

Design and Optimization of Meta-Ferrites for Antenna Applications

A thesis presented for the degree of Doctor of Philosophy in the
Faculty of Engineering of the University of London

By
Philip Beal

School of Electronic Engineering and Computer Science
Queen Mary University of London

September 2018

I, Philip Beal, confirm that the research included within this thesis is my own work or that where it has been carried out in collaboration with, or supported by others, that this is duly acknowledged below and my contribution indicated. Previously published material is also acknowledged below.

I attest that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge break any UK law, infringe any third party's copyright or other Intellectual Property Right, or contain any confidential material.

I accept that the College has the right to use plagiarism detection software to check the electronic version of the thesis.

I confirm that this thesis has not been previously submitted for the award of a degree by this or any other university.

The copyright of this thesis rests with the author and no quotation from it or information derived from it may be published without the prior written consent of the author.

Signature: P. Beal

Date: 14/09/2018

Abstract

This thesis aims to study the design and optimization of Metamaterial Ferrite (metaferrite) structures for antenna applications. These techniques are primarily focused on various forms of the Genetic Algorithm methods. The motivation for this is to provide generic design tools that will be transformed to the industry for further development of the field, especially for the design of low profile antennas. Several possible applications of this technology have been investigated.

Antenna size reduction on platforms with low radar cross sections (RCS) is essential for many practical applications and through metamaterials antennas and associated platforms can be co-designed and optimized. Metamaterials are artificially designed electromagnetic materials or structures which possess properties not existing in nature. Examples include left-handed materials with simultaneous negative permittivity and permeability and zero index media etc. In this project we are particularly interested in the development of two-dimensional metamaterials or “metasurfaces” with the same function as conventional magnetic or magneto-dielectric materials. Ferrites, often used in the design of non-reciprocal microwave devices such as isolators and circulators, are bulky, heavy and lossy for antenna applications. Impedance matched media have been widely used in the design of low-profile antennas and radar absorbers with losses added. Metamaterials and metasurfaces can mimic ferrites at microwave frequencies. These “metaferrites” have attracted interest from academic and industrial communities. In this thesis, a genetic algorithm (GA) is developed incorporated with commercial electromagnetic modelling tools, namely CST Microwave Studio. Antennas with ferrites have been characterized and unique functionalities have been explored. Sample structures have been made based on conventional printed circuit board (PCB) and ink-jet printing technologies. In addition, these structures have been used to augment U-slot patch antennae and the advantages of doing so have been measured.

In addition, a new concept of hyperuniform randomness has been applied to the design of metasurfaces with the intention of comparing metaferrite performance with an alternate method aimed at similar goals. It has been demonstrated by numerical simulations that superior properties such as RCS reduction can be achieved with the consideration of trade-offs between the modelling complexity and the device performance.

Publications

Philip Beal, Yang Hao, “Design of Metaferrites using Genetic Algorithms and Other Optimisation Techniques.” *Innovations in Large Area Electronics conference 2017*, January 2017

Yujie Liu, Philip Beal, Yang Hao, Henry Giddens, “Designing Metamaterial Ferrites via Genetic Algorithm Techniques”, *IEEE Antennas and Wireless Propagation Letters*, to be submitted 2018

Acknowledgement

This thesis was made possible thanks to the aid of many colleagues at Queen Mary's department of Electronic Engineering and Computer Science. Chief among them were Professor Yang Hao, Dr Khalid Rajab, Dr Patrick Bradley, Dr Henry Giddens, Yujie Liu and Haoyang Zhang.

My Supervisor Professor Hao's advice and guidance was, of course, extremely important for keeping this project on track and also for polishing the thesis before you to its current state. His dedication to the field of engineering is an example to strive for. Dr Rajab initially served as my second supervisor, until stepping down from the position partway through the project, but his support until then was likewise of great help.

Dr Bradley provided help with several aspects of the genetic algorithm program, including helping to establish the crucial MATLAB to CST link that the program relies upon. He also gave advice regarding the translation of S11 parameters to other values required by the program.

Dr Giddens' assistance in fabricating the structures designed by the genetic algorithm and testing them was indispensable, and Yujie's aid in validating results was likewise an important factor in ensuring their integrity.

Haoyang performed the bulk of the work on hyperuniform designs. I mostly just helped him by generating the structures he requested in CST, and so his aid in writing chapter 6 was very much appreciated.

I'd also like to thank my parents, for their continued encouragement and moral support.

This thesis was funded by BAE systems and Engineering and Physical Sciences Research Council. Without their generous support none of this would have been possible.

Table of Contents

Abstract	2
Publications	3
Acknowledgement	4
Table of Contents	5
List of Figures	9
Abbreviations	14
Chapter 1: Introduction	15
1.1: Background	15
1.2: State of the Art and Challenges	16
1.3: Contributions	20
1.4: Structure of the Thesis	21
1.5: References	23
Chapter 2: Review of Metaferrites	25
2.1: Overview	25
2.1.1: Metaferrites as High Impedance Surfaces	25
2.1.2: Theoretical Basis	27
2.1.3: Alternative Designs	31
2.1.4: Other Designs	32
2.2: Summary	38
2.3: References	40
Chapter 3: Computational Optimisation Techniques	42
3.1: Overview	42
3.1.1: The Newton-Raphson Method	42
3.1.2: The Quasi-Newton Method	43
3.1.3: Particle Swarm Technique	44
3.1.4: Genetic Algorithm Technique	45

3.1.5: Other Optimisation Techniques	46
3.1.6: Machine Learning	49
3.2: Genetic Algorithm Outline	54
3.1.2: Generation of the Seed Population and Export to CST	55
3.2.2: Building the Structures	57
3.2.3: Analysis of Structures	58
3.2.4: Evaluation by Fitness Function	59
3.2.5: Recombination of Solutions and the Creation of the Second Generation.	60
3.3: Application of Fractal Designs	61
3.4: Final Design and Modifications	63
3.5: Validation of the Genetic Algorithm	65
3.6: Summary	71
3.7: References	73
Chapter 4: Design and Characterisation of Metasurfaces	77
4.1: Basic Design	77
4.2: Determination of Optimal Unit Cell Size	78
4.3: Analysis of Desired Features	84
4.4: Inclusion of Optimisation Parameters	85
4.5: Analysis of the Effects of Metallic Fraction on High-Impedance Surface Behaviour	89
4.6: Post-Correction Designs	91
4.7: Summary	96
4.8: References	99

Chapter 5:	Fabrication and Measurement of Metasurfaces	101
5.1:	Overview	101
5.2:	Silver Ink	102
5.2.1:	The Inkjet Printing Method	102
5.2.2:	Fabrication of Chosen Design	103
5.3:	Milled Copper on FR4	107
5.3.1:	Fabrication and Analysis	107
5.3.2:	Further Simulation of Copper Metaferrite Design	113
5.3.3:	U-Slot Patch Antenna Fabrication	115
5.4:	Summary	117
5.5:	References	119
Chapter 6:	Hyperuniform Metamaterials	120
6.1:	Other Metaferrite Structures: Random vs Periodic vs Hyperuniform	120
6.2:	Summary	134
6.2:	References	135
Chapter 7:	Conclusion and Future Work	137
7.1:	Conclusions	137
7.1.1:	Novel Contributions	137
7.1.2:	Summary	137
7.1.3:	Genetic Algorithm	138
7.1.4:	Fabrication	140
7.1.5:	Other Parameters Studied.	141
7.2:	Future Work	142
7.2.1:	Alternate Materials	142
7.2.2:	Alternate Optimisation Procedure	143
7.2.3:	Alternate Goals	143
7.2.4:	Further Exploration of Previously Touched Upon Designs.	144

List of Figures

FIGURE 2.1:	SIMPLE DIAGRAM OF AN LC CIRCUIT SHOWING CAPACITOR AND INDUCTOR COMPONENTS	26
FIGURE 2.2:	TYPICAL EXAMPLE OF SIMPLE HIGH IMPEDANCE SURFACE STRUCTURE	27
FIGURE 2.3:	UNIT CELL OF METAFERRITE GENERATED BY PRIOR WORK ON THE TOPIC	30
FIGURE 2.4:	EXAMPLE OF RECONFIGURABLE METAFERRITE DESIGN.	32
FIGURE 2.5:	PERSPECTIVE VIEW OF A HEXAGONAL TWO –LAYER PRINTED “MUSHROOM” HIGH-IMPEDANCE SURFACE DESIGN	33
FIGURE 2.6:	RHOMBIC UNIT CELL STRUCTURE DESIGN.	34
FIGURE 2.7:	DEMONSTRATION OF HOW RHOMBIC UNIT CELL STRUCTURE WAS DESIGNED TO ENSURE TESSELLATION WHILST KEEPING THE HEXAGONAL PANELS AS SINGLE OBJECTS.	34
FIGURE 2.8:	RECTANGULAR UNIT CELL STRUCTURE DESIGN.	35
FIGURE 2.9:	A DEMONSTRATION OF THE CONTINUITY OF HEXAGONAL PANELS ACHIEVED USING RECTANGULAR CELL STRUCTURE	36
FIGURE 2.10:	PREDICTIONS OF EFFECTIVE MEDIUM MODEL	37
FIGURE 2.11:	RESULTS OF CST’S EIGENMODE SOLVER ANALYSIS OF THE RHOMBIC UNIT CELL HEXAGONAL PATTERN STRUCTURE.	37
FIGURE 2.12:	RESULTS OF CST’S EIGENMODE SOLVER ANALYSIS OF THE RECTANGULAR UNIT CELL HEXAGONAL PATTERN STRUCTURE.	38
FIGURE 3.1:	DIAGRAM OF ZIKRI’S CONSTRAINED AND UNCONSTRAINED STRUCTURES SHOWING BOTH UNIT CELL AND CONNECTED PINWHEEL STRUCTURE.	48
FIGURE 3.2:	DIAGRAM OF THE GENETIC ALGORITHM USED IN ANOTHER PROJECT	51
FIGURE 3.3:	FLOW CHART DETAILING GENETIC ALGORITHM RUN PROCEDURE.	55
FIGURE 3.4:	FLOWCHART OF THE INITIAL CHROMOSOME GENERATION PROCEDURE.	56
FIGURE 3.5:	MAP OF BIT NUMBERS IN THE STRING TO THE PIXEL IN THE HIGH-IMPEDANCE SURFACE TEMPLATE THEY CORRESPOND TO.	57

FIGURE 3.6:	DETAIL OF THE BUILD AND ANALYSIS STEP, SEPARATED INTO THE PROGRAMS USED FOR EACH COMPONENT	58
FIGURE 3.7:	FITNESS FUNCTION EVALUATION AND RECOMBINATION PROCEDURE.	60
FIGURE 3.8:	THE CONSTRUCTION PROCESS OF A DRAGON CURVE.	62
FIGURE 3.9:	A HILBERT CURVE, ANOTHER FRACTAL PATTERN THAT CAN BE CREATED USING AN L-SYSTEM.	62
FIGURE 3.10:	EXAMPLE OF STRUCTURE TO BE GENERATED BY THE GA PROGRAM	63
FIGURE 3.11:	FINAL ITERATION OF GENETIC ALGORITHM FUNCTION.	65
FIGURE 3.12:	THE FIRST ORDER TWO-DIMENSIONAL GRIEWANK FUNCTION	67
FIGURE 3.13:	THE FIRST ORDER GRIEWANK FUNCTION USED FOR TESTING THE GA	68
FIGURE 3.14:	GRAPH OF THE NUMBER OF GENERATIONS REQUIRED TO CONVERGE UPON A SOLUTION WHEN THE POPULATION SIZE WAS SET TO 8.	69
FIGURE 3.15:	GRAPH OF THE NUMBER OF GENERATIONS REQUIRED TO CONVERGE UPON A SOLUTION WHEN THE POPULATION SIZE WAS SET TO 12	69
FIGURE 3.16:	THE SECOND ORDER TWO-DIMENSIONAL GRIEWANK FUNCTION USED FOR TESTING THE GENETIC ALGORITHM	70
FIGURE 3.17:	GRAPH OF THE NUMBER OF GENERATIONS REQUIRED BY THE PROGRAM TO CONVERGE UPON A SOLUTION WITH A POPULATION SIZE OF 12	71
FIGURE 4.1:	DIAGRAM OF THE CONSTRUCTION PROCESS OF THE METAFERRITE STRUCTURES	77
FIGURE 4.2:	TYPICAL APPEARANCE OF A RANDOMLY-GENERATED 16 X 16 MM STRUCTURE	78
FIGURE 4.3:	EXAMPLE RESONANCE VALUES FOR A RANDOMLY-GENERATED 16 X 16 MM STRUCTURE	79
FIGURE 4.4:	EXAMPLE PHASE FOR A RANDOMLY-GENERATED 16 X 16 MM STRUCTURE	79
FIGURE 4.5:	TYPICAL APPEARANCE OF A RANDOMLY-GENERATED 18 X 18 MM STRUCTURE.	80

FIGURE 4.6: EXAMPLE RESONANCE VALUES FOR A RANDOMLY-GENERATED 18 X 18 MM STRUCTURE.	81
FIGURE 4.7: EXAMPLE PHASE FOR A RANDOMLY-GENERATED 18 X 18 MM STRUCTURE.	81
FIGURE 4.8: TYPICAL APPEARANCE OF A RANDOMLY-GENERATED 20 X 20 MM STRUCTURE	82
FIGURE 4.9: EXAMPLE RESONANCE VALUES AND PHASE FOR A RANDOMLY-GENERATED 20 X 20 MM STRUCTURE	83
FIGURE 4.10: PINWHEEL STRUCTURE REPLICATED FOR THE PURPOSES OF COMPARISON	84
FIGURE 4.11: COMPARISON OF PERMITTIVITY AND PERMEABILITY OF STRUCTURE IN WERNER'S PAPER WITH ONE OF SEVERAL PEAKS IN THE REFLECTION PHASE OF A STRUCTURE RANDOMLY DESIGNED BY THE PROJECT	85
FIGURE 4.12: COMPARISON OF REAL AND IMAGINARY COMPONENTS OF THE PERMEABILITY AND PERMITTIVITY OF THE STRUCTURE	86
FIGURE 4.13: REAL AND IMAGINARY COMPONENTS OF PERMEABILITY OF STRUCTURE WHEN PLACED BEHIND A LAYER OF SUBSTRATE AND PATCH ANTENNA.	87
FIGURE 4.14: MULTIPLE UNIT CELL STRUCTURE, WITH AND WITHOUT A SQUARE PATCH AND ADDITIONAL LAYER OF SUBSTRATE PLACED IN FRONT	88
FIGURE 4.15: RESONANCE PATTERN PRODUCED BY THE STRUCTURE IN FIGURE 4.14	89
FIGURE 4.16: FREQUENCY OF LOWEST RESONANCE POINT OF NUMEROUS STRUCTURES PLOTTED AGAINST THE FRACTION OF SAID STRUCTURE COMPOSED OF METAL	90
FIGURE 4.17: STRUCTURE GENERATED AFTER 35 GENERATIONS BY CORRECTED ALGORITHM	91
FIGURE 4.18: MAGNITUDE PARAMETER OF ALTERNATE STRUCTURE GENERATED AFTER 35 GENERATIONS BY CORRECTED ALGORITHM	92
FIGURE 4.19: PHASE PARAMETER OF ALTERNATE STRUCTURE GENERATED AFTER 35 GENERATIONS BY CORRECTED ALGORITHM	93
FIGURE 4.20: DIELECTRIC RESPONSE PARAMETER OF ALTERNATE STRUCTURE GENERATED AFTER 35 GENERATIONS BY CORRECTED ALGORITHM	93
FIGURE 4.21: ALTERNATE VARIANT OF STRUCTURE GENERATED BY CORRECTED ALGORITHM AFTER 101 GENERATIONS	94
FIGURE 4.22: MAGNITUDE RESPONSE OF ALTERNATE VARIANT OF STRUCTURE GENERATED BY CORRECTED ALGORITHM AFTER 101 GENERATIONS	95

FIGURE 4.23: PHASE RESPONSE OF ALTERNATE VARIANT OF STRUCTURE GENERATED BY CORRECTED ALGORITHM AFTER 101 GENERATIONS	95
FIGURE 4.24: PERMITTIVITY RESPONSE OF ALTERNATE VARIANT OF STRUCTURE GENERATED BY CORRECTED ALGORITHM AFTER 101 GENERATIONS	96
FIGURE 5.1: BASIC COMPOSITION OF TYPICAL METAFERRITE STRUCTURE. ALL OF THE FABRICATED STRUCTURES OUTLINED IN THIS CHAPTER CONFORM TO THIS DESIGN.	102
FIGURE 5.2: DIAGRAM OF INKJET PRINTER USED TO FABRICATE THE STRUCTURE DEPICTED IN FIGURES 57 AND 58	103
FIGURE 5.3: DESIGN OF STRUCTURE THAT WAS FABRICATED FROM SILVER INK USING INKJET PRINTING	104
FIGURE 5.4: EXAMPLE OF STRUCTURE THAT WAS FABRICATED FROM SILVER INK USING INKJET PRINTING	104
FIGURE 5.5: 3 X 3 TILE USING THE UNIT CELL IN FIGURE 72	105
FIGURE 5.6: RESULTS AS PREDICTED BY SIMULATION	106
FIGURE 5.7: RESULTS AS TESTED AT SHORT RANGE.	107
FIGURE 5.8: DESIGN OF STRUCTURE THAT WAS FABRICATED FROM COPPER ON FR4 SUBSTRATE	108
FIGURE 5.9: DIAGRAM OF THE ANECHOIC CHAMBER SETUP.	109
FIGURE 5.10: 10 X 10 UNIT CELL FABRICATED FROM COPPER ON FR4 SUBSTRATE	110
FIGURE 5.11: RESULTS AS PREDICTED BY SIMULATION	111
FIGURE 5.12: RESULTS AS MEASURED AT SHORT RANGE	111
FIGURE 5.13: COMPARISON OF MEASURED (RED) AND SIMULATED (BLUE) S11 RESULTS AFTER PROCESSING VIA THE TIME GATING METHOD	112
FIGURE 5.14: COMPARISON OF MEASURED (RED) AND SIMULATED (BLUE) EFFECTIVE PERMEABILITY AFTER PROCESSING VIA THE TIME GATING METHOD.	112
FIGURE 5.15: SIMULATED RESULTS OF ANTENNA IN FRONT OF VACUUM IN AIR AND EFFECTIVE MEDIUM WITH PARAMETERS OF METAFERRITE WITH 13MM PERIODIC	113
FIGURE 5.16: SIMULATED RESULTS OF ANTENNA IN AIR (GREEN), EFFECTIVE MEDIUM (RED) AND THE ACTUAL METAFERRITE (BLUE) WITH 18MM PERIODIC	114

FIGURE 5.17:	PREDICTED RCS RESPONSE OF THE STRUCTURE, OBTAINED VIA SIMULATION.	114
FIGURE 5.18:	THE LAYERED DESIGN OF THE METAFERRITE ANTENNA COMPARED WITH THE THEORETICAL EFFECTIVE MEDIUM MODEL.	115
FIGURE 5.19:	DIAGRAM OF U-SLOT PATCH ANTENNA OVER METAFERRITE BASE.	115
FIGURE 5.20:	IMAGE OF FABRICATED U-SLOT PATCH ANTENNA OVER METAFERRITE BASE.	116
FIGURE 5.21:	COMPARISON OF SIMULATED AND MEASURED RCS REDUCTION VALUES FOR U-SLOT PATCH ANTENNA	116
FIGURE 5.22:	COMPARISON OF MEASUREMENT AND SIMULATION OF THE NORMALISED E- AND H-PLANE RADIATION PATTERNS AT 8.65, 8.8 AND 9 GHz	117
FIGURE 6.1:	COMPARISON OF RANDOMISED PARTICLE DISTRIBUTION TO HYPERUNIFORM DISTRIBUTION	121
FIGURE 6.2:	PARAMETERS OF DIPOLE UNIT CELLS USED IN METAFERRITE STRUCTURES DESCRIBED BELOW	123
FIGURE 6.3:	S- PARAMETERS RESPONSE OF DIPOLE UNIT CELLS USED IN METAFERRITE STRUCTURES TAKEN AT A RANGE OF VARIABLES AND ANGLES OF INCIDENCE	124
FIGURE 6.4:	PERIODIC STRUCTURE DISPLAYING A PHASE SHIFT AT 9 GHz	125
FIGURE 6.5:	FARFIELD RESPONSE AS 9GHz PERIODIC STRUCTURE	126
FIGURE 6.6:	HYPERUNIFORM STRUCTURE DISPLAYING A PHASE SHIFT AT 9 GHz	127
FIGURE 6.7:	FARFIELD RESPONSE AS 9GHz HYPERUNIFORM STRUCTURE	128
FIGURE 6.8:	COMPARISON OF THE FARFIELD RESPONSE OF PERIODIC AND HYPERUNIFORM STRUCTURES	129
FIGURE 6.9:	PERIODIC STRUCTURE CONTAINING 144 DIPOLES.	130
FIGURE 6.10:	HYPERUNIFORM STRUCTURE CONTAINING 197 DIPOLES.	131
FIGURE 6.11:	RANDOM DISTRIBUTION STRUCTURE CONTAINING 208 DIPOLES.	131
FIGURE 6.12:	TYPICAL FARFIELD RESPONSE OF HIGH-DENSITY METASURFACES	132
FIGURE 6.13:	TABLE DEPICTING RCS REDUCTION BANDWIDTH OF VARIOUS METAMATERIAL STRUCTURES.	133

Abbreviations

ABC	Artificial Bee Colony
CST	Computer Simulation Technology
DE	Differential Evolution
FR4	Flame-Retardant glass-reinforced epoxy laminate material, type 4
FSS	Frequency Selective Surface
GA	Genetic Algorithm
L-system	Lindenmayer-system
PEC	Perfect Electrical Conductor
PET	Polyethylene terephthalate
PMC	Perfect Magnetic Conductor
PSO	Particle Swarm Optimisation
RCS	Radar Cross Section
TE	Transverse Electric
VBA	Visual Basic

Chapter 1: Introduction

1.1: Background

Magnetic materials are a class of material that display the unusual property being able to hold a permanent magnetic field. The origin of this force is the same process that causes electromagnetic phenomena, namely moving electrical currents. In this case it is the orbiting electrons of their atoms that produce small magnetic dipoles. This occurs in all materials, but in most the dipoles are randomly oriented and thus cancel each other out when observed at macroscopic scales. In a magnetic material these dipoles can become aligned, greatly magnifying the effect. These aligned dipoles can do more than simply exert a force, and so magnetic materials have seen their use in electronics for many years in a variety of forms. The most overt uses typically rely either on the forces magnets can exert or their ability to induce currents in metals when in motion. These effects are exploited by motors, generators (essentially a motor run in reverse) and electromagnets. There are also a multitude of less obvious applications, such as transformers, sensors, speakers and various other circuit components. It is this latter category of use that is of interest to this thesis.

Since they were first synthesised in the 1930s by Yogoro Kato and Takeshi Takei of the Tokyo Institute of Technology [1], magnetic materials used for electrical components have most commonly been in the form of ferrites. A ferrite is a ceramic compound of iron oxide and one or more metallic elements [2]. They are so called because they display ferrimagnetic properties, which mean that they possess the properties required to become permanently magnetised. Unlike purely metallic magnetic materials, however, ferrites are electrically non-conductive, and since they can be made to be either low-loss or as good absorbers, this has led to a wide range of applications. The tasks for which a ferrite will be suitable are determined by its composition. Adding nickel, zinc and manganese results in low coercivity compounds known as soft ferrites. Soft ferrites are useful for making transformers or electromagnetic cores because their low coercivity means that their magnetisation can change direction with minimal dissipation of energy. Adding strontium and barium, by contrast, results in hard ferrites, which have high coercivity and high remanence after magnetisation, making such compounds useful in creating permanent magnets [3]. Their low cost ensures such magnets are common, and widely used in household products. Ferrites are also used as inductors, and are used in circuits as low noise amplifiers, filters, and impedance matching networks. [4]

There are, however, a few drawbacks to ferrite materials. They tend to be brittle [5], and devices that use them tend to be rather large and cumbersome, often as a consequence of the gaps that need to be left between components for reasons of destructive interference or to address other concerns such as residual magnetism when the device is switched off, which could potentially interfere with its construction or functionality. For some applications, the persistent magnetic properties are not a problem, and in some they are exactly what you want. It is this property that allows for magnetic data storage, for example. In others devices, on the other hand, the ferrimagnets are used for other purposes that rely on their electronic properties alone. In cases such as this it is often more advantageous to have a ferrimagnet that does not display any actual magnetism, or one which you can switch off entirely. This is where metamaterial ferrites or “metaferrites”, the topic of this thesis, come in.

Genetic algorithms were first proposed in 1950 by Alan Turing [6], and the technique was first put into practice by Nils Aall Barricelli four years later [7]. There are, however, a few drawbacks to the technique. Repeatedly testing via a single fitness function can be limiting for multi-dimensional problems, and the function required may take a prohibitively long time to evaluate by itself. This is, of course, compounded by the repetitive nature of the algorithm. In some cases an exact solution must be forgone in favour of a more computationally efficient approximated function. Another issue is how Genetic Algorithms scale with complexity. Each possible variable investigated increases the design space exponentially, which has a corresponding effect on the computational load. It is for this reason that genetic algorithms commonly focus on designing individual components of items rather than the entire system. Genetic algorithms are also susceptible to an issue seen in real evolution, in that they can easily become trapped at local optima to a problem. They cannot backtrack; passing through the less optimal designs that would lead to a greater global optimum. This issue has a few solutions that can be implemented to combat it, often involving a periodic injection of more randomised solutions into the algorithm in order to promote a greater diversity in the population.

1.2: State of the Art and Challenges

Up to now, metaferrites have already emerged as a fairly well-established subject area of study. While they have been in vogue for a little over a decade and a half, there were a few papers detailing the design process but very few with experimental demonstrations [8] [9]. Most often, these have taken the form of rotationally symmetry such as those patterned high

impedance surfaces, both with and without multiple lines of symmetry. Some of these techniques will be explored with greater detail in Chapter 2.

It is not uncommon for some sort of computational optimisation to be involved in the design of such structures. Computational optimisation can take a large number of different forms, but much of the work on metaferrites specifically has used a particular type of optimisation technique known as the Genetic Algorithm.

At present, there have been several papers detailing the simulation of metaferrites, but very few, if any, that follow this line of thought through to actual fabrication. The most common method of designing these structures is by the use of the Genetic Algorithm technique, which allows for specific goals in regards to material responses to be focused upon.

Ferrimagnetic metamaterials, also known as Metamaterial Ferrites, Metaferrites or Artificial Magnetic Conductors, are a moderately new technology. Their development came about as an attempt to mimic the electromagnetic properties of ferrimagnetic materials, i.e., ones that do not rely on their ability to actually exert forces on objects. Therefore, a key feature of metaferrites is that they are able to provide magnetic properties of materials while being “magnet free”.

In practice, to design “metaferrites” typically involves producing periodic structures with similar permeability and permittivity profiles, so that their characteristic impedance can be matched to that of the free space. Metaferrites often take the form of high-impedance frequency selective surface structures [8] designed in such a way as to mimic the properties of a thin ferromagnetic slab backed by PEC, but there are many other forms they could potentially take.

This thesis chronicles an attempt to design metamaterial ferrites, exploring various possible methods based on computational and optimisation techniques, novel material processing and manufacturing techniques. A particular focus was given to the genetic algorithm method. Once designed by this method, the structures were fabricated through several different techniques using various materials. The results of this fabrication were analysed and compared to those predicted by simulations in order to assess the quality of both, and this analysis informed the further development of the project.

Ultimately, of the various materials and substrates simulated, it was determined that copper on an FR4 substrate would be most optimal for the purposes. This was largely because it was the material whose manufacturing process proved easiest to employ consistently, which is vital for providing consistent results in the finished product. There were other materials, such as silver ink, which performed at a comparable level in simulation but whose application proved difficult and expensive.

The motivation for this project is the range of applications to which metamaterial ferrite surface can be applied. From antennas to absorbers, metaferrites can provide solutions to many problems and their development could open the doors to new technologies in the future.

Genetic Algorithms are a tried and tested optimisation method that, at its most basic, works in a manner not entirely dissimilar to the way a farmer might selectively breed livestock, hence the name. It begins by taking a large sample of possible solutions, and then determines which of them is the closest to a given trait that one hopes to maximise. It generally does this by referring to something called a Fitness Function, which is a function which has a minimum (or maximum, in some cases) at the desired value. Once it has ranked the solutions by quality it then forms a second generation by combining the best solutions together in a manner that will hopefully result in improved performance. This continues over the course of multiple generations, with each hopefully being closer to the goal than the last, until a solution that corresponds to the desired parameters is arrived upon.

A similar nature-inspired optimisation technique is the Particle Swarm [10]. This technique is intended to replicate the behaviour of large groups of animals, such as a flock of birds seeking food. Each particle moves independently, but at the same time is programmed to slowly converge on the best solution any other particle has found in a given solution space. Metaheuristic algorithms such as particle swarms have several advantages and disadvantages. One advantage is that PSO makes very few assumptions about the problem it is presented with, which makes it an extremely versatile technique. It also has the capacity to search extremely large design spaces. The primary downside of the technique, and of metaheuristic algorithms in general, is that there is no way to guarantee that it will ever find the global optimum of the problem.

An Artificial Bee Colony (ABC) algorithm [11] is somewhat similar to a particle swarm, but mimics the foraging techniques of bees specifically, which are somewhat more

organised than that of birds or other flocking animals. Within a real beehive's society there are a number of "scout" bees, whose job it is to wander around more or less at random to find new potential food sources. If they find any they will return to the hive and communicate the location and quality to the other bees via a "waggle dance", before returning to the food source with a small group of other bees in order to harvest it. An Artificial Bee Colony [11] works similarly. It consists of half "employed" bees and half "onlooker" bees, with a number of bees equal to the solutions being analysed. The onlooker bees take the role similar to that of the Fitness Function of genetic algorithms described earlier, evaluating the quality of solutions and directing employed bees to "gather" solutions around the best ones. A subset of employed bees are "scout" bees which, much like their real-world counterparts, move around solutions randomly in order to allow the algorithm to find solutions better than its current crop.

This thesis will primarily focus on discussion of the Genetic Algorithm technique, as this was a technique that had previously been applied to the problem with a degree of success, and it was felt that this would provide a foundation on which to build a novel expansion of the topic. For this reason genetic algorithms were the technique ultimately used in the generation of metaferrite designs. Further details of these techniques, as well as a detailed recount of the process by which a bespoke algorithm was constructed, are presented in chapter 3.

Genetic Algorithms and the various Particle Swarm techniques belong to a class of algorithms collectively known as Machine Learning [11]. In a more general sense, Machine Learning is a term that refers to a variety of computational techniques that have an ultimate goal in common: they are designed to allow computers to autonomously process information and make decisions that they were not explicitly programmed to make. In other words, it forms the basis of artificial intelligence.

Aside from those already mentioned, machine learning techniques also include Cluster Analysis, Dimensionality Reduction, Support Vector Machines, Artificial Neural Networks and Decision Trees [12], [13], [14], [15]. Each of these techniques is further subdivided into several more specialised forms of machine learning designed to focus on a particular type of problem. When a single technique proves insufficient, multiple different methods can also be combined together. This technique is known as Ensemble Learning [16], and it has the advantage of providing a greater predictive accuracy much of the time.

Among the aforementioned alternate machine learning techniques, cluster analysis is most relevant here. Cluster analysis is a process by which the computer can sort data into smaller sets based on various traits. As a form of unsupervised learning these traits would be determined by the program itself through so-called “similarity measures” [17], which simply evaluate traits that certain members of the data population have in common. Cluster analysis can be used to increase the efficiency of a genetic algorithm by identifying key features of generated solutions and mapping them to the generated results. This would allow the computer to predict what result a given solution would give without analysing it directly, saving it from wasting time on non-viable candidates.

Given the high-variable nature of the problem, employing dimensionality reduction would also be an option. Dimensionality reduction is a technique by which high-dimensionality data (that is, data with a high quantity of interconnected variables) is simplified by identifying which of those variables is least relevant, and has least effect on the data, and systematically ignoring them [18]. This can lead to less precise results, but in some cases the approximation is necessary in order to render other techniques viable, such as the aforementioned genetic algorithms and cluster analysis. Both of these require low-dimensionality problems, as too much complexity will increase the computational load at an exponential rate.

The other forms of machine learning also have potential. Neural networks are most often used in data mining and search engines, compiling vast quantities of data and can reach such complexity that not even their creators can entirely explain how they work. This is a drawback of machine learning in general, of course, and one that is very much amplified as the complexity of the algorithms increases, but nonetheless applications of this technique could be explored in future work.

1.3: Contributions

This thesis deals with the construction of a genetic algorithm intended to design metamaterial ferrites for use with antennas and other similar devices.

Several of the structures designed via genetic algorithm were fabricated using a variety of methods, in order to validate the results of simulation and to provide insights into how to improve it. It also served to guide the initial parameters of the simulation in matters such as materials.

Two different conducting materials were used for this fabrication: Silver and Copper. The silver took the form of silver ink, deposited on a PET substrate by the inkjet printing method. The copper structure was instead made by applying a thin layer of copper to an FR4 substrate and carefully removing metal until only the desired design was left behind.

In both cases, a new simulation had to be made to design the structure in question, as the differing materials involved meant that the same design was not guaranteed to have the same response for both materials. This also meant that a direct comparison of the two methods using the same structure was impossible.

1.4 Structure of the Thesis

The thesis is organised as follows:

Following this first introductory chapter, Chapter Two deals with an overview of metaferrites and artificial magnetic conductors. This includes a review of several other papers that have previously covered the topic. A particular focus is given to those primarily dealing with the design similar to the High-Impedance Frequency Selective Surface, along with attempts to obtain optimized results involving other antenna parameters such as size reduction, bandwidth and gain enhancement.

The third chapter concerns itself with various computational optimization techniques, including Genetic Algorithms. It also details an account of the development of a novel genetic algorithm design and several macros to integrate it with commercial software for the purposes of building and analyzing arbitrary electromagnetic structures. It presents a fairly comprehensive rundown of the programming process, as well as some design examples. This chapter also includes the procedure by which the genetic algorithm's function was tested and evaluated.

Chapter Four concerns the development of the particular design of metaferriite produced by the proposed design and optimisation techniques, along with the theoretical basis for the parameters chosen for the final demonstration. It also includes the detail of several measurements to validate the design and optimisation approach with several metaferriites samples.

The fifth chapter deals with the design of hyperuniform metamaterial structures, focusing on a comparison of the performance of angular dipole antennae when arranged in a

periodic, random or hyperuniform distribution. There is particular attention given to the farfield response of these structures.

The sixth chapter details the fabrication procedures used to build physical representations of those designs generated by the genetic algorithm that were deemed to be of sufficient quality, as well as the results derived from the analysis of these structures and the insights thereby gained.

Finally, the seventh chapter provides a summary of the project, assessing what worked, what did not, and evaluating the results produced. It also concerns itself with a discussion of what possible directions that continuations of this project in the future. Several options for future topics of study are presented; along with what scenarios would likely drive projects down that particular road.

1.5: References

- [1] A. Okamoto, "The Invention of Ferrites and Their Contribution to the Miniaturization of Radios", *2009 IEEE Globecom Workshops*, November 2009
- [2] B. C. Carter, M. G. Norton, *Ceramic Materials: Science and Engineering*, Springer Science & Business Media, October 2007
- [3] Z. Ullah, S. Atiq, S. Naseem, "Influence of Pb doping on structural, electrical and magnetic properties of Sr-hexaferrites" *Journal of Alloys and Compounds* 555:263–267, November 2014
- [4] R. Valenzuela, "Novel Applications of Ferrites", *Physics Research International Volume*, March 2012
- [5] Apex Magnets, "The Advantages and Disadvantages of Different Magnets", <https://www.apexmagnets.com/news-how-tos/advantages-disadvantages-different-magnets/> September 2017, visited July 2018
- [6] A. M. Turing, "Computing Machinery and Intelligence", *A Quarterly Review of Psychology and Philosophy*, VOL. LIX. No.236. October, 1950
- [7] Barricelli, Nils, "Numerical examples of evolution processes", *Methodos*, pp. 45-68, 1954
- [8] D. J. Kern, D. H. Werner, M. Lisovich, "Metaferrites: Using Electromagnetic Bandgap Structures to Synthesize Metamaterial Ferrites" *IEEE Transactions on Antennas and Propagation Volume 53 Issue 4*, April 2005
- [9] K. Bi1, G. Dong, X. Fu, and J Zhou, "Ferrite Based Metamaterials With Thermo-Tunable Negative Refractive Index" *Appl. Phys. Lett.* 103, 131915, 2013
- [10] J. Kennedy and R. Eberhart, "Particle Swarm Optimisation", *Neural Networks*, 1995
- [11] D. Karaboga, "An Idea Based on Honey Bee Swarm For Numerical Optimization", *Technical Report - TR06*, 2005
- [12] R., Lior, and O. Maimon. "Clustering methods." *Data mining and knowledge discovery handbook*. Springer US, 2005.
- [13] H.P. Kriegel, E Schubert; A. Zimek, "The (black) art of runtime evaluation: Are we comparing algorithms or implementations?". *Knowledge and Information Systems*. 52: 341–378., 2016
- [14] S. T. Roweis, Lawrence K. Saul "Nonlinear Dimensionality Reduction by Locally Linear Embedding", *Science* 22, December 2000
- [15] J. J. Hopfield "Neural networks and physical systems with emergent collective computational abilities." *Proc. NatL Acad. Sci. USA Vol. 79*, pp. 2554-2558, April 1982

- [16] D. Opitz, R. Maclin, "Popular ensemble methods: An empirical study". *Journal of Artificial Intelligence Research*. 11: 169–198. 1999
- [17] David E Goldberg, John H Holland, "Genetic Algorithms and Machine Learning". *Machine Learning*. 3 (2): 95–99., October 1988
- [18] D. Pfitzner, R. Leibbrandt, D. Powers,. "Characterization and evaluation of similarity measures for pairs of clusterings". *Knowledge and Information Systems*. Springer. 19: 361–394., June 2009

Chapter 2: Review of Metaferrites

2.1: Overview

Metamaterial ferrites or metaferrites, sometimes also referred as to artificial magnetic conductors, consist of periodically repeating structures that mimic the properties of magnetic materials, but do not actually contain any [1]. They are characterised by a frequency band over which surface waves cannot propagate and very high surface impedance within a limited frequency range where the tangential magnetic field is small [1]. Neither of these properties are typically found in nature, and it is for this reason that metaferrites are considered a class of metamaterials [2].

Metaferrites have a number of uses in applications such as low profile, high gain antennas [3] and improved microwave cavities for use in atomic clocks [4]. They typically consist of periodic structures of two or more layers of metal sandwiching a dielectric substrate [2], [4], [5]. The exact pattern of the structure influences its electrical properties and it is this pattern design that is typically the subject of research for metaferrites. This thesis is no exception, and this aspect is elaborated in great detail throughout

2.1.1: Metaferrites as High-Impedance Surfaces

It is common for metaferrites to take the form of a high-impedance surface [6]. High-impedance surfaces work by acting as a densely packed series of capacitors and inductors, forming parallel resonant LC circuits (as seen in figure 2.1). [7] These circuits act as electrical filters, forcing the current to repeatedly charge and discharge the capacitor components, thereby slowing the flow of current and creating the high surface impedance that the structure is named for [8]. This means that when used as the back plate of a one-directional transmitter it will provide the benefit of preventing phase shift during radiative reflection, eliminating the destructive interference which normally occurs if the antenna is placed less than half a wavelength away [9]. Since the antenna can be placed much closer to the back plate, the entire device can in turn be made with a much lower profile than is currently possible with conventional technologies [10].

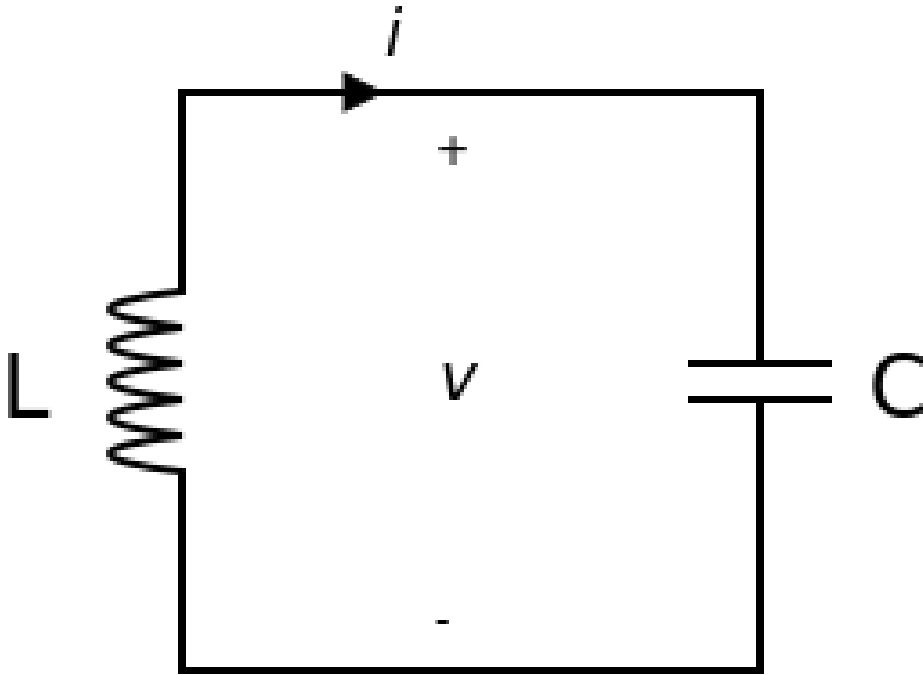


Figure 2.1: Simple diagram of an LC circuit [11] showing the capacitor (right) and inductor (left)

One of the simplest possible forms that a high impedance surface can take consists of a repeating pattern of squares connected through a dielectric substrate to a solid backing plate. [12] This structure is characterised by a high surface impedance at particular resonance frequency, which can be adjusted through the size and spacing of the square patches as seen in figure 2.2.

The downside to such a simplistic design is the limited design space it grants. There are essentially just a few interconnected variables, such as the size and number of the square tiles and how far apart they are. A far greater range of material responses can be achieved with more complex high impedance structures [2], [4], [5].

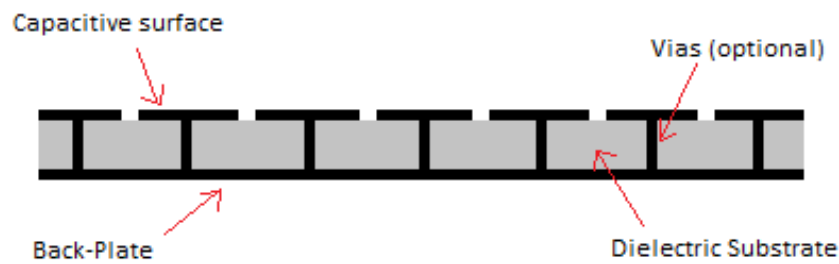
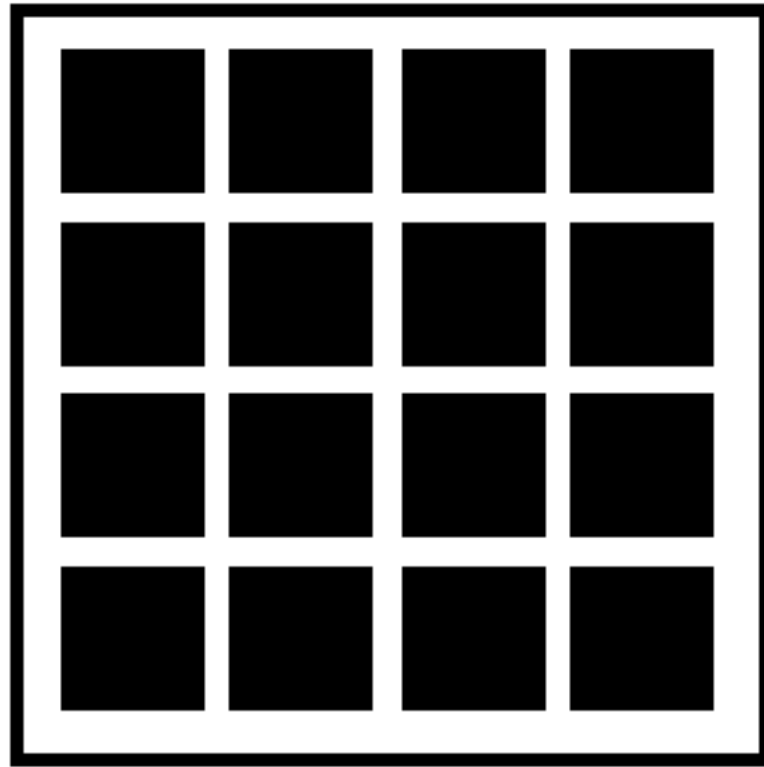


Figure 2.2: Typical example of a simple high impedance surface structure

2.1.2: Theoretical Basis

Mathematically speaking, a metaferrite should behave as a ferrite does in the context of the application it is being used for. This means that different structures may be considered metaferrites in different contexts. The ones designed in this thesis aimed to enhance the performance of antennas placed in front of them by enhancing bandwidth and reducing radar cross section.

