The
University
Of
Sheffield.

Metamodelling for Auxetic Materials

By

Tze Ling Lew

A thesis submitted for the degree of
Doctor of Philosophy
in the
Faculty of Engineering
Department of Mechanical Engineering

October 2005

Acknowledgement

I would first like to thank my family for their unconditional supports and encouragements. I would also like to send my greatest gratitude to my supervisors, Dr. Fabrizio Scarpa and Professor Keith Worden, for their guidance, advice, support, encouragement and teaching in various aspects during my study in the University of Sheffield, and I send my best wishes to Dr. Scarpa for his new career in University of Bristol.

I would also like to thank Mr. Andrew Spencer for assisting me in Java programming with great patience; Dr. Jem Rongong for his time and generous helps in my thesis and work, also many thanks to Dr. Amanda Rutherford from the Los Alamos National Laboratory, for her generous help in providing her simulation data. I would like to acknowledge the University of Sheffield for offering me the Overseas Element Fee Bursary and Exeter Advance Technology Enterprise for manufacturing some of the chiral honeycomb experimental samples with their state of art rapid prototyping technology.

I would also like to extend my thanks to the university technical staffs, Mr. John Goodliffe and Mr. Dave Webster who helped me in experiments; all my colleagues in the Dynamics Research Group and many others within and outside the university, who has contributed a lot of memorable events throughout my stay in Sheffield. Finally I would like to send my appreciation to my friends from all over the world (though some have never met before), whom their generous transfers of knowledge and resources, encouragement and warmth have travelled beyond any geographical constraints.

Summary

The use of Finite Element (FE) based homogenisation has improved the study of composite material properties. A homogenisation is a method of averaging a heterogeneous domain by using a replacement unit cell according to the proportions of constituents in the domain.

However, the homogenisation method involves enormous computational effort when implemented in engineering design problems, such as optimisation of a sandwich panel. The large number of computations involved can rule out many approaches due to the expense of carrying out many runs. One way of circumnavigating this problem is to replace the true system by an approximate surrogate model, which is fast-running compared to the original. In traditional approaches using response surfaces, a simple least-squares multinomial model is often adopted. In this thesis, a Genetic Programming model was developed to extend the class of possible models by carrying out a general symbolic regression. The approach is demonstrated on both univariate and multivariate problems with both computational and experimental data. Its performances were compared with Neural Networks – Multi-Layer Perceptrons (MLP) and polynomials.

The material system studied here was the auxetic materials. The auxetic behaviour means that the structure exhibits a negative Poisson's ratio during extension. A novel auxetic structure, chiral honeycomb, is introduced in this work, with its experiments, analytical and simulations. The implementations of the auxetic material surrogate models were demonstrated using optimisation problems. One of the optimisation problems was the shape optimisation of the auxetic sandwich using Differential Evolution. The shape optimisation gives the optimal geometry of honeycomb based on the desired mechanical properties specified by the user.

The thesis has shown a good performance of numerical homogenisation technique and the robustness of the GP models. A detailed study of the chiral honeycomb has also given insight to the potential application of the auxetic materials.

Table of Content

List of Figures

List of Tables

List of Publications

i. Journals

I. **T. L. Lew**, F. Scarpa and K. Worden, *Homogenisation Metamodelling of Perforated Plates*, Strain, 40(3), 2004, pp.103 - 112.

II. **T. L. Lew**, A. B. Spencer, F. Scarpa, K. Worden, *Surface Response Optimisation of Auxetic Homogenised Cellular Plates using Genetic Programming*, Computational Methods in Science and Technology, 10(2), 2004, pp. 169 – 181.

III. **T. L. Lew**, A. B. Spencer, F. Scarpa, K. Worden, A. Rutherford and F. Hemez, *Identification of Response Surface Method using Genetic Programming*, 2004, accepted in Dec. 2005 by Mechanical Systems and Signal Processing.

IV. F. Scarpa, S. Blain, **T. L. Lew**, D. Perrott, M. Ruzzene and J. R. Yates, *Analytical and Experimental Analysis on The Elastic Buckling of Hexagonal Chiral Cell Honeycombs*, submitted to Composite Part A.

V. Paulhac, F. Scarpa, **T. L. Lew**, D. Perrot and J. R. Yates, *The Linear Elastic Transverse Properties of Hexagonal Chiral Honeycombs*, to be published in Composite, submitted to International Journal of Mechanical Sciences.

ii. Conferences

VI. **T. L. Lew**, A. B. Spencer, F. Scarpa, K. Worden, A. Rutherford and F. Hemez, *Identification of Response Surface Method using Genetic Programming*, Proceedings of ISMA 2004, Leuven, 2004, pp. 3287-3299.

VII. **T. L. Lew**, A. Spadoni, F. Scarpa, M. Ruzzene, *Chiral Hexagonal Cellular Sandwich Structure: A Vibroacoutic Assessment*, to be published in Proceedings of SPIE, Smart Structures and Materials 2004: Damping and Isolation.

iii. Posters

VIII. **T. L. Lew**, F. Scarpa and K. Worden, *Estimation of Chiral Honeycomb Properties Using Neural Network*, Auxetic Materials Network: Abstract of Papers, Poster exhibition in the AUXETNET first annual young researchers forum, University of Sheffield, 2004, pp. 22 – 24. (Best poster Award)

IX. **T. L. Lew**, F. Scarpa, K. Worden and M. Ruzzene, *Estimation of Chiral Honeycomb Out-of-Plane Shear Modulus Using Neural Network (Multi-Layered Perceptrons)*, Auxetic Materials Network: Poster exhibition in the AUXETNET second annual young researchers forum, University of Cambridge, 2005.

1. Introduction

Advances in technology have brought about many innovative creations of new materials with outstanding properties. These newly designed and manufactured material systems have replaced many natural solids in various industrial applications, for example alloys and polymers being used from laboratory tools, engineering structures to household products that can withstand high heat and corrosions. Material design has evolved to obtain not only composite structural materials, but functional, multifunctional (by combining a few functional materials) or even intelligent materials (a more sophisticated combination of functional materials to form an intelligent material system), such as Shape Memory Alloys (SMA), piezoelectric materials, magneto- / electro-rheological fluids, Micro-Electro Mechanical Systems (MEMS) etc. One of the new fields of endeavour is the domain of auxetic materials.

The auxetic materials are material systems that exhibit negative Poisson's ratios under loading conditions. This counterintuitive property has shown some potential in several engineering applications and offered enhanced performance compared to conventional materials in various ways. The auxetic material itself may be a functional material, such as the smart bandage that applies medicine according to the swelling condition [55] or smart filters with automatic defouling capability [73, 82]. The auxetic material may also be combined with a functional material, such as the SMA honeycomb that is discussed in [26] and combined with a piezoelectric sandwich face to form a smart sandwich structure [25]. Other examples of auxetic material applications include enhanced pulling-resistance fibre composites [67]. The current drive is on using auxetic materials as platforms for multifunctional structures, with embedded sensor capabilities for structural health monitoring, and microwave applications [65, 70] and many others [1].

