

# “Enhancing numerical modelling efficiency for electromagnetic simulation of physical layer components”

---

Hugh Granville Sasse, BSc(Hons)

A thesis submitted in partial fulfilment of the requirements of De Montfort University for the Degree of Doctor of Philosophy

26 October 2010

De Montfort University

To my parents, Patricia and Graham

# Abstract

The purpose of this thesis is to present solutions to overcome several key difficulties that limit the application of numerical modelling in communication cable design and analysis. In particular, specific limiting factors are that simulations are time consuming, and the process of comparison requires skill and is poorly defined and understood. When much of the process of design consists of optimisation of performance within a well defined domain, the use of artificial intelligence techniques may reduce or remove the need for human interaction in the design process. The automation of human processes allows round-the-clock operation at a faster throughput. Achieving a speedup would permit greater exploration of the possible designs, improving understanding of the domain.

This thesis presents work that relates to three facets of the efficiency of numerical modelling: minimizing simulation execution time, controlling optimization processes and quantifying comparisons of results. These topics are of interest because simulation times for most problems of interest run into tens of hours. The design process for most systems being modelled may be considered an optimisation process in so far as the design is improved based upon a comparison of the test results with a specification. Development of software to automate this process permits the improvements to continue outside working hours, and produces decisions unaffected by the psychological state of a human operator. Improved performance of simulation tools would facilitate exploration of more variations on a design, which would improve understanding of the problem domain, promoting a virtuous circle of design.

The minimization of execution time was achieved through the development of a Parallel TLM Solver which did not use specialized hardware or a dedicated network. Its design was novel because it was intended to operate on a network of heterogeneous machines in a manner which was fault tolerant, and included a means to reduce vulnerability of simulated data without encryption. Optimisation processes were controlled by genetic algorithms and particle swarm optimisation which were novel applications in communication cable design. The work extended the range of cable parameters, reducing conductor diameters for twisted pair cables, and reducing optical coverage of screens for a given shielding effectiveness. Work on the comparison of results introduced “Colour maps” as a way of displaying three scalar variables over a two-dimensional surface, and comparisons were quantified by extending 1D Feature Selective Validation (FSV) to two dimensions, using an ellipse shaped filter, in such a way that it could be extended to higher dimensions. In so doing, some problems with FSV were detected, and suggestions for overcoming these presented: such as the special case of zero valued DC signals. A re-description of Feature Selective Validation, using Jacobians and tensors is proposed, in order to facilitate its implementation in higher dimensional spaces.

# Contents

Abstract.....	3
Contents .....	4
Table of figures .....	7
Table of Abbreviations .....	9
Table of Symbols .....	10
Acknowledgements .....	11
1 Introduction.....	12
1.1 Numerical Modelling in electromagnetics. ....	12
1.2 Difficulties with Existing Techniques.....	13
1.3 Components of Particular Interest.....	14
1.4 Optimisation.....	14
1.5 Outline of the research .....	18
2 Literature Search.....	19
2.1 Introductory Remarks .....	19
2.2 Parallelisation.....	21
2.3 TLM. ....	24
2.4 Optimisation.....	25
2.4.1 Particle Swarm Optimisation. ....	31
2.5 Validation.....	36
2.5.1 Validation and Feature Selective Validation.....	38
2.5.2 The Feature Selective Validation Method.....	39
2.5.3 How was FSV validated? .....	40
2.5.4 FSV as a tool in knowledge management .....	40
2.6 Conclusion. ....	41
3 Optimisation of Cable Design.....	43

3.1 Genetic Algorithms and The Design of RF Structures. ....	43
3.1.1 Introduction.....	43
3.1.2 Modelling the Electrical Properties of Unshielded Twisted Pair Cable.....	44
3.2 Particle Swarm Optimiser .....	45
3.3 Optimisation and Cable Design Results.....	47
4 Parallel Distributed Implementation of a Transmission Line Modelling Method Solver .....	63
4.1 Introduction and Design Criteria.....	63
4.2 The choice of an object oriented language.....	65
4.3 Linda, Rinda, and the Tuplespace model of programming .....	67
4.4 Organisation of the sharing of work. ....	67
4.5 Security Considerations .....	71
4.6 Node Design.....	73
4.7 Parallel TLM.....	73
5 Two Dimensional Feature Selective Validation .....	81
5.1 Introduction.....	81
5.2 FSV Equations. ....	82
5.3 Problems with FSV Formulation .....	85
5.3 Details of Implementation.....	85
5.4 A Mathematical Foundation for Higher Dimensional FSV .....	89
5.5 Two Dimensional Feature Selective Validation Results.....	95
6 Discussion.....	109
6.1 Introduction.....	109
6.2 Genetic algorithms and computer modelling of cables.....	109
6.3 Parallel TLM modelling.....	110
6.4 Two dimensional feature selected validation.....	111
6.5 Optimisation, Modelling, and Validation as a Toolkit.....	112
7 Conclusions and Future Work .....	114
7.1 Introduction.....	114
7.2 Conclusions from the Research.....	114

7.3 Mainstream Future Work .....	114
7.3.1 Parallel TLM.....	115
7.3.2 Optimisation.....	118
7.3.3 Satisficing .....	118
7.3.4 Mechanical Properties of Communication Cables.....	118
7.3.5 Feature Selective Validation .....	118
7.3.5.3 Filtering in the Fourier Domain .....	119
7.3.6 Human perception.....	120
7.3.7 Mathematical Vulnerabilities.....	120
7.4 Research likely to be of Interest in the Longer Term.....	121
7.4.1 Parallel TLM Solver.....	121
7.4.2 Feature Selective Image Validation.....	122
8 List of Publications .....	124
9 References.....	126
Appendix A – Implementation of FSV method (quoted from IEEE1597.1 “Standard for Validation of Computational Electromagnetics Simulation and Modeling” 2009) .....	140

# Table of figures

Figure 1 View inside a Mode Stirred Chamber .....	19
Figure 2 Simplified illustration of the core of a Genetic Algorithm.....	27
Figure 3 Genetic Algorithm expressed as a flowchart .....	28
Figure 4 Particles searching a "solution landscape" for the highest fitness .....	32
Figure 5 UML diagram showing what a particle and its swarm contain. ....	32
Figure 6 Proportional cumulative count of convergent solutions as a function of the number of breeder/population size for three values of population size.....	47
Figure 7 Solutions for a shielded cable (including non-convergent solutions) for 100Ω operation showing the design variables 'p' and 'q'. ....	48
Figure 8 Reproduction of the previous figure, but only including convergent solutions.....	48
Figure 9 Statistics from GA to optimise a simple twisted pair cable.....	50
Figure 10 Colour map for fitness functions for cable design (if viewed in colour, red = impedance, green = return loss, blue = attenuation). Fitness is proportional to colour intensity. X-axis is dielectric thickness, Y-axis is conductor separation in metres .....	51
Figure 11 Colour map for fitness functions for cable design (if viewed in colour, red = impedance, green = return loss, blue = attenuation). Fitness is proportional to colour intensity. X-axis is dielectric thickness, Y-axis is conductor separation in metres .....	52
Figure 12 The basis of the mathematical model of the UTP cable .....	53
Figure 13 Fuzzy fitness function, to push the GA towards 100 Ohms .....	54
Figure 14 Fitness function for attenuation .....	55
Figure 15 Fitness as a function of phase delay .....	57
Figure 16 Impedance sensitivity with number of wires in each belt as a function of frequency. ....	61
Figure 17 Visualizing the quality of PSO solution as a function of number of particles in a swarm and the first best iteration. ....	62
Figure 18 A heterogeneous network of machines.....	64
Figure 19 A "tuplespace" with various kinds of data in it. ....	68
Figure 20 Workspace divided showing (successive) regions which cannot be solved without connectivity.....	70
Figure 21 A potential Interprocessor boundary problem. A, B, C, D represent individual processors.	73
Figure 22 Four processors with planar boundaries .....	74
Figure 23 Time domain results for setup 1 .....	76
Figure 24 FFT of output from setup 1.....	76
Figure 25 Results from setup 2 .....	77
Figure 26 FFT of results from setup 2 .....	78

Figure 27 Time domain results from setup 3 .....	79
Figure 28 Scanning an ellipse "radially".....	86
Figure 29 Raster scan to cover an ellipse.....	86
Figure 30 The ellipse equation $z = x^2a^2 + y^2b^2$ used as a distance metric .....	87
Figure 31 input 1 .....	95
Figure 32 Input 2.....	96
Figure 33 DC for input 1.....	97
Figure 34 A slice through the DC for input 1 results to show scale .....	97
Figure 35 DC for input 2.....	98
Figure 36 DC results for Input 2 (slice) .....	98
Figure 37 Low frequency part of input 1 .....	99
Figure 38 Low frequency part of input 1 (slice) .....	99
Figure 39 Low frequency part of input 2 .....	100
Figure 40 Low frequency part of input 2 (slice) .....	100
Figure 41 High frequency part of input 1 .....	101
Figure 42 High frequency part of input 1 (slice).....	101
Figure 43 High frequency component of input 2.....	102
Figure 44 High frequency part of input 2 (slice).....	102
Figure 45 Pointwise ADM for inputs 1 and 2.....	103
Figure 46 Pointwise ADM for inputs 1 and 2 (slice).....	103
Figure 47 Pointwise FDM for inputs 1 and 2.....	104
Figure 48 Pointwise FDM for inputs 1 and 2 (slice).....	104
Figure 49 Pointwise GDM for inputs 1 and 2.....	105
Figure 50 Pointwise GDM for inputs 1 and 2.....	105
Figure 51 ADM confidence histogram for inputs 1 and 2 .....	106
Figure 52 FDM confidence Histogram for Inputs 1 and 2.....	106
Figure 53 GDM Confidence Histogram for Inputs 1 and 2 .....	107
Figure 54 Final figures of merit for ADM, FDM, GDM for inputs 1 and 2 .....	107
Figure 55 ADM Results for 1D FSV from the original 1D data.....	108
Figure 56 FDM Results for 1D FSV from the original 1D data .....	108



## Table of Abbreviations

Abbreviation	Meaning
ADM	Amplitude Difference Measure
FDM	Feature Difference Measure
FSV	Feature Selective Validation
GDM	Global Difference Measure
ODM	Offset Difference Measure
GA	Genetic Algorithm
PSO	Particle Swarm Optimisation
RMI	Remote Method Invocation
TLM	Transmission Line Matrix (method)

## Table of Symbols

$g_{best}$	Global best position in PSO
$p_{best}$	Best position for a particular particle in PSO
$rand(\cdot)$	Random number between [0,1]
$Z_t$	Surface Transfer Impedance

# Acknowledgements

I would like to thank my supervisors, Alistair Duffy (First Supervisor) and Chris Oxley (Second Supervisor) and Amar Aggoun (Second supervisor while he was at De Montfort University) for their support, encouragement and friendship.

I would like to thank De Montfort University for granting me permission to pursue this research as part of my Staff Development.

I would like thank my family for their support and encouragement and for letting me drop out of their lives for large amounts of time.

# 1 Introduction.

## 1.1 Numerical Modelling in electromagnetics.

Mathematical modelling in the physical sciences has a long history, going back at least to Galileo and his study of the pendulum, but for many problem domains analytical techniques become intractable. Numerical approximations are better suited to computers, which, although capable of symbolic computation, are designed for numerical work and can display numerical results more usefully than symbolic ones. Many texts that discuss numerical methods do not even define numerical modelling itself, or describe its invention, which demonstrates to some extent how fundamental the idea is [1] [2] [3]. Techniques used in numerical methods such as Gaussian elimination clearly go back as far as Gauss, thus pre-dating the electronic digital computer. The Oxford Dictionary of Mathematics [4] lacks an entry for it, and whilst The Oxford User's Guide to Mathematics has a chapter (Chapter 7) about numerical methods, it does not attempt a definition, but prefaces the chapter with biographical details of Gauss's numerical feats [5]. Since such authorities eschew a definition of numerical modelling, a description of it should suffice to establish what is being discussed.

In attempts to get to grips with problems in the physical world it is useful to simplify the situation using a model that only reflects the aspects of it in which we are interested. This is the traditional, reductionist basis of science. Probably one of the simplest models one might consider is a mass on a spring, which is given the simplification that the spring follows Hooke's law and does not stretch. This is simple enough to be solved analytically, and is shown to be an example of simple harmonic motion. But it is also possible to treat the forces acting on the mass as numerical quantities and "solve" the system, or simulate it, by computing what acceleration the mass experiences, and its resulting position at intervals of time fractions of a second apart, and thus determine from the new position of the mass, what the forces acting in the system will be at that time. An advantage to the programmer of this approach to problem solving is that the mathematics is simplified, from symbolic computation, usually to linear operations.

In modelling electromagnetic systems we are interested in the distribution of charges and fields throughout a space, and current distributions in wires and on surfaces. The numerical techniques used in electromagnetic modelling and simulation are frequently derived from Maxwell's equations, and either use differential or integral methods to solve them. This work concentrates on the Transmission Line Matrix (TLM) method, in which discretised space is treated as interconnected transmission lines, with properties chosen to reflect the media being modelled, with the discretisation chosen to be fine enough that each part of the space may be considered uniform in its properties, and thereby minimise errors due to that discretisation. Whilst each of the different electromagnetic modelling techniques has its own strengths and weaknesses, TLM handles continuous metal surfaces well, rather than having to simulate them as a mesh of wires, and is known to be stable [3].

## **1.2 Difficulties with Existing Techniques**

In performing a simulation of a system, the individual operations are mathematically simple, but because of the complexity of structures of interest to engineers, these simulations involve a lot of such operations for one iteration of the algorithm, and require many iterations to complete a full simulation. Typically, these simulations can take time of the order of days. If exploring how a design will operate takes a day or two, then using repeated design iteration to improve and correct such a design to a point where it is marketable will take a long time, weeks, maybe even months. The increasing speed of computers over the past few decades helps to some extent with this problem, except that increased computational resource usually results in increased desire for more detailed models.

This may prompt the question, "Why model systems at all, then? Why not build and test instead?" There are substantial time and cost implications to such an approach, in terms of materials, inventory, manufacturing time, job shop scheduling, which escalate for complex projects. Also, for particularly large projects, it is much cheaper to throw away six computer models of an aircraft carrier than six aircraft carriers.

## 1.3 Components of Particular Interest.

In terms of the components analysed, the focus of this thesis is communications cables. With Ethernet speeds now into the GHz range, and with the adoption of VOIP in the telephone network, copper cabling is going to remain an important element of the communications infrastructure for the foreseeable future. The communication cable industry is, historically, conservative in its approach to design, using tried and tested methods to produce new products, and this is at least partly due to the mathematical modelling of cables being nontrivial. Yet cable technology still remains competitive with optical fibre, despite increasing bandwidth requirements. *“It has been anticipated that fiber cabling’s superior performance (bandwidth) would be needed in place of copper cabling’s limited performance. Every time, the market has increased its requirements, from 1 Mbps, to 10 Mbps, to 100 Mbps, to 1 Gbps and now to 10 Gbps, it was expected that fiber cabling would be needed. But every time, copper cabling has increased its performance to satisfy the higher bandwidth requirements.”* [6] The majority of cables must by necessity be flexible, and so the insulators and screens will conform during manufacture, and use, to shapes which are mathematically difficult to describe and analyse. Even with simplifying assumptions, parameters such as twist length in a twisted pair cable holding 4 twisted pairs gives a sufficiently large number of configurations to investigate to be both academically and commercially.

## 1.4 Optimisation

The principle strategy for reducing the amount of time people spend making decisions in the design process is to use optimisation techniques. These entail a search over a notional space for some best value (optimum) of some property. The property is usually the performance, or value for money, or value for some other cost (such as effort). The search space is determined by the parameters to be varied in choosing the proposed solution. For example, a multi-element antenna may have parameters that include the dimensions and spacing of the elements. The assumption that the elements are mounted in similar orientations, centred on a boom, might be abandoned by allowing position and orientation to vary as well. Materials for the elements may also be varied (on the basis of weight, or cost). Thus the parameters may be continuous or discrete, and there is a certain amount of freedom about which parameters can be explored. In simple problems there may well exist a single optimum that can be found. In more complex problems optimisation may be regarded as avoiding the most problematic

cases, or getting as good an answer as possible to the question put. The field of optimisation offers many procedures as a means to achieve this, all of which may be considered algorithmic.

If a function is well-behaved and can be differentiated a purely analytical approach may be possible, in which locations in the solution space where the derivative is zero are chosen and evaluated. Often the complexity of the function precludes this, or the evaluation of the performance of a system depends on, for example, simulation. In this case a purely analytical approach is impractical. If the problem is simple enough then linear programming may be used. Nonlinear programming is also a possibility, which extends the idea of a set of linear constraints to a set of constraints which are nonlinear, and where methods of solution may depend on an analytical solution of partial differential equations, gradient descent methods or on linear approximations (cutting plane methods) [7] [8]. However, conjugate methods for nonlinear programming involve expressing the problem in a linear form [9]. As a consequence, this is applicable only when the problem will yield to analysis, or, where it does not, this seems to offer little to deal with the problem of local optima.

Other techniques amount to exploring the solution space, treating it as a fitness or cost “landscape”. The ideas of fitness or cost are effectively opposites: the general aim is to maximise fitness and to minimise cost. So in terms of how one travels over a solution landscape, one can be discussed without lack of generalisation to the other. Considering a “fitness landscape” whose co-ordinates are determined by the problem parameters, and whose “height” is the resulting fitness, the fitness at any one point may be determined by calculation or simulation. However, unlike a real landscape, it is not possible to see the height of neighbouring points without calculating the fitness for each one. This may be likened to exploring a landscape blindfolded or in thick fog.

The first optimisation technique that this model implicitly presents is that of hill climbing [9] [10], in which points in the local neighbourhood of the best point reached are explored, and the climber moves to the best (highest) of those points. [11] makes a distinction between hill climbing and steepest ascent hill climbing; in the former, any point better than the present point is an acceptable place to move to next, whereas in the latter only the best of the nearby evaluated points is acceptable. Sometimes the search for a new point is done by evaluating

the points on a “simplex”, the simplest shape one may make in the given dimensionality (a line segment for 1D data, a triangle for 2D data, a tetrahedron for 3D data, etc), and repositioning the worst point as a result of the tests [12]. This technique has the weakness that the climber may reach the top of a foothill, whilst never reaching the mountain nearby. There are strategies to work around this used by practitioners of this technique, but other methods avoid this problem.

One way of overcoming this problem of local maxima is to use purely random search. Select points in the problem space at random and work out the cost or fitness for each one, recording the best so far. If the surface is basically smooth then this is a particularly inefficient way to explore it, because it takes no account of local information gained already about promising regions in the space. Its weakness is pointed out in [13], which provides Python code for this approach, and points out that this may be considered a baseline for other techniques. Since the systems of interest in computational electromagnetics consist of regions with smoothly varying, even constant, properties, a purely random process is inappropriate because it takes no account of local information. So whilst it is interesting as a theoretical baseline, purely random search may be rejected as impractical.

A strategy which uses this random noise to improve the search to start with is simulated annealing [13], [11]. This is loosely modelled on crystal growth in metals. A single point is chosen, and its fitness evaluated. Then, according to the temperature (which gradually cools as the process progresses) a nearby point is selected if it is better, or as an increasing function of temperature, even if it is worse. That selected point becomes the new focus of interest. As the system cools, the noise level and selection of worse points decreases, but in the early stages the selection of worse points allows the process to leave local minima (remember that in this model we are descending a cost landscape).

An alternative strategy, with a biological basis, is the genetic algorithm [14], [15]. In this approach, co-ordinates in the solution space are mapped to genomes of organisms in a population. This is often done by treating the variables of the problem as coordinates in a space and having the genome of an organism be a bit string representing one point in the space. By crossover (swapping of genes) and mutation (random alteration of genes), pairs of organisms breed, producing new offspring at new positions in the space, thus exploring the fitness of the landscape. By virtue of the “cut and splice” nature of crossover, and to a lesser extent, mutation, the offspring are not restricted to the localities of their parents. This avoids



the problem of being stuck in a local maximum fitness, missing the global maximum as a result.

A further biological model based on the swarming of insects, or the flocking of birds, is particle swarm optimisation. In this approach a number of particles are scattered in the landscape, and each evaluates the fitness of its position. They then move around the landscape, the motion partly random, partly in the direction of the fittest place found so far, and the whole system is dynamic, the swarm changing direction as fitter places are found.

A variation on the theme of Particle Swarm Optimisation based on ant foraging was considered, but not implemented in this research (principally because it seems to be very much agent based, which was a level of abstraction too far, and the ants need a home to return to, which did not seem to fit the system being examined). In this approach each ant explores the landscape leaving a pheromone trail. If it finds an existing pheromone trail it will be likely to follow it. When it finds a “food source” (a particularly fit region of the landscape) it returns to its start, strengthening its trail. As it returns several times the trail gets stronger so that other ants will tend to follow it also. This approach has been successful in controlling telephone networks [16].

In all of these methods, there is an element of randomness in the search. As a consequence there is no guarantee that even if a global optimum exists, it will actually be found. Also, unless the random number generator uses the same seed to start with, then when the process is repeated, the results may be completely different, although trials of sufficiently long duration should, statistically, produce similar results..

So, there are a large number of decisions to take before optimisation can begin: which parameters to include; how much they may vary; their representation; how the optimisation will be carried out; and how long one is prepared to wait for an answer.

The other substantive aim of the research was to speed up the simulation. Most approaches attempting this use supercomputers, multiprocessor systems or blade servers, but it was

intended to use available computer power, given that so many machines in labs and offices are not used for a full 24 hours each day, so an approach splitting the numerical modeller into pieces to run on many machines was adopted.

## 1.5 Outline of the research

The research described in this thesis is aimed at tackling the aforementioned problems of simulation time and removing decisions by people from the design-test cycle. After a TLM solver had been created, investigations proceeded into how to take advantage of available computing power from unused computers, and a distributed TLM modeller was designed. This was shown to work across a number of workstations. The lack of tools to validate the modeller was a stimulus to further research.

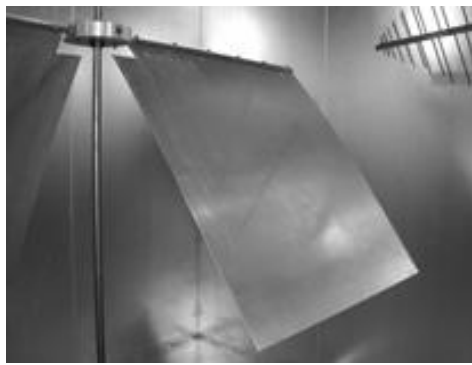
Work was also carried out on the design of cables, in which desirable properties were specified. The possible designs were explored using algorithms from within the discipline of Artificial Intelligence, and these were shown to produce sensible and useable results. However, in so doing, problems were encountered in comparing designs, and this raised the question of what constitutes a good design, especially when many variables are involved, particularly when they conflict. For example, a thicker conductor, will, within the limits of the skin effect, have a lower resistance, and so increasing conductor thickness will tend to reduce return loss. However, copper prices have risen significantly over the past decade or so, and as a result this would significantly increase the cost of a cable, making it uncompetitive against those from other manufacturers, ignoring for this instant Standards compliance. This leads naturally to the idea that some designs may be impossible to optimize, there being no global optimum, and to the investigation of the idea of satisficing [17]. It also leads to the investigation of extending the Feature Selective Validation techniques beyond one dimension.

The structure of the thesis is that the above discussion is expanded in Chapter 2 as part of the literature review. Chapter 3 presents the work on optimization, Chapter 4 presents the work on the parallelization of TLM, and Chapter 5 presents the work on extending Feature Selective Validation to two dimensions. These three themes are drawn together and analysed in Chapter 6, and possible future developments are discussed in Chapter 7. The References are collected together in Chapter 8.

## 2 Literature Search

### 2.1 Introductory Remarks

This thesis is concerned with the development of a toolkit for the optimisation of the physical layer of communication channels, in particular the modelling of communication cables, although the techniques are applicable to structures such as antennas.



**Figure 1 View inside a Mode Stirred Chamber**

Much of the testing carried out in electromagnetics takes place in various kinds of chamber designed to isolate the test apparatus from environmental electromagnetic noise. These chambers may be anechoic, where the walls are designed to absorb radiation (with a metal exterior to act as a Faraday cage to shield the apparatus from external fields), or they may be a room with conducting walls which reflect the radiation internally. This will naturally lead to resonances from the multiple reflections in the structure, and there can thus be modes in the chamber (which at a superficial level may be considered analogous to vibrations on the skin of a drum, or on a string in a musical instrument). These nodes will have nodes and antinodes where there are weak and strong electromagnetic fields set up at different places in the chamber. This makes analysis of the response of a piece of equipment rather difficult, and it would be much better if the average field across some “working volume” in the chamber were as near to uniform as possible. This ideal can be approached by “stirring” the modes using a

metal structure, the stirrer, which can be moved (usually rotated). This can be under continuous rotation, in which case the modes are said to be stirred, or the stirrer can be “stepped” into a sequence of positions, in which case the modes are said to be tuned [18].

The behaviour, characterisation and optimisation of such chambers are also topics of interest. The validation process itself is also a topic of this thesis.

The creation of tools for physical layer development requires the convergence of a number of topics. These are optimisation techniques (genetic algorithms and particle swarm optimisation) and distributed computing, validation and its formalisation in the form of Feature Selective Validation (FSV). This chapter provides a brief introduction to the state of the art in each, providing the necessary background for the more detailed discussions of the novel work presented in the rest of the thesis.

Simulations critical to exploring novel structures take a long time, and as machines become more powerful, the problems people wish to tackle become more complex and detailed, often consuming any gains won from improvements in computational speed. Nonetheless, for a problem of given complexity, a faster simulation will be useful, allowing more changes to be explored in a given timeframe, opening up opportunities afforded by optimisation algorithms in the design process.

Whilst there is little mention of this problem in the literature, some examples of the time taken for practical simulations include reference [19] mentioning simulations taking five or eight hours, meaning only one can be completed per working day, whilst reference [20] states times of 74.53 and 119.6 hours for the two longest simulations, and reference [21] gives “a few 10s of hours” for simulation time in one case. The most extreme case in reference [22] is Case 1 where the TLM modeller needed 677 minutes, more than 11 hours to simulate the problem, though Case 2 only took about half that. There is a competitive need for speed that must be met in any case, because in a commercial environment time to market is critical and reducing simulation times can be a factor in reducing this.

Thus, if the bottleneck were the simulation time, then decreasing that would allow more complex structures to be developed. This would fit in with the ideas of Goldratt about locating the constraint in a system and exploiting it [23]. The TLM algorithm is inherently parallel, so parallelization is an obvious way to speed things up. However, we then find that the bottleneck then moves from the simulation to the design process, because humans have to decide what to do with the results of successive simulations, and when simulations finish outside working hours the machines must wait for the human input. This is where an optimisation algorithm can help, because in algorithms such as the Genetic Algorithm, or Particle Swarm Optimisation, there is a strategy to proceed to successively better results given some measure of quality of the results obtained so far, and this strategy can be automated. This removes the need for human intervention. This is all contingent on the ability to measure the quality of results. Ideally we would like the machines to do just enough work to know if a comparison is favourable or not, and not continue refining the results if that decision can be made with the information available. This is the embodiment of “Satisficing”. To make a decision on whether a result is good or not is closely tied to the problem of validation. Validation is essentially a comparison exercise, between two models, or between a model and a physical experiment, or possibly two physical experiments, and this is founded on what is considered to be a correct result. Usually this decision is made by a person, but using the Feature Selective Validation algorithm we aim to automate this as well. These ideas are explored in the sections that follow.

## **2.2 Parallelisation.**

There have been several approaches to performing parallel computation, and with the availability of multi-core processors introduced in lieu of faster single CPUs this approach continues to be of wide interest. Not all of the approaches are applicable to the parallelisation of TLM, but they have qualities which may be of interest for parallelisation of optimisation algorithms, so they deserve comment even though they have not been used for the work described in this thesis. Some of the approaches are well known, or appear to be well suited to increased performance using the chosen language, so their rejection deserves some explanation.

Timing in simulations is discussed, and then the use of special hardware is covered. Well known libraries for parallelism are then discussed, and the applicability of parallelism to the various simulation techniques is discussed.

The approach to reducing simulation time taken in this thesis is that of developing a distributed parallel solver. This introduces issues with timing, especially when the machines have differing performance. These issues are dealt with in depth in [24], which draws several interesting conclusions for the works it surveys. There are observations about how time is handled in simulations, in particular whilst it notes that simulations are more commonly carried out using spatial parallelism, time parallelism is another possibility that is sometimes used.

Whilst time parallelism does not seem to be applicable to electromagnetics simulation where the output of one time step is the input for the next, at a larger granularity of time, a chamber used for a mode-tuned series of measurements (where the rotation of the stirrer is stepped to several separate positions so that the modes are allowed to settle, as opposed to mode-stirred operation in which the stirrer rotates continuously [25]) could be regarded as several identical chambers with the stirrers in different positions, operating in parallel. In this thesis only distribution by space is considered. [24] goes on to discuss 3 types of time in simulations: the time within the simulation - how much simulated time it runs for, how that time is represented, and real-world “wall clock” time. The relationship between simulated time and wall clock time determines whether a simulation is “real time” or not. It notes that if events are delivered from processors in the order the events are received, then this is different from ensuring that all events are delivered in the correct sequence for simulated time, and this impacts on how the simulation is managed. In the case where things are not kept in step with simulated time, then there may be different results arriving from different processors, out of order. This is similar to the problem of packets arriving out of sequence when using a UDP (Unix Datagram Protocol) connection. If events arrive out of sequence, so that one processor proceeds “into the future” on the basis of its available information, then strategies for correcting the results when they are found to be wrong are discussed. This usually involves rolling back to some checkpoint, and then calculating forwards again, though a few systems are able to reverse the computation from where they reached, and re-compute the correct results from the point where things were correct. The “Recent Advances” section (4) of [24] is of particular interest. It states that recent work makes use of a task pool, rather than explicitly specifying which processor will do which job. (This is the approach taken in the

work described in this thesis.) It notes that parallel distributed processing can be useful on systems with only one processor, in particular, that the use of many priority queues for events to be processed improves caching, and the use of aggregate events (where two events known not to conflict are handled together), and pipelining, where processing of an event occurs on the tail of processing another event, without the use of the priority queues at all. Also discussed in reference [24] is the use of the internet to create parallel machines, This theme has been adopted by the work reported in this thesis. Fault tolerance and security is discussed, considering what happens when processors fail. The system described in this thesis will survive destruction and/or switching off of processors other than the one which collects the results. The point is raised that a model shared over the internet by the machines is no longer secret. This thesis discusses this problem, but an implementation of a cryptographically secure system was considered to be outside the scope of the work, for reasons to be discussed in section 4.5 Security Considerations. That people are using graphics cards for general computation is noted in 7.4.1.1 Use of GPU for parallel TLM .

The use of parallel solvers to reduce simulation time usually involves using special hardware, either dedicated super computers, vector processors, or networks of high power computers. This sort of hardware was not a resource available for this thesis, so an approach using existing hardware was chosen. Examples of the use of special hardware include [26] which uses a DECmpp 12000 supercomputer as the basis for speeding up TLM through parallelisation.

There are a number of existing software platforms for parallelism, including Java's RMI (Remote Method Invocation), MPI (Message passing interface) and PVM (Parallel Virtual Machine) used by [27] for 2D TLM simulation of a waveguide. Both MPI and PVM use an external library, which breaks a constraint for the design of the distributed solver described later described in 4.1 Introduction and Design Criteria .

For the Finite Difference Time Domain method, there is more work on parallelisation.

[28], [29], and [30] use PVM for this purpose, and there are many using the MPI library [31] [32] [33] [34] [35] [36] [37] [38] [39] [40] [41] [42] [43].

For the Method of Moments [44] uses Java's RMI mechanism, or CORBA as an alternative.

The Boundary Element Method has been parallelised for mechanical engineering applications [45].

The work of Tomson [46] using Ruby for distributed computing appeared to be an obvious choice for implementing parallel TLM when it was announced. Further investigation showed that it was unsuitable because it assumes that tasks will always complete successfully. A requirement imposed by the local usage conditions was that machines may become unavailable at any time. This is because the people from whom they would be borrowed may need their full capacity at unexpected times. Also, there was no facility in Tomson's system for machines to compete to complete work. This is needed because as the tasks are completed, faster machines may run out of work and thus be able to take jobs allocated to slower machines, and complete them before the slower machine has finished the remainder of the job. If the CPU power is available, it is better to use it and risk some duplication of effort from the machines than not. Had these constraints not been violated by Tomson's system, it would have been sensible to use it rather than spend time developing something essentially similar.

## 2.3 TLM.

Transmission Line Modelling or Transmission Line Matrix modelling is a time domain method which treats space and volumes of material as a collection of transmission lines through which electromagnetic waves travel, as voltages propagating through the network. The electrical properties of the transmission lines are chosen to produce propagation effects similar to the corresponding materials being modelled. The model is constructed of a series of nodes, each representing a volume of space which is sufficiently small to be able to be considered to have uniform electrical properties. This node, which is usually cuboid, accepts incoming voltages on each face, and these are "scattered" to the other faces of the node through the internal transmission lines, producing new voltages on the faces of the node. There is then a "connect" phase, in which the voltages on the faces of a node propagate to the adjacent faces of its neighbours, and according to the differences between the properties of the nodes, there may be some reflections, as there are at impedance mismatches between joining transmission lines. This is discussed in more detail in **Error! Reference source not found..**



The TLM solver itself started out as an implementation of nodes with properties for Free Space. The theory for TLM is described in [3] and there are a plethora of enhancements to it, including perfectly matched layers [47] [48] [49], wires [50] [51] [52] [53], lumped elements [54] [55], variable mesh, multigriding [56], and many different ways of constructing the nodes (Symmetric Condensed Node, Adaptable Super-Condensed Node, Symmetric Super-Condensed Node...) [57] [58] [59] [60].... The solver developed for this thesis included the Super Symmetric Condensed Node.

## 2.4 Optimisation

Given that simulations may take longer than a working day to complete, and thus finish at times when there are no staff to react to the results, it is desirable to automate the decisions involved in the design process as far as possible, so that the “constraint” of the computing resource can be exploited fully. Some means of comparing results to establish which ones are better, and a way to progress towards still better results is necessary. Two schemes for achieving this are the Genetic Algorithm and Particle Swarm Optimisation. These are the ones explored in this thesis. Other possible ways of proceeding include Simulated Annealing, Tabu Search, Hill Climbing, and as a last resort, random search. Random search is essentially complete guesswork, with no memory of previous results, and as a result is usually considered too inefficient to be worthy of serious consideration, although has been found useful in problems with high dimensionality [61], and recently has shown some utility in software testing [62].

Hill climbing which uses gradient ascent to reach a maximum solution of some function is hindered by local maxima. The algorithm may find the top of a “foothill”, but not the nearby mountain. Simulated Annealing overcomes this with perturbations provided by a Boltzmann “heat source” which gradually “cools”. It has been used in the design of communication channels [63] [64] [65]. Tabu Search is much like hill climbing, in that it is a local search strategy, but instead of always ascending, it always takes the most uphill direction, even if that is merely not as far downhill as the other directions. The reason this might be less than the local optimum is because of the Tabu list. “Tabu” is another way to spell “taboo”, and this list is the collection of points not to revisit. This facilitates avoiding loops when

searching graphs, and allows breaking away from local maxima, a problem with basic hill climbing. This use of a tabu list and the idea of taking the best available choice to progress, means that tabu search is not tied to hill climbing, and may be used in combination with other searching strategies such as genetic algorithms or simulated annealing [66] [67] [68] [69].

The Genetic Algorithm controls the progression towards solutions which are better survivors using a model of “natural selection”. The measure of quality is considered to be a “fitness function” which may give an absolute value for each potential solution, or sometimes the possible solutions compete in “tournament selection”, as animals compete for food or dominance and thus the right to breed. The next generation may entirely replace the current population in a “generational” scheme, or some individuals, usually the worst individuals are replaced with “offspring” created from selected parents while most individuals live on to the next iteration in an “elitist” or “steady state” strategy. The term “elitist” is used because the best individuals will survive into successive generations, whereas in the generational arrangement, they will not, although their children will. The selection of parents is based on fitness. It may be purely by rank, by a weighting system based on fitness (a roulette wheel, where the width of the slot is proportional to the relative fitness of the individual) or by tournament selection (where a selected group of, possibly only 2, individuals selected randomly compete to be parents). [70]

This breeding process is modelled on sexual reproduction at a genetic level, so components from the “chromosomes” coding for parent solutions are combined to create the offspring. This is done using “crossover”, where the chromosomes are aligned, cut at one or more randomly chosen points, and some of the pieces swapped over to create new chromosomes containing parts from each parent. This process is shown graphically in Figure 2.

## Genetic Algorithm

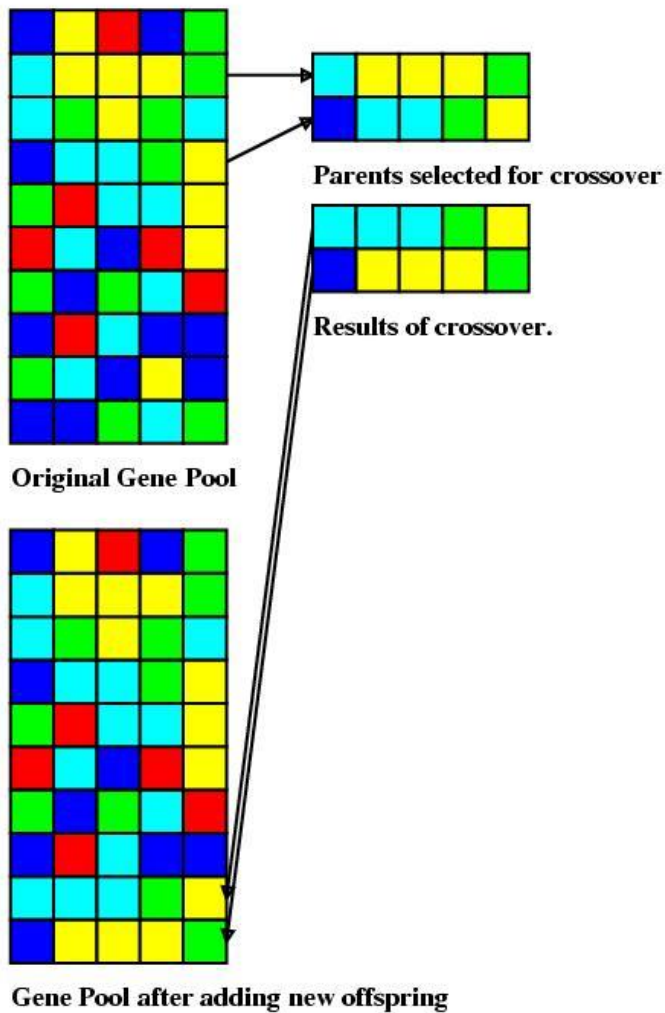


Figure 2 Simplified illustration of the core of a Genetic Algorithm

When the chromosomes are bit strings, (sometimes they are taken as decimal numbers) then crossover can result in a gene representing some characteristic gets split, getting low bits from one parent and the high bits from the other, allowing the value to jump to a completely new place in the solution space. Mutation, at the low rates of about 1%, is not a major contributor to this movement, though it does help. The Genetic Algorithm may be expressed as a flowchart as in Figure 3.