The reflection of a ferrite of this type is related to its surface impedance by the equation

$$Z = Z_0 \times \frac{1 + S_{11}}{1 - S_{11}} \quad (2.1)$$

and so the key to producing a structure that mimics the ferrite in performance is to manipulate the surface impedance in such a way that the two match. This can be done by manipulating the permeability of the structure via its surface resistance and reactance [2], which is given by:

$$\mu'_r = \frac{X_{S1}}{\eta_0 \beta_0 d} \quad (2.2)$$

$$\mu''_r = \frac{R_{S1}}{\eta_0 \beta_0 d} \quad (2.3)$$

where μ'_r is the real permeability of the structure and μ''_r is the imaginary permeability. These are in turn derived from the surface impedance via

$$Z_{S1} = R_{S1} + jX_{S1} \quad (2.4)$$

One method by which metaferrites are commonly designed is through the use of computational optimisation [13]. This is useful due to the increased complexity of the structures that may be required to produce certain material responses [2], [4], [5], [14]. In some cases the amount of potential variables explodes to such a degree that analytical solutions become all but impossible, and there are many cases where human creativity cannot match the brute force trial and error approach a computer can muster.

Computational optimisation was the approach used in a paper by Douglas Kern, Douglas Werner and Mikhail Lisovich. [2] In it, the three outline a method by which they designed a metaferrite structure of high impedance types via the Genetic Algorithm method.

Their objective was to minimise the loss of the system whilst setting the permeability to a pre-defined value of 10. To this end, they determined a fitness function to drive this goal such that

$$FF = - \left[(\mu'_r - 10)^2 + \mu''_r \right] \quad (2.5)$$

As with any Genetic Algorithm, the objective is to find the global minimum of this function. This will correspond to the design with the desired parameters.

Assuming that this equivalent circuit model holds, R_{S1} and X_{S1} can be calculated using the equations

$$R_{S1} = \frac{\omega LA}{A^2 + B^2} \quad (2.6)$$

$$X_{S1} = \frac{\omega LB}{A^2 + B^2} \quad (2.7)$$

where

$$A = \frac{\omega L}{R} \quad (2.8)$$

$$B = 1 - \left(\frac{\omega}{\omega_0}\right)^2 \quad (2.9)$$

$$\omega_0 = \frac{1}{\sqrt{LC}} \quad (2.10)$$

Here R, L and C are the resistance, inductance and capacitance of the structure.

By comparison, the surface impedance for the structure which Kern *et al* was attempting to emulate is given by

$$Z_{S2} = Z \tanh(\gamma d) \quad (2.11)$$

Where

$$Z = \eta_0 \sqrt{\mu_r} \quad (2.12)$$

$$\gamma = j\beta_0 \sqrt{\mu_r} \quad (2.13)$$

Since having the latter emulate the former is the goal here, the next step is to set these two equations to the same value of surface impedance, so the characteristic equation becomes

$$R_{S1} + jX_{S1} = \eta_0 \sqrt{\mu'_r - j\mu''_r} \tanh\left(j\beta_0 d \sqrt{\mu'_r - j\mu''_r}\right) \quad (2.14)$$

Taking the small angle approximation one can assume that $\tanh(x) \approx x$, which allows this equation to be split into equations (2) and (3).

Kern *et al*'s Genetic Algorithm using these parameters has led to the design of such a structure with the unit cell depicted in figure 3. This structure has a characteristic eightfold

symmetry, brought about by the method of its construction. This is useful, as it reduces the computational load on the GA. As such, this feature was utilised in the designs created by the genetic algorithm developed in this project.

Kern *et al*'s paper [2] was largely theoretical in nature. None of the structures were actually fabricated, and as such this project saw an opportunity to extend and validate this research by both simulating a metaferrite structure using similar methods and design parameters and also fabricating said structures for practical applications.

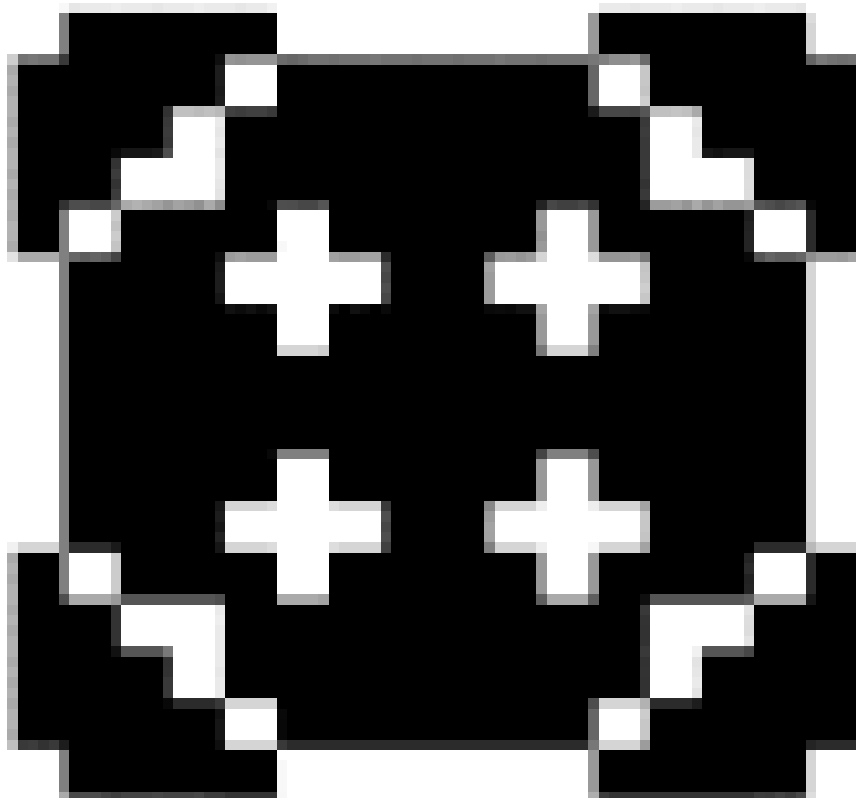


Figure 2.3: Unit cell with size 1.403 x 1.403 cm, a dielectric constant of $12.79 - j 0.0006$ [1] and a substrate thickness of 4.65 mm. These parameters combine to give it the same electrical response of a slab of ferrite material about 1mm thick.

2.1.3: Alternative Designs

This paper was also included as part of a dissertation [15] that went on to expand the idea into matched impedance metamaterials. This involves so-called magneto-dielectric substrates, which are defined as being those that have permeability and permittivity equal for a given frequency or narrow band thereof. The design procedure itself was similar to that used in standard metaferrites.

Another avenue pursued in that particular dissertation [15] concerned reconfigurable Artificial Magnetic Conductors, which like the previous examples were based on Frequency Selective Surface (FSS) structures. The reconfigurable nature of the structures is achieved by building them in the form of a fixed 8x8 grid of metal crosses linked by a matrix of switches which can be used to change the filter response of the device. The genetic algorithm employed to design them was constrained by a fixed unit cell size, dielectric constant and substrate thickness, and cell geometry was only permitted to change based upon the configuration of the aforementioned binary switches.

These parameters allowed the same structure to mimic the configuration derived for multiple mutually-exclusive parameters, switching between them at the touch of a button.

The fitness function chosen for the metaferrite design was given by

$$FF = -\max(\phi_{TE}, \phi_{TM}) \quad (2.15)$$

Where ϕ_{TE} and ϕ_{TM} are the reflection phases of the TE and TM response, respectively. Optimising the structure to act as an absorber, on the other hand, requires a different variant of the fitness function, this time given by equation 15

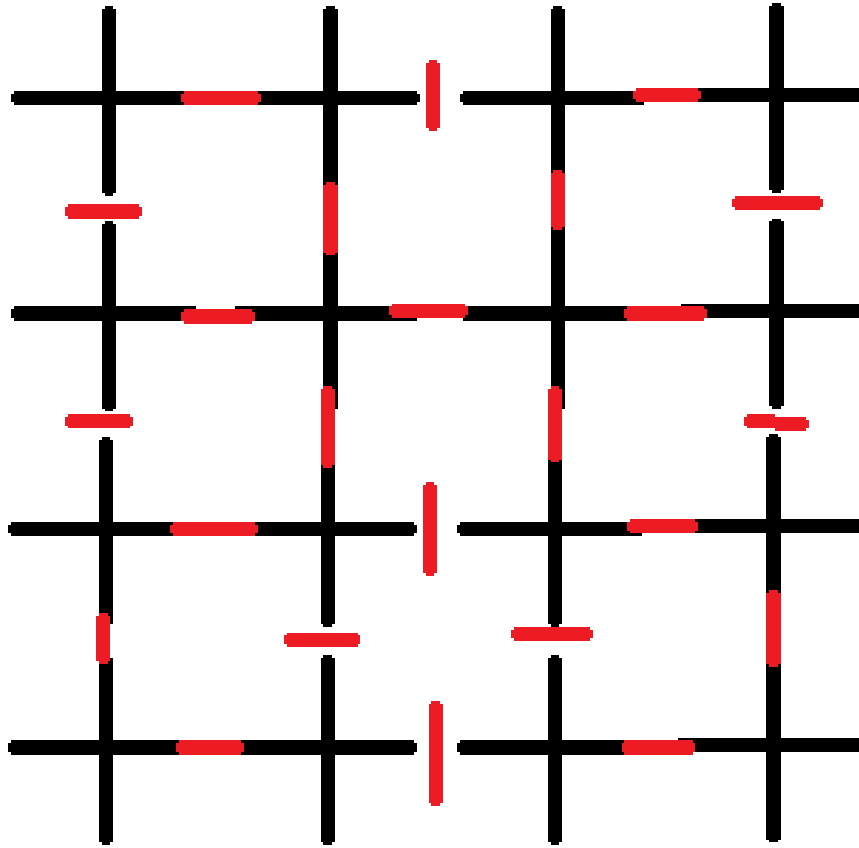


Figure 2.4: Example of reconfigurable metaferrite structure showing fixed cross-shaped components (black) and adjustable switches (red)

$$FF = \frac{1}{\sum_{freq} [0.2|\Gamma| + 0.8\angle\Gamma]} \quad (2.16)$$

which, as one can see, requires an entirely different set of input parameters, this time based on the reflection coefficient directly.

2.1.4: Other Designs

Another variant of metaferrite were presented in Daniel Sievenpiper's thesis on High-Impedance Electromagnetic Surfaces. [4] An attempt was made to replicate the design presented in [4] using CST's eigenmode solver, as at the time there was consideration towards using a solver of this nature throughout the project. This notion was ultimately put to

one side once the GA program approached completion, as the frequency domain solver proved more in line with the project's needs.

The structure dubbed as “Sivenpiper’s Mushroom” is shown in figure 5. The hexagonal pattern of the structure meant that the standard square unit cell would be difficult to implement. It could theoretically be done by carefully selecting a square portion of the structure to copy, but the precision required to ensure proper tessellation would be impractical.

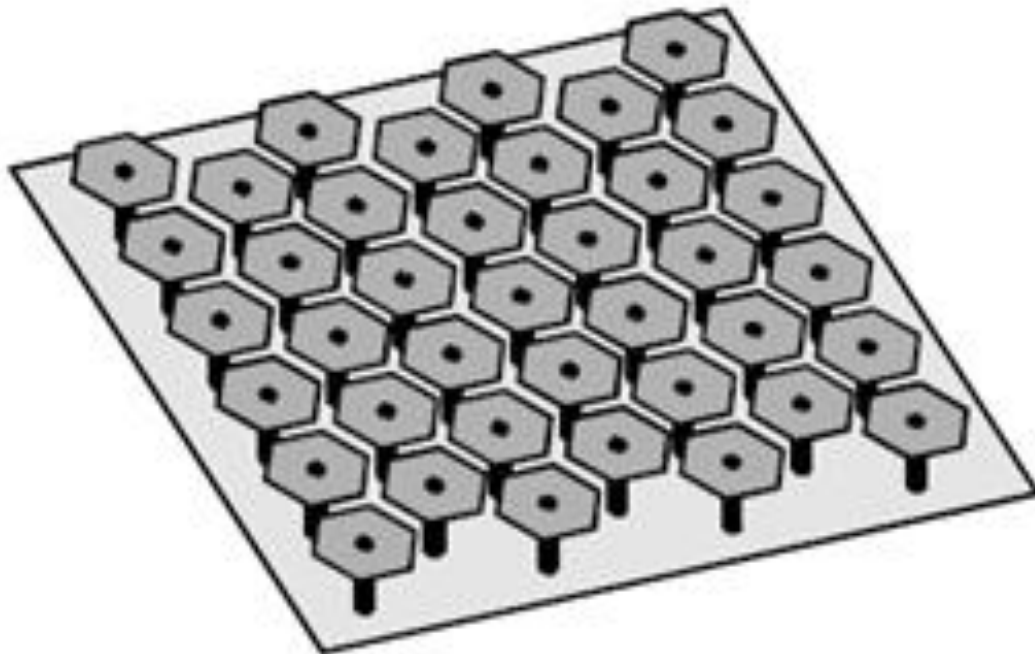


Figure 2.5: Perspective view of a hexagonal two-layer printed “mushroom” design, as seen in Sievenpiper’s thesis paper. [3]

As such, a modified approach was deemed to be necessary. To that end two competing solutions to that issue rose to prominence, and it was decided that it would be worthwhile to implement both in parallel in order to properly assess their respective merits and flaws.

The first solution was to use a rhombic unit cell shape rather than a rectangular one. Doing this also required that unit cell structure be designed such that it would tessellate properly when repeated across the boundaries to form the full structure. This ultimately led to a three-hexagon structure carefully aligned with the edges of the rhombic boundary.

Said structure is shown in figure 2.6, below.

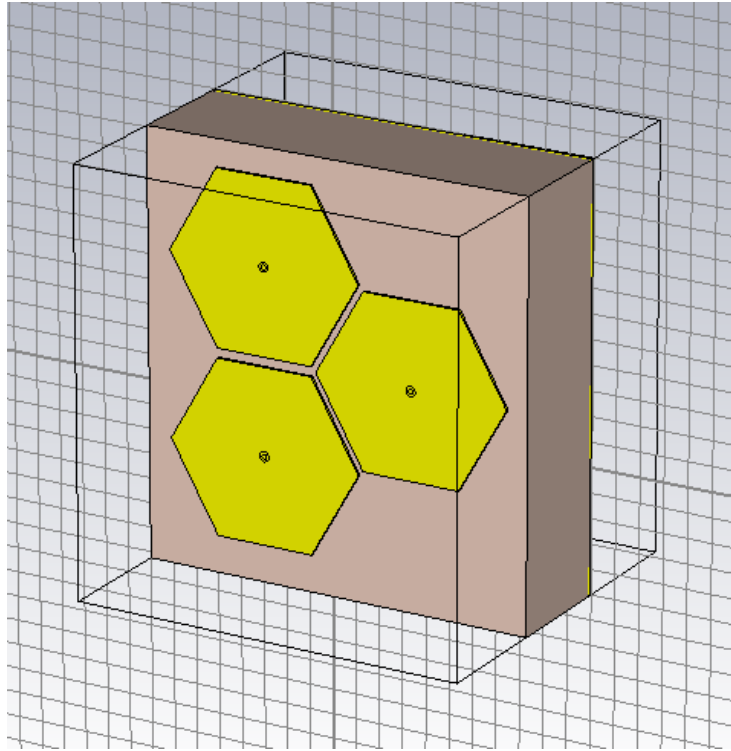


Figure 2.6: Rhombic unit cell structure.

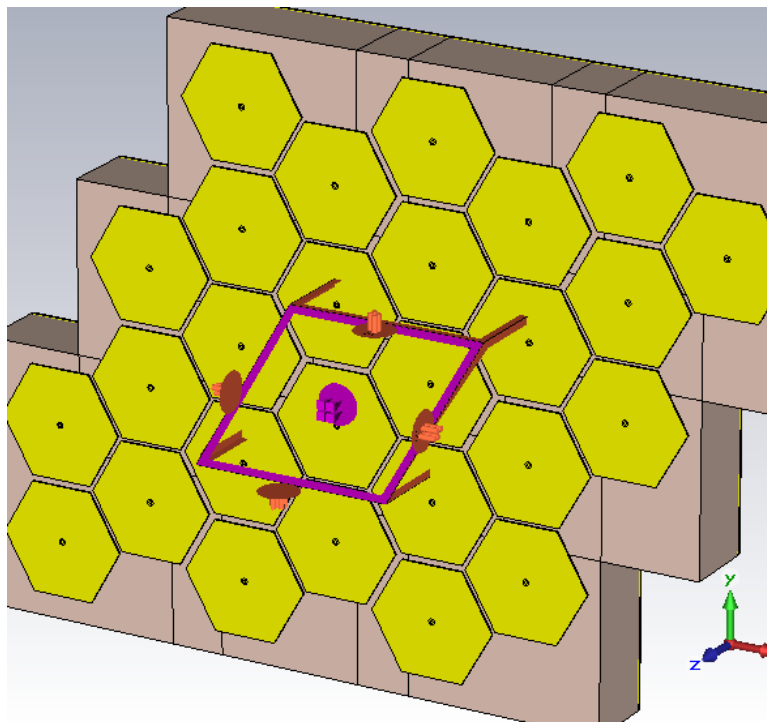


Figure 2.7: The structure designed to ensure tessellation whilst keeping the hexagonal panels as single objects (below)

A unit-cell design is inherently not quite the same as the original, since it goes on infinitely instead of having a finite size, and in the rhombic unit cell specifically there are several points where the substrate overlaps with the substrate of adjacent cells. The results gathered by this simulation, shown in figure 2.11 and figure 2.12, seemed to indicate that these were not huge problems, but there was some evidence of an amplification of the results brought about by this compromise.

The other solution involved preserving a more conventional rectangular cell design by slightly altering the structure itself. This took the form of strategically bisecting the hexagonal sections and vias such that the same pattern would be constructed when the unit cell was expanded to the full structure. This had the downside of introducing small discontinuities in the pattern at the edges of the unit cells.

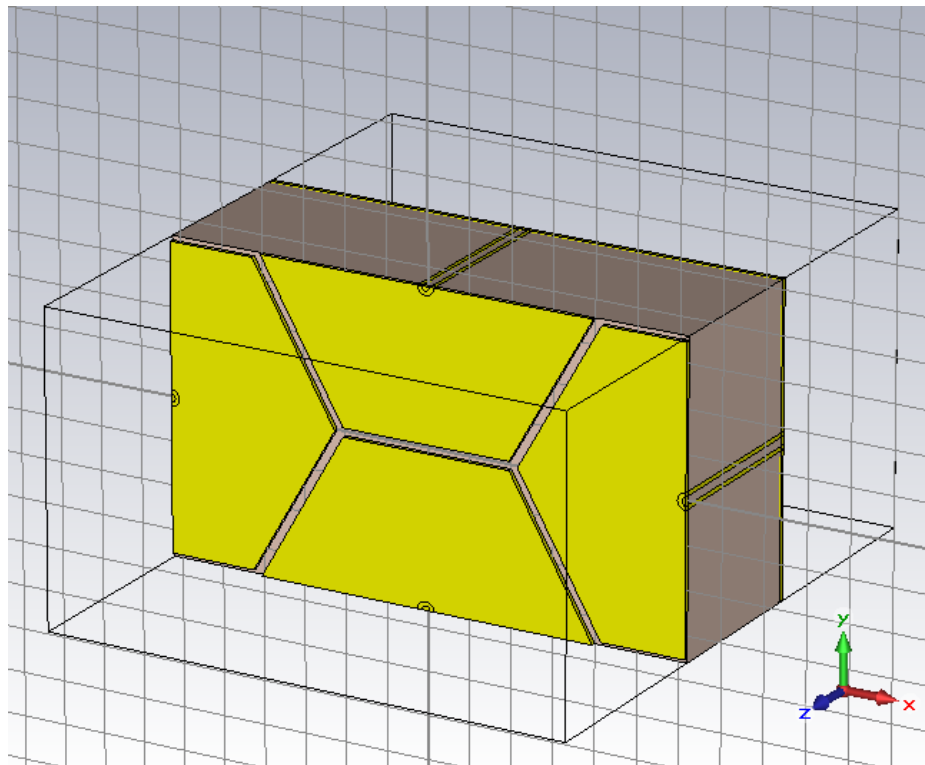


Figure 2.8: Rectangular Unit cell structure

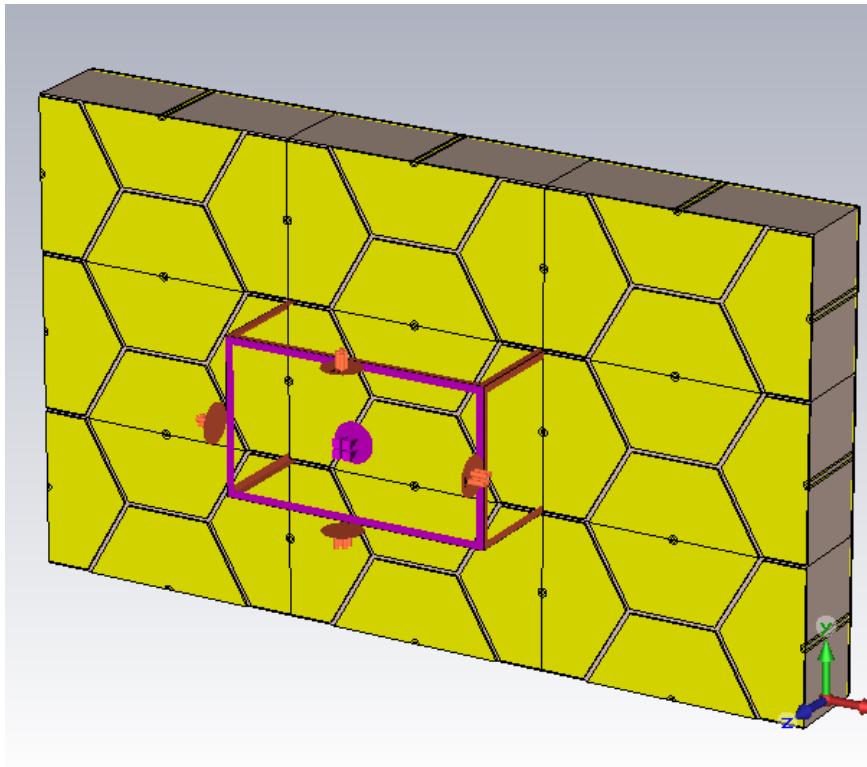


Figure 2.9: A demonstration of the continuity of hexagonal panels achieved using rectangular cell structure

Both of the methods presented here rely on approximations that distort the design of the surface in some way. Naturally, these approximations could have a knock-on effect in the simulated results. One of the purposes of this exercise was to ascertain which method produced the greatest deviation from reality, and how large those deviations were.

To this end, a dispersion diagram was constructed for each method. In both cases, this was compared to one theoretically-derived via the Effective Medium Model, as seen in figure 2.10. In both cases, the dispersion bears a resemblance to that model, as shown in figures 2.11 and 2.12, although only the rhombic unit cell could really be considered a strong match, with clearly identifiable analogous features. On a quantitative level, both predict a resonance frequency for their respective structure at around 3 to 4 GHz. This is consistent with other measurements taken of the two. Both diagrams also predict several more TE waves at higher frequencies. The rhombic unit cell response bears closer resemblance to the theoretical model than the rectangular one regarding the overall shape of these TE response diagrams, possibly as a result of the lack of discontinuities in the metallic sections.

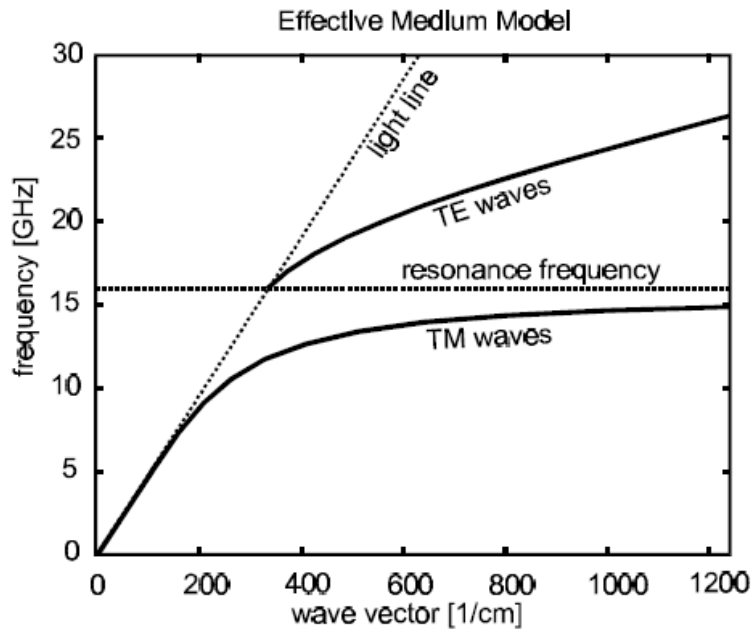


Figure 2.10: Prediction of effective medium model [3] of the relationship between wave vector and frequency for a structure like that in Sievenpiper's paper.

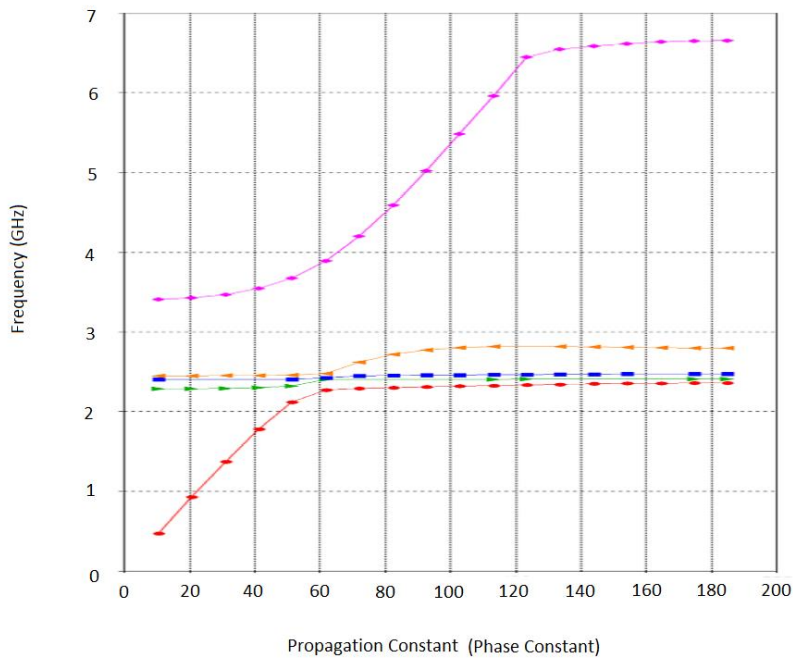


Figure 2.11: Results of CST's Eigenmode solver analysis of the rhombic unit cell hexagonal pattern structure.

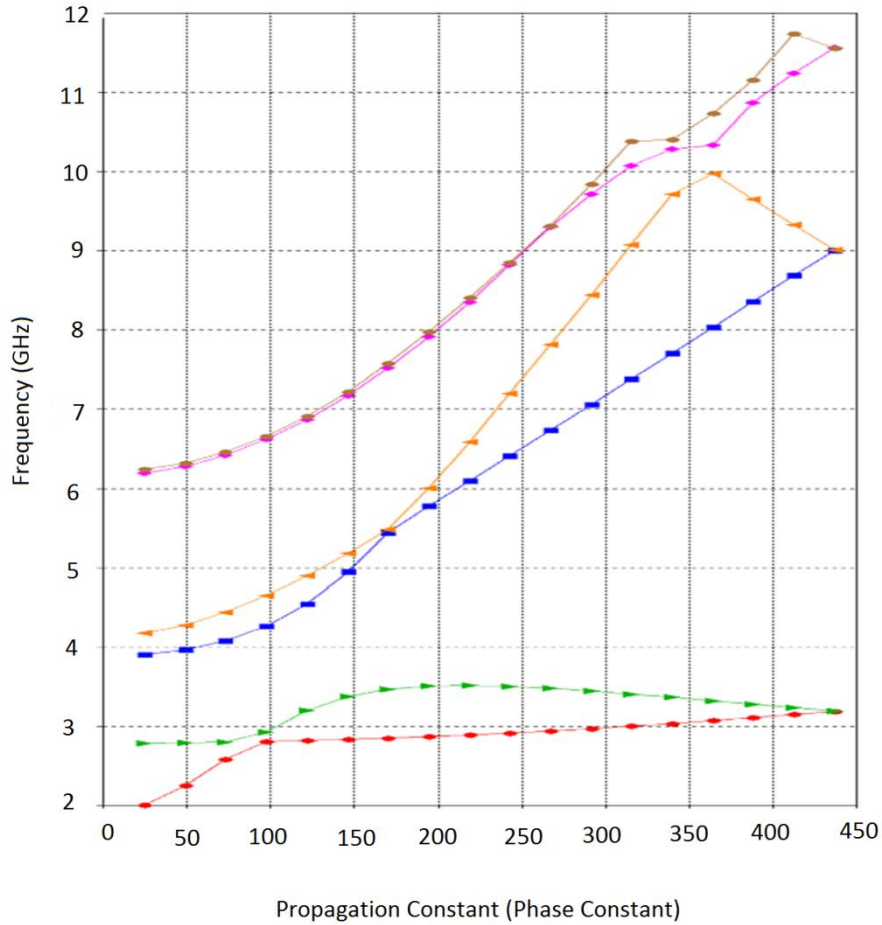


Figure 2.12: Results of CST’s Eigenmode solver analysis of the rectangular unit cell hexagonal pattern structure.

2.2: Summary

This chapter deals with the theory behind metaferite structures, with a focus on high impedance surfaces which comprise the bulk of such structures attempted in the past. High impedance surfaces can be understood using the parallel LC circuit model, which casts them as an array of interconnected inductors and capacitors.

The chapter then details several papers that focused on their design and manufacture, along with an attempt to replicate one such structure in CST, which took the form of an array of hexagonal tiles connected to a backplate with vias. Due to the shape of the structure impeding the use of a standard unit cell design, an exploration was made of two alternate methods of using an infinitely repeating unit cell structure, either with a conventional

rectangular cell which bisected the surface and a rhombic unit cell that left it intact. These two methods were compared and evaluated.

2.3: References

- [1] H. Zhou, F. Xu, "Artificial magnetic conductor and its application", *2013 Proceedings of the International Symposium on Antennas & Propagation*, October 2013
- [2] D.J. Kern, D. H. Werner, M. Lisovich, "Metaferrites: Using Electromagnetic Bandgap Structures to Synthesize Metamaterial Ferrites" *IEEE Transactions on Antennas and Propagation Volume 53 Issue 4*, April 2005
- [3] S.A. Attachi, C. Saleh, M Bouzouad, "Microstrip Antenna Gain Enhancement With Metamaterial Radome", *Applied Physics A 123(1)* January 2017
- [4] D. F. Sievenpiper, "High-Impedance Electromagnetic Surfaces: A dissertation in partial satisfaction of the requirements for the degree of Doctor of Philosophy in Electrical Engineering," 1999.
- [5] O. Rybin, "Effective microwave magnetic response of two-component metaferrite" *International Journal of Applied Electromagnetics and Mechanics*, vol. 40, no. 3, November 2012
- [6] F. Linot R. Cousin X. Begaud, "Design and measurement of High Impedance Surface" *Proceedings of the Fourth European Conference on Antennas and Propagation*, April 2010
- [7] G. Expósito, J.M. Fernández P. Padilla M. Sierra-Castaner, "EBG Size Reduction for Low Permittivity Substrates" *International Journal of Antennas and Propagation 2012:1-8*, December 2012
- [8] N. Friedrich, "High-Impedance Electromagnetic Surface improves antenna performance". *Microwaves & RF magazine*. pp. 62, May 2007
- [9] D. Sievenpiper, L. Zhang, R.F.J Broas. N.G Alexopolous, E Yablonovitch,.; et al. "High-Impedance Electromagnetic Surfaces with a Forbidden Frequency Band" *IEEE Transactions on Microwave Theory and Techniques*. 47, November 1999
- [10] J. R Sohn, K. Y. Kim, H-S Tae, H. J Lee, et al.. "Comparative study on various artificial magnetic conductors for low-profile antenna" *Progress in Electromagnetics Research*. 61: 27–37, 2006
- [11] First Harmonic, <https://commons.wikimedia.org/w/index.php?curid=21991221>, October 2012, accessed June 2018
- [12] Z. Hao, S. Saadat, H. Mosallaei, "FDTD-SPICE for Characterizing Metamaterials Integrated with Electronic Circuits" *International Journal of Antennas and Propagation 2012(1687-5869)*, August 2012
- [13] X.S. Yang S. Koziel, "Computational Optimization: An Overview" *Computational Optimization, Methods and Algorithms pp 1-11*, June 2011

[14] Z.Bayraktar, M. D. Gregory, X. Wang, D. H. Werner, “Matched Impedance Thin Planar Composite Magneto-Dielectric Metasurfaces”, *IEEE Transactions on Antennas and Propagation*, Volume 60 Issue 4, April 2012.

[15] D. Kern, “Advancements in Artificial Magnetic Conductor Design for Improved Performance and Antenna Applications”, *Dissertation Abstracts International*, Volume: 70-11, 2009

Chapter 3: Computational Optimisation Techniques

3.1: Overview

Computational optimisation is a powerful technique that allows researchers to design new and innovative solutions to problems by rapidly generating and testing potential solutions at a rate that no human could ever accomplish [1]. It essentially allows for automated “trial and error” exploration of problems at an industrial scale, potentially providing unique solutions that many would otherwise not even think to attempt [1].

By and large, all computational optimisation techniques have two prominent features: a means of generating solutions, and a means of testing and improving those solutions in order to converge on a solution that generates the best possible results [2], [3]. The manner in which they perform these tasks can vary dramatically, and a few of the more common methods are detailed below.

3.1.1: The Newton-Raphson Method

One of the oldest methods used in computational optimisation is Newton’s method, which is also known as the Newton-Raphson method. Named after Isaac Newton and Joseph Raphson, the Newton-Raphson method can be regarded as a root-finding algorithm [4]. The Newton-Raphson method actually pre-dates computers; it was first published in 1865 by a mathematician named John Wallis. Wallis published it under the name of Newton’s method [5], as it was based on Newton’s work on polynomials. It was simplified and popularised by Raphson five years later [6], resulting in its alternate moniker.

The Newton-Raphson method is efficient and effective, but it relies on the variables sought being a) real and b) the minima of a function capable of being differentiated at least twice [7]. Naturally, these requirements mean that it is not applicable to all problems, but for those that can make use of it, it can be quite simple to apply.

The Newton-Raphson method works by taking information about the gradient of a function (generally derived by differentiation) and using that to identify the most direct route to the function’s stationary points, which characterise both maxima and minima. The formula takes the form

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad (3.1)$$

where x_n is the previously derived x_{n+1} , and f and f' are the function and its derivative, respectively.

To start the algorithm off an initial value is required, of course, and one is generally estimated based on where the user thinks the minima will be, setting the first x_n to be somewhere near that point [5]. The resulting x_{n+1} is then re-inserted producing a new value, and the process is repeated until the minima is found [5]. The Newton-Raphson method does have one limitation: generally, only one solution will be found, even if the function has multiple minima. Which this is will depend heavily on the initial point chosen. [8]

3.1.2: The Quasi-Newton Method

If the derivative of the function is unavailable, or the Newton-Raphson method is inapplicable for some other reason, there are a number of other, similar methods that can be used instead. These are collectively known as Quasi-Newton methods [9].

One form of Quasi-Newton method is known as the secant method. [10] The secant method operates on the same principles as the Newton-Raphson, but replaces the exact Jacobian function with an approximation, calculated via the formula:

$$x_n = x_{n-1} - f(x_{n-1}) \frac{x_{n-1} - x_{n-2}}{f(x_{n-1}) - f(x_{n-2})} \quad (3.2)$$

Which can also be written as:

$$x_n = \frac{x_{n-2}f(x_{n-1}) - x_{n-1}f(x_{n-2})}{f(x_{n-1}) - f(x_{n-2})} \quad (3.3)$$

This method requires the user to select two initial values rather than one [11], and with each successive step the previously calculated result and the one preceding it are used to calculate the new value [11]. The method's main weakness is that it is somewhat slower than the Newton-Raphson, which is the primary reason it is generally only employed when that method will not work [12].

Other quasi-Newton methods use different approximations, and naturally the exact parameters of said approximation can have a significant effect on the performance of a given technique [13].

3.1.3: Particle Swarm Technique

Another commonly-used type of optimisation method is known as the Particle Swarm technique. This is a much younger technique than Newton's method, being first used in 1995 by Russel Eberhart and James Kennedy in [14]. The Particle swarm technique focuses on the optimisation of nonlinear functions. It is computationally inexpensive, requiring only simple mathematical operators [15], and works by creating a population of candidate solutions (the "swarm" that the technique's name refers to) that explore an allotted space according to simple formulae, [15] somewhat like the way a flock of birds or school of fish will move. The particles are further guided by a combination of their own best position and the swarm's best position. If a particle stumbles onto a better solution than those known it will take control of the swarm as a whole, and in this way the technique will theoretically converge on the global minima of the search space [15].

The success of a particle swarm is heavily dependent of the parameters used to create it, such as the motions each individual member of the swarm takes and the various ways in which they can be linked [16], either with every member responding to every other member or merely responding to their nearest members or those of a smaller group [17]. In general, regardless of specifics, particle swarms are more computationally efficient than genetic algorithms [18]. Finding optimal parameters for a swarm has been subject to much research, including several attempts at what is known as Meta-Optimisation [19]. Convergence of a particle swarm can be defined as one of two different situations. Either it is the case where all particles have arrived at the same point, or it can instead be when all personal bests have achieved sufficient proximity to the local optimal value.

A variant of Particle Swarm is the Artificial Bee Colony, or ABC. [20] Proposed in 2005, it is based on the foraging behaviour of a honey bee swarm. Bees locate food in a manner for more organised than birds or other creatures after which conventional particle swarms are modelled. Instead, an ABC algorithm splits the particles into three groups; employed bees, onlookers and scouts [20]. The process begins by generating a potential solution or "food source" for each employed bee. The bee then goes to that solution, determines a neighbour source and then evaluates the "nectar amount", which is essentially the fitness of the solution. They then return to the "hive" where the quality of the solutions that were investigated is compared. Onlooker bees then select one of the higher-quality solutions and find another neighbour source to evaluate. At this point, some food sources are

abandoned, and they are replaced by new sources by the scout bees. The best source is logged and the process repeats.

The Artificial Bee Colony has a few advantages over the standard particle swarm model [21]. The dedicated “scout” bees provide a more reliable source of unexplored solutions, and the ability for the swarm to investigate multiple solutions with a high degree of difference simultaneously rather than having to slowly drift between them allows for greater efficiency [21].

3.1.4: Genetic Algorithm Technique

The technique used most prominently in this project, however, was the Genetic Algorithm method. In many ways, the Genetic Algorithm technique is the opposite of a Particle Swarm, since Genetic Algorithms are based on a competitive function, while Particle Swarms are based on a cooperative one [22].

Genetic algorithms were first proposed in the early 1950s by Alan Turing. [23] The primary adopters of the technique were biologists, who would typically use GAs to model the evolutionary process that inspired them [24]. Over the course of the next couple of decades use expanded to other problems such as those found in engineering, with sufficient success that the technique is most commonly used in that area today [25].

Genetic Algorithms work by mimicking the principles of natural selection. To begin with, one will typically create an initial population of solutions, either from approximations of what it is thought that the final design should resemble, or simply through random generation, and then run these potential solutions through a so-called fitness function to assess their quality.

Those deemed successful enough are randomly recombined or “bred” to form the next generation [26], generally with a few random “mutations” thrown in to allow the GA to explore the design space, as it is quite possible for the initial population to have missed the best possible solution in such a way that simply recombining the ones that were generated will never arrive at it. Introducing a random element prevents the algorithm from getting stuck at a local minimum.

Once a new population of solutions has been constructed from the old one, the entire process repeats, the new generation is tested, and the best are recombined [26]. The GA

continues on in this fashion until a solution that conforms to a goal set by the designers beforehand is reached. This can be very time consuming, with Genetic Algorithms running for dozens or even hundreds of generations before convergence. For this reason, efficiency is a common topic of study and innovation [27], and is something that this project could not go without addressing in at least some capacity.

The goal of this project is to construct a method by which novel metamaterial ferrite structures can be designed using computational optimisation techniques. To this end, a genetic algorithm was designed that used both MATLAB and CST in conjunction to build and test various high-impedance surfaces for the desired properties. This algorithm's structure and an account of its development process are detailed below.

3.1.5 Other Optimisation techniques

Occasionally, one finds that a single optimisation process is insufficient to the task at hand. When this happens, there are a few avenues of approach one can take.

Meta-Optimisation is one such approach [28]. It is a technique where one uses one optimisation algorithm to determine the ideal parameters for another. Theoretically, one could continue stacking layers of optimisation in this manner, but the process would quickly hit the point of diminishing returns. One of the earliest known uses of meta-optimisation was applied to a genetic algorithm in the late 1970s by Mercer and Sampson [29] In their paper, they developed an adaptive procedure to modify the parameter values of the GA while it was running, which allows one to do things like reduce the tolerances of the GA as it approaches a solution, allowing for faster convergence upon the solution.

Often, optimisation finds itself unable to completely define a problem it is attempting to find solutions for [30]. Sometimes this is because the data is incomplete, other times the number of variables is so large that it would be impossible to cover every combination in a reasonable amount of time no matter how powerful a computer was used. In these cases, a technique known as Surrogate Modelling can help by simplifying models and filling in the gaps [31]. There are multiple approaches to this, depending on what exactly needs to be tweaked to get the program in a usable state, and which method of simulation one is using in the first place. One example of such a technique applied to the subject of metamaterials is the Quasi-Newton Model-Trust Region Approach outlined in a paper by Patrick Bradley. [32] In it Bradley outlines a model-trust region optimisation approach employed to reduce

computational burden in a Visual Basic for applications-linked MATLAB/commercial solver software approach to metamaterial simulations. A model-trust region method is a heuristic procedure based on both the steepest-descent method and the quasi-Newton method [33]. The former is what pushes the solution toward a local minimum, while the latter will speed convergence when it is in sufficient proximity (as it would diverge or cycle when too far away on its own).

One potential issue with Genetic Algorithms and other optimisation techniques is that attempting to solve for too many variables at once can cause the running time to become unmanageably large. As such, steps to reduce the number of variables the program has to deal with are common. One example of such a method was presented by Zikri Bayraktar in the form of a GA-designed metasurface that limited the algorithm's parameters to the length of thin strips of metal that formed an almost key-like structure. [34] These structures were linked in a pinwheel pattern such that a single unit cell would consist of the central axle of the wheel surrounded by four unconnected key segments. This structure was compared directly with a near-identical set up that was not constrained in this manner. It proved far less likely to overwhelm the system, but this came at the cost of design flexibility, as one would naturally expect.

3.1.6: Machine Learning

Machine learning is a broader category under which genetic algorithms find themselves [35]. Generally speaking, a machine learning algorithm can be considered any technique by which a program can be taught to find solutions to problems by itself. [36] This is a useful feature, as there are some problems, such as Email spam filters, search engines and pictographic recognition systems, which are extremely difficult to simply program a machine to be able to do, but can be achieved via machine learning algorithms [37]. The technique is also used alongside datamining by financial companies and cyber-surveillance, and by marketing on websites such as Amazon to predict what customers may be interested in to provide better targeted advertising.