Composites with auxetic materials have unusual mechanical properties, such as higher shear rigidity, enhanced indentation resistance and bending stiffness [1, 82]. Auxetic material systems with extreme hardness, polymers of negative thermal expansion, and composites with auxetic cores have demonstrated potential applications in multifunctional sandwich structures and are within the reach of current manufacturing and process technology. The details of auxetic materials, including their background and current applications will be discussed in later sections.

The investigations of the auxetic cellular materials described in this work lead to a ground breaking understanding of their mechanical behaviour in the sense of fundamental understanding and their possible engineering applications. Analytical, computational and experimental studies of such material systems offer a better understanding of the known

mechanisms responsible for their unusual properties. It also helps suggest new mechanisms, which may lead to achieving novel designs of material systems under given conditions.

The theme of this thesis is to develop novel metamodelling techniques to improve the effectiveness of cellular material simulations (with sepecial emphasis on auxetic), which are based on a Finite Element Analysis (FEA) approach, for engineering design optimisation. Complex simulation models of proposed real systems are often used to make decisions on the actual alteration to a system design. The FE models related to auxetic materials and structures may be quite complex, therefore reduced order approximations of the material properties are constructed via data fitting. The metamodel method proposed in this work allows a synthesis of the FE computed output with respect to the auxetic structural geometries, in order to create an accurate approximation of the FE models, also known as the metamodel.

Numerical computational tools like the Finite Element Analysis (FEA) have significantly increased the progress of engineering analysis over the last 50 years. However, their intensive computational requirements can still be a hurdle for today's engineering design. Efforts have been made over the years to overcome this issue, such as taking advantage of symmetry, or, for periodic objects, by modelling only the periodic / unit cell. Despite these approaches, researchers in the FE domain are still involved in searching for ways to boost the computational speed in order to cope with the increasingly complex design problems.

When the design problem is incorporated with explicit simulations, typically FE simulations, or experiments that determine the design parameters; depending on their complexity, they can be bottlenecks, which slow the optimisation process of the design problem. A typical engineering design problem would involve the handling of many design parameters; some parameters may be complicated functions instead of plain single variables. The function can be a description of a simulation or experimental data in a predefined range. Therefore, researchers identified polynomials, splines, wavelet functions etc. to describe these datasets efficiently. This process is called metamodelling.

The intention of metamodelling is to introduce a fast running surrogate / replacement model of the simulation or experiment. There are various metamodels to choose from, each having different complexity and a performance extent. For example, the parametric models (eg. empirical formulae, polynomials etc) are fast to implement but require prior knowledge of the system (e.g. the simulation / experiment); while the flexible models (examples given below) can fit any system with very little knowledge of the actual system, but with more effort put in to construct the model (training effort), depending on the complexity of the system. Figure 1.1

briefly demonstrates the standpoint of different metamodels in the area of structural dynamics problems [17].

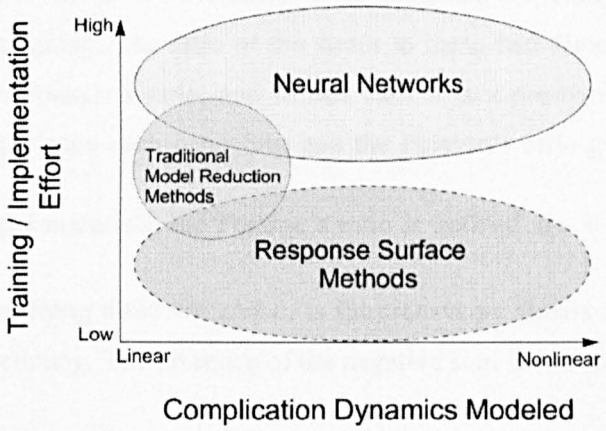


Figure 1.1 Implementation effort and complexity of various classes of metamodels [17].

The neural network showed here is just an example among various flexible models; other examples of the flexible models include wavelet functions, splines, genetic programming and other evolutionary programming approaches. The response surface method indicated by [17] is a physics – based parametric model, with extensive prior knowledge of the system to accommodate for nonlinearities included in the model. The physics – based parametric model used by [17] is a nonlinear polynomial function describing the damage condition of a multi-DoF system. The traditional reduced models are mostly linear parametric models and do not model stochastic phenomenon and nonlinear dynamic behaviour effectively.

In this work, the data used for developing the metamodels consist mainly of FE simulations of periodic cellular structures such as honeycombs, with auxetic properties (experiencing negative Poisson's ratio), used for sandwich composite applications. The FE simulations were performed using the homogenisation method, namely a generalised method for representing a periodic medium. The data were then collected to create surrogate models using Neural Networks (NN) or Genetic Programming (GP). The NN model used here is a Multi-Layered Perceptron (MLP), developed as in [81]. The reason for choosing MLP and GP is due to their high flexibility to fit most functions. Though the traditional honeycombs (hexagonal in shape, or inverted hexagonal shown in Figure 1.2) have their analytical formulae developed by [23], most other structures, especially complicated ones, may not have such convenience. For example one of the structures studied in this work, the chiral honeycomb, is still not widely understood. The character of the chiral honeycomb is still at its early stage of being investigated. Therefore, it is handy to have these flexible metamodels to give rough estimations of the complex structures' behaviours in relation to their geometry setup, in order to avoid intensive FE modelling for every variation of the structures' behaviour that correspond to a minor change of its geometry.

i. An Introduction to Auxetics

When a conventional material is stretched; it is expected to contract in the direction perpendicular to the stretching. The ratio of the strain in these two directions (with the minus sign) is defined by the Poisson's ratio, and in this case it is a positive ratio. Most naturally occurring materials experience such behaviour and the Poisson's ratio generally falls between 0.2 and 0.4. For isotropic materials, the Poisson's ratio is defined as $\nu = -\frac{\varepsilon_y}{\varepsilon_x}$, where ε_x is the tensile strain in the stretching direction and ε_y is the transverse tensile strain in the direction perpendicular to the stretching. The presence of the negative sign is to ensure a positive value of the ratio.

In theory, the elastic isotropic materials can have their Poisson's ratio ranging between -1.0 and 0.5. When a material with a negative Poisson's ratio is stretched, it will expand in one if not all the other directions, exhibiting therefore an unusual volumetric deformation. Lakes first found a material with negative Poisson's ratios in 1987 [82], named as anti-rubber, dilatational, and auxetic by later researchers. The term auxetic is used throughout this thesis, it is named by [18] as $\alpha\nu\xi\varepsilon\tau\sigma\sigma$ (pronounced as 'auxetic'. Originating from the Greek: 'that may be increased', referring to the width and volume increase when stretched), to avoid the cumbersome phrase: 'material with negative Poisson's ratio'.