**Figure 3 Genetic Algorithm expressed as a flowchart**

Considered as exploration of the solution space, it is clear that the successful propagation of genes increases the population in some area or volume of the solution space, and the crossover mechanism will allow some dimensions to be explored in greater depth. The factors affecting the performance must therefore include how well the chromosomes get mixed (which is related to the population's diversity) and how quickly the populations in the good parts of the solution space get built up, which is related to the selection of parents, the selection of which individuals do not survive, and on how long the good individuals live.

The selection of better individuals and the (gradual) replacement of below average survivors results in an increasingly fit population over time. At the most basic, this was effectively demonstrated in Donald Michie's MENACE (Matchbox Educable Noughts And Crosses Engine) [71], in which each configuration of the noughts and crosses ("tic-tac-toe" in the USA) board was represented by a matchbox, and coloured beads in the box represented possible legal moves. Play was determined by selection of a bead at random from the appropriate box. If the game were lost, the selected bead would be removed from each box. If there were a win, a bead of the same colour as the winning bead would be added to the box. (A draw may be taken as a win if it is against an "expert", and the "credit assignment problem" features, because it is not clear how much the early moves contribute to winning the

game.) Indeed, he characterises it as Darwinian evolution of organisms inhabiting each matchbox (regarded as a habitat) in the system. If the selections made during each game are considered to be a complete chromosome, it looks more like the genetic algorithms as they are described now, only with the complete solution space covered and the use of an elitist strategy in that the good solutions survive to the next generation implicitly. This system lacked crossover and mutation, but covering the whole solution space already, it would not need them. Darwinian evolution is often mischaracterised as progression towards some particular better thing, some “higher form”, whereas in reality it only produces better survivors from the available stock. There is no goal as such. It is the job of those applying genetic algorithms to ensure that the means of determining fitness expresses what they mean by a good solution to the problem, so that the long term result of this selection process is fit for purpose.

The fact that the algorithm has these variants (how successive generations are created by complete replacement or partial replacement, how the selection of parents is carried out, and how individuals are chosen to be eliminated from the next population) demonstrates that the algorithm is very robust, and over many generations individuals fit enough for the problem at hand will be found, almost despite these variations. Because it does not rely on local fitness gradients it will cope with the optimisation of non-differentiable functions, and move towards a maximum that is globally good, if not the best solution to the problem being explored.

The use of genetic algorithms to design the shape of Radio Frequency absorbing material in a screened room is discussed in [72] [73], with the novel aspects being particularly that the material is modelled with the metal backing actually used (the walls of the chamber to block out external RF emissions), and that oblique radiation is considered. Reference [73] also discusses the use of uniform crossover, whereby the way crossover takes place is a uniform probability of each bit being swapped, rather than a choice of a number of places to cut the chromosomes before crossing over. The point about multiple frequencies being expensive to simulate could be overcome with a time domain code, provided that a sufficiently fine gridding is acceptable. Fine gridding will lead to a long simulation time, which may not be acceptable.

The use of a genetic algorithm for the design of a stirrer for a reverberation chamber is discussed in [74]. It notes that the low frequencies are important because that is where stirrers are poor. Earlier remarks about long simulation times in electromagnetics research are confirmed by the stated 42 hour simulation time for each stirrer design and that this was overcome with another way of modelling them. They conclude that more complex stirrer designs are better, as well as larger ones. The genetic algorithm itself was of steady state design, using tournament selection which was chosen for speed of convergence.

The use of modified GAs to explore the design space is discussed in reference [75]. From the point of view of the algorithm, one novel contribution is the idea of treating the chromosomes as circular so crossover can occur between any parts of the two chromosomes. The paper is unclear as to whether this actually improves the efficiency of the GA. It would seem to imply that good genes would not necessarily cross over with the corresponding good genes in the other parent, making selection weaker. The paper discusses other variations on the design of the algorithm including variable mutation rate, continuous rather than discrete genes, and the generation of an initial population using a genetic algorithm. The conclusion does not fully explore how these experiments turned out but its strength is in addressing non-standard ways of addressing the solution to these problems.

The use of decimal numbers to represent the genes is explored in reference [76], and performing crossover and mutation in that space, rather than in the space of binary numbers for each gene. The results show better convergence and better results overall for the fixed number of generations used, which was 50. Given the long simulation times experienced for a significant number of applications in electromagnetics this technique would prove useful. The operators for crossover and mutation are interesting; crossover is regarded as a perturbation of a gene, and thus can be modelled by operations on the parent genes  $P1$  and  $P2$  to give the child genes  $C1$  and  $C2$ , by  $C1 = (1 - F) P1 + F P2$  and  $C2 = F P1 + (1-F)P2$ , where  $F$  lies in the open interval  $(0,1)$ . If the interval were closed, the 1 and 0 cases would result in no crossover. Mutation is performed by adding or subtracting some factor of the range to that parameter or gene. A further advantage of this technique is that there need be no conversion between binary and decimal forms of the genes, which should be a performance improvement.

There has been some work on distributed computing with genetic algorithms. [77] is a system for Unix workstations, written in C++ with the parallelism modelled on that of the Transputer on the basis that “OCCAM is notably the first language to be developed explicitly for parallel processing”. Occam seems to date from 1982 [78], but reference [79] describes Ada with its rendezvous communication mechanism in 1981. The genetic algorithm has a number of parts which are not explicitly serial, and can thus be parallelised, for example the computation of fitness prior to roulette wheel selection. The paper discusses the use of several sub-populations evolving separately on different machines. The underutilisation of computer resources outside office hours is also mentioned as a motivating factor in the design. The use of genetic algorithms to configure a reconfigurable patch antenna is discussed in [80]. The antenna consists of small metal patches connected by switches that will control the connectivity between them. The gain and power in a specific direction were optimised in this case.

It is clear from this collection of papers that much of the research is in the design of antennas, some on the design of screened rooms and almost none on cable design.

### **2.4.1 Particle Swarm Optimisation.**

In the same way that genetic algorithms use a biological model to perform optimisation, Particle Swarm Optimisation (PSO) uses the collective intelligence of swarms or flocks as a biologically inspired model for exploring the solution space. Such a model has the appeal that people will have observed flocks of birds or animals, and how they move in search of resources, whereas evolution usually takes place on too long a timescale to be observed directly. Thus the particle swarm model of searching seems particularly intuitive, and can be visualised in the manner of Figure 4.

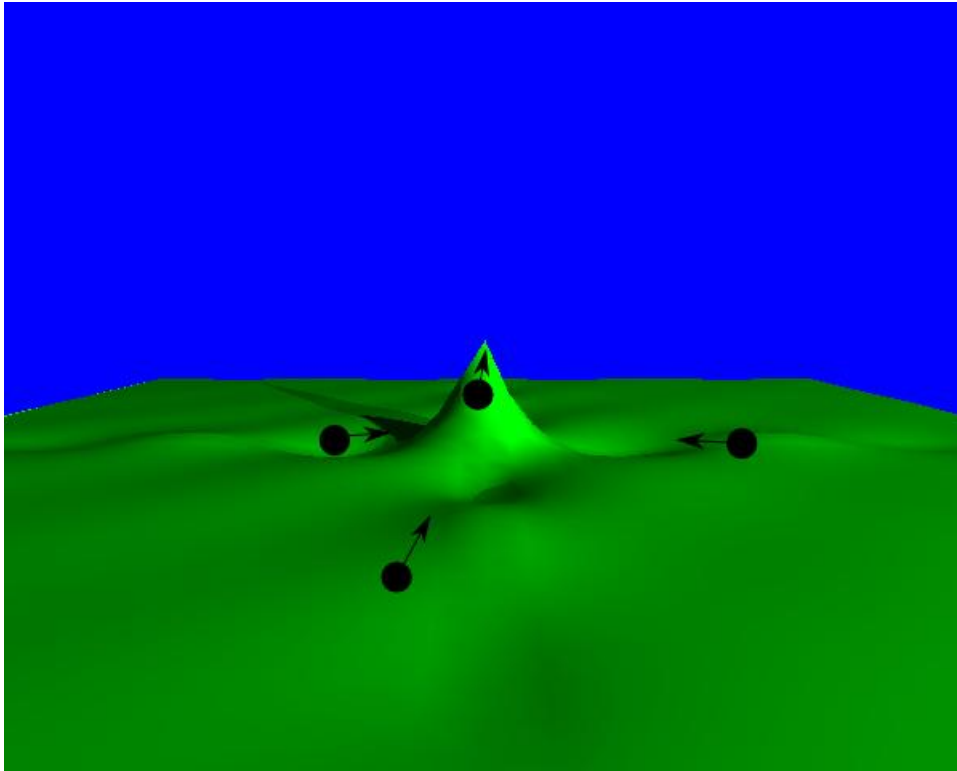


Figure 4 Particles searching a "solution landscape" for the highest fitness

The essential properties of the animal, insect or bird in the swarm are its position, its memory and its ability to communicate with the rest of the swarm. As the animal is reduced to a particle, so its memory is reduced to a record of the best place (as determined by some fitness function) it has visited and the corresponding fitness, and its ability to communicate with its neighbours is reduced to sharing information about its best position so that the location and fitness of the globally best position found so far may be calculated. This information is implemented as attributes of the particle, and of the swarm, and the communication takes places as method calls as shown in Figure 5.

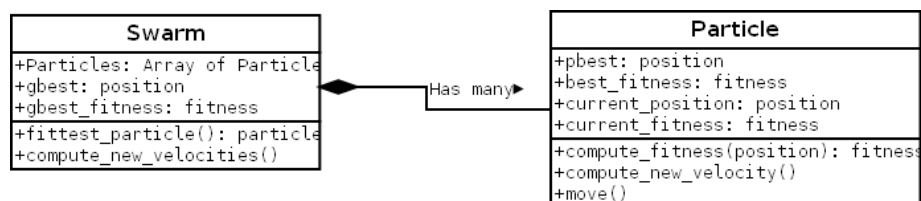


Figure 5 UML diagram showing what a particle and its swarm contain.

The particles are distributed randomly through the space, and each particle explores its neighbourhood. As it moves it has a random component to its movement, a randomly weighted tendency to return to the best place it found (called *pbest* in the literature), and a



randomly weighted tendency to head in the direction of the global best position (called *gbest* in the literature) for all particles. In this way the better parts of the space should be explored more thoroughly. The particle whose *pbest* equals the *gbest* is effectively the leader of the swarm until another particle finds a better place. Because each particle is influenced by the global properties of the swarm it should be able to avoid stagnation at, or premature optimisation to, local optima.

The possible weaknesses of PSO relate to premature convergence to an optimum which is global for the swarm, but which is not the global optimum for the solution space; and the possibility that particles will explore too widely, but as a result overlook nearby solutions which are better than *pbest* for that particle. Most of the adaptations to PSO are to address these issues.

Reference [81] showed that most of the use of PSO in electromagnetics was in the design of antennas. There was almost no work on communication cable design, and only one paper mentioning transmission lines. Work on PSO did not really pick up until about 2003, according to this source. Reference [82] also makes no mention of cable design, most of the cited applications being in the construction of neural networks, and various sorts of control and classification problems. As an example of the application of PSO to antennas, reference [83] is of interest because it compares PSO with genetic algorithms, using the problem of creating an antenna with a significant specific null in its radiation pattern to defeat interference during reception. The results show that the approaches of PSO and genetic algorithm are broadly comparable. Reference [84] explores biomedical image registration, using normalized mutual information optimised by PSO. Mutual information gives a measure as to how much Shannon information about one data set is contained in another data set. It is determined by

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \quad (2.1)$$

From equation 2.28 in [85] where  $p(x,y)$  is the joint probability of  $x$  and  $y$ , which may be obtained by considering the combination as a vector. It may be considered as the reduction in entropy of  $X$  as a result of knowing  $Y$ . This is of interest because it is similar to the validation problems faced in electromagnetics research in so far as the similarity of traces is of concern. The use of mutual information for such validation seems to be absent from the literature. Reference [86] discusses the use of PSO in the design of radio frequency absorbers for use in

anechoic chambers, whilst reference [87] discusses its use in the design of the membership functions of a fuzzy controller for traffic lights. PSO is used in [87] to define a neural network for image classification. The rest of the PSO literature surveyed here is concerned with the optimisation of mathematical functions, as benchmarks for development of the algorithms themselves.

There are a number of variations on PSO in the literature, intended to overcome the aforementioned problems. Reference [82] explores adding a local neighbourhood of  $k$  particles to the algorithm, to give a local optimum as well as a global one, and it also explores the use of an inertial weight (as described in [88]), and the use of a constriction factor. Inertial weights are used by [84], and it discusses the use of a neighbourhood based on a distance metric. It also discusses the application of crossover and mutation operators borrowed from evolutionary strategies and genetic algorithms, the use of subpopulations and constriction factors. Reference [86] uses Pareto optimisation for a multi-objective PSO. Pareto optimisation is based on the idea of dominance, in which one solution,  $a$ , dominates another,  $b$ , if there is no objective in  $b$  better met than in  $a$ , and there is at least one in  $a$  that is better met than in  $b$ . So at least one is better and none are worse. The Pareto front is the set of solutions that cannot be bettered in the sense of Pareto dominance. The PSO algorithm is further modified in that the space is divided into hypercubes, and those that contain more than one particle are given less weight in a roulette wheel to select a particle which will contribute to the velocity. This is intended to preserve diversity in the population in order to prevent premature convergence. The particles then undergo mutation: the details of which are not described in the paper. Reference [87] handles multiple constraints using a count of the constraints violated to measure the fitness of solutions that violate any constraints, thus providing direction for improvement. The constraints are relaxed as a function of time, as a means of preventing only local optima being found. Reference [89] uses a “chaotic wind” consisting of two components with random magnitudes, one in the direction of  $gbest$ , and one in the opposite direction. This is to prevent premature convergence. Reference [88] introduces the inertial weight into PSO. Reference [90] uses inertial weight decreasing with time, and also, when there is stagnation the velocities are reset to random values to reinitialise the process. Reference [91] uses crossover based on the idea of surrounding multiple points with a “convex hull” which is a convex shape which surrounds a collection of points, like a bounding box, or a bounding sphere, but which may be any shape without concavities. The new point generated lies within the hull enclosing the particle’s position, the position of  $pbest$  for that particle and  $gbest$ . A combination of PSO with tabu search is used in reference [92] to ensure that previously visited solutions are not revisited. The production of new positions

in PSO is framed in terms of crossover and mutation; it is in the mutation that this scheme uses the tabu search to avoid local optima. It suggests using one slave processor per particle, but because of the problems of communication that would result it actually groups a number of particles onto one processor. An adapted PSO algorithm is taken in reference [93] where the range that each particle may cover in its explorations decreases with time, and performs simulated annealing on each particle between iterations of the PSO algorithm. The reasoning is that simulated annealing is better at finding local optima, whilst PSO is better at finding global optima. The convergence is better in terms of the iteration count, but the wall clock time for convergence is not discussed. In reference [94] the whole population of particles is divided into subpopulations, and then the PSO algorithm is run on those. The *gbest* values from all of these are then added to the subpopulations for the next iteration of the whole process. References [95], [96] and [97] may be taken together. They attempt to improve the search by introducing more random numbers into the process, essentially. They use the concept of a cloud which is a set of points taken from a set of normal distributions whose means and standard distributions are themselves normally distributed. Each particle is assumed to be searching its locality and the clouds are used to determine the position and range of the search for that particle. Each particle's memory includes its best position, the global best position for the swarm, but also the best position of the group of  $r$  particles within the search distribution defined by the cloud. A Manhattan, or "City Block", distance metric ( $d(x, y) = \sum_i (|y_i| - |x_i|)$ ) is used in reference [98] to measure the distance between particles, the maximum distance being used to measure the spread of the whole swarm, and, if it is not very spread out then it is considered to have converged on a solution. This is used to detect premature convergence. Every  $k$  iterations of the PSO algorithm this metric is checked, and if the swarm is too clumped then the positions and velocities of the particles are re-initialized, but their memories (*gbest*, *pbest*) are preserved. There is also a scheme to mutate the position of *gbest* with a Gaussian noise source  $\sigma$  by setting  $gbest := gbest * (1 + \eta\sigma)$  with  $\eta$  being between 0.01 and 0.9, to drive the search out of local optima. With the random component of motion for the particles in the optimiser developed in this thesis, this was unnecessary. It is clear that there are a number of strategies for dealing with the limitations of PSO, and therefore further refinement of the optimiser described in this thesis is worth exploring.

## 2.5 Validation.

The “scientific method” is based on testing hypotheses and results. Validation is an example of this, showing that an experiment or model behaves as expected. Any method that will reveal defects in the model is useful, because progress is often founded on uncovering and removing hidden assumptions. However, there are practical limits on how this can be applied. Every model and variation of a model cannot be tested because this would eliminate any savings in time and effort that modelling is intended to give compared to that for building a suitable experiment [99] [100]. Modelling may show up aspects which cannot be inspected easily in a real experiment, because the distribution of currents cannot be seen, for example. So in some cases it may be useful to do both. But this still leaves the validation of the class of models which are used to replace the experiments in order to show that they will provide the required results.

As [99] and [100] point out, there are various problems to take into account when doing validation within the domain of electromagnetics. There is the modelling technique itself, and whether it represents the aspects of the model that are of interest. Depending on the application, that may be sufficient. There is the numerical stability of the modelling technique. There is whether the particular problem has been modelled accurately, including the gridding of the space in sufficient detail; this can be established by convergence testing. Comparison with other applicable techniques is also useful. And, intermediate results such as currents on wires may be used for validation, checking that they actually make sense in the context. Often this can be done in simulation more easily than by experiment.

For the experiment in which testing is done, there is some degree of measurement error. In EMC this can run to a “budget” which can reach values 10 dB, which can be achieved through about 4.5 dB in emission, and about the same noise floor in reception, together with any calibration errors and mismatches (from the antenna at the edge of its band, etc). See Table 6.5 “CISPR Uncertainties according to CISPR 16-4-2” in [101].

Essentially, the crux of the problem of validation is what do we mean by “good”? The ability to compare results and see which ones are good is fundamental to testing and developing any kind of design, and fundamental to Genetic Algorithms and other optimisation strategies.

For simulations where the requirements were a set of values all to be maximised, and where these do not conflict, the fitness models used for GAs and PSO are reasonably easy to implement, and reasonable to use in validation. When there are conflicts then there are issues of scaling, depending on the relative importance of the values, or some concept like the Pareto front may be used. When the problem is a frequency distribution to be optimised, as is the usual case in EMC and signal integrity problems, then the problem is more difficult. In such a case, the usual approach is that of using experience to decide whether something looks good. This has the advantage of utilising the skill of experienced engineers and their “tacit knowledge” to good effect, but it has the difficulty that agreement among a group of engineers can be hard to obtain. Tacit knowledge is that knowledge which is hard to put into words or numbers, the sort of knowledge that allows an experienced cook to know what “cook until done” means [102]. It is not too vague to be defined, it is simply difficult to describe, in the case of cooking because it will involve smell, and colour, for which we don’t have precise language.

This is a subjective process, and it involves having some background in the area in which validation takes place in order to interpret results correctly. In the same way that statisticians develop some “feel” for how good the “95% confidence level” is, so engineers working in EMC and related fields develop experience from looking at spectra whether two sets of results are close enough to be considered valid. Examining the literature examples can be found where the validity and the quality of results is simply asserted with no more than an appeal to visual inspection to justify it [103] [104] [105] [106], because the intended audience of peers in the field will know what one is talking about, so anything more might be considered to be labouring the point. However, it has been observed that sometimes the figures quoted verbally in workshops and conferences for what is good enough can vary between 3dB difference to 10dB difference, and clearly since repeatability is fundamental to the scientific endeavour, it would be useful to have a standard people can use for comparisons, if only so one can reduce the variability in how results are reported. Reference [107] reports that the visual rating scale in FSV achieves this, and 1-dimensional FSV is now an IEEE standard. .

When modelling a mode-stirred reverberation chamber to optimise its design, the field distribution over the enclosed volume will be important. When establishing the working volume of a mode-stirred chamber, considering peak fields, stirring ratio, etc, the dimensions will include the stirrer position, ratio, and peak field strength, as well as the fields, because

these are variables in the problem space. In such circumstances, simple validation is no longer possible, because it becomes difficult to perceive an  $n$ -dimensional display, or to create one for that matter. There are ways to represent higher dimensions on a 2 dimensional display, but even the interpretation of simple high-dimensional figures, such as the hypercube, is not immediately intuitive. The Feature Selective Validation (FSV) technique, which this thesis extends to two dimensions, appears to be extensible to higher dimensions, and may assist with this problem, but to ensure that engineers treat high-dimensional spaces in the same way as the algorithm will entail research into display techniques.

### 2.5.1 Validation and Feature Selective Validation

The essence of Feature Selective Validation (FSV) is a method for performing comparisons automatically which broadly agrees with what engineers would decide, and is repeatable. Within the body of work on FSV is an algorithm which people can apply to data sets to perform the comparison themselves, and if this is used by most engineers it will agree with the computed comparison. The work described in this thesis concentrates on extending it from one dimensional plots  $y = f(x)$  to two dimensional plots  $z = f(x, y)$ .

There are advantages to the automation of the decision process. Firstly, the decision becomes repeatable, regardless of psychological state of the person performing the experiment. Secondly, the integration of an automated decision process into the design process allows progress to be made outside working hours, because human intervention is reduced. Where the configuration of an antenna is altered electrically, as was done in reference [79], the whole experiment may be run according to a genetic algorithm, or similar optimisation procedure, with fitness decided using FSV, but this is more practical in the electrically alterable case than in the case where rearrangement of components occurs physically.

Automated comparisons in EMC work have been attempted prior to the development of FSV, but the approaches that are usually suggested for comparisons have been found to be ineffective for various reasons. Reference [108] points out that correlation usually only yields one number, which is inadequate to describe the complexities of the result sets engineers

usually deal with. Also, using “least squares” to measure the error fails to convey any information about resonances apparent in the results. It is pointed out by reference [109] that the statistical assumptions of Pearson correlation, assuming the data comes from normally distributed data sets, is certainly going to be incorrect for much of the data of interest, but Spearman correlation which drops this assumption still only yields one number. The paper then explores the use of correlograms, which the paper refers to as correlelograms. This usage led to the discovery of reference [110], which may be of use in the display of 2 dimensional FSV displays. Correlograms are based on autocorrelation and cross correlation. Whilst these are more informative than just single numbers, the results are difficult to relate to physical reality, and are thus non-intuitive. This makes it unlikely that they will reflect the views of large numbers of engineers studying experimental results.

Reliability Factors which arose from X-ray crystallography and surface diffraction are discussed by references [111], [109] and [109]. They rely on taking a function of the signal, the signal’s derivative, and second derivative, integrated over a frequency range, all normalised to the integral of the signal over that frequency range. These components, and combining them in a similar way, turned out to be fundamental to the construction of the FSV method. The derivatives have physical significance as high pass filters.

An alternative approach, discussed in reference [112] as well as reference [109], is the Integrated Error against Log Frequency measure, based on the realisation that results are frequently displayed against log frequency, leading to a weighting in the results not normally affecting the computation of other comparison measures. This method handles noisy (“grassy”) data better than FSV, in which the derivatives used in FSV are larger than the actual features in the graph. It cannot be used with time domain data, or where the independent axis is spatially related, because of the log frequency requirement. The papers show it has good comparison to what engineers would decide in those cases where FSV breaks down. Clearly a full explanation of FSV is now in order.

## **2.5.2 The Feature Selective Validation Method.**

Supported by evidence of how people examine graphical data [113], the FSV method is based on separating out the trend information from the detailed information in two data sets. It is based on the usefulness of the reliability factors, and the equations are have similarities.

Reference [114] describes the procedure, see Appendix A – Implementation of FSV method (quoted from IEEE1597.1 “Standard for Validation of Computational Electromagnetics Simulation and Modeling” 2009).

The Amplitude Difference Measure and the Feature Difference Measure combine to form the Global Difference Measure. See also [115] for another extensive treatment of the FSV method.

There have been a few enhancements to the method, in particular the idea of a Global Difference Tolerance in [116] as a threshold for acceptable experimental repeatability. A recent development is the introduction of the Offset Difference Measure, which makes use of the DC data (the first 4 points which were removed to separate that information out), discussed in [117], and the introduction of the grade spread diagram in [118].

### **2.5.3 How was FSV validated?**

FSV is for validation. One may ask how it was validated itself. It is intended to produce results similar to a group of engineers, and there have been a number of studies where people have been given data sets to compare and the results tested against the results provided by FSV itself. [113] has already been mentioned, in which the design of the visual rating system was changed in the light of responses from engineers, and then re-tested to ensure then was an improvement. [119] has a more extensive survey. [120] surveyed 50 engineers and demonstrated the effectiveness of the system, showing that there was a smaller standard deviation among the results for good than for fair.

### **2.5.4 FSV as a tool in knowledge management**

FSV could be said to fill the role of making tacit knowledge explicit, in so far as the visual rating scale and the numerically computed measure quantify something which has hitherto been tacit knowledge, and in so doing make it explicit. It turns something difficult to measure into something which can be measured. As a result the knowledge can be transferred between



different groups of engineers or preserved for use (possibly by the same group of engineers) in the future.

It also may be considered in relation to fuzzy logic as providing membership functions for the different quality levels of the data comparison. Fuzzy logic's classic example is a qualitative judgement whether someone is tall, and yielding a degree of tallness, that is, a degree of membership of the tall category for that individual [121]. The shape of this membership function may be obtained by supplying different subjects to one judge, or it may gain a broader applicability by asking a group of judges to decide on the set of subjects, and aggregating their decisions, somehow. This should result in a membership function which can be agreed on by more people than could a membership function decided by one person. The parallel between such a process and the decision whether a comparison is good seems to be quite striking. This is not a probability; it is not some chance that this decision is good [122].

## **2.6 Conclusion.**

The challenges faced by engineers using Computational Electromagnetics include the need for increasingly sophisticated designs, the pressure to reduce time for such studies, and to do so on a budget. Whilst there has been work on parallel solvers, much of it has been to use special hardware, or dedicated collections of computers, rather than to use existing machines outside of hours in a manner that is flexible and fits in with the needs of the primary users of those machines. In the drive to improve designs, whilst attention has been paid to genetic algorithms in the electromagnetics community, there seems to be little evidence from the surveyed literature that much has been done with genetic algorithms in the design of cables, and this is also true for particle swarm optimisation.

The problems of making a decision on whether results are good or bad continue to be an issue, and although FSV is beginning to gain acceptance, it presently only deals with one dimensional data sets, and the problems handled by engineers usually cover more dimensions than this. The original work on 2D FSV actually scanned the areas to make them into one-

dimensional problems. A fully fledged development of 2D FSV would be a useful step on the way to higher dimensions.

# 3 Optimisation of Cable Design

## 3.1 Genetic Algorithms and The Design of RF Structures.

### 3.1.1 Introduction

One of the themes of the research described in this thesis has been the application of “artificial intelligence” to the design of physical layer components in communication systems. The work concentrates on Genetic Algorithms and Particle Swarm Optimisation with a particular emphasis on the design of communications cables.

When designing a cable, there are a number of factors in the design process which often conflict. The electronic engineer will naturally consider the electrical properties to start with, such as impedance, return loss, and attenuation. These properties do not have simple linear relationships with the dimensions of the cable, and in themselves make a topic worthy of some study. The fundamental questions are “How can the designer optimise the cable performance given that the dimensions need to be ‘reasonable’? Can this be balanced with economic factors? Furthermore, can this be done automatically?”

The problem domain naturally extends itself to nonelectrical properties of the cable. For a person trained in electronics the mechanical properties are more difficult to tackle, but the financial constraints would seem to be realisable: is it possible to come to a good trade-off with cost when creating a design? Copper is probably the most expensive material in most cables, and minimising the amount of copper, provided the electrical properties are met would be desirable for any commercial organization. Then there is the question of manufacturing tolerances: how much precision can actually be required of a design?

Prior to the work described in this thesis commercial development of cables for communication systems was rather conservative. Most products were based on existing designs, with minor variations. Computational modelling, in particular computer generated design, was not a strong

feature of the industry. The work on cables described in this thesis was intended to bring some of the techniques from artificial intelligence into the electromagnetics community and to demonstrate their practical application. As described in Chapter 2 much of the genetic algorithm research in the field of electromagnetics had been focused on antenna design. It is believed that applying genetic algorithms and particle swarm optimisation to this area is novel.

### 3.1.2 Modelling the Electrical Properties of Unshielded Twisted Pair Cable.

Consider a cable which has 2 inner conductors, of radius  $d$ , the centres of which are separated by a distance  $s$ . The internal diameter of the outer screen is  $D$ . The impedance is given by reference [123]:

$$Z_0 = \frac{\eta}{\pi} \left( \ln \left( 2p \left( \frac{1-q^2}{1+q^2} \right) \right) - \frac{(1+4p^2)}{16p^4} \right) (1-q^2) \quad (3.1.2.1)$$

where  $p = \frac{s}{d}$  and  $q = \frac{s}{D}$  and  $\frac{\eta}{\pi} = 120\Omega$ .

Certainly automation of the design of a cable is possible, and the early stage of the research was to produce a program using a genetic algorithm (GA) to optimise these parameters. However, these results would need to be validated by some other means, in order to show that the genetic algorithm was functioning correctly. Given that there were three properties being optimised, it was possible to regard the problem as being optimisation of a 3-dimensional property over the space of cable dimensions. For unshielded twisted pair, the simplified model which did not take the twist length into account was used, and the dimensions of the problem space became the diameter of wire used and the distance between the centres. (Note that the solution space has three dimensions.)

Visual representations of the results requires the ability to combine three variables in a way that can be clearly discerned by eye. It therefore seemed natural to regard the solution space, having the two dimensions of wire diameter and spacing, as being analogous to a colour image. The amount of any one of the red, green, or blue components would correspond to impedance, attenuation and return loss, producing a coloured pixel in the given place in the (sampled) 2-dimensional space. As a result, the points selected by the GA could be compared with a colour map, with lighter regions being more optimal than darker regions of the same hue. The hue itself would convey which of the three

properties was most optimal. For example a redder hue would mean the region had better impedance properties than attenuation and return loss.

While this is a novel way to represent this information, there are clearly limitations to such an approach, because such a display is only effective for two or three variables plotted over a two dimensional surface, although the surface need not be flat. Applying such an approach over a three dimensional volume, or extending it to higher dimensions, introduces more difficulties, starting with the occlusion of voxels (volumetric pixels) further back from the viewer, and the selection of Alpha transparency so as to balance the displaying of the obscuring data with displaying that which is obscured.

The approach to designing the genetic algorithm was essentially selectionist; about the top 10% of the genomes were selected to the breeders. As has been discussed in chapter two, there is a tension between picking good solutions for future generations and discounting useful characteristics too early. In an algorithm that essentially depends on randomness it is difficult to make choices about the best strategy to choose for selecting breeding pairs. As a way of broadening the search, in the genetic algorithms discussed here, the implementation inserted one new randomly generated chromosome into each new generation. This amounts to a search that is purely random, and is therefore inherently inefficient. However since the initial population is chosen randomly this seems to be a sensible way of “stirring” the mix of characteristics. During the literature search no other examples of this technique were found. Its utility is arguable, but it seems difficult to demonstrate convincingly. One is essentially comparing two randomised processes, and in a large population size, the insertion of one individual with random characteristics which will most likely be eliminated in the next iteration is difficult to demonstrate statistically. On the other hand, it would seem to have little detrimental effect, because the selection process used is based upon a roulette wheel whereby unfit individuals have a very small chance of being selected for breeding.

## **3.2 Particle Swarm Optimiser**

Continuing the development of cable models, optimisation techniques were applied to the design of shielded coaxial cables. As well as obtaining results using a genetic algorithm, Particle Swarm Optimisation was also explored. This technique was investigated to see if it provided any advantages over genetic algorithms, but it appears the results are comparable. A novel aspect of this work was the attempt to limit the velocity by using an approximate mechanical model of the flying particles.

The swarms of particles are compared in the literature to swarms of insects or flocks of birds. These would have a maximum speed, which is imposed by drag forces on the creature, a force which increases as the square of the velocity. The literature generally uses some constant to limit the speed, rather than some model of a drag force, which was the approach proposed in this work. The value of the drag coefficient was chosen so as to achieve a velocity which meant that a particle travelling at the top speed would sample the space about ten times in transit across it.

If the drag force for a drag coefficient of  $C_d$ , for a object with cross sectional area  $A$ , in a medium of density  $\rho$  is given by:

$$F = \frac{1}{2} C_d \rho A v^2 \quad (3.1)$$

but let  $k = \frac{1}{2} C_d \rho A$  to collect all the constants together. Assume the mass of the particle is 1 so that

$$v_{t+1} = v_t + \text{attractor} - k v_t^2 \quad (3.2)$$

Then *attractor* is determined by the maximum velocity that can be generated for the two points as input to the velocity algorithm. That algorithm in this PSO model was to take a vector based on the memory:

$$\text{diff}_{\text{memory}} = p_{\text{best}} - p_{\text{current}} \quad (3.3)$$

$$\text{diff}_{\text{global}} = p_{\text{global\_best}} - p_{\text{current}} \quad (3.4)$$

$$\text{range} = \text{maximum} - \text{minimum} \quad (3.5)$$

and leaving aside the purely random component of the motion,

$$v_{t+1} = v_t + 2 \text{rand}(\cdot) \text{diff}_{\text{memory}} + 2 \text{rand}(\cdot) \text{diff}_{\text{global}} \quad (3.6)$$

where  $\text{rand}(\cdot)$  generates a random number between 0 and 1 inclusive. For this term to be maximised that would be 1, and the differences would have to be maximised as well, then the maximum change to the velocity we would get would be when the destination is at the opposite end of the space from the particle, so both differences are maximised. Then the maximum possible value of *attractor* would be  $4(x_2 - x_1)$ , where  $x_2$  is the maximal position vector and  $x_1$  is the minimum position vector. So we have

$$v_{t+1} = v_t + 4(x_2 - x_1) + k v_t^2 \quad (3.7)$$

At the maximum velocity we want the drag to cancel the attractive force:

$$v_t = v_t + 4(x_2 - x_1) - k v_t^2 \quad (3.8)$$

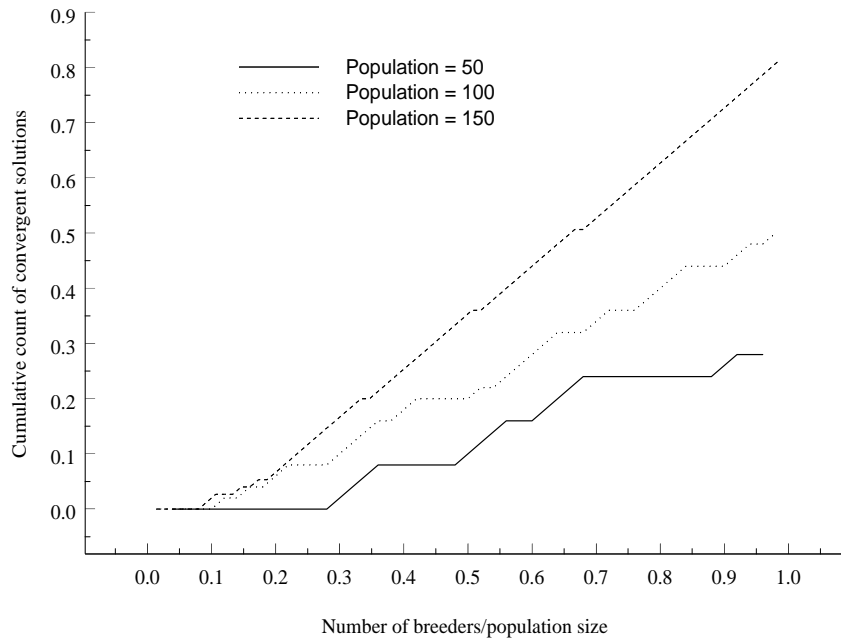
$$0 = 4(x_2 - x_1) - k v_t^2 \quad (3.9)$$

$$k = \frac{4(x_2 - x_1)}{v_t^2} \quad (3.10)$$

A particle travelling at the maximum possible speed should sample the space a reasonable number of times, say 10 times. Then  $v_m = \frac{x_2 - x_1}{10}$  and  $k = \frac{400}{x_2 - x_1}$ .

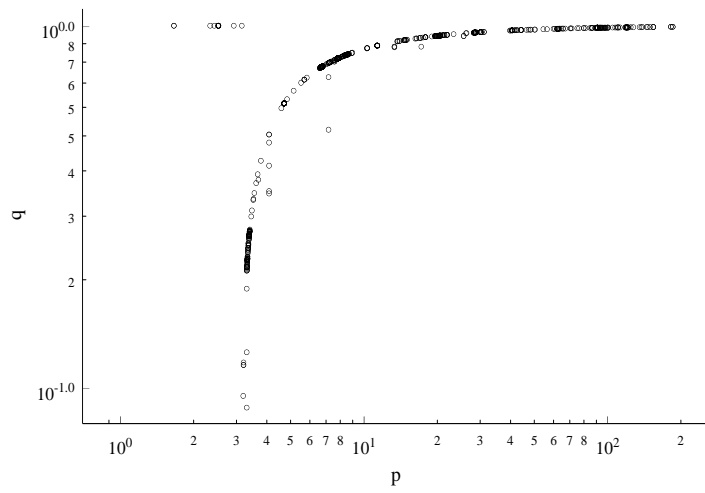
### 3.3 Optimisation and Cable Design Results.

Initially an analysis of how the number of breeders in a population affected the results was performed for three population sizes as show in Figure 6.

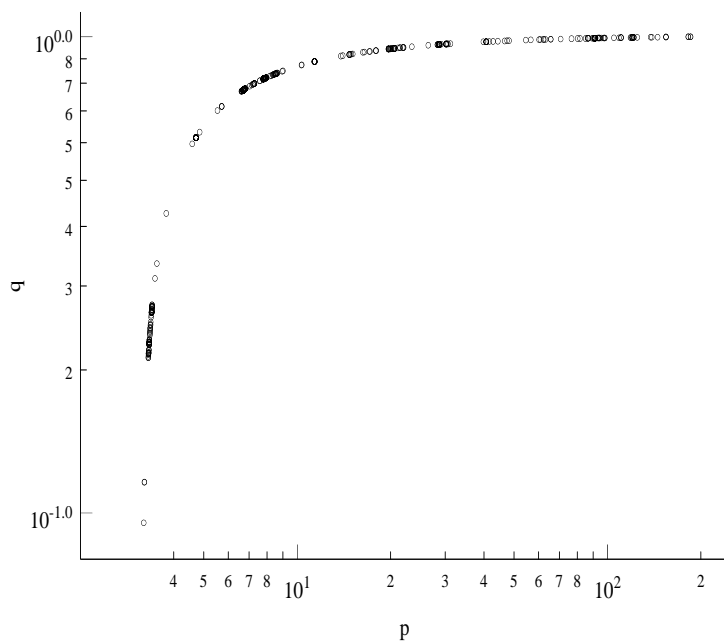


**Figure 6 Proportional cumulative count of convergent solutions as a function of the number of breeder/population size for three values of population size.**

It is clear that convergence is reached even for a small nuber of breeding pairs in the population. That not all the population converged can be seen from the following two figures, Figure 7 and Figure 8 .



**Figure 7** Solutions for a shielded cable (including non-convergent solutions) for 100Ω operation showing the design variables ‘p’ and ‘q’.



