$. In the aforementioned $M$ is the number of array elements and $N_i$ the number of basis functions for each element. The additional term that is added to the complexity, stems from the $M^2$ multiplications involving the $\alpha$ weighting coefficients to each of the $\mathbf{Z}_{ns}\mathbf{Z}'_{sn}$ terms. The matrix-matrix multiplication, $\mathbf{Z}_{ns}\mathbf{Z}'_{sn}$, scales as $\mathcal{O}(N_i^2 \times N_{\text{s}})$. Calculating $\mathbf{Z}'_{sn} = \mathbf{Z}_{\text{ss}}^{-1}\mathbf{Z}_{sn}$, can be done by using the LU-factored matrix, $\mathbf{Z}'_{\text{ss}}$, followed by multiple backward substitutions involving the columns of $\mathbf{Z}_{sn}$. Obtaining the current on the static domain, $\mathbf{J}_{\text{s}}$ in (6.17), scales as $\mathcal{O}(N_{\text{s}} \times N_i)$ as a result of the matrix-vector multiplication, $\mathbf{Z}'_{sn}\mathbf{J}_n$.

The memory usage for the hybrid NGF-DGFM scales as $\mathcal{O}(N_{\text{s}}^2 + (N_{\text{s}} \times N_i) + N_i^2)$ which is significantly less than that associated with the full-wave solution of the problem, i.e., $\mathcal{O}((N_s + M \times N_i)^2)$.

### 6.2.3 Test cases

In this section, the hybrid NGF-DGFM method will be applied to array geometries in the presence of different structures to illustrate the accuracy as well as the computational benefit of the technique.

#### Dipole array above finite ground plane

The hybrid NGF-DGFM formulation was applied to the 5 × 1 array of strip dipole antennas above a finite ground plane, as illustrated in Figure 6.5 (a). The height of the antennas above the ground plane, $h$, was varied from $0.1\lambda$ to $5\lambda$. The simulated far field was calculated along $\phi = 90°$ and $0° \leq \theta \leq 90°$ and is illustrated in Figure 6.5 (b). Likewise, the near field patterns were calculated by the MoM and the hybrid NGF-DGFM respectively for $h = 0.1\lambda$ and $h = 5.0\lambda$, along the $X$-axis of the array and is illustrated in Figure 6.5 (c). In each case, the i-DGFM with the $\boldsymbol{\alpha}$ matrix were used. The results obtained by the hybrid NGF-DGFM technique compare well to that obtained with the full-wave MoM, even for a closely coupled spacing of $h = 0.1\lambda$.

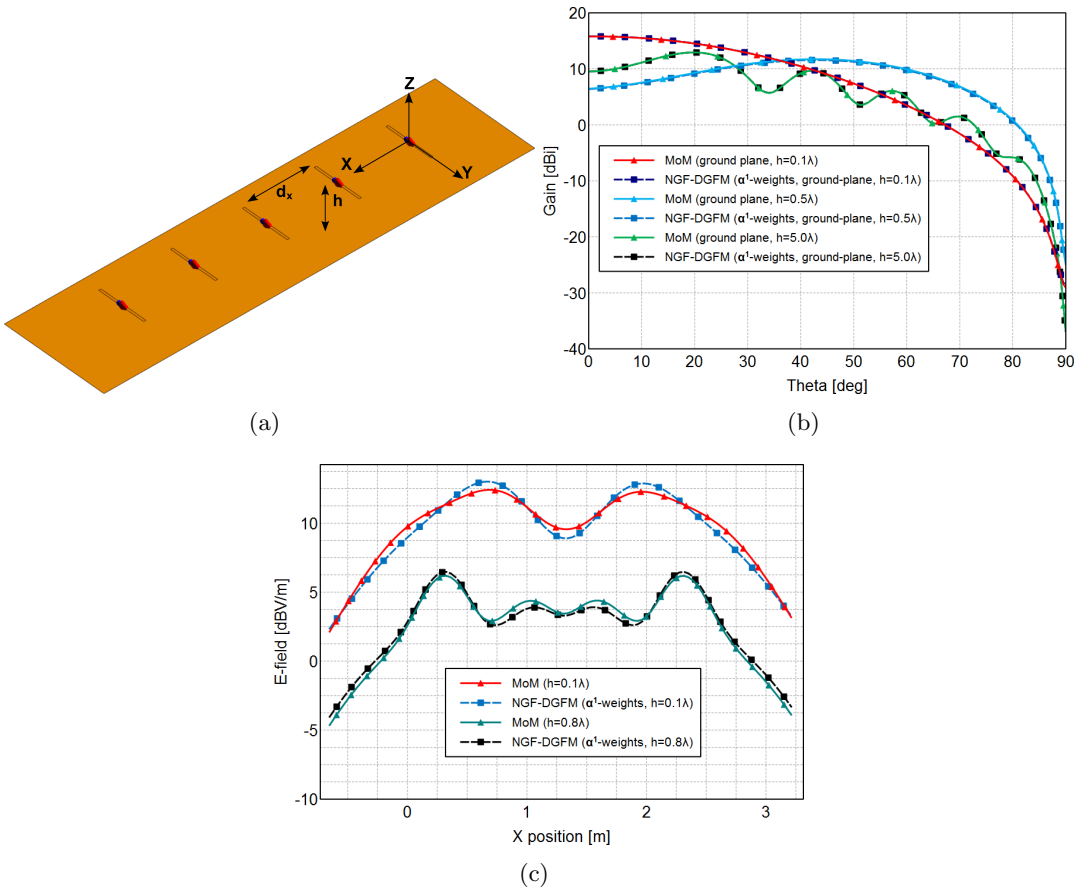



**Figure 6.5:** (a) (a) A 5 × 1 strip dipole antenna array above a finite ground plane. (b) Comparison of the far field gain patterns, measured along $\phi = 90°$ and $0° \leq \theta \leq 90°$, and (b) the near field patterns, measured along the $X$-axis at $z = \lambda$ and $y = 0$ for different spacing between the antenna array and the ground plane.