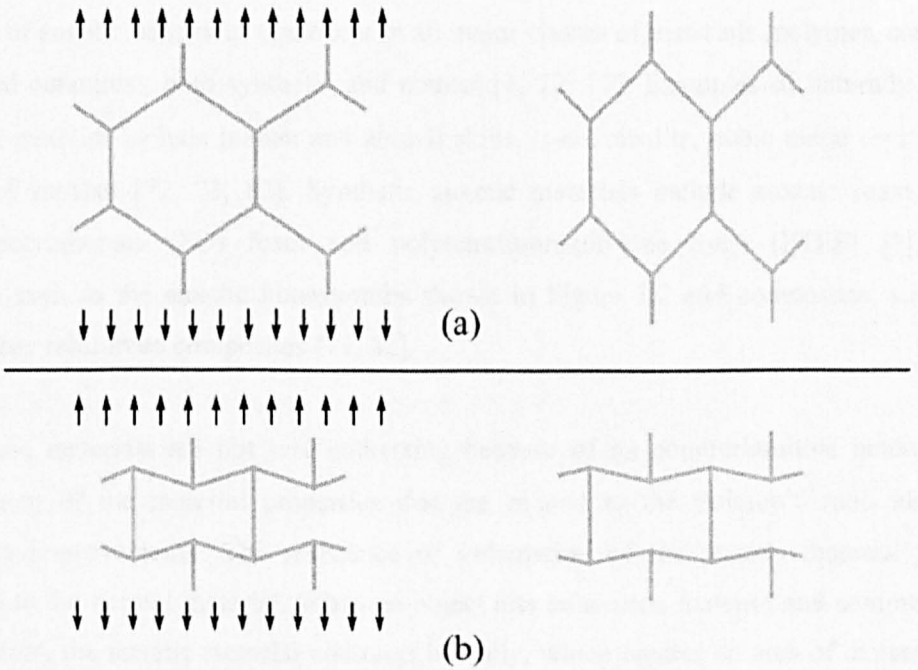


Figure 1.2 (a) Conventional Honeycomb (b) Re-entrant / inverted (auxetic) Honeycomb under deformation.

The conventional hexagonal unit cell of a regular honeycomb in Figure 1.2(a) can be made auxetic with a re-entrant / inverted angle as shown in Figure 1.2(b), and it exhibits lateral expansion when pulled along one direction. The re-entrant / inverted honeycombs are highly anisotropic and feature a stronger shear resistance compared to conventional honeycombs [82, 63], other properties include directional band-gaps in flexural wave propagations and strong dielectric anisotropic properties which can be used in conjunction with the mechanical anisotropic properties to design electromagnetic compatibility characteristics in microwave absorbers [59, 65].

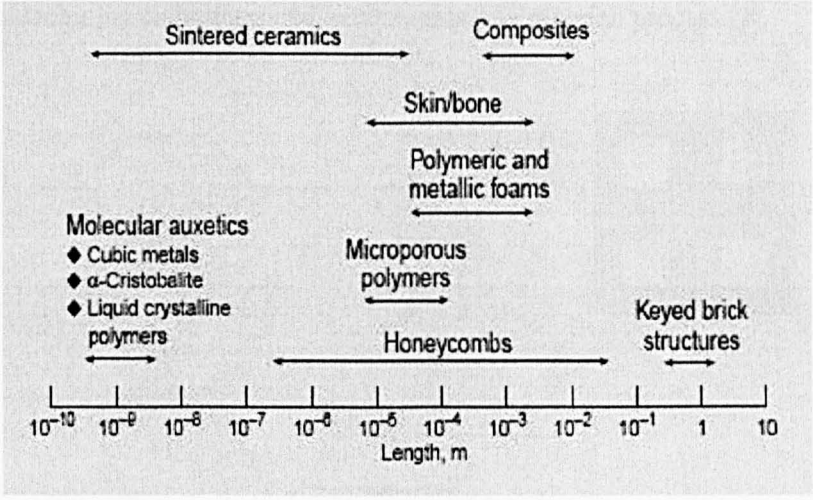


Figure 1.3 Examples of auxetic materials in different scales and different material systems [1].

Auxetic behaviour does not only occur in man-made structures, Figure 1.3 shows some examples of auxetic behaviour that occur in all major classes of materials (polymer, composites, metals and ceramics), both synthetic and natural [1, 72, 73]. Examples of naturally occurred molecular auxetics include human and animal skins, α-cristobalite, cubic metal crystals and a number of zeolites [72, 73, 82]. Synthetic auxetic materials include auxetic foam, such as auxetic polyutherane (PU) foam and polytetrafluoroethylene foam (PTFE) [1], auxetic structures such as the auxetic honeycombs shown in Figure 1.2 and composites, such as the auxetic fibre reinforced composites [73, 82].

The auxetic materials are not just interesting because of its counterintuitive behaviour, the enhancement of the material properties that are related to the Poisson's ratio also shows significant improvements. The resistance of indentation of the auxetic material is higher compared to the normal material. When an object hits an auxetic material and compresses it in one direction, the auxetic material contracts laterally, which creates an area of denser material around the vicinity of impact and resists the impact. Also, for material with Poisson's ratio of -1, it corresponds to a very high shear modulus with respect to the bulk modulus, which is ideal for

most structural design [1]. It is also known that the acoustic damping of the auxetic materials overtakes the normal materials [1, 59, 65].

The current applications and manufacturing of auxetic materials are still to be explored, some existing applications include pyrolytic graphite for thermal protection in aerospace applications, auxetic composite materials for crashworthiness in structures, some foreseen medical applications such as prosthetic materials, surgical implants, dilators to open up blood vessels, auxetic filters with enhanced defouling capabilities, etc [73, 82]. The manufacturing of auxetic foams include LIGA technology, laser stereolithography, molecular self-assembly, silicon surface micromachining techniques and nanomaterials fabrication process [1].

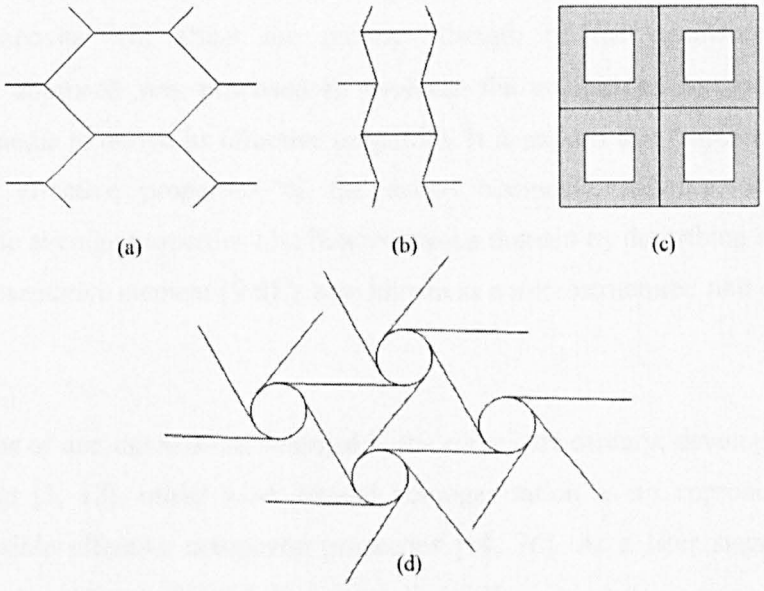


Figure 1.4 Examples of centresymmetric and non-centresymmetric models used in this work. (a) Conventional hexagonal honeycomb; (b) Auxetic hexagonal honeycomb; (c) Square cells and (d) Chiral honeycomb.