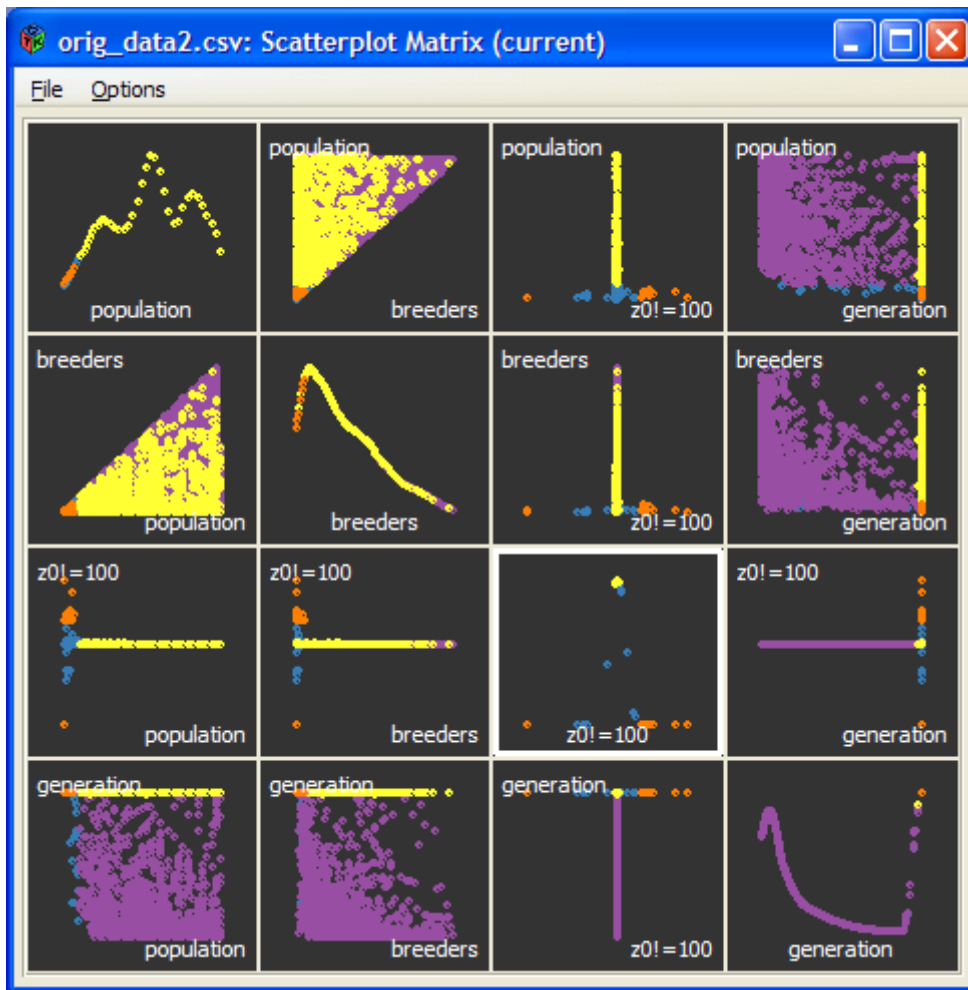
**Figure 8** Reproduction of the previous figure, but only including convergent solutions



The genetic algorithm was developed in C++ with a number of assumptions built into it which, while actually weakening it, provided further evidence to the robustness of the method, because the technique works even in sub-optimal circumstances. In particular the random number generator used in the GA was not initialized to a random value upon startup. This allowed repeatable trials, but meant that there was some inherent bias in the tests. Also the mutation function would only alter the value of one bit in the chromosome at any time, not each bit with equal probability (in which case more than one could be mutated at one time). Despite this the algorithm with mutation converged successfully in many cases. This demonstrates that although the impedance equation was one equation in two unknowns ( $p$ ,  $q$ , to give the desired  $Z_0$ ) a solution was still found.

The data produced was subsequently analysed in greater depth.

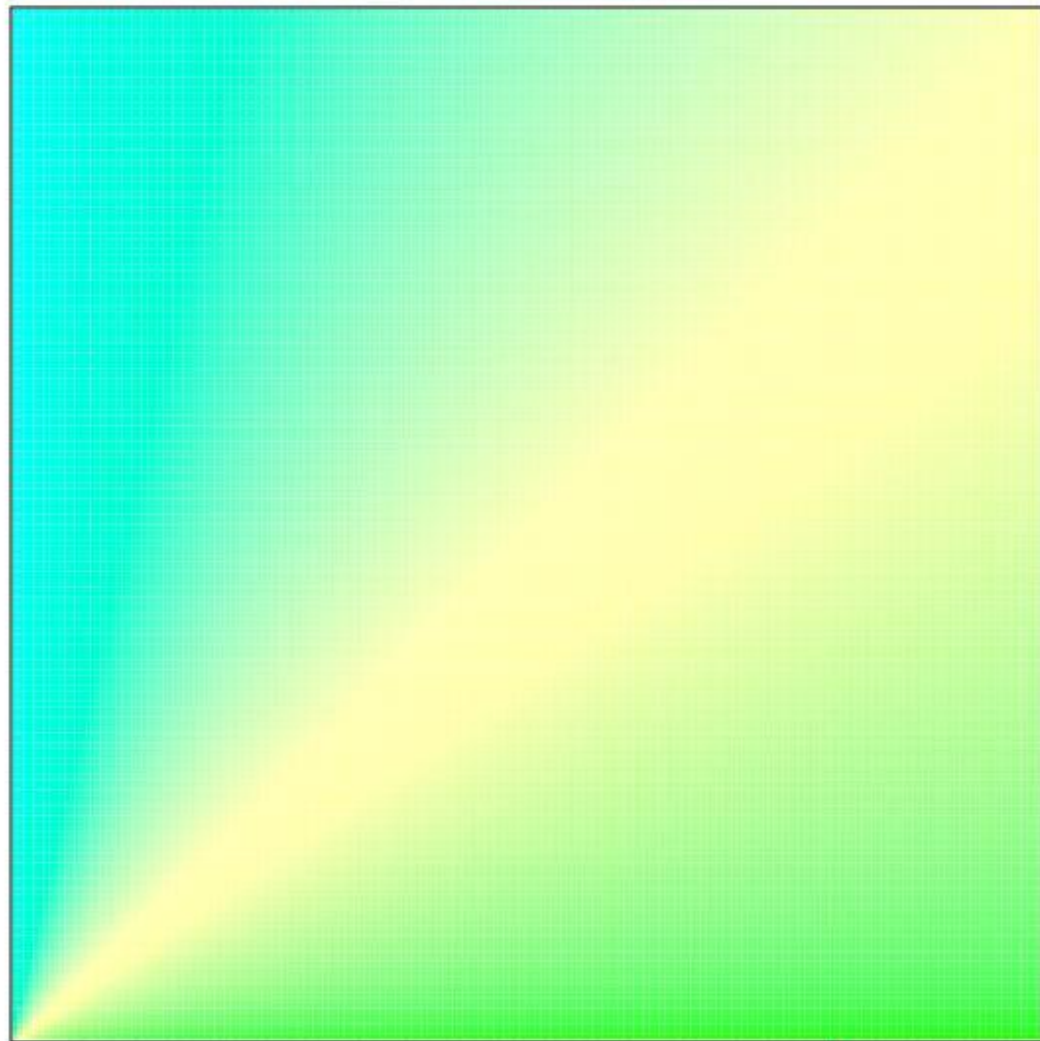
Figure 9 shows the scatter plot matrix for the genetic algorithm to optimise the impedance of a twisted pair cable. It was produced using the free software ggobi, which is the cross-platform version of XGobi [124]. The label “ $z_0 \neq 100$ ” refers to the original data where the final value of the impedance was only recorded when it was not equal to  $100\Omega$  making this a plot of  $Z_0$ . The leading diagonal of the matrix is a histogram of the particular variable of interest. The figure was coloured using the brush mode of ggobi in the following way: the top few generations were coloured yellow, is it can be seen which solutions ran for many iterations. The smallest population sizes were coloured blue, and the extremes of impedance were coloured orange. It seems clear from this display that increasing the number of parents (breeders) in a population speeds up the convergence. Very small population sizes do not converge to the desired impedance, but the population does not have to be vast to achieve convergence. The histogram of generations suggests that in many cases convergence is achieved rather soon. With suitable statistical tools this could be expanded into a more generally useful study, to explore repeated trials for each population size and number of parents, and to explore whether the use of a more sophisticated random number generator is worth the programmer effort to build it.



**Figure 9 Statistics from GA to optimise a simple twisted pair cable**

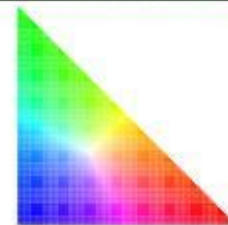
This scatter plot matrix is a way to represent points in a high dimensional space, by looking at points plotted onto planes whose two dimensions are selected from the available dimensions in this space. Where in the matrix those two dimensions would be the same, a histogram is plotted, showing the distribution of points as a function of the single variable. Thus plots on one side of the leading diagonal are reflections of the plot on the other side of the leading diagonal. So, from the breeders versus the population size plots, it can be seen that there were more breeders chosen for the larger populations, but for a large enough population size not all of them remain until the end. For small population sizes there are more extremes of impedance, because there was less chance to converge.

The plot of fitness, in terms of impedance, attenuation and return loss can be seen in for two different frequencies in Figure 10 and Figure 11. These colour maps are formed by mixing additive colour (that is, mixing works as it does for light rather than paint) as depicted in the included triangular diagram at the lower right of the plots. The colour bars give an idea of the contribution to each colour for the values of the appropriate variables, and the main image is the plot over the x and y values, with the origin at the lower left corner.



Plot from X = 1e-05 to 0.002      Y = 1e-05 to 0.001

Fitness values at f=5e+07 data file test\_data1

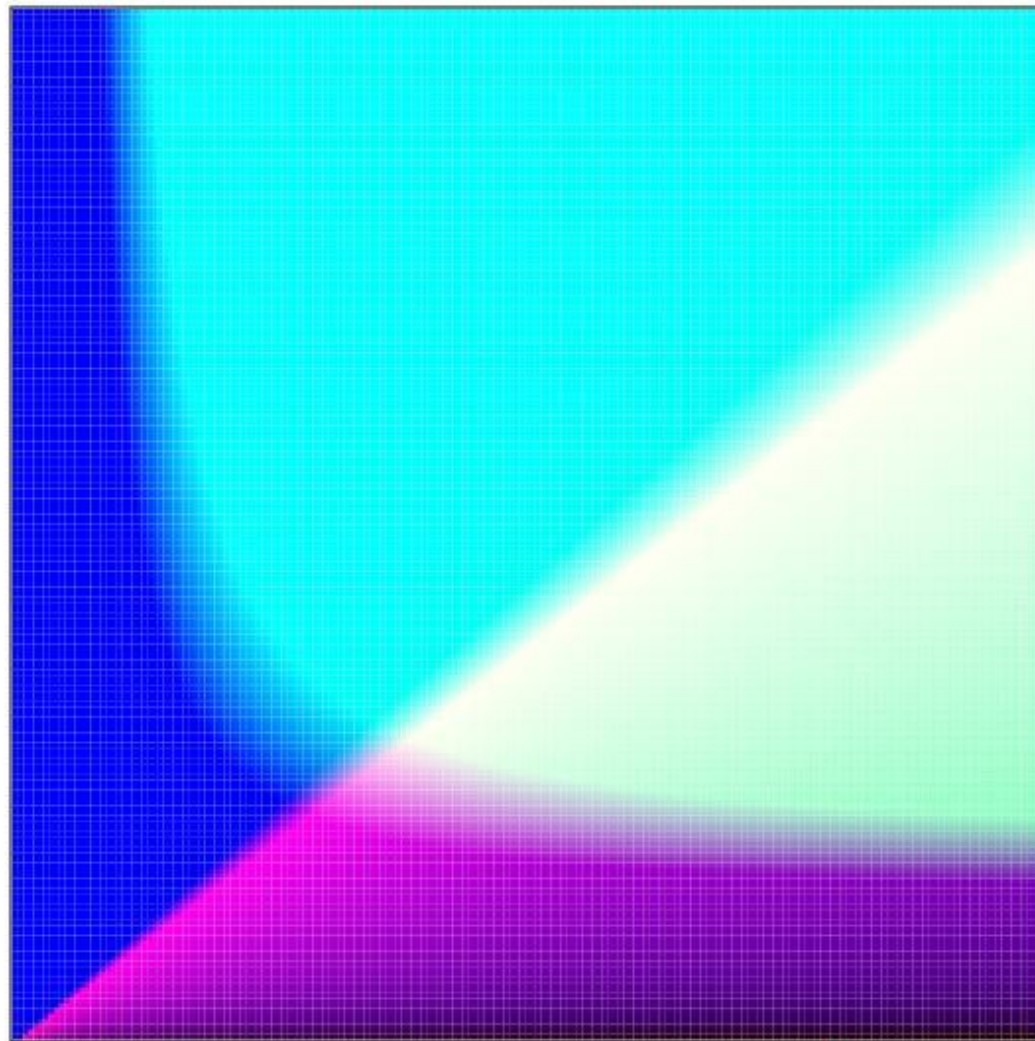


from red = 0  1

from green = 0  1

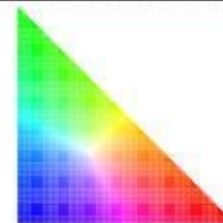
from blue = 0.528045  0.955285

Figure 10 Colour map for fitness functions for cable design (if viewed in colour, red = impedance, green = return loss, blue = attenuation). Fitness is proportional to colour intensity. X-axis is dielectric thickness, Y-axis is conductor separation in metres



Plot from X = 1e-05 to 0.002 Y = 1e-05 to 0.001

Fitness values at f=1.5e+08 data file test\_dataH



from red = 0 0.999958

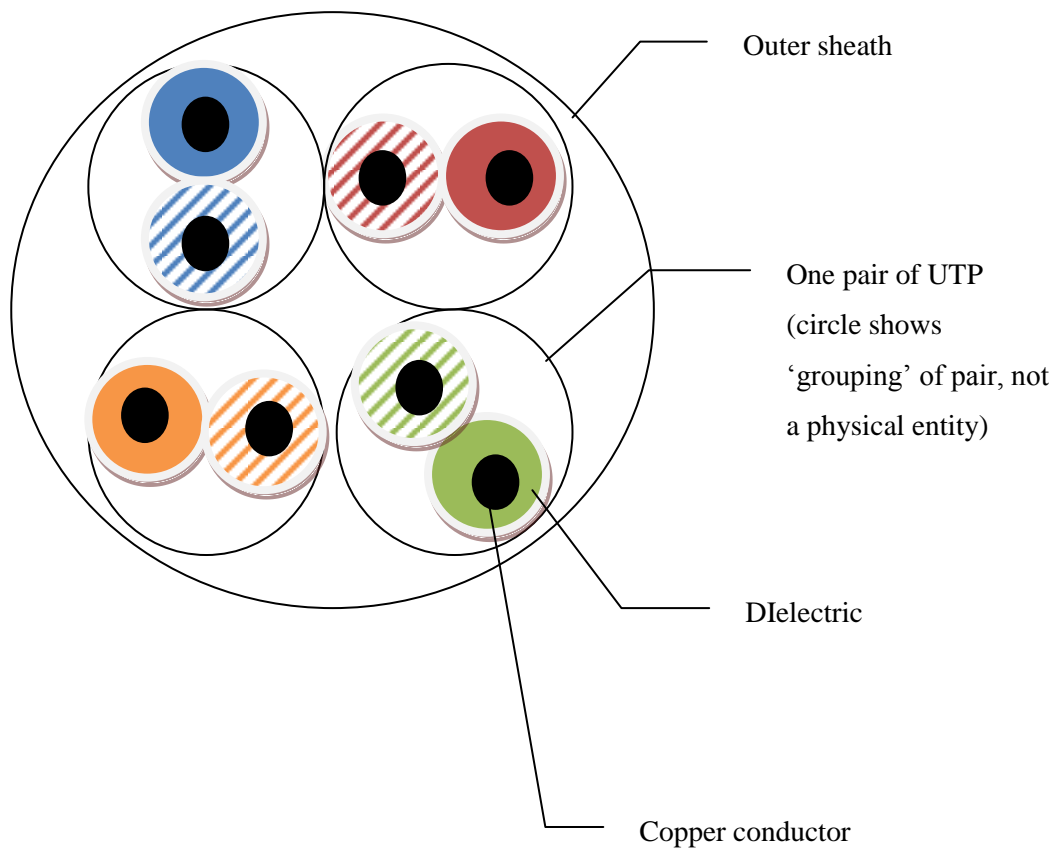
from green = 0 1

from blue = 0.0012769 1

Figure 11 Colour map for fitness functions for cable design (if viewed in colour, red = impedance, green = return loss, blue = attenuation). Fitness is proportional to colour intensity. X-axis is dielectric thickness, Y-axis is conductor separation in metres

The immediate visual impact of the approach is apparent. Extension to higher dimensions is nontrivial, both for reasons of the display techniques which are still in development, and the perception aspects, including how to handle data which is occluded by other data.

The work was further extended to cover several shielded twisted pairs in one cable [125]. These had a high density polyethylene dielectric. After the genetic algorithm, this time using a form of tournament selection with randomly selected pairs competing to “breed”, had been tested with a basic resistance model, then an impedance model, a cable with four twisted pairs was modelled. The basis of the model is shown in Figure 12, and this effectively provides a worst case for the diameter, given the desire for compactness, with an outer diameter of less than 5mm being desired, larger values allowed.



**Figure 12 The basis of the mathematical model of the UTP cable**

The fitness for impedance was initially a function

$$fitness = \begin{cases} 1 & \text{if } 85 \leq Z_0 \leq 115 \\ 0 & \text{otherwise} \end{cases} \quad (3.11)$$

and as a result the properties selected for the cable were:

Conductor diameter	0.4 mm
Dielectric thickness	0.182 mm
$Z_0$	100.892 Ohms
Overall Diameter	3.693 mm

This was then further extended to use a fuzzy fitness function, as depicted in Figure 13. The fitness here is to be regarded as membership of the fuzzy set of fit solutions, 1 being full membership, 0 being full non-membership. If only the values 0 or 1 were valid, then this would be a crisp set, rather than a fuzzy set.

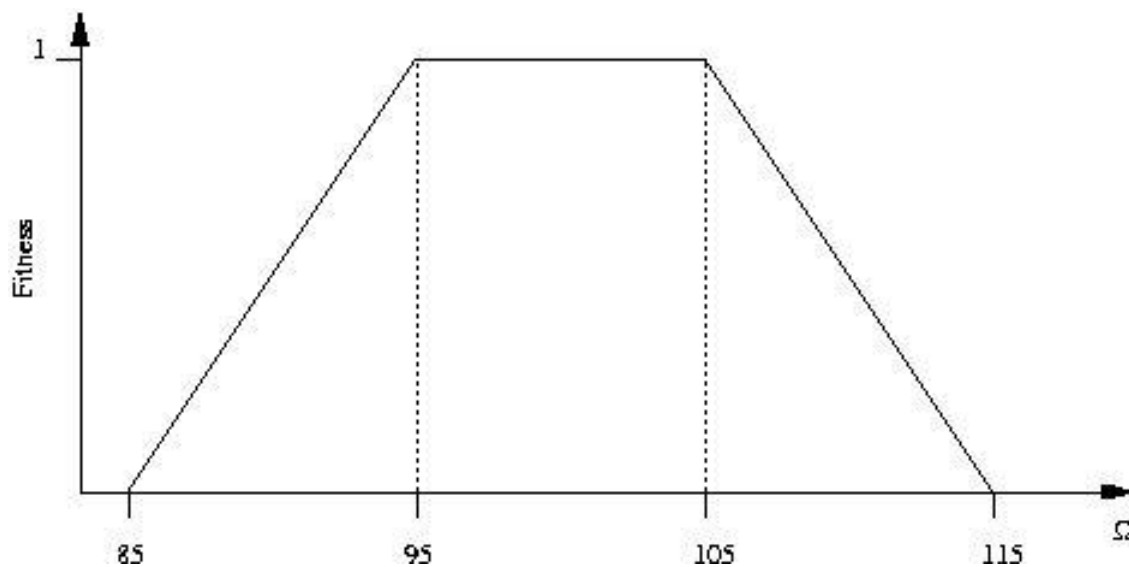


Figure 13 Fuzzy fitness function, to push the GA towards 100 Ohms

These fuzzy functions are combined using a minimum operator to represent AND, and a maximum operator to represent OR. This means that if the fuzzy membership function is replaced by a Boolean function, the results are still the same, i.e 1 OR 1 is 1, 1 OR 0 is 1, 0 OR 0 is 0, and for AND, the result is 0 except for 1 AND 1 which is 1. The results for this genetic algorithm were as follows:

Conductor diameter	0.475 mm
Dielectric thickness	0.217 mm
$Z_0$	100.932 Ohms
Overall Diameter	4.384 mm

The fitness function for attenuation shown in Figure 14 was added.

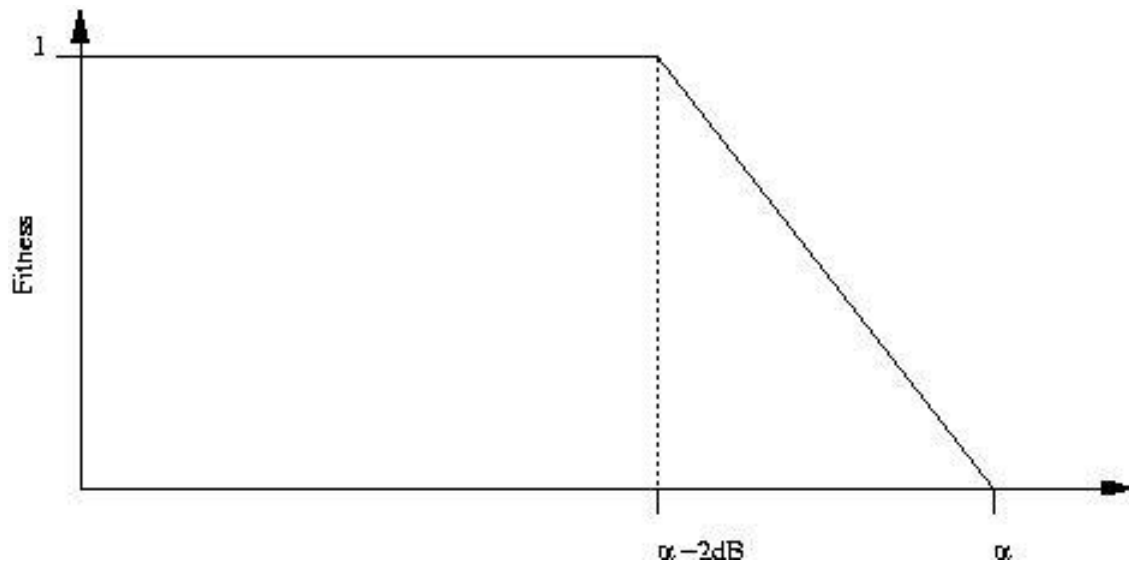


Figure 14 Fitness function for attenuation

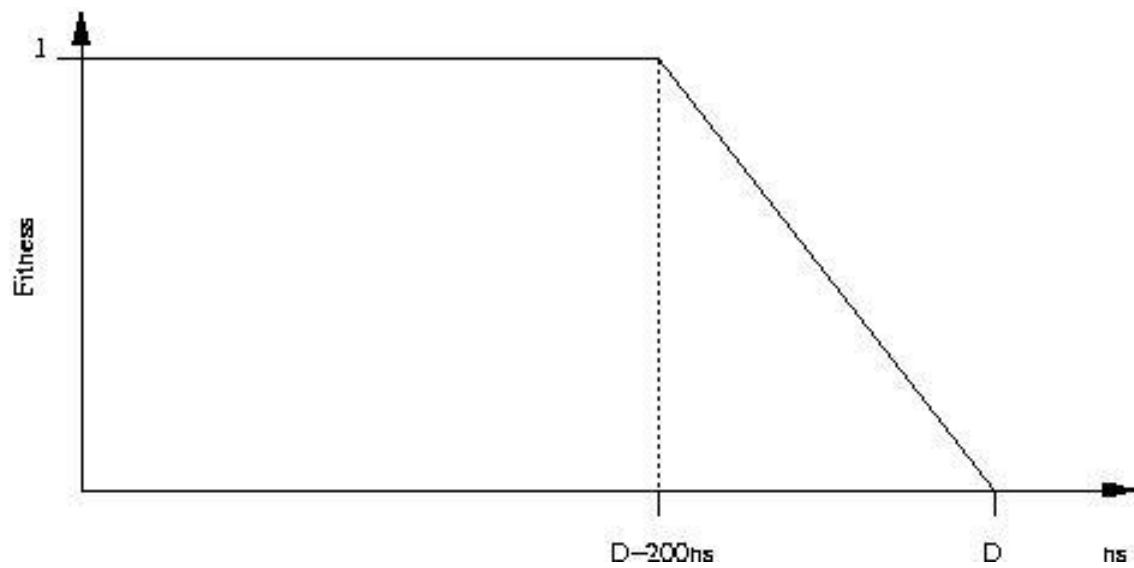
This produced some selective pressure to increase the conductor diameters to decrease the attenuation. The results of this change in the design parameters were:

Table 1 Results after applying fitness function from previous figure

Conductor Diameter	0.8 mm
Dielectric Thickness	0.357 mm
$Z_0$	102.369 Ohms
Overall Diameter	7. 505 mm
Attenuation @ 1MHz	1.042 dB / 100 m
Attenuation @ 10MHz	3.395 dB / 100 m
Attenuation @ 100MHz	11.631 dB / 100 m
Velocity of propagation @ 1MHz	501.442 ns / 100 m
Velocity of propagation @ 10MHz	501.448 ns / 100 m
Velocity of propagation @ 100MHz	501.451 ns / 100 m

A further criterion of cable design is the velocity of propagation. This may be measured as a phase delay  $v = \beta/\omega$ , and the fitness function for this is shown in Figure 15





**Figure 15 Fitness as a function of phase delay**

This was further extended by considering that there are manufacturing tolerances such as a cable, and so the diameter of the core was perturbed by  $\pm 0.05\text{mm}$  at each node in a 1D TLM model of the cable, with the nodes 1mm apart for 1 metre of cable. Following the TLM simulation the resultant time domain voltages were subjected to a Discrete Fourier Transform (DFT) so as to give both the incident and reflected signals at each end of the cable in the frequency domain. This meant that the attenuation could be computed directly. The results for this simulation were:

**Table 2 Results after applying phase fitness function**

Conductor Diameter	0.787 mm
Dielectric Thickness	0.377 mm
$Z_0$	103.129 Ohms
Overall Diameter	7.440 mm
Velocity of propagation @ 1MHz	501.442 ns / 100 m

Velocity of propagation @ 10MHz	501.449 ns / 100 m
Velocity of propagation @ 100MHz	501.451 ns / 100 m

A further experiment was carried out in which the cost of the copper in the cable was considered, and increasing it was a penalty on fitness. This would tend to decrease the thickness of the copper cores, a selective pressure in the opposing direction to that of attenuation.

**Table 3 Results after applying copper cost fitness function**

Conductor Diameter	0.519 mm
Dielectric Thickness	0.271 mm
$Z_0$	107.082 Ohms
Overall Diameter	5.127 mm
Attenuation @ 1MHz	1.502 dB / 100 m
Attenuation @ 10MHz	4.850 dB / 100 m
Attenuation @ 100MHz	16.332 dB / 100 m
Velocity of propagation @ 1MHz	501.438 ns / 100 m
Velocity of propagation @ 10MHz	501.448 ns / 100 m

Velocity of propagation @ 100MHz	501.451 ns / 100 m
----------------------------------	--------------------

This demonstrates the applicability of Genetic Algorithms to complex problems with conflicting requirements, and shows that trade-offs can be included with some success.

The use of optimisation techniques was applied to coaxial cables in [126], in which the effectiveness of the shielding was optimized. Broadband cabling is particularly sensitive to the performance of its shield, which is the reason for shielding performance being used as an illustration of the optimization techniques. Shield design may require a combination of up to two pairs of braids and shielding tapes. However if we were to ask “Can we get sufficient performance from a single braid and if so what is the best we can get?” we need to identify the combination of design factors that give the best shielding effectiveness or surface transfer impedance.

The basis for optimizing the Surface Transfer Impedance ( $Z_t$ ) has been to adopt the model that  $Z_t$  for braided cables is a balance between inductance caused by the braid itself ( $L_b$ ) and the diamond shaped apertures caused by the spacing between the belts ( $L_h$ ). The predominantly DC component of resistance has been ignored here.

$$Z_t = j\omega |L_b - L_h| \quad (3.12)$$

Using the model developed by Katakis [127] (based on that by Tyni [128]) the transfer impedance can be described as:

$$Z_t = j\omega \left| \left( \frac{-\mu h}{4\pi D_m} (1 - \tan^2 \alpha) \right) - \frac{\mu 2N}{\pi \cos \alpha} \left( \frac{b}{\pi D_m} \right)^2 e^{\left( \frac{-\pi d}{b} - 2 \right)} \right| \quad (3.13)$$

Where  $\alpha$  is the braid angle,  $N$  is the number of braid wires per belt,  $b$  is the separation distance of the belts and  $d$  is the diameter of each of the braid wires. The number of belts is implicit,  $h$  is the ‘gap’ between porpoised (the interweaving of the belts of wires) braids and can be given in terms of the parameters  $b$  and  $d$ ,  $D_m$  is another derived parameter based on the dielectric diameter ( $D_o$ ) and the braid wire thickness - and is the outer diameter of the cable - as given below. All other parameters have their usual meanings.

$$h = \frac{2d}{1 + b/d} \quad (3.14)$$

$$D_m = D_o + 2d + h \quad (3.15)$$

The diameter of the coaxial cable's dielectric has been held fixed at 1 cm,  $\alpha$  was allowed to vary between 0 and 90 degrees,  $b$  was allowed to vary between 0 and three times the braid wire diameter,  $N$  was allowed to vary between 1 and 10 wires and  $d$  was allowed to vary between  $10^{-6}$  m and  $10^{-4}$  m. It is acknowledged that these may not be appropriate constraints, but at this stage the question being addressed is "what's the best that can be obtained?". Other important factors such as power handling capability, return loss performance, mechanical robustness were not included at this stage.

Several simulations were run and the best of these was then used to obtain the  $Z_t$ .

The prediction of the PSO and the GA is given in Table I. The impedance is calculated at 6GHz and is in milliohms per meter ( $m\Omega m^{-1}$ ).

**Table 4 Optimised values for cable shielding**

	<b>PSO</b>	<b>GA</b>
Braid angle $\alpha$ , radians (degrees)	0.785 (44.9)	1.0 (57)
Braid wire diameter, $d$ , mm	0.1	0.075
Braid wire separation, $b$ , mm	0.1	0.023
Number of braid wires per belt, $N$	3	1
Surface transfer impedance at 6 GHz, $m\Omega m^{-1}$	10	11

The similarities and differences are very clear. On the one hand, the braid needs to be laid as close to  $45^\circ$  as possible but the other solution requires a wider lay. One requires a single  $75\mu m$  braid wire, the other requires multiple wires per belt separated by close to the wire diameter. However, they both produce results that are close to each other, which underlines the complexity of the problem space.

It should also be noted that, with these techniques, there is some dependency of the results on the starting conditions, so it is always worth running several simulations with different initial values in order to check that the optimum has been achieved. Limited simulations were run; for example, the genetic algorithm only used 50 individuals over 100 generations.

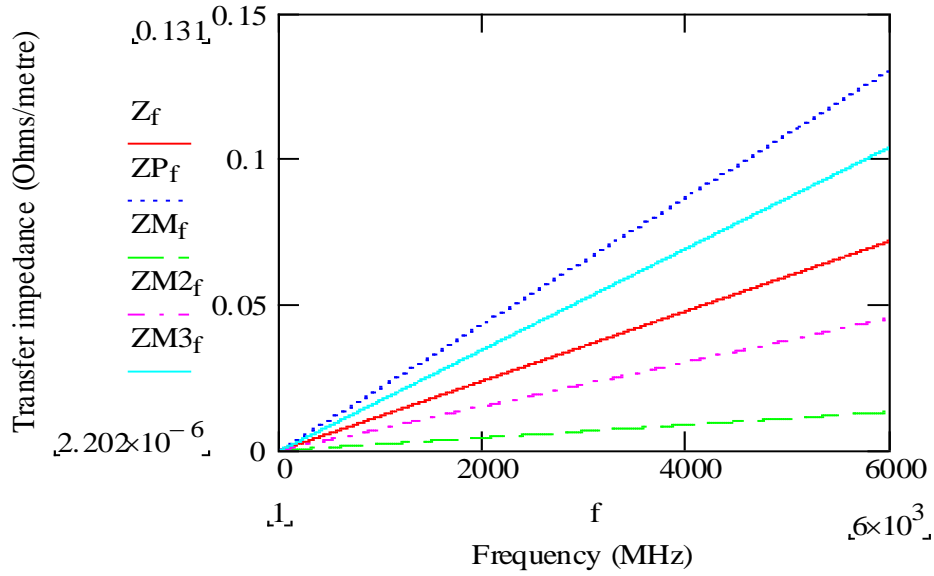
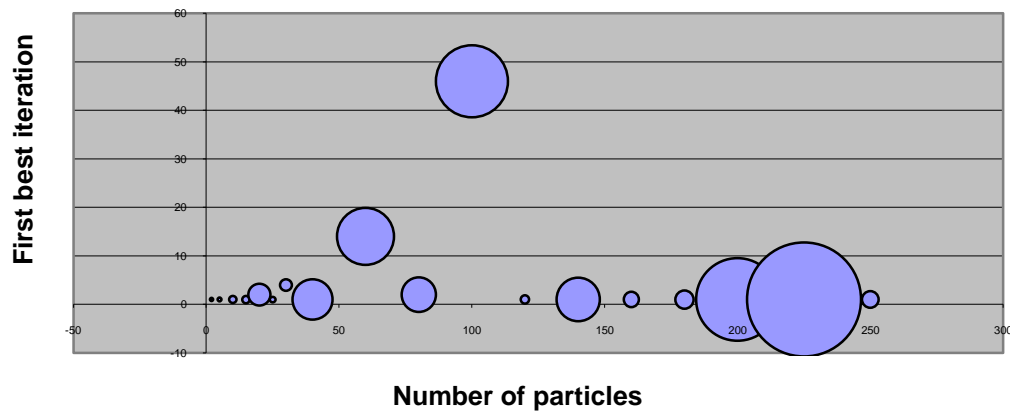


Figure 16 Impedance sensitivity with number of wires in each belt as a function of frequency.

It is helpful to understand the effects of small changes in any of the parameters because particular care needs to be taken in accepting results where there is a very narrow distinction between an excellent result and an unacceptable one as this allows informed design judgments to be made. An interesting illustration of this comes from one early run of the PSO, with very few particles, which predicted a transfer impedance of  $72 \text{ m}\Omega\text{m}^{-1}$  with  $\alpha=0.789$ ,  $d=0.0001$ ,  $b=0.000214$ ,  $N=7$  (which would also give good optical coverage). Subsequent runs improved on this as more particles were brought into the problem space. However, an investigation of the sensitivity to the number of wires in a belt showed that an even better result could be obtained. Figure 16 shows the effect of changing the number of braid wires on the transfer impedance up to 6GHz. The figure shows the effect of increasing by 1 (labeled ZP) or decreasing by 1, 2 and 3 (labeled ZM, ZM2 and ZM3 respectively). It can be seen that the results will improve when the optimizer flies to  $N=6$  or  $N=5$ , but  $N=3$  will increase the impedance.

The best value obtained was with  $N=6$ , giving an impedance of  $13 \text{ m}\Omega\text{m}^{-1}$ , very close to the value presented in Table 4, but with entirely different parameters. This is a clear illustration of the plurality of possible solutions in the problem space.



**Figure 17 Visualizing the quality of PSO solution as a function of number of particles in a swarm and the first best iteration.**

An indication of how well the schemes work can be seen by reference to Figure 17 which compares the number of particles in a swarm with the first iteration to reach the best fitness function. The size of the bubbles is a representation of the quality of the final result (the larger the bubble, the lower the transfer impedance). It shows that good solutions were obtained early in most cases but that the random nature of the probing of the problem space means that not all simulations will reach an optimum value in a limited number of iterations. Parameters such as velocity will contribute to how well the optimizer works for a given problem space structure. As a means of finding a solution in some space, particle swarm optimisation seems to perform comparably to genetic algorithms, although the model of particles actively exploring the space may be more intuitive when trying to understand its operation.

# 4 Parallel Distributed Implementation of a Transmission Line Modelling Method Solver

## 4.1 Introduction and Design Criteria

One of the issues facing the development of models for electromagnetics is the long running time of simulations. The aim of this part of the work was to tackle this by means of distributed computing. The search of the literature showed that most of the work in this area used special hardware, either commercial parallel computers, or dedicated networks of sophisticated workstations. Budgetary constraints mean that this sort of solution is not so widely applicable. On the other hand many organisations have PCs which are generally unused outside office hours. If a way could be found to utilise this resource, or even only part of it, this would be a useful contribution to a computational facility. However, this meant that the system had to meet several specialised requirements which would not apply to those systems that were already published.

1. The person whose machine is to be used must have full control over its availability. Otherwise they may be unwilling to lend us the CPU for the work.
2. The system must be simple to install.
3. The installation must avoid interference with other software on the system.
4. Because of the variety of platforms available, it must be portable.

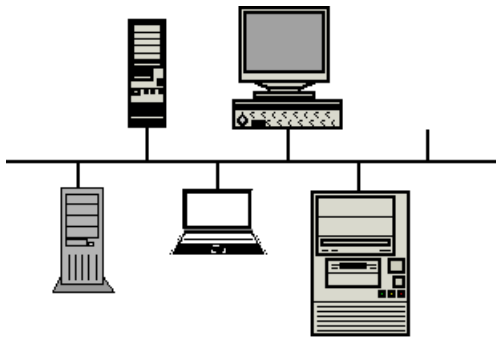
The consequences of these constraints are significant.

1. The requirements on machine availability mean that the network of machines could grow and shrink *during* the simulation. Also, people may allow the use of their machine when they leave their desks, perhaps for a few hours, but they will need to be able to get it back

immediately when they return. Therefore simulations on their machine may not be finished.

2. The need for a simple installation ruled out the use of libraries for parallelisation. This is because such libraries usually depend on other libraries, and when one considers this in combination with the desire to be portable across various Unix flavours and Windows versions, this would make maintenance difficult.
3. The avoidance of interference with other software also demands a minimal set of dependencies because if there are specific versions of libraries needed, then they may conflict with the requirements of other software. This is particularly the case for Dynamic Link Libraries (DLLs) under Windows, and there is a colloquialism among programmers to describe this: “DLL hell”. It was also important not to introduce libraries that could interact unpleasantly with existing libraries, because whatever was installed should not be detrimental to the owner’s system.
4. There must be a way to notify machines that the simulation has finished so that those who have provided machines may withdraw them at the end of one simulation, if they choose to keep the machine available for one, or more, whole simulations.

The result would be a heterogeneous network of machines of differing power, some of which may leave during the simulation, rather as depicted in Figure 18.



**Figure 18 A heterogeneous network of machines.**

Initially a TLM solver was developed in C++. An object oriented language was chosen in order to simplify the design, to separate aspects such as the different node types from what happens during the connect phase, and to allow the design to be compartmentalised into manageable chunks. This was validated rather simply by constructing a closed metal box, and exciting it. The resonances agreed with expectation.



The requirement for performance led to the investigation of UDP as a lightweight protocol to support the communication between the parallel machines, but the complexity of overcoming the problems of packets arriving out of sequence, and possibly not arriving at all was considered too great. Instead, communication was carried out over TCP/IP, which has a protocol stack to overcome these problems.