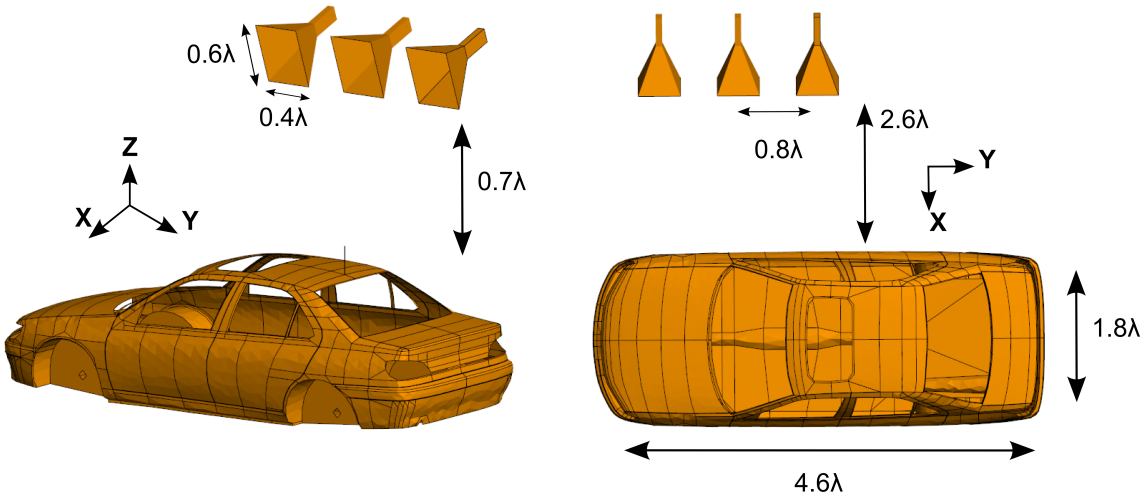



**Figure 6.6:** Three horn antenna illuminating a vehicle.

### Antenna array near a vehicle

The previous example illustrated the accuracy of the hybrid NGF-DGFM approach but was not representative of the computational efficiency thereof. In Figure 6.6, a more complex example is shown, consisting of a three element horn cluster illuminating a vehicle. The DGFM with $\alpha$ weighting coefficients was used in the analysis. The vehicle is also excited by a monopole on the roof, such that $\mathbf{V}_s \neq \mathbf{0}$. The analysis frequency is set to $f = 900$ MHz and the total number of RWG basis functions is 17,278 of which $11,212$ is allocated to the vehicle and $6,066$ to the finite array, i.e. $2,022$ per horn array element.

The computational complexity of the various phases of the hybrid NGF-DGFM and the MoM is summarised in Table 6.2. The results were obtained using MATLAB on a Windows machine with an Intel Core2 Quad processor (operating at 2.66 GHz) with 8 GByte of RAM.

The benefit of storing the factorisation, $\mathbf{Z}'_{\text{ss}}$, to a file and reusing it for subsequent runs, is clearly evident in Table 6.2. The runtime cost for the full-wave MoM is not added to the results, due to memory limitations imposed by MATLAB. The NGF-DGFM, being a more memory efficient method as explained in Section 6.2.2, was able to solve the problem with the runtime and memory usage as shown.

A relative error norm percentage (as defined in (4.18)) of $\epsilon_{\%} = 4.27\%$ was achieved when comparing the NGF-DGFM to the full-wave MoM solution in FEKO.

**Table 6.2:** Runtime and memory requirement for the analysis of the horn cluster and vehicle configuration in Figure 6.6 using the MoM and the hybrid NGF-DGFM, respectively.

| Method | Solution time | Memory usage |
|---|---|---|
| Factoring $\mathbf{Z}_{\text{ss}}$ and saving to disk | 6.9 min | 1.87 GByte |
| Reading the factorisation, $\mathbf{Z}'_{\text{ss}}$, from disk | 51.02 sec | 1.87 GByte |
| MoM | —- hr | 2.224 GByte |
| NGF-DGFM ($\alpha^0$-weights) | 3.52 hr | 1.87 GByte |

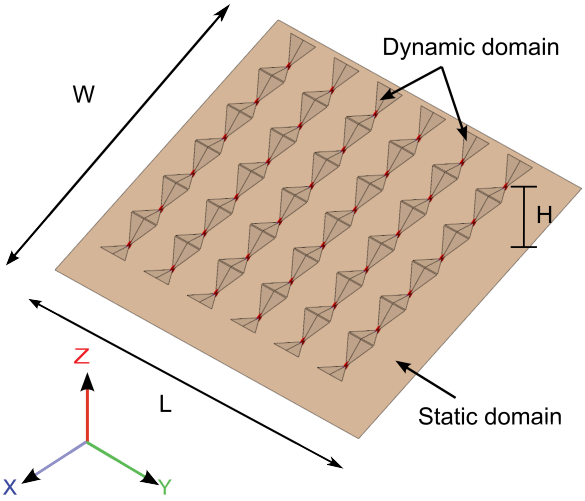
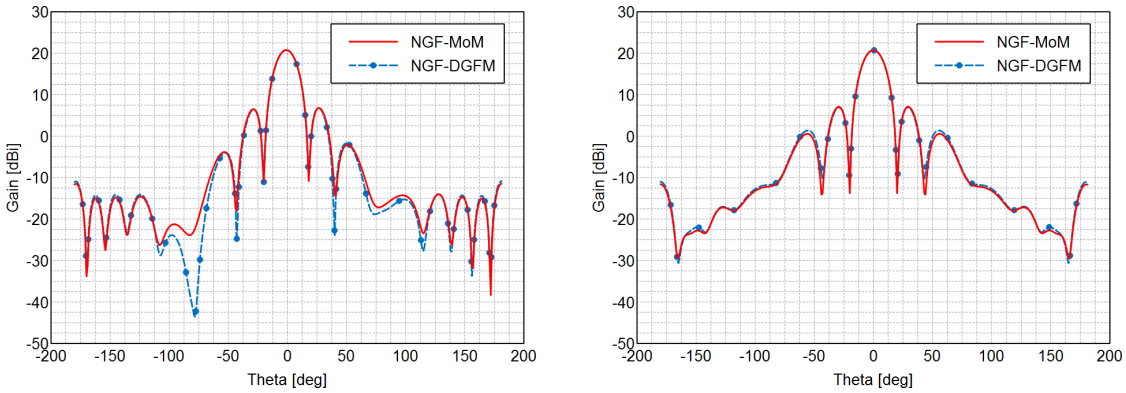



**Figure 6.7:** A 36 element bow-tie array above a finite ground plane.




(a) (b)

**Figure 6.8:** (a) The far field gain measured in the $ZX$-plane, and (b) the $ZY$-plane, for the 36 element bow-tie array, as shown in Fig. 6.7.

### Bow-tie antenna array above a finite ground plane

Consider the $6 \times 6$ bow-tie array geometry shown in Figure 6.7 that is discretised with 6,449 RWG basis functions of which 3,605 are associated with the finite ground plane and 2,844 with the bow-tie array. For this particular case, the ratio between the number of unknowns in the static and dynamic domain is more or less equal and results in a poor scaling for the NGF method (referred to as the NGF-MoM in this subsection) when compared to the MoM. An initial investigation indicates that the poor performance is associated with calculating the term $\mathbf{Z}'_{\text{sd}} = \mathbf{Z}_{\text{ss}}^{-1}\mathbf{Z}_{\text{sd}}$ in (6.10), i.e., the multiple backward substitutions that are applied to the LU-factored $\mathbf{Z}_{\text{ss}}$. The NGF-MoM will therefore be compared to the NGF-DGFM in this example.