The honeycombs used in this work, as shown in Figure 1.4, include the above mentioned centresymmetric hexagonal honeycomb and also the non-centresymmetric honeycomb. The centresymmetric models include squares, hexagons and re-entrant hexagons (Figure 1.4 (a) – (c)), while the example for non-centresymmetric is the chiral honeycomb (Figure 1.4 (d)). The centresymmetric characteristic indicates that the unit cell model is symmetric at its centre; on the other hand, the non-centresymmetric, like the chiral honeycomb, does not have symmetry along the normal horizontal and vertical directions, however, it has a rotational symmetry in the local cells.

ii. Homogenisation

Composite materials are well known for their excellent characteristics, such as high strength ratios, corrosion resistivity etc, which cannot be achieved by uniform materials. However, it is impossible to theoretically evaluate the exact properties of a highly heterogeneous composite. Conventional studies mainly assume the heterogeneous composite material is a homogeneous orthotropic material using the law of mixtures [36]. The information obtained using such a method evaluates only the mean value of the microscopic information.

The microstructure information for the composite is required in order to improve the design of the composite material. For example, determining the stress concentration in a material interface within the composite will affect the precise strength of the resultant structure. The homogenisation approach was proposed to evaluate the composites by approximating the heterogeneous media to derive its effective properties. It is an idea that 'replaces by a material' possessing the effective properties of the actual composite mixture, in other words, characterising the average properties of a heterogeneous domain by describing it as an effective volumetric representative element (VRE), also known as a microstructure / unit cell [2, 3, 12, 30, 60, 61].

The physical idea of homogenisation emerged in the early 20th century, developed by Rayleigh, Reuss and Voigt [3, 12]. Initial work treated homogenisation as an approach to define the bounds for possible effective composite properties [14, 76]. At a later stage, it became an averaging method to estimate the effective properties [12].

A mathematical approach to homogenisation theory in engineering composite analysis appeared in the 1970's [3]. Since then, a considerable amount of research activities have been carried out in the analytical and numerical homogenisation domain. Periodic homogenisation, as used in this work, is a heuristic method; it was proposed about a decade ago by using the double scale asymptotic expansion and periodicity assumption. Homogenisation was then used as an approximation method [3, 83] to determine the global properties through the knowledge of the effective VRE / unit cell properties.

Fundamental references on the theory can be found, among others, in [2, 3, 12, 27, 30, 37, 60, 61]. This method was later proposed for topology optimisation [24, 38, 39, 45, 52, 75], in order to obtain an optimum shape and topology of a continuum structure. The details of the double scale asymptotic expansion / periodic homogenisation theory will be discussed in a later chapter.

iii. Metamodelling

The design of a composite sandwich panel involves more than one FE simulation to determine its properties. The situation becomes worse when the design involves a constant update of the parameters that affect the properties of the composite. For a design problem that is associated with some experimentally or numerically determined parameters, the heavy computational effort to determine such parameters makes the problem non-trivial. Therefore metamodelling techniques such as MLPs, polynomials, GP and others are introduced, in order to study the query-and-response of the experiment / simulation, and to reduce the computational time. The query-and-response of the experiment / simulation is the input-output relationship of a system. For example, in a FE model, which is a system, the geometry of the model affects its property; therefore the input-output relationship of the FE model is the geometry-property relationship.

The query-and-response procedure is often by trial and error and the researcher may never fully understand the underlying input-output relationship and therefore never be able to identify the best setting for input values, for the specific design problem.

The basic approach to this problem is to develop an approximation of the analyses that is more efficient to execute, and to obtain more 'insight' into the relationship between the input and output. This approximate / surrogate model of the original is called a metamodel.

The creation of the metamodel involves choosing a function in order to represent the data in the most appropriate way. Whether linear regression, quadratic or even Fourier analysis, all require some form of background knowledge of the experimental/simulation data that one is attempting to fit. Once the proper function is chosen, one proceeds to a series of analyses.

A traditional metamodelling approach often involves using a multivariate response surface model. A general response surface method would be first to screen the data to reduce the set of design variables that are most influential to the output; this will require some prior knowledge of the system. Then with the chosen function, one would start to perform a general fitting, followed by some refining procedures. If the researcher decided that such function doesn't give a promising fit, alternative functions may then be considered.

The research in traditional metamodelling reached its mature state some time ago, but there are still some known problems, such as difficulties with the polynomial basis functions, problems in trying to fit a highly nonlinear model, explosion of coefficients etc.

In short, the creation of the surrogate model requires both the identification of the correct function that fits the data and the appropriate estimation of the function's coefficients. The problem can be seen as finding a proper function, in symbolic form, which fits a given finite sample of data, or symbolic regression. [41]

Alternative methods include Artificial Neural Networks (ANN), Genetic Programming (GP) and other evolutionary and adaptable algorithms. These methods are well known for their wide range of applications, flexibility and independence of prerequisite knowledge of the problems.

The metamodels used in this work include NNs and GP. The NN model used here is the Multi-Layer Perceptron (MLP) [80, 81], while the GP was created using the Java programming language [19] during the research period. The performances of these evolutionary metamodels are compared with the traditional polynomial least square models.

In the following chapter, theory and detailed background regarding homogenisation, metamodelling, neural networks – MLPs and genetic programming (GP) will be discussed. A GP code was also developed in this work, and its performance is benchmarked with the MLP in Chapter 3. The efficiency of using the numerical homogenisation method to develop the FE models in representing the structures and the convergence of the numerical homogenisation method is benchmarked in Chapter 4. Application of the GP and NN in structural design was also studied in this chapter. Subsequent testing on the numerical homogenisation in cellular solids is demonstrated in Chapter 5. Also a novel application of the metamodels (GP and NN) in a hexagonal honeycomb is included, whereby a case study utilising the Differential Evolution (DE, another example of flexible model but not used as a metamodel here) is proposed for a shape optimisation of the cellular structure. Chapter 6 studies the chiral honeycomb out-of-plane properties. Derivations of analytical formulae and simulations on the chiral honeycomb out-of-plane structural modulus, shear modulus and critical buckling stress were developed and were also validated by some experiments. The chapter is also accompanied by a shape design optimisation case study of a chiral sandwich panel, which involves cascading the metamodels into an empirical sandwich modal density formula taken from [56].

2. Theoretical Background

In this chapter, the theory and detailed background regarding homogenisation, metamodelling, neural networks – MLPs and Genetic Programming (GP) will be discussed for the completeness of the thesis.

The homogenisation method used here is a double scale asymptotic equation developed heuristically [3, 83]. It is used to simplify the simulation of a complex heterogeneous domain, by representing it using a unit cell, which is composed of all the components of the heterogeneous domain according to their proportion. The unit cell also requires a periodicity constraint to define its continuity in the heterogeneous domain. This technique was chosen to study the material structures in this work as it is widely used among researchers and its effectiveness has also been proven in the literatures [12, 14, 24, 27, 37, 38, 39, 40, 41, 61, 83, 85]. The numerical implementation of the homogenisation is also included in Appendix 6. The theory of the homogenisation is referenced from [12, 24, 27].