The DRuby, or DRb library is part of the standard library for Ruby. It is a distributed communication system for Ruby objects using TCP/IP, allowing one object on one machine to be accessed from another machine. This library implements Rinda, which is an implementation of the tuplespaces used in the Linda programming language. The initial R signifies that it is a Ruby implementation (in the same way that many Java based projects have names beginning with J), but it is also a joke, because the Ruby language comes from Japan, and the nearest sound Japanese has to R is also the nearest sound they have to L, hence the confusion for some Japanese people between L and R when speaking English [129].

## **4.2 The choice of an object oriented language**

The choice of an object oriented (OO) language was made in order to assist in the creation of more maintainable code, and this choice has had impacts on how the system was developed.

A brief discussion of Ruby's style of Object Oriented (OO) programming may be helpful. Whilst a comprehensive discussion of OO programming is outside the scope of this work, it is fair to say that the idea of OO programming is a shift of emphasis from what a program does to what a program operates on, i.e. the data, and the inherent properties of that data [130]. It is a logical extension of abstract data types: one is not overly concerned with how an object works, but one is concerned with the operations one may perform on that data type. With objects, the relationship to what they are and what they can do is closer than that, because the operations belong to the object type as "methods". In most languages, the "class" and the object's type are synonymous. Languages such as Smalltalk, and to some extent Ruby, regard the method calls as messages sent to the object, evoking some response, possibly an error if it does not respond to that method. Methods are implemented as subroutines. For example, an array will provide a way to insert and delete elements, but some other types will not.

Three aspects of OO are important in pushing the code in a more maintainable direction: encapsulation, inheritance, and polymorphism. Encapsulation means that an object may be composed from several objects (or items of data) like a structure in C or a record in Pascal, but these are treated as being invisible to users of the object, because the user sees only the interface to the object, not its insides. A driver observing another car on the road need not be concerned with that car's engine, but will have to be concerned with the speed and temperature of the engine in their own car. One calls the methods of the object to change its state, so how that state is implemented is permitted improvement without affecting the interface. Inheritance means that when a class is derived from another class it inherits the methods and data types of its parent. When an instance of some class responds to a method, it may ask the parent class to respond, which therefore may ask its parent to respond. The new class then gets its own methods to make it more specialised. (This is to save representing the same information in two or more places in a program, which means that changes have to be performed more than once, and if they are not, then inconsistencies arise.) "Polymorphism" means that classes derived from a ((great-) grand-) parent class can be used in places where that ((great-) grand-) parent may be used, without rewriting that code. A consequence of this is that specialisation means that all properties of the parent class must be supported by the child class. The result of this is that designs which are object oriented will consist of many pieces of code which are self contained and have minimal interaction between each other, thereby allowing change to be localised. Keeping information local makes changes to individual objects easier, because the information is collected in one place and can be seen all at once. More importantly, because objects are designed to present an interface to the outside world, this facilitates re-use of objects in different contexts. (It has sometimes been argued that this is overstated, and that the "re-use" goal of OO programming has not really been as successful as was hoped, but such a discussion is outside the scope of this thesis. It seems only fair to note that there are other viewpoints about this technique.) Furthermore, substitution of alternate designs which conform to a given interface is possible, allowing (for example) the various types of TLM node to be interchanged as needed.

All this flexibility comes at the price of more complicated management of the whole design: the class hierarchies must be designed to support this kind of loose interaction between many discrete data types. As Fred Brooks points out in [131], "There is no silver bullet" that will simplify software design. Therefore the complexity must reside somewhere. Thus, the choice of programming environment affects the design of the system. An example would be how the Numerical Electromagnetics Code (NEC) takes its input on notional punched cards, a constraint existing now because earlier code was constrained by what was practical in Fortran some years ago. The choice of an object orientated language was intended to help manage the complexity of the design and reduce debugging time.

Ruby is a dynamic language, meaning that a variable is not bound to any one type (one does not declare variables as being character, or integer, or voltage), and may take several during its “life” in the program. The classes, and the instances of classes, are open to modification, and thereby extension: extra methods may be added to a class or one object, and this is used to make objects conform to some interface, for example, the ability to “serialize” an object as a string in the YAML format is done in Ruby by providing a `to_yaml` method. The inheritance model is one of single inheritance with “mixins”, in that each class has one parent class but one may “mix in” a module to supply it with extra methods. (This contrast with C++, for example, which allows multiple parent classes, the search for methods and properties having a defined ordering). Therefore, when considering the “type” of an object, its class is insufficient to completely describe it, because it may respond to methods obtained from one or more modules, or declared for it directly. Classes are constants; it is the instances of the class or the referenced class variables in a class that can vary. So it is possible to have a list, in Ruby that is an Array, with classes listed in it: they are just constants which may be used in the program. This is the way in which the Rinda system represents patterns to be matched.

### 4.3 Linda, Rinda, and the Tuplespace model of programming

The Linda programming language uses a tuplespace as a blackboard for communication between processes [132]. A blackboard in computing terms is some object that is readable and writable by the different processes, and manages the synchronisation inherent in this communication. A tuple is a list. So there exists a space into which processes may write lists, and they may also test for the presence of lists by matching, and they may also remove lists from the tuplespace. In the Ruby implementation the matching is done using `nil` as a wildcard to represent anything, and using the Object Oriented inheritance “is a” relationship for matching.

### 4.4 Organisation of the sharing of work.

Let there be a number of jobs in a pool. The pool in this case is a tuplespace, a collection of tuples, and in Ruby these are arrays. The jobs may be represented as being available to be worked on, or being in progress, and the results from a job may also be stored in the same tuplespace. The space may be imagined as something like **Error! Reference source not found.**, i.e. in an unordered space.

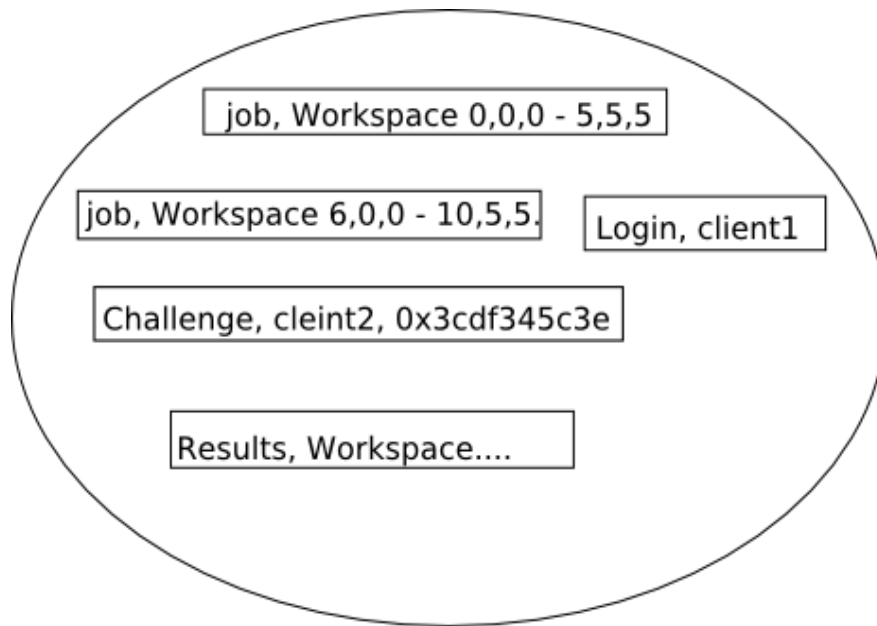


Figure 19 A "tuplespace" with various kinds of data in it.

Processes look for jobs in the pool by matching the appropriate pattern and finding which ones are available. If there are no jobs which have no processes working on them, then in the software described in this thesis it was decided that the processor may select a job which is already being worked on. In this way there is scope for competition between machines to complete a job, if there is sufficient computing resource available. There is then the possibility that a faster machine can complete a job before a slower machine, and thereby speed up the whole process. Also, the process scales downwards as well: if there is only one CPU to do the processing, it will just plough through the work alone. Given that any of the machines may be removed from the pool without notice, this also increases reliability and performance of the system.

Because it cannot be determined *a priori* which machines will be available to take a job from the pool, it is necessary to ensure that a job can be executed by the machine with the smallest expected memory size. This will influence how the workspace in a simulation is partitioned into jobs. It is not necessary to worry about the speed of the machine taking the job in a system where unfinished jobs may be taken by other machines.

When a job is completed by a fast machine, it would be efficient to stop the job on any slower machine running it. This can be done by means of a "poison pill". At present this mechanism of a

poison pill is only used to terminate the server when that is appropriate, but its extension is a logical step to take.

There is also the issue of what to do when all work is complete. There may be work submitted after a pause, so there is a case for keeping the server running. Alternatively, the server may be terminated after a suitable pause. The client machines may be withdrawn by their owners at any time, but it became clear that it would be useful to provide a means to notify them that now might be a good time to leave because there is no more work at the moment. This allows people who are otherwise content to run the client in the background to be aware that they may stop the client.

When a machine makes a request to join the system, it is given a limited time in which may do this. In the present design this time is 120 seconds. This is in order to prevent a malicious user from creating a machine that would make successive attempts to join the pool of machines until it managed to break in. This is achieved by use of the nonce (a value created and used only once), and only allowing it to remain within the tuplespace for the specified time.

Terminology that would fit the metaphor was difficult to arrive at. A clear metaphor for a Software System helps when modelling the objects necessary to make up the system. It was clear that there was work to be “contracted” out and that they will work pieces to be worked on, but the provider of work was described as a server because this was basically a client server model. Thus, the objects created were a server (to provide the work), contractors (to do the work) each of which had a skill set (a set of operations it can perform) and work pieces which were essentially inert. The skill set object was there to provide for different classes of contractor. It was envisaged that the parallel system would be used for more work than just TLM simulations. For example, in a parallel genetic algorithm there could be workers operating on the gene pool as if the items were to be manipulated according to a genetic model (with crossover and mutation), and workers operating on the individual chromosomes in order to determine their fitness, which would treat them as sets of parameters for input to a computational model. Clearly this would be different classes of worker. The design of the code of that implemented the parallelism was based on this being more general purpose than for a TLM simulation alone.

There are issues in how to divide up the TLM workspace, some of which remain unexplored. One of the difficulties faced by distributed computing systems, a difficulty which provides a “force” pushing people towards dedicated networks or parallel hardware, is the necessary intercommunication between the component parts, and the resulting bandwidth requirements. In a system which models wave propagation, there is going to be significant information flow between the parts. This presents several possibilities for reducing the amount of necessary traffic, even before one considers how the

data might be compressed. One may choose to keep the volumes on each process as large as possible, and minimise the surface area of each volume, which is where the parts come into contact. This presents the possibility of automatic segmentation. Division of the workspace according to material type may also be useful, especially in the cases where some of the interfaces would contain metal or perfect electrical conductor, through which no waves propagate. Boundaries inside metal would have no information flowing across them. Another solution is to only compute what can be computed without reference to neighbouring processors. Where the voltages on a node are dependent on unavailable inputs, then such computation may be delayed until such information is available. This distributes the bandwidth over time. Thus, a notional region held by one CPU, would lose one layer of nodes each timestep into the future. **Error! Reference source not found.** shows this for a workspace divide unevenly between three machines, the darker regions being more out of date without connectivity. Then, when communication is possible, there would be some work to “catch up” using the data from the other parts.

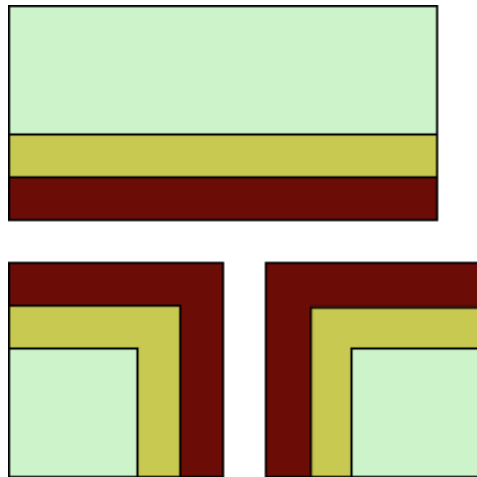


Figure 20 Workspace divided showing (successive) regions which cannot be solved without connectivity

Boundaries were implemented to have a combination of various types. “Absent” means the boundary is electrically absent; its presence is for some other function. “Electric” and “magnetic” boundaries have reflection coefficients of -1 and 1 respectively and may be used for boundary conditions as well as thin metal structures. “Dielectric” boundaries signify a change in dielectric constant between two non-conducting regions. “Processor” boundaries mean this is the point a workspace is split across machines, or between processes running on the same machine. There were also “Absorbing” boundaries implemented so as to absorb all energy that goes into them. Boundaries marked as “intelligent” were included to allow for digital filtering, etc, which remain unused.

The complexity in implementing a system which can segment the space into non-rectangular regions was deferred, in order that a simpler system could be shown to work first. The system described in this thesis relies on splitting the space across each axis, ensuring the resulting segments are cuboid. This is done manually at present: automatic segmentation has not been developed. Clearly greater automation of this kind is desirable.

## 4.5 Security Considerations

Another consideration that became apparent during the design was the need for security features in the code. Given that the system would be used on a University or corporate network, it was initially considered to be something for future work. However, these networks are part of the internet, and breaking through firewalls is not completely unknown. It was therefore necessary to find some way to counter at least some kinds of threat to make sure that it was not trivial to damage the work being done.

There were two conflicting requirements: the need to be able to collaborate easily with people sufficiently trusted; and the need to keep out everyone else. Given the need for the widest possible reach of collaborators geographically, in order to get as much CPU power as practical, the system would run into problems with getting help from countries not allowed to use cryptography because the authorities have regarded it as a munitions technology. Whilst the countries which are not allowed to trade in such technologies has varied, the USA has restricted such traffic to Iran, Iraq, Cuba, and Libya in the past. The UK is a signatory to the Wassenaar agreement, which has similar restrictions [133], and the wording on the UK government website is the same [134]. Also, as the work was commencing, in the United Kingdom, the Regulation of Investigatory Powers act was coming into force, requiring the handing over of cryptographic keys to the police on request. This was a level of complication that was undesirable. The solution chosen was not to encrypt anything, but to use a hashing function to ensure the integrity of the data being sent. The system known as HMAC does exactly this, and it does not send the key with the data.

Hashing functions are commonly used in computing for accessing “contents addressable stores”, in which a number is computed from the data to be stored, and this number becomes the address or index into the data structure. A cryptographic hash function is one in which it is extremely difficult to work

out what the data was from the hash itself. These are used as checksums for files to ensure that the file downloaded is the correct file, not a compromised copy containing a virus or other malicious software. Popular examples include the MD5 (message digest) hashing function, which recently has been found to be less secure than expected, and SHA-1. The insecurity lies in the ability to create another data set with the same hash in feasible amounts of time; this allows the faking of the data which the hash still checks correctly.

HMAC [135] works by generating a hash from the data together with a secret key, known to both ends of the communication. The receiving end uses the data sent and the key to regenerate the hash, confirming that it is correct. The third party, attempting to attack the system (for example by injecting faked data into the communications channel) will need to have access to the key to be able to generate the correct hash for the transmitted data. If the cryptographic properties of the hashing function are weak then it becomes possible to infer the key from the messages and their corresponding hashes. A famous historical parallel would be the attacks on the Enigma cipher during the Second World War. The implementation of HMAC in Ruby was novel at the time, in so far as no existing libraries were found that did this, but the code was not published.

This scheme was taken and extended further by adding a “nonce” into the message. A nonce is a word that is used only once (according to the Concise Oxford Dictionary.) This nonce included a timestamp to facilitate the expiry of data, so that messages could not be faked with a given timestamp without this taking place within a certain time of that stamp. Thus the hash of the message, nonce, and password is generated. The recipient has enough information to check the hash is valid, a third party does not. This is used in CRAM-MD5, see [136]. CRAM stands for Challenge Response Authentication Mechanism. Computing a faked message far into the future would rely on the job still being available to be worked on. The use of a timestamp was also to prevent repetition, by an attacker, of a previous collection of data. This is known as a replay attack. The nonce used in the code used the current time, the current process identifier, and the object identifier for the current thread.

The hashing algorithm used for all these cryptographic hashes was SHA-256, i.e. a 256 bit hash using the SHA-2 algorithm, thus more difficult to attack than SHA-1 which is only 160 bits. These functions are part of the Ruby standard library, and thus no external dependencies are generated by using them.



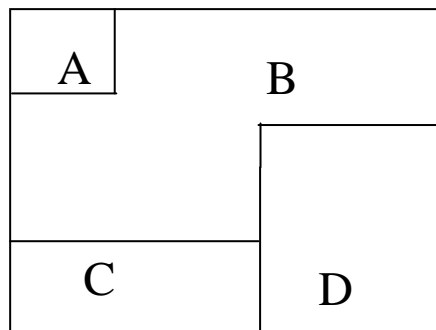
## 4.6 Node Design.

The nodes started out as metal/air nodes, and after the model was shown to be basically working, and that the distribution across machines was working, a node based on the Symmetrical Super Condensed Node [42] was developed. Because of the structure of the code and the OO nature of the design, this was effectively a drop-in replacement for the previous metal/air type of node.

The implementation of the metal node was achieved by simply negating the voltages on all the ports. Similarly, an absorbing node (for use in boundaries) was implemented by setting all the voltages to zero. For the other types of material, a “factory method” [137] is used to construct a node with the correct properties. These properties determine the numbers in the scattering matrix.

## 4.7 Parallel TLM

The preliminary work on distributed TLM was to consider the means of distributing TLM in the first place, and this was presented in reference [138]. The issues of distribution were developed further in reference [139] where the problems of dividing up the workspace were discussed. In particular it was noted that irregular regions such as in Figure 21 are difficult to analyse usefully, in order to determine how the geometry affects performance.



**Figure 21 A potential Interprocessor boundary problem. A, B, C, D represent individual processors.**

In the final implementation only planar boundaries between processors were used, as shown in Figure 22 for four processors A, B, C and D, with performance related costs of  $a$ ,  $b$ ,  $c$  and  $d$ , where the effects of changing  $\alpha$  as a proportion of  $l$ , or  $\beta$  as a proportion of  $m$  are much simpler to reason about than in Figure 21. Moving these dynamically in a system where any processor could choose to work on part of the workspace remains an unsolved problem.

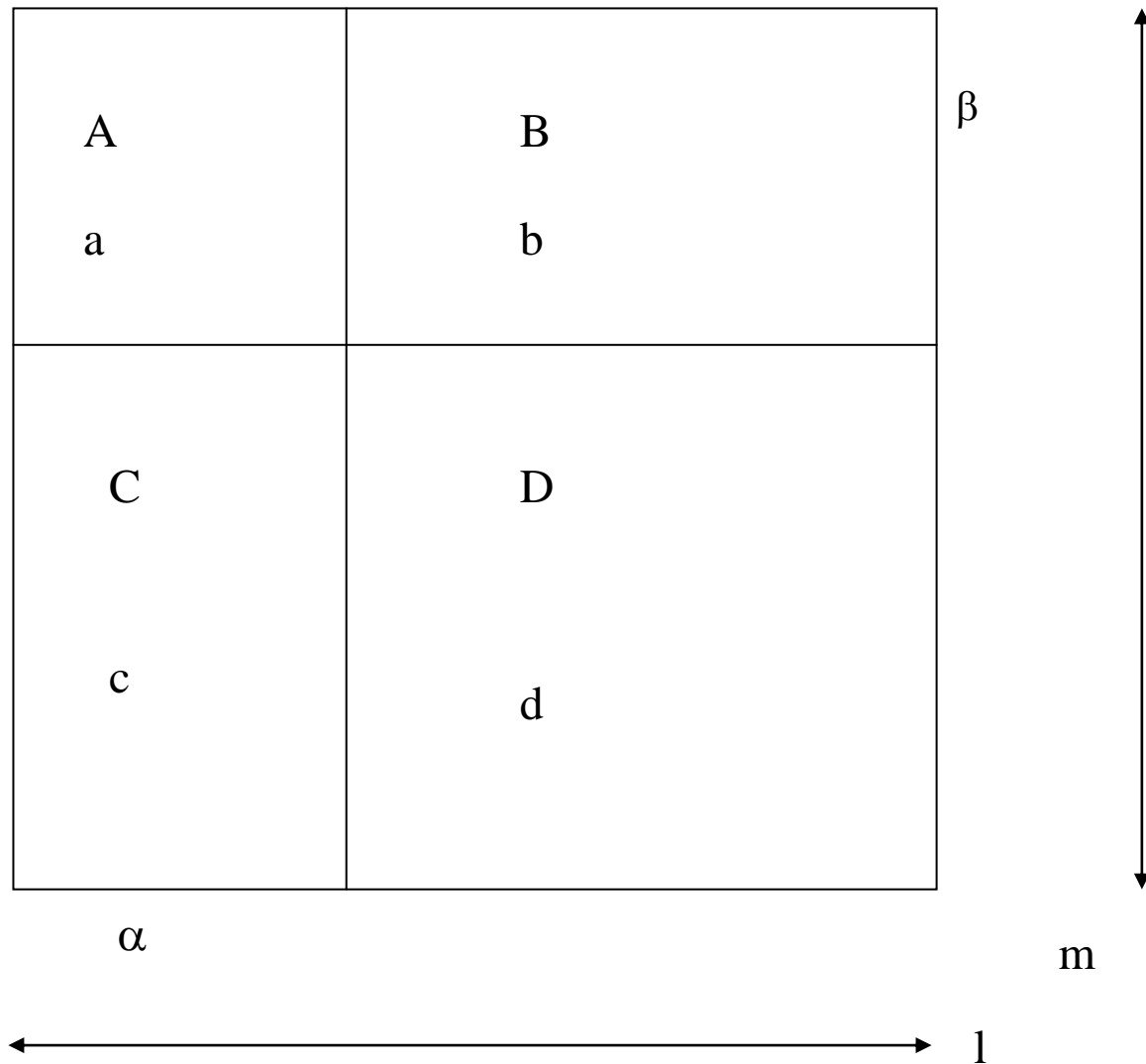


Figure 22 Four processors with planar boundaries

The system was extended to cope with more materials than the initial design of metal and air, using the Symmetric Super Condensed Node [58]. The system was shown to work across a trial network consisting of several Sun workstations and a PC. To improve efficiency in utilising the network, the data was compressed using Gzip [140], using a C library which ships with Ruby as standard, ensuring

that it was as fast as possible for a Ruby implementation. To ensure network transparency, the binary compressed data was encoded using bas64 encoding [141], in which a binary sting is encoded as printable characters in the set {'0'-'9', 'A'-'Z', 'a'-'z', '+', '/'}, as is used for transmission in electronic mail systems. It may be possible to drop this requirement, but it was included in order to reduce the possibility of obscure bugs being introduced.

One of the difficulties with the system was validation. Some tests were performed on rectangular metal boxes to start with, of varying dimensions. The results were inspected by eye, and agreed with theory. The first example was created with a workspace size of 7x8x9 nodes, with the boundaries all metal (perfect conductor), and an initial vertical pulse of one volt at the junction of the nodes at (4, 4, 4) and (5, 4, 4). The output was the sum of the vertical voltages between (5, 5, 5) and (6, 5, 5) for 10010 timesteps.

Setup 1 was:

Created Thu Aug 08 14:57:17 BST 2002

With this initialisation:

```
c.create_workspace(7,8,9) # Dimensions of workspace
```

```
plane_boundaries = [
    ["x",0,["Metal", "External"]],
    ["x",6,["Metal", "External"]],
    ["y",0,["Metal", "External"]],
    ["y",7,["Metal", "External"]],
    ["z",0,["Metal", "External"]],
    ["z",8,["Metal", "External"]],
]
```

```
c.set_voltage(4,4,4,"vxpy",1.0);    c.set_voltage(5,4,4,"vxny",1.0);
$timesteps = 10 * 7 * 11 * 13
```

With outputs taken from:

```
v = c.get_voltage(5,5,5,"vxpy") + c.get_voltage(6,5,5,"vxny")
```

for 10010 timesteps

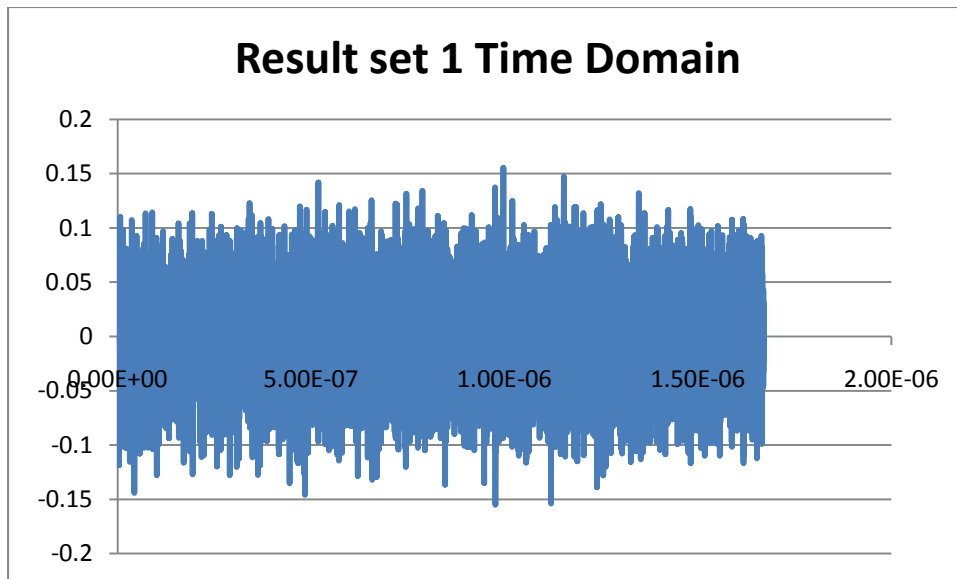


Figure 23 Time domain results for setup 1

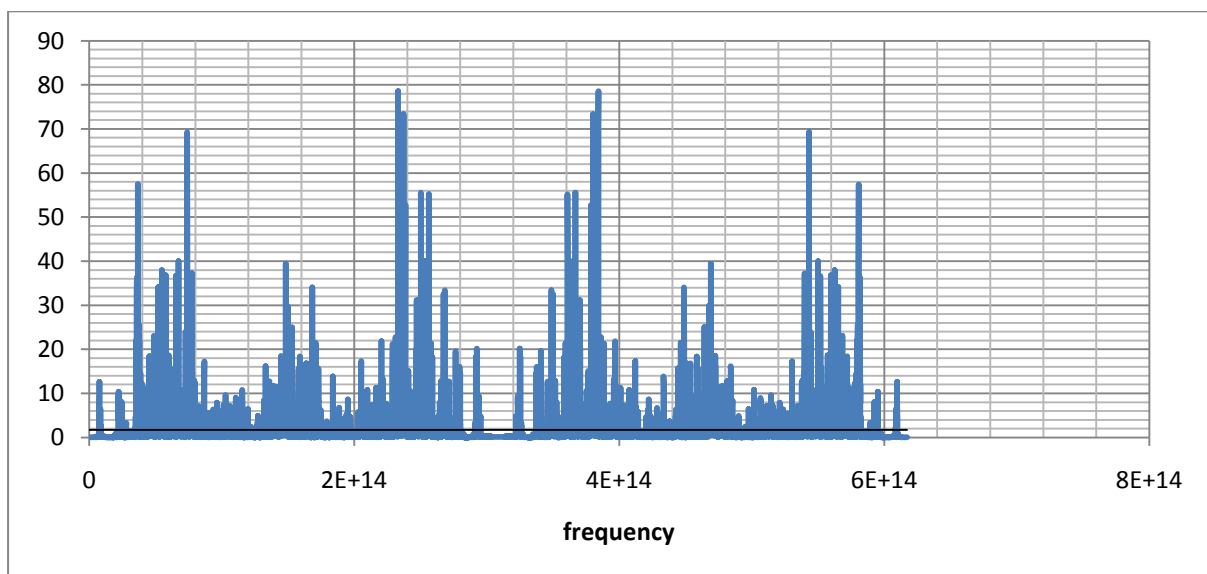


Figure 24 FFT of output from setup 1

Setup two was similar:

Created Thu Aug 08 16:07:31 BST 2002

with this initialisation:

```
c.create_workspace(7,8,10)
plane_boundaries = [
    ["x",0,["Meta1", "External"]],
```

```

["x",6,["Meta1", "External"]],
["y",0,["Meta1", "External"]],
["y",7,["Meta1", "External"]],
["z",0,["Meta1", "External"]],
["z",9,["Meta1", "External"]],
]

c.set_voltage(4,4,4,"vxpy",1.0)
c.set_voltage(5,4,4,"vxny",1.0)
$timesteps = 10 * 7 * 11 * 13
with outputs taken from:
    v = c.get_voltage(5,5,5,"vxpy") + c.get_voltage(6,5,5,"vxny")
for 10010 timesteps

```

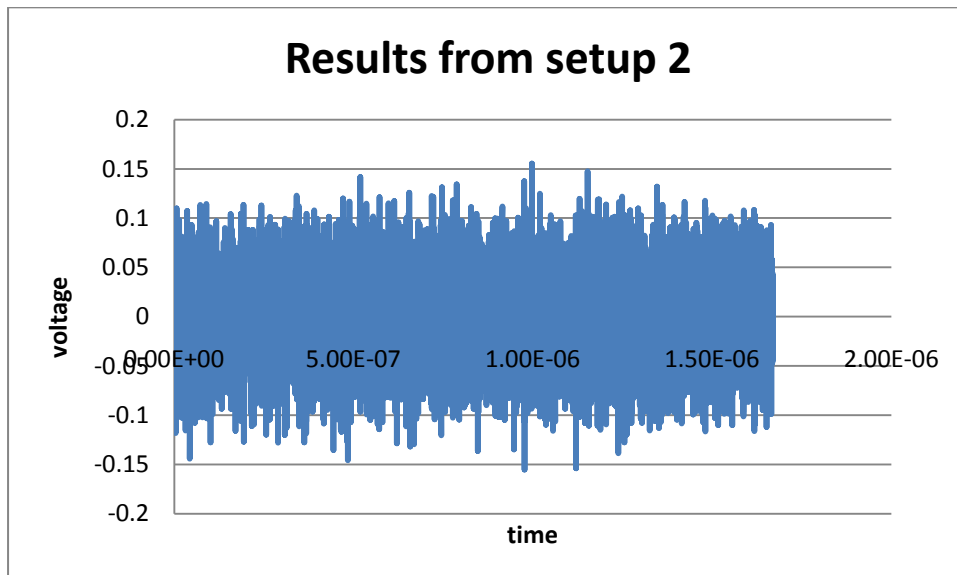


Figure 25 Results from setup 2

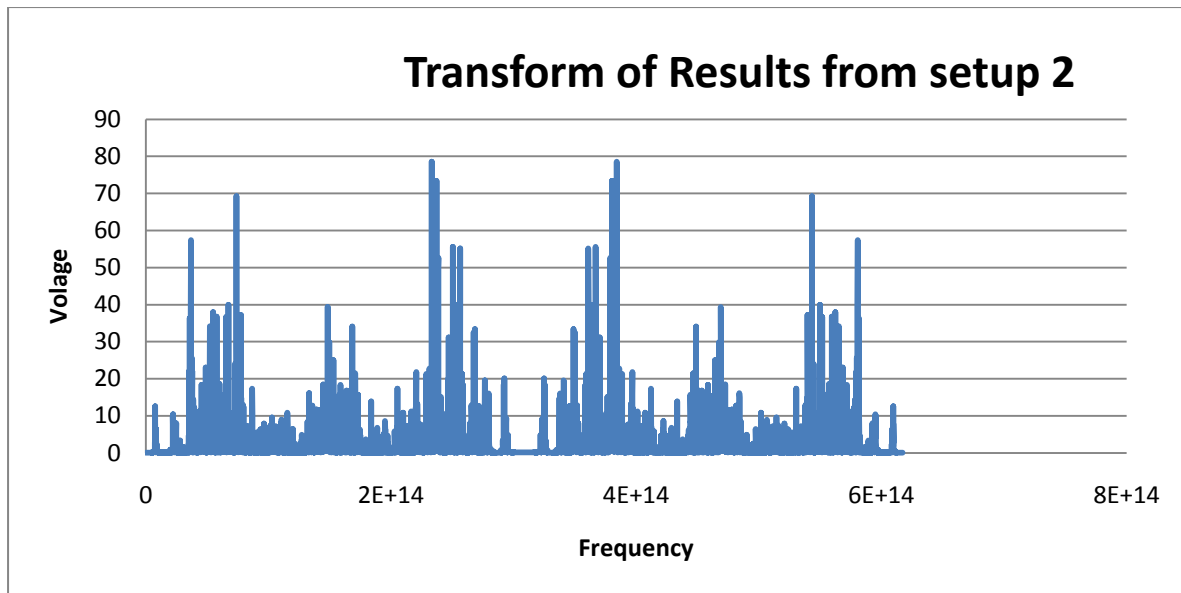


Figure 26 FFT of results from setup 2

Setup 3 was also cuboid

Created Thu Aug 08 17:26:28 BST 2002

With this initialisation:

```
c.create_workspace(7,8,14)
```

```
plane_boundaries = [
    ["x",0,["Metal", "External"]],
    ["x",6,["Metal", "External"]],
    ["y",0,["Metal", "External"]],
    ["y",7,["Metal", "External"]],
    ["z",0,["Metal", "External"]],
    ["z",13,["Metal", "External"]],
]
```

```
c.set_voltage(4,4,4,"vxpy",1.0)
```

```
c.set_voltage(5,4,4,"vxny",1.0)
```

```
$timesteps = 10 * 7 * 11 * 13
```

With outputs taken from:

```
v = c.get_voltage(5,5,5,"vxpy") + c.get_voltage(6,5,5,"vxny")
```

for 10010 timesteps

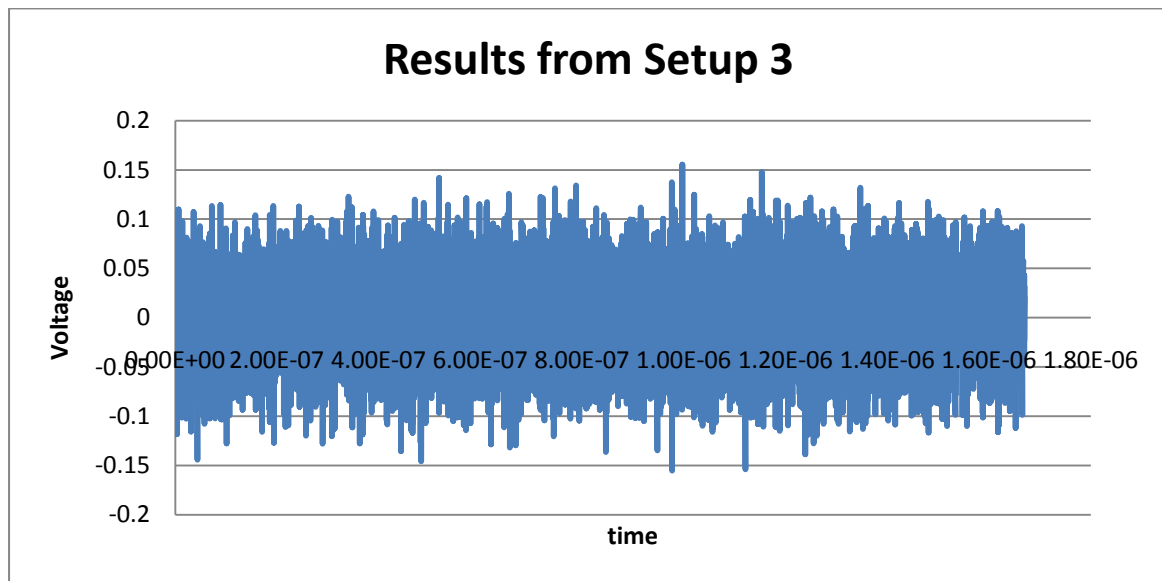
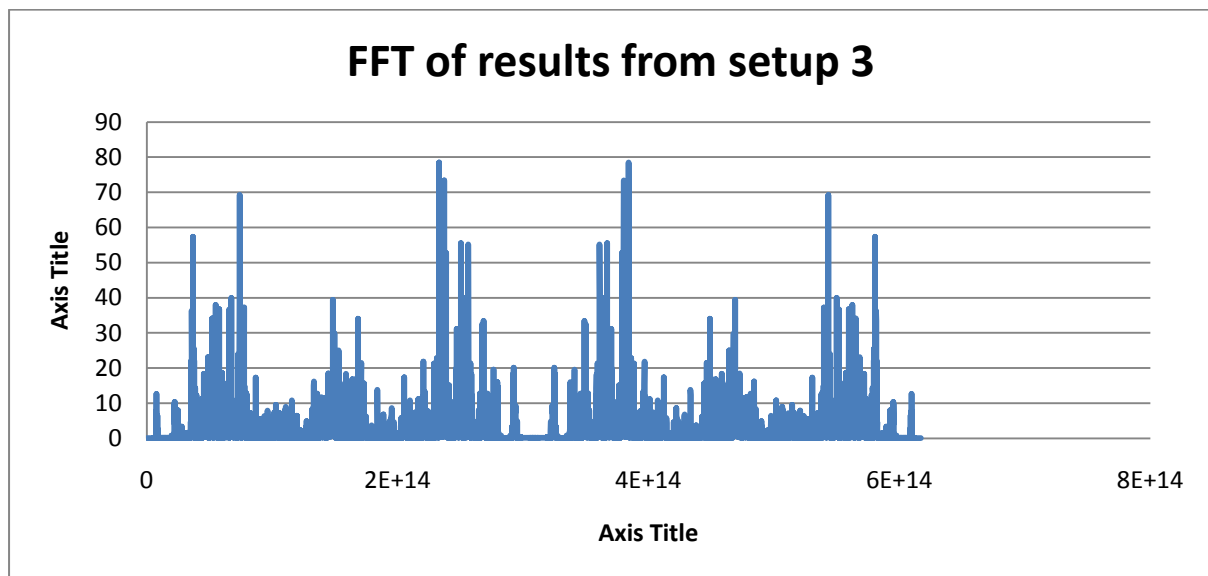


Figure 27 Time domain results from setup 3



However, it was apparent that this was a severely limited test when what was really of interest was not the output at one point, but the distribution of fields over the modelled structure. It was clear that there were resonances and that they were similar, but it was still rather difficult to interpret. This problem occurred more than once, and was a stimulus to development of the two-dimensional FSV method.

As part of the construction of the software, unit tests were used to confirm the correct behaviour of components, such as the workspace, copying data in and out of it, the functioning of the individual nodes, and the propagation of voltage pulses through the node. Thus confidence in the correct function was founded on more than a few outputs.



# 5 Two Dimensional Feature Selective Validation

## 5.1 Introduction

The one dimensional feature selective validation process was developed to assist the decision making process of EMC engineers when comparing data sets from measurements and from simulations [122]. The aim of the research described here was to extend the process from one-dimensional data sets to two-dimensional sets, as a prelude to the necessary extension to higher dimensions, and to identify and solve the main obstacles to such a transition.

Most of the graphs used in comparisons for electromagnetics work are of one parameter such as field strength against a single variable, which is often, but not exclusively, frequency. These variables are obviously chosen to be representative of the problem in some way. However, of increasing interest are situations with multiple independent axes, such as the distribution of the electromagnetic field about a piece of equipment under test, for certain test conditions, or current densities in the ground plane of a circuit board. In such cases, the independent axes are clear: Cartesian axes and/or frequency. Also, the conditions themselves, such as the configuration of the equipment, its layout, etc., may be considered variables in the problem, and each of these variables may be considered to a dimension of the problem in some configuration space. So, to make decisions using most or all of the information available, it is necessary to be able to perform comparisons where the result is determined by “position” in a space of more than one dimension.