The far field gain pattern was calculated along the $E$-plane (the $ZX$-plane) and the $H$-plane (the $ZY$-plane), for the case where all the elements are excited with unity amplitude and zero phase. The results are illustrated in Fig. 6.8(a) and (b). The runtime and memory usage of the NGF-MoM and NGF-DGFM are summarised in Table 6.3 and pertains to the case where the

**Table 6.3:** Runtime and memory requirement for the analysis of the 36 element bow-tie array configuration in Fig. 6.7 using the NGF-MoM and NGF-enhanced DGFM, respectively.

| Method | Runtime | Memory |
|---|---|---|
| NGF-MoM | 208.07 sec | 634.61 MByte |
| NGF-DGFM | 226.14 sec | 198.30 MByte |

NGF is being read from disk.

Excellent agreement can be observed between the far field gain patterns for the various cases. The runtime associated with the NGF-DGFM is slightly larger than that of the NGF-MoM for this example. A big improvement in memory usage can however be seen for the NGF-DGFM compared to that of the NGF-MoM, which is the result of working with the active impedance matrices for the array elements, and not a dense matrix to model the entire array environment (as is the case with the NGF-MoM). The memory usage associated with the NGF-DGFM is dominated by storing the LU-factored matrix of the static domain.

The results were calculated for this example using the DGFM with the scalar $\alpha$ weighting coefficients, although the improved $\boldsymbol{\alpha}$ weight matrix, as discussed in Chapter 5, can also be used. This should yield similar computational results with improved accuracy.

## 6.3 Conclusion

Two advanced applications for the DGFM and i-DGFM were considered in this Chapter, viz., that of analysing large arrays consisting of subarray tiles using the hybrid CBFM-DGFM approach and also the analysis of finite arrays in the presence of arbitrary structures with the hybrid NGF-DGFM. The hybrid NGF-DGFM can be used with both the DGFM and i-DGFM.

The hybrid CBFM-DGFM, presented in Section 6.1, provides an easy and efficient approach for analysing large finite antenna arrays consisting of electrically large actively phased-steered subarrays. The approach is based on a perturbation from the infinite array solution, as introduced by the DGFM, and is focussed on solving localised matrix equations formulated for each subarray respectively. Mutual coupling between the subarrays are considered with the formulation of a scan impedance matrix for each domain. The results presented for a test case in Section 6.1.2 showed that the accuracy of the CBFM-DGFM deteriorates for larger scan-angles, which can possibly be improved through the use of $\boldsymbol{\alpha}$ weighting coefficients as derived with the i-DGFM.

The hybrid NGF-DGFM solution technique discussed in Section 6.2, present a means of simulating finite array geometry in the presence of other structures, e.g. finite sized ground planes. The NGF is a full-wave domain decomposition approach and is particularly useful for problems that contain a large static region and a smaller changing dynamic part. In the hybrid NGF-DGFM, the dynamic domain is formed by the finite antenna array and is modelled using the DGFM (or i-DGFM). To account for mutual coupling between the static and the dynamic domain, a correction term is introduced for each of the active impedance matrices, as discussed in Section 6.2.1. Various test cases were presented that illustrate good accuracy and memory

usage for the DGFM when compared to results obtained by the MoM. Promising runtime results indicate that the method performs well for particular cases, i.e. where the number of unknowns in the static domain is much larger than that of the dynamic domain (the array).

# 7

# Conclusion and future work

## 7.1 General conclusions

In this work, a MoM-based domain decomposition technique, the DGFM and also its accuracy improved version, the i-DGFM, were developed for the analysis of finite antenna arrays. The DGFM reduces the problem to the analysis of a single array element, resulting in significant runtime and memory savings when compared to the full-wave MoM solution of the entire antenna array. Mutual coupling between domains is accounted for by calculating an active impedance matrix equation for each array element. This matrix equation is formulated by using, initially, zeroth order weighting coefficients for the coupling terms, which is based on the work done by Skrivervik *et al.* in [43–46]. A rigorous comparison with the CBFM, showed that the DGFM equivalently only accounts for the interaction between primary CBFs.

The DGFM is limited to problems for which the currents between array elements are smoothly varying. This limitation stems from the fact that the DGFM is a perturbation approach where the currents on array elements are assumed equal at first, apart from a constant scaling factor. This assumption is then perturbed through the solution of the active impedance matrix equations for each of the domains. The accuracy limitation of the DGFM was addressed through the formulation of the i-DGFM. The improvements introduced for the i-DGFM include the derivation of a coefficient matrix used in the active impedance matrix calculations. This coefficient matrix is calculated using the ratio of the expansion coefficients of the individual basis functions between the array elements. Inherent in the aforementioned, is that the current on domains be calculated as accurately (and computationally efficiently) as possible. For this, the Jacobi method was used to approximate the current on each array element as the summation of primary and secondary CBFs. The accuracy improvements introduced by the i-DGFM allows for the calculation of embedded element patterns in passive arrays, as well as for the investigation of larger scan-angles in the case of phased array analysis, which was illustrated at the hand of different test cases. Efficient acceleration strategies for the active impedance matrix calculations, as well as a hybrid MPI/OpenMP parallelisation technique were implemented for the DGFM (that are also applicable to the i-DGFM). Of these strategies, the use of the ACA algorithm to approximate the coupling terms in the active impedance matrices with low-rank representations, proved very efficient. The class of array problems that can be analysed with

the DGFM and i-DGFM include both regular and irregular array geometries.

Finally, the DGFM was also combined with existing domain decomposition methods, viz. the CBFM and NGF, for the analysis of electrically large array geometries consisting of subarray tiles, as well as finite antenna arrays in the presence of other structures, e.g. finite sized ground planes. The DGFM now forms part of the array analysis tool in the commercial CEM software package, FEKO.

## 7.2 Future work

Future work can be directed towards the following extensions for the DGFM/i-DGFM, as well as the hybrid CBFM-DGFM and NGF-DGFM methods:

1. The treatment of interconnected array structures, as done for the CBFM [30], or the SFx [32]. A proposal for how this can be treated is to extract the unknowns on the interfaces between domains, and to reduce the problem into two sub-problems, each consisting of identical domains. Each of these sub-problems can then be treated independently using the DGFM/i-DGFM.

2. An efficient manner in which multiple excitations can be treated, e.g. for monostatic RCS calculations, or multiple scan angles for phased array analysis. For the CBFM, this is achieved through the formulation of the reduced impedance matrix equation, as explained in Section 3.3.2.

3. Investigate the use of Krylov subspace solvers to obtain an initial estimate for the current distribution on the array elements. The latter can be used to derive the improved weighting coefficients for the i-DGFM, which can be compared to the existing method of using two iterations of the Jacobi method as discussed in Section 5.2.

4. A method to indicate the uncertainty level in the DGFM (or i-DGFM) results would be beneficial. A possible way in which this can be done is to consider an additional post-processing step where the perturbation assumption (as explained in Section 4.1) is evaluated. This can be done by extracting the resulting $\alpha_{qp}$ coefficient *after* the current has been calculated by the DGFM (or i-DGFM) and then comparing it to the value that was estimated (initially) for the ratio of the currents on domains $p$ and $q$, respectively. A quantitative analysis should be performed regarding the maximum perturbation allowed and the accuracy of the results.