Machine Learning has existed in some form since the 1950s, but in recent years and with the advent of the internet it has really taken off. This is because generally, the process begins with developers building a database of solutions which the program can access and compare to the problem at hand, which can be sourced through datamining or other methods, which the internet has dramatically facilitated. However it's not enough for the system to simply check for answers it has already come across. True machine learning requires a more sophisticated approach [38]. A true machine learning program needs to be able to combine data in order to correctly predict results they have not previously encountered or solve problems with little or no input from humans [39]. This is generally achieved via optimisation algorithms such as GAs and other methods. In supervised learning, the most straightforward method, the algorithm is given data it has never seen before to process, and how well it does this is scored by the optimisation program, which has a "cheat sheet" of the expected results of the data [40]. This is commonly used for training machines for tasks such as photograph recognition. To give a somewhat simplified example, imagine that one has an algorithm that is required to identify pictures of bees. To test it a large database of images of bees and not-bees would be created, with a key identifying which were bees that the optimiser can see. This would then compare the real answers to those given by the algorithm, and a fairly objective score can be given. For other tasks, testing an algorithm's quality is not as simple, but will still follow a similar process. Occasionally so-called semi-supervised learning is used for the same tasks that supervised learning can be used to tackle. Semi supervised learning uses both labelled and unlabelled data, typically a large amount of the latter supplemented by a small amount of the former. The advantage of this method is that unlabelled data is less expensive and far easier to acquire than labelled data due to nobody

needing to go through and apply said labels. Unsupervised learning does not rely on labelled training data at all. Instead, the algorithm seeks patterns in whatever it is presented with. This has the same advantages in expense and data availability as semi supervised learning for largely the same reasons. A complete lack of labelled data means that it is often even cheaper. Unsupervised learning is better suited to organising data, sorting a large number of data points into manageable clusters. This can be used to aid things like marketing, identifying key demographics which can be targeted, or to tailor recommendations on streaming sites such as Netflix or Youtube.

One such technique that combined machine learning with Genetic Algorithms and similar optimisation methods was addressed in a 2014 paper by Bo Liu [41]. In it, they and their team proposed an evolutionary algorithm incorporating both surrogate modelling and machine learning, intended to provide an efficient method of antenna synthesis. The surrogate modelling method chosen was Gaussian Process Machine Learning, which assumes a Gaussian character for the objective function of the program [42]. This means that the distribution of a new point can be predicted based on that of those already available to the program, which allows it to screen points for those most likely to produce good results for computationally expensive thorough analysis whilst discarding those less likely to produce fruitful results. The Evolutionary Algorithm used was the Differential Evolution or DE algorithm. A DE algorithm uses a differential operator create new candidates, along with a one-to-one competition scheme to select the best among them. The program combines these by first running the DE, then passing the results to the surrogate model before selecting the best result and running as rigorous calculation of its exact parameters before passing it to the database. At this point, if the stopping criterion has not been reached, the program loops back around and begins again. Otherwise, it terminates and delivers the best solution as the output.

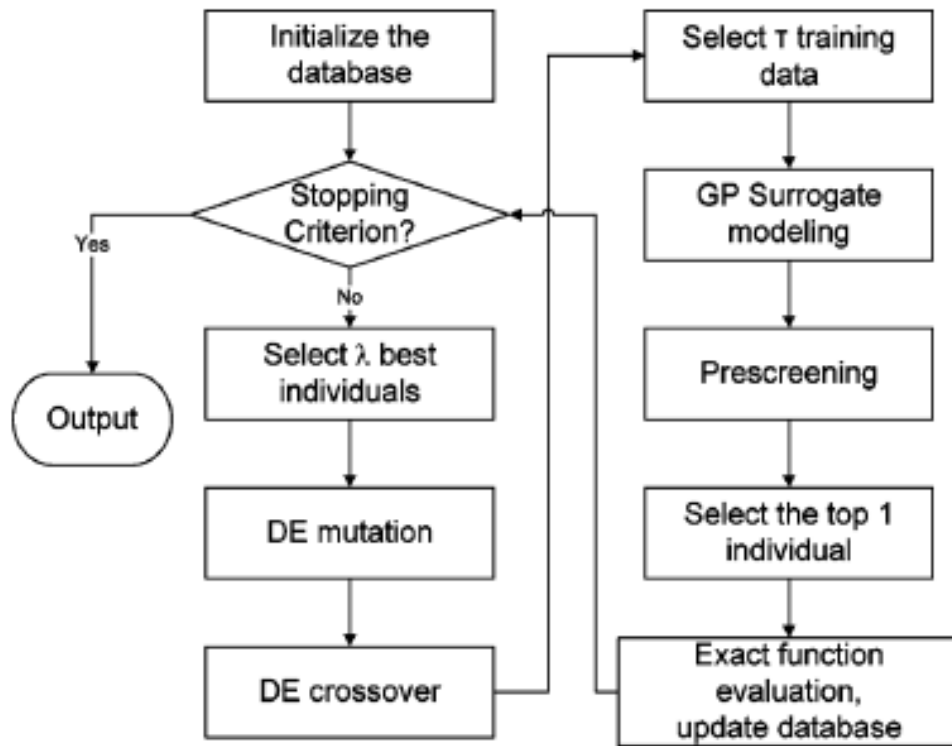


Figure 3.2: Diagram of the Algorithm used in [11]

Another common form of machine learning is known as Cluster Analysis [43]. Cluster Analysis is a form of Unsupervised Learning, which differs from the types of machine learning described above as the teaching program has no direct “answer key”. Instead, Cluster analysis deals with the classification of unlabelled data. It does this by using so-called “similarity measures” to group data into sets that share common properties [44]. There are two primary goals that cluster analysis is used to achieve: complexity reduction and exploratory analysis. In complexity reduction the algorithm seeks to divide the data into small groups that can then be assigned a representative data value, effectively reducing the number of data points to something more manageable [45]. With exploratory data analysis the aim is instead to present the data in a way that aids the analyst in seeing patterns and formulate hypotheses by themselves [46].

In either case, there are several methods that can be used to achieve this. A common method is Hierarchical Clustering, in which the clusters are slowly built up in stages. This can be done either with an agglomerative method, which takes a bottom-up approach and slowly merges the data into larger and larger groups, or with a divisive method, which is a

top-down approach where the data is taken first as a whole and then split into smaller groups as one descends the hierarchy. [43]

Another method is known as K-means clustering [47], which uses the k-means or Lloyd's algorithm to partition the data space into uniformly-sized Voroni cells as a method of vector quantisation. It does this by dividing observations among a set number of clusters, with each individual observation being assigned to the cluster with the nearest mean value. [48] A Voroni cell is defined a region in which no point is closer to this mean than any other, which means that this method can be quite slow, as much like genetic algorithms it involves an iterative process. In the case of this method, the "centroid" of each Voroni cell is computed, and then the site of each cell is moved be evenly distributed around its centroid and a new Voroni diagram is created, for which the centroid is again computed until an acceptably uniform distribution is achieved. Perfect convergence is often impossible with real data, so usually there is some defined point deemed "close enough".

In the case where one has a lot of high-dimensional data, which is to say data points with multiple independent parameters, a technique known as Dimensionality Reduction may be employed to simplify it [49]. Most commonly, Dimensionality Reduction works by identifying the parameter with the least variance and eliminating it, reducing the problem to its most relevant variable. This method is known as Latent Variable or Principle component Analysis. Another method, commonly used in image processing, is known as Feature Extraction, and instead works by building informative, non-redundant derived values from the initial set of data [50].

Dimensionality Reduction is also useful to apply as a precursor to Cluster Analysis, since there are many possible clustering algorithms that work with low-dimensionality data but only a few that are effective with high-dimensionality data [51].

Applications of Principle Component Analysis and K-means Clustering were explored by Chris Ding and Xiaofeng He in the appropriately titled "K-means Clustering via Principal Component Analysis" [52] In it, they outline methods for 2-way clustering, before generalising to K-way. They then demonstrated the technique with two examples: Gene expression and Internet Newsgroups.

In 2-way clustering, they began by defining the sum of the square of the distances between two clusters, denoted C_k and C_l , as

$$d(C_k, C_l) \equiv \sum_{i \in C_k} \sum_{j \in C_l} (x_i - x_j)^2 \quad (3.4)$$

Equation 16 can be rearranged to give

$$J_K = \sum_{k=1}^K \sum_{j \in C_k} \frac{(x_i - x_j)^2}{2n_k} = n\bar{y}^2 - \frac{1}{2}J_D \quad (3.5)$$

and

$$J_D = \frac{n_1 n_2}{n} \left[2 \frac{d(C_1, C_2)}{n_1 n_2} - \frac{d(C_1, C_1)}{n_1^2} - \frac{d(C_2, C_2)}{n_2^2} \right] \quad (3.6)$$

Since \bar{y}^2 is a constant, this means that the minimum value of J_K is equivalent to the maximum value of J_D . Since it can be also shown that

$$\frac{d(C_1, C_2)}{n_1 n_2} = \frac{d(C_1, C_1)}{n_1^2} + \frac{d(C_2, C_2)}{n_2^2} + (m_1 - m_2)^2 \quad (3.7)$$

It is possible to show that J_D is always positive by substituting equation 19 into equation 18.

This all means that it is possible to maximise the distance objective J_D by minimising the objective function J_K , thereby separating the clusters as much as possible.

Generalising to K -way clustering is a matter of first using (17) to write J_K in the form

$$J_K = \sum_i x_i^2 - \sum_k \frac{1}{n_k} \sum_{i,j \in C_k} x_i^T x_j \quad (3.8)$$

Here, the first term is a constant, while the second is the sum of K diagonal block elements of an $X^T X$ matrix representing within-cluster similarities. Using K non-negative indicator vectors, this can be written in the form

$$J_K = \text{Tr}(X^T X) - \text{Tr}(H_k^T X^T X H_k) \quad (3.9)$$

This equation has several redundancies in H_k , which can be removed by a two-step process. Step one is to perform a linear transformation T such that $Q_k = H_k T$. The second step is to require that the last column of the matrix T is given by

$$\mathbf{t}_n = \left(\sqrt{n_1/n}, \dots, \sqrt{n_k/n} \right)^T \quad (3.10)$$

This means that

$$\mathbf{q}_k = \sqrt{\frac{n_1}{n}} \mathbf{h}_1 + \dots + \sqrt{\frac{n_k}{n}} \mathbf{h}_k = \sqrt{\frac{1}{n}} \mathbf{e} \quad (3.11)$$

This linear transformation is always possible, which allows the 2-way case to be generalised to any value of K desired. The K=2 case itself becomes

$$T = \begin{pmatrix} \sqrt{n_2/n} & -\sqrt{n_1/n} \\ \sqrt{n_1/n} & \sqrt{n_2/n} \end{pmatrix} \quad (3.12)$$

3.2: Genetic Algorithm Outline

To begin with, the general structure of the Genetic Algorithm itself had to be considered; how large a population to use, the parameters being tested, how stringent the cut off points should be. In addition, the tools used to build the algorithm can have a great effect on the structure of the algorithm. This was particularly notable in the case of this project, as the mechanism proposed involved the use of two different pieces of software running in parallel.

With all these variables under consideration, the general outline of the Genetic Algorithm program for this project was initially designed in the form shown in figure 3.3. Ultimately, this design was modified slightly over the course of the project, albeit mostly due to the differences required in the initial loop to set the initial parameters. It was found that the easiest way to implement this was to make that first generation a separate sequence. By and large the procedure was nonetheless unchanged.

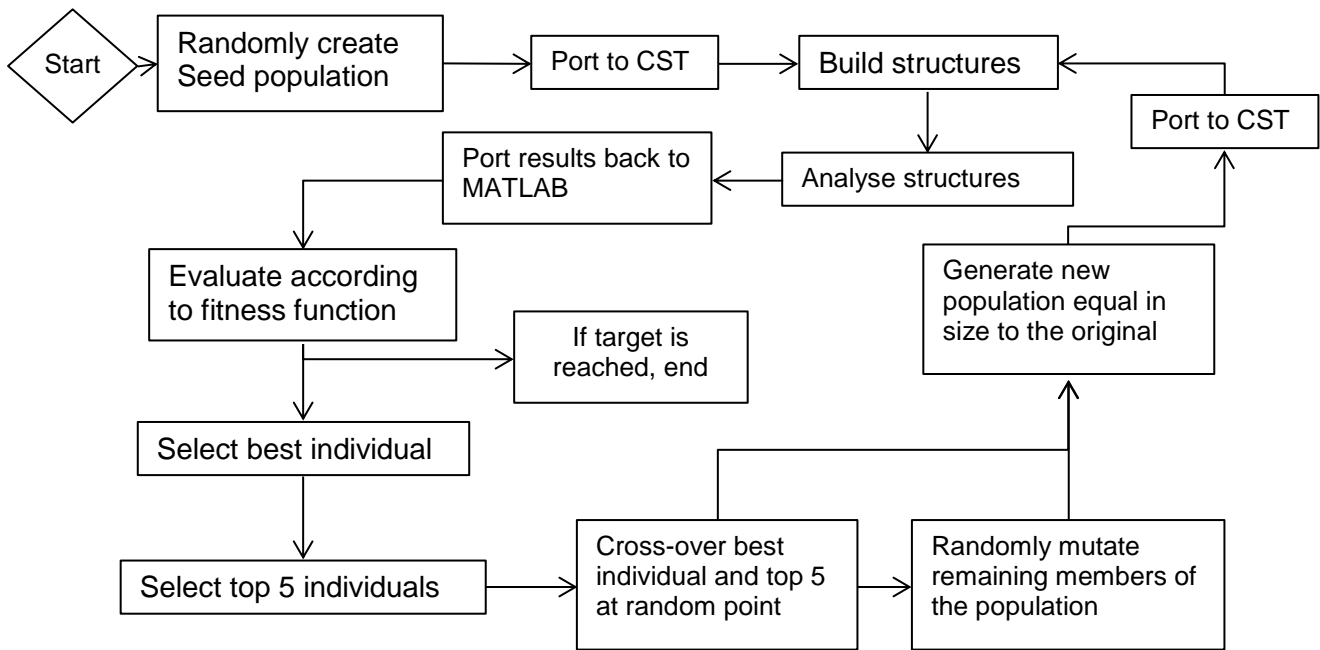


Figure 3.3: Flow chart detailing Genetic Algorithm run procedure.

3.2.1: Generation of the Seed Population and Export to CST

The Genetic Algorithm program begins by randomly generating a series of chromosomes that defines the “genetic code” of the structures. In this case, each consisted of a string nine of ones and zeroes, created by taking a series of random decimal numbers created using the random number generation function of MATLAB. It converted the decimal fractions produced by this function into a binary 1 and 0 scheme by simply setting the chromosome to read 0 if the number was less than 0.5 and 1 if it was greater. However, the nine-bit chromosomes generated this way are far too short to form the basis of a usable high-impedance surface as they are. Instead, once they have been generated they are immediately run through an L-system algorithm [53] whose purpose is to expand them into a larger string that is of sufficient size for the intended purpose, which in this case 45 bits in length.

Once the Genetic Algorithm has created a sufficient number of 45 bit structure codes (in the case of this program the number was typically 12) the program can then export the strings as a text files, which are a format that can be read by a Visual Basic macro in CST. Said macro now takes over control of the Genetic Algorithm program.

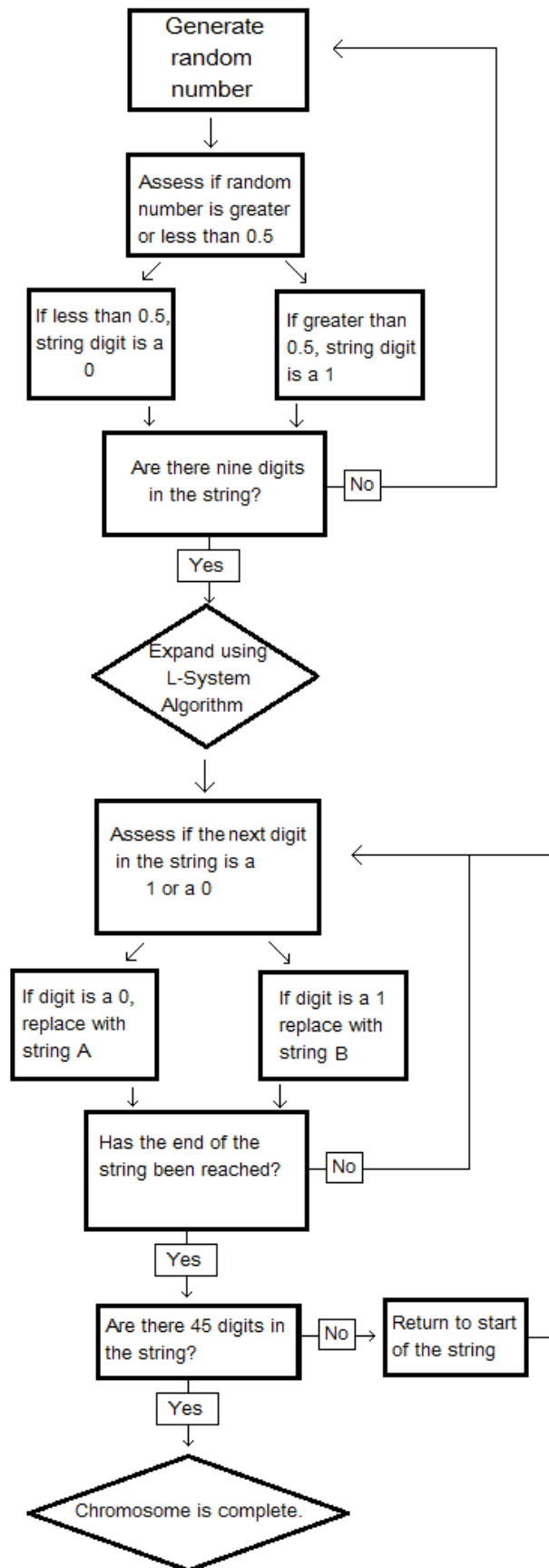


Figure 3.4: Flowchart of the initial chromosome generation procedure.

3.2.2: Building the Structures

1	10	18	25	31	36	40	43	45
2	11	19	26	32	37	41	44	
3	12	20	27	33	38	42		
4	13	21	28	34	39			
5	14	22	29	35				
6	15	23	30					
7	16	24						
8	17							
9								

Figure 3.5: Map of bit numbers in the string to the pixel in the high-impedance surface template they correspond to. If the bit is a 1 there will be a 1mm x 1mm square of material inserted. If it is a 0, this space will be left blank.

Once a full length bit string has been exported as a text file, MATLAB triggers a Visual Basic file to run and sits back. This VBA file then imports the text file as a variable, and runs through the bits applying 1mm by 1mm squares of material (usually a metal such as copper or silver) to predefined template (seen in figure 3.5) based on whether the bit is a 1 or a 0. Once the template has been filled, it is reflected along its diagonal axis before being rotated around the centre of the structure to produce a rotationally symmetrical shape.

Initially, this part of the code produced a 36-digit string of numbers which were applied (truncated) grid directly. It was only later that the aforementioned L-system techniques were employed to reduce the number of randomly generated variables to nine, whilst maintaining the original string length. This also allowed the structure to be easily expanded and as such, later tests were performed on the performance of 45 and 54 digit strings. These numbers are important as they correspond to values that fit in the above triangular grid pattern (figure 3.5), simply adding a row each time. The reason that this was done was as an attempt to reduce the complexity of the problem and allow the Genetic

Algorithm to reach a solution more quickly, as together with the excessively long run time of CST, the whole optimisation process will be very time-consuming, which make it impractical for engineering applications.

3.2.3: Analysis of Structures

Once a structure has been built in CST, it is then analysed using the software's frequency domain solver, which extracts the real and imaginary parts of the S11 parameters. It then exports these back to MATLAB using the same method of having text files as an intermediary. MATLAB then uses these to derive other parameters such as the permeability or permittivity using the equations which have been derived in chapter 2 and [34]

$$\epsilon_r = \frac{1}{i\beta_0 d} \tanh^{-1} \frac{Z_{imp}}{Z_0} \quad (3.13)$$

$$Z_{imp} = Z_0 \frac{(1+S)}{(1-S)} \quad (3.14)$$

Where d is the effective thickness of the structure, Z_0 is the impedance of free space, Z_{imp} is the impedance of the structure, and S is the reflection parameters.

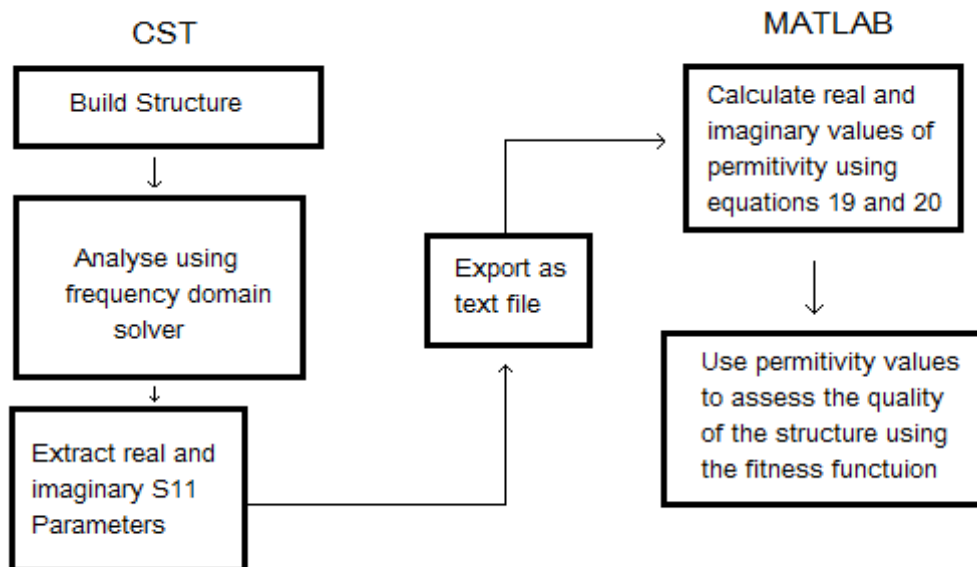


Figure 3.6: Detail of the build and analysis step, separated into the programs used for each component

3.2.4: Evaluation by Fitness Function

The permittivity parameters calculated by MATLAB from the CST-derived reflection values are then inserted into the Fitness Function in order to assess them and determine which population member has them at the most optimal values. The criterion for this is based on how close to a minimum of the function they approach. Those closest, within a tolerance level set by the program, are selected to form the next generation. Several iterations of the program experimented with different tolerance levels, in order to establish a good balance between finding too many solutions and not finding enough. It is important to have a good selection of solutions for recombination whilst not being so generous that solutions with insufficient chance of producing the desired end result slip in.

The Fitness function used here was based on the one used by Kern *et al* in [54], modified slightly to allow for differences in the program structure and design constraints. The base goals were the same, however: a dielectric constant of 10 and minimal loss. As such, the function took the form

$$FF = -[(\epsilon_{REAL} - 10)^2 + \epsilon_{IMAGINARY}] \quad (3.15)$$

with the two variables being those taken from the calculated permittivity.

3.2.5: Recombination of solutions and the creation of the second generation.

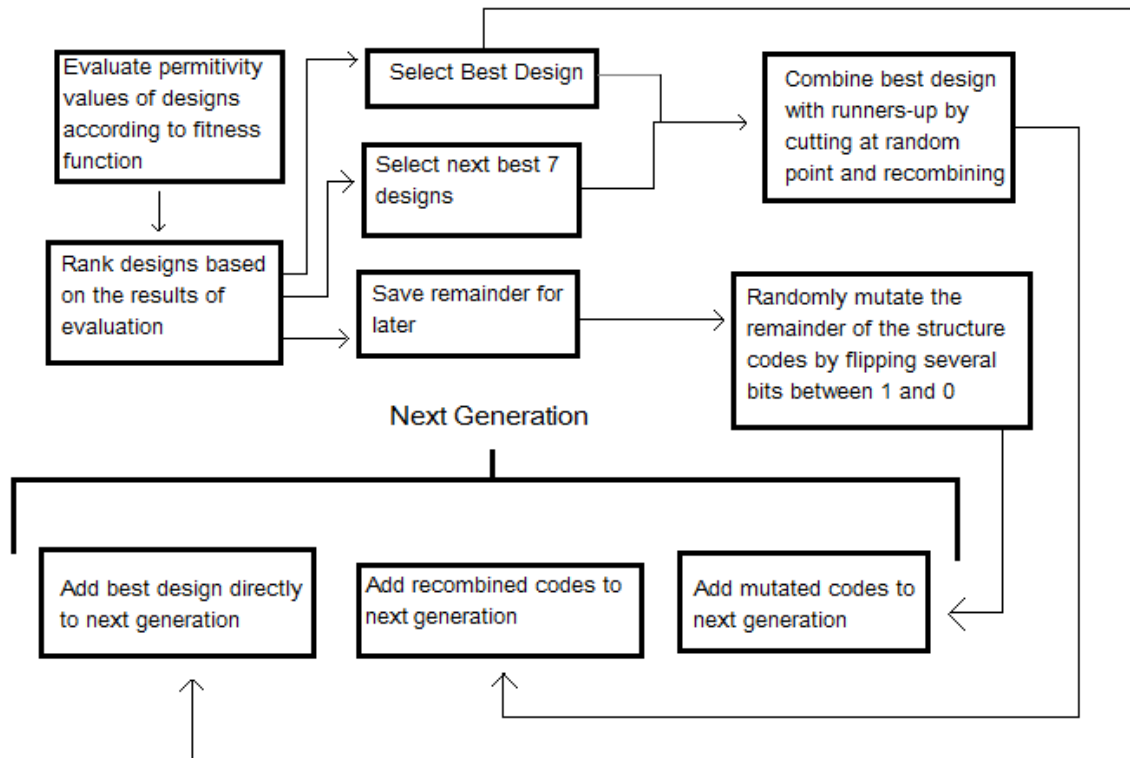


Figure 3.7: Fitness function evaluation and recombination procedure.

The solutions that meet the standards as chosen by the fitness function and other parameters defined within the program are then recombined (with the best function also remaining intact, so the program does not go backwards and lose it) by a process which takes two solution chromosomes, cuts them in half at a semi arbitrary point (typically somewhere in the middle, to ensure a good combination of features) and swaps the pieces over. The members who failed to achieve parameters within the prescribed tolerance limit are not without use, however. Since the population needs to remain constant (at least with this example of the algorithm) the remaining chromosomes are randomly mutated by flipping a few of their bits from either 1 to 0 or vice versa and used to fill the population back up to its initial size. This is not the only reason for doing this, however. Rather, it is also done in order to maintain diversity in the solutions and so avoid becoming trapped at a local minimum.

The process then begins again with the new population, theoretically producing better and better results until finally a solution sufficiently close to the target is reached, at which point the program will terminate and output that design.

3.3: Application of Fractal Designs

The Genetic Algorithm program uses a Lindenmayer-system algorithm, also known as an L-system algorithm [53], to expand the 9-bit seed generated in MATLAB into a 45-bit string useable by the CST construction macro. This is done in order to reduce the number of variables that the genetic algorithm must deal with, with the goal of increasing the algorithm's efficiency and the speed with which it converges upon a solution.

An L-system algorithm works by applying a certain series of rules under which the variables in a string are altered. For example it could be that every 0 in the string is replaced by a 0 and a 1 and every 1 is replaced by a 0. The algorithm then takes the new string and applies the rule again, repeating the process until the string has reached a desired length.

This is a somewhat unusual use of the technique, however. By far the most common use for L-system algorithms, and the purpose they were designed for, is as a means of constructing fractal patterns [53]. This is done by assigning some sort of function to each variable, such as drawing a line pointing in one direction or another. The repeating variables will naturally cause these simple designs to form deceptively complex patterns such as the well-known dragon curve. Other L-systems can be used to form different types of pattern such as the Hilbert curve, a space-filling curve useful in a wide variety of applications [55] [56].

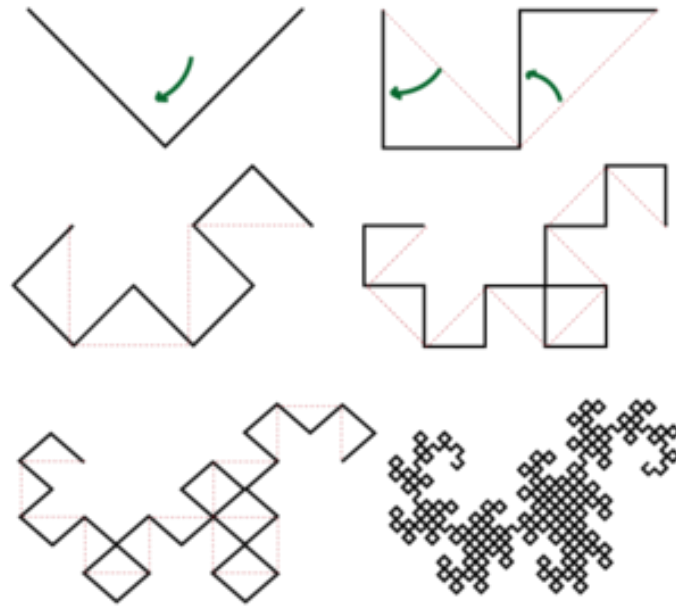


Figure 3.8: The construction process of a Dragon Curve. It can most simply be thought of as the result of folding a piece of paper in half many times (far more than the maximum of 7 or so that is possible in reality) before unfolding and opening all the folds at 90 degrees. It can also be modelled by an L-system of the form $X \rightarrow X+YF+$ and $Y \rightarrow -FX-Y$ using an initial string FX . [57]

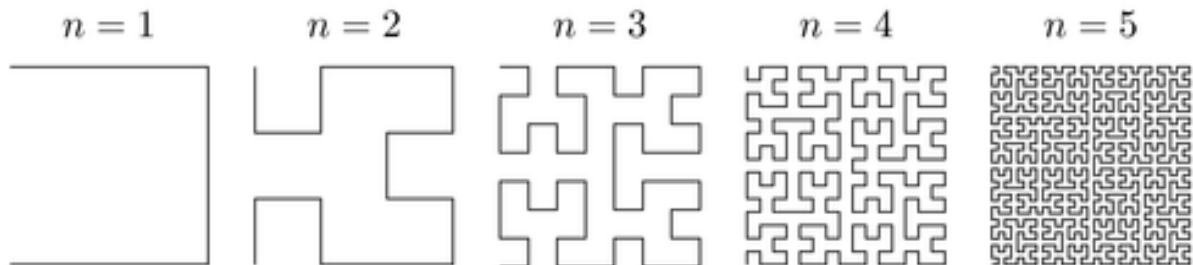


Figure 3.9: A Hilbert curve, another fractal pattern that can be created using an L-system. A Hilbert curve is also an example of a space filling curve, which can also have applications in high impedance surface design. [58]

As one can see from figures 3.8 and 3.9, Fractal Patterns, which is a term coined by mathematician Benoit Mandelbrot in 1975, in relation to his study of what would later be known as the Mandelbrot set, are patterns that exhibit what has been described by some as “expanding symmetry”. What this means is that in much the same way that an object with reflectional symmetry will appear the same when reflected along a certain axis and an object

with rotational symmetry will appear the same when rotated through a particular angle, an object with expanding symmetry will appear the same when magnified by a particular amount. In some cases this may only hold true at a few distinct magnifications but others can be astoundingly complex. The dragon curve, for example, will tessellate with itself, and several of them can be combined into a larger version of the pattern.

3.4: Final Design and Modifications

As mentioned earlier, the structures initially created by the program were 16 x 16 structures, produced via a direct 36 bit chromosome. In other words, each pixel was completely independent of all the others, which increased the potential design space at the cost of increased computational load. One example of a design from this version of the program is shown in figure 3.10. Visually speaking, the differences between that and the other two types of structure are subtle at best, but there are fundamental differences between the three. When combined with the L-system expansion and slight rearrangement of the grid structure of the template, the small increase in size also means that a given string will produce dramatically different high-impedance surface structures.

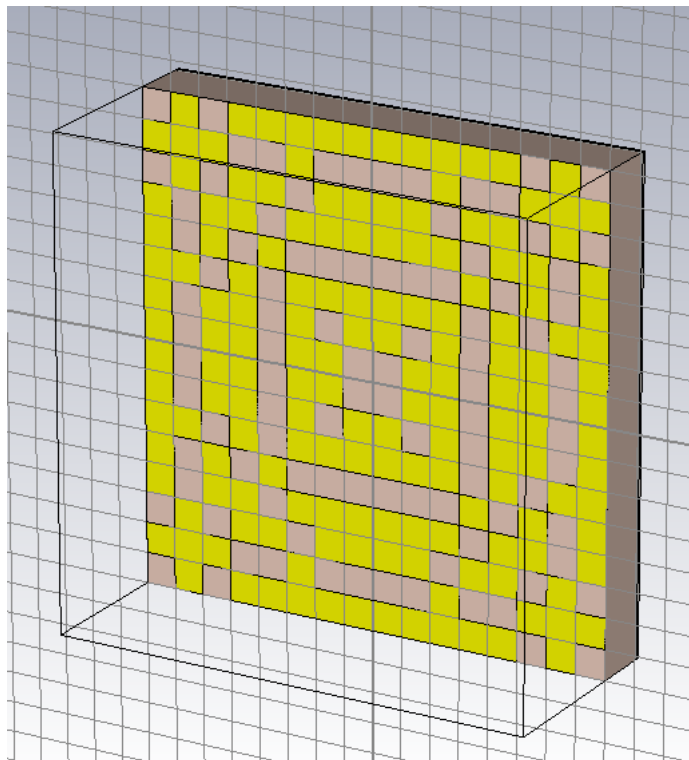


Figure 3.10: Example of the type of structure that will be generated by the GA program (non-optimised, as this was taken from one of the early generations)

It is for this reason that when the seed generation algorithm was redesigned in such a way that it became possible to increase the size of the structure without increasing the complexity of the randomly generated portion, the opportunity to explore what differences such a change would make, and which were most beneficial to the goal of creating an artificial magnetic conductor was quickly seized upon.

Since the structure creation method involves generating a small randomised segment of the design and reflecting it in several axes until it reaches the desired shape and size, it requires the final shape to be an even number of units in width. As such, when it came to selecting larger sizes to attempt an exploration of it was decided that exploring 18 x 18 and 20 x 20 squares would prove most fruitful. This required increasing the length of the input string from 36 bits to 45 and 54 bits, respectively.

All L-systems require at least one of the replacement strings be longer than what it replaces, so that the overall string increases in size. Often, one of the replacement strings of an L-system can be longer than the other. For example you may have an algorithm that replaces “0” with “01” and “1” with “0”. In cases like this, the length of the string after a given number of passes through the algorithm will display a degree of variation, based on the values in the initial string. One that began with more 0s will be larger after a given number of cycles, as each zero is replaced with two characters rather than one. In cases where the replacement strings are of equal length the final length of the full string they will produce will be consistent, but even then it is often difficult to set these up so that the final value will be exactly what is required by the program.

So, either way, the program will likely end up with either too few or too many values after a given number of iterations. This issue was particularly noticeable at first, as the program initially used a for-loop to run the L-system for a set number of iterations. In order to prevent having too few, the program was instead set to deliberately overshoot and then discard any excess values. This had the downside several of the variables of the base string would inevitably be lost, with a severity depending on the L-system used, and meant that if care was not taken the structure would be dominated by just the first few random values, truncating the design space to an unacceptable degree.

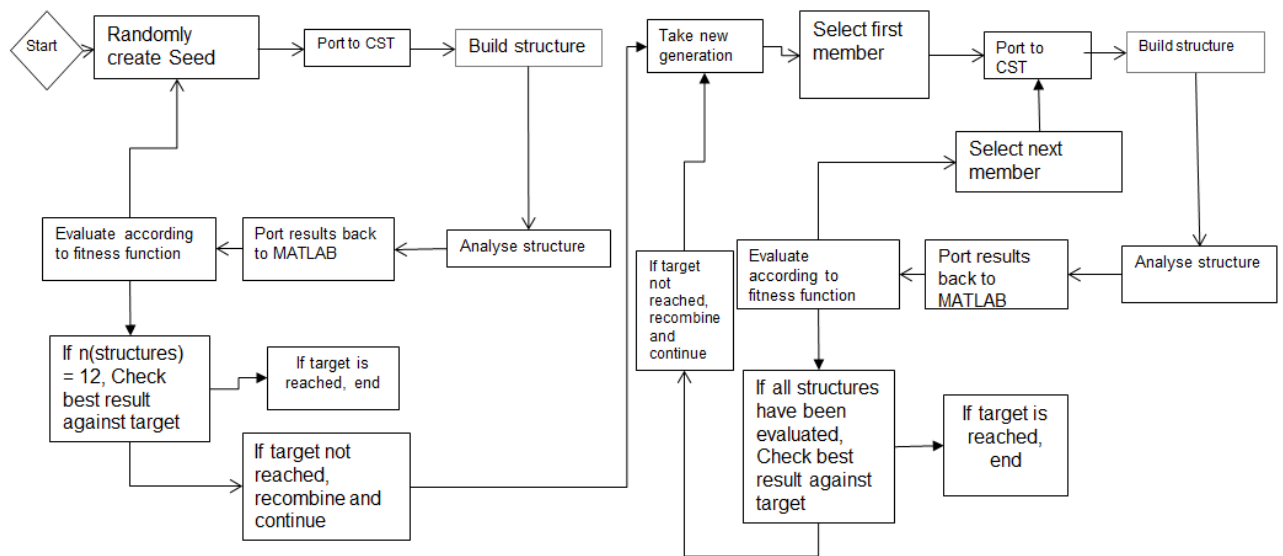


Figure 3.11: Final Iteration of Genetic Algorithm function.

This issue was partially solved by switching to a while-loop that terminated the process as soon as the required value was met or exceeded, allowing the program to have flexibility which minimised both run time and waste chromosome clippings. Eliminating lost values entirely was, unfortunately, almost impossible.

The design of the Genetic Algorithm remained largely intact throughout production, but there were some alterations to the design once it was complete. The final Genetic Algorithm's software structure proposed in this thesis is shown in figure 3.11. The primary difference between the final version of the Genetic Algorithm depicted here in figure 3.11 and the initial design shown in figure 3.3 is the addition of the initialisation loop depicted on the left of the diagram. Its function is largely identical to the rest of the algorithm, but variations in the design that arose from generating a population of solutions from scratch instead of receiving them from a prior loop meant that it proved to be far easier to construct that step from a separate set of functions.

3.5: Validation of the Genetic Algorithm

Validation is important in any project, and it is important to reassess that validation from time to time as the project progresses. Something may have worked originally, but additions and increasing complexity have the potential to have unforeseen consequences that affect the function of those original components, and when this happens it is important to

catch those issues and fix them. This is especially true in a complex function like the Genetic Algorithm designed for this project, as it contains a multitude of moving parts.

Initially, the MATLAB portion of the program was built and tested by tasking it to optimise the raw chromosomes of the structures, in absence of any CST analysis. After several rounds of modifications to connect the program to CST and general improvements, the Genetic Algorithm had been altered sufficiently from its initial form that it was felt prudent to test the program in order to validate the results of the current iteration and to ensure that these modifications had not negatively impacted its function.

This was done by carefully deactivating the additional lines of code without disturbing the now-modified original structure. This reduced the goal to one similar to that of the original non-linked program, but rather than the simple gradient fitness function it was initially using it was instead tested using the two-dimensional form of the Griewank function, seen in figure 3.12. The Griewank function is a function designed for this purpose with an abundance of local minima intended to trap insufficiently robust genetic algorithms. It is defined by the equation

$$y = 1 + \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) \quad (3.16)$$

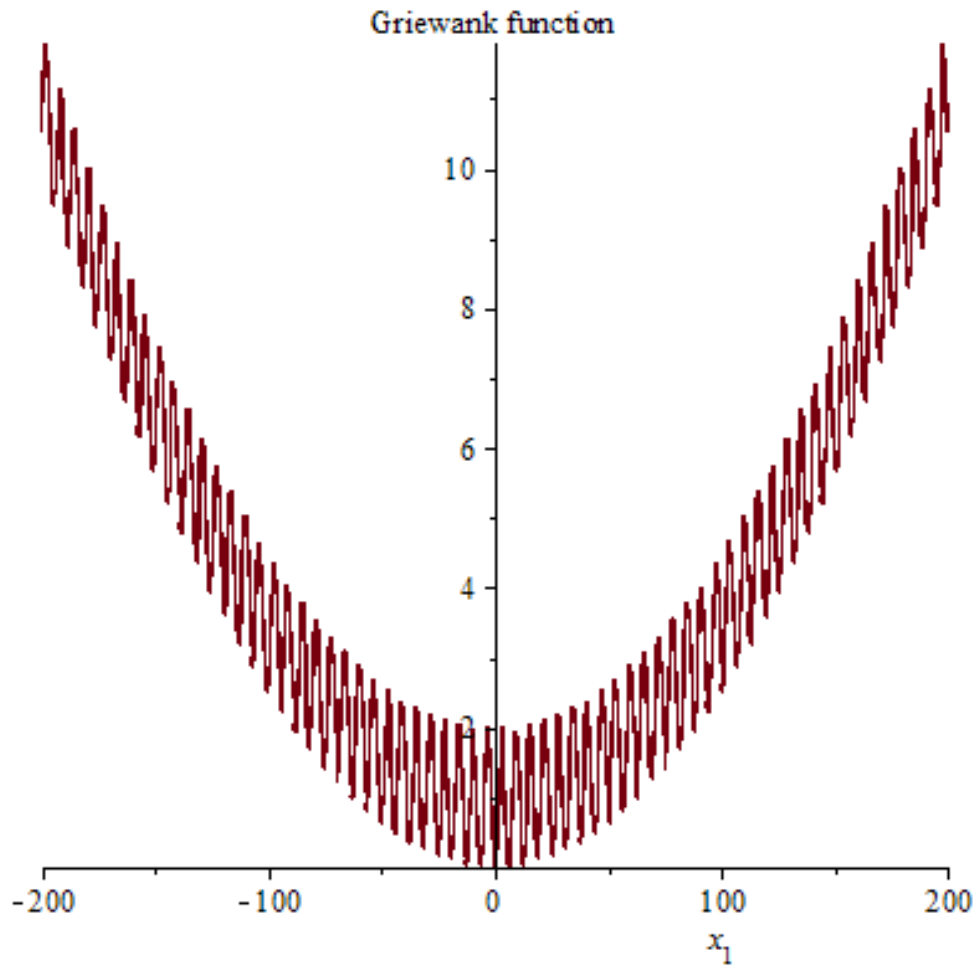


Figure 3.12: The first order two-dimensional Griewank function, a function with a multitude of local minima designed to trap optimisation algorithms. [59]

The Genetic Algorithm was tested against both a first order and second order Griewank function. Since there was no processing of the generating bit string in CST for this evaluation, the GA was simply tasked with minimising the sum of the digits in the string. This did mean that the values for x could only go from 0 to 9, and thus a lot of the Griewank function's complexity was lost, but what remained proved sufficient for the purposes required.

It was decided that this was also a good opportunity to test the effect of population size on the speed of convergence for the algorithm. As such, tests were done with both a 12 member population and an 8 member population of solutions. A lower number of members in the population would result in each individual generation being completed faster, at the expense of more generations being required to converge upon a solution, on average. What

the experiment intended to discover was which of these effects was larger, as this would inform the optimal point of maximum efficiency.

In each case, the Genetic Algorithm was run numerous times and the number of generations required to converge on the solution was logged. In addition, the program was set to terminate after 1000 generations regardless of whether a solution had been found, and this eventuality was also recorded. Then, a bar chart of the frequency of each instance of a given number of generations was constructed, grouped into sets of 100.

In the case of population size 8, the number of generations required to converge on a solution is somewhat erratic. This size of population also displayed a high frequency of runs (over 25) which hit the 1000 generation cut-off point and were terminated without finding a solution.