Returning to the one dimensional case, where only the frequency is considered as an independent variable for the given setup, experience has shown that whilst a consensus can be reached among a group of engineers about the quality of a graphed result, or a comparison of two graphs, this agreement is not immediately attained, and some discussion is necessary first. Feature Selective Validation can help in reaching this agreement, and the heuristic is based on how engineers examine the data; in particular, which features they look at when making comparisons. It is thus aligned with how they usually work, and is now an IEEE standard [142].

The examination of the data by engineers usually consists of looking at the general trends in the data, the level offset between the graphs, as well as the detail (how corresponding peaks in two data sets agree, often accompanied by observations about how “grassy” the data is in regions of the graph, a description of the noise levels apparent in the data). The one dimensional FSV method operates in a low and high frequency domain also. The data is transformed into the Fourier domain, the lowest five points about DC are separated out, and the rest divided such that 40% of the energy in the graph is in the low frequency region, the rest in the high frequency region. For the two data sets being compared, the lowest cutoff is used for both data sets. The inverse Fourier transforms are performed for the 3 regions (DC, Lo, Hi) for both data sets and the rest of the calculations are performed in the original domain of the data. These calculations involve finding the first and second derivatives of the data ( $Lo'$ ,  $Hi'$ ,  $Hi''$ ) and thus the Amplitude Difference Measure, the Feature Difference measure, and the overall Global Difference Measure, as detailed below.

## 5.2 FSV Equations.

$$ADM(n) = \left| \frac{(|Lo_1(n)| - |Lo_2(n)|)}{\frac{1}{N} \sum_{i=1}^N (|Lo_1(i)| + |Lo_2(i)|)} \right| \quad (5.1)$$

$$FDM_1(n) = \frac{|Lo'_1(n)| - |Lo'_2(n)|}{\frac{2}{N} \sum_{i=1}^N (|Lo'_1(i)| + |Lo'_2(i)|)} \quad (5.2)$$

$$FDM_2(n) = \frac{|Hi'_1(n)| - |Hi'_2(n)|}{\frac{6}{N} \sum_{i=1}^N (|Hi'_1(i)| + |Hi'_2(i)|)} \quad (5.3)$$

$$FDM_3(n) = \frac{|Hi''_1(n)| - |Hi''_2(n)|}{\frac{7.2}{N} \sum_{i=1}^N (|Hi''_1(i)| + |Hi''_2(i)|)} \quad (5.4)$$

$$FDM(n) = 2(|FDM_1(n)| + |FDM_2(n)| + |FDM_3(n)|) \quad (5.5)$$

$$GDM(n) = \sqrt{(ADM(n))^2 + (FDM(n))^2} \quad (5.6)$$

This has been refined, somewhat, to include the DC offset within the ADM [117].

$$ADM(n) = \left| \frac{\alpha}{\beta} \right| + \left| \frac{\chi}{\delta} \right| \exp \left\{ \frac{\chi}{\delta} \right\} \quad (5.7)$$

where

$$\alpha = \{|Lo_1(n)| - |Lo_2(n)|\} \quad (5.8)$$

$$\beta = \frac{1}{N} \sum_{i=1}^N (|Lo_1(i)| + |Lo_2(i)|) \quad (5.9)$$

$$\chi = (|DC_1(n)| - |DC_2(n)|) \quad (5.10)$$

$$\delta = \frac{1}{N} \sum_{i=1}^N (|DC_1(i)| + |DC_2(i)|) \quad (5.11)$$

The two dimensional forms of these equations are essentially the same, except that the indices into the data are two dimensional.

$$ADM(n, m) = \left| \frac{\alpha}{\beta} \right| + \left| \frac{\chi}{\delta} \right| \exp \left\{ \frac{\chi}{\delta} \right\} \quad (5.12)$$

where

$$\alpha = \{|Lo_1(n, m)| - |Lo_2(n, m)|\} \quad (5.13)$$

$$\beta = \frac{1}{NM} \sum_{i=1}^N \sum_{j=1}^M (|Lo_1(i, j)| + |Lo_2(i, j)|) \quad (5.14)$$

$$\chi = (|DC_1(n, m)| - |DC_2(n, m)|) \quad (5.15)$$

$$\delta = \frac{1}{NM} \sum_{i=1}^N \sum_{j=1}^M (|DC_1(i, j)| + |DC_2(i, j)|) \quad (5.16)$$

$$FDM_1(n, m) = \frac{|Lo'_1(n, m)| - |Lo'_2(n, m)|}{\frac{2}{N} \sum_{i=1}^N \sum_{j=1}^M (|Lo'_1(i, j)| + |Lo'_2(i, j)|)} \quad (5.17)$$

$$FDM_2(n, m) = \frac{|Hi'_1(n, m)| - |Hi'_2(n, m)|}{\frac{6}{N} \sum_{i=1}^N (|Hi'_1(i, j)| + |Hi'_2(i, j)|)} \quad (5.18)$$

$$FDM_3(n, m) = \frac{|Hi''_1(n, m)| - |Hi''_2(n, m)|}{\frac{7.2}{N} \sum_{i=1}^N \sum_{j=1}^M (|Hi''_1(i, j)| + |Hi''_2(i, j)|)} \quad (5.19)$$

$$FDM(n, m) = 2(|FDM_1(n, m)| + |FDM_2(n, m)| + |FDM_3(n, m)|) \quad (5.20)$$

$$GDM(n, m) = \sqrt{(ADM(n, m))^2 + (FDM(n, m))^2} \quad (5.21)$$

## 5.3 Problems with FSV Formulation

There are a number of problems with this formulation which have shown up during testing the two dimensional code. The simplest test to perform was to supply inputs of 0. Equation (21) then results in a delta of 0, resulting in a division by 0 in Equation (17). Clearly this is actually a perfect comparison, so the division by zero is unhelpful here. The next most obvious test was to have two DC inputs that differ. For data sets which happen to be DC only, the derivative terms and the low frequency terms will sum to zero, causing the same problem for the FDM. To ensure that FSV still reflects how people judge comparisons, it will be necessary to conduct more experiments to determine how engineers react to these kinds of problems with the input.

## 5.3 Details of Implementation

When the Fourier transform is performed with the Fast Fourier Transform (FFT), data must be zero padded if it is too short, because the FFT algorithm requires that the size of the input be a power of 2. In the case of a two dimensional FFT, both dimensions of the input need to be powers of two, but the input need not be square. This becomes important when considering what to do about the cutoff for the 40% of the energy being in the low part of the data, and the rest being in the high part of the data. If the sizes of both dimensions are not equal, then it will be necessary to use something other than a circular “cut-out” to determine the membership of the Hi and Lo regions. Clearly, an ellipse should be used as the generalisation of a circle, also because it extends naturally to higher dimensions as an ellipsoid, becoming a spheroid when sizes in each direction are identical. (A rectangular function was considered, but although this made the computations simpler, it was rejected on the basis that it does not reflect what happens to diagonally oriented frequency information in the input.)

The inclusion within an ellipse would determine which parts of the data were considered to be low frequencies, and which were high frequencies. This inclusion was calculated by raster scanning the whole of the input array and determining which points fall within the ellipse using the equation for an ellipse, that is,  $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$ , although this meant processing points which clearly lay outside the ellipse. While the use of polar co-ordinate mapping may have some computational advantage, simple

raster scanning is considered to be sufficiently straightforward to facilitate adoption and validation of the technique by others.. These are illustrated in Figures 1 and 2.

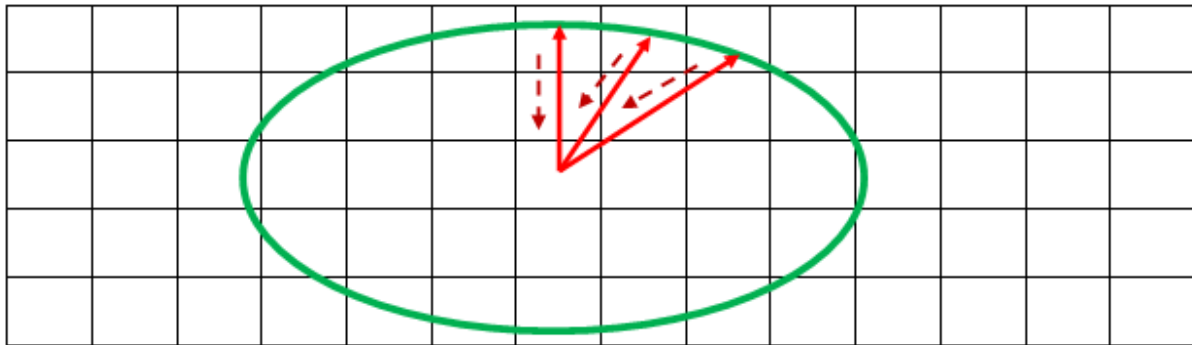


Figure 28 Scanning an ellipse "radially"

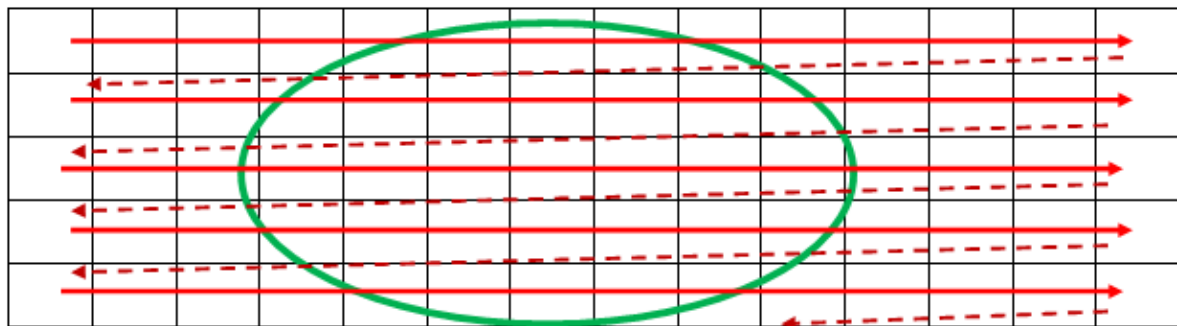


Figure 29 Raster scan to cover an ellipse

The FSV method uses a 5 pixel roll-off between the low and high parts of the filter. The mathematics of computing the distance from an ellipse, which needed to work in the general case, is computationally expensive.

This is a consideration because the FSV process involves filtering the data in the transform domain after one forward FFT, to produce the DC part, the low frequency part and the high frequency part. Direct solutions for the calculation of the distance from an ellipse to a point involve solving a quartic equation which is too expensive for a raster based method where each point will have to be processed. One solution is to approximate the roll-off using the equation  $z = \frac{x^2}{a^2} + \frac{y^2}{b^2}$ , using the value of  $z$  to calculate the roll-off, because inside the ellipse it will be less than 1, on the ellipse it will be 1 exactly, and outside the ellipse it will be greater than 1. The shape of this curve is shown in Figure 30



**Figure 30** The ellipse equation  $z = \frac{x^2}{a^2} + \frac{y^2}{b^2}$  used as a distance metric

A further approximation, and the one used in the Ruby implementation, was to consider the relative position of each point in the two dimensional space measured from the centre of the ellipse, and using polar co-ordinates, determine the angle  $\theta$  and then calculating  $x = a \cos(\theta)$  and  $y = b \sin(\theta)$  to work out the position of the nearest point on the ellipse to this point. Given high eccentricity in an ellipse is this will not be completely accurate, but as this is only intended to approximate a roll-off of 5 pixels, this is sufficient, given that previous work [143] has shown that FSV is robust to small changes in the number of points in the roll-off.

The boundaries of the roll-off were considered to be “concentric” ellipses, with major and minor axes in the same proportions, one 5 pixels smaller and one 5 pixels larger than the cutoff ellipse that has been calculated. The 5 pixels were taken as the distances on the major axis of the ellipse, and the minor axes of the larger and smaller ellipses were scaled to maintain that proportionality.

Once the distance from the cutoff was calculated, the actual scaling of the values was done linearly, according to the distance across the region where the roll-off occurs.

There is another difference between the one-dimensional case and the two-dimensional case which is fundamental to how the derivatives are calculated. The derivatives are well defined for the one-dimensional case. For 2D images there are a variety of possibilities. If the data is considered as being 4-connected, that is, diagonals can be ignored, then essentially there are just vertical and horizontal one-dimensional cases to consider. If the data is considered to be 8-connected so that diagonals count, then derivatives may be calculated in the same way that edges are found in image processing applications. This is done by convolving the image with a kernel, which is often a  $3 \times 3$  matrix, but theoretically may be larger. One such kernel is the Sobel filter [144]:

$$\begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} \quad (5.22)$$

which will give 0 for a constant (flat) input, but will enhance vertical edges. Similarly a horizontal version exists, and variants with more emphasis on the 4 connected points than the 8 connected points, reflecting the greater distances to the diagonal points, such as this horizontal edge detector:

$$\begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} \quad (5.23)$$

This matrix is usually convolved with the image, the middle pixel in the output being replaced with the sum of the products of the individual pixels with the corresponding numbers in the matrix, that is, a matrix product as that is normally understood is not used.

Given that the data had to be transformed by Fourier transform anyway, an alternative is to use the relationship that differentiation in the time domain may be accomplished by multiplication by  $j\omega$  in the Fourier domain. In the discrete case this becomes  $j(n\omega_1 + k\omega_2)$ , where  $n$  and  $k$  are the indices in the x and y directions. In the final implementation the derivatives were calculated in the Fourier domain and the Inverse Fast Fourier Transform was performed to create the derivatives in the domain of the original inputs.



## 5.4 A Mathematical Foundation for Higher Dimensional FSV

Derivatives are an important part of the Feature Difference Measure in Feature Selective Validation (FSV). Recording the local rate of change, they give a clue as to the roughness of the graph in a region, something engineers are interested in when comparing two traces. When the curves are smooth, the slope is important, and the second derivative distinguishes between peaks and troughs. With the extension of the domain to higher dimensions, i.e. more than one independent variable, it would be useful to explore the mathematics of differential operators in higher dimensional spaces. Such an analysis may lead to better explanations of how the algorithm performs, and illuminate possible avenues to explore or improve the technique, and its implementation.

To be able to consider derivatives in the FSV equations, the issue of sampled data must be considered. All the data obtained from instruments is sampled: in radio frequency; in the configuration space of the equipment, such as stirrer position controlled by a stepper motor; etc. In order for derivatives to have some meaning, it must be possible to treat the signal as being continuous.

In the discussion which follows, let  $f(\cdot)$  be an arbitrary function, let  $s(\cdot)$  be another function, for example, the signal. For experiments in electromagnetics this is often a function of radio frequency, which shall be written as  $f$ . However, for sampled systems, such as those to which FSV has been applied, there is also a sampling frequency. Let this be denoted by  $g_s$ .

A derivative is usually defined in terms of a local “slope”, where the region is as small as possible.

$$\lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h} \cong f'(x) \quad (5.24)$$

For the types of experiment under consideration the input data, coming as it does from a computer controlled chamber, is discretised, so that  $x$  is not continuous, and thus the signal is not differentiable.

The conventional engineering approach is that for a signal which is sampled at half the minimum period (that is, the sampling frequency  $g_s$  is twice the maximum frequency (data points per unit of axis)  $g$  of interest in the data) then the system is beyond the Nyquist limit, and it is possible to reconstruct the signal as follows [145]. Let

$$g(x) \stackrel{\text{def}}{=} \frac{\sin 2\pi g_s x}{2\pi g_s x} \quad (5.25)$$

i.e., a sinc function, then

$$f(t) = \sum_{n=0}^{\infty} f\left(\frac{n}{g_s}\right) g\left(t - \frac{n}{g_s}\right) \quad (5.26)$$

This recovers the continuous signal from the discrete signal, and this permits the differentiation of the signal.

$$\frac{df}{dt} = \sum_{n=0}^{\infty} f(n) \frac{d g(t - \frac{n}{g_s})}{dt} \quad (5.27)$$

Let

$$d(x) = \frac{\frac{\pi^2 x}{T^2 \cos(\frac{\pi x}{T})} - \frac{\pi}{T} \sin(\frac{\pi x}{T})}{\left(\frac{\pi}{T}\right)^2} \quad (5.28)$$

Then

$$\frac{df}{dt} = \sum_{n=0}^{\infty} f(n) d(t - kT) \quad (5.29)$$

Sampling Equation (5.29)

$$\begin{aligned} \frac{df}{dt}[n] &= \left. \frac{df(t)}{dt} [n] \right|_{t=nT} \\ &= \sum_{n=0}^{\infty} f(n) d(t - kT) \\ &= \sum_{n=0}^{\infty} f(n) d_T(t - k) \end{aligned} \quad (5.30)$$

where  $d_T(\cdot)$  is the T sampled sinc derivative.

Consider the transformation to the sampled domain from the continuous domain. If it can be shown that the sampled signal is equivalent to the continuous one, then it is possible to transform back and forth between the two representations with impunity. Let  $X = x - lT$  and  $x = nT$ . Then

$$\frac{\sin\left(\frac{\pi X}{T}\right)}{\frac{\pi X}{T}} = \frac{\sin\left(\frac{\pi}{T}(x-lT)\right)}{\frac{\pi}{T}(x-lT)} \quad (5.31)$$

$$= \frac{\sin((n-l)\pi)}{(n-l)\pi} \quad (5.32)$$

$$= \delta(n-l) \quad (5.33)$$

It must be emphasized that this only works for the T sampling sinc function.

For multidimensional cases, the derivative is replaced by the gradient, which is the vector of partial derivatives along each axis. With similar assumptions about sampling a continuous function, and meeting the requirements of the Nyquist sampling theorem, it is possible to obtain (in two dimensions for example)

$$f(u, v) = \sum_{n=0}^{\infty} \sum_{m=0}^{\infty} f(n, m) \cdot g(u - nT, v - mT) \quad (5.34)$$

$T$  is the sample spacing, (here assumed to be the same on both axes, but this makes little difference), and

$$g(x, y) = g(x)g(y) \\ = \frac{\sin\left(\frac{\pi x}{T}\right)}{\frac{\pi x}{T}} \frac{\sin\left(\frac{\pi y}{T}\right)}{\frac{\pi y}{T}} \quad (5.35)$$

Because of the sampling theorem it may be assumed that both sides of (5.34) are differentiable. Without loss of generality, consider the partial derivative with respect to  $u$ :

$$\frac{\partial f(u, v)}{\partial u} = \sum_{n=0}^{\infty} \sum_{m=0}^{\infty} f(n, m) \frac{\partial}{\partial u} g(u - nT, v - mT) \quad (5.36)$$

The definition of the derivative of the discrete function may be arrived at by sampling both sides of Equation (5.36):

$$\begin{aligned} \frac{\partial f(n, m)}{\partial u} &= \left. \frac{\partial f(u, v)}{\partial u} \right|_{u=nT, v=mT} \\ &= \sum_{n=0}^{\infty} \sum_{m=0}^{\infty} f(n, m) \frac{\partial}{\partial u} g(u - nT, v - mT) \\ &= \sum_{n=0}^{\infty} \sum_{m=0}^{\infty} f(n, m) d_T(u - n) g(v - m) \\ &= \sum_{n=0}^{\infty} \sum_{m=0}^{\infty} f(n, m) d_T(u - n) \delta(v - m) \end{aligned} \quad (5.37)$$

Let the following operators be defined:

$$D_x(\cdot, \cdot) = \frac{\partial}{\partial x}(\cdot, \cdot) \quad (5.38)$$

$$D_y(\cdot, \cdot) = \frac{\partial}{\partial y}(\cdot, \cdot) \quad (5.39)$$

And these operate over the sampled space.

It is now possible to define an operator which is a combination of these two to give an overall derivative operator. So, for some vector  $\bar{\alpha}$  representing the “slope”, it can be said that

$$D_{\bar{\alpha}} f(n, m) = \alpha_x D_x f + \alpha_y D_y f \quad (5.40)$$

It is then possible to proceed to obtain a second derivative operator:

$$D_{\bar{\alpha}}^2 f(n, m) = (\alpha_x D_x + \alpha_y D_y)^2 f(n, m) \quad (5.41)$$

$$= (\alpha_x^2 D_x^2 + 2\alpha_x \alpha_y D_x D_y + \alpha_y^2 D_y^2) f(n, m) \quad (5.42)$$

All the tools necessary to compute the FSV components in an N-Dimensional space are now available..

It may prove fruitful to examine other ways of considering the derivatives.

Let  $\bar{y} = f(\bar{x}_i): i = 1 \dots N$ , where  $\bar{y}$  is a vector, and where  $f(x_i)$  is a continuous representation of the signal. The derivatives can be formed with the Jacobian matrix

$$J(\bar{x}) = \begin{pmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \dots & \frac{\partial y_1}{\partial x_N} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} & \dots & \frac{\partial y_2}{\partial x_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y_N}{\partial x_1} & \frac{\partial y_N}{\partial x_2} & \dots & \frac{\partial y_N}{\partial x_N} \end{pmatrix} \quad (5.43)$$

which may be considered to be the result of a differential operator  $D_j = \frac{\partial}{\partial x_j}, i \in 1..N$  operating on  $\bar{y}$ .

If  $y$  were a scalar, then applying this operator again would give the Hessian matrix

$$H(\bar{x}) = \begin{pmatrix} \frac{\partial^2 y}{\partial x_1^2} & \frac{\partial^2 y}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 y}{\partial x_1 \partial x_N} \\ \frac{\partial^2 y}{\partial x_2 \partial x_1} & \frac{\partial^2 y}{\partial x_2^2} & \dots & \frac{\partial^2 y}{\partial x_2 \partial x_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 y}{\partial x_N \partial x_1} & \frac{\partial^2 y}{\partial x_N \partial x_2} & \dots & \frac{\partial^2 y}{\partial x_N^2} \end{pmatrix} \quad (5.44)$$

The Hessian matrix is the second derivative, describing whether there is a local maximum or minimum. However, since  $\bar{y}$  is a vector quantity in the more general case considered here, a different approach is necessary.

A vector is one case of a tensor, a tensor of order 1. Tensor notation has the advantage that summation over subscripts is implied by their repetition; this is known as the Einstein summation convention [146]. For example, in tensor notation, divergence is expressed as:

$$\nabla \cdot A = \frac{\partial A_i}{\partial x_i}. \quad (5.45)$$

The Laplacian  $\nabla^2 A$  may be obtained [147] from

$$\nabla^2 A = \frac{\partial^2 A}{\partial x_i \partial x_i} \quad (5.46)$$

Thus we have the means to obtain  $Lo_1'(x)$ ,  $Lo_2'(x)$ ,  $Lo_1''(x)$ ,  $Lo_2''(x)$ ,  $Hi_1'(x)$ ,  $Hi_2'(x)$ ,  $Hi_1''(x)$  and  $Hi_2''(x)$ . These are sufficient to compute FSV, by direct extension from the 1-dimensional case and to be able to do so with a neat mathematical representation. If experiment shows that engineers examine other properties in high dimensional data sets, such as divergence and curl, these may be expressed in tensor form, and the approach would need to be further refined.

Having these derivatives presents the possibility of modelling the data using a Taylor approximation of  $f(\bar{x})$ , by considering a small offset from  $\bar{x}$ ,  $\Delta\bar{x}$ , which is also a vector.  $f(\cdot)$ , itself, may be a vector. Considering a scalar function  $g$  of the scalar  $y$ , the Taylor series is of the form:

$$g(y + \Delta y) \approx g(y) + \Delta y g'(y) + \frac{\Delta y^2}{2!} g''(y) + \dots \quad (5.47)$$

In our case all these terms would be vectors, so care would need to be taken over the tensor products chosen to achieve this. For tensors, there are Kronecker products and inner products to consider.

This would allow certain types of interpolation in which derivatives are used. There is a need for interpolation in the FSV method when the data points of the input sources are not sampled over the same points. Then only the region of overlap is processed, and interpolation is done to ensure that the points are coincident. The use of the derivatives in the FDM equations occurs after the data have been split into DC, low and high frequency parts, so computing the derivatives as is done at the time of interpolating the inputs may have to be repeated. However, the filtering in the Fourier domain is linear, so there may be a time domain form of the filter which could be calculated. Since the choice of parameters for this filter would require knowledge of the calculated cut-off, it seems that this would be more computationally expensive than re-computing the Taylor series for the DC, Lo, and Hi datasets themselves after the Fourier filtering is done.

With an equation approximating the data sets, it should be possible (for small data sets) to compare the results of numerical computation of the FSV with symbolically computed analytical solutions, bearing in mind the smallest practical data size for a two-dimensional FSV based on the Fast Fourier Transform is 32 by 32 points. This size limit is imposed by the FFT needing a power of two points to process, the DC terms removing the first 4 points, and the cutoff having to extend over 5 points. The derivatives, if calculated directly rather than in the transform, need three points, and there is positive and negative frequency to account for, so 15 must be doubled, giving 30 points, the next power of two being 32.

## 5.5 Two Dimensional Feature Selective Validation Results

The two dimensional version of Feature Selective Validation is demonstrated below as a series of figures corresponding to the elements in the calculation. The results were modelled as 3D Alias Wavefront .OBJ files and displayed in 3D graphics programs, Meshlab [148] (a tool developed with the support of the 3D-Coform [149]), and Anim8or [150], because this allowed free manipulation of the objects. The cross sectional graphs are provided in order to provide some scale to the images, lacking in the 3D displays. These are based on 1D input data sets extruded to be 2 dimensional in order to permit comparison with the original 1D results.

Figure 31 and Figure 32 show the two input signals.

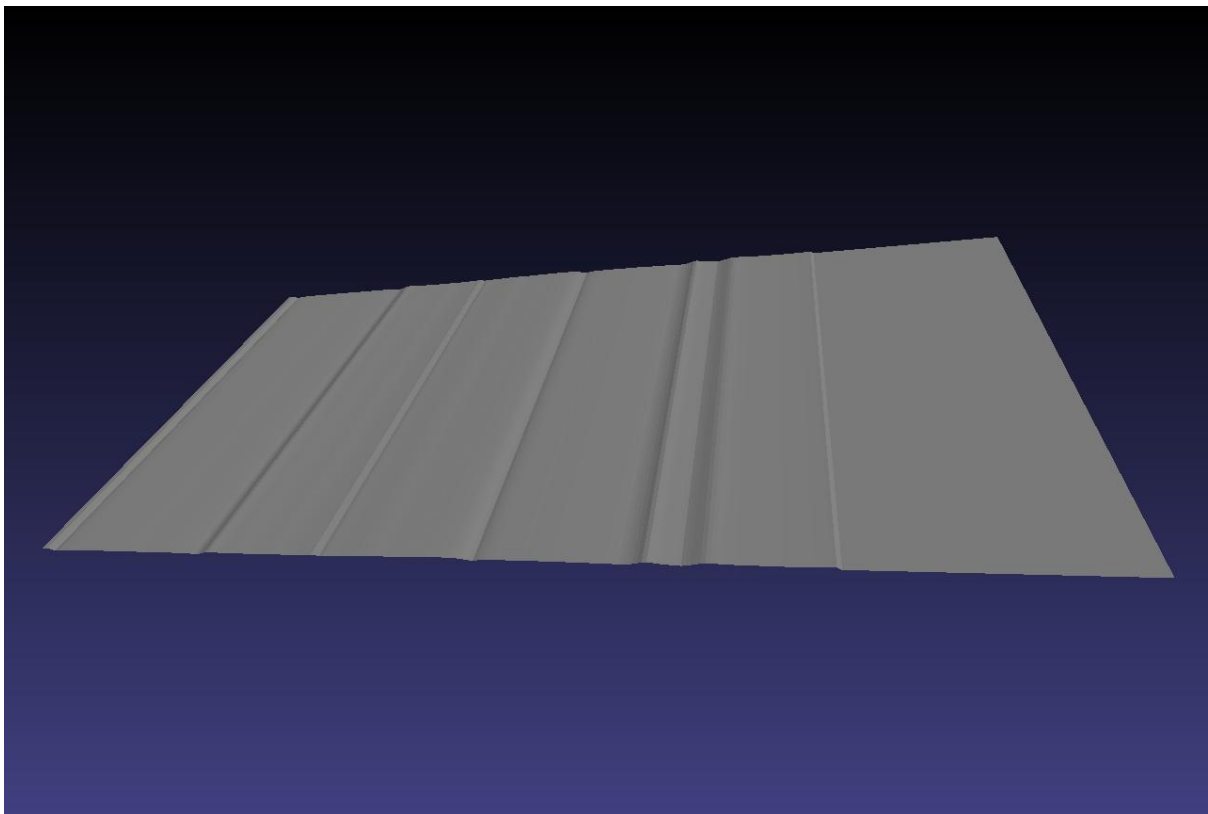
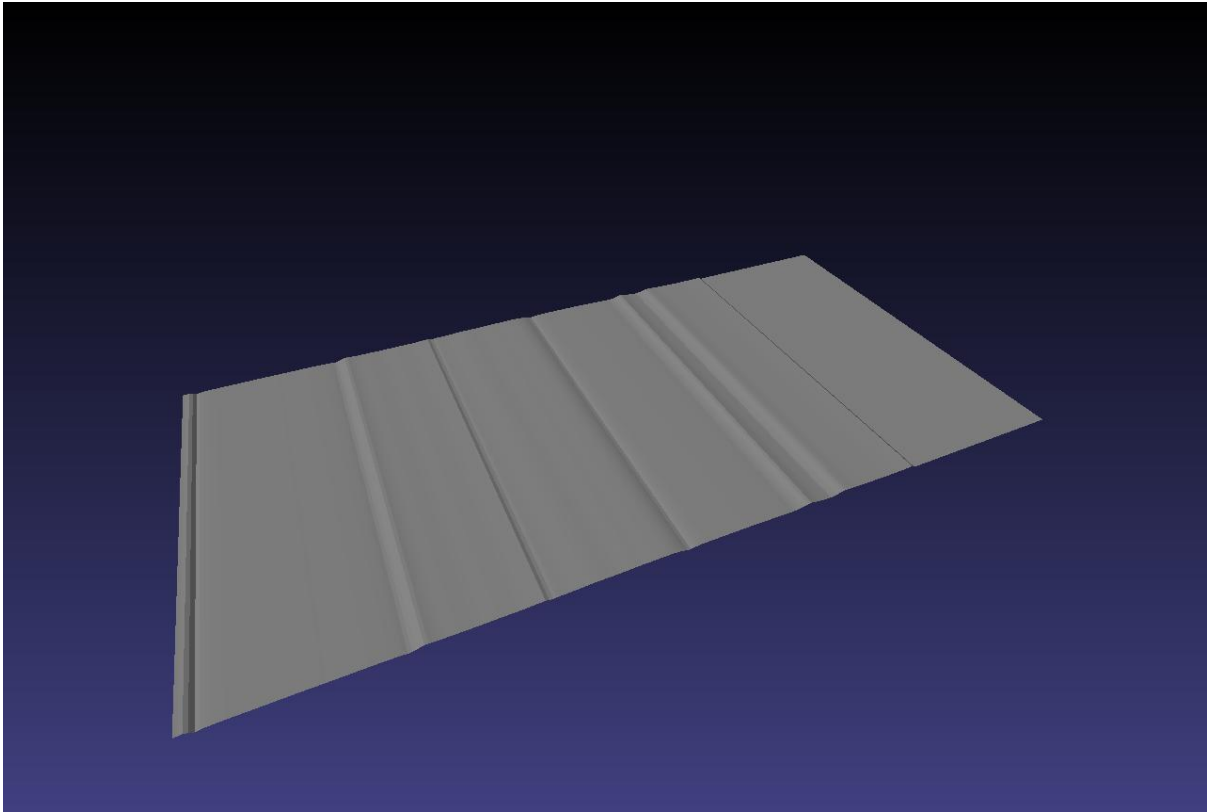


Figure 31 input 1



**Figure 32 Input 2**

The results after filtering are shown in the following figures, DC in Figure 33, Figure 34, Figure 35, Figure 36 (as a 3D display and then a cross-section in each case), then the low frequency part (i.e. the bandpass filtered part) is shown in Figure 37, Figure 38, Figure 39, Figure 40.

Finally the high frequency components are depicted, again as 3D images and then as cross sections, in Figure 41, Figure 42, Figure 43, Figure 44.



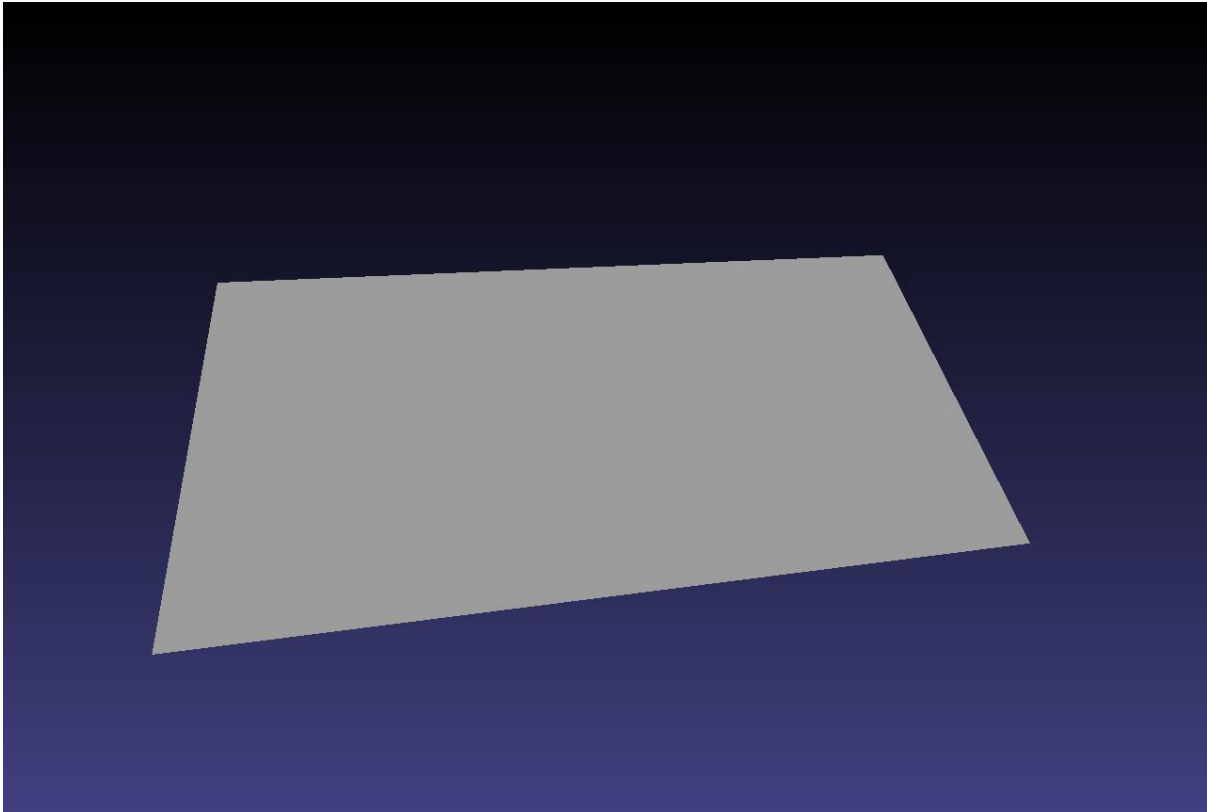


Figure 33 DC for input 1

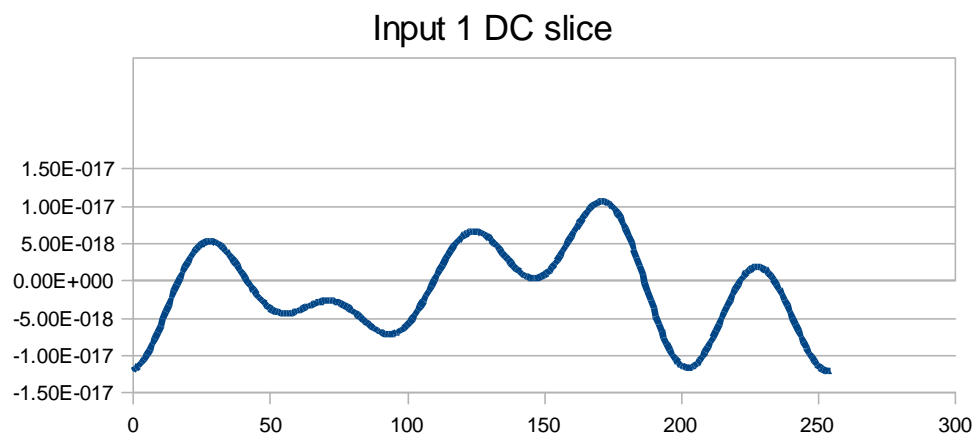


Figure 34 A slice through the DC for input 1 results to show scale

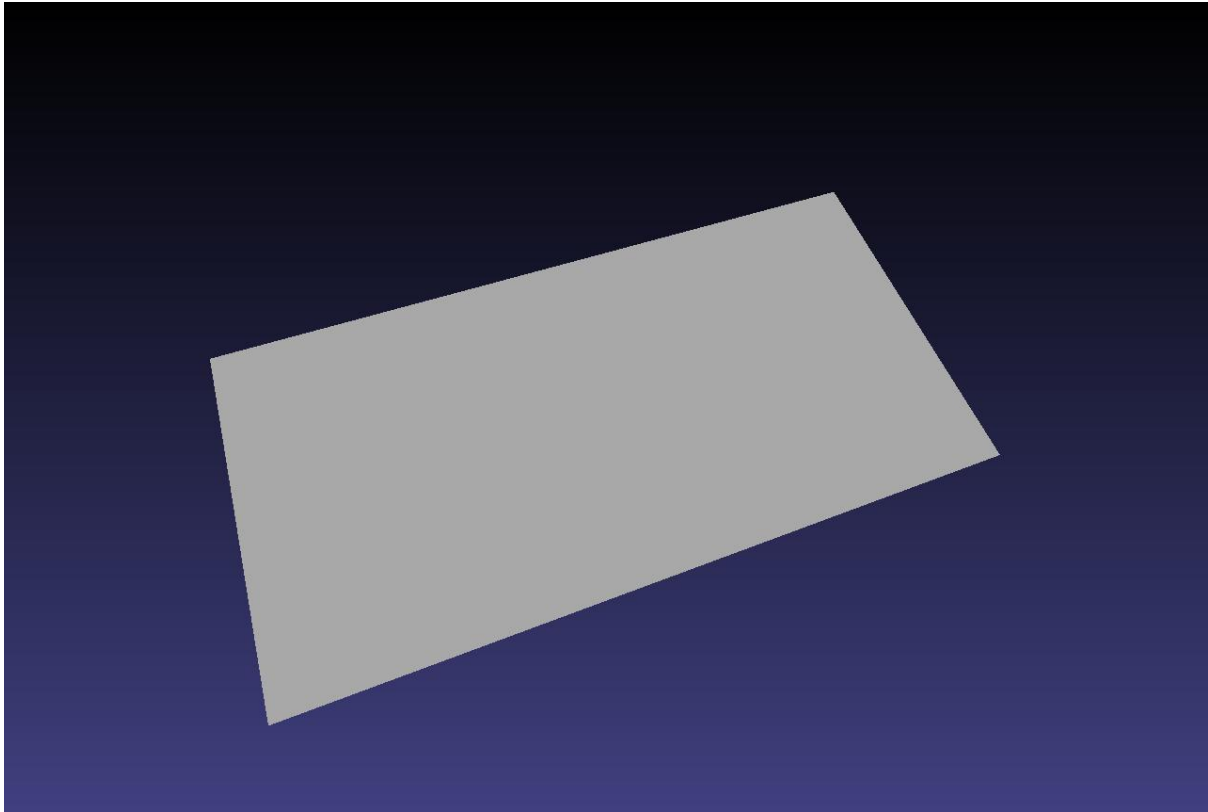


Figure 35 DC for input 2

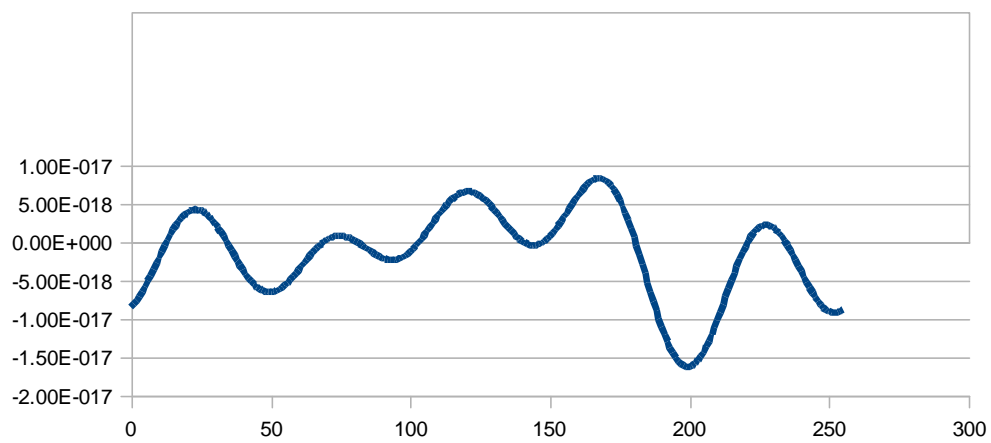


Figure 36 DC results for Input 2 (slice)