5. For the hybrid CBFM-DGFM, the extension to use also the i-DGFM should improve the numerical accuracy of the method.

6. For the hybrid NGF-DGFM, a strategy to calculate the correction factors that are introduced for the active impedance matrix equations, as derived in Section 6.2.1, more efficiently. This should allow for better runtime scaling, regardless of the ratio of the number of unknowns between the static and dynamic domains.

# List of References


[1] van Tonder, J. and Jakobus, U.: Introduction of curvilinear higher-order basis functions for MoM and MLFMM in FEKO. *2013 Computational Electromagnetics Workshop (CEM)*, pp. 13–14, 2013.

[2] The LOFAR website.
Available at: `http://www.lofar.org/`

[3] Skolnik, M.I.: Introduction to Radar. *Radar Handbook*, p. 1990, 1962.

[4] Bucci, O., Isernia, T., Morabito, A., Perna, S. and Pinchera, D.: Aperiodic arrays for space applications: An effective strategy for the overall design. In: *EuCAP 2009. 3rd European Conference on Antennas and Propagation*, pp. 2031–2035. 2009.

[5] The South-African Square Kilometre Array Project. 2012.
Available at: `www.ska.ac.za`

[6] *FEKO Suite 7.0 — Field Computations Involving Bodies of Arbitrary Shape*. EM Software & Systems - S.A. (Pty) Ltd, Stellenbosch, South Africa, 2014. `http://www.feko.info`.

[7] Harrington, R.F. and Harrington, J.L.: *Field computation by Moment Methods*. Oxford University Press, 1996.

[8] Asvestas, J.S.: The physical optics method in electromagnetic scattering. *Journal of Mathematical Physics*, vol. 21, p. 290, 1980.

[9] Kline, M.: Electromagnetic theory and geometrical optics. Tech. Rep., DTIC Document, 1962.

[10] Kouyoumjian, R.G. and Pathak, P.H.: A uniform geometrical theory of diffraction for an edge in a perfectly conducting surface. *Proceedings of the IEEE*, vol. 62, no. 11, pp. 1448–1461, 1974.

[11] Chew, W.C., Michielssen, E., Song, J. and Jin, J.: *Fast and efficient algorithms in computational electromagnetics*. Artech House, Inc., 2001.

[12] Gumerov, N.A. and Duraiswami, R.: *Fast multipole methods for the Helmholtz equation in three dimensions*. Access Online via Elsevier, 2005.

[13] van Tonder, J.J. and Jakobus, U.: Fast multipole solution of metallic and dielectric scattering problems in feko. In: *Wireless Communications and Applied Computational Electromagnetics, 2005. IEEE/ACES International Conference on*, pp. 511–514. 2005.

[14] Bebendorf, M.: Approximation of boundary element matrices. *Numerische Mathematik*, vol. 86, no. 4, pp. 565–589, 2000.

[15] Kurz, S., Rain, O. and Rjasanow, S.: The adaptive cross-approximation technique for the 3D boundary-element method. *IEEE Transactions on Magnetics*, vol. 38, no. 2, pp. 421–424, 2002.

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

[17] Zhao, K., Vouvakis, M.N. and Lee, J.-F.: The adaptive cross approximation algorithm for accelerated method of moments computations of EMC problems. *IEEE Transactions on Electromagnetic Compatibility*, vol. 47, no. 4, pp. 763–773, 2005.

[18] Rjasanow, S. and Steinbach, O.: *The fast solution of boundary integral equations. Mathematical and analytical techniques with applications to engineering*, vol. 152. Springer, New York, 2007.

[19] Jakobus, U.: Parallel computation of electromagnetic fields based on integral equations. In: *High Performance Computing in Science and Engineering'98*, pp. 377–386. Springer, 1999.

[20] Jakobus, U.: Application of integral equation and hybrid techniques to the parallel computation of electromagnetic fields in a distributed memory environment. *ACES Journal (Special Issue on Computational Electromagnetics and High Performance Computing)*, vol. 13, pp. 87–98, 1998.

[21] Jakobus, U., Bingle, M., Burger, W., Ludick, D., Schoeman, M. and van Tonder, J.: Method of Moments Accelerations and Extensions in FEKO. *Proceedings of the 11th International Conference on Electromagnetics in Advanced Applications - ICEAA'11*, September 2011.

[22] Lezar, E. and Davidson, D.: GPU-Accelerated Method of Moments by Example: Monostatic Scattering. *IEEE Antennas and Propagation Magazine*, vol. 52, no. 6, pp. 120–135, December 2010. ISSN 1045-9243.

[23] Balanis, C.A.: *Antenna Theory and Design*. 3rd edn. John Wiley & Sons, Inc., Hoboken, New Jersey, 2005. ISBN 0-471-66782-X (paperback).

[24] Prisco, G. and D'Urso, M.: Maximally sparse arrays via sequential convex optimizations. *IEEE Antennas and Wireless Propagation Letters*, vol. 11, pp. 192–195, 2012.

[25] Bencivenni, C., Ivashina, M.V. and Maaskant, R.: A simple method for optimal antenna array thinning using a broadside maxgain beamformer. In: *2013 7th European Conference on Antennas and Propagation (EuCAP)*, pp. 1799–1802. 2013.

[26] Bartal, Y.: *Divide-and-Conquer Methods, Wiley Encyclopedia of Electrical and Electronics Engineering*. John Wiley & Sons, Inc., 1999.

[27] Volakis, J. and Carr, M.: Domain decomposition by iterative field bouncing. *IEEE Transactions on Antennas and Propagation*, vol. 56, pp. 298–301, 2002.

[28] Prakash, V.V. and Mittra, R.: Characteristic Basis Function Method: A new technique for efficient solution of Method of Moments Matrix equations. *Microwave and Optical Technology Letters*, vol. 36, pp. 95–100, 2003.

[29] Maaskant, R.: *Analysis of Large Antenna Systems*. Ph.D. thesis, Eindhoven University of Technology, Eindhoven, 2010.

[30] Maaskant, R., Mittra, R. and Tijhuis, A.: Fast Analysis of Large Antenna Arrays using the Characteristic Basis Function Method and the Adaptive Cross Approximation Algorithm. *IEEE Transactions on Antennas and Propagation*, vol. 56, no. 11, pp. 3440–3451, 2008.

[31] Mittra, R. and Du, K.: Characteristic Basis Function Method for iteration-free solution of large Method of Moments problems. *Progress In Electromagnetics Research B*, vol. 6, pp. 307–336, 2008.

[32] Matekovits, L., Laza, V. and Vecchi, G.: Analysis of Large Complex Structures with the Synthetic-Functions Approach. *IEEE Transactions on Antennas and Propagation*, vol. 55, pp. 2509–2521, 2007.

[33] Matekovits, L., Vecchi, G., Dassano, G. and Orefice, M.: Synthetic function analysis of large printed structures: the solution space sampling approach. In: *IEEE Antennas and Propagation Society International Symposium*, vol. 2, pp. 568–571. 2001.