For the metamodelling, a general overview of the backbones of metamodels will be discussed. Followed by the theory of the MLP and GP, the latter was developed during this research, in order to test for its robustness when compared to the MLP. The theory of the MLP is referenced to [81], while the GP theory is mostly taken from [41]. The reader is suggested to refer to [15, 41] for further details of metamodelling and the GP topic respectively.

i. Homogenisation

A complex heterogeneous domain is numerically expensive to model using Finite Element Methods (FEM) as mentioned in the previous chapter, and the homogenisation method was developed to cope with this issue, as it computes the heterogeneous domain by looking into only its 'periodic unit cell', much like seeing the world through a grain of sand.

A domain is considered periodic when it obeys the following relationship [27]:

$$\Phi(\mathbf{x}+\mathbf{NY})=\Phi(\mathbf{x}) \tag{2.1}$$

where Φ is an arbitrary function describing the property of the domain, it can be a scalar, vectorial or tensorial function of a point $\mathbf{x}$, such as the stiffness or compliance matrix. The fast varying property at the microscopic level is on the left hand side, and the slow varying property at the macroscopic level is on the right. Considering a $\mathbf{R}^3$ space of the heterogeneous domain, the position vector $\mathbf{x}$ is (1×3), it is the coordinates of a point $\mathbf{x}$ in the domain, $\mathbf{Y}$ is also (1×3), it is the constant vector determining the period of the structure in each direction and $\mathbf{N}$ is a (3×3)

diagonal matrix containing arbitrary integers acting as a scale-factor of **Y**. As shown in Figure 2.1, the periodicity also requires that the functions have equal values at each of the points P_1, P_2, P_3 and P_4.

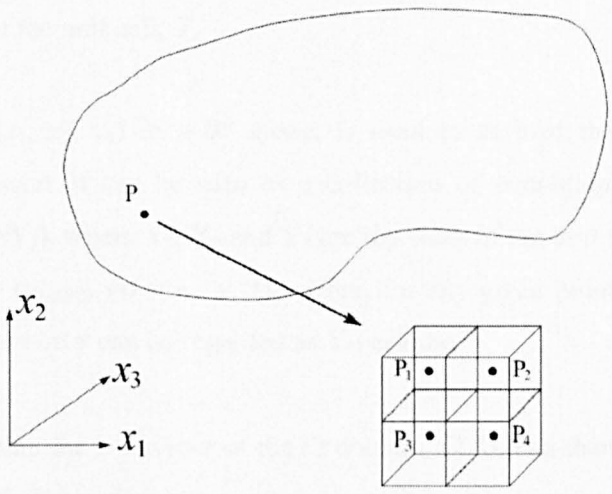


Figure 2.1 A representation of a R^3 domain, with point P magnified to show its periodicity.

The theory of homogenisation suggests that the period **Y** compared to the dimensions of the overall domain is very small, and the heterogeneity of the domain is varying rapidly within a very small interval, $\partial \mathbf{x}$. This can be viewed as a double-scale dependence: one slow variation, large scale describing the macroscopic level x, and one fast variation, small scale describing the microscopic level y of the composite domain. The ratio of these two coordinates is denoted by:

$$\varepsilon = x/y \tag{2.2}$$

It indicates the proportion between the dimensions of the unit cell and the whole domain. Consequently, if Φ is a general function describing the domain, then it also has the double-scale dependence:

$$\Phi(\mathbf{x}) = \Phi(\mathbf{x}, \mathbf{y}) = \Phi(\mathbf{x}, \varepsilon/\mathbf{x}) \tag{2.3}$$

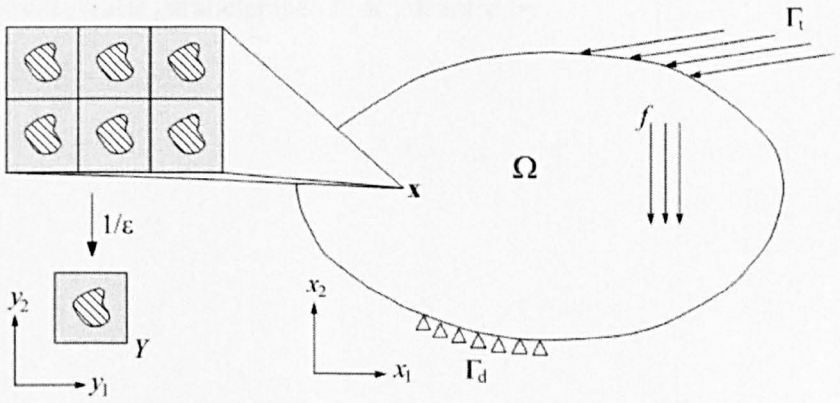


Figure 2.2 An equilibrium domain with different periodicity at point x.

To illustrate the technique, consider the domain Ω in Figure 2.2, with this double-scale expansion, the domain can be seen as the unit cell in the figure. The parameter ε indicates the proportion between the unit cell and the whole domain. The inverse of ε can be seen as a magnification factor of the unit cell, $\boldsymbol{Y}$.

If a coordinate $\mathbf{x} = (x_1, x_2, x_3)$ in a $\mathbf{R}^3$ space, is used to defined the Ω domain; with the periodicity of the domain, it can be seen as a collection of parallelepiped cells of identical dimension (εY_1, εY_2, εY_3), where Y_1, Y_2 and Y_3 are the sides of the unit cell $\boldsymbol{Y}$ in a microscopic coordinate system $\mathbf{y} = (y_1, y_2, y_3) = \mathbf{x} / \varepsilon$. Therefore, for any given point $\mathbf{x}$ in the macroscopic domain, any dependency on $\mathbf{y}$ can be regarded as $\mathbf{Y}$-periodic.

The function determining the behaviour of the Ω domain, (2.3), can then be expanded (using a double-scale asymptotic expansion) as:

$$\Phi^{\varepsilon}\left(\mathbf{x}, \mathbf{y} = \frac{\mathbf{x}}{\varepsilon}\right) = \Phi^{0}(\mathbf{x}, \mathbf{y}) + \varepsilon\Phi^{1}(\mathbf{x}, \mathbf{y}) + \varepsilon^{2}\Phi^{2}(\mathbf{x}, \mathbf{y}) + \ldots \qquad \textbf{(2.4)}$$

when $\varepsilon \to 0$, the right hand side of the above equation are smooth with respect to $\mathbf{x}$ and periodic in $\mathbf{y}$. The smoothness of the unit cell within the domain means that, for different points $\mathbf{x}$ within the domain, the composition may vary, but if one looks through a microscope at a point, a periodic pattern can be found.

For simplicity, a linear elastic domain comprised of only a single unit cell without pre-stresses is considered. Before explaining the theory of homogenisation, the problem illustrated in Figure 2.2 is first defined: Assume that the domain, Ω, is an open subset of $\mathbf{R}^3$. The domain has a body force f, surface traction Γ_t and displacement constraints Γ_d at its boundary, Γ.