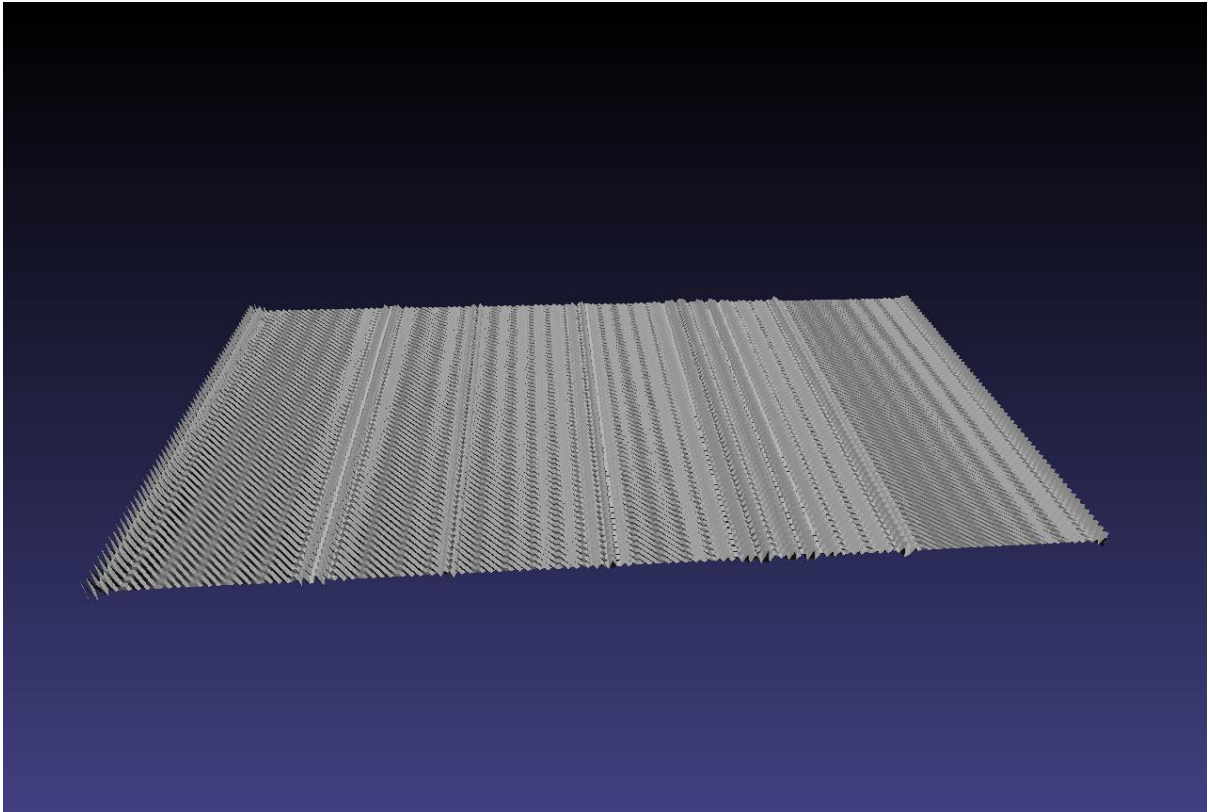


Figure 37 Low frequency part of input 1

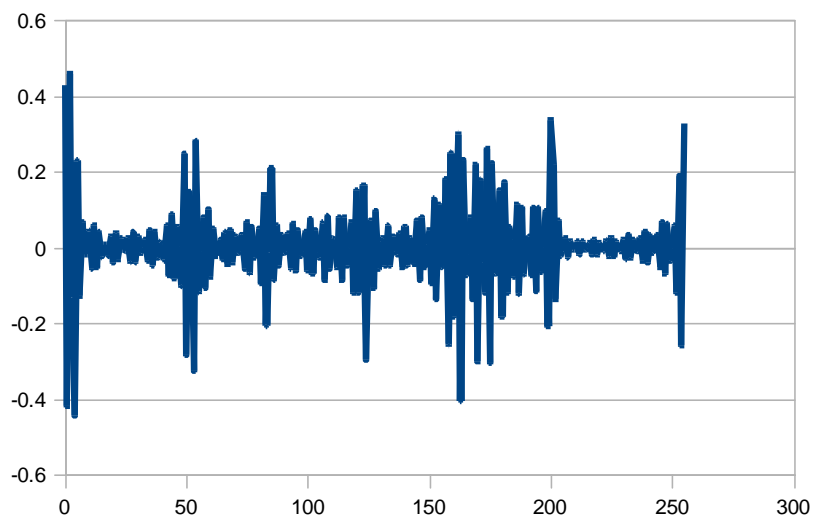


Figure 38 Low frequency part of input 1 (slice)

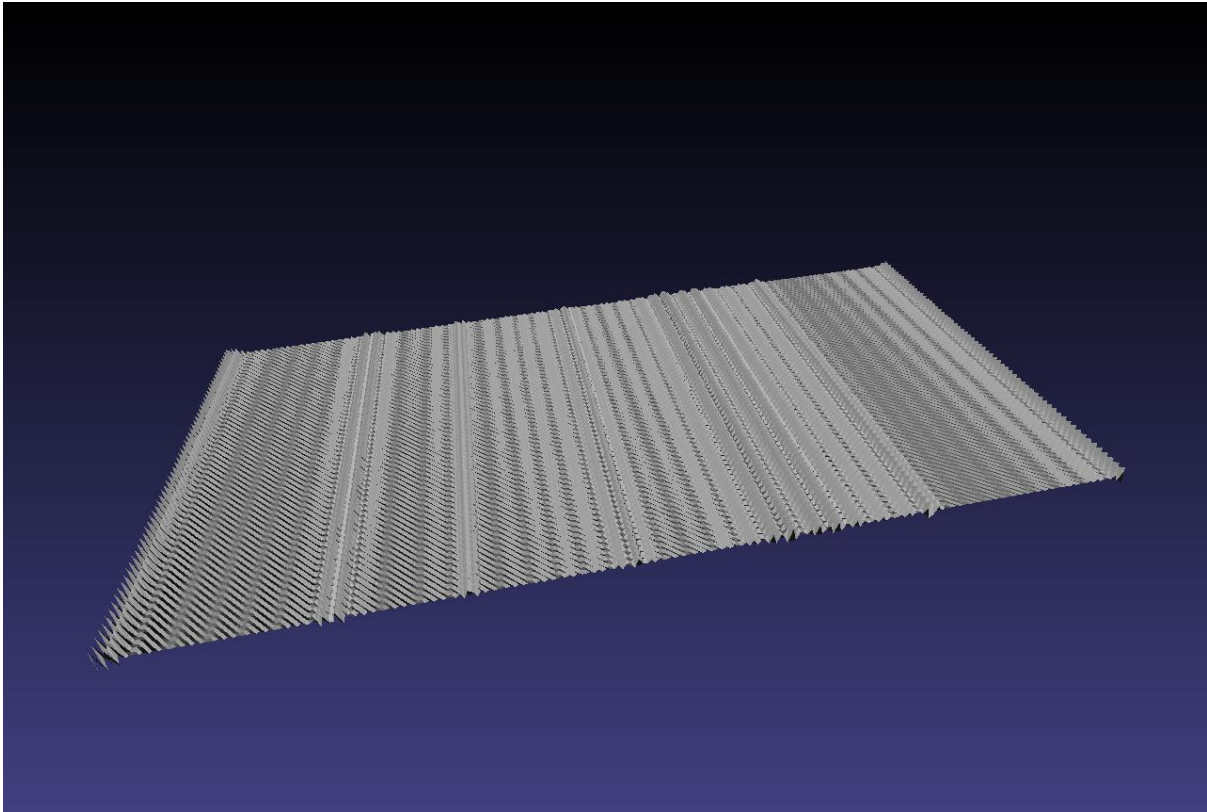


Figure 39 Low frequency part of input 2

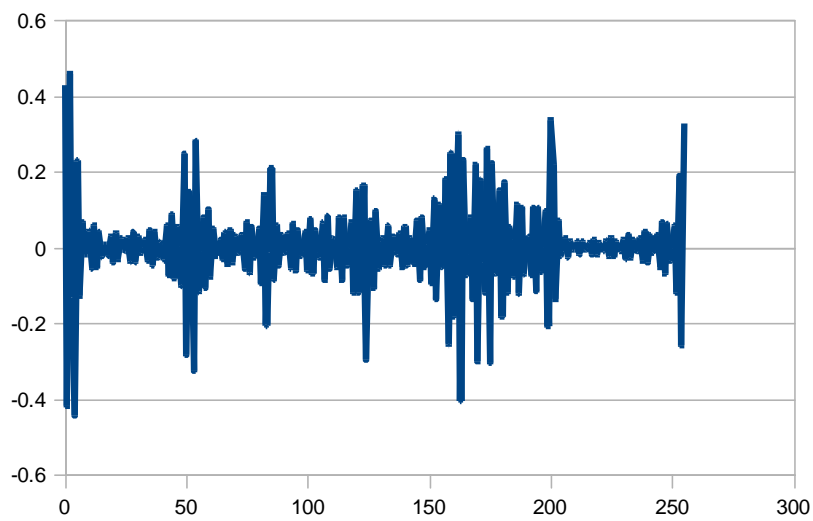


Figure 40 Low frequency part of input 2 (slice)

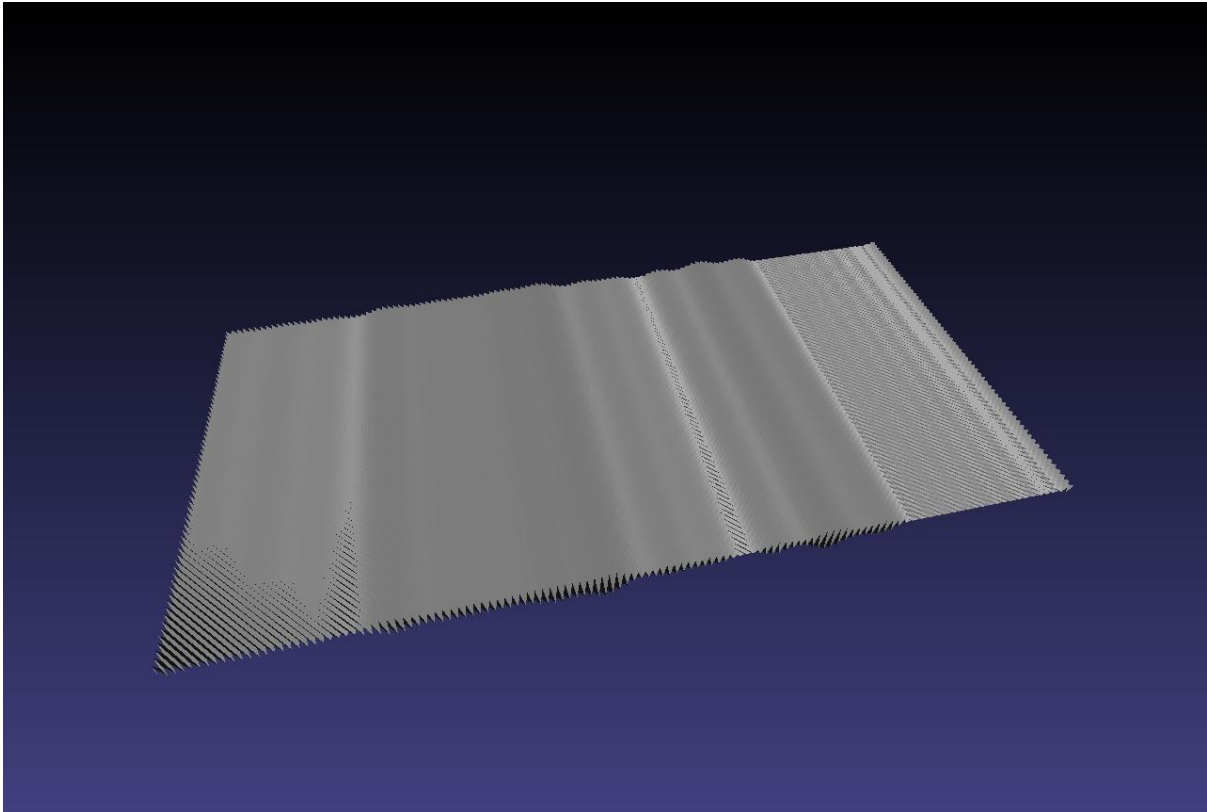


Figure 41 High frequency part of input 1

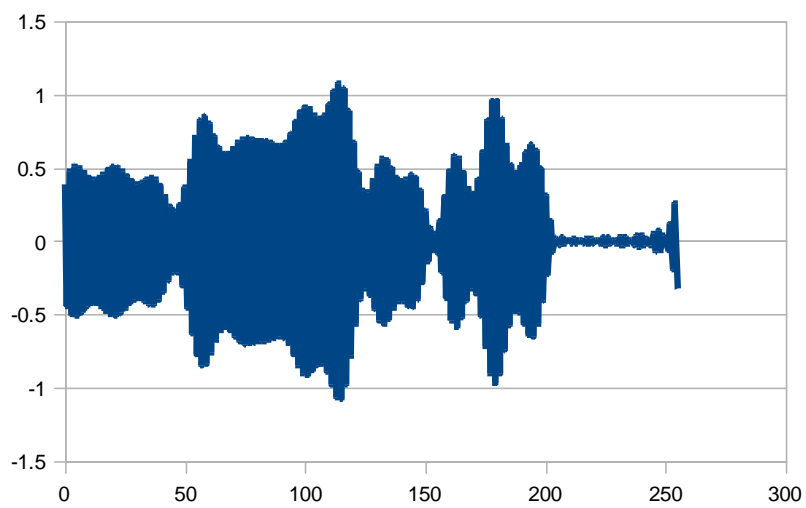


Figure 42 High frequency part of input 1 (slice).

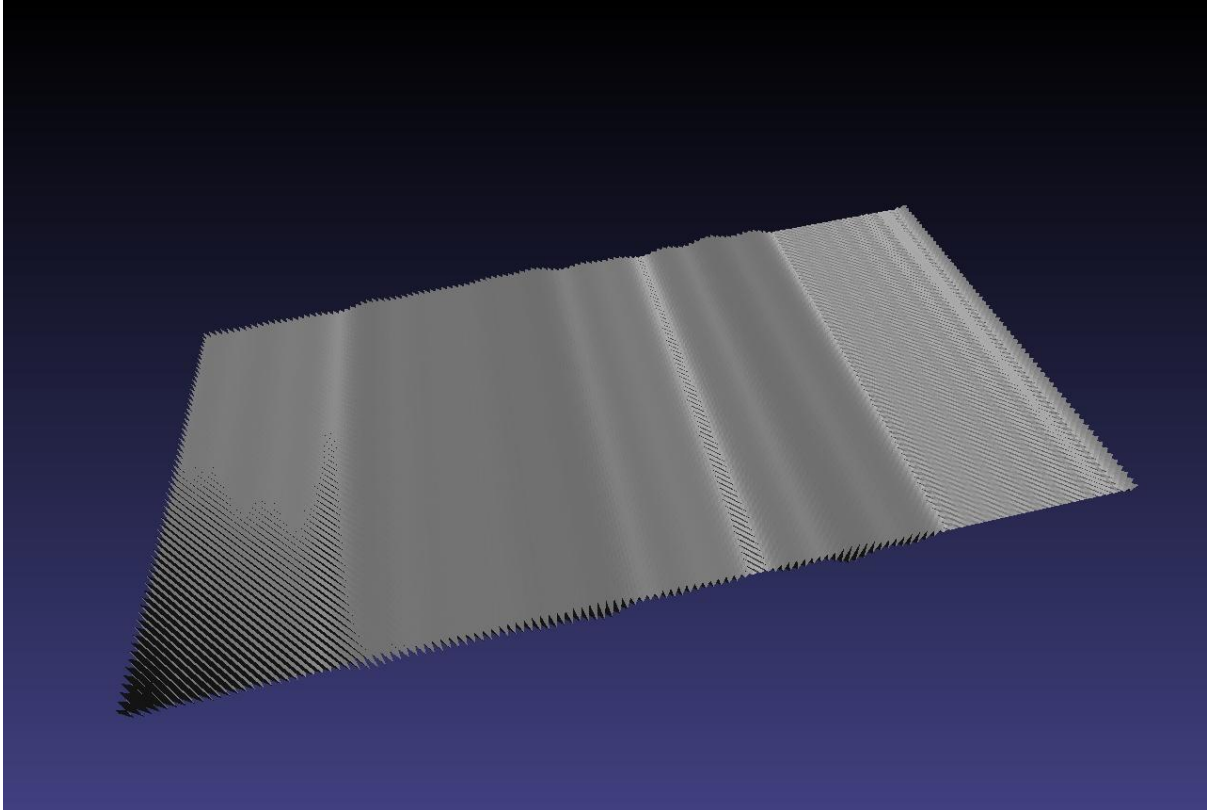


Figure 43 High frequency component of input 2

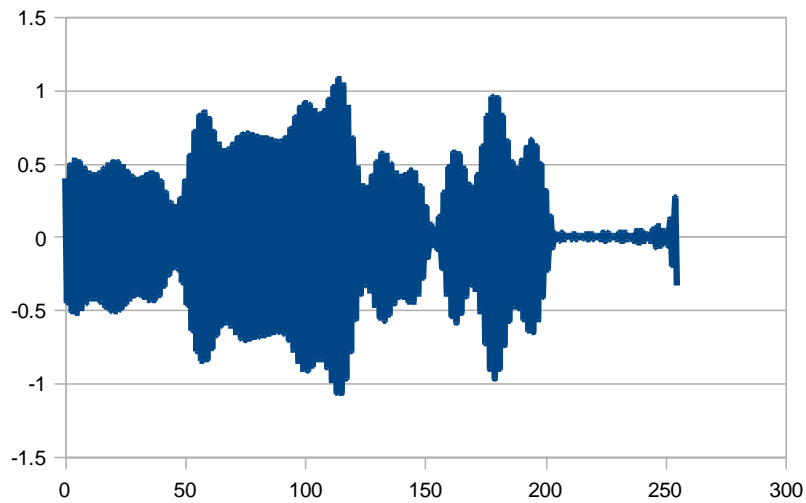


Figure 44 High frequency part of input 2 (slice)

The computed FSV values follow: the ADM in Figure 45 and Figure 46; the FDM in Figure 47 and Figure 48; the GDM in Figure 49 and Figure 50; and then the confidence histograms in Figure 51, Figure 52 and Figure 53. The final figures of merit are shown in Figure 54.

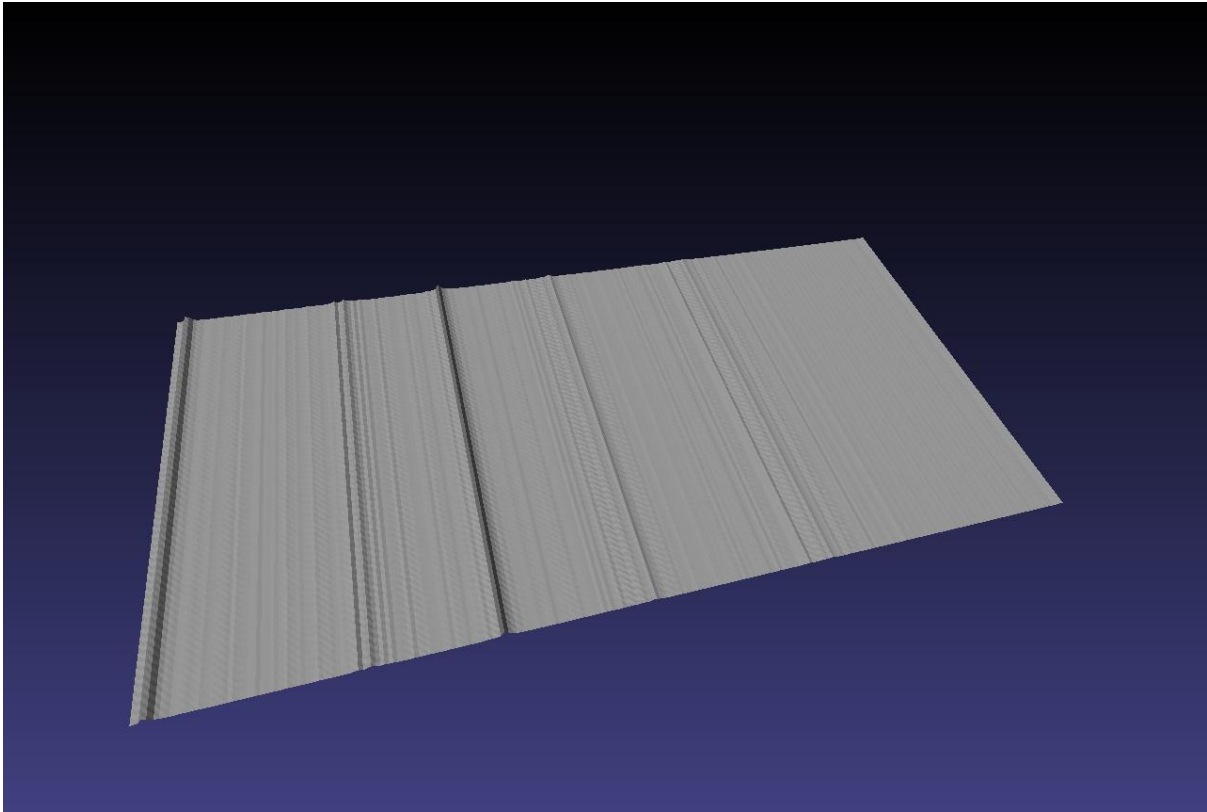


Figure 45 Pointwise ADM for inputs 1 and 2

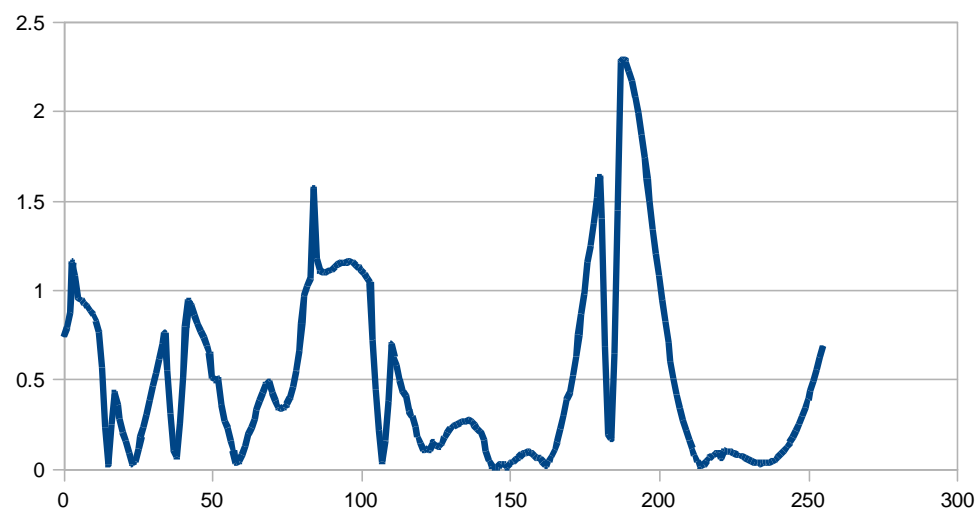


Figure 46 Pointwise ADM for inputs 1 and 2 (slice)

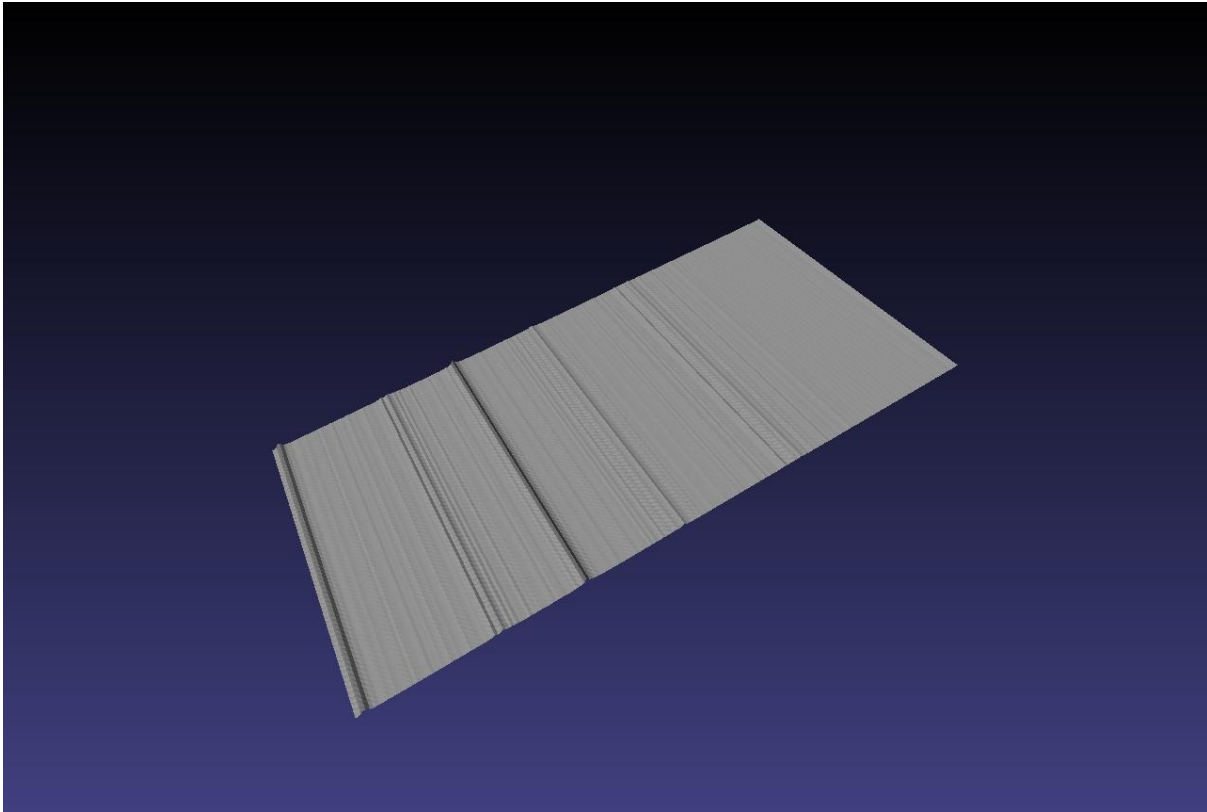


Figure 47 Pointwise FDM for inputs 1 and 2

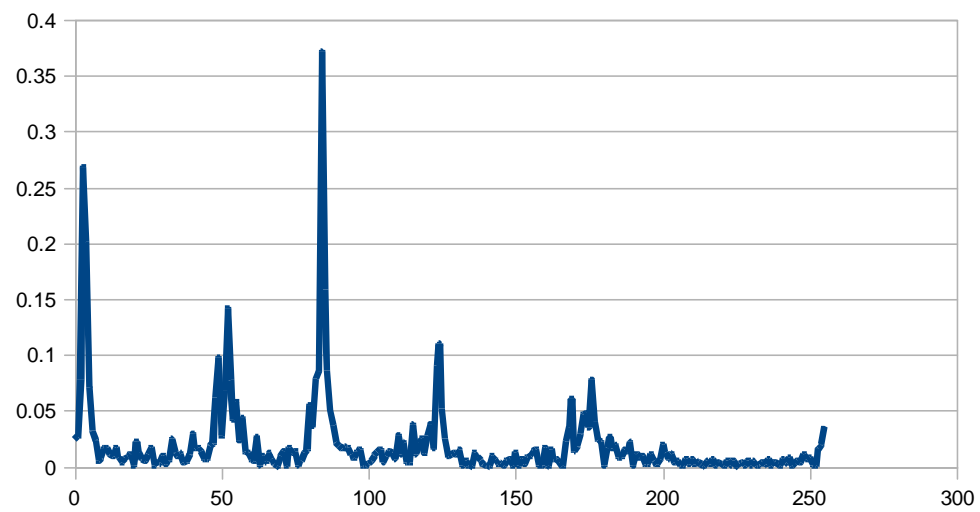


Figure 48 Pointwise FDM for inputs 1 and 2 (slice)



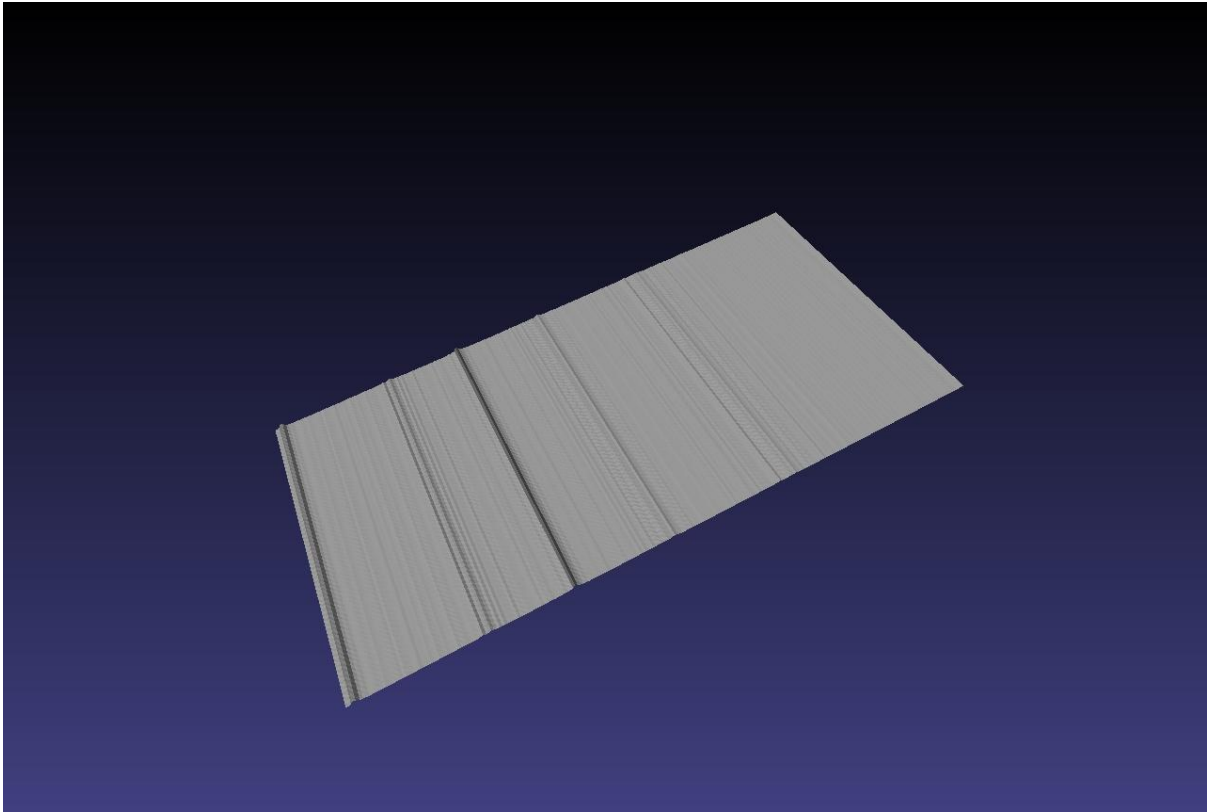


Figure 49 Pointwise GDM for inputs 1 and 2

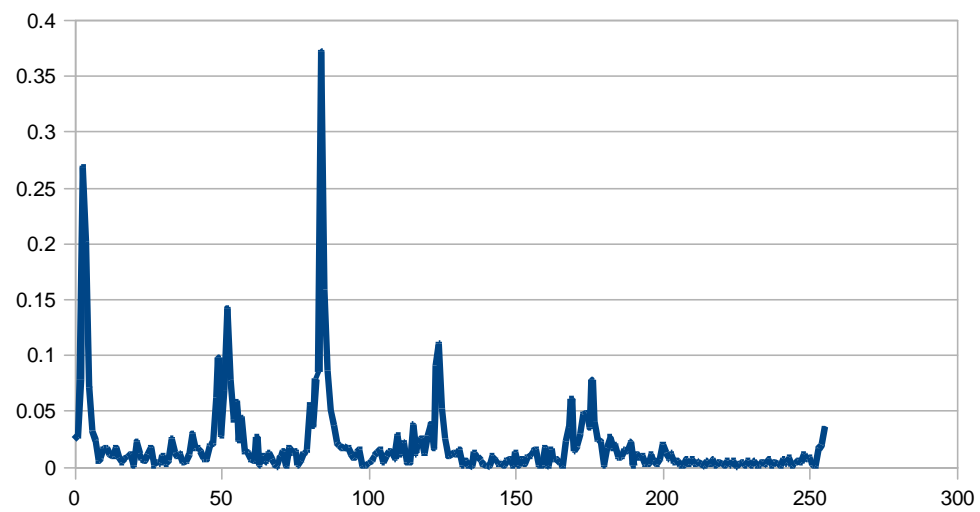


Figure 50 Pointwise GDM for inputs 1 and 2

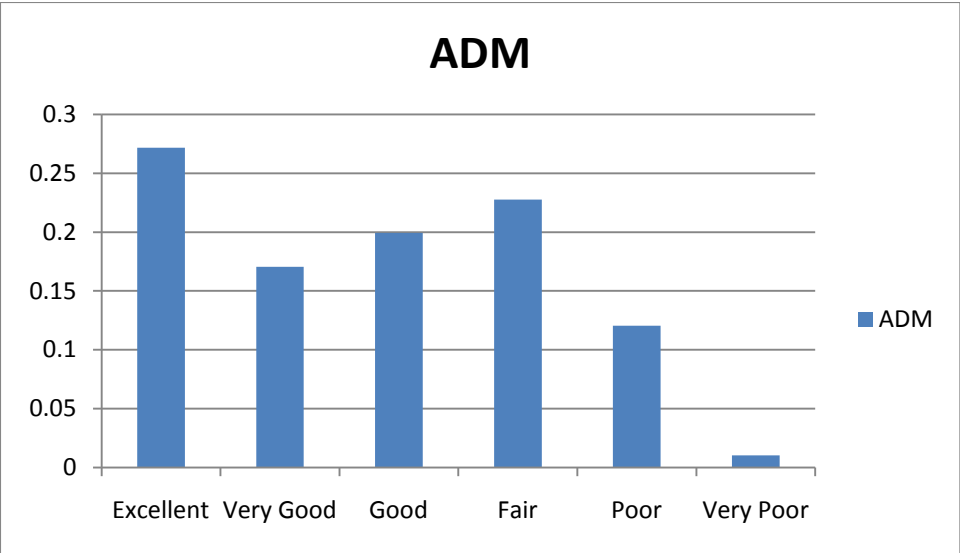


Figure 51 ADM confidence histogram for inputs 1 and 2

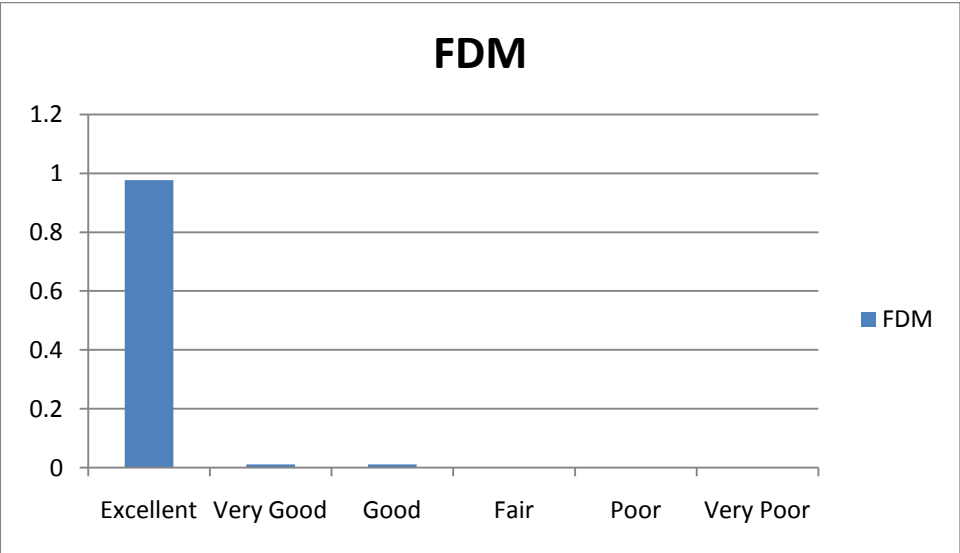


Figure 52 FDM confidence Histogram for Inputs 1 and 2

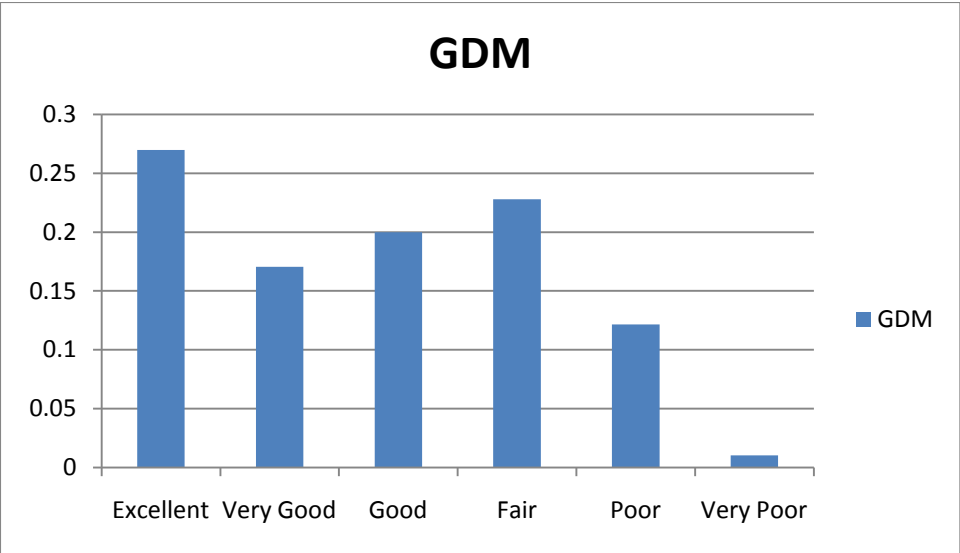


Figure 53 GDM Confidence Histogram for Inputs 1 and 2

ADM	0.38
FDM	0.02
GDM	0.38

Figure 54 Final figures of merit for ADM, FDM, GDM for inputs 1 and 2

These were for 2D data derived from a 1D data set, to permit comparisons to be made. The results for the 1D data sets follow in Figure 55 and Figure 56.