[34] Peng, Z., Wang, J., Lei, F.-R. and Lee, J.-F.: New computational strategies for electromagnetic modeling of multiscale heterogeneous composites. In: *Proceedings of the 5th European Conference on Antennas and Propagation (EuCAP)*, pp. 3226–3229. 2011.

[35] Wang, X., Peng, Z. and Lee, J.: A full-wave solution strategy for computing antenna couplings on a mockup fighter aircraft at Ku band. In: *2012 Proceedings ESA Workshop on Aerospace EMC*, pp. 1–6. 2012.

[36] Wang, X., Peng, Z., Lim, K.-H. and Lee, J.-F.: Multisolver Domain Decomposition Method for Modeling EMC Effects of Multiple Antennas on a Large Air Platform. *IEEE Transactions on Electromagnetic Compatibility*, vol. 54, no. 2, pp. 375–388, 2012.

[37] Jakobus, U., Marchand, R.G. and Ludick, D.J.: Aspects of and Insights Into the Rigorous Validation, Verification, and Testing Processes for a Commercial Electromagnetic Field

Solver Package. *IEEE Transactions on Electromagnetic Compatibility*, vol. 56, no. 4, 2014.

[38] Jakobus, U.: Overview of recent extensions in the electromagnetics computer code FEKO and their application. *20th Annu. Review of Progress in Applied Computational Electromagnetics*.

[39] Bingle, M., Ludick, D., Jakobus, J. and Schoeman, M.: Enhanced Electromagnetic Modeling Features available in FEKO Suite 6.1. *IEEE Symposium on Antennas and Propagation*, 2012.

[40] van Tonder, J. and Jakobus, U.: Infinite Periodic Boundary Conditions in FEKO. *ACES Journal*, vol. 24, pp. 584–591, 2009.

[41] van Tonder, J. and Jakobus, U.: Infinite Periodic Boundaries in FEKO. *25th Annual Review of Progress in Applied Computational Electromagnetics*, 2009.

[42] Ludick, D. and Davidson, D.: Investigating Efficient Parallelization Techniques for the Characteristic Basis Function Method (CBFM). In: *Electromagnetics in Advanced Applications, 2009. ICEAA'09. International Conference on*, pp. 400–403. 2009.

[43] Skrivervik, A.: *Réseaux périodiques d'antennes microruban*. Ph.D. thesis, Lausanne, 1992. Available at: `http://library.epfl.ch/theses/?nr=1032,http://vpaa.epfl.ch/page14974-fr.html`

[44] Skriverik, A.K. and Mosig, J.R.: Analysis of finite phase arrays of microstrip patches. *IEEE Transactions on Antennas and Propagation*, vol. 41, no. 8, pp. 1105–1114, 1993.

[45] Skriverik, A.K. and Mosig, J.R.: Analysis of printed array antennas. *IEEE Transactions on Antennas and Propagation*, vol. 45, no. 9, pp. 1411–1418, 1997.

[46] Skriverik, A. and Mosig, J.: Arrays of multilayered printed antennas: a space domain analysis. *Proc. J. Int. Nice Antennes*, pp. 638–641, 1994.

[47] Rao, S., Wilton, D. and Glisson, A.: Electromagnetic scattering by surfaces of arbitrary shape. *IEEE Transactions on Antennas and Propagation*, vol. 30, no. 3, pp. 409–418, 1982.

[48] Graglia, R.D., Peterson, A.F. and Andriulli, F.P.: Curl-conforming Hierarchical Vector Bases for Triangles and Tetrahedra. *IEEE Transactions on Antennas and Propagation*, vol. 59, no. 3, pp. 950–959, 2011.

[49] Zah, L.P., Hu, Y.Q. and Su, T.: Efficient Surface Integral Equation Using Hierarchical Vector Bases for Complex EM Scattering Problems. *IEEE Transactions on Antennas and Propagation*, vol. 60, no. 2, pp. 952–957, 2012.

[50] Ludick, D.J., Jakobus, U. and Davidson, D.B.: Efficient analysis of finite antenna arrays using the Domain Green's Function Method. In: *2012 IEEE Antennas and Propagation Society International Symposium (APSURSI)*, pp. 1–2. 2012.

[51] Ludick, D.J., Jakobus, U. and Davidson, D.B.: Numerical analysis of finite antenna arrays using the Domain Green's Function Method. In: *2012 International Conference on Electromagnetics in Advanced Applications (ICEAA)*, pp. 216–219. Sept 2012.

[52] Ludick, D.J., Van Tonder, J. and Jakobus, U.: Combining domain decomposition solution techniques with higher order hierarchical basis functions. In: *2013 International Conference on Electromagnetics in Advanced Applications (ICEAA)*, pp. 70–73. 2013.

[53] Ludick, D.J., Maaskant, R., Davidson, D.B., Jakobus, U., Mittra, R. and de Villiers, D.: Efficient Analysis of Large Aperiodic Antenna Arrays using the Domain Greens Function Method. *IEEE Transactions on Antennas and Propagation*, vol. 62, no. 4, 2014.

[54] Ludick, D.J., Maaskant, R., Davidson, D.B., Jakobus, U. and Mittra, R.: A Comparison of Domain Decomposition Techniques for Analysing Disjoint Finite Antenna Arrays. *8th European Conference on Antennas and Propagation (EuCAP)*, pp. 2994–2998, 2014.

[55] Ludick, D.J., Maaskant, R., Mittra, R., Jakobus, U. and Davidson, D.B.: Applying the CBFM-Enhanced DGFM to the Analysis of Large Finite Antenna Arrays. In: *2013 International Conference on Electromagnetics in Advanced Applications (ICEAA)*. 2013.

[56] Ludick, D.J., Maaskant, Davidson, D.B. and Jakobus, U.: Applying the NGF-Enhanced Domain Green's Function Method to the Analysis of Antenna Arrays and Ground Planes of Finite Sizes. In: *2014 International Conference on Electromagnetics in Advanced Applications (ICEAA)*. 2014.

[57] Ludick, D.J., Maaskant, Davidson, D.B. and Jakobus, U.: Accelerating the Domain Green's Function Method through Adaptive Cross Approximation. In: *2014 International Conference on Electromagnetics in Advanced Applications (ICEAA)*. 2014.

[58] Woo, A.C., Wang, H.T., Schuh, M.J. and Sanders, M.L.: EM programmer's notebook-benchmark radar targets for the validation of computational electromagnetics programs. *IEEE Antennas and Propagation Magazine*, vol. 35, no. 1, pp. 84–89, 1993.

[59] Davidson, D.B.: *Computational Electromagnetics for RF and Microwave Engineers*. 2nd edn. Cambridge University Press, Cambridge, 2011.

[60] Makarov, S.: MoM antenna simulations, with Matlab: RWG basis functions. *IEEE Antennas and Propagation Magazine*, vol. 43, no. 5, pp. 100–107, 2001.