The unit cell $\boldsymbol{Y}$, which makes up the domain, is illustrated in Figure 2.3, and it is also assumed to be an open rectangular parallelepiped in $\mathbf{R}^3$, denoted by

$$\boldsymbol{Y} =]0, y_1[\times]0, y_2[\times]0, y_3[\qquad \textbf{(2.5)}$$

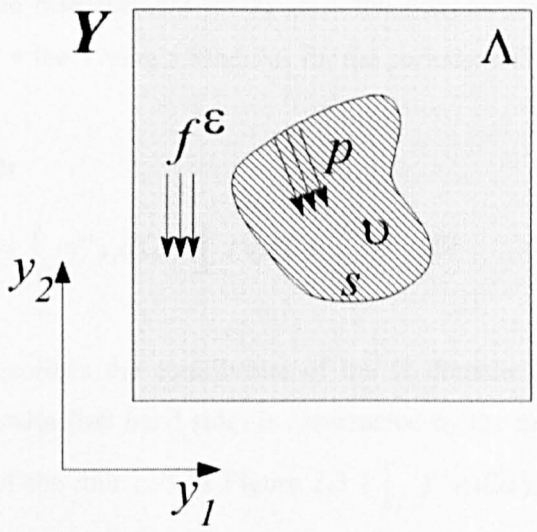


Figure 2.3 A unit cell of a homogeneous domain.

Also, assume that there is a hole υ in the unit cell; the boundary of the hole is defined as $\partial\upsilon = S$. The hole is assumed to be smooth in order not to have a traction force occur within the hollow. The solid part of the unit cell is Λ. Therefore the solid part of the macroscopic domain can be described as

$$\Omega^\varepsilon = \left\{ \mathbf{x} \in \Omega \,|\, \left(\mathbf{y} = \mathbf{x}/\varepsilon \right) \in \Lambda \right\}$$

Also define

$$S^\varepsilon = \bigcup_{i=1}^{\text{all cells}} s_i$$

which states that every hole, s_i, in every unit cell, belongs to the set S^ε. None of the holes s_i interacts with the boundary Γ of the Ω domain. Also, let the *admissible linear manifold* be

$$\mathbf{V}^\varepsilon = \left\{ \upsilon \in \left(H^1\left(\Omega^\varepsilon\right) \right)^3 \,\&\, \upsilon|_{\Gamma_d} = 0 \right\}$$

where $\upsilon|_{\Gamma_d}$ is the value of υ on the boundary of Γ_d [24] and H^1 is a Sobolev space [27], which grades the continuity.

Now, consider the deformation **u** of the body Ω^ε, subject to the body force *f*, surface traction *t* on boundary Γ_t and prescribed displacement on Γ_d, together with the traction *p* inside the holes S^ε, this can be stated as:

Consider the stress-strain and strain-displacement relationship:

$$\sigma_{ij}^\varepsilon = E_{ijkl}^\varepsilon e_{kl}^\varepsilon \tag{2.6}$$

$$e_{kl}^\varepsilon = \frac{1}{2}\left(\frac{\partial u_k^\varepsilon}{\partial x_l} + \frac{\partial u_l^\varepsilon}{\partial x_k} \right) \tag{2.7}$$

where u^ε = the periodic displacement on Y; σ^ε = the periodic axial stress on Y; e^ε = the periodic strain on Y; E^ε = the Young's Modulus for the periodic cell Y.

Find $\mathbf{u}^\varepsilon \in \mathbf{V}^\varepsilon$, such that

$$\int_{\Omega^\varepsilon} E_{ijkl} \frac{\partial u_k^\varepsilon}{\partial x_l} \frac{\partial v_i}{\partial x_j} d\Omega = \int_{\Omega^\varepsilon} f_i^\varepsilon v_i d\Omega + \int_{\Gamma_t} t_i v_i d\Gamma + \int_{S^\varepsilon} p_i^\varepsilon v_i dS \qquad \forall \mathbf{v} \in \mathbf{V}^\varepsilon \tag{2.8}$$

The equation above describes the equilibrium of the Ω domain (Figure 2.2), whereby the displacement of the domain (left hand side) is constructed by the displacement caused by the internal / body force of the unit cell in Figure 2.3 ($\int_{\Omega^\varepsilon} f_i^\varepsilon v_i d\Omega$), the displacement due to traction of the solid part ($\int_{\Gamma_t} t_i v_i d\Gamma$), and the displacement due to the traction of the hole ($\int_{S^\varepsilon} p_i^\varepsilon v_i dS$).

Also the elastic constants have symmetry and coercivity properties:

$$E_{ijkl} = E_{jikl} = E_{ijlk} = E_{klij}$$

$$\exists \alpha > 0 : E_{ijkl}^\varepsilon e_{ij} e_{kl} = \alpha e_{ij} e_{ij}, \forall e_{ij} = e_{ji}$$

To solve this problem, a homogenisation method is used. The method involves first assuming a double-scale asymptotic expansion to be the solution of the above problem. Upon substituting this double-scale expanded solution into the statement, compare both sides with identical terms. A 'unit cell problem' is deduced from the comparisons. With this 'unit-cell problem', a final homogenised equation will be obtained.

To begin with, a unique solution of $\mathbf{u}^\varepsilon$ is assumed to exist when the functions t, f and p are smooth and boundaries Γ_t, Γ_d and S^ε are regular [24]. Since the traction p and body forces and elastic constants vary within a small region, where the dimensions are much smaller than the macroscopic scale of the composite domain, i.e. they are **Y**-periodic / two-scale functions. Therefore, the solution for $\mathbf{u}^\varepsilon$ should also be a two-scale function:

$$\mathbf{u}(\mathbf{x}) = \mathbf{u}(\mathbf{x}, \mathbf{y}) = \mathbf{u}(\mathbf{x}, \mathbf{x}/\varepsilon) \tag{2.9}$$

Note that it is assumed that the traction t on the Γ_t boundary only depends on $\mathbf{x}$.

Expanding the periodic displacement u^ε using a double scale asymptotic method gives:

$$\mathbf{u}^\varepsilon\left(\mathbf{x}, \mathbf{y} = \frac{\mathbf{x}}{\varepsilon}\right) = \mathbf{u}^0(\mathbf{x}, \mathbf{y}) + \varepsilon \mathbf{u}^1(\mathbf{x}, \mathbf{y}) + \varepsilon^2 \mathbf{u}^2(\mathbf{x}, \mathbf{y}) + \ldots \tag{2.10}$$

Now, the task is to find each **u** term in the above equation. It is important to know that $\frac{\partial \mathbf{u}^{\varepsilon}}{\partial \mathbf{x}} = \frac{\partial \mathbf{u}}{\partial \mathbf{x}} + \frac{1}{\varepsilon}\frac{\partial \mathbf{u}}{\partial \mathbf{y}}$. Also, for a Y-periodic function, $\Psi(\mathbf{y})$ (e.g. structural properties of the domain), is said to be the integration of microscopic properties of the unit cell over the whole domain:

$$\lim_{\varepsilon \to 0^+} \int_{\Omega^{\varepsilon}} \Psi\left(\frac{\mathbf{x}}{\varepsilon}\right) d\Omega \to \frac{1}{|\mathrm{Y}|} \int_{\Omega} \int_{\Lambda} \Psi(y) dY d\Omega \tag{2.11}$$

and

$$\lim_{\varepsilon \to 0^+} \int_{S^{\varepsilon}} \Psi\left(\frac{\mathbf{x}}{\varepsilon}\right) d\Omega \to \frac{1}{\varepsilon |\mathrm{Y}|} \int_{\Omega} \int_{s} \Psi(y) dS d\Omega \tag{2.12}$$

each representing the solid part and hollow part of the unit cell respectively, where |Y| is the volume of the unit cell.