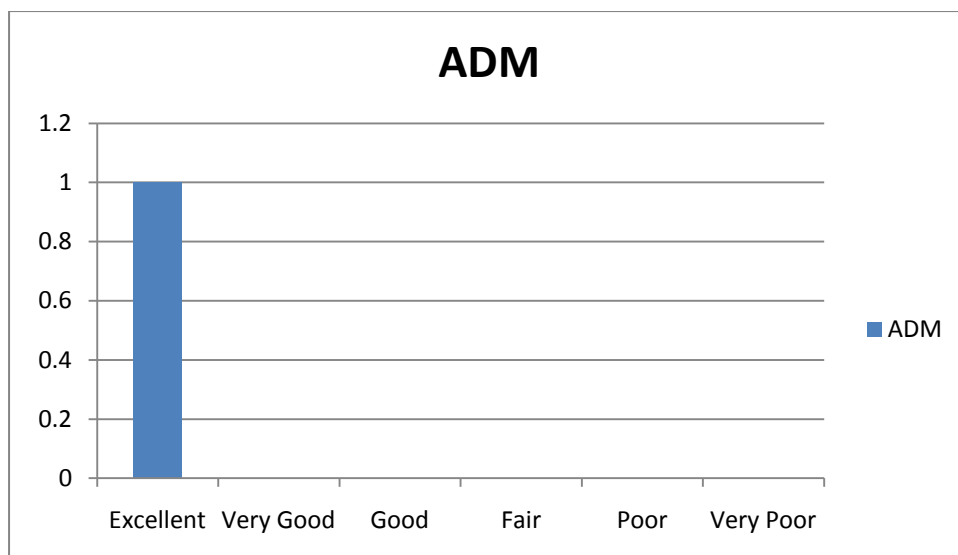


Figure 55 ADM Results for 1D FSV from the original 1D data

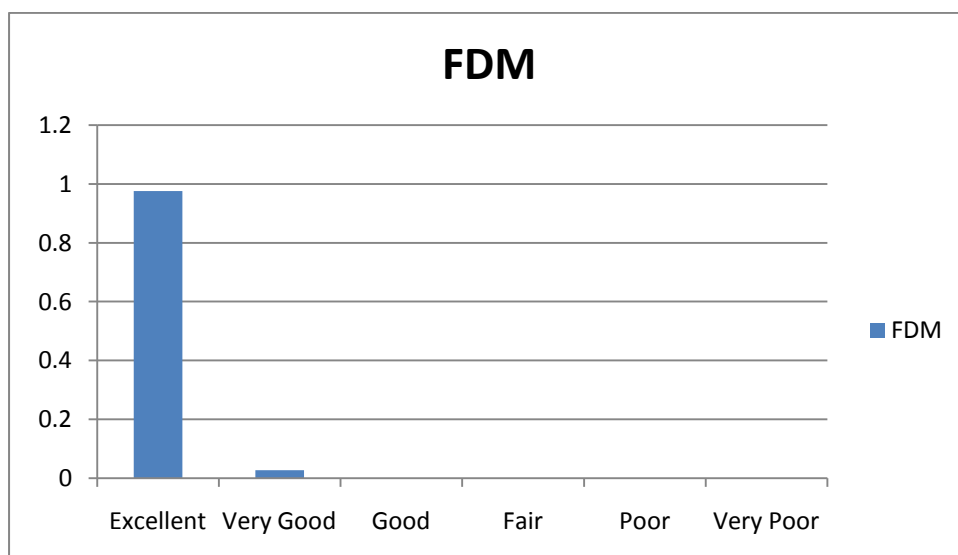


Figure 56 FDM Results for 1D FSV from the original 1D data

The ADM was larger than expected. This underlines the need, as discussed in Future Work, for evaluating the user interpretations of 2D data in order to tune the coefficients used in the FSV formulation.

# 6 Discussion

## 6.1 Introduction

The fundamental objective of this thesis is to present research, and draw together a number of inter-related topics of current importance in the modelling of novel RF structures. This chapter discusses this work and investigates topics of contemporary interest which have been discovered in the process of undertaking this research.

In this chapter the modelling of computer cables and the work on genetic algorithms will be explored, followed by the work of parallel transmission line modelling, and then Feature Selective Validation. Finally the interconnections between these themes will be discussed.

## 6.2 Genetic algorithms and computer modelling of cables.

The application of genetic algorithms to the computer modelling of cables was novel when originally carried out and has demonstrated the extreme design limits for cables optimised for shielding, demonstrating the effectiveness of the counter-intuitive design involving extremely low optical coverage [126]. The use of colour maps to display three properties over a two dimensional area to demonstrate optimisation output is also a novel contribution to electromagnetics practice.

The number of variables involved in creating a genetic algorithm provides scope for considerable exploration of how these techniques may be applied. The number of chromosomes selected to breed, the population size, the way in which chromosomes are selected, and whether there is any benefit in having a random chromosome inserted at each generation, all interact with each other in ways which are difficult to predict intuitively. The algorithm itself is drawn from nature, and the variety of reproductive strategies so far discovered therein suggests that there is no single “best” way to create a genetic algorithm. In the end, the engineer must choose a strategy which appears to give satisfactory results. The genetic algorithm is sufficiently robust that it can cope with errors in implementation, a poor choice of random number generator, and even a rather ineffective selection strategy, and yet yield useful results. The desire to optimize an optimisation algorithm appeals in the same way as recursion does, but such an abstraction should have some utility in speeding up the process of automated design.

The particle swarm optimizer has been shown to be a useful and intuitive algorithm for solving electromagnetic design problems. There would seem to be scope for further work to investigate the merits of both systems but, this thesis suggests that the selection of technique is a matter of user

preference, because the results seem comparable with those from the genetic algorithm. The work on using particle swarm optimizers in cable design appears to be novel and the results have been used commercially.

Provided a means can be found to express a problem in terms of optimising some function these techniques may be applied to the design of more complex structures. For example, a number of properties may be chosen when judging the quality of an antenna, such as gain, bandwidth, and the complexity of the structure itself. This latter property could be chosen to reflect the ease of manufacture of the antenna.

### **6.3 Parallel TLM modelling**

The development of a parallel TLM modeller raised a number of interesting design issues. Initially, the need to use existing hardware, as and when it became available set up a set of constraints different from those of others working in this field, for example [44] which assumes all tasks will complete successfully. The current implementation has shown that the need for portability has been met, and the system has been shown to work on both PCs and Unix Systems. The initial choice of a scripting language to implement the design can be improved in the future, particularly, now that Ruby has newer, faster implementations. Implementation based on the java virtual machine will allow distribution of the software in binary form, as Java bytecode, which opens up the possibility of commercialisation of the software. Clearly more work is needed before this software could be released as a commercial product; there are issues in the security aspects and the user interface which are obviously not at a commercial stage of development. At the very least it will be necessary to construct a parser for some kind of input language, to facilitate the construction of TLM models. However, these are implementation issues rather than research issues, and a certain amount of work has been done on this already.

The drive towards parallelization has been propelled by the need for fast computation. The present TLM model does not incorporate multigriding which would help increase performance for structures with areas of fine detail. Again, this is an implementation issue. There has been considerable work in this area, and whilst has not been possible to reap the benefits of this work during this programme of research it is clearly desirable that the parallel modeller be extended to include multigriding, thin wires, and lumped elements, if commercialisation is ever to be pursued.

The distribution of the computation over the Internet raised issues relating to the security of the computational system as a whole. The desire to avoid cryptography constrained the solution to the use of cryptographic check sums. This was preferable to the implementation of a cryptographic system which was only apparently secure because of the introduction of unknown weaknesses. Mistakes which are all too commonly made include: “security through obscurity”, the belief that

hiding information means that it cannot be discovered; creating one's own algorithm on the premise that well known algorithms will have well known weaknesses (a fallacy because there will be weaknesses which a creator of a new algorithm will not have thought of, which will be known from other algorithms), or possibly the choice of a known algorithm which is insufficiently secure (for example the cipher in the Enigma machine is too weak now). Furthermore such a system would require some practical solution to the key distribution problem. If someone determined to attack the system were able to get hold of one of the keys then an attack would be indistinguishable from a genuine attempt to use the system. The subtleties used in defeating ciphers are such that the development of a secure system should be considered a topic for future work.

## **6.4 Two dimensional feature selected validation**

The evaluation of designs necessitates comparisons. However, for example, when the goal is to achieve a certain field pattern within a structure such as a mode stirred chamber, then we are faced with the problem of comparing data sets with two or more independent variables (dimensions). Understanding these comparisons is, at present, a poorly understood area, particularly in terms of the human psychology. The difficulties of comparing one dimensional data sets have already been demonstrated, and features selective validation has described in IEEE 1597.1 has shown its utility in overcoming many such problems for the one dimensional case. In this thesis this technique has been extended to the two dimensional case. The use of an elliptical shaped filter to account for different sizes in the two directions is novel, and a necessary precursor to expansion of FSV into higher levels of dimensionality. However, at this stage the association between the natural language descriptors used in the standard 1D case and the numerical results generated by the 2D implementation must be regarded with some caution. There is a whole body of work that must follow this which simply addresses the question of how engineers can and do 'look' at high dimensionality data. Given how poorly one-dimensional comparisons were understood in the original development of FSV, it is clear that two-dimensional comparisons will be considerably more difficult. Because engineers are not used to performing such comparisons as single comparison events, we lack even the terminology to describe the features often found therein. It is possible to borrow terminology from topography; however knowledge of physical geography is not a prerequisite for engineering courses, so one cannot assume a common vocabulary. Furthermore geographical features are often tied to their formation (such as a glaciated valley) or to their physical properties (oxbow lake). Even for one-dimensional sets, when the data gets noisy, we are reduced to using terms like "grassy" to describe what we are seeing. In order to provide a heuristic for two dimensional comparisons which makes sense to people in the same way that one dimensional features selective validation makes sense to people, the psychology will need to be much better understood. This includes the way the information is

perceived by those comparing it and also the psycholinguistic factors that come into play during the decision process. It may be that a whole new vocabulary will have to be defined in order to allow intelligent debate about numerical results over a two dimensional domain.

Mathematically, it makes sense to extend the comparison approach to higher dimensions, as previously alluded to. In the case of a mode stirred chamber the "dimensions" of the problem include the spatial dimensions of the chamber, the position of the stirrer, and the frequency, and this is before any properties of the equipment under test have been considered. So, even for such a seemingly innocuous situation an 11 dimensional comparison could be required (three components each of E and H fields over three spatial directions over a given frequency range and over a number of stirrer steps). Representing data over a two dimensional domain is challenging enough at present, and moving to a three dimensional domain adds new layers of complexity. For a species evolved in a three dimensional universe (ignoring time, and setting aside the work of string theory for the time being) imagining higher dimensions and designing ways to display them is nontrivial. Finding terminology to describe the features in such a space, especially in such a way that agreement can be reached between groups of engineers, will be a major research topic for some time to come.

The description of the FSV method in more rigorous mathematical terms is a worthwhile goal because it will facilitate discoveries about identities comparable to "multiplication in the time domain is the same as convolution in the frequency domain" for Fourier analysis. It has been shown that it makes sense to consider the data as approximated by a Taylor series, using the Jacobian matrix and the Hessian Matrix, and that for vector fields, unless considering each component in turn as a function of all variables, an approach based on tensors seems worthy of investigation. Also, the derivatives that are included in the Taylor series will be filtered by the Fourier domain operations, and that the properties of linearity in the Fourier domain may lead to a more analytical grasp of the whole process.

## **6.5 Optimisation, Modelling, and Validation as a Toolkit**

The components developed during this programme of research complement each other as part of a framework for developing physical layer models. The design of the parallel solver was such that the parallelism was achieved by a component which could be re-used in the other parts of the toolkit. Clearly the different chromosomes in the genetic algorithm could be stored in a tuplespace, and the fitnesses be evaluated in parallel, but similarly the particle swarm could be shared among machines using the same techniques. Furthermore, in the development of the two-dimensional Feature Selective Validation algorithm, the Fast Fourier Transform plays an important role, and the 2D implementation of this is clearly amenable to parallelisation, and this could be achieved using the



same tuplespace model. The various components of the FSV calculations could also be shared amongst machines,.

In order to make this into a completely unified toolkit, more work will be required on the interfaces between the different components. This will take more work because the limitations of each form will have to be characterised in order to provide a consistent Application Programming Interface (API) which is simple enough to be useful, and yet flexible enough to be able to handle the problems which are likely to be of interest.

# 7 Conclusions and Future Work

## 7.1 Introduction

It seems to be in the nature of research that when exploring a topic a large number of avenues are opened up. Sometimes these are interesting curiosities, and sometimes they appear to be major topic areas which would certainly merit deeper exploration. The purpose of this chapter is to discuss the possible avenues for further work which have appeared during the course of this research.

This chapter first draws conclusions from the research and then describes work which could immediately follow the research discussed in this thesis, and finally goes on to describe longer term research topics which may be investigated.

## 7.2 Conclusions from the Research

The novel aspects of the research are: the introduction of colour maps into cable design as a means of displaying up to three variables over a two dimensional domain; the development of a parallel TLM solver which did not depend on specialised hardware, with a design suited to institutions where the availability of computer power is not exactly predictable and the use of encryption is inappropriate, and where it is expected that clients of the server will fail to deliver computational results from time to time; the introduction of particle swarm optimisation and genetic algorithms to the communications cable design community; the development of a model of swarming where the speed of the particles is limited by drag forces to ensure coverage of the domain in detail; the development of two-dimensional feature selective validation with filters which were elliptical, to cover non-square domains; and the application of vector calculus to provide the beginnings of an analytical basis for the FSV method. The research has opened up a number of avenues for further exploration. Most of these novel aspects will require further development to reach their potential.

## 7.3 Mainstream Future Work

This section will concentrate on future work which is a direct continuation of the work already carried out. It will cover areas which are obviously needed in order to further the work. Work which is related, but not necessary to continue the research will be reserved for a later section. This section will be broken down into the broad topic areas reflected in the structure of this thesis.

## 7.3.1 Parallel TLM

### 7.3.1.1 Security Concerns

One of the weaknesses of the parallel TLM system is its susceptibility to attack by those on the Internet with malicious intent. Given that a goal of the project was to produce a system which would allow many people to contribute computer time to the modelling of electromagnetics problems, the use of the Internet is essentially a prerequisite; a private network which is secure is not really an option in this case. Therefore any vulnerabilities must be addressed by means of protocols or algorithms.

The obvious approach of using strong cryptography has the disadvantage that users whom we made trust working in countries which the United States do not trust would be excluded from contributing to the system because of the export laws affecting cryptographic software. If one is prepared to accept that loss than the technical difficulties of correctly implementing cryptographic algorithms come into play. This requires a solid grounding in cryptography, code-breaking, and cracking (what many journalists referred to as hacking, a term which many programmers prefer to use for recreational software development or less structured program development with no implication of malicious intent). The reason for this is that there are subtle ways to break cryptographic algorithms. For example, if it is known that an algorithm depends on a random number generator to generate cryptographic keys, and if it is suspected that this generator uses the linear congruential algorithm, and this turns out to be correct, then a weakness has been exposed. The linear congruential algorithm relies on multiplying one number, which is initially the seed, by a constant, and forming the modulus with respect to another constant. This means that for a given seed the numbers generated will be a repetitive sequence, and also that if the current random number can be determined as well as the two constants, then the whole of the sequence can be generated. This effectively gives away your keys. A clear example might be a poker program using such a random number generator in which knowledge of five cards allows the sequence of all cards dealt to be known [151]. There are also problems with ciphers based purely on the exclusive or operation. If a key is regarded as a string of bits and a bitwise exclusive operation is performed with the "plain text" to produce the cipher text (the encrypted message) then an attacker with access to the system as a "black box" can perform the "encryption" operation using the cipher text and the plain text to get the key. This is because if  $A \otimes B = C$ , then  $B \otimes C = A$ , and  $A \otimes C = B$ , where  $\otimes$  represents the exclusive or operation and where  $\{A, B, C\}$  are bits. The reversibility of the exclusive or operation makes it useful as a component in symmetric ciphers where the same key is used to encrypt the information as is used to decrypt it. It is also useful in generating a pseudorandom bitstream as a key for encryption by using a binary stream and some known plain text. This plain text becomes the password which generates the key which is

actually used in the encryption process [151]. Clearly it is important to treat such a system as a “one time pad” where the plain text used to generate the key is not reused, for this reason.

Another kind of attack on the parallel TLM system would be an attempt to get hold of the actual simulation data by attacking the central server which controls the parallel operation. This is a traditional form of cracking, where one is attempting to break into a computer. The security considerations are probably more to do with firewall configuration, auditing, recognition of attack patterns and so forth [152]. To assume that this data is not of sufficient interest to others for them to want to steal it is to assume that one’s work is not of great importance, hence, by contradiction, this type of attack should be taken seriously.

At present no consideration has been given to detection of faked results by code maliciously designed to damage the TLM simulation. The only obvious means to overcome this at the moment would be to ensure that work is guaranteed to be replicated across servers, some of which can be trusted. This would seem to run counter to the idea that contributors are welcome. At present people can only add their machines if they are invited, so the trust element is human mediated. There are difficulties with having  $k$  malicious machines in a set of  $n$ , for  $k < n$ , because if a simple majority is used to define truth, then if that majority is picked from the  $k$  machines then a false result will be accepted as true. If trusted machines can be used to test other machines to see if they are trustworthy, then those marked as untrustworthy can be ignored. Lying part of the time will be sufficiently destructive and sufficiently difficult to detect to make this worthy of research in itself.

For the current implementation the key exchange problem still exists. That is to say for the communication to be established in a secure way between the server and client, the key is for encryption must somehow be passed to the client. This needs to be done in a secure way. At present it is assumed that the key will be passed by some other means than using the Internet, entirely because of this problem. Solutions like public key infrastructure should make this practical. Perhaps there are other means of achieving this.

Distributed Ruby, in recent releases, provides for access control lists (ACLs) to enhance the security of the system. What this can provide, and how practical ideas to implement, needs to be investigated further, though it is known to limit access to trusted machines.

### **7.3.1.2 Acceptable Simulation Times.**

It would be very useful to gather evidence of what is an acceptable simulation time. As machines increase in capacity, we can simulate more detailed models. But people have been working with simulation times of days, so that is in some sense manageable. What is the trade-off for complexity against time that people are willing to accept? When does a simulation take so long that the problem is regarded as intractable? People always want a simulation to be faster, but an answer to this would

allow a useful trade-offs to be calculated. Factors that would affect this include time pressure such as time-to-market on a product, the number of computes available to simulate the problem, how long it takes to develop and verify a model prior to running the simulation.

### **7.3.1.3 Further Development of the TLM Solver.**

The system developed was based upon TLM. However the parallel code, which distributes the work between machines, was not written with any particular modelling technique in mind. It is therefore possible to change the modelling technique to method of moments, finite difference time domain or whatever. This would facilitate validation of models. Given the importance of validation in the research, specifically the place of Feature Selective Validation, this would be an important step to take in furthering the research.

The current implementation of the TLM modeller lacks many facilities available in commercial systems. For example, there is no multigriding, thin wires, correctly implemented perfectly matched layers, and moving data into and out of the system is tedious at present. A parser for the input language has only been partially implemented, and for serious amounts of work to be done this needs to be finished. Facilities such as better graphics often come at the price of portability, but perhaps the use of a web based interface would overcome this.

To ensure the effectiveness of the parallel implementation, and as a means of detecting and removing bottlenecks, it is important that a large scale test be done across tens of machines, for workspaces segmented in many different ways, and with different workspace sizes. The performance measures would provide some guide as to how many machines it is sensible to use for a given model size. It would permit one to determine whether compression of the data on the network is a critical need or not. (Compression is assumed to be important in the present implementation, and this is partly to do with concern for other users of the limited bandwidth available.) It would permit a study of how the segmentation of the workspace into pieces for each machine to operate on affects performance. Clearly these effects would interact with the implementation details, particularly the speed of the code itself, and so an extensive study would be necessary.

### **7.3.1.4 Visualisation tools for the solver**

At present there is no toolkit for visualising the models input to the solver, and also no tools for visualising the resulting fields. In terms of practical verification and for the most basic form of validation of the results, visualisation tools would be an important development. With recent developments in gaming technology it should be possible to develop a system which is particularly intuitive to use.

### **7.3.2 Optimisation**

Much of the research in Chapter 2 discusses population sizes of around one hundred individuals, be that for Particle Swarm Optimisation (PSO) or for Genetic Algorithms (GA). In nature, such a small population size would be considered endangered. It would be useful to do more research on a wide range of population sizes for various problems, as well as the various strategies (velocity modifications for PSO, number of breeders and selection strategy for GA) to try to optimise the optimisation algorithms for quality of results and for speed. Beyond this, the algorithms are fairly robust, so there seems little else to change because the strategies generally result in good solutions. It would seem to be sensible to attempt to use an optimisation algorithm to try and improve the optimisation algorithms.

### **7.3.3 Satisficing**

A topic of interest has been satisficing, which in some ways may be regarded as the antithesis of optimisation, in so far as it is about the termination of a search for better solutions when one which is good enough has been found. In terms of simulation, this might correspond to stopping the simulation when we have enough information to know that a proposed solution is better or worse than another, for however we are defining better or worse (which is clearly problem dependent). This has the advantage that it saves on time, which has been an important goal of most of the work carried out in this research programme. This is a topic which is worth exploring further, and also ties in to the theme of how to perform effective comparisons on results, or in this case, partial results.

### **7.3.4 Mechanical Properties of Communication Cables.**

The mechanical properties of a cable are more difficult for an electronic engineer to consider, and these would make a good goal for further work. The consideration of weight and minimum bend radius as obvious starting points, but properties such as fragility under rough handling would be worthy of consideration. These would clearly be affected by dimensions of the physical materials used in the design.

### **7.3.5 Feature Selective Validation**

#### **7.3.5.1 Interpolation in FSV, and the filter cutoff**

Once a Taylor approximation has been constructed for the data, derivatives will have been obtained. The Fourier filtering is a linear process, so it should be possible to determine how the Taylor series is affected by the filtering directly, which would save having to recalculate the derivatives after the filtering is complete. However, at present the cutoff of the filter is a function of the input data sets, and the roll-off is fixed. These were chosen as heuristics because they worked. For the roll-off, work discussed earlier shows there is some latitude in how that is chosen, and a roll-off more suited to an

analytical approach may be more appropriate. If the basis of FSV can be given a more analytical footing, it may lead to a better understanding of the decision making process.

Other possibilities for interpolation exist, such as that based on a Lagrange polynomial to approximate  $y = f(x)$  thusly:

$$L(x) = \sum_{i=1}^N y_i l_i(x) \quad (7.1)$$

$$l_i(x) = \prod_{j=1, j \neq i}^N \frac{x - x_j}{x_i - x_j} \quad (7.2)$$

This could be extended to higher dimensions by holding the indices of all but one dimension constant, as is done for surface plots. This approach requires no differential operators. It would be worth comparing the approaches, and seeing if there are better approaches.

### 7.3.5.2 Investigation of Wavelets as a basis for FSV

The author believes that wavelet theory will prove to be very useful when trying to understand why FSV is successful where it is. In wavelet theory a signal is decomposed into component waves which are not purely sinusoidal as in the Fourier transform, but the shape of the wavelet reflects the structure of the signal. However, these waves are dilated so as to provide low resolution and high resolution information about the signal. The importance of this is that one of the ways people discuss graphs in comparisons is by observations that shifts and stretches applied to regions of one graph would make it fit the other one, which was discussed by Anthony Martin and Alistair Duffy in connection with the one-dimensional FSV work. This author believes that a wavelet model should support this viewpoint well. In the fast wavelet transform this scaling is based on powers of two, so initially the single is broken down into a high frequency part and a lower frequency part (roughly speaking). The wavelet coefficients are computed by differences between neighbouring points and from the average of all points at each scale [153]. The similarity between these techniques and the mathematics of Reature Selective Validation seem to the author to be very striking.

### 7.3.5.3 Filtering in the Fourier Domain

The need to process rectangular data sets led naturally to the extension of a 1 dimensional filter as an ellipse in the two dimensional case and an ellipsoid in the N-dimensional case, with differing dimensions of its principal axes reflecting that of the data set. However, if augmenting zeroes were added to the input data sets for each dimension less than the maximum, then the size of the Fourier domain would be the same in all directions (a square, a cube, a hypercube), allowing the use of circular filters. This would simplify the calculations for the cut-off and roll-off because the distances from the edge of the circle or (hyper-)sphere would simply be Euclidean distances from the centre,

The differences between the results from the two-dimensional FSV and the one dimensional FSV may be due to what is and what is not included by the elliptical filter. It is more likely to be a result of the small numbers and rounding errors in the ODM calculation. However, it would be instructive to compare an implementation of two dimensional FSV with augmented zeroes with one implemented with elliptical shaped filters, to see if that is a contributor to inaccuracies. It would also be instructive to implement FSV using an arbitrary precision arithmetic library to see whether that improves the comparison with 1D FSV in either case. These libraries have the disadvantage that they slow the calculations down, but this would be a necessary trade off against more accurate results.

### **7.3.6 Human perception.**

Work needs to be done for the two dimensional FSV to replicate and extend the studies which were performed for the one dimensional FSV. It will be necessary to see how engineers examine two dimensional plots, and to perform statistical studies to help the selection of the numerical constants in the equations, so decisions agree with those made by engineers. To enable the work to move to higher dimensions, and to some extent for the two dimensional case, work will need to be done to determine the correct way to display the datasets. Given the varieties of display for groups of one dimensional data sets (bar charts, pie charts, box plots, line graphs, scatter plots and the rest), and that the choice is known to affect how well the information is conveyed, it seems clear that in the case of two or more dimensions, the possibilities will be more numerous. Maintaining simplicity and clarity with so much more information will not be easy. It may be that displays will need to become interactive, and more like immersive gaming environments, for a full appreciation of the data to be reached.

Eye tracking studies will be essential to the study of visual displays, but an exploration of surround sound is of interest. The blind author of [154] describes how rain illuminates his surroundings with different textures of sound, giving a sense of space otherwise difficult to achieve. It would be interesting to determine to what extent the different senses affect the perception of data of three or more dimensions.

There will also be a need for clearer terminology, as understanding grows which features engineers find important. Methods of display will affect how well such features are perceived and described by the users.

### **7.3.7 Mathematical Vulnerabilities.**

The FSV method breaks down mathematically under some circumstances. If both inputs are 0 there is a division by zero in the Lo and Hi calculations. The derivatives break down for very noisy data; in the one-dimensional implementations this has been considered to be due to needing more points than the feature size to calculate the derivatives. Computing them in the Fourier domain may help with



this, because differentiation is performed by pointwise multiplication by  $j\omega$ . The effects of this kind of noise need to be explored in the two-dimensional case. When there is such high frequency noise in the traces, the one-dimensional plots (functions of one independent variable) end up as regions on the graph, rather than lines. There may be some merit in using the plot as a signal with a two-dimensional domain, where there is a 1 where there is data plotted, and a 0 elsewhere. This will allow these noisy regions which end up as areas to be compared more directly. Clearly a larger (or more detailed) plot would overcome this, but this brings in important questions about how much detail is deemed to be important, and having chosen the level of detail, how comparisons can be carried out despite this noise.

## **7.4 Research likely to be of Interest in the Longer Term**

### **7.4.1 Parallel TLM Solver**

#### **7.4.1.1 Use of GPU for parallel TLM**

Directly related to this is the trend to move towards the use of GPU boards for general purpose computing. These graphics processor units consist of massively parallel processors with impressive performance for handling rendering of visual scenes. Their development has been driven by the market for computer games, with their increasingly sophisticated rendering of graphics. In some sense this is essentially electromagnetics modelling. Since TLM at least is a matrix based method, this should map well on to the vector type operations involved in ray tracing computations which make up the core of the functionality of these boards.

#### **7.4.1.2 Centralised control is a form of weakness**

At present the distributed system depends on one machine holding the tuplespace in which the jobs and results are stored. The hardware failure of the system, or a malicious attack on it would result in the destruction of the work being undertaken at the time. It would be worth exploring mechanisms to provide redundancy for the server itself to overcome this weakness. Perhaps developments in peer to peer networking and ad hoc networks would facilitate this.

#### **7.4.1.3 Better use of protocols**

The present implementation of network based parallelism uses distributed ruby on top of TCP/IP. Given the variety of protocols there have been created for use on the Internet, including realtime audio, realtime video, and UDP, it seems possible that a more efficient implementation of parallelism could be achieved by a specific protocol, or by better use of the features of TCP/IP.

#### **7.4.1.4 Development of a screensaver**

As a possible way to increase the number of participating machines it would be worth investigating the practicality of making this into a screensaver. In this way uptake might be encouraged like it was for the seti@home project. According to their website, the seti@home project is based upon BOINC [155], which faces many of the issues of computing distribution discussed earlier. BOINC does seem to depend on the use of compiled languages, so consideration would need to be given to programmer effort if this approach were used to continue this work.

#### **7.4.2 Feature Selective Image Validation.**

##### **7.4.2.1 Comparison**

FSV is based on how engineers compare graphs of numerical data. Images intended to be viewed solely as pictures may be considered to have a two dimensional domain (the area of the picture) and a one-dimensional (monochrome), three-dimensional ({red, green, blue}, or {hue, saturation, brightness}) or possibly four-dimensional ({cyan, magenta, yellow, black}, a subtractive colour space used in printing) range. The effectiveness of FSV as an image comparison method is an interesting possibility. Some of the constants in the method may need to be changed to make the results agree with how people consider images to compare, since the criteria for comparison are likely to be different, but the underlying method, which looks at overall changes in the image, and detail, can be argued at least by the way people describe images verbally; they refer to large areas of colour and details.

##### **7.4.2.2 Beyond 2 Dimensional Domains.**

The use of elliptical regions to segment the data into low and high frequencies naturally extends into higher dimensions with (hyper-)ellipsoids. The algorithms should be relatively straightforward to extend to higher dimensions given that Fourier Transforms are well defined for 3 and more dimensions. The difficulties are going to be the display of the data in an intuitive manner, and understanding how people will interpret the results. For brain scans in medicine, people display the 3D data set as a collection of slices. The morphology of the brain is well known and can be navigated relatively easily by a clinician. When the result of the comparison of two data sets has a form that is not obvious before hand, this is more difficult to explore. The use of 3D displays will be a useful avenue of research, with a number of issues worthy of exploration. How should the alpha transparency of voxels (volume pixels) be handled in order to allow the background data to be seen clearly? What ways may one manipulate (rotate, zoom, change the optical properties of) the display to allow it to be viewed effectively. The psychological aspects of how people perceive the data, and what impact this knowledge has on the design of displays will also be an issue for data in a 3 dimensional domain.

All the above applies to higher dimensions still. The use of time as a fourth dimension, utilising animation, may be helpful, but at 5 dimensions and beyond, useful detail will be much more difficult to convey.

## 8 List of Publications

Archambeault, Bruce ; Duffy, Alistair; Sasse, Hugh; Li,Xin Kai; Scase, Mark; Shafiullah, Mohammed; “Challenges in developing a multidimensional Feature Selective Validation implementation” IEEE Int Symp on EMC, July 2010, Fort Lauderdale, FLA, USA

Telleria, Ricardo Jauregui; Silva, Ferran; Orlandi, Antonio; Sasse, Hugh; Duffy, Alistair, “Factors influencing the successful validation of transient phenomenon modelling” Electromagnetic Compatibility (APEMC), 2010 Asia-Pacific Symposium on Publication Year: 2010 , Page(s): 338 - 341

H.G. Sasse, A.P. Duffy, A. Orlandi "Applying the Feature Selective Validation (FSV) method to quantifying rf measurement comparisons" ARMMS - RF & Microwave Society Conference, Stevenston Oxforshire UK 21-22 April 2008

Orlandi, A. Antonini, G. Polisini, C. Duffy, A. Sasse, H., "Progress in the development of a 2D Feature Selective Validation (FSV) method" EMC 2008. IEEE International Symposium on Electromagnetic Compatibility 18-22 Aug. 2008

A Duffy, H Sasse, A Coates, C-S Lee, K Hodge, “Cable Coupling Measurements in a Reverberation Chamber”, 2007 International Wire and Cable Symposium

A Coates, H Sasse, D Coleby, A Duffy, A Orlandi, “Validation of a three dimensional Transmission Line Matrix (TLM) model implementation of a mode stirred reverberation chamber” IEEE Transactions on EMC, 2007 .

A Duffy, A Orlandi and H Sasse, “Offset Difference Measure (ODM) Enhancement for the Feature Selective Validation (FSV) Method”, IEEE Transactions on EMC,

HG Sasse and AP Duffy, “Implementation of a parallel distributed TLM Solver”, CEM2004 April 2004, Stratford-on-Avon, UK

C Dole, HG Sasse, AP Duffy “Particle Swarm Optimisation of Broadband Coaxial Cabling” , 53rd IWCS, Nov 2004

H. G. Sasse, B. G. Tunstall, D. Coleby, M. M. Al-Asadi, A. P. Duffy, K. Hodge, A. J. Willis, "A Genetic Algorithm Toolkit for Cable Design", International Wire and Cable Symposium - 50 Florida, USA Nov 12-15 2001. pp143-150

H. G. Sasse, A. P. Duffy: "Topics in the Design of a Parallel TLM Solver" 14th International Zurich Symposium and Technical Exhibition on EMC, Zurich, 20-22 February 2001, pp397-402

H. G. Sasse, A. P. Duffy; "Issues in Distributed Modelling Package Design" Paper No 63.5 EUROEM 2000 Conference Edinburgh UK, 29 May to 2 June 2000

H. G. Sasse, M. M. Al-Asadi, A. P. Duffy, K. Hodge, A. J. Willis; "Computer Assisted Optimisation of Cable Design" 49th IWCS Symposium, Atlantic City, NJ, USA 13-16 November 2000 pp 380-384

## 9 References

- [1] William S. Dorn and David D. McCracken, *Numerical Methods with Fortran IV case studies.*: John Wiley & Sons, 1972, ISBN:0-417-21919-3.
- [2] Neil Gershenfield, *The nature of Mathematical Modelling.*: Cambridge University Press, 1999, ISBN:0521-57095-6.
- [3] Christos Christopoulos, *The Transmission Line Modeling Method.*: IEEE Press, 1995, ISBN:0-7803-1017-9.
- [4] James Nicholson Christopher Clapham, *The Concise Oxford Dictionary of Mathematics*, Third Edition ed.: Oxford Universtiy Press, 2005.
- [5] Erhard Zeidler, Ed., *Oxford User's Guide to Mathematics.*: Oxford University Press, 2004, ISBN:0-19-8057-63-1.
- [6] Information Gatekeepers, Inc. (2009) igigroup.com. [Online].  
<http://igigroup.com/st/pages/certified/cert14.html>
- [7] David G. Luenberger, *Linear and Nonlinear Programming.*: Addison Wesley, 1984, ISBN: 0-201-15794-2.
- [8] Thomas L Saaty and Joseph Bram, *Nonlinear Mathematics*. New York: Dover Publications, Inc, 1964, ISBN:0-486-64233-X.
- [9] Donald A. Pierre, *Optimization Theory with Applications*. New York: Dover Publications, Inc, 1986, ISBN:0-486-65205-X.
- [10] A Cauchy, "Méthode Générale pour la Résolution de Système d' Equations Simultanées," in *Comptes Rendus Hebdomadaires des Sèances de l' Académie des Scieences*, Paris, 1847, pp. 25 536-538.
- [11] Kevin Knight Elaine Rich, *Artificial Intelligence.*: McGraw-Hill, 1991, ISBN:0-07-100894-2.
- [12] J.A. Nelder and R. Mead, "A simplex method for function minimization," *Computer Journa*, vol. 7, no. 4, pp. 308-313, 1965.
- [13] Toby Segaran, *Programming Collective Intelligence.*: O'Reilly Media, Inc, 2007, ISBN:978-0-

596-52932-1.

- [14] John Holland, "Genetic algorithms," *Scientific American*, vol. 267, no. 1, pp. 66-72, July 1992.
- [15] Glenn Seemann David M. Bourg, *AI for Game Developers*.: O'Reilly Media, Inc, 2004, ISBN:0-596-00555-5.
- [16] Mark Ward, ""There's an Ant in My Phone"," *New Scientist*, no. 2118, January 1998.
- [17] Wynn C. Stirling, *Satisficing Games and Decision Making (with applications to Engineering and Computer Science)*.: Cambridge University Press, 2003, ISBN: 0521817242.
- [18] Perry F. Wilson and Mark T. Ma, "Techniques for Measuring the Electromagnetic Shielding Effectiveness of Materials : Part 11-Near-Field Source Simulation," *IEEE Transactions on Electromagnetic Compatibility*, vol. 4, no. 3, pp. 251-259, August 1988.
- [19] J.C.P. Kong and I Lau, "Efficiency Evaluation of Full-Wave Electromagnetic Simulation Techniques in 3D WB-PBGA Package Modeling," in *8th International Conference on electronic Packaging Technology ICEPT*, 2007, pp. 1-5.
- [20] Hsueh-Yung Chao, "The multilevel fast multipole algorithm for electromagnetic compatibility analysis," in *IEEE International symposium on Electromagnetic Compatibility*, 1999, pp. 844-847.
- [21] V.P. Gopinath and T.A. Grotjohn, "Three-dimensional electromagnetic PIC model of a compact ECR plasma source," *IEEE Transactions on Plasma Science*, vol. 23, no. 4, pp. 602-608, August 1995.
- [22] Hua Zeng, Changyi Su, Haixin Ke, and Todd Hubing, "Modeling experiences with full-wave time-domain modeling software," in *IEEE International Symposium on Electromagnetic Compatibility*, 2008, pp. 1-4.
- [23] Jeff Cox Eliyahu M. Goldratt, *The Goal: A Process of Ongoing Improvement*, (Revised Edition) ed.: Gower Publishing Ltd, 2004, ISBN10: 0566086654.
- [24] Kalyan S. Perumalla, "Parallel and Distributed Simulation: Traditional Techniques and Recent Advances," in *Proceedings of the 2006 Winter Simulation Conference*, 2006, pp. 84-95.
- [25] BSI, "Electromagnetic Compatibility (EMC) - Part 4-21: Testing and Measurement Techniques - Reverberation Chamber Test Methods," British Standards Institution, London, Standard BS EN

61000-4-21:2003, 2003.