[61] Dunavant, D.A.: High degree efficient symmetrical gaussian quadrature formulas for the triangle. *International Journal for Numerical Methods in Engineering*, vol. 21, pp. 1129–1148, 1985.

[62] Savage, J.S. and Peterson, A.F.: Quadrature rules for numerical integration over triangles and tetrahedra. *IEEE Antennas and Propagation Magazine*, vol. 38, no. 3, pp. 100–102, 1996.

[63] Graglia, R.D.: On the numerical integration of the linear shape functions times the 3-D Green's function or its gradient on a plane triangle. *IEEE Transactions onAntennas and Propagation*, vol. 41, no. 10, pp. 1448–1455, 1993.

[64] Khayat, M.A. and Wilton, D.R.: Numerical evaluation of singular and near-singular potential integrals. *IEEE Transactions on Antennas and Propagation*, vol. 53, no. 10, pp. 3180–3190, 2005.

[65] Vipiana, F. and Wilton, D.: Numerical Evaluation via Singularity Cancellation Schemes of Near-Singular Integrals Involving the Gradient of Helmholtz-Type Potentials. *IEEE Transactions on Antennas and Propagation*, vol. 61, no. 3, pp. 1255–1265, March 2013.

[66] Botha, M.M.: Analysis and augmentation of the Duffy transformation for near-singular integrals. In: *2012 IEEE Antennas and Propagation Society International Symposium (APSURSI)*, pp. 1–2. 2012.

[67] Graglia, R.D., Wilton, D.R. and Peterson, A.F.: Higher order interpolatory vector bases for computational electromagnetics. *IEEE Transactions on Antennas and Propagation*, vol. 45, no. 3, pp. 329–342, 1997.

[68] Notaros, B.M.: Higher order frequency-domain computational electromagnetics. *IEEE Transactions on Antennas and Propagation*, vol. 56, no. 8, pp. 2251–2276, 2008.

[69] Jakobus, U.: Comparison of different techniques for the treatment of lossy dielectric/magnetic bodies within the method of moments formulation. *AEU-international journal of electronics and communications*, vol. 54, no. 3, pp. 163–173, 2000.

[70] Clarke, S. and Jakobus, U.: Dielectric material modeling in the MoM-based code FEKO. *IEEE Antennas and Propagation Magazine*, vol. 47, no. 5, pp. 140–147, 2005.

[71] Michalski, K.A. and Mosig, J.R.: Multilayered media Green's functions in integral equation formulations. *IEEE Transactions on Antennas and Propagation*, vol. 45, no. 3, pp. 508–519, 1997.

[72] Munk, B.A.: *Finite antenna arrays and FSS*. Wiley-IEEE Press, 2003.

[73] Yang, F. and Rahmat-Samii, Y.: *Electromagnetic band gap structures in antenna engineering*. Cambridge University Press New York (NY), 2009.

[74] Smith, D., Pendry, J. and Wiltshire, M.: Metamaterials and negative refractive index. *Science*, vol. 305, no. 5685, pp. 788–792, 2004.

[75] Agrawal, V. and Lo, Y.: Mutual coupling in phased arrays of randomly spaced antennas. *IEEE Transactions on Antennas and Propagation*, vol. 20, no. 3, pp. 288–295, 1972.

[76] Bruno, O., Elling, T., Paffenroth, R. and Turc, C.: Electromagnetic integral equations requiring small numbers of Krylov-subspace iterations. *Journal of Computational Physics*, vol. 228, no. 17, pp. 6169–6183, 2009.

[77] Kelley, D.F. and Stutzman, W.L.: Array antenna pattern modeling methods that include mutual coupling effects. *IEEE Transactions on Antennas and Propagation*, vol. 41, no. 12, pp. 1625–1632, 1993.

[78] Pozar, D.M.: The active element pattern. *IEEE Transactions on Antennas and Propagation*, vol. 42, no. 8, pp. 1176–1178, 1994.

[79] Eibert, T.F., Volakis, J.L., Wilton, D.R. and Jackson, D.R.: Hybrid FE/BI modeling of 3-D doubly periodic structures utilizing triangular prismatic elements and an MPIE formulation accelerated by the Ewald transformation. *IEEE Transactions on Antennas and Propagation*, vol. 47, no. 5, pp. 843–850, 1999.

[80] Ding, D.-Z., Yung, E.K.-N., Wang, D.-X. and Chen, R.-S.: Efficient Analysis of Periodic Structures with Arbitrary Shape Using Volume-surface Integral Equation Method. *PIERS Online*, vol. 1, no. 6, pp. 677–680, 2005.

[81] Mathis, A.W. and Peterson, A.F.: Efficient electromagnetic analysis of a doubly infinite array of rectangular apertures. *IEEE Transactions on Microwave Theory and Techniques*, vol. 46, no. 1, pp. 46–54, 1998.

[82] Capolino, F., Wilton, D.R. and Johnson, W.A.: Efficient computation of the 2-D Green's function for 1-D periodic structures using the Ewald method. *IEEE Transactions on Antennas and Propagation*, vol. 53, no. 9, pp. 2977–2984, 2005.

[83] Capolino, F., Wilton, D.R. and Johnson, W.A.: Efficient computation of the 3D Greens function for the Helmholtz operator for a linear array of point sources using the Ewald method. *Journal of Computational Physics*, vol. 223, no. 1, pp. 250–261, 2007.

[84] Shubair, R. and Chow, Y.: Efficient computation of the periodic Green's function in layered dielectric media. *IEEE Transactions on Microwave Theory and Techniques*, vol. 41, no. 3, pp. 498–502, 1993.

[85] Ishimaru, A., Coe, R.J., Miller, G.E. and Geren, W.P.: Finite Periodic Structures Approach to Large Scanning Array Problems. *IEEE Transactions on Antennas and Propagation*, vol. 33, no. 11, pp. 95–100, 1985.

[86] Ling, F., C.-F., W. and Jin, J.-M.: An efficient algorithm for analyzing large scale microstrip structures using adaptive integral method combined with discrete complex-image method. *IEEE Transactions on Microwave Theory and Technology*, vol. 48, no. 11, pp. 832–839, 2000.

[87] Yuan, T., Li, L.W. and Leong, M.S.: Efficient analysis and design of finite phased arrays of printed dipoles using fast algorithm: Some case studies. *Journal of Electromagnetic Waves and Applications*, vol. 21, no. 6, pp. 737–754, 2007.

[88] Skrivervik, A. and Mosig, J.: Printed antenna arrays: a perturbation analysis. In: *Antennas and Propagation Society International Symposium (1995. AP-S. Digest)*, vol. 1, pp. 610–613. 1995.

[89] Suter, E. and Mosig, J.R.: A subdomain multilevel approach for the efficient MoM analysis of large planar antennas. *Microwave and Optical Technology Letters*, vol. 26, no. 4, pp. 270–277, 2000.

[90] Bekers, D.J.: *Finite antenna arrays: An eigencurrent approach.* Ph.D. thesis, Technische Universiteit Eindhoven, 2004.