Substituting Equation (2.10) into (2.8) and comparing both sides with similar terms, will lead to the following:

$$\varepsilon^{-2}: \quad \frac{1}{|\mathrm{Y}|} \int_{\Omega} \int_{\Lambda} E_{ijkl} \frac{\partial u_k^0}{\partial y_l} \frac{\partial v_i}{\partial y_j} dY d\Omega = 0 \tag{2.13}$$

$$\varepsilon^{-1}: \quad \int_{\Omega} \left\{ \frac{1}{|\mathrm{Y}|} \int_{\Lambda} E_{ijkl} \left[\left(\frac{\partial u_k^0}{\partial x_l} + \frac{\partial u_k^1}{\partial y_l} \right) \frac{\partial v_i}{\partial x_j} + \frac{\partial u_k^0}{\partial y_l} \frac{\partial v_i}{\partial x_j} \right] dY \right\} d\Omega = \int_{\Omega} \left(\frac{1}{|\mathrm{Y}|} \int_{s} p_i v_i dS \right) d\Omega \tag{2.14}$$

$$\begin{aligned} \varepsilon^{0}: \quad & \int_{\Omega} \left\{ \frac{1}{|\mathrm{Y}|} \int_{\Lambda} E_{ijkl} \left[\left(\frac{\partial u_k^0}{\partial x_l} + \frac{\partial u_k^1}{\partial y_l} \right) \frac{\partial v_i}{\partial x_j} + \left(\frac{\partial u_k^1}{\partial x_l} + \frac{\partial u_k^2}{\partial y_l} \right) \frac{\partial v_i}{\partial y_j} \right] dY \right\} d\Omega \\ & = \int_{\Omega} \left(\frac{1}{|\mathrm{Y}|} \int_{\Lambda} f_i v_i dY \right) d\Omega + \int_{\Gamma_t} t_i v_i d\Gamma \end{aligned} \tag{2.15}$$

$\forall \boldsymbol{v} \in \mathbf{V}_{\Omega \times \Lambda}$. $\boldsymbol{v}$ is an arbitrary function representing the admissible displacement [12].

The next problem is to solve equations (2.13) - (2.15):

Multiplying (2.13) by ε^2 and taking the limit $\varepsilon \to 0^+$ of (2.11), then, integrating by parts and applying periodicity (i.e. the terms in opposite faces of the unit cell can be cancelled), yields

$$\frac{1}{|\mathrm{Y}|} \int_{\Omega} \left\{ \int_{\Lambda} \left[-\frac{\partial}{\partial y_j} \left(E_{ijkl} \frac{\partial u_k^0}{\partial y_l} \right) \right] v_i dY + \int_{s} E_{ijkl} \frac{\partial u_k^0}{\partial y_l} n_j v_i dY \right\} d\Omega = 0 \tag{2.16}$$

As $\boldsymbol{v}$ is arbitrary, this gives

$$-\frac{\partial}{\partial y_j} \left(E_{ijkl} \frac{\partial u_k^0}{\partial y_l} \right) = 0, \ \forall \mathbf{y} \in \Lambda \tag{2.17}$$

$$E_{ijkl} \frac{\partial u_k^0}{\partial y_l} n_j = 0 \ \text{ on } s. \tag{2.18}$$

As (2.17) has a unique form, a proposition is applied to solve this form of equation, which is stated as in [24, 27, 51]:

The problem $\frac{\partial}{\partial y_j}\left(E_{ijkl}(\mathbf{y})\frac{\partial \Phi(\mathbf{y})}{\partial y_l}\right) = F(\mathbf{y})$ has a solution in Y (i.e. the unit cell domain), where Y is periodic and has a smooth boundary, defined up to an additive constant, for a regular F, iff $\int_\Lambda F(\mathbf{y})dY = -\int_S E_{ijkl}\frac{\partial \Phi(\mathbf{y})}{\partial y_l}n_j dS = 0$.

Applying the above proposition, it follows from (2.17) and (2.18) that

$$\mathbf{u}^0 = \mathbf{u}(\mathbf{x}) \tag{2.19}$$

This indicates that the first term of the $\mathbf{u}^\varepsilon$ expansion only depends on the macroscopic scale $\mathbf{x}$.

Now, solving the ε^{-1} term:

Multiplying (2.14) by ε and taking the limit $\varepsilon \to 0^+$ of (2.11) and (2.12), then, integrating by parts and applying periodicity, yields

$$\int_\Omega\left[\frac{1}{|Y|}\int_\Lambda E_{ijkl}\left(\frac{\partial u_k^0(x)}{\partial x_l}+\frac{\partial u_k^1}{\partial y_l}\right)\frac{\partial v_i}{\partial y_j}dY\right]d\Omega = \int_\Omega\left(\frac{1}{|Y|}\int_S p_i v_i dS\right)d\Omega \tag{2.20}$$

Since the above equation is satisfied for any v, choosing $v = v(y)$ results in:

$$\int_\Lambda E_{ijkl}\left(\frac{\partial u_k^0(x)}{\partial x_l}+\frac{\partial u_k^1}{\partial y_l}\right)\frac{\partial v_i(y)}{\partial y_j}dY = \int_S p_i v_i(y)dS \tag{2.21}$$

One integrates (2.21) by parts and applys the divergence theorem (i.e. in the absence of the creation / destruction of matter, the density within a region of space can change only by having it flow into or away from the region through its boundary [33]), then on applying the periodicity condition, it becomes:

$$\int_\Lambda E_{ijkl}\left(\frac{\partial u_k^0(x)}{\partial x_l}+\frac{\partial u_k^1}{\partial y_l}\right)v_i dY + \int_\Lambda E_{ijkl}\left(\frac{\partial u_k^0(x)}{\partial x_l}+\frac{\partial u_k^1}{\partial y_l}\right)n_j v_i dY = \int_S p_i v_i dS \tag{2.22}$$

As v is arbitrary, canceling it from both sides, will lead to:

$$-\frac{\partial}{\partial y_j}\left(E_{ijkl}\frac{\partial u_k^1}{\partial y_l}\right) = \frac{\partial}{\partial y_j}\left(E_{ijkl}\frac{\partial u_k^0(x)}{\partial x_l}\right), \text{ on } \Lambda \tag{2.23}$$

$$E_{ijkl}\frac{\partial u_k^1}{\partial y_l}n_j = E_{ijkl}\frac{\partial u_k^0(x)}{\partial x_l}n_j + p_i, \text{ on } S \tag{2.24}$$

On the other hand, if $v = v(x)$ is chosen for (2.20), then it will yield

$$\int_\Omega\left(\frac{1}{|Y|}\int_S p_i v_i dS\right)d\Omega = 0 \tag{2.25}$$

which implies that

$$\int_S p_i v_i dS = 0 \tag{2.26}$$

This means that the tractions within the holes are in self-equilibrium and it restricts the possibility of applied traction occuring in the hole [24].