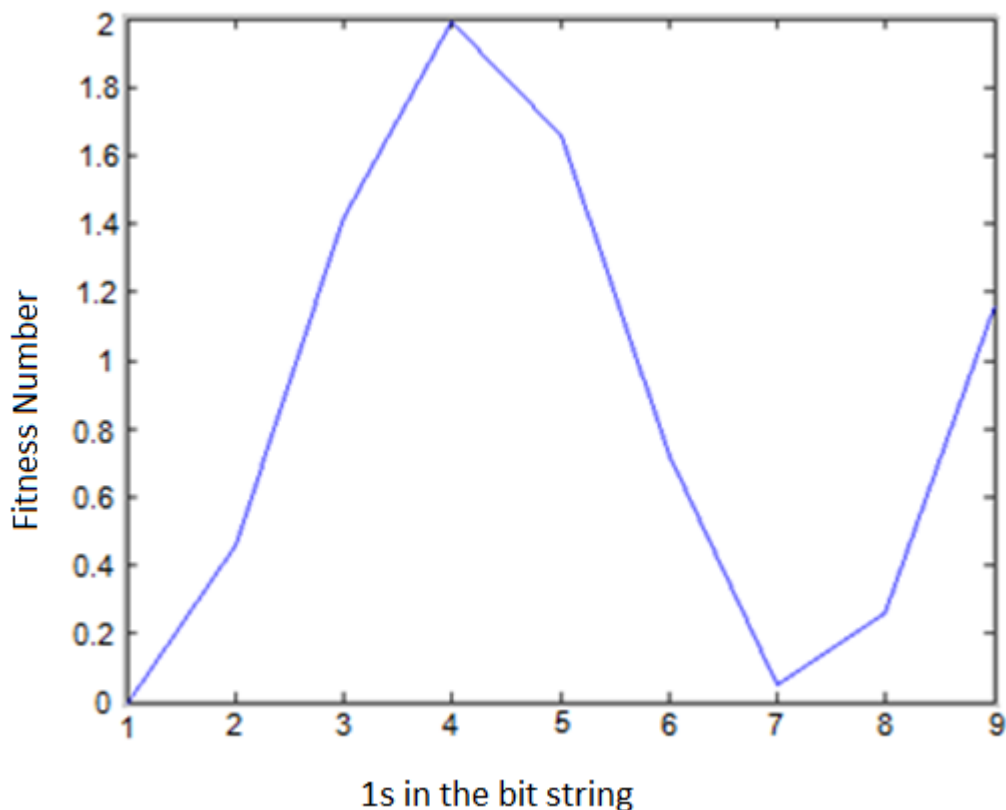


Figure 3.13: The first order griewank function used for testing the GA

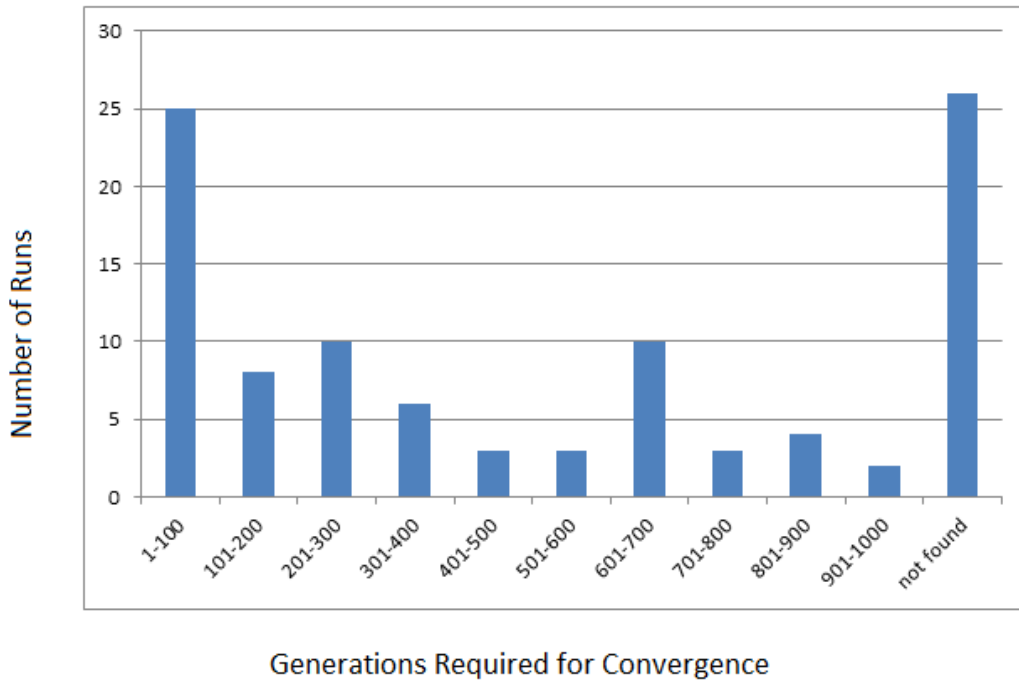


Figure 3.14: The number of generations required to converge upon a solution when the population size was set to 8.

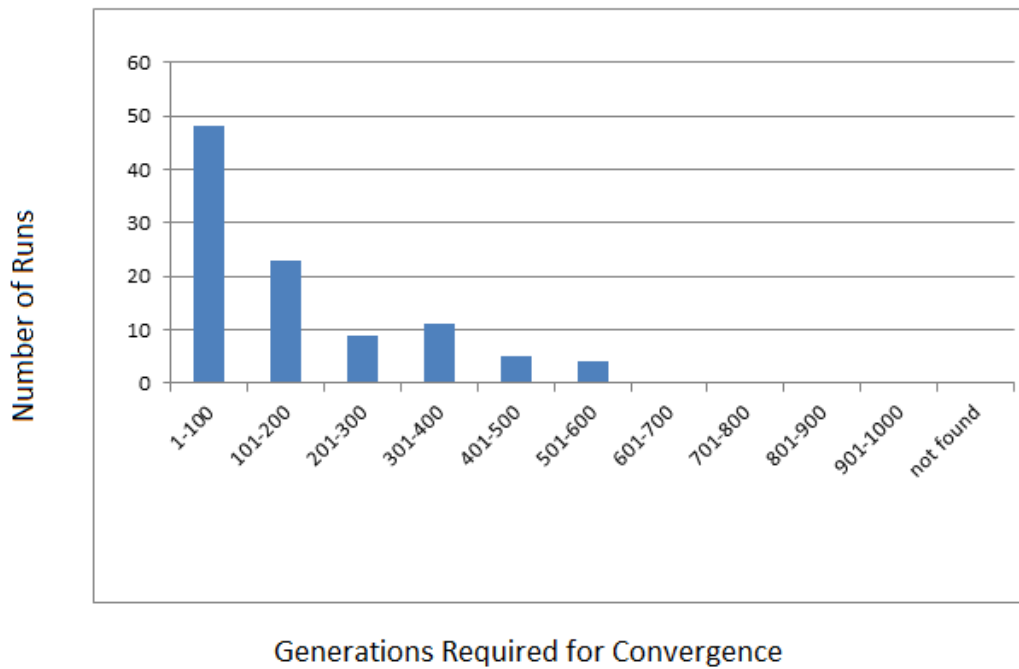


Figure 3.15: The number of generations required to converge upon a solution when the population size was set to 12

By comparison, the 12 member population shows a distinct improvement in efficiency. None of the runs failed to converge on a solution, and the longest any of them took was between 500 and 600 generations, a little over half the maximum allowed. There is also a clear progression of rarity as the required number of generations increases, with the vast majority of runs taking far less than this, with most lying in the 0-200 range. Almost 50% of the optimisation attempts converged within 100 generations of the algorithm being started.

Having excluded the eight-member population as a viable option, the 12-member population was subjected to a more rigorous variant of the Griewank function, that being the second-order version. This had an additional local minimum in the available design space, and was therefore more likely to trap the algorithm than the first-order version.

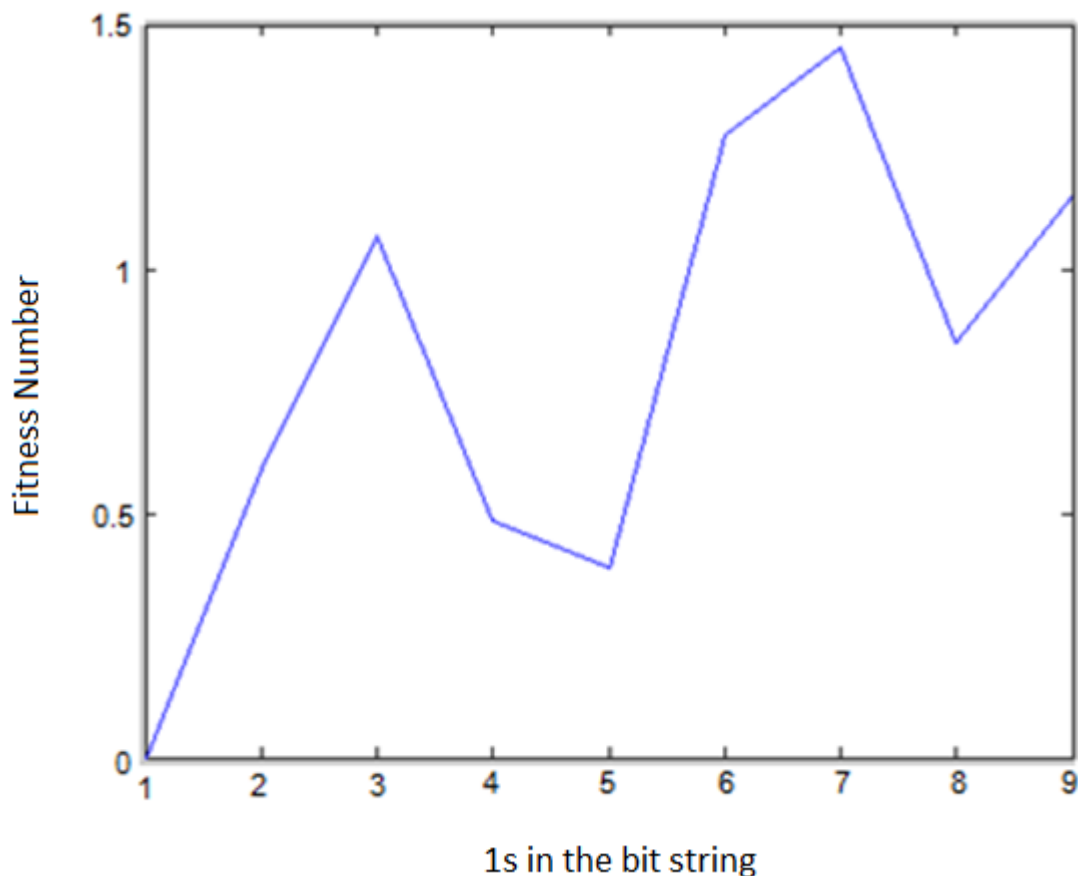


Figure 3.16: The second order two-dimensional griewank function used for testing the Genetic Algorithm

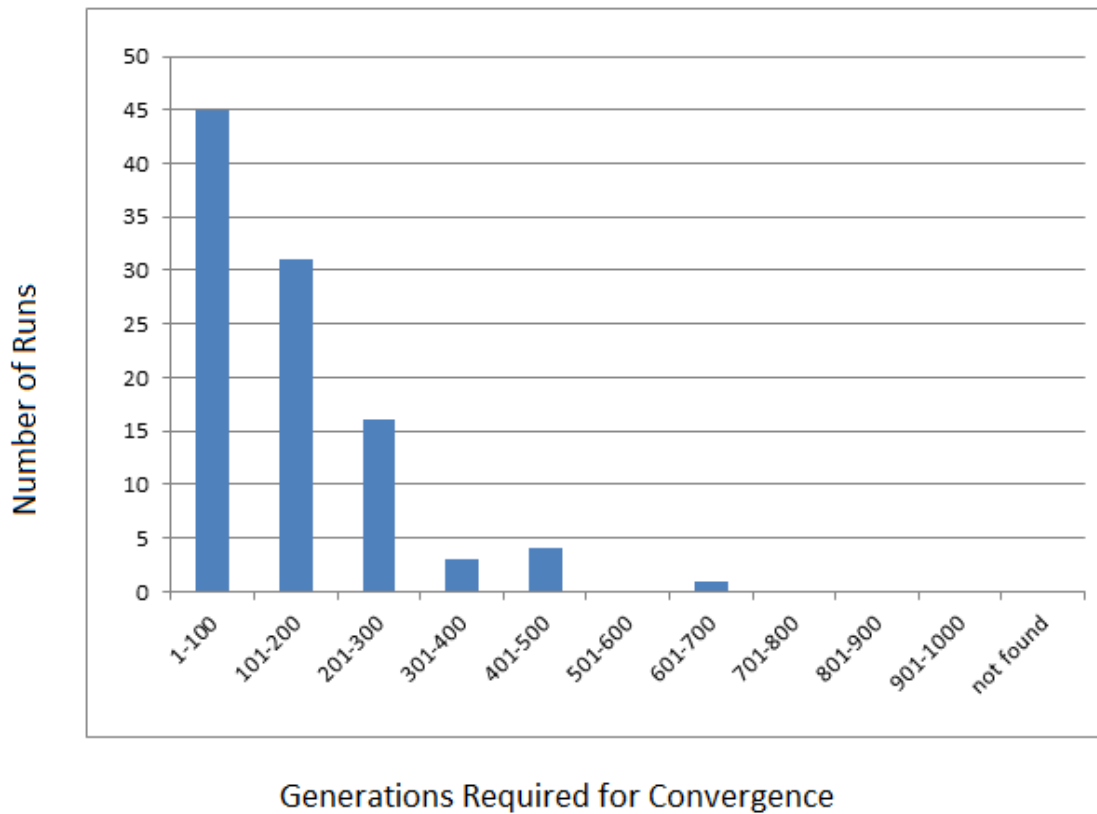


Figure 3.17: Number of Generations required by the program to converge upon a solution with a population size of 12

Oddly, while there was a slight increase in the maximum number of generations required, this was represented by only a very small number of runs. The vast majority actually seemed to cluster more at the low-generation end of the scale than they did with the first order function. Regardless, it was considered at this point that the base program had proven itself to have remained functional, so work could continue on the program with confidence.

3.6: Summary

In this chapter, several potential forms of computational optimisation are presented and studied, their pros and cons evaluated and their applicability to certain problems is assessed. These included the Newton-Raphson method, Quasi-Newton method, Particle Swarm technique and Genetic Algorithm method. The chapter then goes into particular detail about the latter method, as this was the sort used by this project. It details the process by

which Genetic Algorithm was constructed, based on previous attempt to create one with the intention of designing Metamaterials, along with a step-by-step guide to how it functions.

Details are presented regarding the process by which the Genetic Algorithm created for this thesis project uses a combination of MATLAB and CST to design, build and evaluate potential metaferrite structures. It uses a technique derived from fractal generation methods to expand a small base chromosome into a larger structure, in order to reduce complexity and increase efficiency.

The chapter also concerns itself with how the Genetic Algorithm went through several iterations and several rounds of testing before arriving at its current form. These testing methods are also presented in detail.

3.7: References

- [1] P. A. Vikhar, "Evolutionary algorithms: A critical review and its future prospects". *In proceedings of the 2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC)*, December 2016
- [2] J Cohoon, et al. "Evolutionary algorithms for the physical design of VLSI circuits" *Advances in Evolutionary Computing: Theory and Applications*. Springer, pp. 683-712, 2003.
- [3] J.Clune, B.Beckmann, C.Ofria, and R.Pennock. "Evolving Coordinated Quadruped Gaits with the HyperNEAT Generative Encoding". Proceedings of the IEEE Congress on Evolutionary Computing Special Section on Evolutionary Robotics, 2009.
- [4] WH Press, SA Teukolsky, WT Vetterling, BP Flannery, "Chapter 9. Root Finding and Nonlinear Sets of Equations Importance Sampling". *Numerical Recipes: The Art of Scientific Computing (3rd ed.)*, 2007
- [5] J.Wallis, *A Treatise of Algebra both Historical and Practical*, 1685
- [6] J.Raphson, *The Universal Equation Analysis*, 1690
- [7] V. S. Ryaben'kii, S, V, Tsynkov, "A Theoretical Introduction to Numerical Analysis", *CRC Press*, p. 243, 2006
- [8] R. Sureshkumar and G. C. Rutledg, "Solution of Non-Linear Algebraic Equations", http://web.mit.edu/10.001/Web/Course_Notes/NLAE/index.html 1998. accessed June 2018
- [9]R, Haelterman,. "Analytical study of the least squares quasi-Newton method for interaction problems". *PhD Thesis, Ghent University*. 2009
- [10] A.K. Kaw, E.E. Kalu, *University Numerical Methods with Applications*, March 2011
- [11] Ž. Povalej, "Quasi-Newton's method for multiobjective optimization" *Journal of Computational and Applied Mathematics Volume 255*, January 2014
- [12] R. Schoenberg, "Optimization with the Quasi-Newton Method" *Aptech Systems, Inc.* September 2001
- [13] R. Haelterman, J. Degroote, D. Van Heule, J. Vierendeels "The quasi-Newton Least Squares method: a new and fast secant method analyzed for linear systems". *SIAM J. Numer. Anal.* 47, 2009
- [14] J.Kennedy, R. Eberhart, "Particle Swarm Optimisation", *Neural Networks*, 1995
- [15] Y.Zhang, "A Comprehensive Survey on Particle Swarm Optimization Algorithm and Its Applications". *Mathematical Problems in Engineering*, 2015
- [16] J. Kennedy,. R Mendes, "Population structure and particle swarm performance". *Evolutionary Computation*, 2002.

- [17] D. Bratton, J. Kennedy, "Defining a Standard for Particle Swarm Optimization" *Proceedings of the 2007 IEEE Swarm Intelligence Symposium (SIS 2007)* 2007
- [18] R. H. Babak, O. Cohanım, L. de Weck, "A Comparison of Particle Swarm Optimization and the Genetic Algorithm" *Proceedings of the 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Material Conference*, April 2005
- [19] K. Mason, J. Duggan, E. Howley, "A Meta Optimisation Analysis of Particle Swarm Optimisation Velocity Update Equations for Watershed Management Learning". *Applied Soft Computing*. 62: 148–16, 2018
- [20] D. Karaboga, "An Idea Based on Honey Bee Swarm For Numerical Optimization", *Technical Report - TR06*, 2005
- [21] B. Akay, "A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding", *Applied Soft Computing Volume 13, Issue 6*, June 2013
- [22] L. Panait, Sean Luke, "A comparison of two competitive fitness functions", *Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation*, July 2002
- [23] A. M. Turing, "Computing Machinery and Intelligence", *A Quarterly Review of Psychology and Philosophy*, VOL. LIX. No.236. October, 1950
- [24] N. A. Barricelli, "Symbiogenetic evolution processes realized by artificial methods". *Methodos*: 143–182. 1957
- [25] H.P. Schwefel, "Numerical optimization of computer models (Translation of 1977 Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie. Chichester", *New York: Wiley*. 1981
- [26] J. McCall, "Genetic algorithms for modelling and optimisation", *Journal of Computational and Applied Mathematics Volume 184, Issue 1*, December 2005
- [27] H. Safak, M.t Sahin, B. Gülveren, M. Tomak, "Efficiency of Genetic Algorithm and Determination of Ground State Energy of Impurity In a Spherical Quantum Dot" *International Journal of Modern Physics C* Vol. 14, No. 06, pp. 775-784, 2003
- [28] J.J. Grefenstette, "Optimization of control parameters for genetic algorithms". *IEEE Transactions Systems, Man, and Cybernetics*. 16 (1): 122–128, 1986
- [29] Mercer and Sampson, "Adaptive Search Using a Reproductive Metaplan", *Kybernetes* 7(3) 215-228, 1978.
- [30] V. I. Levin, "Optimization in terms of interval uncertainty: The determinization method" *Automatic Control and Computer Sciences*, Volume 46, Issue 4, pp 157–163, July 2012
- [31] Y. Jin "Surrogate-assisted evolutionary computation: Recent advances and future challenges", *Swarm and Evolutionary Computation*, 1(2):61–70, 2011

- [32] P. J. Bradley, "Quasi-Newton Model-Trust Region Approach to Surrogate-Based Optimisation of Planar Metamaterial Structures" *Progress In Electromagnetics Research B*, Vol. 47, 1–17, 2013
- [33] D. C Sorensen, "Newton's Method with a Model Trust Region Modification", *SIAM J. Numer. Anal.*, 19(2), 409–426, 1982
- [34] Z. Bayraktar, M. D. Gregory, X. Wang, and D. H. Werner, "Matched Impedance Thin Planar Composite Magneto-Dielectric Metasurfaces", *IEEE Transactions on Antennas and Propagation*, April 2012.
- [35] D. Michie, D. J. Spiegelhalter, C. C. Taylor, *Machine Learning, Neural and Statistical Classification*. Ellis Horwood, 1994
- [36] A. Samuel, "Some Studies in Machine Learning Using the Game of Checkers". *IBM Journal of Research and Development*. 3 (3): 210–229, 1959
- [37] W. B. Frakes, R. Baeza-Yates, "Information Retrieval Data Structures & Algorithms.", *Prentice-Hall, Inc*, 1992
- [38] C. M. Bishop, *Pattern Recognition and Machine Learning* , Springer, 2006
- [39] R. Kohavi, Foster Provost "Glossary of terms". *Machine Learning*. 1998
- [40] M.r Mohri, A. Rostamizadeh, A. Talwalkar "Foundations of Machine Learning", *The MIT Press*, 2012
- [41] B. Liu, H. Aliakbarian , Z.Ma, G. A. E. Vandenbosch, G. Gielen and P. Excell, "An Efficient Method for Antenna Design Optimisation Based on Evolutionary Computation and Machine Learning Techniques" 2014.
- [42] C.E. Rasmussen, M. Kuss, "Gaussian Processes in Reinforcement Learning", *Advances in Neural Information Processing Systems 16Proc. Ann. Conf. Neural Information Processing Systems*, pp. 751-759, 2004.
- [43] R., Lior, and O. Maimon. "Clustering methods." *Data mining and knowledge discovery handbook*. Springer US, 2005.
- [44] A.Y. Ng, M.I. Jordan, Y. Weiss, "On Spectral Clustering: Analysis and an Algorithm", *Advances in Neural Information Processing Systems*, MIT Press, 14: 849–856, 2001
- [45] S. Firdaus, Md. A. Uddin, "A Survey on Clustering Algorithms and Complexity Analysis", *IJCSI International Journal of Computer Science Issues*, Volume 12, Issue 2, March 2015
- [46] C. Chatfield, *Problem Solving: A Statistician's Guide (2nd ed.)*, Chapman and Hall, 1995
- [47] S. P. Lloyd, "Least square quantization in PCM". *Bell Telephone Laboratories Paper*, 1957

- [48] HP Kriegel, E. Schubert, A. Zimek, "The (black) art of runtime evaluation: Are we comparing algorithms or implementations?". *Knowledge and Information Systems*. 52: 341–378. (2016).
- [49] S. T. Roweis, L. K. Saul "Nonlinear Dimensionality Reduction by Locally Linear Embedding", *Science* 22, December 2000
- [50] E. Alpaydin, "Introduction to Machine Learning", *London: The MIT Press*. p. 110, 2010
- [51] H.P. Kriegel, P. Kröger, A. Zimek, "Subspace clustering". *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*. 2 (4): 351–364. July 2012
- [52] C. Ding, X. He, "K-means Clustering via Principal Component Analysis", 2004
- [53] P. Prusinkiewicz, A. Lindenmayer, *The Algorithmic Beauty of Plants*, Springer, 1990
- [54] D. J. Kern, D. H. Werner, M. Lisovich "Metaferrites: Using Electromagnetic Bandgap Structures to Synthesize Metamaterial Ferrites", *IEEE Transactions on Antennas and Propagation*, vol. 53, issue 4, pp. 1382-1389, April 2005
- [55] T. Eavis, D. Cueva,. "A Hilbert space compression architecture for data warehouse environments". *Lecture Notes in Computer Science*. 4654: 1–12. 2007
- [56]D. Lemire, O. Kaser, "Reordering Columns for Smaller Indexes". *Information Sciences*. 181 (12). 2011
- [57] S Hall, *Fractals in D3: Dragon Curves* 2014
- [58] M. van Dongen, <http://www.texample.net/tikz/examples/hilbert-curve/> 2016, accessed January 2018
- [59] Tango - <https://commons.wikimedia.org/w/index.php?curid=38823099>, 2015

Chapter 4: Design and Characterisation of Metaferrites

4.1: Basic Design

Once the replicated structures had been studied and confidence of the parameters and responses expected of simulated metaferrite structures had been developed, the next step was to develop some sample structures to validate the design approach. To that end, the Genetic Algorithm program detailed in chapter 3 was used to design and optimise metaferrite structures, based on a template and parameters set beforehand. This chapter deals with the process of how the nature of that template and those parameters was arrived upon.

The basic shape of the metaferrite arose partially in imitation of previous designs, such as [1] and partially as a consequence of the constraints of the genetic algorithm. It consisted of a square, rotationally symmetrical structure with additional lines of symmetry bisecting each edge and the diagonals. This facilitated a construction method that began with a small, triangular patch of square pixels. These pixels were either filled or left empty according to a data string generated by the genetic algorithm, and once complete this triangular patch was reflected along the diagonal line of its hypotenuse forming a square. This square was then rotated around the central point of the unit cell, forming the final shape.

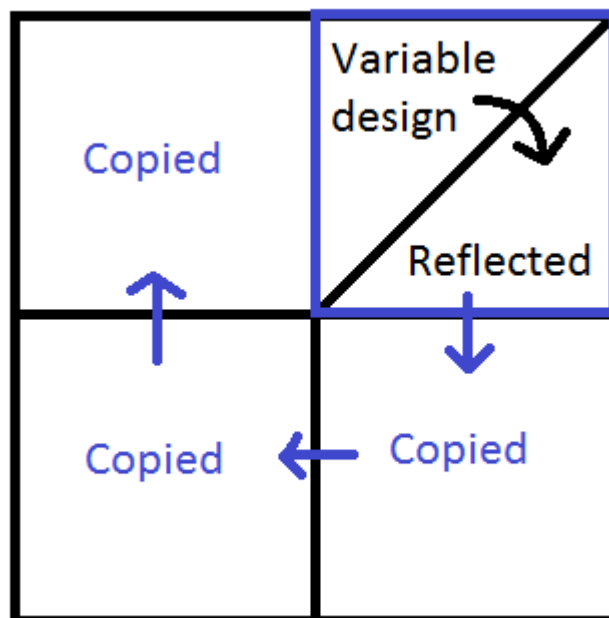


Figure 4.1: Diagram of the construction process of the metaferrite structures

Beyond this basic shape, however, there were numerous parameters that needed to be considered.

4.2: Determination of Optimal Unit Cell Size.

To begin with, an assessment needed to be made of the optimal unit cell size for the structures. Three sizes were studied, each varying by changing the number of layers on the core triangular pattern of the structure. When expanded to full size, this resulted in square unit cells 16, 18 and 20 mm in width. Owing to differences in how the pixels were arranged for each size of structure, there were a number of subtle differences in both the typical shapes they produced, even when using identical templates. As a result of this the S11 parameters each of these structures was correspondingly different.

The 16 x 16 structure had a randomised patch consisting of 36 pixels. This was the size of the structure used prior to the introduction of the L-system expansion method [2], when each pixel was randomly determined. As such there was already available data regarding the typical parameters and limitations of a structure this size, and this formed the standard by which the other variants were compared.

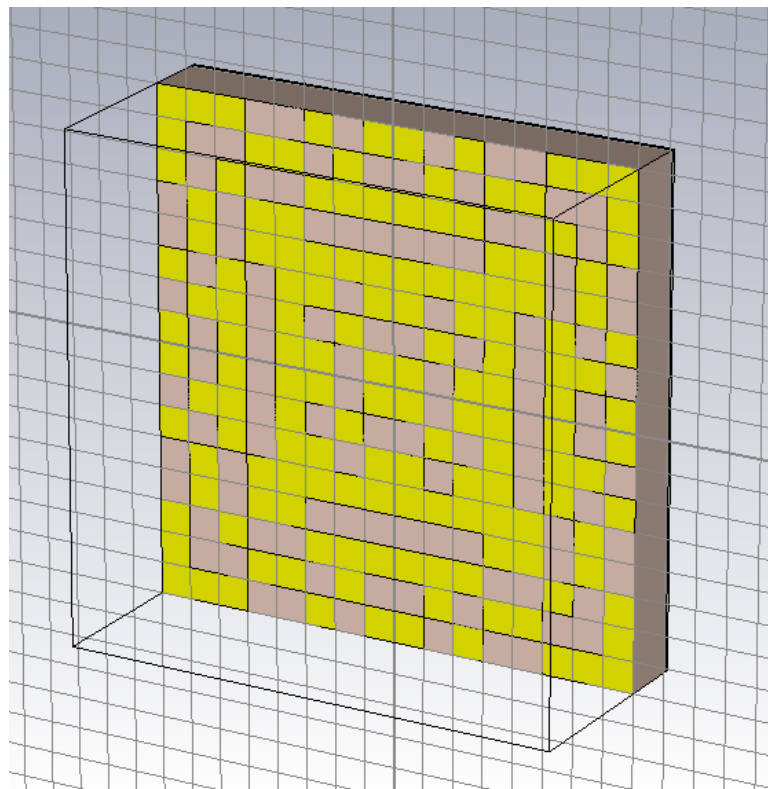


Figure 4.2: Typical appearance of a randomly-generated 16 x 16 mm

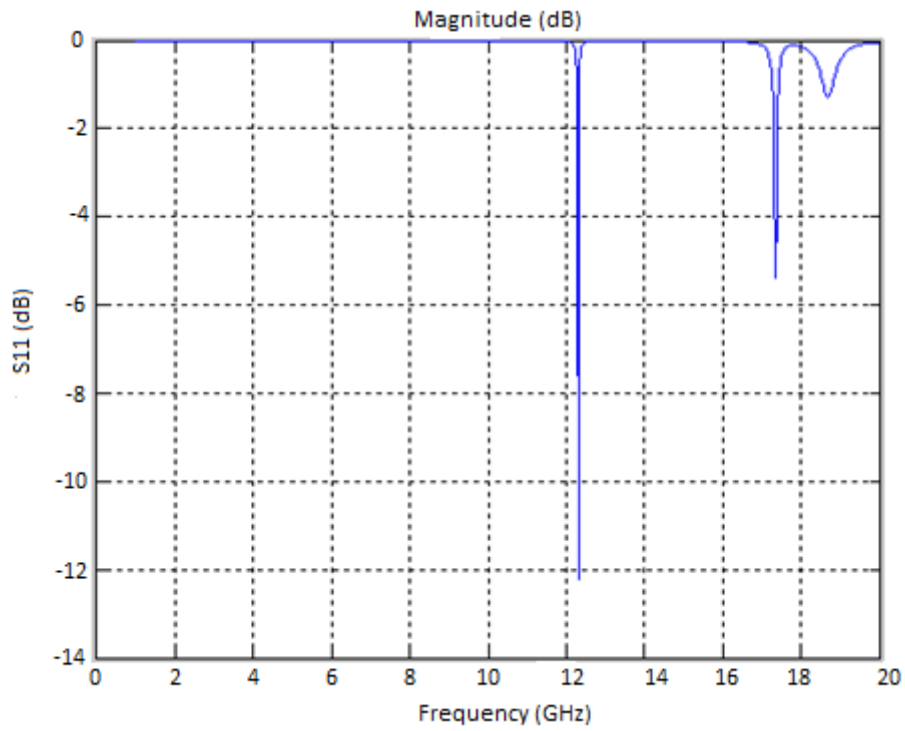


Figure 4.3: Example of resonance values for a randomly-generated 16 x 16 mm structure

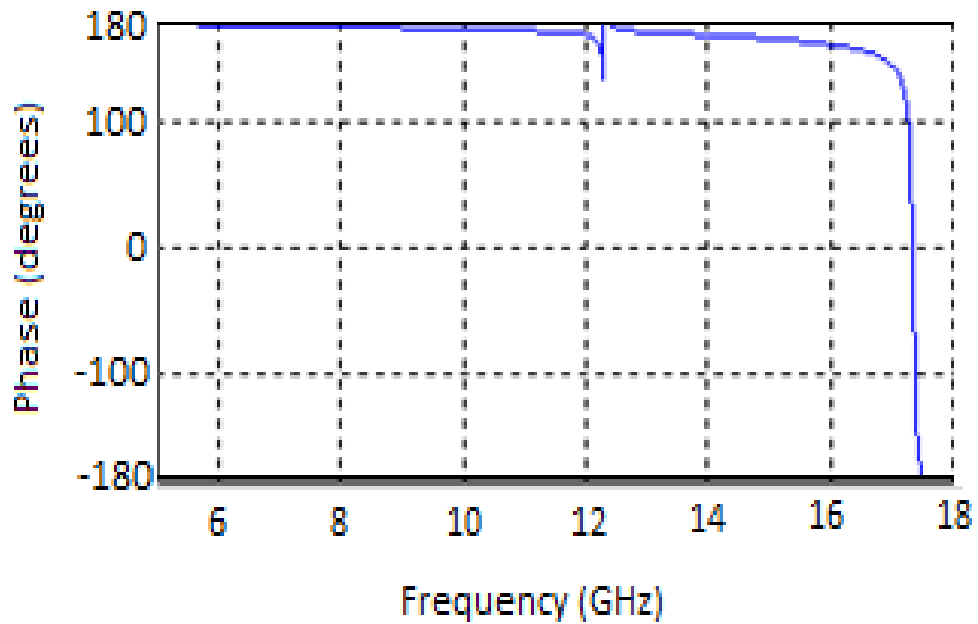


Figure 4.4: Example phase for a randomly-generated 16 x 16 mm structure

Structures of this size typically displayed responses similar to the example depicted in figure 4.5. There are a few advantages associated with smaller structures, such as greater simulation speed, but the concern was that the inclusion of the L-system (itself designed to improve this factor) would reduce the possible design space [3], and it was felt that a larger structure could alleviate this issue. As such, other two sizes of structure presented here were simulated.

The 18 x 18 mm structure used a 45 pixel patch. This struck a balance between the smaller design space and the larger time taken for analysis, in addition to the other issues with a larger structure seen below.

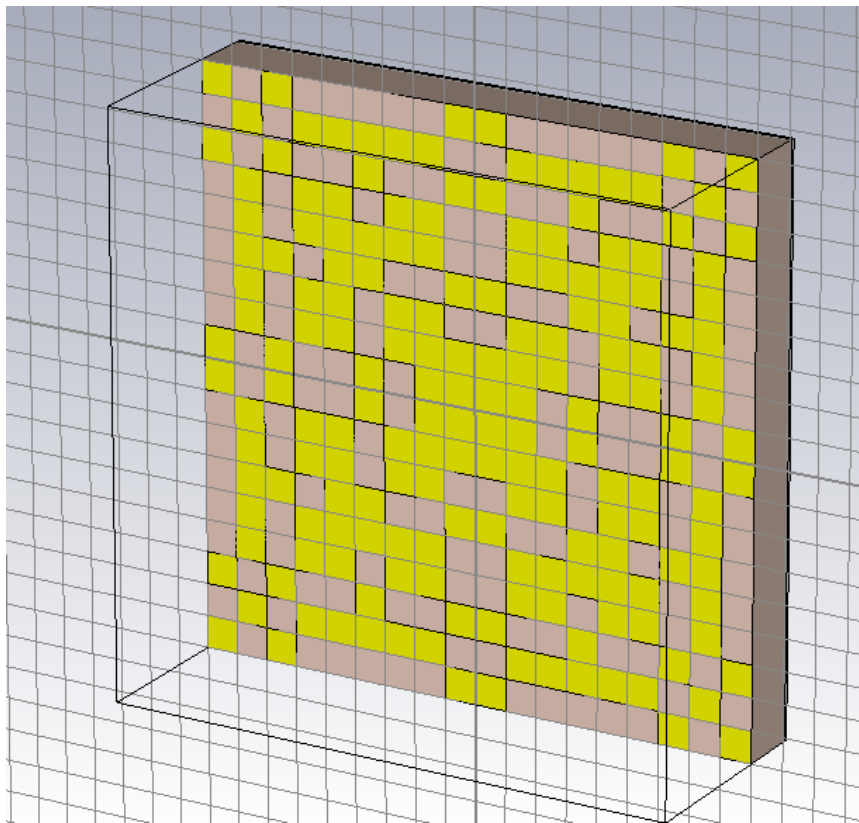


Figure 4.5: Typical appearance of a randomly-generated 18 x 18 mm structure.

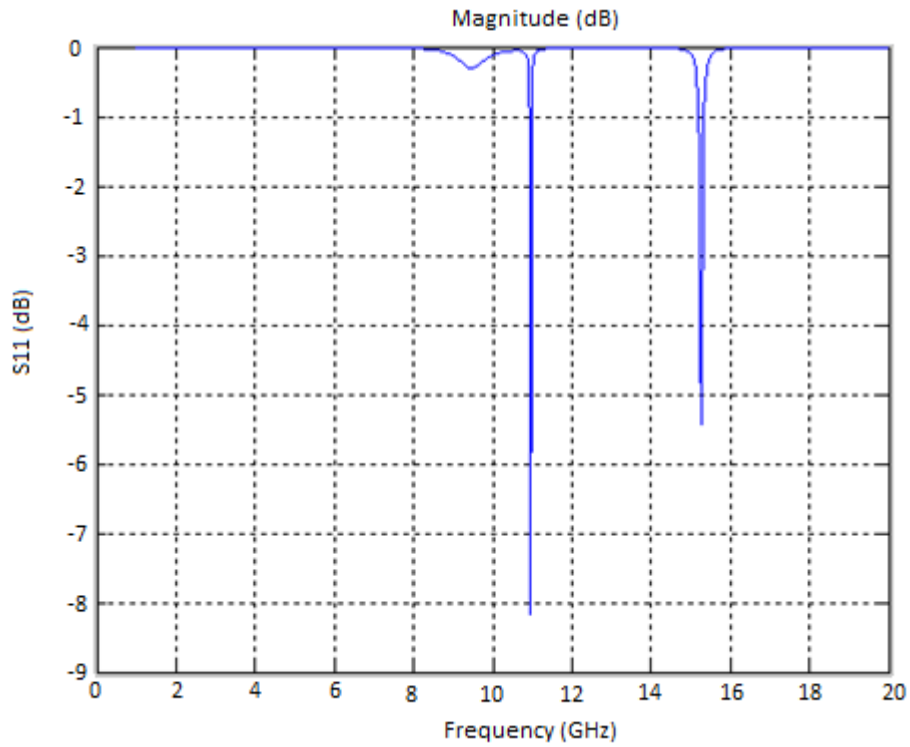


Figure 4.6: Example of resonance values for a randomly-generated 18 x 18 mm structure.

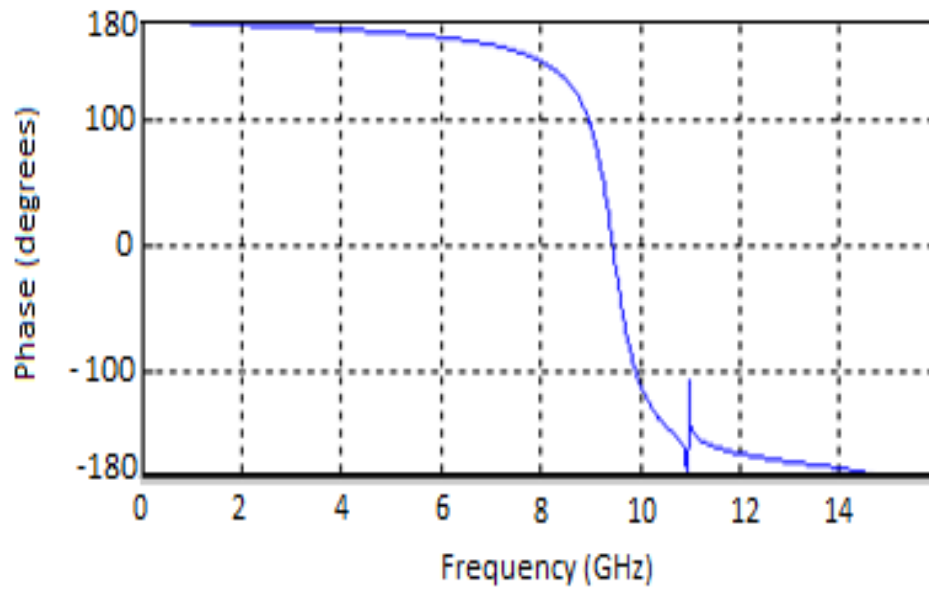


Figure 4.7: Example phase for a randomly-generated 18 x 18 mm structure. This was deemed to be the one most likely to produce a working metaferrite structure with the desired properties in the lowest number of generations.

The 20 x 20 structure used a 54 pixel variable patch. This granted the structure an even larger potential design space than the previous design, but was not without problems. Analysis of the reflection showed a greatly increased number of reflections scattered around a wide range of frequencies. There was concern that having multiple frequency bands in this manner would cause issues with assessment by the fitness function, so it was judged that a structure of this size should probably not be used despite the greater design space afforded. As such, it was ultimately judged that the 18 x 18 structure presented previously was the optimal form for the metaferrite designs to take.

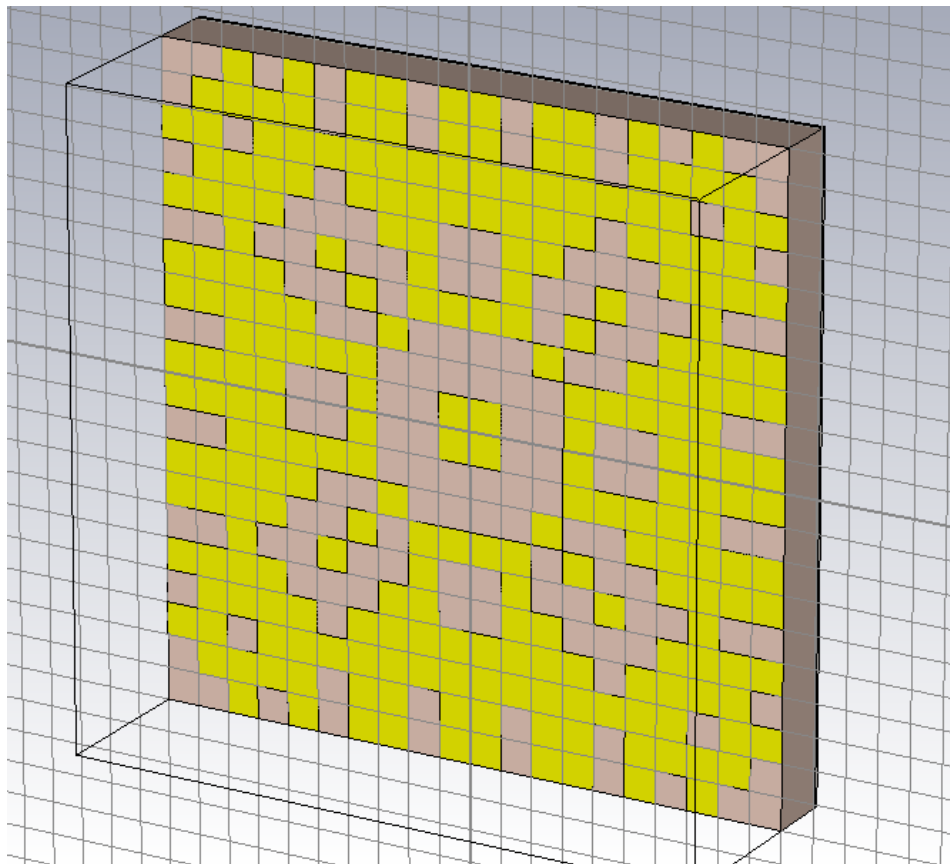


Figure 4.8: Typical appearance of a randomly-generated 20 x 20 mm structure

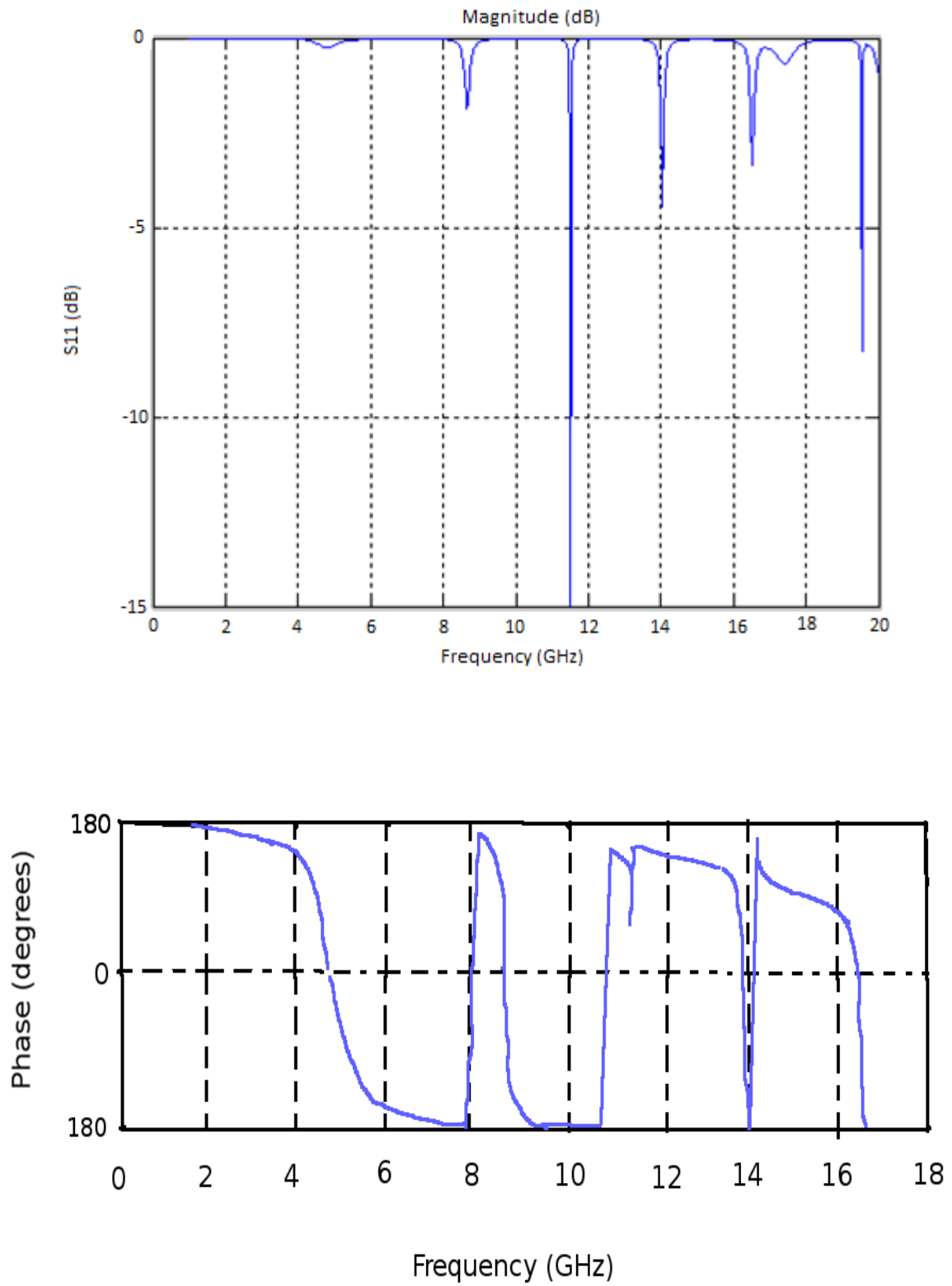


Figure 4.9: Example resonance values and phase for a randomly-generated 20 x 20 mm structure

A perfect magnetic conductor (PMC) provides 0 degrees reflection phase. Therefore, a metaferrite structure aiming to replicate the material response of a magnetic material should aim to achieve the same value. Since a metasurface can theoretically achieve any reflection phase desired, this is well within the bounds of feasibility. [4] In practice, however, such control is bound to a narrow operating frequency, dependent on the structure in question. Changing the structure size will, naturally, have an effect on this frequency. While the two smaller structures made it easy to find a functional operating frequency, the multiple resonances displayed by the 20 x 20 pixel structure made it hard to pin down.