[91] Bekers, D.J., van Eijndhoven, S.J., van de Ven, A.A., Borsboom, P.-P. and Tijhuis, A.G.: Eigencurrent analysis of resonant behavior in finite antenna arrays. *IEEE Transactions on Microwave Theory and Techniques*, vol. 54, no. 6, pp. 2821–2829, 2006.

[92] Bekers, D.J., van Eijndhoven, S.J. and Tijhuis, A.G.: An eigencurrent approach for the analysis of finite antenna arrays. *IEEE Transactions on Antennas and Propagation*, vol. 57, no. 12, pp. 3772–3782, 2009.

[93] Hay, S.G., O'Sullivan, J.D. and Mittra, R.: Connected patch array analysis using the Characteristic Basis Function Method. *IEEE Transactions on Antennas and Propagation*, vol. 59, no. 6, pp. 1828–1837, 2011.

[94] Tai, C.-T.: *Dyadic Green functions in electromagnetic theory*, vol. 272. IEEE press New York, 1994.

[95] Konno, K., Chen, Q., Sawaya, K. and Sezai, T.: Optimization of Block Size for CBFM in MoM. *IEEE Transactions on Antennas and Propagation*, vol. 50, no. 10, pp. 4719–4724, 2012.

[96] Gonzalez-Ovejero, D. and Craeye, C.: Interpolatory Macro Basis Functions analysis of non-periodic arrays. *IEEE Transactions on Antennas and Propagation*, vol. 59, no. 8, pp. 3117–3122, 2011.

[97] Press, W., Teukolsky, S., Vetterling, W. and Flannery, B.: *Numerical recipes 3rd edition: The art of scientific computing.* Cambridge University Press, 2007.

[98] Argonne National Laboratory, Mathematical and Computer Science Division: The Message Passing Interface (MPI) standard. 2010.
Available at: `http://www.mcs.anl.gov/research/projects/mpi/index.htm`

[99] The OpenMP API Specification for parallel programming. 2012.
Available at: `www.openmp.org`

[100] DuHamel, R. and Ore, F.: Logarithmically periodic antenna designs. In: *IRE International Convention Record*, vol. 6, pp. 139 – 151. 1958.

[101] DuHamel, R. and Isbell, D.: Broadband logarithmically periodic antenna structures. In: *IRE International Convention Record*, vol. 5, pp. 119–128. 1957.

[102] Schoeman, D.H.: *Full scale low-cost ultra wide band antenna for SKA low frequency array.* Master's thesis, Stellenbosch: Stellenbosch University, 2013.

[103] Dewdney, P.E., T.W., Millenaar, R., McCool, R., Lazio, J. and Cornwell, T.J.: SKA1 System Baseline Design. Tech. Rep., SKA Office, 2013.

[104] Burke, C. and Poggio, A.: Numerical Electromagnetics Code (NEC) - Method of Moments. 1981.

[105] Maaskant, R., Mittra, R. and Tijhuis, A.G.: Fast solution of multi-scale antenna problems for the square kilometre array (SKA) radio telescope using the characteristic basis function method (CBFM). *The Applied Computational Electromagnetics Society (ACES) Journal*, vol. 24, no. 2, pp. 174–188, 2009.

[106] Davidson, D.B.: *Parallel algorithms for electromagnetic moment method formulations.* Ph.D. thesis, Stellenbosch: Stellenbosch University, 1991.

[107] de Lera Acedo, E.: SKALA: A log-periodic antenna for the SKA. In: *2012 International Conference on Electromagnetics in Advanced Applications (ICEAA)*, pp. 353 –356. Sept 2012.

[108] Melais, S.E. and Weller, T.M.: A multilayer jerusalem cross frequency selective surface. In: *IEEE 10th Annual Wireless and Microwave Technology Conference (WAMICON'09)*, pp. 1–5. 2009.

[109] Brand, Y., Skrivervik, A.K., Mosig, J.R. and Gardiol, F.E.: New iterative integral equation technique for multilayered printed array antennas. In: *Mathematical Methods in Electromagnetic Theory*, pp. 615–617. Kharkov, Ukraine, June 1998.

[110] Polycarpou, A.C.: Evaluation of stationary block iterative techniques for the solution of finite arrays using the FE-BI method and domain decomposition. In: *Proc. European Conference on Antennas and Propag. (EuCAP)*, pp. 1–6. Nice, France, 2006.

[111] Brand, Y., Skrivervik, A.K. and Mosig, J.R.: An iterative scheme solution for the analysis of printed arrays. *Microwave and Optical Technology Letters*, vol. 16, no. 2, pp. 106–115, 1997.

[112] de Vos, M., Gunst, A.W. and Nijboer, R.: The LOFAR telescope: System architecture and signal processing. *Proceedings of the IEEE*, vol. 97, no. 8, pp. 1431–1437, 2009.

[113] van Ardenne, A., Bregman, J.D., van Cappellen, W.A., Kant, G.W. and de Vaate, J.B.: Extending the field of view with phased array techniques: Results of European SKA research. *Proceedings of the IEEE*, vol. 97, no. 8, pp. 1531–1542, 2009.

[114] Lawson, C.L., Hanson, R.J., Kincaid, D.R. and Krogh, F.T.: Basic Linear Algebra Subprograms for Fortran Usage. *ACM Trans. Math. Softw.*, vol. 5, no. 3, pp. 308–323, 1979.

[115] Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A. and Sorensen, D.: *LAPACK Users' Guide.* 3rd edn. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1999.

# Appendix A

# Partially pivoted ACA algorithm

The key steps that are used in the partially pivoted ACA algorithm are summarised here. The aforementioned is used to accelerate the active impedance matrix calculation for the DGFM or i-DGFM, as discussed in Section 4.3.3 and the implementation is based on that explained in [17].

The ACA algorithm constructs the rectangular matrices $\mathbf{U}_p$ and $\mathbf{V}_q$ such that the dense MoM coupling matrix, $\mathbf{Z}_{pq}$, can be approximated as follows,

$$\mathbf{Z}_{pq} \simeq \tilde{\mathbf{Z}}_{pq} = \mathbf{U}_p^{N_i \times r} \mathbf{V}_q^{r \times N_i} = \sum_{i=1}^{r} \mathbf{u}_i^{N_i \times 1} \mathbf{v}_i^{1 \times N_i}. \tag{A.1}$$

In (A.1) $N_i$ is the number of unknowns in the domains $p$ and $q$, respectively. Here it is assumed that domains $p$ and $q$ are identically discretised. The quantity $r$ is the effective rank of $\tilde{\mathbf{Z}}_{pq}$ and decreases as the mutual coupling between domains $p$ and $q$ reduces. Furthermore, $\mathbf{u}_i$ and $\mathbf{v}_i$ are the $i$th column and row of the matrices $\mathbf{U}_p$ and $\mathbf{V}_q$, respectively.