- [26] C. Eswarappa, P.P.M. So, and W.J.R. Hoefer, "Efficient Field-Based CAD of Microwave Circuits on Massively Parallel Processor using TLM and Prony's Method," in *IEEE MTT-S International Microwave Symposium Digest, 1994*, 1994, pp. 1531-1534.
- [27] P.J. Parsons, S.R.: Pulko, S. H. Jaques, and F.A. Rabbi, "TLM modeling using distributed computing," *Microwave and Guided Wave Letters*, vol. 6, no. 3, pp. 141-142, March 1996.
- [28] V. Varadarajan and R. Mittra, "Finite-difference time-domain (FDTD) analysis using distributed computing," *IEEE Microwave and Guided Wave Letters*, vol. 4, no. 5, pp. 144-145, May 1994.
- [29] Xue Zhenghui, Gao Benqing, and Zhang Zejie, "A strategy for parallel implementation of the FDTD algorithm," in *3rd International Symposium on Electromagnetic Compatibility*, 2002, pp. 259-263.
- [30] S.T. Nguyen, B.J. Zook, and Xiaodong Zhang, "Distributed computation of electromagnetic scattering problems using finite-difference time-domain decompositions," in *Proceedings of the Third IEEE International Symposium on High Performance Distributed Computing*, 1994, pp. 85-93.
- [31] Feng Ju and Cheng Xing, "A study of parallel FDTD for simulating complex antennas on a cluster system," in *Microwave Conference Proceedings, 2005, APMC 2005 Asia-Pacific Conference*, Chengdu, 2005, p. 4.
- [32] Xu Zhang, Xiaoming Liu, Shaobua Liu, Junsheng Yu, Ran Ding, Bingran Liu, and Rui Xiao, "A more efficient mechanism of FDTD method based on message passing interface," in *ICMMT 2008 International Conference on Microwave and Millimeter Wave Technology*, 2008, pp. 692-695.
- [33] Dan Yang, Cheng Liao, Lang Jen, and Jie Xiong, "parallel FDTD algorithm based on domain decomposition method using the MPI library," in *PDCAT2003 Proceedings of the Fourth International Conference on Parallel and Distributed computing, Applications and Technologies*, 2003, pp. 730-733.
- [34] G.A. Schiavone, L. Codreanu, R. Palaniappan, and P. Wahid, "FDTD speedups obtained in distributed computing on a Linux workstation cluster," in *IEEE Antennas and Propagation*



*Society International Symposium*, 2000, pp. 1336-1339.

- [35] C. Cuiffaut and K. Mahdjoubi, "A parallel FDTD algorithm using the MPI library," *Antennas and Propagation Magazine*, vol. 43, no. 2, pp. 94-104, April 2001.
- [36] Meilian Xu and Parimala Thulasiraman, "Parallel Algorithm Design and Performance Evaluation of FDTD on 3 Different Architectures: Cluster, Homogeneous Multicore and Cell/B.E.," in *HPCC'08 10th IEEE International Conference on High Performance Computing and Communications*, 2008, pp. 174-181.
- [37] A.J. Woods, D.L. Rhoades, L.W. Cavey, G.R. Hall, and L.D. Ludcking, "MAGIC electromagnetic FDTD PIC code linux cluster performance," in *IVEC 2008 IEEE International Vacuum Electronics Conference*, 2008, pp. 221-222.
- [38] Xiaohe Chen, M. Cracraft, Yauojinang Zhang, Jiannin Zhang, J.L. Drewniak, B. Archambeault, and S. Connor, "An Efficient Implementation of Parallel FDTD," in *IMC 2007 IEEE International Symposium on Electromagnetic Compatibility*, 2007, pp. 1-5.
- [39] J.S. Araujo, R.O. Santos, C.L.S.S. Sobrinho, J.M. Rocha, L.A. Guedes, and R.Y. Kawasaki, "Analysis of antennas by FDTD method using parallel processing with MPI," in *IMOC2003 Proceedings of the 2003 Microwave and Optoelectronics Conference*, 2003, pp. 1049-1054.
- [40] A. Ray, G. Kondayya, and S.V.G Menon, "Developing a finite difference time domain parallel code for nuclear electromagnetic field Simulation," *IEEE Transactions on Antennas and Propagation*, vol. 54, no. 4, pp. 1192-1199, April 2006.
- [41] O. Ramadan, O. Akaydin, M. Salamah, and A.Y. Oztoprak, "Parallel implementation of the finite-difference time-domain method on local area network using the message passing interface," in *2004 International Conference on Information and Communication Technologies: From Theory to Applications*, 2004, pp. 651-652.
- [42] D.A. Gorodetsky and P.A. Wilsey, "Innovative approaches to parallelizing finite-difference time-domain computations," in *Proceedings of Xth International Seminar/Workshop on Direct and Inverse Problems of Electromagnetic and Acoustic Wave Theory.*, 2005, pp. 43-47.
- [43] J. Costa, K. Roy, F. Costen, and A. Brown, "Toward large scale Frequency Dependent FDTD for UWB systems," in *IEEE Antennas and Propagation International Symposium, 2007*, 2007, pp. 3724-2727.

- [44] M.K. Perdikeas, D.I. Kaklamani, and I.S. Venieris, "Distributed computation models for aggregating a dynamic set of machines to solve complex electromagnetic problems," in *Proceedings of the Second International symposium of Trans Black Sea Region on Applied Electromagnetism, 2000*, 2000, p. 75.
- [45] N. Phan-Thien, Ka Yan lee, and D. Tullock, "Large Scale Simulation of Suspensions with PVM," in *ACM/IEEE Conference on Supercomputing*, 1997, p. 60.
- [46] Phil Tomson, "Distributed computing with Ruby," *Dr Dobb's Journal*, vol. 27, no. 9, pp. 16,18,20,22, September 2002.
- [47] N. Peila and M.M. Ney, "A new TLM node for Berenger's perfectly matched layer," *IEEE Microwave and Guided Wave Letters*, vol. 6, no. 11, pp. 410-412, November 1996.
- [48] C. Eswarappa and W.J.R. Hoefer, "Implementation of Berenger absorbing boundary conditions in TLM by interfacing FDTD perfectly matched layers," *Electronics Letters*, vol. 31, no. 15, pp. 1264-1266, July 1995.
- [49] C. Eswarappa and W.J.R. Hoefer, "Implementation of Berenger absorbing boundary conditions in TLM by interfacing FDTD perfectly matched layers," *Electronics Letters*, vol. 31, no. 15, pp. 1264-1266, July 1995.
- [50] A.P. Duffy, T.M. Benson, C. Christopoulos, and J.L. Herring, "New methods for accurate modelling of wires using TLM," *Electronics Letters*, vol. 29, no. 2, pp. 224-226, January 1993.
- [51] A.P. Duffy, J.L. Herring, T.M. Benson, and C. Christopoulos, "A new wire representation for TLM," in *IEEE International Symposium on Electromagnetic Compatibility, 1995, Symposium Record*, 1993, pp. 434-435.
- [52] A.P. Duffy, T.M. Benson, and C Christopoulos, "A computationally efficient method for studying the behaviour of complex systems containing thin wires," in *Ninth International Conference on Electromagnetic Compatibility, 199 (Conf Publ. No 396)*, 1994, pp. 274-281.
- [53] A.P. Duffy, J.L., Benson, T.M. Herring, and C Christopoulos, "Improved wire modelling in TLM," *IEEE Transactions on Microwave Theory and Techniques*, vol. 42, no. 10, pp. 1978-1983, October 1994.
- [54] P. Russer, B. Isele, M. Sobby, and E.A. Hosny, "A general interface between TLM models and lumped element circuit models," in *IEEE MTT-S International Microwave Symposium Digest*

1994, 1994, pp. 891-894.

- [55] J.W. Park, P.P.M. So, and W.J.R. Hoefer, "Lumped and distributed device embedding techniques in time domain TLM field models," in *IEEE MTT-S International Microwave Symposium Digest 2001*, 2001, pp. 2047-2050.
- [56] J.L. Herring and C. Christopoulos, "The use of graded and multigrid techniques in transmission-line modelling," in *Second International Conference on Computation in Electromagnetics*, 1994, 1994, pp. 142-145.
- [57] V. Trenkic, C. Christopoulos, and T.M. Benson, "Advanced node formulations in TLM-the adaptable symmetrical condensed node," *IEEE Transactions on Microwave Theory and Techniques*, vol. 44, no. 12, pp. 2473-2478, December 1996.
- [58] V. Trenkic, C. Christopoulos, and T.M. Benson, "Theory of the symmetrical super-condensed node for the TLM method," *IEEE Transactions on Microwave Theory and Techniques*, vol. 43, no. 6, pp. 1342-1348, June 1995.
- [59] V. Trenkic, C. Christopoulos, and T.M. Benson, "Generally graded TLM mesh using the symmetrical supercondensed node," *Electronics Letters*, vol. 30, no. 10, pp. 795-797, May 1994.
- [60] V. Trenkic, C. Christopoulos, and T.M. Benson, "A graded symmetrical super-condensed node for the TLM method," in *Antennas and Propagation society International Symposium*, 1994, pp. 1106-1109.
- [61] M. Schumer and K. Steiglitz, "Adaptive step size random search," *IEEE Transactions on Automatic Control*, vol. 13, no. 3, pp. 270-276, June 1968.
- [62] T. Menzies, D. Owen, and B. Cukic, "Saturation effects in testing of formal models," in *ISSRE Proceedings 13th International Symposium on Software Reliability Engineering*, 2002, pp. 15-26.
- [63] G. Cherubini, S. Olcer, and G. Ungerboeck, "Adaptive analog equalization and receiver front-end control for multilevel partial-response transmission over metallic cables," *IEEE Transactions on Communications*, vol. 44, no. 6, pp. 675-685, June 1996.
- [64] M. Alaydrus and T.F. Eibert, "Optimizing Multiband Antennas Using Simulated Annealing," in *INICA '07, 2nd International ITG Conference on Antennas*, 2007, pp. 86-90.

- [65] Gao Huotao, Li Jie, Zheng Xia, and Wei Zhongwei, "Optimal design of broadband loaded receiving antenna for HFSWR using simulated annealing," in *4th International Symposium on Antennas, Propagation and EM Theory*, 2003, pp. 30-33.
- [66] Fred Glover, "Tabu Search—Part I," *ORSA Journal on Computing*, vol. 1, no. 3, pp. 190-206, Summer 1989.
- [67] F. Glover, "Tabu Search -- Part II.," *ORSA Journal on Computing*, vol. 2, no. 1, pp. 4-32, 1990.
- [68] V. Karamcheti and M. Malek, "A TSP engine for performing tabu search," in *Proceedings of the International Conferenc on Application Specific Array Processors*, 1991, pp. 309-321.
- [69] C. Fleurent, F. Glover, P. Michelon, and Z. Valli, "A scatter search approach for unconstrained continuous optimization," in *Proceedings of IEEE International Conference on Evolutionary Computation*, 1996, pp. 643-648.
- [70] J.M. Johnson and V. Rahmat-Samii, "Genetic algorithms in engineering electromagnetics," *IEEE Antennas and Propagation Magazine*, vol. 39, no. 4, pp. 7-21, August 1997.
- [71] Donald Michie, "Trial and Error," in *On Machine Learning*.: Edinburgh University Press, 1974.
- [72] Suomin Cui and D.S. Weile, "Robust design of absorbers using genetic algorithms and the finite element-boundary integral method," in *IEEE Antennas and Propagation Society International Symposium*, 2002, pp. 326-329.
- [73] S. Cui and D.S. Weile, "Robust design of absorbers using genetic algorithms and the finite element-boundary integral method," *IEEE Transactions on Antennas and Propagation*, vol. 51, no. 12, pp. 3249-3258, December 2003.
- [74] J. Clegg, A.C. Marvin, J.F. Dawson, and S.J. Porter, "Optimization of stirrer designs in a reverberation chamber," *IEEE Transactions on Electromagnetic Compatibility*, vol. 47, no. 4, pp. 824-832, November 2005.
- [75] A. Varahram, J. Rashed-Mohassel, and K. Mafinezhad, "Design of matching transformers for UHF band power splitters using modified genetic algorithms," in *IEEE Antennas and Propagation Society International Symposium*, 2003, pp. 96-99.
- [76] Y.H. Lee, A.C. Marvin, and S.J. Porter, "Genetic algorithm using real parameters for array antenna design optimisation," in *High Frequency Postgraduate Student Colloquium*, 1999, pp.

- [77] D. Patrick, P. Green, and T York, "A distributed genetic algorithm environment for UNIX workstation clusters," in *GALENA 97 Second International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*, 1997, pp. 69-74.
- [78] R. Taylor, "Signal processing with occam and the transputer," in *Ccommunication, Radar and Signal Processing, IEE Proceedings. Part F*, 1984, pp. 610-614.
- [79] R.F. Brender and I.R. Nassi, "What is Ada?," *Computer*, vol. 14, no. 6, pp. 17-24, June 1981.
- [80] Zhan Min, Luo Xiao-Wu, and Wang Guang-Hui, "Preliminary research of the reconfigurable antenna based on genetic algorithms," in *ICCEA 2004 Computational Electromagnetics and Its Applications*, 2004, pp. 137-140.
- [81] Riccardo Poli, "Analysis of the publications on the applications of particle swarm optimisation," *Journal of Artificial Evolution and Applications*, vol. 8, no. 2, pp. 1-10, January 2008.
- [82] Russel C. Eberhart and Yuhui Shi, "Particle swarm optimization: developments, applications and resources," in *Proceedings of the 2001 Congress on Evolutionary Computation*, 2001, pp. 81-86.
- [83] D.W. Boeringer and D.H. Werner, "A comparison of particle swarm optimization and genetic algorithms for a phased array synthesis problem," in *IEEE Antennas and Propagation Society International Symposium*, 2003, pp. 181-184.
- [84] M.P. Wachowiak, R. Smolikova, Yufeng Zheng, J.M. Zurada, and A.S Elmaghraby, "An approach to multimodal biomedical image registration utilizing particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 289-301, June 2004.
- [85] Thomas M. Cover and Joy A. Thomas, *Elements of Information Theory*.: John Wiley and Sons, Inc, 1991, ISBN:0-471-06259-6.
- [86] S. Chamaani, S.A. Mirtaheri, M. Teshnehlal, and M.A. Shooredeli, "Modified Multi-objective Particle Swarm Optimization for electromagnetic absorber design," in *APACE 2007 Asia-Pacific Conference on Applied Electromagnetics*, 2007, pp. 1-5.
- [87] Juan Chen and Lihong Xu, "Road-Junction Traffic Signal Timing Optimization by an adaptive Particle Swarm Algorithm," in *ICARVC '06, 9th International Conference on Control*,

*Automation, Robotics and Vision*, 2006, pp. 1-7.

- [88] Y Shi and R. Eberhart, "A modified particle swarm optimizer," in *IEEE World Congress on Computational Intelligence, The 1998 IEEE International Conference on Evolutionary computation*, 1998, pp. 69-73.
- [89] K. Chandramouli and E. Izquierdo, "Image Classification using Chaotic Particle Swarm Optimization," in *IEEE International Conference on Image Processing*, 2006, pp. 3001-3004.
- [90] A. Ratnaweera, S.K. Halgamuge, and H.C. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 240-255, June 2004.
- [91] J. Togelius, R. De Nardi, and A. Moraglio, "Geometric PSO + GP = Particle Swarm Programming," in *CEC 2008 (IEEE World Congress on Computational Intelligence) IEEE Congress on Evolutionary Computation*, 2008, pp. 3594-3600.
- [92] Lei Xu and Fengming Zhang, "Parallel Particle Swarm Optimization for Attribute Reduction," in *SNPD 2007 Eighth ACIA International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel Distributed Computing*, 2007, pp. 770-775.
- [93] Xi-Huai Wang and Jun-Jun Li, "Hybrid particle swarm optimization with simulated annealing," in *Proceedings of 2004 International Conference on Machine Learning and Cybernetics*, 2004, pp. 2402-2405.
- [94] I.M. Supratid, "A Multi-Subpopulation Particle Swarm Optimization: A Hybrid Intelligent Computing for Function Optimization," in *ICNC 2007 Third International Conference on Natural Computation*, 2007, pp. 679-684.
- [95] Chaohua Dai, Weirong Chen, and Yunfang Zhu, "Seeker Optimization Algorithm," in *International Conference on Computational Intelligence and Security*, 2006, pp. 225-229.
- [96] Xiaolan Wu, Bo Cheng, Jianbo Cao, and Binggang Cao, "Particle swarm optimization with normal cloud mutation," in *WCICA 2008 7th World Congress on Intelligent Control and Automation*, 2008, pp. 2828-2832.
- [97] Jiangping Wen and Binggang Cao, "A modified particle swarm optimizer based on cloud model," in *AIM 2008 IEEE/ASME International Conference on Advanced Intelligent Mechanics*, 2008, pp. 1238-1241.

- [98] Yang Guangyou, "A Modified Particle Swarm Optimizer Algorithm," in *ICEMI '07 8th International Conference on Electronic Measurement and Instruments*, 2007, pp. 2-675 - 2-679.
- [99] B. Archambeault, "Concerns and approaches for accurate EMC simulation validation," in *IEEE International Symposium on Electromagnetic Compatibility*, 2003, pp. 329-334.
- [100] Bruce. Archambeault and Samuel. Connor, "Proper model validation is important for all EMI/EMC applications," in *EMC 2008 IEEE International Symposium on Electromagnetic Compatibility*, 2008, pp. 1-8.
- [101] Tim Wilson, *EMC for Product Engineers.*: Newnes (an imprint of Elsevier), 2007, ISBN:0-7506-8160-5.
- [102] Andy Hunt, *Pragmatic Thinking and Learning: Refactor Your Wetware.*: Pragmatic Bookshelf, 2008, ISBN:193435050.
- [103] D.R. Pflug, "A novel test article for validation of EMC codes for antenna-to-antenna coupling and antenna isolation: the Transformable Scale Aircraft-Like Model," in *IEEE International Symposium on Electromagnetic Compatibility*, 1998, pp. 1161-1165.
- [104] S.P. Watkins, R. Hoad, L.C. Chirwa, J.F. Dawson, and M.P. Robinson, "Validation of intermediate level circuit models (ILCM) of a generic system through measurement," in *IEE Seminar on Validation of Computational Electromagnetics*, 2004, pp. 65-69.
- [105] L. Foged, F. Mioc, B. Bencivenga, E. Di Giampaolo, and M. Sabbadini, "High frequency numerical modelling using measured sources," in *IEEE Antennas and Propagation society International symposium*, 2006, pp. 1837-1840.
- [106] D.G. Beetner, Haixiao Weng, Meilin Wu, and T. Hubing, "Validation of Worst-Case and Statistical Models for an Automotive EMC Expert System," in *IMC 2007 IEEE International Symposium on Electromagnetic Compatibility*, 2007, pp. 1-5.
- [107] V. Rajamani, C.F. Bunting, A. Orlandi, and A. Duffy, "Introduction to feature selective validation (FSV)," in *IEEE Antennas and Propagation Society International Symposium*, 2006, pp. 601-604.
- [108] A. Ruddle, D. Ward, A. Williams, and A. Duffy, "Objective validation of automotive EMC models," in *IEEE International Symposium on Electromagnetic Compatibility*, 1998, pp. 475-479.

- [109] A. Duffy, D. Coleby, A. Martin, M. Woolfson, and T. Benson, "Progress in quantifying validation data," in *IEEE International Symposium on Electromagnetic Compatibility, 2003*, 2003, pp. 323-328.
- [110] W.J. Dallas, "The Correlelogram: a two transmission layer display device," *Applied Optics*, vol. 24, pp. 3945-3947, 1985.
- [111] A.J.M. Williams, M.S. Woolfson, T.M. Benson, and A.P. Duffy, "Quantitative validation of numerical technique results against experimental data," in *Tenth International Conference on Antennas and Propagation (Conf. Publ. No. 436)*, 1997, pp. 532-535.
- [112] R.J. Simpson, C.R. Jones, I. MacDiarmid, A. Duffy, and D. Coleby, "The integrated error against log frequency (IELF) method for CEM validation," in *EMC 2005 International Symposium on Electromagnetic Compatibility*, 2005, pp. 296-300.
- [113] D.E. Coleby and A.P. Duffy, "Development of a visual rating scale to aid validations," in *IEE Seminar on Validation of Computational Electromagnetics*, 2004, pp. 75-79.
- [114] A. Duffy, A. Martin, G. Antonini, A. Orlandi, and C. Ritota, "The feature selective validation (FSV) method," in *EMC 2005 International Symposium on Electromagnetic Compatibility*, 2005, pp. 272-277.
- [115] A.P. Duffy, A.J.M. Williams, T.M. Benson, and M.S. Woolfson, "Quantitative assessment of experimental repeatability," *IEE Proceedings Science Measurement and Technology*, vol. 145, no. 4, pp. 177-180, July 1998.
- [116] A.P. Duffy, A.J.M. Martin, A. Orlandi, G. Antonini, T.M. Benson, and M.S. Woolfson, "Feature selective validation (FSV) for validation of computational electromagnetics (CEM). part I-the FSV method," *IEEE Transactions on Electromagnetic Compatibility*, vol. 48, no. 3, pp. 449-459, August 2006.
- [117] A. Duffy, A. Orlandi, and H. Sasse, "Offset Difference Measure Enhancement for the Feature-Selective Validation Method," *IEEE Transactions on Electromagnetic Compatibility*, vol. 50, no. 2, pp. 413-415, May 2008.
- [118] A. Orlandi, G. Antonini, and C. Duffy, A.P. Ritota, "Enhancing feature selective validation (FSV) interpretation of EMC/SI results with grade-spread," in *EMC 2006 IEEE International Symposium on Electromagnetic Compatibility*, 2006, pp. 362-267.



- [119] A. Orlandi, A.P. Duffy, B. Archambeault, G. Antonini, D.E. Cooley, and S. Connor, "Feature selective validation (FSV) for validation of computational electromagnetics (CEM). part II- assessment of FSV performance," *IEEE Transactions on Electromagnetic Compatibility*, vol. 44, no. 3, pp. 460-467, August 2006.
- [120] B. Archambeault, S. Connor, and A. Duffy, "Comparing FSV and human responses to data comparisons," in *EMC 2005 International Symposium on Electromagnetic Compatibility*, 2005, pp. 284-289.
- [121] Bart Kosko, *Fuzzy Thinking.:* Flamingo (Harper Collins), 1994, ISBN:0-00-654713-3.
- [122] A.J.M. Martin, "Quantitative Data Comparisons - Automated Visual Evaluations," De Montfort University, Leicester, UK, PhD Thesis 1999.
- [123] S Ramo, J. R. Whinny, and T Van Duzer, *Fields and Waves in Communications Electronics*, 3rd ed. New York, USA: J. Wiley and Sons, 1994.
- [124] A. Buja, D. Asimov, C. Hurley, and J.A. McDonald, "Elements of a Viewing Pipeline for Data Analysis," in *Dynamic Graphics for Statistics*, , W.S. Cleveland and M.E. McGill, Eds., 1988, pp. 277-308.
- [125] Hugh G. Sasse, Benjamin G. Tunstall, Dawn E. Coleby, Mohammed M. Al-Asadi, Alistair P. Duffy, Kenneth Hodge, and Arthur J. Willis, "A Genetic Algorithm Toolkit for Cable Design," in *International Wire and Cable Symposium*, Lake Buena Vista, Florida, 2001, pp. 143-150.
- [126] C. Dole, H.G. Sasse, and A.P. Duffy, "Particle Swarm Optimisation of Broadband Coaxial Cabling," in *53rd International Wire and Cable Symposium*, 2004, pp. 41-46.
- [127] J.N. Katakis, "Leakage through and optimization of braids for coaxial cables," University of Sheffield, Sheffield, MEng Thesis 1981.
- [128] M Tyni, "The Transfer Impedance of coaxial Cables With Braided Outer Conductor," in *Wroclaw IMC Symposium, Poland*, Wroclaw, 1976, pp. 410-418.
- [129] Dave Thomas, Chad Fowler, and Andy Hunt, *Programming Ruby: The Pragmatic Programmer's Guide*, 2nd ed.: Pragmatic Bookshelf, 2004, ISBN-10: 0974514055 ISBN-13: 978-0974514055.
- [130] Bertrand Meyer, *Object Oriented Software Construction.:* Prentice Hall, 1997, ISBN:0-13-

629155-4.

- [131] Frederick P. Brooks Jr, *The Mythical Man Month and Other Essays on Software Engineering*, 2nd ed.: Addison Wesley, 1995, ISBN-10:0201835959 ISBN-13: 978-0201835953.
- [132] Andrew S. Tanenbaum, *Distributed Operating Systems*.: Prentice Hall, 1995, ISBN:0-13-143934-0.
- [133] The European Union. (2008, December) The Wassenaar Arrangement on Export Controls for Conventional Arms and Dual Use Goods and Technologies. [Online].  
[http://www.wassenaar.org/controllists/2008/WA-LIST%20\(08\)%201/WA-LIST%20\(08\)%201.pdf](http://www.wassenaar.org/controllists/2008/WA-LIST%20(08)%201/WA-LIST%20(08)%201.pdf)
- [134] Export Control Organisation. (2006, January) Guidance on Export of Cryptographic Items. [Online]. <http://www.berr.gov.uk/whatwedo/europeandtrade/strategic-export-control/licensing-policy/guidance/page8489.html>
- [135] H. Krawczyk, M. Bellare, and R. Canetti, HMAC: Keyed-Hashing for Message Authentication, 1997, RFC 2104.
- [136] J Klensin, R. Catoe, and P Krumviede, "IMAP/POP AUTHorize Extension for Simple Challenge/Response", 1997, RFC 2195.
- [137] Erich Gamma, Richard Helm, Richard Johnson, and John Vlissides, *Design patterns: elements of reusable object-oriented software*.: Addison Wesley, 1995, ISBN-10: 0201633612, ISBN-13: 978-0201633610.
- [138] Hugh G. Sasse, Mohammed M. Al-Asadi, Alistair P. Duffy, Kenneth G. Hodge, and Arthur J. Willis, "Computer assisted Optimisation of Cable Design," in *49th International Wire and Cable Symposium*, Atlantic City, NJ, USA, 2000, pp. 380-384.
- [139] H.G. Sasse and A.P. Duffy, "Topics in the Design of a Parallel TLM Solver," in *14th International Zurich Symposium and Technical Exhibition on EMC (EMC Zurich)*, Zurich, 2001, pp. 397-402.
- [140] L. Peter Deutsch, GZIP file format specification version 4.3, 1996, RFC 1952.
- [141] N. Freed and N. Borenstein, Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies, 1996, RFC 2045.

- [142] IEEE, IEEE 1597.1 Standard for Validation of Computational Electromagnetics Computer Modeling and Simulation., 2009.
- [143] A. Duffy, A. Orlandi, and G. Antonini, "Testing the robustness of the feature selective validation (FSV) method for EMC/SI applications," in *EMC Europe*, Barcelona, 2008.
- [144] E.R. Davies, *Machine Vision: Theory, Algorithms, Practicalities.*: Morgan Kaufmann, 2005, ISBN: 0-12-206093-8.
- [145] John G. Proakis and Dimitris G. Manolakis, *Digital Signal Processing: Principles, Algorithms and Applications*, 4th ed.: Pearson Education, 2007, ISBN:0-13-187374-1.
- [146] John R. Tyldesley, *An Introduction to Tensor Analysis for Engineers and Scientists.*: Longman Group, Ltd, 1975, ISBN:582-44354-7.
- [147] D.E. Bourne and P.C. Kendall, *Vector Analysis and Cartesian Tensors.*: Van Nostrand Reinhold (UK) Co. Ltd., 1977, ISBN:0-442-30743-8.
- [148] (2009) Meshlab. [Online]. <http://meshlab.sourceforge.net>
- [149] Fraunhofer-Gesellschaft. 3D-FoForm Project. [Online]. <http://www.3d-coform.eu/>
- [150] R. Steen Glanville. (2009) Anim8or. [Online]. <http://www.anim8or.com/main/index.html>
- [151] Michael Howard and David LeBlanc, *Writing Secure Code.*: Microsoft Press, 2002, ISBN: 0-7356-1588-8.
- [152] Uyless Black, *Internet Security Protocols: Protecting IP Traffic.*: Prentice Hall, 2000, ISBN: 0-13-014249-2.
- [153] Barbara Burke Hubbard, *The World According To Wavelets.*: AK Peters Ltd, 1998, ISBN: 1-56881-072-5.
- [154] John M. Hull, *Touching the Rock: an experience of blindness.*: SPCK, 1990, ISBN: 0-281-04444-9.
- [155] D.P. Anderson, "BOINC: A System for Public-Resource Computing and Storage," in *5th IEEE/ACM International Workshop on Grid Computing*, Pittsburgh, PA, 2004, pp. 365-372.

# Appendix A – Implementation of FSV method (quoted from IEEE1597.1 “Standard for Validation of Computational Electromagnetics Simulation and Modeling” 2009)

## 7.4 Procedure

Calculation of the ADM and FDM requires access to a Fourier Transform capability. The procedure is presented hereafter, and is as found in *Feature Selective Validation (FSV) for validation of computational electromagnetics (CEM). Part I – The FSV method* [B24]<sup>1</sup>.

The procedure assumes that two separate data sets are available for comparison. Both data sets share a common independent variable (plotted as an x-axis) and a common dependent variable (plotted as a y-axis). The two data sets may:

- contain different numbers of data points;
- exist over different ranges of the common independent variable;
- have either coincident or non-coincident values of the common independent variable.

The procedure refers to these data sets as the input data sets. These are manipulated during the procedure to generate the working data sets, i.e. the data sets used in the calculation of the ADM and FDM.

### 1. Generation of the Working Data Sets

This step sees the input data sets manipulated to create the working data sets. The working data sets contain the same number of data points and have coincident values of the common independent variable.

- a. Compare the input data sets to establish the region of overlap within the common independent variable;
- b. Identify those data points within each data set that exist within the region of overlap;
- c. Identify the number of data points within each data set that exist within the region of overlap;
- d. For the input data set with the smallest number of data points within the region of overlap, extract the data points within the region of overlap to form the first working data set;
- e. Copy the values of the independent variable from the first working data set to the second working data set;
- f. Use the remaining input data set (that with the highest number of data points within the region of overlap), apply linear interpolation within the region of overlap to generate the value of the independent variable at each coincident value of the second working data set;

---

<sup>1</sup> Reference [B24] is: Duffy, A. P., Martin, A. J. M., Orlandi, A., Antonini, G., Benson, T. M., and Woolfson, M. S., *Feature Selective Validation (FSV) for validation of computational electromagnetics (CEM). Part I – The FSV method*, IEEE Trans-EMC, Vol. 48 No. 3, Aug. 2006, pp. 449 – 459.

Note that when both input data sets have the same number of data points within the region of overlap, either data set may be used in step d, above.

2. Fourier Transform both working data sets. This creates the transformed working data sets.
3. The following procedure is to be performed for both transformed working data sets:
  - a. Perform an Inverse Fourier Transform on the content of the first four data points within the transformed data set. The vectors returned shall be labeled as  $DC_s(n)$ , where the subscript,  $s$ , is either 1 or 2 and denotes the transformed working data set number.
  - b. Calculate the sum of the values,  $S$ , of the independent variable, starting at the fifth data point and continuing to the end of the data set. Thus:

$$S = \sum_{i=5}^N TWDS(i)$$

where

- |           |   |
|-----------|---|
| $S$       | is the sum of the values of the independent variable;               |
| $i$       | is the data set element;  |
| $N$       | is the total number of elements within the data set;                |
| $TWDS(i)$ | is the $i$ th independent variable within the transformed data set. |

- c. Determine the ‘40%’ location. This is the  $i$ th data point, at which the sum of the values of the independent variable, starting at the fifth data point and continuing to the  $i$ th data point, is greater than or equal to 40% of the value calculated during step (b). The value of  $i$  is systematically increased from the fifth data point until this condition is reached. Thus:

$$\sum_{i=5}^{i_{40\%}} TWDS(i) \geq 0.4 \cdot S$$

where

- |            |  |
|------------|--|
| $S$        | is the sum of the values of the independent variable;                            |
| $i$        | is the data set element;   |
| $TWDS(i)$  | is the value of the $i$ th independent variable within the transformed data set; |
| $i_{40\%}$ | is the element containing the ‘40%’ location.                                    |

- d. Determine the ‘break-point’ location. This is five data points higher than the ‘40% location’ determined in step (c), above. Thus:

$$i_{bp} = i_{40\%} + 5$$

where

- |            |   |
|------------|---|
| $i_{40\%}$ | is the element number of the 40% location;  |
| $i_{bp}$   | is the element number of the ‘break point’. |

- e. Apply the filter displayed as Table 1 to the independent variable values contained within the indicated elements of the transformed working data set to generate the filtered transformed working data set.

**Table 1 – Filter Definition – a**

Element Number	Filter Value
$I_{bp}-3$	1.000
$I_{bp}-2$	0.834
$I_{bp}-1$	0.667
$I_{bp}$	0.500
$I_{bp}+1$	0.334
$I_{bp}+2$	0.167
$I_{bp}+3$	0.000

- f. Perform an Inverse Fourier Transform on the filtered transformed working data set. The vectors returned shall be labeled as  $Lo_s(n)$ , where s is either 1 or 2 and denotes the data set number.
4. The following procedure is to be performed for both transformed working data sets:
  - a. Apply the filter displayed as Table 2 to the independent variable values contained within the indicated elements of the transformed working data set to generate the filtered transformed working data set.

**Table 2 – Filter Definition – b**

Element Number	Filter Value
$I_{bp}-3$	0.000
$I_{bp}-2$	0.167
$I_{bp}-1$	0.334
$I_{bp}$	0.500
$I_{bp}+1$	0.667
$I_{bp}+2$	0.834
$I_{bp}+3$	1.000

- b. Perform an Inverse Fourier Transform on the filtered transformed working data set. The vectors returned shall be labeled as  $Hi_s(n)$ , where s is either 1 or 2 and denotes the data set number.
5. For each set of data points within the Working Data Sets, the ADM is to be calculated using Equation (1).

$$ADM(n) = \left| \frac{\alpha}{\beta} \right| + \left| \frac{\gamma}{\delta} \right| \exp \left\{ \left| \frac{\gamma}{\delta} \right| \right\} \quad (1)$$

where

$$\alpha = \left( \left| Lo_1(n) \right| - \left| Lo_2(n) \right| \right)$$

$$\beta = \frac{1}{N} \sum_{i=1}^N \left( \left( |Lo_1(i)| + |Lo_2(i)| \right) \right)$$

$$\chi = \left( |DC_1(n)| - |DC_2(n)| \right)$$

$$\delta = \frac{1}{N} \sum_{i=1}^N \left( \left( |DC_1(i)| + |DC_2(i)| \right) \right)$$

$n$  is the  $n$ th data point

6. Calculate the mean value of ADM from Equation (2).

$$ADM = \frac{\sum_{n=1}^N ADM(n)}{N} \quad (2)$$

where

The mean value of the  $ADM(n)$  gives a single-figure measure of ‘goodness-of-fit’

7. Calculate the ADM confidence histogram. The range of values for the ADM and, in fact, the FDM and GDM can be divided into six categories, each with a natural language descriptor: Excellent, Very Good, Good, Fair, Poor, and Very Poor. These are the terms that are most often used in descriptions of the quality of comparisons. The confidence histogram, like a probability density function, provides some intelligence as to how much emphasis can be placed on the single figure of merit. There is some evidence to show that this mirrors the overall group assessment of any data pair by a number of engineers. The determination of the histogram is simply a case of counting the proportion of points that fall into one of the categories, according to the rule base in Table 3.

**Table 3 – FSV interpretation scale**

FSV value (quantitative)	FSV interpretation (qualitative)
Less than 0.1	Excellent
Between 0.1 and 0.2	Very good
Between 0.2 and 0.4	Good
Between 0.4 and 0.8	Fair
Between 0.8 and 1.6	Poor
Greater than 1.6	Very poor

8. Calculate derivatives in preparation for the FDM calculation. The following components need to be calculated:
- The first derivatives of the  $Lo(f)$  and  $Hi(f)$  data sets
  - The second derivatives of the  $Hi(f)$  data sets.

The derivatives accentuate the high rate-of-change features in the original data, and differences based on the derivatives are combined in the determination of the FDM. The first derivatives are obtained using a central difference scheme as in Equation (3).

$$Lo'(f) = Lo(f + N_d) - Lo(f - N_d) \quad (3)$$

where

$N_d = 2$  for the first derivative

The second derivatives of the Hi data sets are obtained from the first derivatives using a similar approach again, found in Equation (4)

$$Hi''(f) = Hi'(f + N_d) - Hi'(f - N_d) \quad (4)$$

where

$N_d = 3$

Equation (5) shows the derivative operators (from Equation (3) and Equation (4)) in terms of templates that 'slide across' the original data to generate the first and second derivatives.

$$\begin{aligned} Hi' &= [1 \ 0 \ 0 \ 0 \ -1]^T \\ Hi'' &= [1 \ 0 \ 0 \ 0 \ -1 \ 0 \ -1 \ 0 \ 0 \ 1]^T \end{aligned} \quad (5)$$

where

superscript  $T$  stands for transpose

9. Calculate the point-by-point FDM. The FDM is formed from three parts based on the derivatives calculated in step 8. The numerical values in the equations are part of the heuristic and have been determined empirically. Note that because the following equations describe point-by-point values in the numerator, the denominator is based on a mean-value approach, similar to the determination of the ADM. It should be noted that the combination of Equations (6), Equation (7), and Equation (8) results in Equation (9) resolving to the difference between a function of Dataset 1 and a function of Dataset 2.

$$FDM_1(f) = \frac{|Lo_1'(f)| - |Lo_2'(f)|}{\frac{2}{N} \sum_{i=1}^N (|Lo_1'(i)| + |Lo_2'(i)|)} \quad (6)$$

$$FDM_2(f) = \frac{|Hi_1'(f)| - |Hi_2'(f)|}{\frac{6}{N} \sum_{i=1}^N (|Hi_1'(i)| + |Hi_2'(i)|)} \quad (7)$$

$$FDM_3(f) = \frac{|Hi_1''(f)| - |Hi_2''(f)|}{\frac{7.2}{N} \sum_{i=1}^N (|Hi_1''(i)| + |Hi_2''(i)|)} \quad (8)$$

$$FDM(f) = 2(|FDM_1(f)| + |FDM_2(f)| + |FDM_3(f)|) \quad (9)$$

10. Calculate the single value of FDM. This is done in exactly the same way as for the ADM.



11. Calculate the FDM confidence histogram. This is done in exactly the same way as was done for the ADM.
12. Obtain the point-by-point GDM value. The GDM is premised on the ADM and FDM being largely independent, which means that:

$$GDM(f) = \sqrt{ADM(f)^2 + FDM(f)^2} \quad (10)$$

13. Calculate the overall GDM value and the GDM confidence histogram. This follows the same procedure as the ADM and FDM.
14. Determine the equivalent visual scale values for ADM, FDM, and GDM. As discussed previously, the FSV values can be categorized into a six point, natural language scale. The VRS is presented in section 7.3. This is done using a piecewise conversion. The piecewise approach for this is given in Table 4, where X is the ADM, FDM, or GDM (total or point-by-point), and V is the transformed (visual) scale equivalent.

**Table 4 – Piecewise visual conversion**

If $X \leq 0.1$ , then $V = 1 + 10X$
If $X > 0.1$ and $X \leq 0.2$ , then $V = 2 + 10(X - 0.099)$
If $X > 0.2$ and $X \leq 0.4$ , then $V = 3 + 5(X - 0.199)$
If $X > 0.4$ and $X \leq 0.8$ , then $V = 4 + 2.5(X - 0.399)$
If $X > 0.8$ and $X \leq 1.6$ , then $V = 5 + 1.25(X - 0.799)$
If $X > 1.6$ , then $V = 6$