4.3: Analysis of desired features

The structure depicted in figure 4.10 was replicated using CST to validate that the same result that was presented in [6] could be produced by the software. This was successful. The same technique was then applied to other designs shown below. It was quickly concluded that the low loss structures the project was aiming for have a profile characterised by a fairly short, wide-based peak. Since the structure generally produced other resonances of higher amplitude, this meant that isolating the desired one was slightly more complex than simply using the minimum-finding function packaged with MATLAB. Instead the approach used was to identify each peak in turn, using a copy of the magnitude matrix and removing each peak as it is found, before displaying the values at the same point in the real matrix.

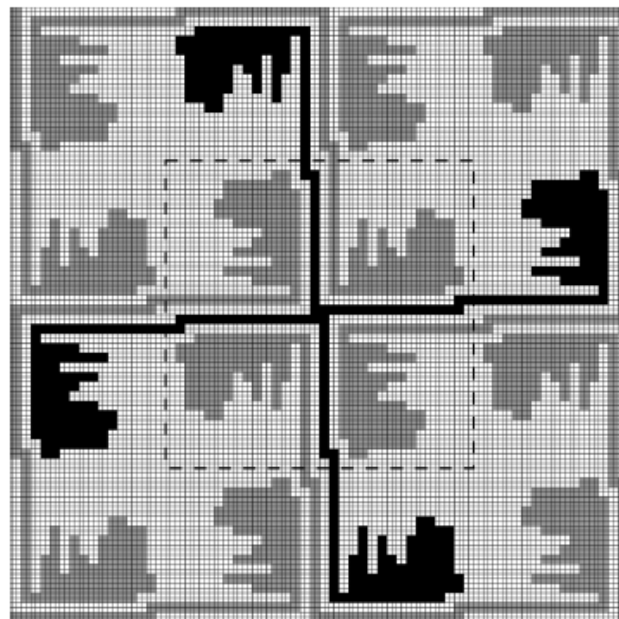


Figure 4.10: Structure replicated for the purposes of comparison [14]

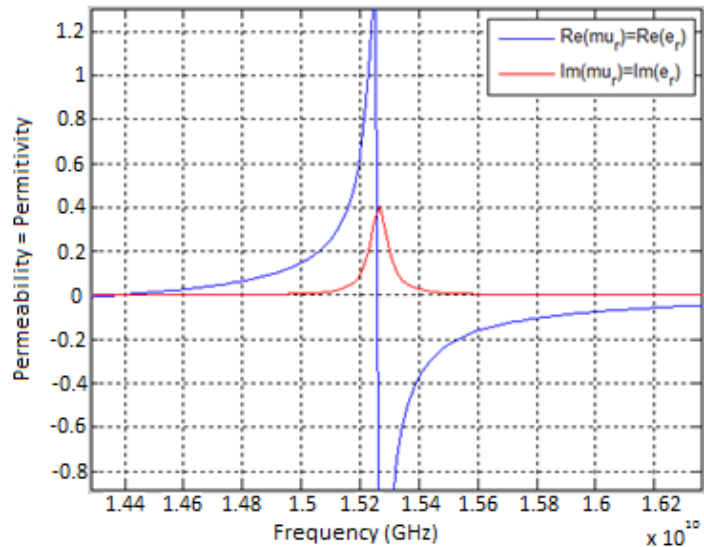
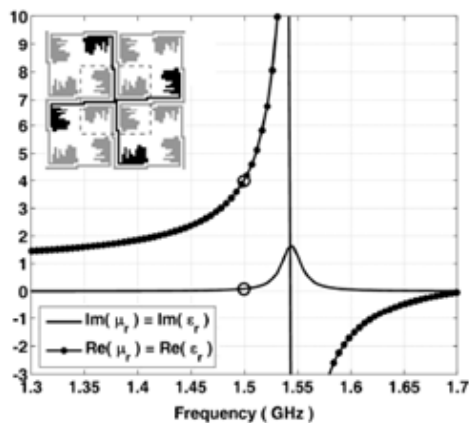


Figure 4.11: Comparison of Permittivity and permeability of structure in Gonzalo's's paper [5] with one of several peaks in the reflection phase of a structure randomly designed by the project, characterised by being of the shorter and wider variety, indicating that this is the type of pattern to aim for. There is a notable difference regarding the amplitudes of the two peaks, but since this is the focus of the fitness function it is thought that this will be corrected in the fully optimised structure.

It was also determined that, if possible, it was best to aim for patterns with fewer independent pixels. These produce better reflection profiles, as they possess a more coherent pseudo-capacitor/inductor mesh [7]. This condition could be enforced by manipulating the L-system generator so that it always produces sections of a minimum length, but this has the downside that it can interfere with the repeating pattern as it would allow for fewer iterations of the algorithm. One must be careful to avoid overly restricting the design space when designing a genetic algorithm, as the entire point is to explore options that would not normally be considered.

4.4: Inclusion of Optimisation Parameters

Once a rough idea of what it should be aiming at had been cemented, a fitness function based on the one in Kern *et al's* paper [1], [8] was inserted into the Genetic Algorithm and true optimisation could begin. As would be expected, this yielded more consistent results at resonances far closer to those desired, even after only a few generations. There was an issue with the magnitude of the parameters predicted being far higher than they

should be at first, but this problem was eventually corrected. An example of the responses predicted by the program at this point is shown in figure 4.12. It was felt that these results could be improved upon though. In pursuit of this, variations on the structure were also designed and analysed. One modification studied was the addition of a layer of substrate and a square patch of metal in front of the metallic structure. This initially seemed to have the effect of reducing the resonance frequency values by approximately 4 GHz, as shown in figure 4.13.

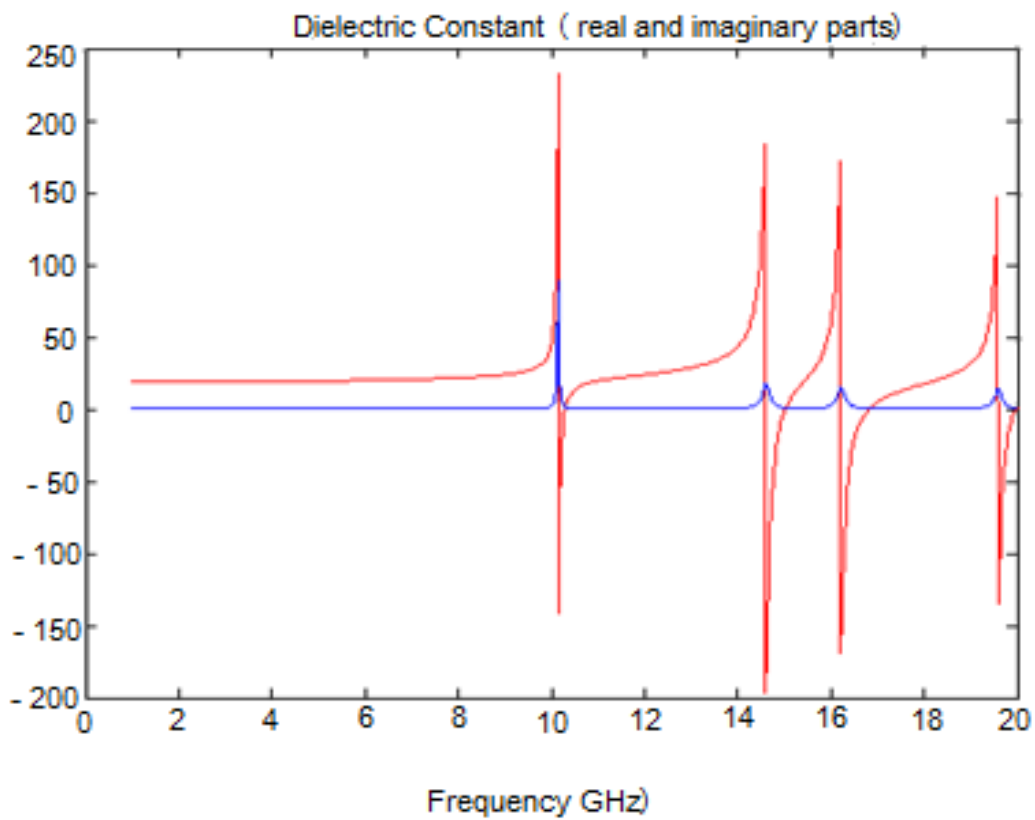


Figure 4.12: Comparison of real (red) and imaginary (blue) components of the permeability and permittivity of the structure (which are assumed by equations to be equal to each other) as predicted at an early stage of the project.

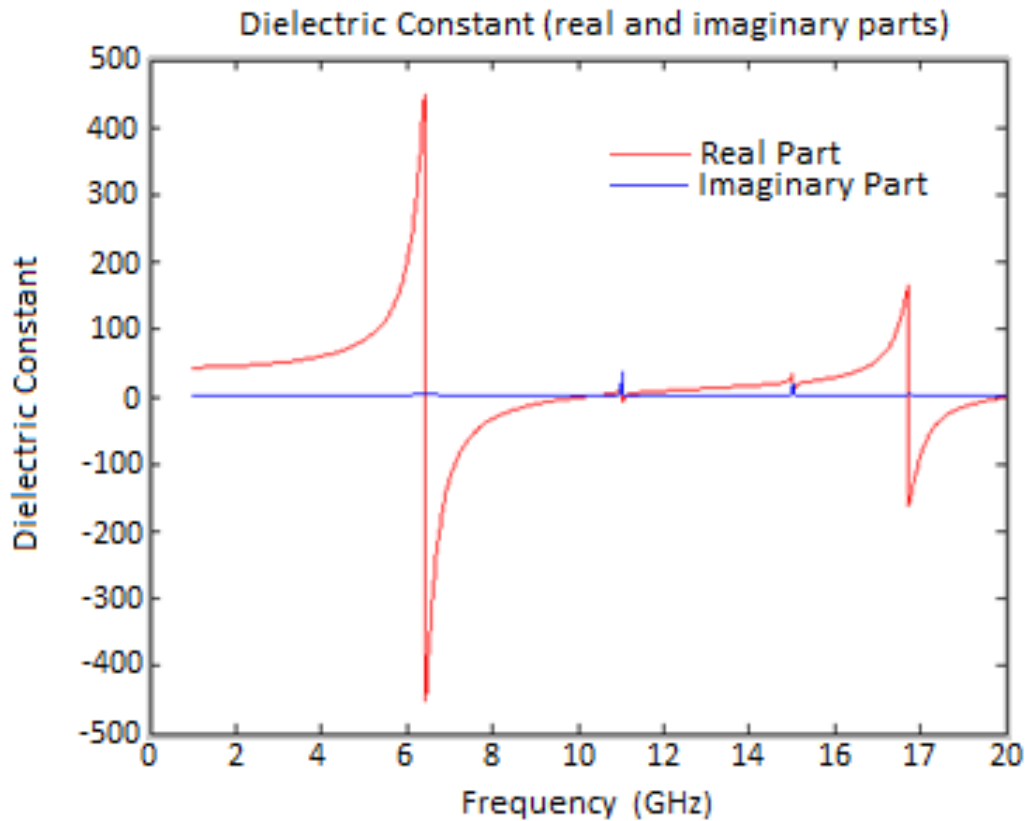


Figure 4.13: Real (red) and imaginary (blue) components of permeability of structure when placed behind a layer of substrate and patch antenna.

This result is far closer to the response seen in the paper [8] that the structure was attempting to emulate, aside from the yet-to-be-corrected magnitude problem, and while the optimisation process in this case was not completed, it can be seen that it was already approaching the desired form. Unfortunately, subsequent simulations involving such a structure proved to not merely be similar, but highly identical. This caused some concerns to be raised regarding how much influence the underlying high-impedance surface actually had. It seemed that very little of the incoming wave was reaching the second layer of the structure, and the response was almost entirely due to the square patch on the front.

This was studied further using a variant consisting of a large patch covering several of the rotationally symmetric unit cells. A 3 x 3 structure was chosen for this purpose as it was felt that this provided a good balance between a single unit cell and a continuous boundary. It displayed several additional resonances of decreasing amplitude at high frequencies. This was

somewhat unexpected, but was at least different to the results seen with the previous square patch.

There are a few possible explanations for the result. It may have been due to the increased size of the structure [9], or the fact that rather the structure had a finite size rather than an infinitely repeating pattern [9]. If the latter was the root cause it possibly indicates that slight adjustments to the analysis configuration were required, as the values were calculated based off the s11 parameters of the structure and so would be vulnerable to reflection, which the repeated, fading resonance points resemble. Since the commercial software used employs Finite Integral Time Domain modelling [10], a flaw in the boundaries of the simulation could conceivably cause this.

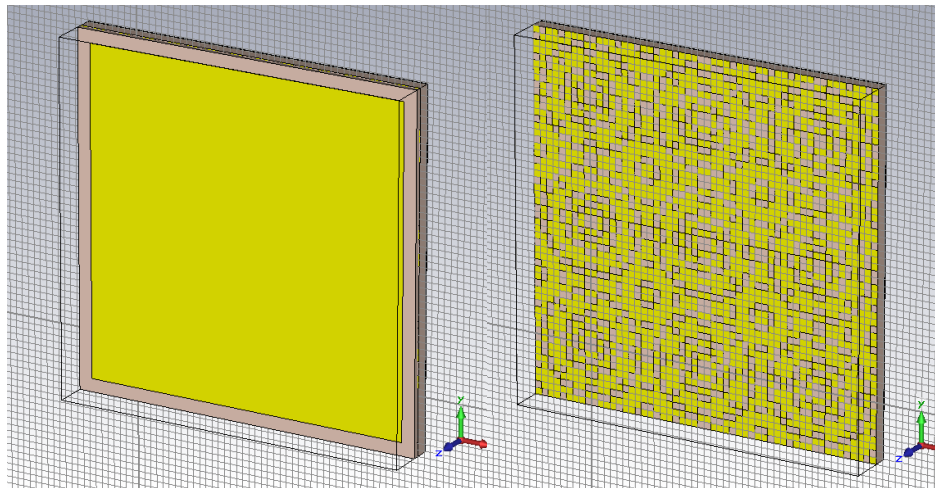


Figure 4.14: Multiple unit cell structure, with and without a square patch and additional layer of substrate placed in front

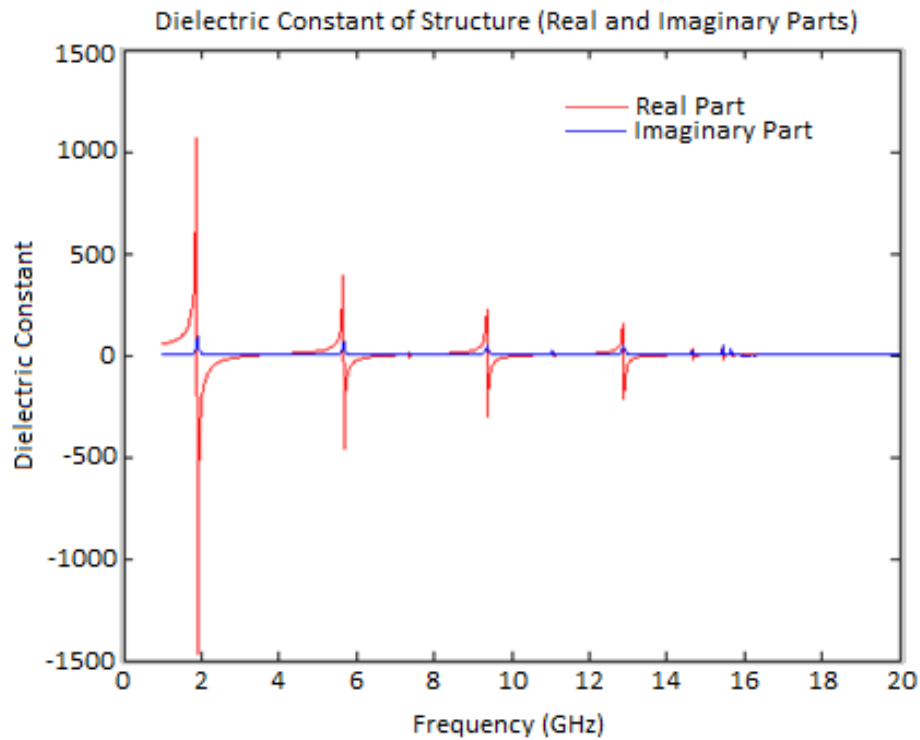


Fig 4.15: the resonance pattern produced by the structure in figure 4.14

4.5: Analysis of the Effects of Metallic Fraction on High-Impedance Surface Behaviour

Later, tests were performed to ascertain the effect of various parameter changes on the results generated by the structures designed by the Genetic Algorithm. One such test was to compare the amount of metal (measured as the fraction of the total possible area of coverage that the structure actually covers) with the value of the lowest frequency resonance point, within the range analysed.

This was done by calculating the fraction for each structure generated by the GA and then plotting said fraction against the lowest frequency value of the resonance point of the dielectric response. When a sufficiently large number of data points are collected in this manner, the following relation is seen:

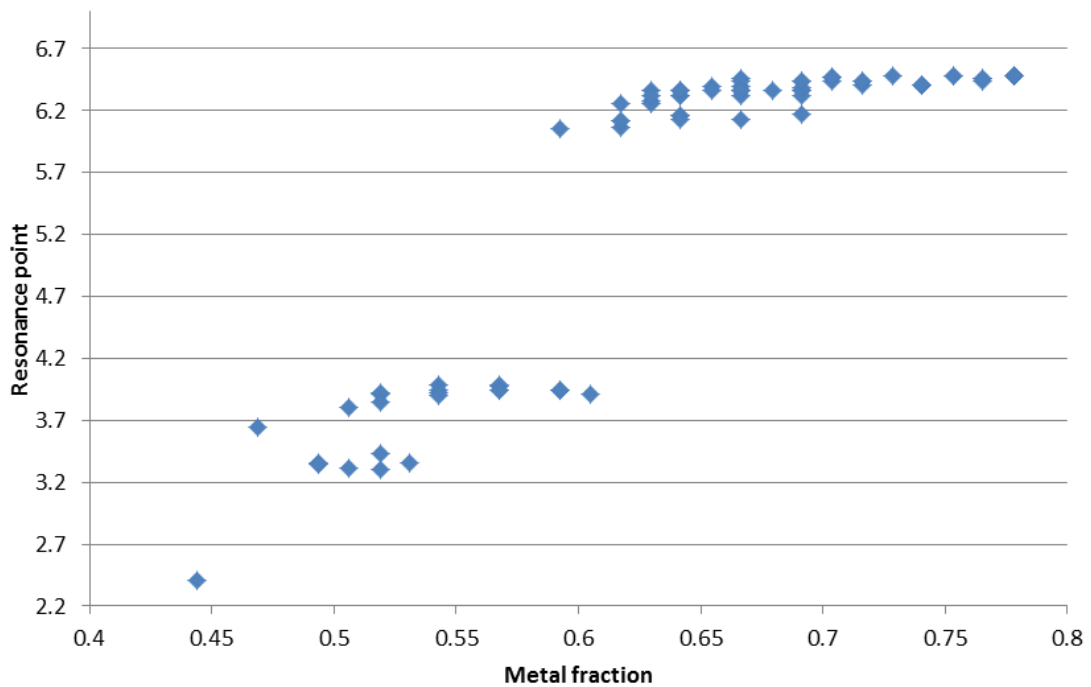


Figure 4.16: frequency of lowest resonance point of numerous structures plotted against the fraction of said structure composed of metal

As can be seen in figure 4.16, there appeared at first to be a positive correlation between the lowest resonance frequency and the metal fraction of the structure. The groupings seen are thought to have arisen due to the data points depicted being comprised of several different variants on the program, mostly centred on alterations made to the L-system expansion. This was done because an individual L-system actually turned out to have minimal variance in the metal fractions it produced, and so running multiple simulations with different ones was required to change the metallic fraction by a notable degree and provide a usable range of samples. The fact that this was required is actually potentially rather useful, as it provides a means of controlling the expected metal fraction and thus the resonance point expected on average. However, it may be more due to the varying designs that accompanied the different tiers of metal fraction rather than an effect of metal fraction directly.

Another test performed dealt with the effect of the dielectric constant of the substrate and its thickness. Several combinations were tested, but ultimately while this proved to some

effect on the location of the point of resonance, said effect was too small to really be of any practical use when designing a metaferrite structure. The main goal of the project is to make artificial ferrites that can be used for low-profile antennas, after all [13]. If the substrate was several centimetres thick this would defeat the point, as any gains to be had by placing components closer together would be cancelled out by their sheer bulk.

4.6: Post-correction Designs.

Later on, the accuracy of certain equations within the genetic algorithm were called into question. Once these faults were corrected, it was deemed prudent to run another round of simulations in order to ascertain the exact effects the modification had upon the function of the program. Several examples of the structures produced by this modified program are presented below, in figures 4.18 to 4.25.

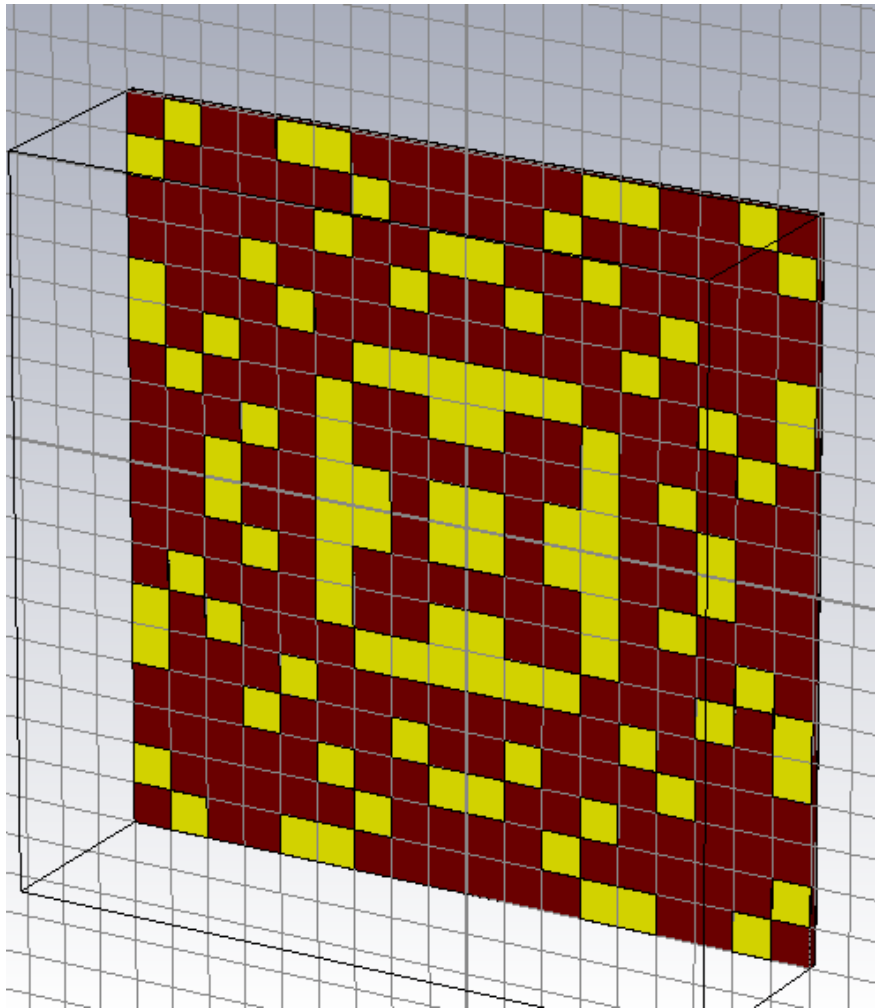


Figure 4.17: Structure generated after 35 generations by corrected algorithm

To begin with the L-system used produced a structure with a fairly low metallic fraction. The metal portion of structures seen in the initial population was notably even sparser than that seen later ones, indicating that this level of metallic fraction was a little too low. Clearly this was not something that was beyond the power of the GA to correct. as long as it had sufficient design space this should not cause trouble. The reason for choosing such a low-density L-system parameter was in order to avoid the issues that could arise at high metal fractions, most notably the potential for the entire structure to be made out of a single piece of metal with just a few holes cut out of it. This would eliminate the capacitor nature of the structure, a vital component of high impedance surfaces, and instead render it more of a slightly modified PEC.

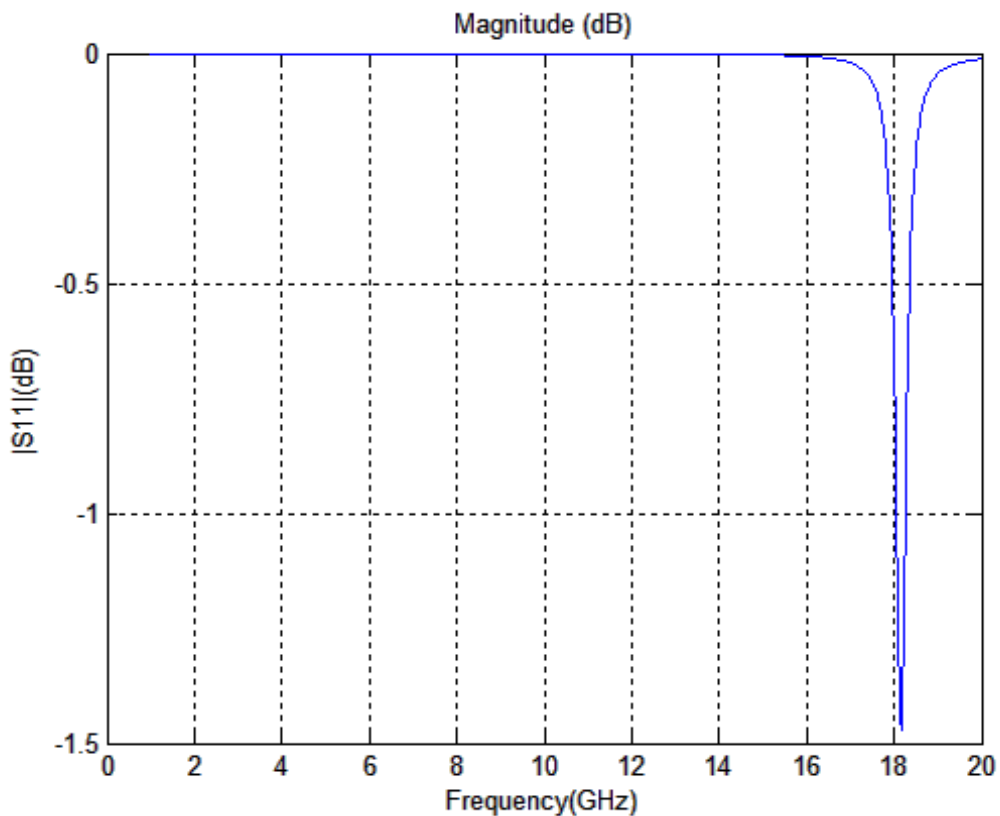


Figure 4.18: Magnitude parameter of alternate structure generated after 35 generations by corrected algorithm

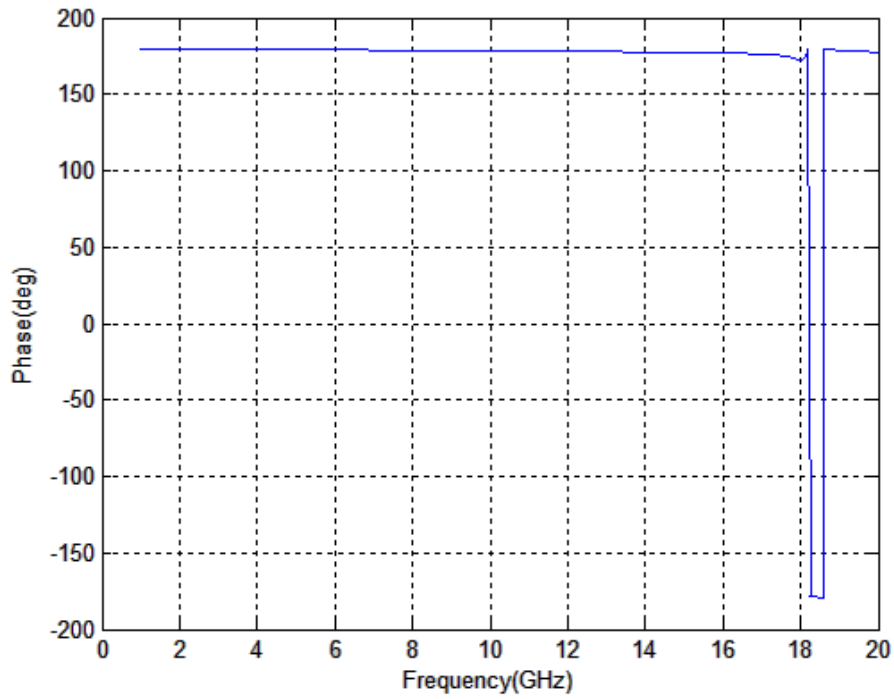


Figure 4.19: Phase parameter of alternate structure generated after 35 generations by corrected

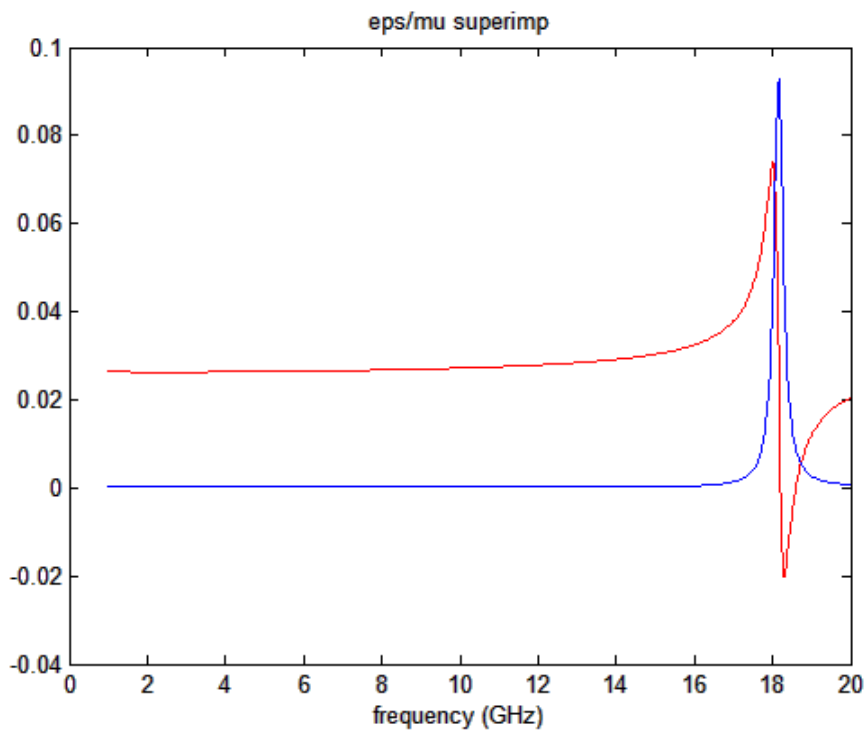


Figure 4.20: Dielectric response parameter of alternate structure generated after 35 generations by corrected algorithm

These structures displayed a small resonance peak at a fairly high frequency, far higher than that seen in the structures designed prior to the corrections to the GA. This was merely generation 35, however, so it was entirely possible that leaving the program to run for longer would correct any issues seen here, particularly those related to the magnitude of the material responses, such as the dielectric response showing greater levels of losses than desired. However, given how low the metallic fraction generated by the L-system used here was, it was felt that it would be better to modify it and run the algorithm from the start than allow this version to continue.

This version of the program was left to run for far longer than the previous iteration, and one of the best examples this alternate variant produced was the following structure, seen after 101 generations:

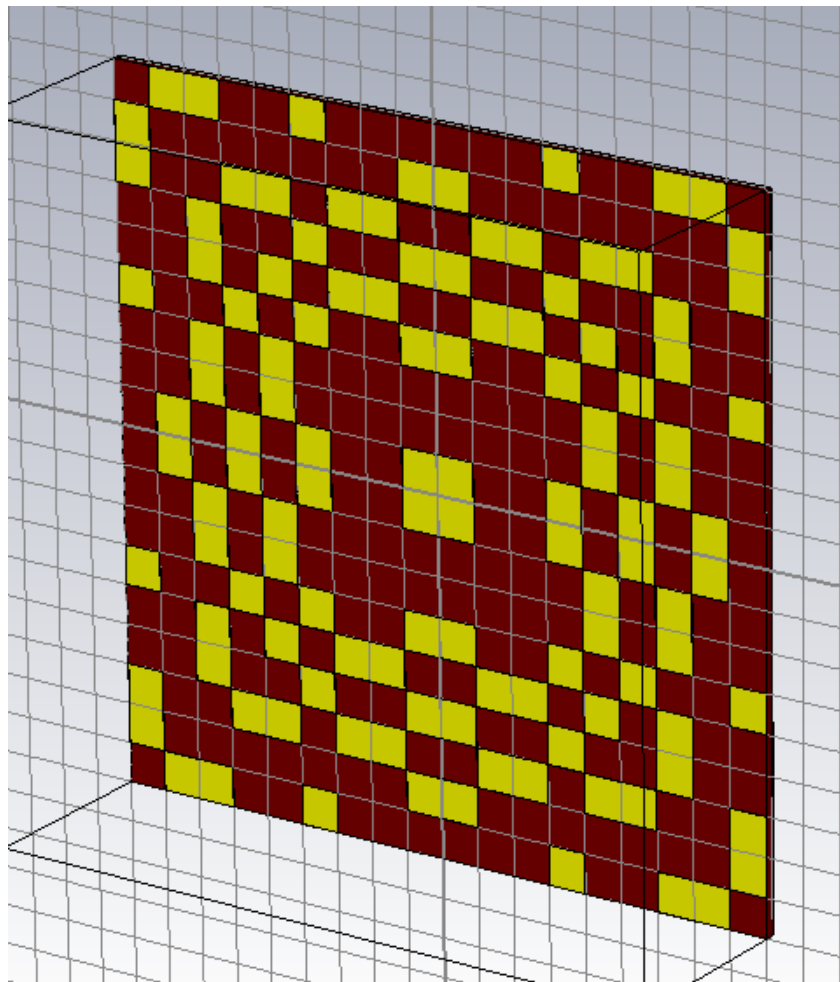


Figure 4.21: Alternate variant of structure generated by corrected algorithm after 101 generations

This structure had the following material responses:

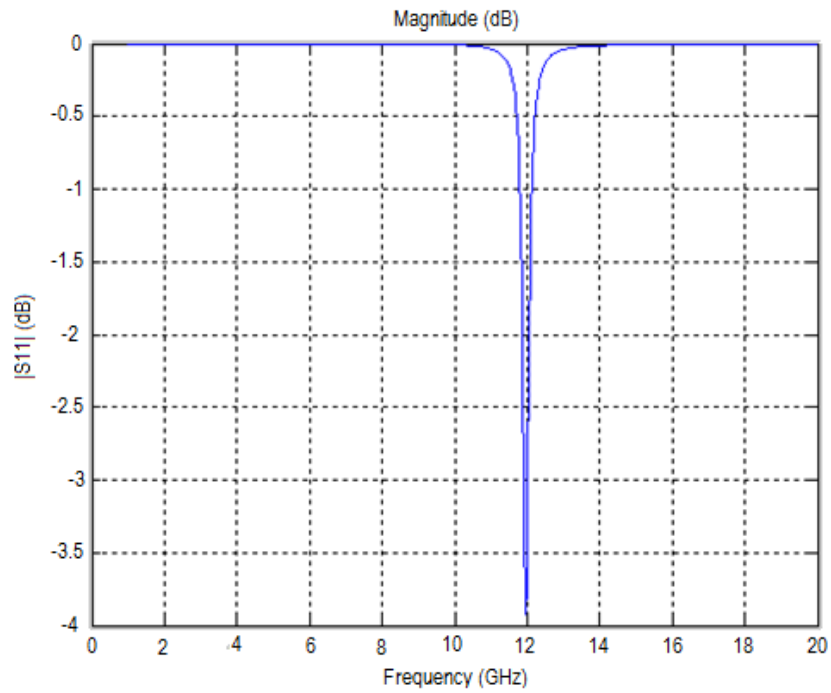


Figure 4.22: Magnitude response of alternate variant of structure generated by corrected algorithm after 101 generations

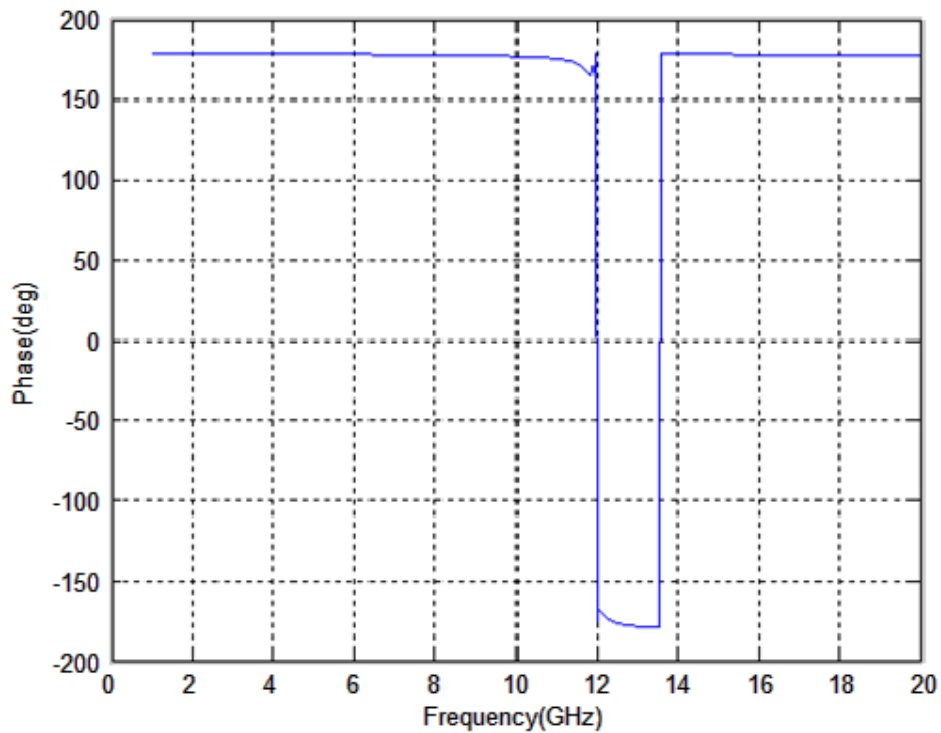


Figure 4.23: Phase response of alternate variant of structure generated by corrected algorithm after 101 generations

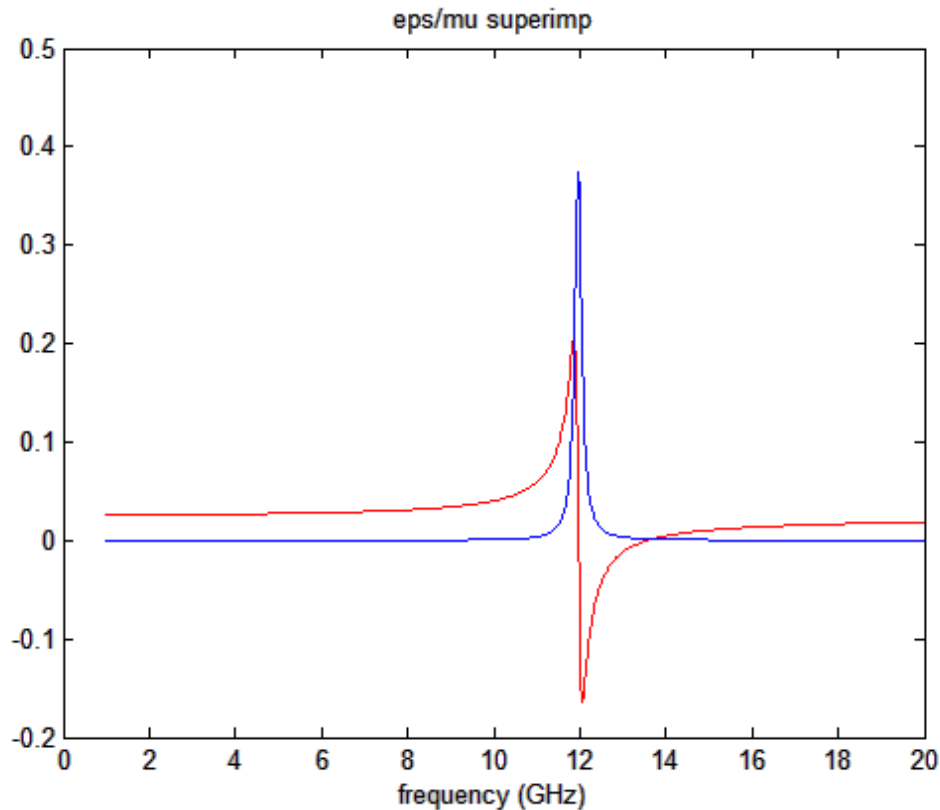


Figure 4.24: Permittivity response of alternate variant of structure generated by corrected algorithm after 101

These results showed a good resonance peak at 12 GHz, as well as a wide phase shift at the same point, indicating a good bandwidth. These were not an exact match for the goal, but it was felt that they were close enough that further optimisation would be able to close the gap.

Results of this nature seem to persist indefinitely from this point onwards, indicating that this may be the best that this particular configuration of the program can manage. Modifications to the base parameters may facilitate a better agreement with the goal at this stage.

4.7: Summary

A range of Artificial Magnetic Conductor designs were experimented with in order to determine the optimal parameters to serve as a basic template for the project. The parameters tested included structure size, substrate thickness, the materials used and even the presence of additional components such as a back plate and square patch antenna. These results were

compared with those of similar projects described in academic papers. Other more unusual designs such as placing a large patch over multiple unit cells were also investigated.

Other tests on the various parameters were also performed. The effect of metal fraction on the response of the structure was investigated, but after an initial false positive was found to be of negligible effect. The same was true of substrate thickness. Whilst there was a correlation seen, it was determined that capitalising on it would have been impractical given the ratio of variables involved.

The final conclusion was that an 18 x 18 mm unit cell was optimal for the purposes of the project, ideally constructed using a metal such as silver or copper. The reason for this was that a structure of this size had a greater design space than smaller structures, owing to the particular method of construction involving multiple 1mm by 1mm squares, but also remained free of the distortions in the signal seen when the structure became too large. The metals chosen are due to the fairly large conductivity required to generate the desired results. Conductive ink was also investigated as an option, but ultimately simulations indicated that it was unfeasible, as the lower conductivities available were insufficient to produce those same results.

Once the basic parameters necessary for the goals of the project were established, designs following that template were then subjected to computational optimisation using the genetic algorithm method in order to refine them and design a novel metaferrite structure that could be fabricated. This took several forms, as multiple designs were fabricated for the purposes of comparison between fabrication techniques. The materials available for these techniques also informed the initial parameters of the simulation. This is another reason that silver and copper were tested, and why conductive ink was explored.

Initially a high-metallic-fraction series of structures were generated, but these lacked the characteristic features sought by the project. Another L-system design with less metal was attempted, and this did produce the desired responses. This indicates a consequence of high metallic fraction not brought to light by the previous study of the topic: In cases where the fraction is so high that the structure is composed of a single, linked piece of metal, the capacitance that the high impedance surface relies upon is compromised. This can lead to it not functioning as intended. For this reason, it has become apparent that lower-fraction surfaces are preferable. However, by the same token the L-system required to produce the lowest fraction structures appears to generally favour higher resonance points. The genetic

algorithm is capable of fine-tuning a structure to some degree, but first one has to ensure that its design space can easily reach the required parameters.