The ACA algorithm constructs the matrices $\mathbf{U}_p$ and $\mathbf{V}_q$ by successively selecting rows and columns of the original matrix, $\mathbf{Z}_{pq}$. An error matrix is calculated each time a new row or column of $\mathbf{Z}_{pq}$ is selected. This is defined as follows,

$$\left\|\mathbf{R}\right\|_F = \left\|\mathbf{Z}_{pq} - \tilde{\mathbf{Z}}_{pq}\right\|_F. \tag{A.2}$$

In (A.2) $\left\|\cdot\right\|_F$ denotes the Frobenius norm. The process of selecting rows and columns is based on the largest computed error entry in (A.2). The ACA algorithm terminates naturally after $N_i$ iterations, as then all the rows and columns of $\mathbf{Z}_{pq}$ have been processed and are reconstructed exactly [17]. For runtime efficient implementations, the iterative process can be terminated after $r$ iterations, where $r << N_i$, or alternative when $\left\|\mathbf{R}\right\|_F \leq \kappa \left\|\mathbf{Z}_{pq}\right\|_F$ where $\kappa$ is some threshold (typically selected in the order of $10^{-2}$ to $10^{-3}$, depending on the required accuracy).

During each iteration, we do not reconstruct $\mathbf{Z}_{pq}$, as it is only partially known. The error matrix $\mathbf{R}$ in (A.2) is therefore approximated after the $k$th iteration as,

$$\left\|\mathbf{R}\right\|_F \simeq \left\|\tilde{\mathbf{R}}\right\|_F = \left\|\mathbf{u}_k\right\|_F \left\|\mathbf{v}_k\right\|_F. \tag{A.3}$$

What follows is a pseudocode overview of the algorithm in MATLAB notation, as done in [17].

**The following notations and conventions are used:**

1. The original matrix $\mathbf{Z}_{pq}$ is written as $\mathbf{Z}$ that is of size $N_i \times N_i$. Likewise, the approximate matrix $\tilde{\mathbf{Z}}_{pq}$ is written as $\tilde{\mathbf{Z}}$.
2. Let $\mathbf{I}^{\text{row}} = [I_1^{\text{row}}, I_2^{\text{row}}, \ldots, I_r^{\text{row}}]$ and $\mathbf{I}^{\text{col}} = \left[I_1^{\text{col}}, I_2^{\text{col}}, \ldots, I_r^{\text{col}}\right]$ be the arrays containing the selected row and column indices of $\mathbf{Z}$.
3. The vectors $\mathbf{u}_k$ and $\mathbf{v}_k$ are the $k$th column and row of the matrices $\mathbf{U}$ and $\mathbf{V}$, respectively.
4. The notation $\mathbf{R}(I_1^{\text{row}}, :)$ indicates the $i$th row of the matrix $\mathbf{R}$.
5. The matrix $\tilde{\mathbf{Z}}^{(k)}$ is the matrix $\tilde{\mathbf{Z}}$ at the $k$th iteration.

The ACA algorithm can now be written as follows:

***Initialisation (k=1):***

1. Initialise the 1st row index $I_1^{\text{row}} = 1$ and set $\tilde{\mathbf{Z}} = 0$.
2. Initialise the 1st row of the approximate error matrix: $\tilde{\mathbf{R}}(I_1^{\text{row}}, :) = \mathbf{Z}(I_1^{\text{row}}, :)$.
3. Find the 1st column index $I_1^{\text{col}}$ such that $|\tilde{\mathbf{R}}(I_1^{\text{row}}, I_1^{\text{col}})| = \max_j(|\tilde{\mathbf{R}}(I_1^{\text{row}}, j)|)$.
4. Set $\mathbf{v}_1 = \tilde{\mathbf{R}}(I_1^{\text{row}}, :)/\tilde{\mathbf{R}}(I_1^{\text{row}}, I_1^{\text{col}})$.
5. Initialise the 1st column of the approximate error matrix: $\tilde{\mathbf{R}}(:, I_1^{\text{col}}) = \mathbf{Z}(:, I_1^{\text{col}})$.
6. Set $\mathbf{u}_1 = \tilde{\mathbf{R}}(:, I_1^{\text{col}})$.
7. $\left\|\tilde{\mathbf{Z}}^{(1)}\right\|_F^2 = \left\|\mathbf{u}_1\right\|_F^2 \left\|\mathbf{v}_1\right\|_F^2$.
8. Find the 2nd row index $|\tilde{\mathbf{R}}(I_2^{\text{row}}, I_1^{\text{col}})| = \max_i(|\tilde{\mathbf{R}}(i, I_1^{\text{col}})|), i \neq I_1^{\text{row}}$.

***kth iteration:***

1. Update the $(I_k^{\text{row}})$th row of $\tilde{\mathbf{R}}$: $\tilde{\mathbf{R}}(I_k^{\text{row}}, :) = \mathbf{Z}(I_k^{\text{row}}, :) - \sum_{l=1}^{k-1} (\mathbf{u}_l)_{I_k^{\text{row}}} \mathbf{v}_l$.
2. Find the $k$th column index $I_k^{\text{col}}$ such that $|\tilde{\mathbf{R}}(I_k^{\text{row}}, I_k^{\text{col}})| = \max_j(|\tilde{\mathbf{R}}(I_k^{\text{row}}, j)|), j \neq I_1^{\text{col}}, \ldots, I_{k-1}^{\text{col}}$.
3. Set $\mathbf{v}_k = \tilde{\mathbf{R}}(I_k^{\text{row}}, :)/\tilde{\mathbf{R}}(I_k^{\text{row}}, I_k^{\text{col}})$.
4. Update the $(I_k^{\text{col}})$th column of $\tilde{\mathbf{R}}$: $\tilde{\mathbf{R}}(:, I_k^{\text{col}}) = \mathbf{Z}(:, I_k^{\text{col}}) - \sum_{l=1}^{k-1} (\mathbf{v}_l)_{I_k^{\text{col}}} \mathbf{u}_l$.
5. Set $\mathbf{u}_k = \tilde{\mathbf{R}}(:, I_k^{\text{col}})$.
6. Calculate $\left\|\tilde{\mathbf{Z}}^{(k)}\right\|_F^2 = \left\|\tilde{\mathbf{Z}}^{(k-1)}\right\|_F^2 + 2\sum_{j=1}^{k-1} |\mathbf{u}_j^T \mathbf{u}_k| \cdot |\mathbf{v}_j^T \mathbf{v}_k| + \left\|\mathbf{u}_k\right\|_F^2 \left\|\mathbf{v}_k\right\|_F^2$.
7. Check whether convergence is achieved and end iteration if $\left\|\mathbf{u}_k\right\|_F \left\|\mathbf{v}_k\right\|_F \leq \kappa \left\|\tilde{\mathbf{Z}}^{(k)}\right\|_F$.
8. If not converged then find next row index : $|\tilde{\mathbf{R}}(I_{k+1}^{\text{row}}, I_k^{\text{col}})| = \max_i(|\tilde{\mathbf{R}}(i, I_k^{\text{col}})|), i \neq I_1^{\text{row}}, \ldots, I_k^{\text{row}}$.