Later on, further designs were simulated after it came to light that there was an error in the calculation used by the genetic algorithm program. These results display many of the same features as those later fabricated, but seem to suffer from a design space issue that leaves them stuck at a local minimum of the fitness function.

4.8: References

- [1] D. J. Kern, D. H. Werner, M. Lisovich, “Metaferrites: Using Electromagnetic Bandgap Structures to Synthesize Metamaterial Ferrites” *IEEE Transactions on Antennas and Propagation Volume 53 Issue 4*, April 2005
- [2] P. Prusinkiewicz and A. Lindenmayer “The Algorithmic Beauty of Plants”
- [3] I. Y. Kim, B. M. Kwak, “Design space optimization using a numerical design continuation method”, *International Journal For Numerical Methods In Engineering*, 2002
- [4] B. O. Zhu, J. Zhao, Y. Feng, “Active impedance metasurface with full 360° reflection phase tuning” *Scientific Reports volume 3, Article number: 3059*, 2013
- [5] Z. Bayraktar, M. D. Gregory, X. Wang, D. H. Werner, “Matched Impedance Thin Planar Composite Magneto-Dielectric Metasurfaces”, *IEEE Transactions on Antennas and Propagation Volume: 60 Issue: 4*, April 2012.
- [6] G. Expósito, J.M. Fernández P. Padilla M. Sierra-Castaner, “EBG Size Reduction for Low Permittivity Substrates” *International Journal of Antennas and Propagation 2012:1-8*, December 2012
- [7] A. Arrióla, G. Sasiain, J. I. Sancho, J. Parrón, J. Gemio, R. Villarino, “Compact high impedance surface based on interdigital capacitors”, *Proceedings of the Fourth European Conference on Antennas and Propagation*, April 2010
- [8] D. Kern, “Advancements in Artificial Magnetic Conductor Design for Improved Performance and Antenna Applications”, 2009
- [9] F. Costa, S. Genovesi, A. Monorchio, “On the Bandwidth of High-Impedance Frequency Selective Surfaces” *Ieee Antennas And Wireless Propagation Letters, Vol. 8*, 2009
- [10] C. Warren, L. Pajewski, D. Poljak, A. Ventura, A. Giannopoulos, S. Sesnic, “A comparison of finite-difference, finite-integration, and integral-equation methods in the time-domain for modelling ground penetrating radar antennas”, *2016 16th International Conference on Ground Penetrating Radar (GPR)*, June 2016
- [11] <http://www.ukinsulations.co.uk/dupont-nomex/nomex-brand-papers>, accessed June 2017

- [12] F. Bonaccorso, L. Colombo, G. Yu, M. Stoller, V. Tozzini, A. C Ferrari, R. S. Ruoff, V. Pellegrini, "Graphene, related two-dimensional crystals, and hybrid systems for energy conversion and storage". *Science*. 347, 2015
- [13] M. K. T. Al-Nuaimi, W. G. Whittow, "Novel planar AMC for low profile antenna applications", *2009 Loughborough Antennas & Propagation Conference*, November 2009
- [14] Z. Bayraktar, M. D. Gregory, X. Wang, and D. H. Werner, "Matched Impedance Thin Planar Composite Magneto-Dielectric Metasurfaces", *IEEE Transactions on Antennas and Propagation*, April 2012.

Chapter 5: Fabrication and Measurement of Metasurfaces

5.1: Overview

Fabrication is an important step in the development of any computer-assisted design. After all, a metaferrite design is no good if it stays as a simulation forever. So, when it was felt that a sufficient number of simulations had been made, the clear next step for the project was to choose a design to actually fabricate. This would allow testing of the material responses and validating the predictions made by the genetic algorithm.

There were several methods of fabrication considered. These included screen printing, inkjet printing, chemical etching, and a few other techniques [1], [2]. Screen printing is a process by which ink is transferred onto a substrate via a mesh. Currently, synthetic threads such as polyester are most commonly used for the process, but nylon and even stainless steel meshes are available for more specialised applications. Historically, silk was a common material to use for such a purpose. The desired pattern is achieved by blocking certain areas of the mesh with a stencil.

Inkjet printing is a method commonly used in commercially available printers. It involves a moving ink cartridge that passes across a stationary substrate and coats it via a series of jets, hence the name. Its popularity is due to the relative speed and ease of use with which it can print documents and other such common items. Its major weakness, as will be explored later, is a lack of uniformity in its deposition profile. As such it's best suited for tasks where this doesn't matter.

Chemical etching [3] is a process by which the substrate is entirely coated with the material intended to cover it, and the unwanted portions are chemically dissolved until only the desired design remains. The exact chemicals involved vary depending on the materials used, but generally include some sort of corrosive solvent, which can make the process tricky to use. A similar technique is laser etching, which involves the same coat and remove process but which is done using a laser instead of chemical solvents. A similar result can be achieved through pcb milling, where the material is removed mechanically rather than by chemicals. This has the advantage of producing less hazardous waste and being easier to do in a lab without specialised equipment.



Figure 5.1: Basic composition of typical metaferrite structure. All of the fabricated structures outlined in this chapter conform to this design.

It was the techniques of inkjet printing and pcb milling that were eventually put into practice, however, mostly due to the availability of the required equipment. This chapter will concern itself with the results that were produced by these methods.

In both cases, the structure had the same basic composition: A high impedance surface placed on a dielectric substrate with a PEC back plate behind it, as shown in figure 5.1. What differed were the materials involved. The first material attempted was silver ink.

5.2: Silver Ink

5.2.1: The Inkjet Printing Method

The first fabrication attempt used silver ink on a PET substrate, fabricated using the inkjet printing method. Inkjet printing is a technique that allows a thin layer of conductive materials to be deposited on a substrate, making it ideal for designs like metaferrites, high-impedance surfaces and other printed electronics, where the material only needs to be a millimetre or two thick. [4]

The material to be deposited is prepared in a liquid form (the titular “ink”) and placed in a cartridge in the printer. The substrate is placed on a flat surface inside, held in place via vacuum suction. Deposition is controlled by moving the printer head containing the ink cartridge over the substrate, depositing one line of the material at a time, not dissimilar to how a standard computer printer functions.

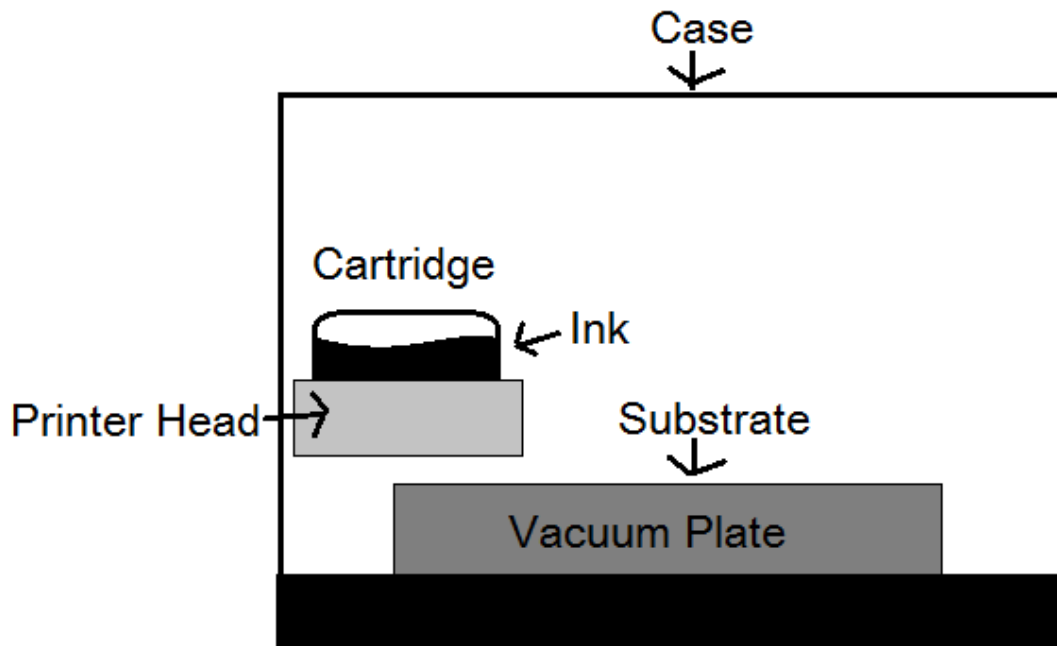


Figure 5.2: Diagram of inkjet printer used to fabricate the structure depicted in figures 5.3 and 5.4

5.2.2: Fabrication of Chosen Design

To begin with, a design that was predicted to display the desired sort of response was selected and a structure with that as the unit cell was created.

The structure depicted in figure 5.3 was chosen for this purpose. It was not a fully optimised example, but since the main purpose of this first round of fabrication was to assess the viability of the inkjet printing method for the fabrication of later structures, that was considered sufficient.

A number of settings and approaches were implemented when creating the larger structures. Simply going down in rows of unit cells was the most straightforward method, but this meant that if something went wrong then the resulting structure was awkwardly shaped and unsalvageable for analysis. A second approach instead worked outwards, making first a 2 x 2 unit cell structure, then expanding it to a 3 x 3 structure, and then (it was intended) to 4 x 4 and so on. Unfortunately issues with the process meant that the largest structure created was a 3 x 3 unit cell.

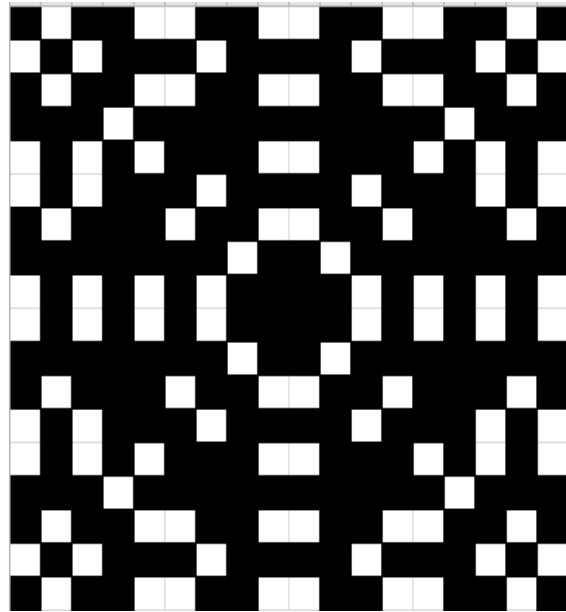


Figure 5.3: Design of structure that was fabricated from silver ink using inkjet printing.

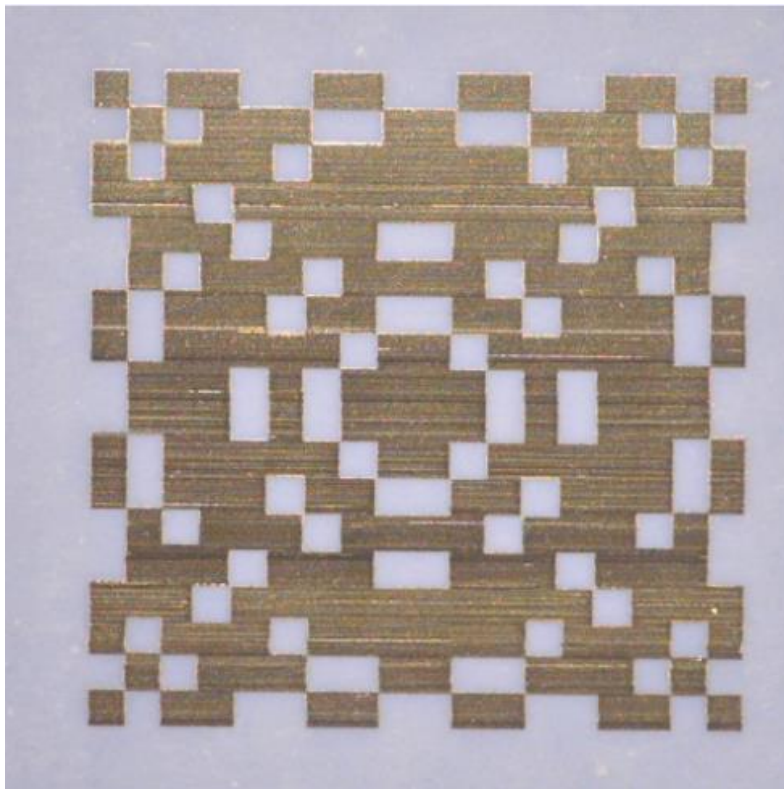


Figure 5.4: Example of structure that was fabricated from silver ink using inkjet printing.

Despite above the structure produced lacking the ideal dimensions for analysis, an attempt was nonetheless made to measure its S11 parameters using a horn antenna, and this produced the parameters depicted in Figure 5.6. The results are not particularly close to those expected, as shown in figure 5.7. The issues encountered whilst attempting the inkjet printing process are likely to blame for this, and so whilst it could theoretically produce structures of the sort desired, if it were performed successfully, this would likely have taken multiple attempts and have been overly expensive.

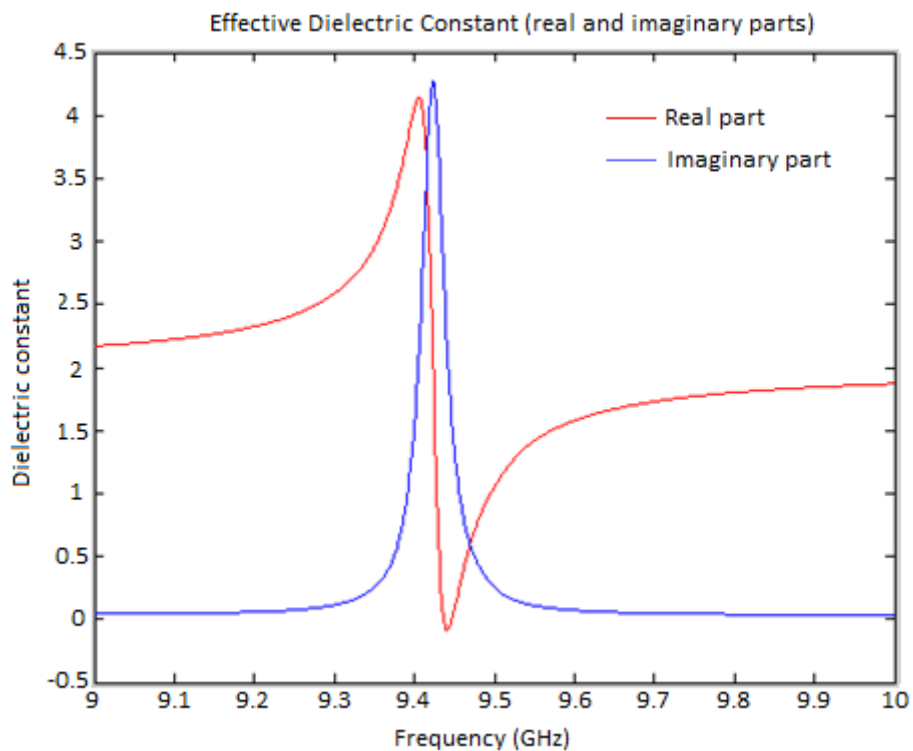


Figure 5.6: Results as predicted by simulation

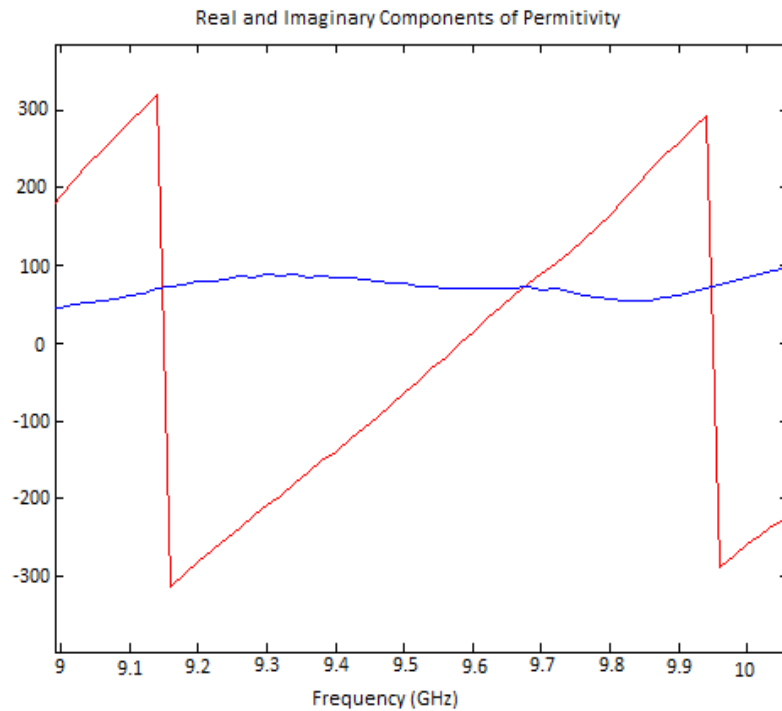


Figure 5.7: Results as tested at short range.

5.3: Milled Copper on FR4

5.3.1: Fabrication and Analysis

Since the inkjet printing method proved to be both problematic and expensive, other avenues of fabrication were also investigated. One such alternative that proved more viable was milled copper on FR4 substrate.

Rather than depositing the material, the surface came pre-covered and the design was created by removing material until only the desired shapes remained. This required some of the work to be done by hand, which meant that it took a fair bit longer than the inkjet printing method. The payoff was that it was also far more reliable, allowing for the creation of a far larger structure suitable for long range analysis in an anechoic chamber.

Since the materials used for this design were different to the one fabricated through the previous method, that pattern would not have generated the same material response. For this reason a new, more optimised structure was designed via the Genetic Algorithm program to replace it. In this case, it predicted an optimal performance for a design such as that in Figure 5.8.

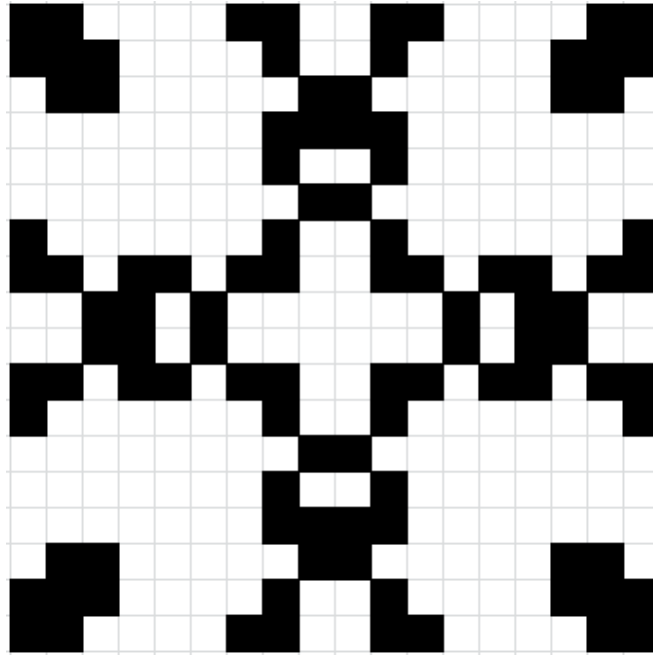


Figure 5.8: Design of structure that was fabricated from copper on fr4 substrate

A 10 x 10 grid with this unit cell was constructed. A photograph of the structure is depicted in figure 5.10, below. Like the previous example, this structure's reflection parameters were measured using a horn antenna, both at short range and at a distance of about 1 metre in an anechoic chamber. These results were compared to those predicted by the simulation (shown in figure 5.11). While the short range measurement (figure 5.12) displayed a strong resemblance to the simulated values, the measurement from the chamber (figures 5.13 and 5.14) displayed a far greater imaginary component, which entirely eclipsed the real values in magnitude.

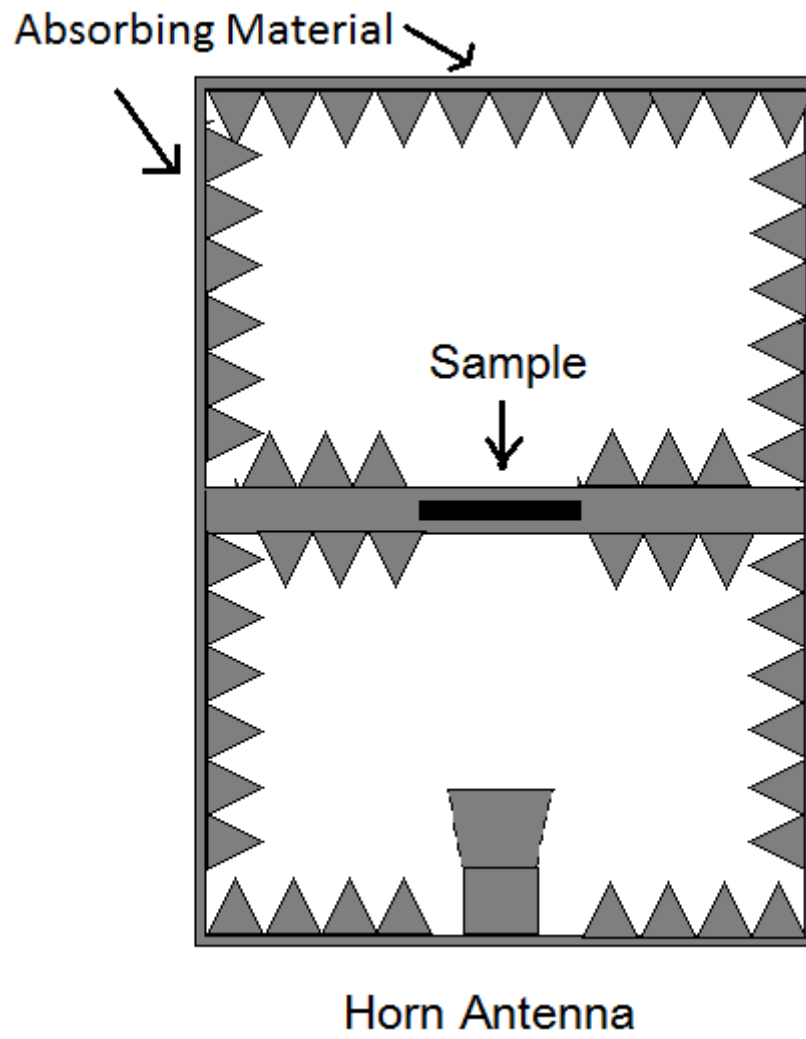


Fig 5.9: Diagram of the anechoic chamber setup.

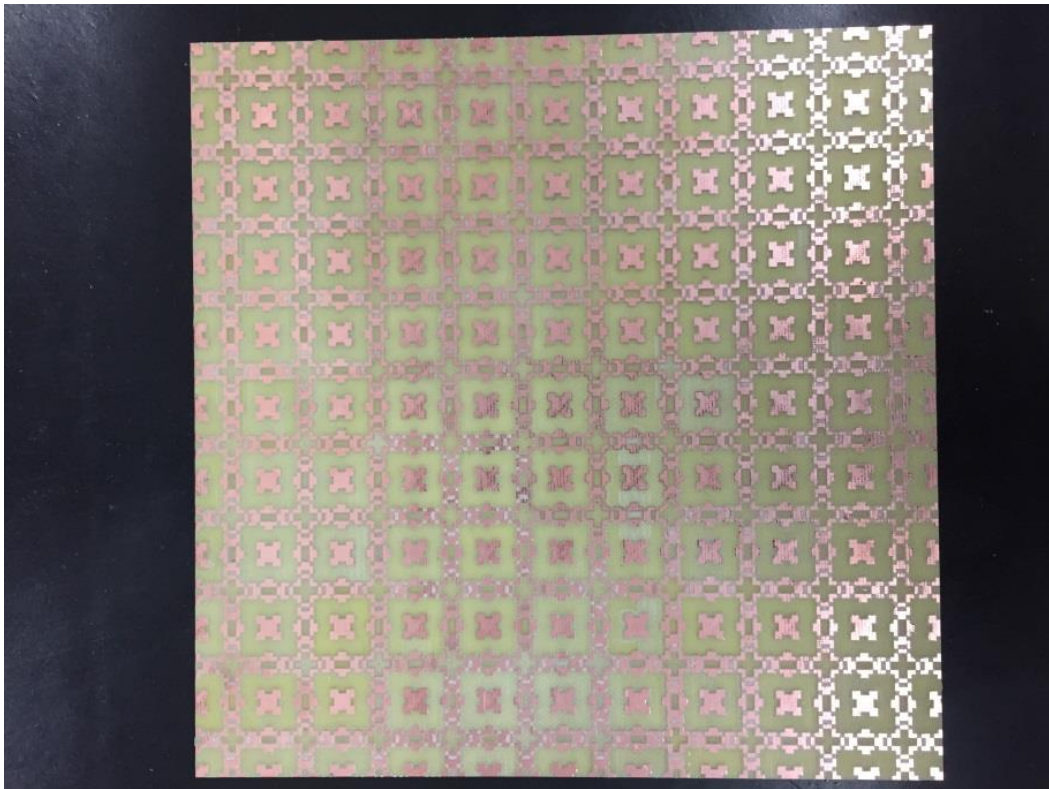


Figure 5.10: 10 x 10 unit cell structure that was fabricated from copper on FR4 substrate

The anechoic chamber [5] used for this analysis consisted of a large wooden cabinet with absorbing materials attached to the inner surfaces. Halfway up, approximately one meter from a horn antenna set in the base, there is a shelf on which to place a sample. The sample shelf has an aperture in the centre, over which the sample is placed in order to expose it to the signal transmitted from the horn antenna. This particular feature limits the sample size under the test.

The computer simulation of this structure predicted a strong resonance at around 9.7 GHz, with a very small imaginary peak at the same point. The numerical values of these points are somewhat higher than one would expect, but it is believed that this is largely due to a slight error in single variable in the S11 to permittivity conversion equation, which was later corrected to a more appropriate value for subsequent simulations. This variable only affects the scaling of the values, so it should not have any effect on the qualitative features of the prediction.

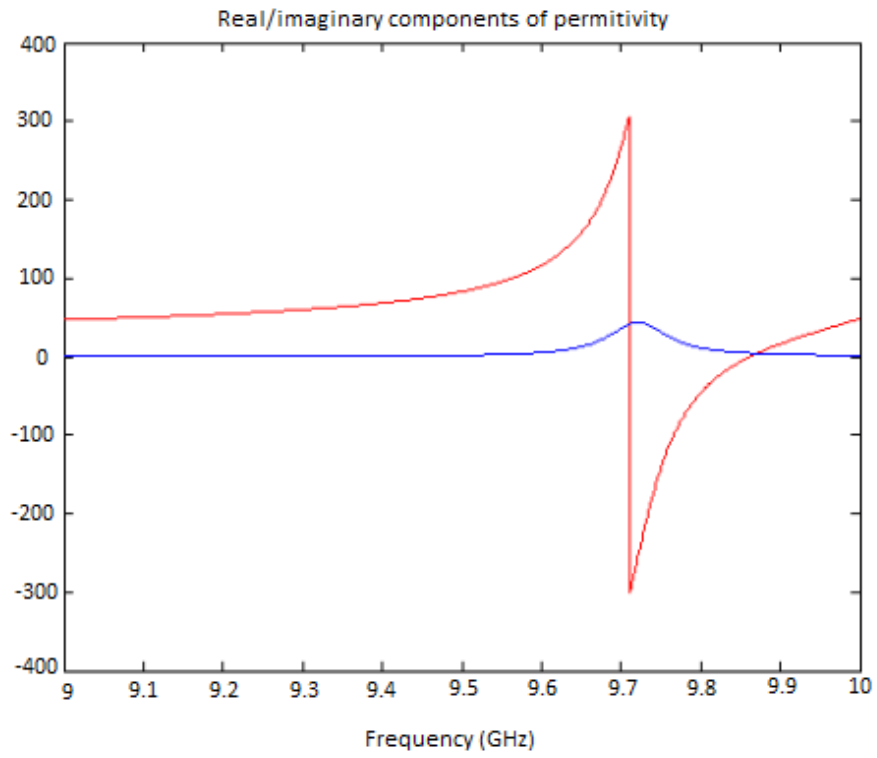


Figure 5.11: Results as predicted by simulation

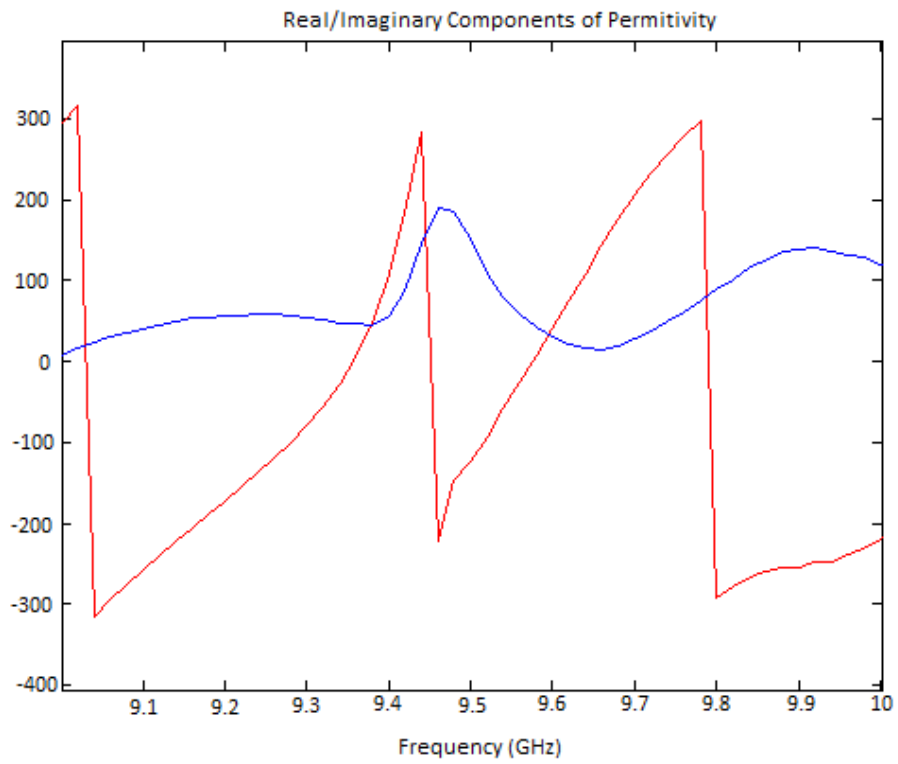


Figure 5.12: Results as measured at short range

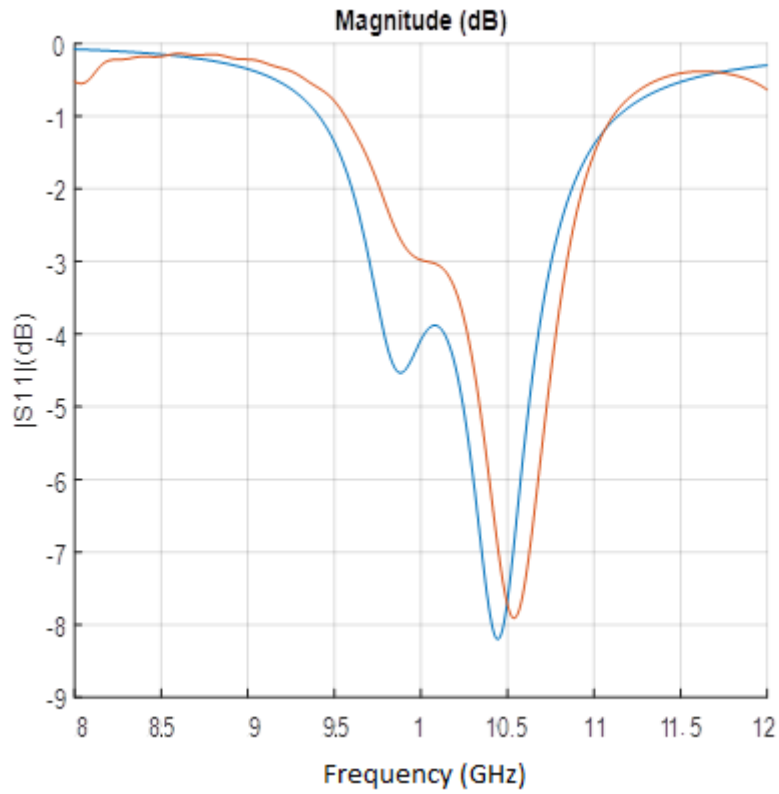


Figure 5.13: Comparison of measured (red) and simulated (blue) S11 Results after processing via the time gating method. [6], [7]

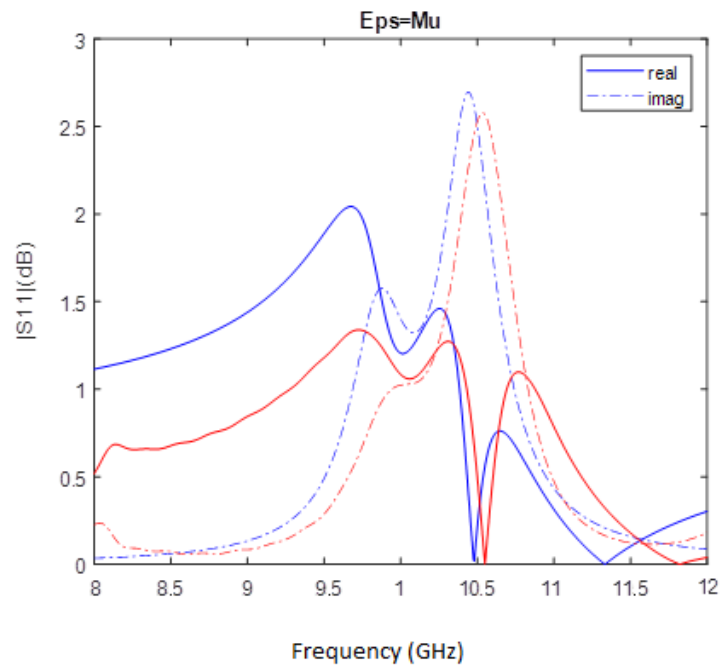


Figure 5.14: Comparison of measured (red) and simulated (blue) effective permeability after processing via the time gating method. [6], [7]

The close-range analysis of the fabricated structure is similar in appearance to the simulated responses predicted, and far closer to them than the measured response of the inkjet-printed structure was. The resonance is smaller than predicted (it was calculated from the measured s11 parameters by the same equation as was used for the simulation, so ratios will be preserved) and the major resonance has shifted down about 0.2 GHz, but overall the agreement appears to be quite high.

Simulation and fabrication of structures is ongoing. The later modifications to the GA and subsequent simulations will provide new, improved designs to fabricate and these new fabricated structures should better adhere to the goals of this project than those already created.

5.3.2: Further Simulation of Copper Metaferrite Design

The etched copper structure was subjected to further study later on in the project. For this, the CST model of the structure was subjected to further tests by a colleague, Mr Yujie Liu.

First, the metaferrite structure's performance when placed behind a bow tie antenna [8] was compared with that of the same antenna placed in front of a vacuum.

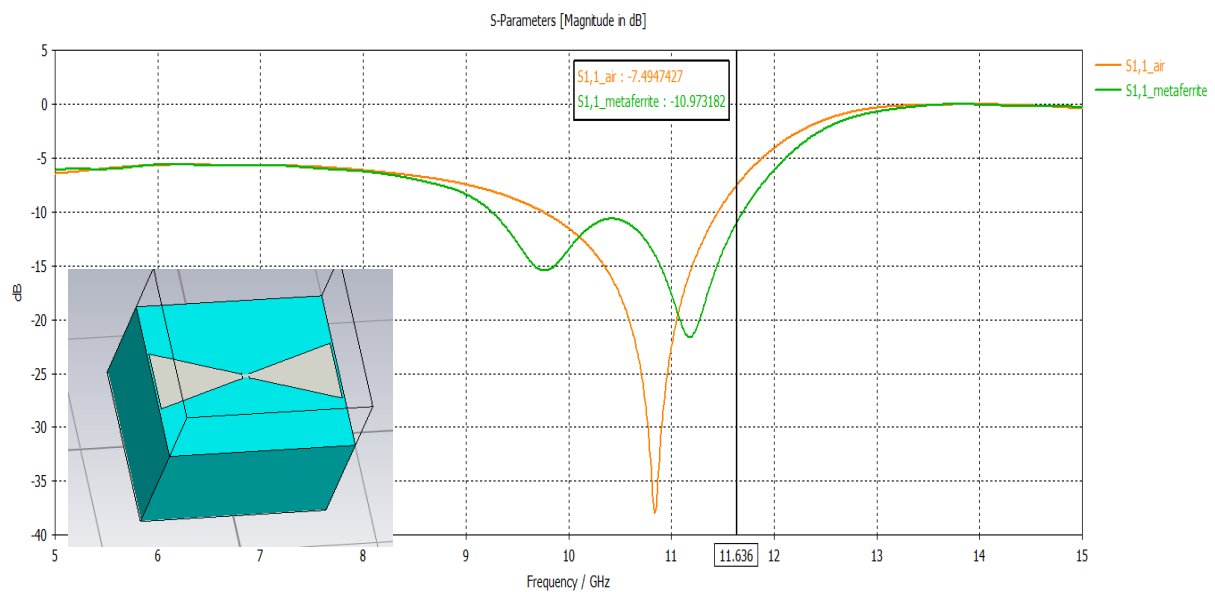


Figure 5.15: Simulated results of antenna in front of vacuum in air (orange) and effective medium with same parameters as the metaferrite (green) with 13mm periodic [7]

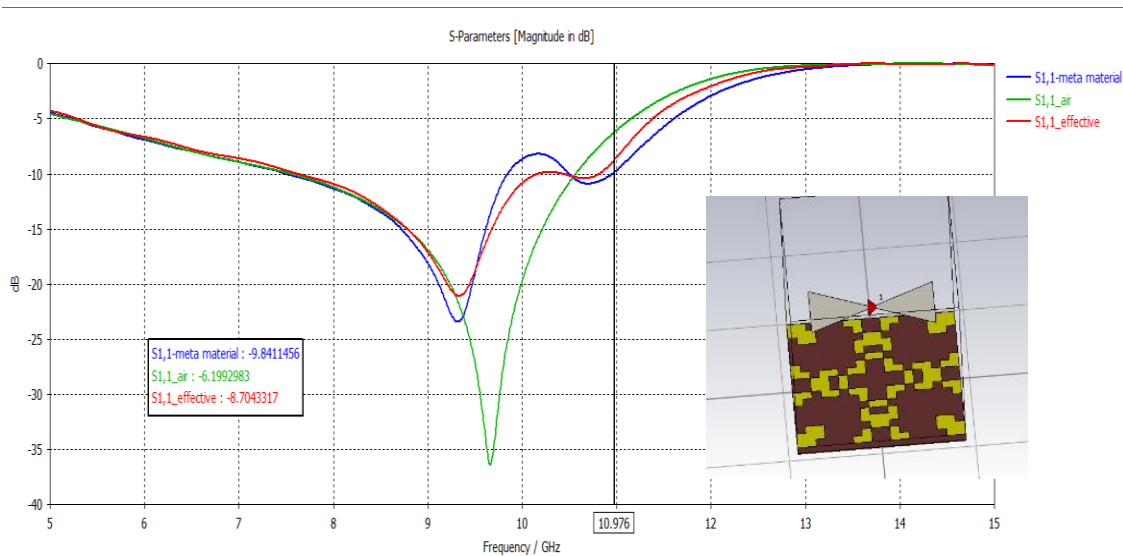


Figure 5.16: Simulated results of antenna in air (green), effective medium (red) and the actual metaferrite (blue) with 18mm periodic [7]

The effective medium [9] shown in figure 5.15 has the same permeability and permittivity as the metaferrite structure, and was found to increase the bandwidth from 1.5 GHz to 2.5 GHz, compared to vacuum. In figure 5.16 the actual metaferrite structure was compared to the same effective medium and air, and was found to increase bandwidth from 2.7 GHz to 3.17 GHz. This means that a structure of this type could potentially see use in antenna applications.

In addition to this, the simulated Radar Cross-Section [10] of the structure was calculated. It was predicted that the structure would display reduced RCS when compared with that of a metal surface. This could also be of use in some applications, but further study is probably required, including refinements.

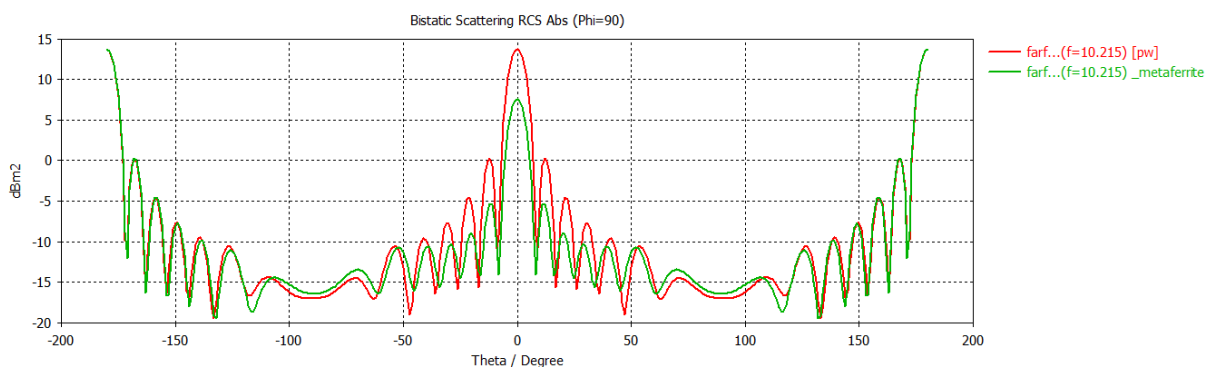


Figure 5.17: Predicted RCS response of the structure, obtained via simulation. [5]

5.3.3: U-Slot Patch Antenna Fabrication

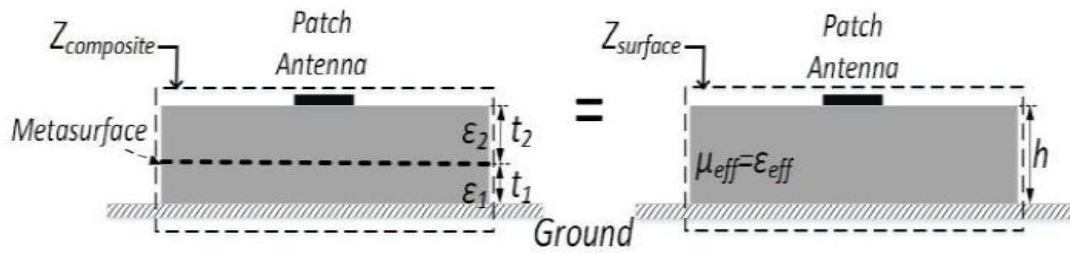


Figure 5.18: The layered design of the metaferrite antenna compared with the theoretical effective medium model.

In addition to the bowtie antenna, a second antenna was simulated, this time a U-slot patch antenna. This antenna was also fabricated, and practical tests were performed to validate the results of the simulation. The copper structure in figure 5.10 formed the base plate of this structure, and a second layer of substrate was placed on top, as shown in figure 5.18. On top of this was a metal patch antenna with a U-shaped slot cut out of it, as shown in figure 5.19 and 5.20

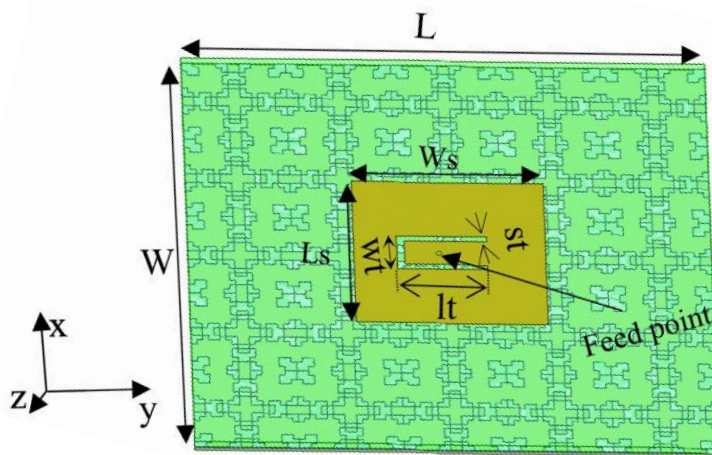


Figure 5.19: Diagram of U-slot patch antenna over metaferrite base.

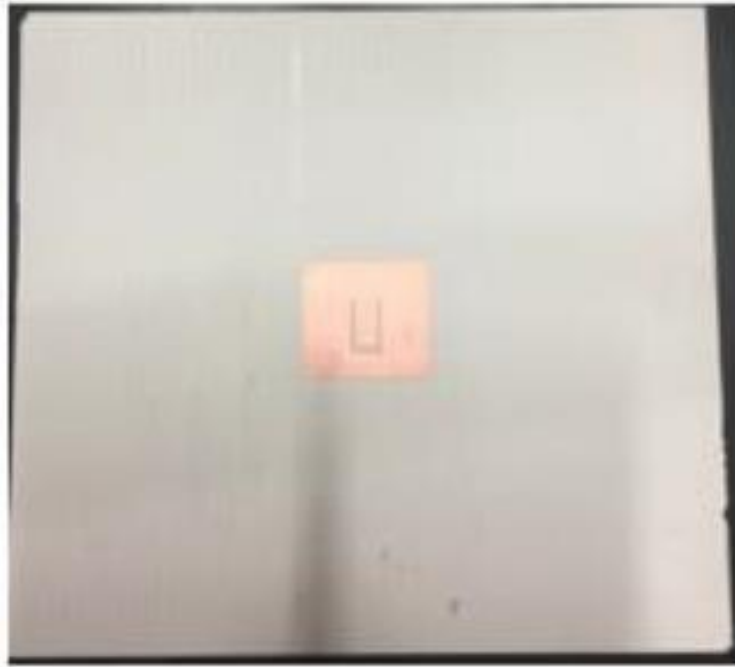


Figure 5.20: Image of fabricated U-slot patch antenna over metaferrite base.

The antenna's response was measured and compared to that predicted by simulation. The S_{11} parameters showed good agreement, taking into account a small frequency shift caused by an air gap in the fabricated structure. Both indicated that the desired value of $\epsilon_r = \mu_r = 3 - i$ for RCS reduction was achieved at around 9 GHz, as shown in figure 5.21.

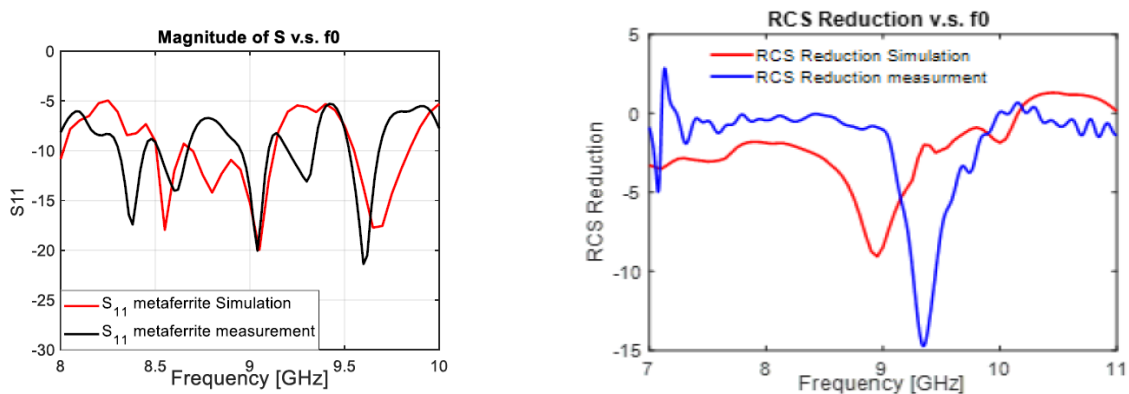


Figure 5.21: Comparison of simulated and measured RCS reduction values for U-slot patch antenna

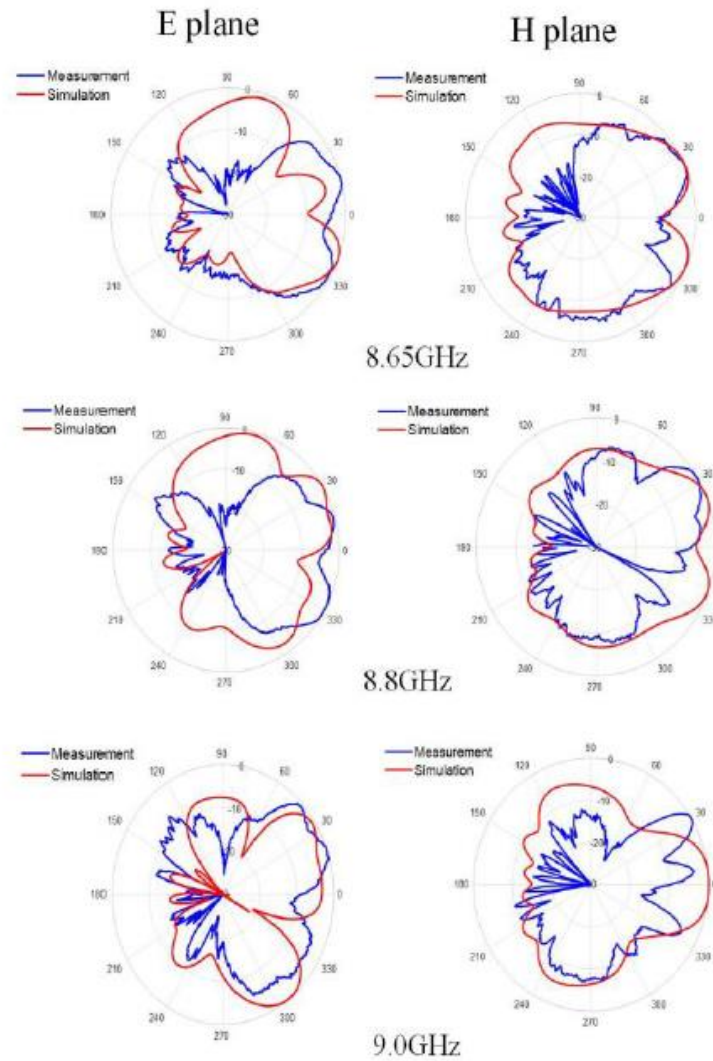


Figure 5.22: Comparison of measurement and simulation of the normalised E- and H-plane radiation patterns at 8.65, 8.8 and 9 GHz

The farfield response of this antenna was also measured and compared to simulation at three distinct frequencies, as shown in figure 5.22.

5.4: Summary

Several structures designed by the genetic algorithm program were fabricated using a variety of techniques, generally depending on the parameters on which the simulation was based. This was done both to validate the accuracy of the simulations and to assess the viability of the fabrication techniques used.

The primary techniques studied were inkjet printing using silver ink on a PET substrate, and etching using copper foil on an FR4 substrate. Of these two the latter was found to be the preferable technique, owing to its greater robustness and reliability, and lesser cost. Silver is, after all, expensive, and performing several trial runs using an inkjet printer in order to establish the proper settings required to create a full structure proved to quickly use up valuable material.

The results obtained were approximately those expected, but there is certainly room for improvement, both in conformity to the results predicted and in the conformity of the results predicted to the goals. It is hoped that the results presented here might form a basis for improvements to the genetic algorithm and manufacturing methods that will facilitate both.

Following this, simulations and experiments were performed to explore the performance of the copper structure when placed behind both a bow tie antenna and a u-slot patch antenna. These were found to aid in RCS reduction.

The current structures have proved to be able to enhance the bandwidth of an antenna if placed behind one, but overall they function better as an absorber than a reflector. It is hoped that new designs taking the new parameters into account will better conform to the goal. Other designs involving multi-layer structures and transmitters are also being studied.

5.5: References

- [1] S. Fresner, "Basic Screen Printing Techniques 101 by Scott Fresener", <http://t-biznetwork.com/articles/screenprinting/basic-screen-printing-techniques-101/>, March 2016, accessed June 2017
- [2] C. Woodford, "Inkjet printers." <https://www.explainthatstuff.com/inkjetprinters.html>, 2006, accessed June 2017
- [3] O. Çakir, A. Yardimeden, T. Özben, "Chemical machining" *Archives of Materials Science and Engineering*. 28 (8): 499–502, August 2007
- [4] A. Sridhar, T. Blaudeck, R. R. Baumann "Inkjet Printing as a Key Enabling Technology for Printed Electronics" *Material Matters Volume 6 Article 1*, 2011
- [5] R. H. Randall, *An Introduction to Acoustics*, Dover Publications, 2005
- [6] B. Archambeault, S. Connor, J. C. Diepenbrock "Time Domain Gating of Frequency Domain S-parameter Data To Remove Connector End Effects for PCB and Cable Applications" *2006 IEEE International Symposium on Electromagnetic Compatibility*, August 2006
- [7] Y. Liu, 2018
- [8] B. Kaur, L. Singh Solanki, "A Brief Review on Bowtie Antenna", *Conference Paper*, December 2012
- [9] M Wang, N Pan, "Predictions of effective physical properties of complex multiphase materials" *Materials Science and Engineering: R: Reports*. 63:1, 2008
- [10] F. Ulaby, *Microwave Remote Sensing: Active and Passive, Volume 2*, Artech House, Inc. 1986

Chapter 6: Hyperuniform Metamaterials

6.1: Other Metaferrite Structures: Random vs Periodic vs Hyperuniform

To date, almost all forms of metamaterials and metasurfaces including metaferrites are periodic structures with repeating patterns of unit cells [1], [2]. These unique topologies offer peculiar electromagnetic properties as they introduce constructive and/or destructive interference at certain directions due to Bragg diffractions [3]. They can be used to manipulate angular properties and polarisations of transmitted and reflected waves. This is not the only way such a goal may be realised, however. Randomly dispersed structures occur in the formation of conventional dielectrics where their inclusions are also irregular [4], and so, some questions are raised over the role of periodicity and randomness of point distributions of meta-atoms within metasurfaces.

Recently, increasing interests have been found in the study of hyperuniform randomness. Hyperuniform Structures work differently to metaferrites exploiting high impedance effects, instead exploiting the “ordered randomness” they possess to scatter incoming waves. This “ordered randomness” means that hyperuniform structures possess traits of both periodic and random structures [5], [6]. This means that they are characterised by a reduction in density fluctuations compared to random structures, but also lack the Bragg peaks seen in periodic structures. This causes them to resemble a hybrid of crystal and isotropic liquid. The way they work differs from high-impedance surface based metaferrites as well, scattering rather than absorbing radiation. This means that their performance is orientation dependent, being most effective at particular angles of incidence whilst not reducing the global reflection when all angles are considered. Hyperuniform structures also prevent situations that can arise in a truly random arrangement such as low-entropy states like all the particles in a room ending up in one corner, which are theoretically possible, if unlikely. Such a state is not generally what is being sought by those who elect to generate a random distribution, however. More commonly, what someone employing a “random” distribution desires is a reasonably even distribution that is not bound to a rigid periodic lattice [7]. A significant number of papers have been published in exploring novel structures such as photonic crystals with a wide bandgap. [8], [9], and demonstrating the technology’s usefulness in the field of optical waveguides and High-Q polarizers, on chip spectrometers and Luneberg lenses, to name a few.

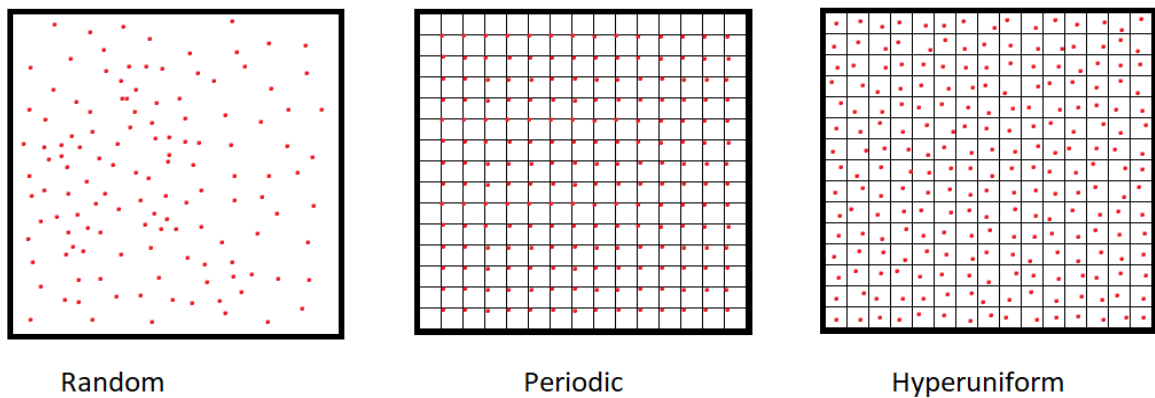


Figure 6.1: Illustration of the difference between randomised, hyperuniform and periodic particle distribution.

Hyperuniform structures can be designed by allowing each particle to occupy a random position within a segment of a grid. In terms of Fourier space, the structure factor $S(k)$ approaches zero as $|k|$ approaches zero. This means that there will be a reasonably even distribution of particles, free of any clumping that a truly random arrangement could produce. Indeed, hyperuniform structures can often appear more consistently “random” than a truly random distribution.

In order to evaluate the differences between hyperuniform and high impedance metaferitesm, several metasurfaces based on the former were designed with the goal of minimising the back-scattering from an incident plane wave reflected from the surface. This took the form of multiple dipole structures consisting of two metallic arms of 8-10 mm in length at an angle of between 0 and 90 degrees from each other. The values were randomly determined for each antenna. These structures were designed in CST using a VBA macro to quickly translate hyperuniform distributions into simulated structures ready for analysis. The analysis itself was performed by a colleague, who also provided the distributions to input. Following this, further testing aimed at reducing the radar cross section of the structures was also performed. Periodic, random and hyperuniform structures were compared for these purposes. The hypothesis was that hyperuniform structures would perform better than either of the alternatives, and comparing them in this manner is obviously the most straightforward method of either confirming or disproving that notion. The structure of the unit cell is shown in figure 6.2, along with the various parameters that could be varied within a structure. [10]

The structures consisted of a number of simple v-shaped dipole antennas of various, randomly determined, tine lengths and opening angles. The purpose of this was to alter the refractive index of the material, since structures of this nature can act as a plasmonic metasurface [11]. Snell's law of refraction states that the ratio of the sines of the angles of incidence and refraction of light entering a medium is equal to the ratio of the phase velocities of that medium and the medium that the light entered from, which is also equal to the inverse ratio of the two media's refractive indices [12], [13]. Or in other words:

$$\frac{\sin \theta_2}{\sin \theta_1} = \frac{v_2}{v_1} = \frac{n_1}{n_2} \quad (6.1)$$

In the case of an array of v-shaped antenna the plasmonic interfaces impart abrupt phase shifts in the optical paths, which allows for more flexibility in how one moulds the optical wavefront [14]. This allows Snell's law to be generalised to the form

$$[k_0 n_i \sin(\theta_i) dx + (\Phi + d\Phi)] - [k_0 n_t \sin(\theta_t) dx + \Phi] = 0 \quad (6.2)$$

which can then be rearranged to give

$$\sin(\theta_t) n_t - \sin(\theta_i) n_i = \frac{\lambda_0}{2\pi} \frac{d\Phi}{dx} \quad (6.3)$$

Here, λ_0 is the vacuum wavelength, θ_i and θ_t are the angles of incidence and refraction respectively, and n_i and n_t are similarly the refractive indices of the materials in which the incident and refracted radiation propagate. k_0 is $\frac{2\pi}{\lambda_0}$, and Φ and $\Phi + d\Phi$ are the phase discontinuities where the paths cross the interface.

Figure 6.2 depicts the design and parameters of the v-antenna structures. They were placed on a substrate which had a thickness of 2mm and a dielectric constant of 2.65. The metal v-antenna itself had a thickness of 0.018mm, as did the ground plate behind the structure. The structures were arranged in either a periodic, random or hyperuniform arrangement and farfield simulations of each were made. In order to understand the scattering field of the structure as a whole, it's useful to understand that of a single lattice. The theoretical formula for the E-field of a dipole antenna is given by

$$E(\theta) = \frac{\cos\left(\frac{kl}{2} * \sin \theta * \sin \varphi\right) - \cos\left(\frac{kl}{2}\right)}{\sqrt{1 - \sin^2 \varphi \sin^2 \theta}} \quad (6.4)$$

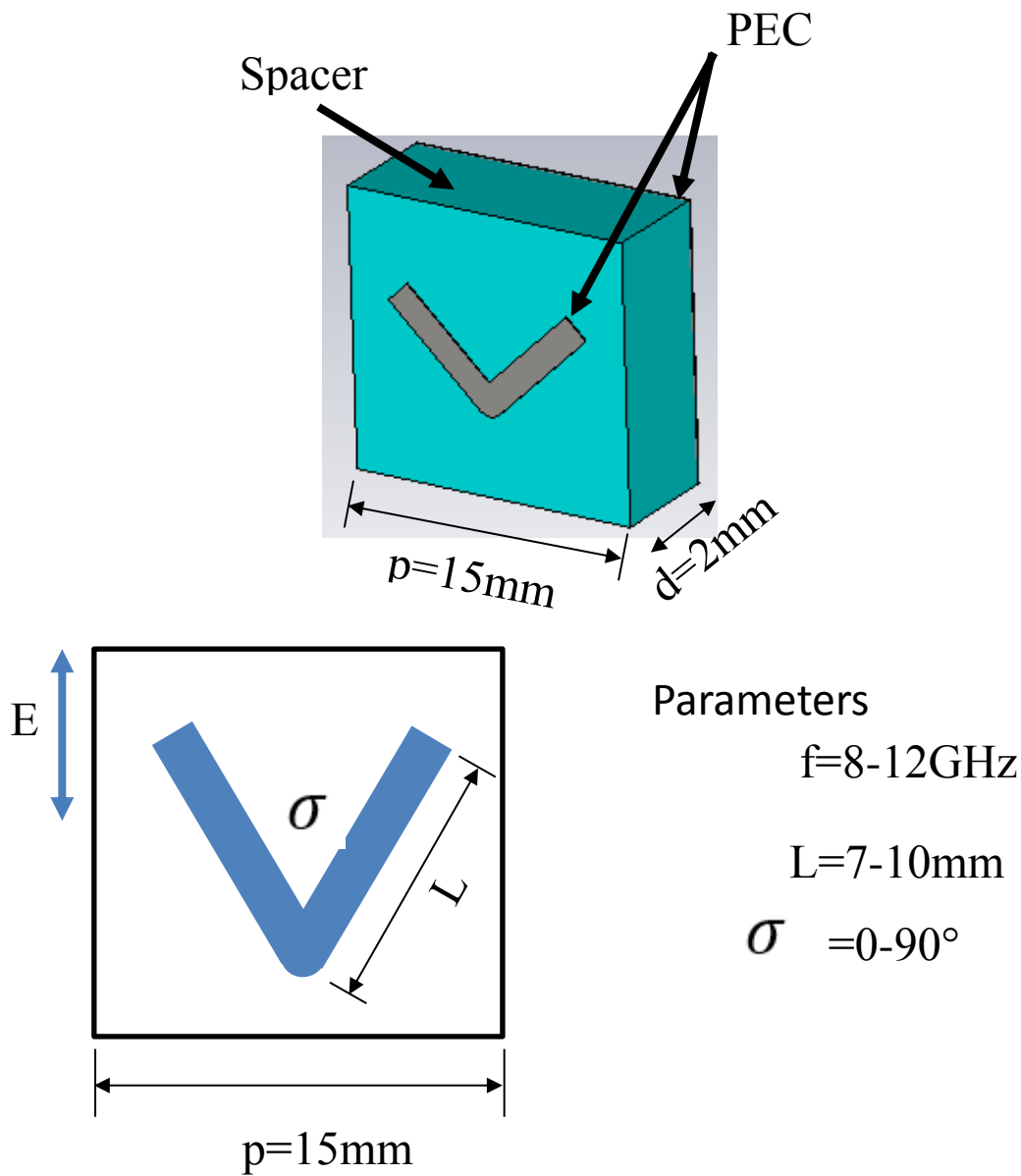


Figure 6.2: Parameters of dipole unit cells used in metaferriite structures described below

In the case where there is no coupling between dipoles and each of them rotates by half the opening angle σ (28) can become

$$E(\theta) = \frac{\cos\left(\frac{kl}{2} * \sin \theta * \sin(\varphi \pm \sigma/2)\right) - \cos\left(\frac{kl}{2}\right)}{\sqrt{1 - \sin^2(\varphi \pm \sigma/2) * \sin^2 \theta}} \quad (6.5)$$

The S-parameters of this structure were taken at a range of incidence values and for various parameters of the length and opening angle. These are detailed below in figure 6.3:

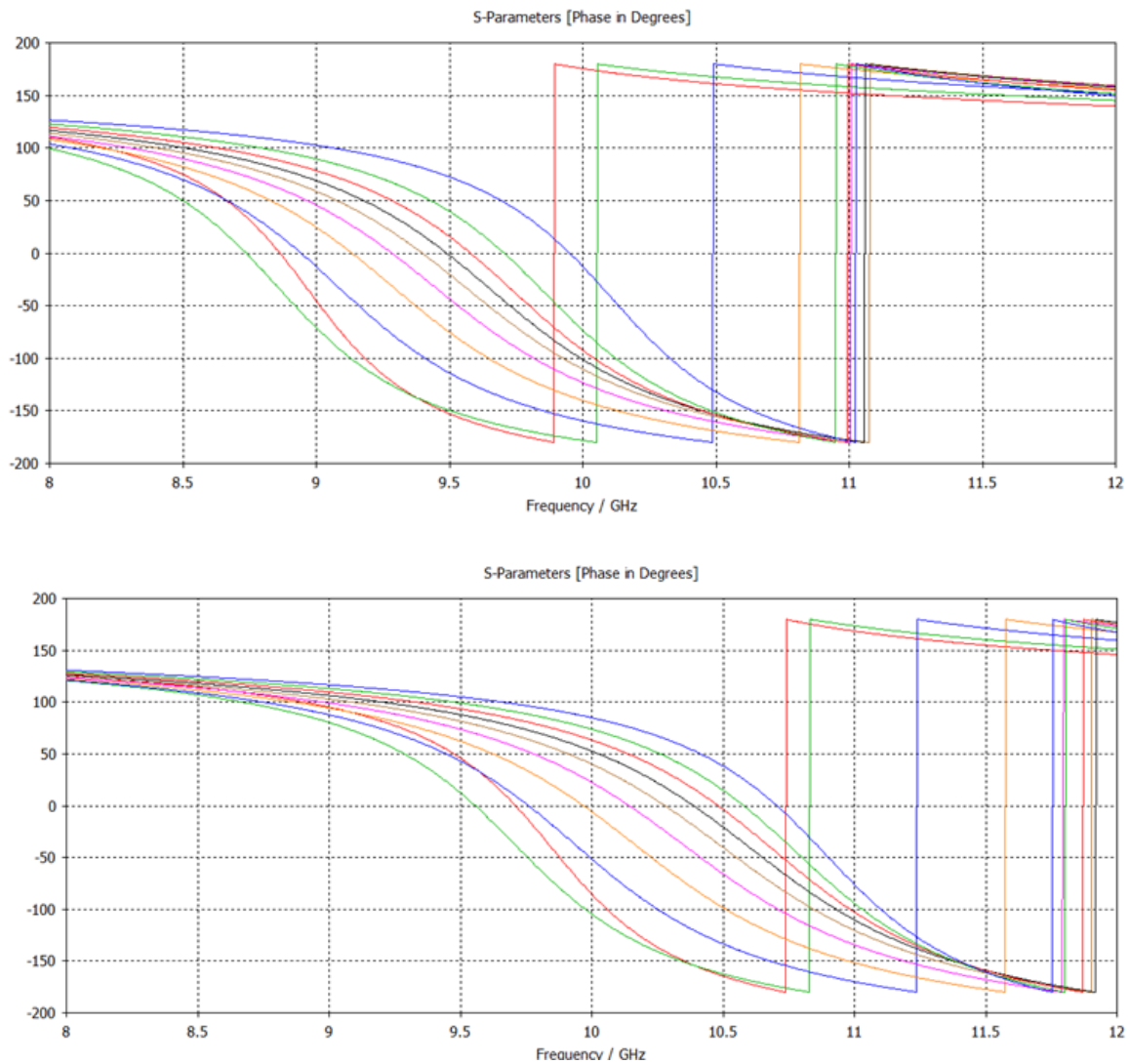


Figure 6.3: S- Parameters response of dipole unit cells used in metaferriite structures taken at a range of variables and angles of incidence

As can be seen, the individual unit cells display a range of possible phase shift frequencies, ranging from just under 10 GHz at the lower end to just under 12 GHz at the upper one. The greatest values are typically those that correspond to higher angles of incidence.

Following this analysis, periodic and hyperuniform structures were created. Each had the form of an $M \times N$ array of v-shaped antennas. The scattering pattern of such a structure can be characterised as a superposition of the scattered waves of its constituents, meaning that the normal incidence can be derived as

$$f(\theta, \varphi) = f_e(\theta, \varphi) \sum_{m=1}^M \sum_{n=1}^N \exp\{-i\} [\delta(n) + kD \sin \theta \times ((m - 1/2) \cos \varphi + (n - 1/2) \sin \varphi)] \quad (6.6)$$

In the case of a non-periodic distribution, this is modified to

$$f(\theta, \varphi) = f_e(\theta, \varphi) \sum_{n=1}^N \exp\{-i[\delta(n) + \sin \theta (u * \cos \varphi + v * \sin \varphi)]\} \quad (6.7)$$

$f(\theta, \varphi)$ is the vector far-field of the element at position [n], while θ and φ are respectively the polar and azimuthal angle. When the initial phase of the elements is applied between 0 and 360 degrees randomly the far-field scattering should undergo complex interference in an equally random manner and thus act similarly to light illuminating a rough surface, theoretically producing a low backward scattering.

Naturally, structures of the nature described here were constructed in CST and tested in order to verify this theoretical analysis. First to be studied was the periodic structure seen below, in figure 6.4. This displayed a phase shift at 9 GHz.

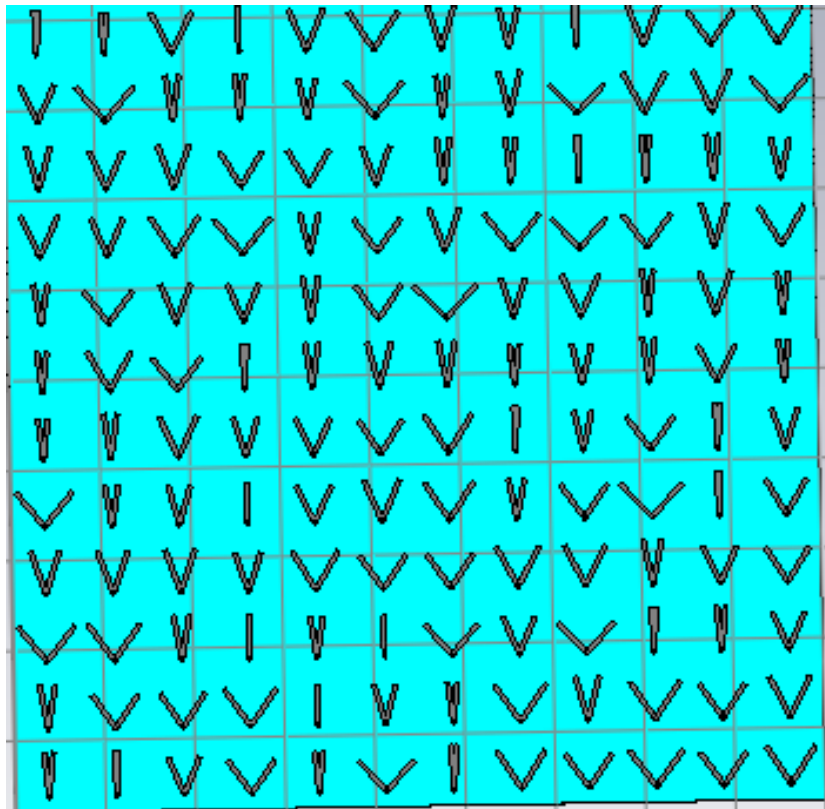


Figure 6.4: Periodic structure displaying a phase shift at 9 GHz

It also displayed the following farfield parameters, seen in figure 6.5:

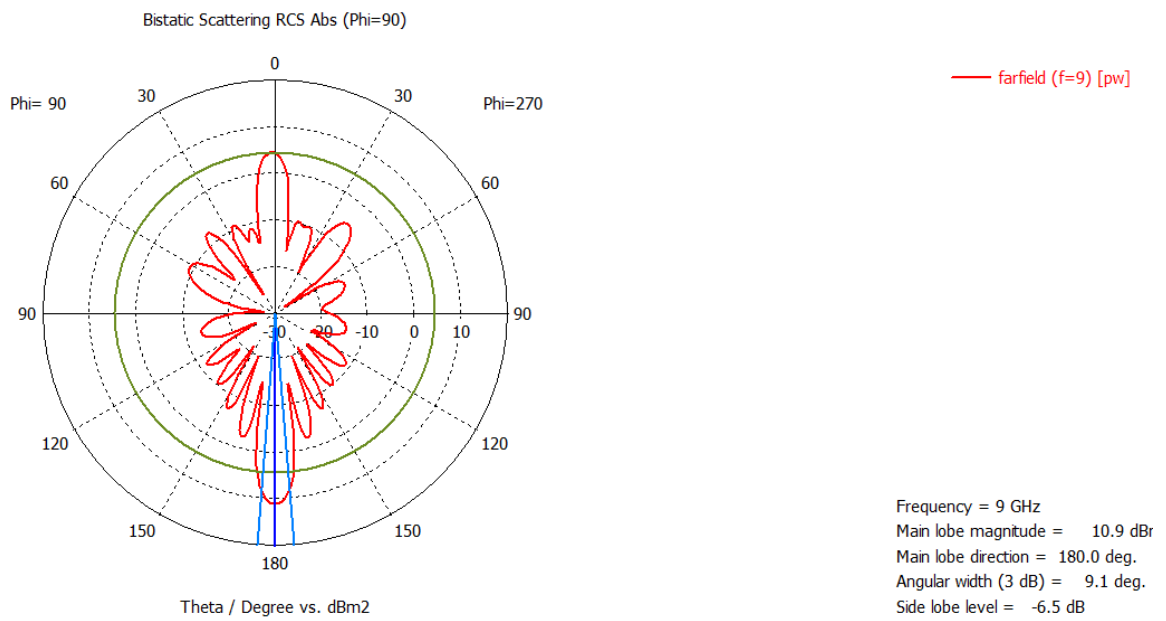
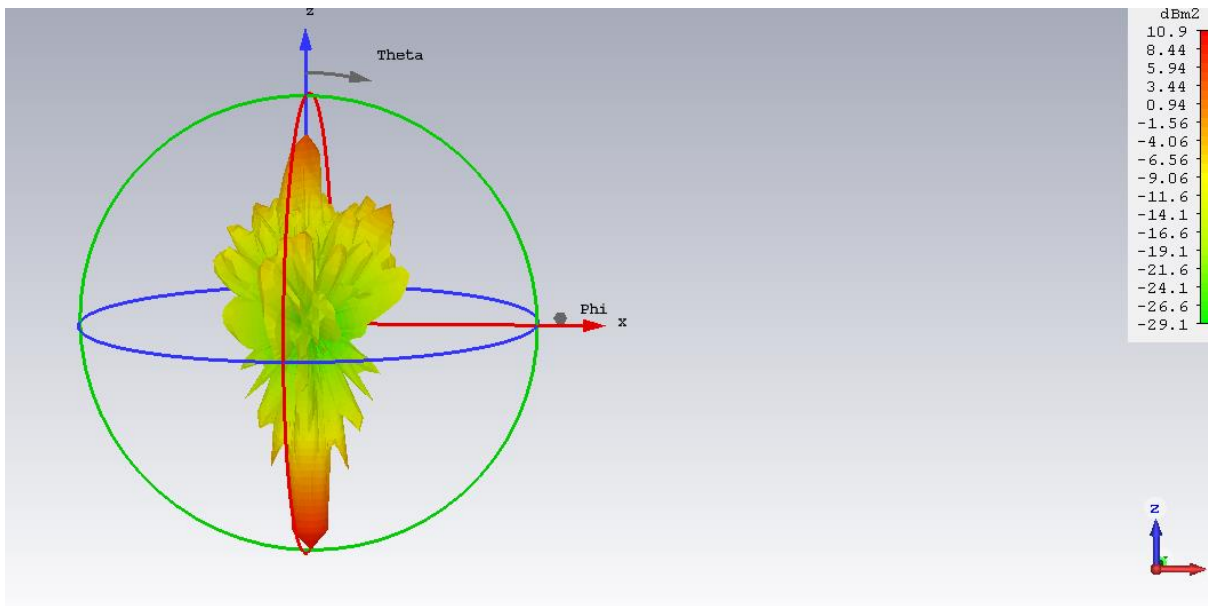


Figure 6.5: Farfield response as 9GHz periodic structure

By comparison, the hyperuniform structure is depicted in figure 6.6, below. This structure also displayed a phase shift at 9 GHz, making it ideal for comparison with the structure in figure 6.6.

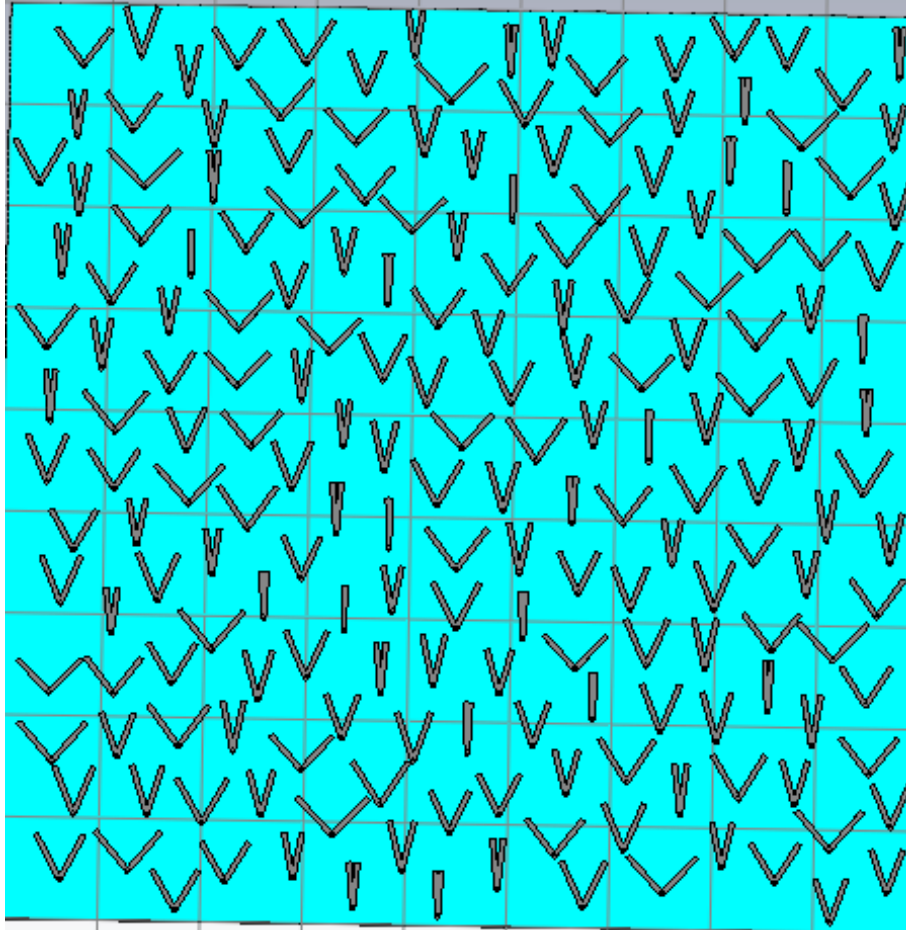


Figure 6.6: Hyperuniform structure displaying a phase shift at 9 GHz

This structure displayed the farfield parameters seen in figure 6.7:

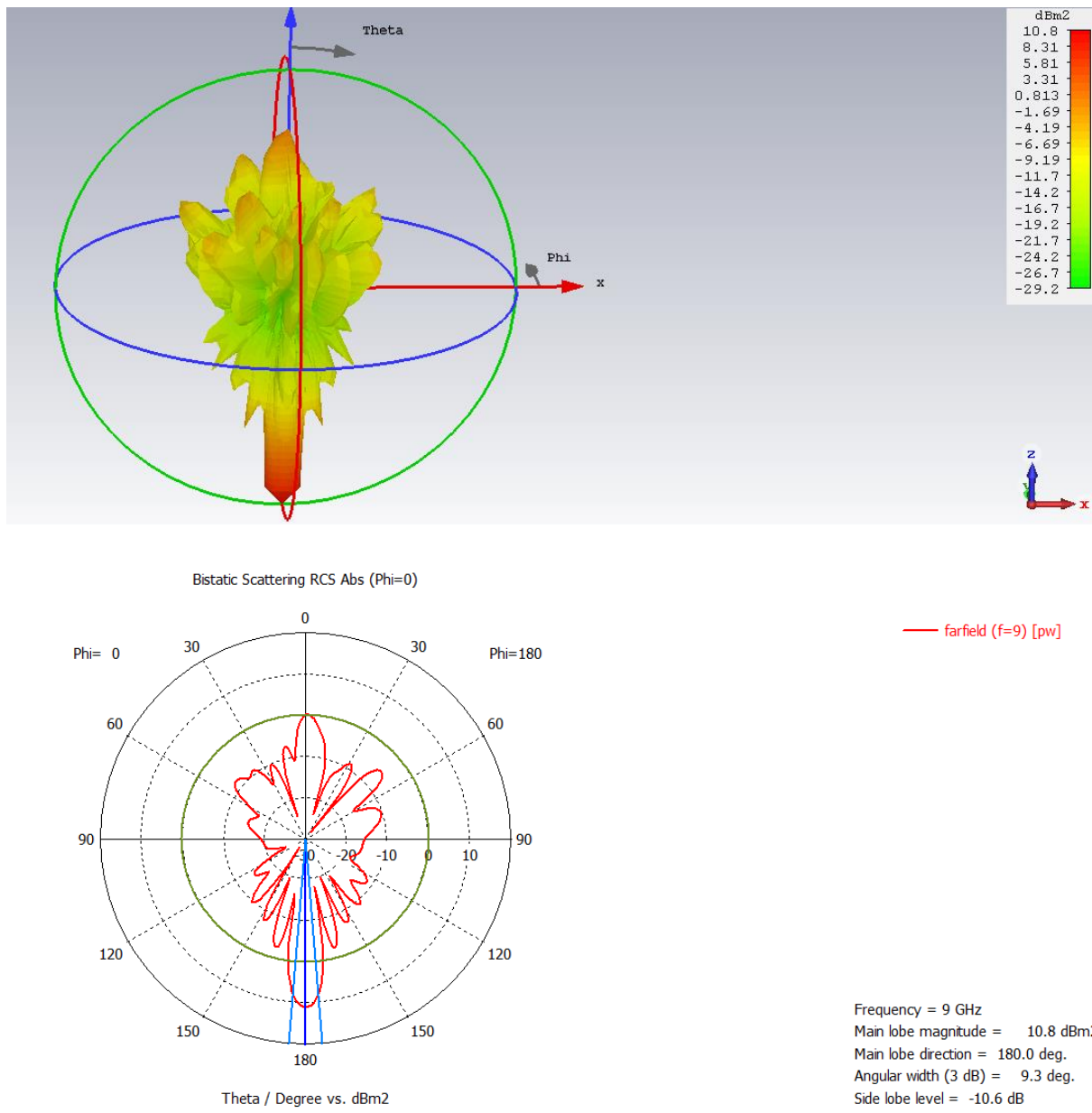


Figure 6.7: Farfield response as 9GHz hyperuniform structure

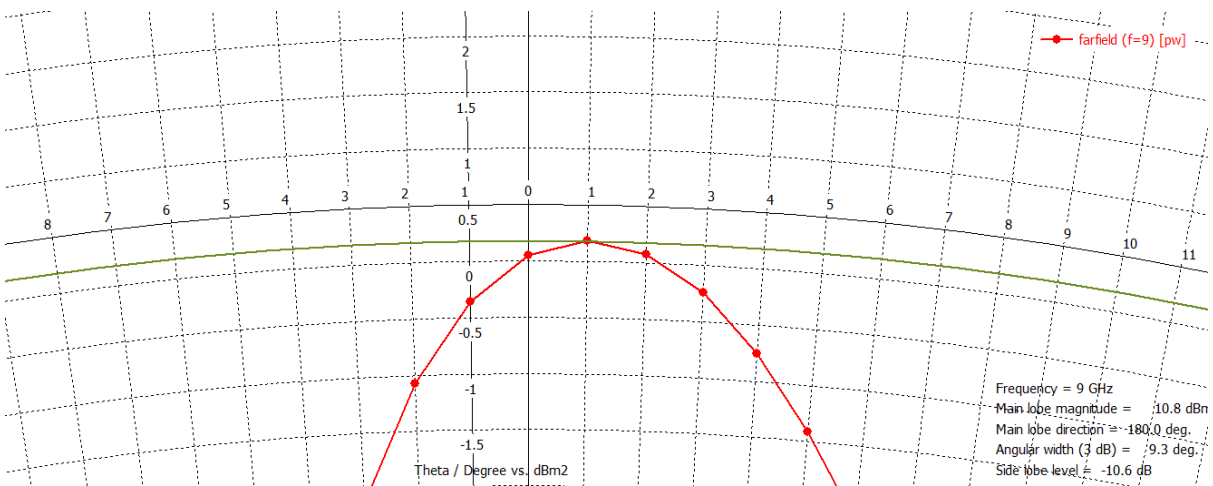
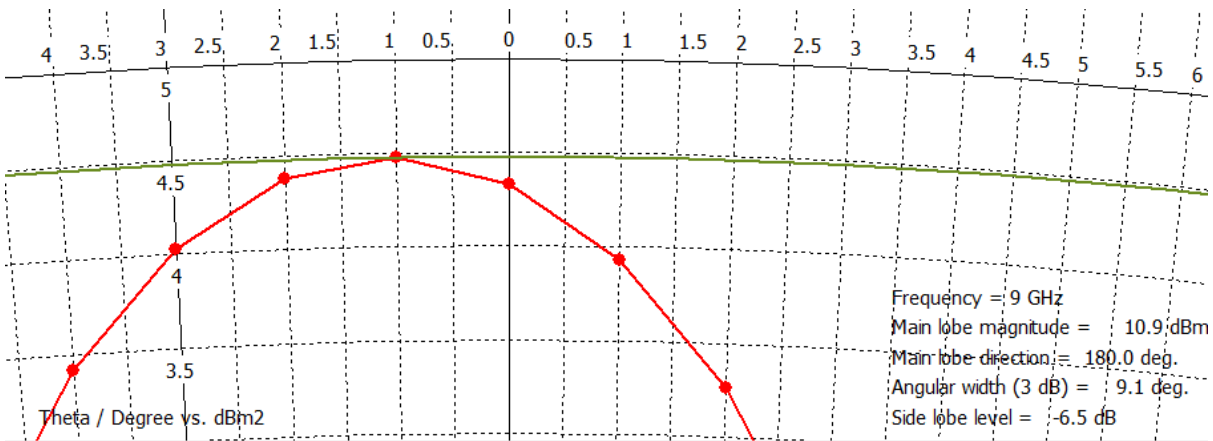
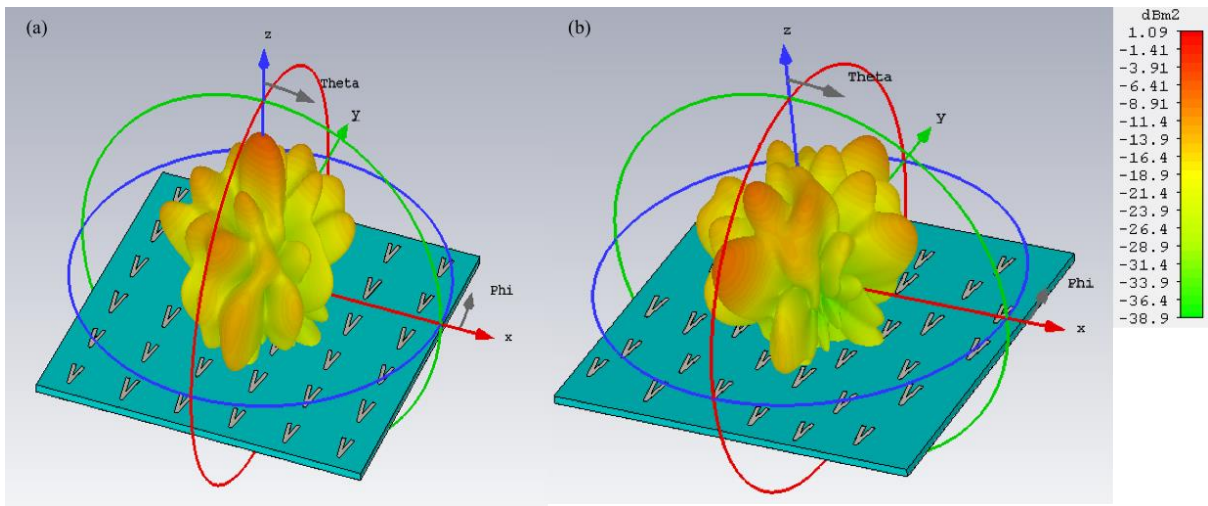


Figure 6.8: Comparison of the farfield response of periodic and hyperuniform structures

As can be seen here, the hyperuniform structure studied does indeed display a lower far-field response than the periodic structure in the 0 degree direction, whilst maintaining an

equally high magnitude at 180 degrees. The results indicate that the hyperuniform structure is better at reducing the back scattering from the plane wave, as predicted.

In the structures seen above, the periodic and hyperuniform variants both possessed the same number of particles. However, hyperuniform structures allow one to fit more particles in a given area, thereby increasing the bandwidth possible. This feature was explored with the use of 144, 197 and 208 dipole structures with periodic, hyperuniform and randomised distributions respectively.

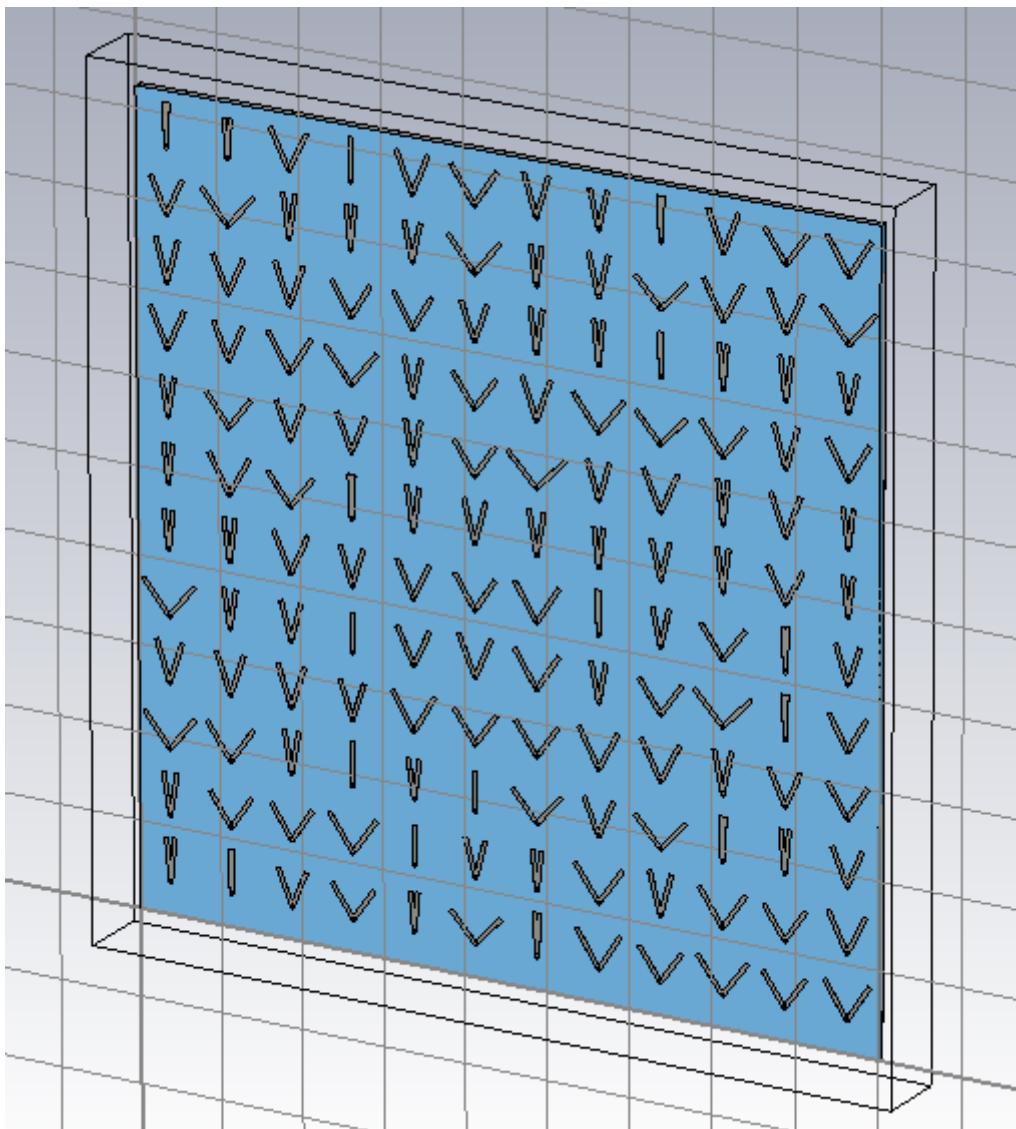


Figure 6.9: Periodic structure containing 144 dipoles.

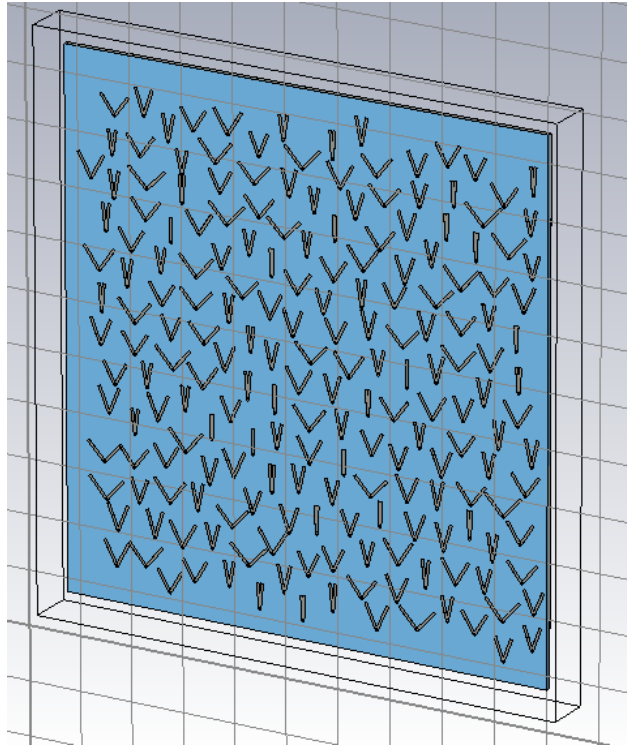


Figure 6.10: Hyperuniform structure containing 197 dipoles. Note the far less ordered arrangement.

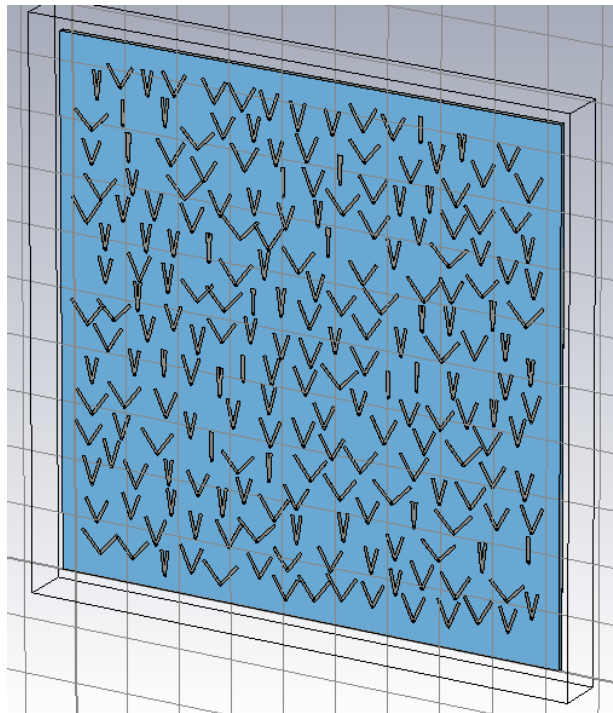


Figure 6.11: Random distribution structure containing 208

The simulated structures were subjected to more in depth farfield analysis, this time taking into account Bistatic Scattering, with the aim of determining which best achieved the goal of reduced far-field reflection. The typical response of the structures generally resembled that shown in figure 6.12, however the differences between the various metasurfaces are more clearly seen in tabular form. Such a table is depicted below in figure 6.13.

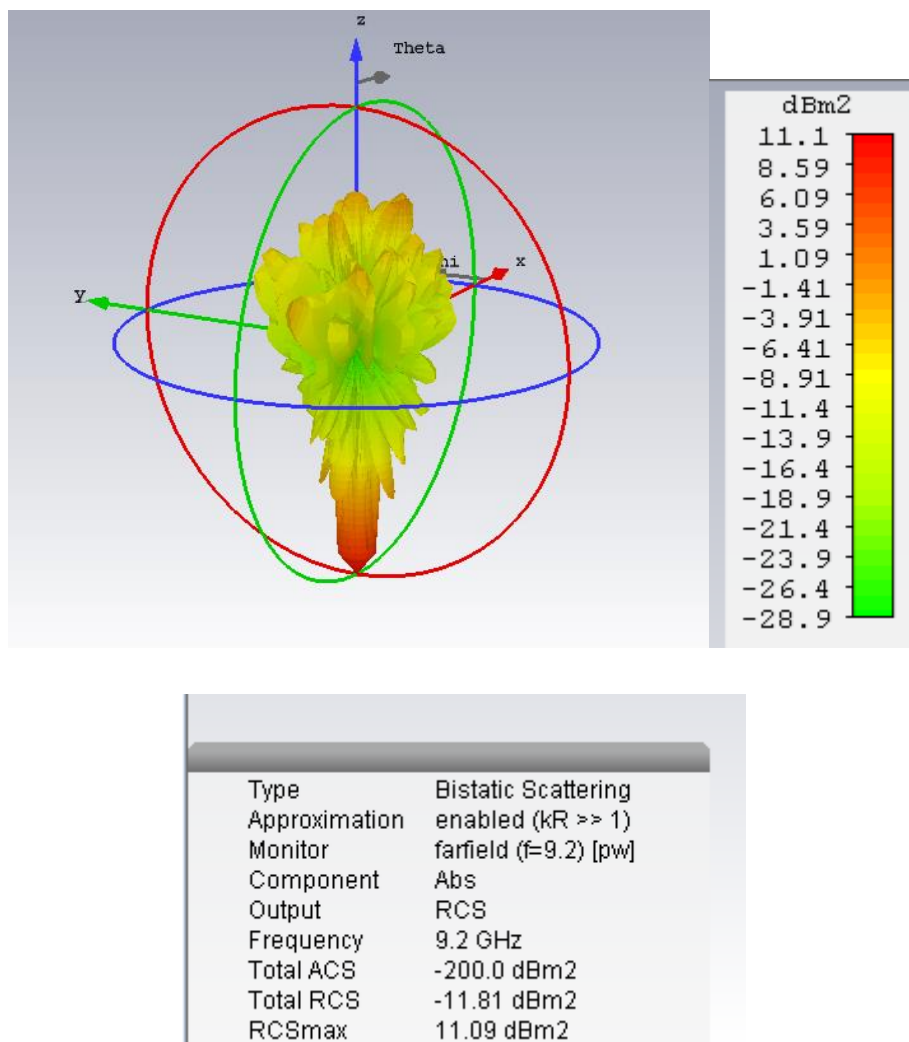


Figure 6.12: Typical far-field response of high- density metasurfaces

	8 GHz	8.2 GHz	8.4 GHz	8.6 GHz	8.8 GHz	9 GHz	9.2 GHz	9.4 GHz	9.6 GHz	9.8 GHz	10 GHz	10.2 GHz	10.4 GHz	10.6 GHz	10.8 GHz	11 GHz	11.2 GHz	11.4 GHz
periodic	0.3	0.3	0.9	3.8	6.3	6.6	8.2	12.7	14.1	15.2	13.8	7.6	3.8	1.9	1.2	0.9	0.7	0.6
hyper1	0.4	0.6	1.1	3	6.5	8.3	9.9	12.8	10.6	9.7	8.3	8.6	6.3	2.9	1.8	1.3	1	0.8
hyper2	0.4	0.6	1	3	8.3	10.3	13.5	12.9	12.7	11.8	10.9	8.6	5.7	2.7	1.8	1.3	1	0.8
hyper3	0.4	0.9	1.5	3.8	9.9	10.9	11.3	12.8	11.3	11.3	12.2	8.5	5.5	2.7	1.8	1.4	1.1	0.9
hyper4	0.4	0.6	0.8	1.9	5.3	7.7	9.1	11.5	10.2	11.6	10.4	8.7	5.4	2.6	1.7	1.3	1	1

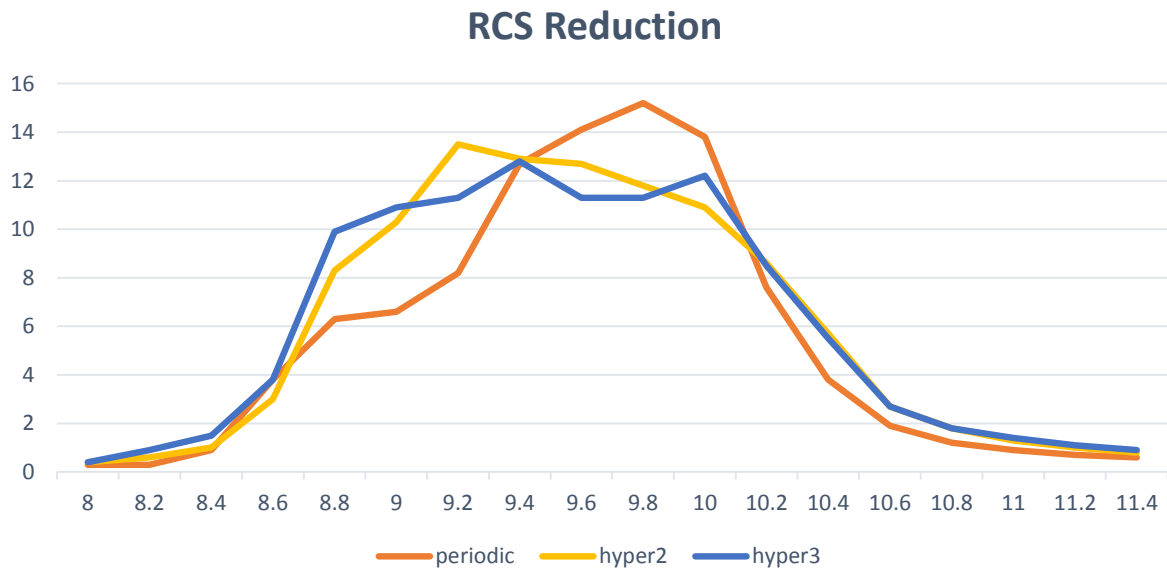


Figure 6.13: Table depicting RCS reduction bandwidth of various metamaterial structures.

As can be seen, periodic structures do appear to have a disadvantage over hyperuniform ones, most prominently at the 9.6 to 10 GHz range. These results are somewhat preliminary, and further study would be required to confirm them.

6.2: Summary

This chapter details a study of hyperuniform metamaterial structures, with the aim of reducing back-scattering and radar cross-section. These structures consisted of multiple angular dipole antenna arranged in periodic, hyperuniform and random arrangements, and at various numbers in a given structure. The various effects of these differences were compared to one another, with a focus on their far-field responses.

The primary hypothesis under investigation was that hyperuniform structures would have an advantage in achieving the desired goal of reduced far-field reflection over their periodic and random counterparts for certain angles of incidence. The results, albeit preliminary at this stage, appear to bear that hypothesis out.

In the future, it may be possible to apply the genetic algorithm to this problem, optimising for hyperuniform distribution, structure tine length, structure opening angle or some combination of the three. This could improve the performance of the structure and lend itself to many applications.

6.3: References

- [1] D. J. Kern, D. H. Werner, M. Lisovich, "Metaferrites: Using Electromagnetic Bandgap Structures to Synthesize Metamaterial Ferrites" *IEEE Transactions on Antennas and Propagation Volume 53 Issue 4*, April 2005
- [2] Z. Bayraktar, M. D. Gregory, X. Wang, D. H. Werner, "Matched Impedance Thin Planar Composite Magneto-Dielectric Metasurfaces", *IEEE Transactions on Antennas and Propagation*, April 2012.
- [3] W.H. Bragg, W.L. Bragg, "The Reflexion of X-rays by Crystals". *Proc. R. Soc. Lond. A.* 88 (605): 428–38, 1913
- [4] S. Priya, "Importance of structural irregularity on dielectric loss in $(1-x)\text{Pb}(\text{Mg}_{1/3}\text{Nb}_{2/3})\text{O}_3-(x)\text{PbTiO}_3$ crystals" *Appl. Phys. Lett.* 80, 421, 2002
- [5] G. Gkantzounis, T. Amoah, M. Florescu, "Hyperuniform Disordered Phononic Structures", *Physical Review B*, March 2017
- [6] E. C. O'guz, J. E. S. Socolar, P. J. Steinhardt, S. Torquato, "Hyperuniformity and anti-hyperuniformity in one-dimensional substitution Tilings", June 2018
- [7] B. Barak, R. Shaltiel, E. Tromer, "True Random Number Generators Secure in a Changing Environment" *International Workshop on Cryptographic Hardware and Embedded Systems*, 2003
- [8] F. Wen, S. David, X. Checoury, M. El Kurdi, P. Boucaud, "Two-dimensional photonic crystals with large complete photonic band gaps in both TE and TM polarizations", *Optics Express Vol. 16, Issue 16*, pp. 12278-12289, 2008
- [9] G. Zhang, F. H. Stillinger, S. Torquato "The Perfect Glass Paradigm: Disordered Hyperuniform Glasses Down to Absolute Zero", *Scientific Reports volume 6, Article number: 36963*, 2016
- [10] B. Wu, Y. Hao *PLOS one*, 2018
- [11] N. Yu, F. Capasso, "Flat optics with designer metasurfaces". *Nat. Mater.* 13: 139–150, 2014

[12] N. Bloembergen, P. S. Pershan, "Light waves at the boundary of nonlinear media" *Physical Review*. 128: 606. 1962

[13] E. Hecht, *Optics Fourth Edition*, Addison Wesley, 2002

[14] N. Yu, P. Genevet, M. A Kats, F. Aieta, J.P. Tetienne, F. Capasso, Z. Gaburro, "Light Propagation with Phase Discontinuities: Generalized Laws of Reflection and Refraction." *Science (New York, N.Y.)*. 334. 333-7. [10.1126/science.1210713](https://doi.org/10.1126/science.1210713), 2011

Chapter 7: Conclusion and Future Work

7.1: Conclusions

7.1.1: Novel Contributions

Like most projects of this nature, this thesis sought to provide a novel contribution to the field of antenna engineering. It has achieved this by providing a new, integrated approach to metaferriite design as well as several designs that could potentially see use as absorbers and bandwidth-enhancing back plates for antenna applications. It has also laid the groundwork for designing newer, improved versions of the same at a later date

The more significant product, however, is the genetic algorithm created over the course of this work. It's this genetic algorithm that was used to design the structures mentioned and that could be used to generate improved structures. It is a flexible tool that can be reconfigured for other design goals with minimal alteration, and incorporates a multi-program approach to analysis and data collection. It is this latter feature that provides a novel approach to certain problems, as incorporating a program to simulate the structures designed instead of relying on analytical functions is something that has not been done before. The GA was also highly modular in nature, allowing sections to be easily replaced. This allows it to easily be adapted to a range of problems with minimal alterations to the surrounding algorithm.

The hyperuniform structures, albeit preliminary, could also yield useful contributions at a later date, and could also be the subject of optimisation to improve their performance beyond what has already been seen.

7.1.2: Summary

The outputs of this project can essentially be divided into two groups: those being the genetic algorithm program designed to aid in the creation of a variety of metaferriites, and the structures that this algorithm designed that were subsequently fabricated.

The former of the two is the more substantial product. While there were a few setbacks along the way, the GA has slowly developed as the project progressed. The surfaces produced were created not simply for their own end, but also to test and improve the genetic algorithm, and the knowledge gained by studying them will ultimately go back into the

algorithm that created them to upgrade and improve it, in order to produce new, superior structures.

That is not to say that the fabricated structures were unimportant, of course. There are multiple fabrication methods to choose from and studying them to determine which would be most efficient and reliable was an important part of the process, as was understanding the processes involved in those that were new and unfamiliar. Likewise, the techniques used to analyse the structures, and the design limitations that need to be taken into account to accommodate that, are also important features.

This was also the reason behind some of the early metaferrite design exercises. Not only were they intended to explore the ins and outs of the mechanics of metamaterial surfaces, they also served to increase familiarity and understanding of the tools used to create them. The hexagon-based structure that was so difficult to make at the time would likely have been far easier to recreate had that been attempted at the tail-end of the project, but that learning experience was necessary to ensure that the commercial software was set up properly and produced the correct results when used to analyse the metasurface structures produced by the genetic algorithm, even though some of the techniques introduced during that exercise, such as the non-rectangular unit cell, were ultimately not used in the course of the main project. They may, however, be of use in further exploration of the topic by future research.

7.1.3: Genetic Algorithm

As stated previously, this project sought to create a genetic algorithm that could aid the design of metamaterial ferrite structures. To this end, it was largely successful. The genetic algorithm created by this project is a flexible tool that can be fine-tuned to produce a number of metamaterial high-impedance surface structures. It is also modular in nature, allowing it to be quickly and easily reconfigured to a new task simply by swapping out pre-created modules by altering a few lines of code. Both the fitness function and the vba construction macros can be interchanged this way, permitting almost any alteration of parameters desired.

This flexibility of the genetic algorithm has already been capitalised on in various ways. Numerous materials can potentially be used to form the conductive surface of the structure, and several of these have been simulated using the GA with limited alteration to its

core structure. The incorporation of commercial software provides further flexibility, as it means that there will be no shortage of parameters for the simulation of even unconventional substances, available in the built in material library. Those that are not can be easily constructed using the available parameter-entry tools.

The ease with which the fitness function can be replaced facilitates the search for solutions with different parameters, as replacing a given function with a slightly different equation can be done quickly, simply, and in an easily reversible manner. One simply needs to create a new file for the new function and swap out the prompt to run it in the parent file. Since it is possible to make reconfigurable devices that can act as either a high impedance surface or an absorber, this should not require much more than an alteration of the fitness function. As such turning the algorithm towards the design of absorbers or other similar structures could be done with ease.

There is room for improvement, however. A notable issue was the long run times involved with the program. This often led to it needing to be left running for days or even weeks before it would produce a properly optimised structure, and such an extended period of operation left the program vulnerable to network failure or general computer errors. This frequently stymied attempts to run the program for extended lengths of time.

There was also some concern about the specifics of the results derived. While qualitatively what was expected, they did display a numerical issue, with the actual values predicted being inflated to far beyond what was expected for structures of this type. This meant that additional simulations were required before the structures generated could be used in antennas. It was eventually deemed that an error in the calculation of β_0 may have been behind it, and this was rectified. Subsequently a new round of simulation was performed in order to assess how this change had altered the function of the genetic algorithm.

This testing involved multiple variants of the L-system in order to determine which of them was best suited to the new parameters. The first was tended to produce high metal structures that did not display the desired properties at all. It was determined that this was in fact due to the high fraction leading to a tendency for the segments of the design to link together, depriving the structure of its capacitor nature. This is an important observation and is something to look out for in the future, but for now it meant that such an L-system was unsuitable.

The second variant went in the opposite direction, producing extremely sparse designs. This did allow the structures to resonate with a degree of consistency, but generally at a very high frequency. This is likely due to the large gaps produced between pieces of metal, and their small size.

A third variant sought to compromise between the two, and was largely successful. It produced the desired middle ground of metal to substrate ratio, and reduced the resonant point to within the desired 8 to 12 GHz range with a degree of consistency, although a few outliers persisted. The program did however appear to get stuck at a local minima that produced responses with a higher imaginary component than desired, which may be a consequence of the reduced design space.

It is possible that, upon reflection, the reduction in complexity and computer load the L-system granted did not pay off in the long run. The benefits in speed seem to be outweighed by the cost in overall versatility. Future work to increase design space may be worth pursuing.

7.1.4: Fabrication

The genetic algorithm produced multiple potential designs for a metaferrite structure. Several of these designs produced by the genetic algorithm were selected to progress to the fabrication stage, and ultimately two such structures were created. These were then analysed and compared to one another at both short range and long range, in and outside an anechoic chamber. The results showed moderate agreement with those predicted by simulation, although in one case time gating had to be employed in order to make this apparent. The results found here could be put towards improvements to the genetic algorithm, or possibly refine the parameters of the analysis performed by the commercial software.

The structures created provide a solid foundation for the topic, but more could certainly be done using the genetic algorithm. More adventurous design challenges and other, more novel approaches could be addressed, and the results of these endeavours could also provide insights on how the algorithm could be improved further. Perhaps refinements could be made to the fitness function, or alterations to the parameters of recombination could be made.

Overall, it was concluded that the best material to use for fabrication as of yet attempted was Etched Copper on an FR4 substrate. This was the material combination that

could be fabricated with the most consistency, compared to inkjet-printed silver. This allowed for a larger structure to be created, which was in turn easier to analyse with the available equipment. It is possible that further exploration could yield a process that is even better, either in terms of ease of use or consistency.

Subsequent simulations of the structure also indicated that it could reduce the radar cross-section of a device compared to a solid metal substrate. This has a number of potential applications, and may be useful to study in depth at a later date.

7.1.5: Other Parameters Studied.

During the construction of the algorithm, several parameters were studied in order to ascertain the effects they would have on the finished structure. One example is the metallic fraction of the structure. This was studied by logging the metal fraction of structures generated by the genetic algorithm and comparing them to the parameters of those structures.

At first, it appeared that there was a correlation between the fraction of metal and the resonance point of the structure's permittivity. However, as data accumulated this correlation slowly disappeared, leading to the conclusion that it was in fact other features of the structure, which metal fraction is often determined by, that was causing the pattern. This initial false positive likely came about because the data was accumulated through multiple runs using different L-system configurations. Certain l-systems do lend themselves to particular resonance points more easily, so when there were only a few systems in the data, each tending towards a different resonance point and metal fraction, it just so happened that these two variables lined up with each other to a certain degree. As other systems which tend to the same fraction appeared in the data this spurious correlation was corrected.

Later on it also became clear that a sufficiently high metal fraction can potentially disrupt the function of the device, as a continuous metal structure does not possess the capacitance necessary for the high-impedance required to arise. As with the previous examples, this is more a consequence of the structures a given metal fraction lends itself to than the fraction itself.

So, in conclusion, it was determined that metallic fraction can have an indirect effect on the parameters of structure created by influencing the structure itself, but that it does not possess any direct connection to the resonance point.

7.2: Future Work

There are a number of directions in which research on the topic of this thesis could progress further. The materials, design parameters, design goal or even the optimisation algorithm itself could be altered. Each of these would provide a point of comparison with which the existing algorithm could be contrasted and potentially improved. As previously mentioned, the flexibility of the genetic algorithm facilitates some of these avenues. Replacing the algorithm itself would not be among them, obviously, as it would require rebuilding from the ground up.

First and foremost, the project could simply continue on its present course. There is room for improvement in the genetic algorithm, certainly. It could be faster, more elegant, and there are likely many refinements to its function that could be made. New designs could be generated and fabricated, and the results used to upgrade the GA further. The structures previously fabricated are certainly valuable points of study, but given the refinements that have already been made to the GA since they were created, it is unlikely that they will achieve the project's initial goal. New structures could approach those parameters to a better extent.

7.2.1: Alternate Materials

One possible alternate material that could be investigated is graphene. This was studied to a small degree by this project, but ultimately nothing came of it. Graphene is a material with a range of interesting properties, primarily derived from its unique "two-dimensional" nature due to being merely an atom thick. A more in-depth study could yield better results, and possibly take advantage of alternative design techniques.

Other metals might also be an option. Silver has numerous beneficial properties, but it is expensive, and the main fabrication method using it proved difficult to use consistently. If another method of fabrication using it was found, that may be worth investigating. Alternately metals with varying degrees of losses could be compared to one another, or other forms of conductive ink.

Another component that could be altered is the substrate. This is often dictated by the constraints of a given intended fabrication method, but this has shown to have some degree of influence on the structure in simulation, and where there are multiple options available it

would be worth making a comparison between them to determine which was optimal for a given problem.

7.2.2: Alternate Optimisation Procedure

Another possible avenue of research could be to attempt another form of optimisation procedure aimed at the same design goal or, at the very least, a similar one. This would facilitate an appraisal of the two by allowing a comparison the derived solutions. If time is available it might even be worth comparing the solutions produced by both to multiple problems, in order to get a comprehensive overview of which is better suited to a given design goal.

The Particle Swarm technique would be a likely candidate for such an endeavour, given that it has a similar concept at its core (mimicry of processes found in nature) and is a reasonably well-established technique. It is also sufficiently different in execution that the final results would probably have a reasonably distinct form, and if this were not in fact the case that itself would be interesting as well. Alternate optimisation could also be applied to other materials which the genetic algorithm struggled to find solutions for, such as graphene, and comparisons between them on a material by material basis could be constructed.

This would, of course, represent a project of comparable magnitude to the one described in this thesis. The algorithm would need to be constructed from scratch, and an equivalent design parameter to that used here would need to be created. This perhaps stretches the definition of what could be considered further work on this project a little too far. But, as a project to spin off of this one, it is worth considering.

7.2.3: Alternate Goals

On the other hand, attempting to generate another form of structure with the same method rather than attempt the same goal with a different method would certainly qualify as an expansion. In this case, the bulk of the genetic algorithm program could remain intact, as only the export and fitness function modules would need to be replaced.

The new goal could also be an attempt to replicate one done by others in the past. Switching the design to an absorber or some other component would provide a demonstration of the solver's adaptability and the robustness of the algorithm. On the other hand, a completely novel design would have more impact and possibly provide new insights into the

topic, so perhaps that avenue would be the better one to pursue. This would require a far more substantial reconstruction of the build modules, of course, but if one was careful the new design could still be based off the same seed code, or one with only minor alterations to the L-system to produce one of a slightly different length.

Another avenue of research could be to investigate the results of some of the design constraints being loosened or tightened. A full fractal pattern could be implemented, rather than the somewhat limited version used in the project. Another possibility is that variations on the structure's symmetry could be explored.

Machine Learning could also receive a more in-depth exploration. A comprehensive data bank of solutions would greatly speed up analysis, not only by excising repeated structures but also by potentially allowing the program to make intuitive leaps about the expected results of those it has yet to examine. The latter approach would likely require extensive re-working of the program, however, so much like the alternate optimisation approaches its implementation would be very dependent on the time remaining in the project.

7.2.4: Further exploration of previously touched upon designs.

Over the course of this project other designs and techniques have received a degree of attention, such as hyperuniform structures and multi-layer metamaterials. These are both avenues that could receive further exploration. The benefits of the multilayer structure appear to be somewhat orthogonal to the goals of this project, so a separate study may be the best approach there, while applying a hyperuniform approach to the metaferrite structures designed by this project would require a certain degree of re-working.

This could take many forms, such as a hyperuniform distribution of pixels in the initial structure generation as opposed to the randomised generation seen currently, or perhaps a hyperuniform distribution of unit cells, or perhaps both, although this may not have any real benefit.

Appendix: MATLAB code

GenetAlg_cluster

```
clear, clc
Dexcount=3800;
Dexno=Dexcount;
Dexno2=Dexcount;
dbno=1;
run nextgen_cluster
runs=1;
G=1;
match=0;

while Elite<0 && runs<500
    run nextgen_cluster2
    runs=runs+1;
    [Elite,Position]=min(T);
    disp(Elite)
    disp(Position)
    Alpha=Newpop(:,Position);

end
disp('Best value is')
disp(Elite)
Plant=Alpha;
while length(Plant)<45
    k=length(Plant);
    l=1;
    for j=1:k

        if Plant(j)==1
            Y(l)=0;
            Y(l+1)=1;
            l=l+2;
        else
            Y(l)=1;
            Y(l+1)=0;
            l=l+2;
        end
    end
    Plant=Y;
    disp(Plant)
end
disp ('SCREENCAP THIS BEFORE IT VANISHES -->')
```

nextgen_cluster

```
run seedgen_cluster
[Length,Width]=size(Cross);
```

```

for n=1:Width
    Cutpoint=round(rand*(Length-1)+1);
    for p=1:Cutpoint
        Newpop(p,n)=Alpha(p,1);
    end
    for q=Cutpoint+1:Length
        Newpop(q,n)=Cross(q,n);
    end
    disp('next1a')
end

[Length2,Width2]=size(Mut);
mutgene=3;
t=1;
for r=Width+1:Population
    Newpop(:,r)=Mut(:,t);
    for s=1:mutgene
        Mutpoint=round(rand*(Length2-1)+1);
        if Newpop(Mutpoint,r)==1
            Newpop(Mutpoint,r)=0;
        else
            Newpop(Mutpoint,r)=1;
        end
    end
    disp('next1b')
    t=t+1;
end

```

seedgen_cluster

```

Population=12;
Tolerance=5;
k=1;
m=1;
i=1;
T(1)=0;
match=0;
fitnum=1;
while i<=Population
    run seed
    if i==1 || seednum ~= varnum
        Seed(:,i)=var;
        seedflip(1,:)=Seed(:,i);
        seedchar=num2str(seedflip);

        g=1;
        for f=1:length(seedchar)
            if seedchar(f)=='1' || seedchar(f)=='0'
                seedcharshort(g)=seedchar(f);
                g=g+1;
            end
        end
    end
    i=i+1;
end

```

```

    end
end

seednum=str2num(seedcharshort);
PlantX(:,i)=var;
end
i=i+1;
disp('I am working')
end
disp('This is Generation 1')
i=1;
while i<=Population
    Plant=PlantX(:,i);
while length(Plant)<45 %put this back in when done debugging.
    k=length(Plant);
    l=1;
    for j=1:k

        if Plant(j)==1
            Y(l)=0;
            l=l+1;
        else
            Y(l)=1;
            Y(l+1)=0;
            l=l+2;
        end
    end
    Plant=Y;

    disp(Plant)

end
Plantsum = sum(Plant(1:45));
Plantdiag = [Plant(9), Plant(17), Plant(24), Plant(30), Plant(35), Plant(39), Plant(42),
Plant(44), Plant(45)];
diagsum =sum (Plantdiag);
    Plantfrac = ((Plantsum*2)-diagsum)/81;
    disp('Plantfrac equals')
    disp(Plantfrac)
PlantY(:,i)=Plant;
i=i+1;
end
i=1;
run program_selector_fr4
while i<=Population
    Plant=PlantY(:,i);

if Pha0<150 T(i)=FitMin(i);
    disp('T-Value is')

```

```

disp(T(i))

end
i=i+1;
disp(length(T))
end
%end

[Elite,Position]=min(T);
Alpha=Seed(:,Position);
q=0;
o=1;
Tbest=T(Position);
for j=1:Population
if T(j)<=Tbest+Tolerance && q<8
    CrossID(o)=j;
    Cross(:,o)=Seed(:,j);

    [Length,Width]=size(Cross);
    disp('Cross is')
    disp(Width)
    o=o+1;
    q=q+1;

else
    T(j)= Tbest+Tolerance+1000;
end
end

for l=1:Population
    if T(l)>Tbest+Tolerance %||dupe(l)==1
        Mut(:,m)=Seed(:,l);
        MutID(m)=l;
        m=m+1;
        mutcheck=1;
    end
end

seed

for q=1:9
y=round(rand*9+1);
for z=1:y
x=rand;
end
if x>0.5
    var(q,1)=1;
else
    var(q,1)=0;

```

```

end
end

varflip(1,:)=var(:,1);
vchar=num2str(varflip);

g=1;
for f=1:length(vchar)
    if vchar(f)=='1' || vchar(f)=='0'
        vcharshort(g)=vchar(f);
        g=g+1;
    end
end

varnum=str2num(vcharshort);

```

Nextgen_cluster_2

run seedgen_cluster2

```

[Length,Width]=size(Mut);
[Length2,Width2]=size(Cross);
mutgene=3;
Mutmax=Population-Width2;
if Mutmax<=4
    Mutpop=4;
else
    Mutpop=Mutmax;
end
clear Newpop
for r=1:Mutpop
    Newpop(:,r)=Mut(:,r);
    for s=1:mutgene
        Mutpoint=round(rand*(Length-1)+1);
        if Newpop(Mutpoint,r)==1
            Newpop(Mutpoint,r)=0;
        else
            Newpop(Mutpoint,r)=1;
        end
    end
end

end

```

```

t=1;
for n=Mutpop+1:Population
    Cutpoint=round(rand*(Length-1)+1);
    for p=1:Cutpoint
        Newpop(p,n)=Alpha(p,1);
    end
    for q=Cutpoint+1:Length
        Newpop(q,n)=Cross(q,t);
    end
end

```

```

    end
    t=t+1;
end

```

seedgen_cluster2

```

k=1;
m=1;
G=G+1;
r=1;
clear PlantX
while r<=Population
% run seed
% Seed(:,i)=var;
PlantX(:,r)=Newpop(:,r);
disp('seed2a')
r=r+1;
end
disp('this is generation')
disp(G)
r=1;
while r<=Population
    Plant=PlantX(:,r);
while length(Plant)<45
    q=length(Plant);
    l=1;
    for j=1:q

        if Plant(j)==1
            Y(l)=0;
            l=l+1;
        else
            Y(l)=1;
            Y(l+1)=0;
            l=l+2;
        end
    end
    Plant=Y;
    disp(Plant)
end
Plantsum = sum(Plant(1:45));
Plantdiag = [Plant(9), Plant(17), Plant(24), Plant(30), Plant(35), Plant(39), Plant(42),
Plant(44), Plant(45)];
diagsum =sum (Plantdiag);
Plantfrac = ((Plantsum*2)-diagsum)/81;
disp('Plantfrac equals')
disp(Plantfrac)
PlantY(:,r)=Plant;

r=r+1;

```

```

disp('seed2b')
end
r=1;
run program_selector_fr4
while r<=Population
    Plant=PlantY(:,r);

    if Pha0<150 T(r)=FitMin(r);
        disp('T-Value is')
        disp(T(r))

    end
    r=r+1;
    disp('seed2c')
    end
    [Elite,Position]=min(T);
    Alpha=Newpop(:,Position);
    Abort=8;
    dump=1;
    k=1;
    for j=1:Population
        if T(j)<T(Position)+Tolerance
            if k<Abort
                Cross(:,k)=Newpop(:,j);
                CrossID(k)=j;
                disp('Cross is')
                disp(Width2)
                k=k+1;
            else
                Mut(:,dump)=Newpop(:,j);
                MutID(dump)=j;
                dump=dump+1;
            end
        end
    end
    disp('seed2d')
    end

    for l=1:Population
        if T(l)>=T(Position)+Tolerance
            Mut(:,m+dump-1)=Newpop(:,l);
            MutID(m+dump-1)=l;
            m=m+1;
        end
    end
    disp('seed2e')
end

```


program_selector_fr4

```
Dexno=Dexno+1;  
psscount=1;
```

```
run impexclusterfr4_1  
run Dbasefnconv  
Dexno=Dexno+1;  
run impexclusterfr4_2  
run Dbasefnconv  
Dexno=Dexno+1;  
run impexclusterfr4_3  
run Dbasefnconv  
Dexno=Dexno+1;  
run impexclusterfr4_4  
run Dbasefnconv  
Dexno=Dexno+1;  
run impexclusterfr4_5  
run Dbasefnconv  
Dexno=Dexno+1;  
run impexclusterfr4_6  
run Dbasefnconv  
Dexno=Dexno+1;  
run impexclusterfr4_7  
run Dbasefnconv  
Dexno=Dexno+1;  
run impexclusterfr4_8  
run Dbasefnconv  
Dexno=Dexno+1;  
run impexclusterfr4_9  
run Dbasefnconv  
Dexno=Dexno+1;  
run impexclusterfr4_10  
run Dbasefnconv  
Dexno=Dexno+1;  
run impexclusterfr4_11  
run Dbasefnconv  
Dexno=Dexno+1;  
run impexclusterfr4_12  
run Dbasefnconv
```

```
X=1;  
Dexno2=Dexno2+1;  
run impexcluster21  
run Dbasefnconv2  
run Fitfun1  
Dexno2=Dexno2+1;  
X=X+1;  
run impexcluster22
```

```
run Dbasefnconv2
run Fitfun1
Dexno2=Dexno2+1;
X=X+1;
run impexcluster23
run Dbasefnconv2
run Fitfun1
Dexno2=Dexno2+1;
X=X+1;
run impexcluster24
run Dbasefnconv2
run Fitfun1
Dexno2=Dexno2+1;
X=X+1;
run impexcluster25
run Dbasefnconv2
run Fitfun1
Dexno2=Dexno2+1;
X=X+1;
run impexcluster26
run Dbasefnconv2
run Fitfun1
Dexno2=Dexno2+1;
X=X+1;
run impexcluster27
run Dbasefnconv2
run Fitfun1
Dexno2=Dexno2+1;
X=X+1;
run impexcluster28
run Dbasefnconv2
run Fitfun1
Dexno2=Dexno2+1;
X=X+1;
run impexcluster29
run Dbasefnconv2
run Fitfun1
Dexno2=Dexno2+1;
X=X+1;
run impexcluster30
run Dbasefnconv2
run Fitfun1
Dexno2=Dexno2+1;
X=X+1;
run impexcluster31
run Dbasefnconv2
run Fitfun1
Dexno2=Dexno2+1;
X=X+1;
run impexcluster32
```



```

Top=max(B.data(:,2));
Bottom=min(B.data(:,2));

Dif=Top+Bottom;

if Dif<Top
    disp('Resonance achieved')
else
    disp('resonance not achieved')
end

[Max,Place]=max(A.data(:,2));

disp(Max)
disp(A.data(Place,1))

Fz= A.data(:, 1);
S11real = A.data(:, 2);
S11imag = B.data(:, 2);

S = S11real+1i*S11imag;
Z0 = 376.73 ;
Z_imp = Z0*((1+S)/(1-S));
n = length(S11real) ;

Mu0=4*pi*10^-7;
Eps0=8.854*10^-12;
c = 299792458;
lambda = c./(Fz*10^9);
Beta0=((2*pi)/lambda);
d=0.0024;

Epsr=(1./(1i.*Beta0.*d)).*atanh(Z_imp/Z0);
EpsRE=real(Epsr);
EpsIM=imag(Epsr);

ph = angle(S)*180/pi;%unwrap();
figure(3)
plot(Fz,ph)
xlabel('Frequency(GHz)')
ylabel('Phase(deg)')
grid on

figure(5)
plot(Fz,20*log10(abs(S)))
xlabel('Frequency(GHz)')
ylabel('|S11|(dB)')
title('Magnitude (dB)')
grid on

```



```

Plant(33),Plant(34),Plant(35),Plant(36),Plant(37),Plant(38),Plant(39),Plant(40),Plant(41),Plant(42),Plant(43),Plant(44),Plant(45));
!"C:\Program Files (x86)\CST STUDIO SUITE 2014\CST DESIGN ENVIRONMENT.exe"
-m structmac-solver2_ink2.bas
A = importdata('S11_Real_CST.txt');
B = importdata('S11_Imag_CST.txt');
C = importdata('S11_Phase_CST.txt');
Top=max(B.data(:,2));
Bottom=min(B.data(:,2));

Dif=Top+Bottom;

if Dif<Top
    disp('Resonance achieved')
else
    disp('resonance not achieved')
end

[Max,Place]=max(A.data(:,2));

disp(Max)
disp(A.data(Place,1))

Fz= A.data(:, 1);
S11real = A.data(:, 2);
S11imag = B.data(:, 2);

S = S11real+1i*S11imag;
Z0 = 376.73 ;
Z_imp = Z0*((1+S)/(1-S));
n = length(S11real) ;

Mu0=4*pi*10^-7;
Eps0=8.854*10^-12;
Beta0=(Fz*1e9/(2*pi))*sqrt(Mu0*Eps0);
d=0.001;

% This is where the problem is, I think.
Epsr=(1./(1i.*Beta0.*d)).*atanh(Z_imp/Z0);
EpsRE=real(Epsr);
EpsIM=imag(Epsr);

ph = angle(S)*180/pi;%unwrap();
figure(3)
plot(Fz,ph)
xlabel('Frequency(GHz)')
ylabel('Phase(deg)')
grid on

figure(5)

```



```
plot(Fz,20*log10(abs(S)))
xlabel('Frequency(GHz)')
ylabel('|S11|(dB)')
title('Magnitude (dB)')
grid on
```

```
figure(1)
plot(A.data(:,1),A.data(:,2))
title('Real Part')
figure(2)
plot(B.data(:,1),B.data(:,2))
title('Imaginary Part')
```

```
figure(7)
plot(Fz,EpsRE)
title('Eps/MU Real')
```

```
figure(8)
plot(Fz,EpsIM)
title('Eps/Mu Imaginary')
```

```
figure(9)
plot(Fz,EpsRE,'r')
hold on
plot(Fz,-EpsIM)
title('eps/mu superimp')
xlabel('frequency (GHz)')
hold off
```

```
Database(:,dbno)= importdata('gridstore.txt');
dbno=dbno+1;
```