Solving for the ε term:

Introducing (2.19) into (2.15), and choosing $v = v(x)$, yields

$$\begin{aligned}&\int_\Omega \left[\frac{1}{|Y|}\int_\Lambda E_{ijkl}\left(\frac{\partial u_k^0}{\partial x_l}+\frac{\partial u_k^1}{\partial y_l}\right)dY\right]\frac{\partial v_i(x)}{\partial x_j}d\Omega \\ &= \int_\Omega\left(\frac{1}{|Y|}\int_\Lambda f_i dY\right)v_i(x)d\Omega + \int_\Gamma t_i v_i(x)d\Gamma\end{aligned} \tag{2.27}$$

This is the statement of overall equilibrium in the macroscopic scale.

Choosing $v = v(y)$, yields

$$\int_\Omega \left[\frac{1}{|Y|}\int_\Lambda E_{ijkl}\left(\frac{\partial u_k^1}{\partial x_l}+\frac{\partial u_k^2}{\partial y_l}\right)dY\right]\frac{\partial v_i(y)}{\partial x_j}d\Omega = \int_\Omega\left(\frac{1}{|Y|}\int_\Lambda f_i dY\right)v_i(y)d\Omega \tag{2.28}$$

which is equivalent to

$$\int_\Lambda E_{ijkl}\left(\frac{\partial u_k^1}{\partial x_l}+\frac{\partial u_k^2}{\partial y_l}\right)\frac{\partial v_i(y)}{\partial x_j}dY = \int_\Lambda f_i v_i(y)dY \tag{2.29}$$

The above equation represents the unit cell at the microscopic level, which is known as the 'unit cell problem'.

In order to obtain the homogenised equation that describes the macroscopic equilibrium with the similar trend as equation (2.8), the following two problems are considered:

$$\int_\Lambda E_{ijpq}\frac{\partial \chi_p^{kl}}{\partial y_q}\frac{\partial v_i(y)}{\partial y_j}dY = \int_\Lambda E_{ijkl}\frac{\partial v_i(y)}{\partial y_j}dY \tag{2.30}$$

$$\int_\Lambda E_{ijkl}\frac{\partial \psi_k}{\partial y_l}\frac{\partial v_i(y)}{\partial y_j}dY = \int_S p_i v_i(y)dY \tag{2.31}$$

where χ^{kl} is the first order correction factor [51], also known as the microscopic characterisation displacement field [27], it plays the role of a parameter.

The existence of solution of the above two problems is assured by the aforementioned proposition. Therefore, from equation (2.22), the second term of the **u** expansion, u^1 can be written as:

$$u_i^1 = -\chi_i^{kl}(x,y)\frac{\partial u_k^0(x)}{\partial x_l} - \psi_i(x,y) + \tilde{u}_i^1(x) \tag{2.32}$$

where $\tilde{u}_i^1$ is an arbitrary constant of integration in y.

Introducing equation (2.32) into (2.27) yields

$$\int_\Omega \left[\frac{1}{|Y|} \int_\Lambda \left(E_{ijkl} - E_{ijpq} \frac{\partial \chi_p^{kl}}{\partial y_q} \right) dY \right] \frac{\partial u_k^0}{\partial x_l} \frac{\partial v_i(x)}{\partial x_j} d\Omega$$
$$= \int_\Omega \left(\frac{1}{|Y|} \int_\Lambda \left(E_{ijkl} \frac{\partial \psi_k}{\partial y_l} \right) \frac{\partial v_i(x)}{\partial x_j} dY \right) d\Omega + \int_\Omega \left(\frac{1}{|Y|} \int_\Lambda f_i dY \right) v_i(x) d\Omega \quad (2.33)$$
$$+ \int_{\Gamma_t} t_i v_i(x) d\Gamma$$

Now, defining each term in the above equation:

$$E_{ijkl}^H(x) = \frac{1}{|Y|} \int_\Lambda \left(E_{ijkl} - E_{ijpq} \frac{\partial \chi_p^{kl}}{\partial y_q} \right) dY \quad (2.34)$$

$$\tau_{ij}(x) = \int_\Lambda \left(E_{ijkl} \frac{\partial \psi_k}{\partial y_l} \right) dY \quad (2.35)$$

$$b_i(x) = \frac{1}{|Y|} \int_\Lambda f_i dY \quad (2.36)$$

each representing the homogenised elastic constants, average 'residual' stress within the unit cell due to the traction p inside the holes and b_i represents for the average body force. The above equations simplify (2.33) to become:

$$\int_\Omega E_{ijkl}^H \frac{\partial u_k^0}{\partial x_l} \frac{\partial v_i(x)}{\partial x_j} d\Omega$$
$$= \int_\Omega \tau_{ij}(x) \frac{\partial v_i(x)}{\partial x_j} d\Omega + \int_\Omega b_i(x) v_i(x) d\Omega + \int_{\Gamma_t} t_i v_i(x) d\Gamma \quad (2.37)$$

This is very similar to equation (2.8), and it represents the macroscopic equilibrium.

The following procedure is to solve for the elasticity problem stated in (2.8):

1. Find χ and ψ within the unit cell by solving the equation (2.30) and (2.31).
2. Find E_{ijkl}^H, τ_{ij} and b_i using equation (2.34) - (2.36).
3. Construct equation (2.37) in the macroscopic domain.

As an example of the implementation of the homogenisation method, equation (2.6) can be determined at each point of the domain, by using the expansion of $\mathbf{u}^\varepsilon$:

$$\sigma_{ij}^\varepsilon = \sigma_{ij}^0 + \varepsilon \sigma_{ij}^1 + \varepsilon^2 (\cdots) \quad (2.38)$$

where

$$\sigma_{ij}^0 = E_{ijkl} \left(\frac{\partial u_k^0}{\partial x_l} + \frac{\partial u_k^1}{\partial y_l} \right) \quad (2.39)$$

$$\sigma_{ij}^1 = E_{ijkl} \left(\frac{\partial u_k^1}{\partial x_l} + \frac{\partial u_k^2}{\partial y_l} \right) \quad (2.40)$$

...

Introducing (2.32) into (2.39) will lead to the approximation of the macroscopic stress as

$$\sigma_{ij}^{0}(x,y)=\left[E_{ijkl}(x,y)-E_{ijpm}(x,y)\frac{\partial\chi_{p}^{kl}(x,y)}{\partial y_{m}}\right]\frac{\partial u_{k}^{0}(x)}{\partial x_{l}}-E_{ijkl}(x,y)\frac{\partial\psi_{k}(x,y)}{\partial x_{l}}$$

A similar procedure also applies for solving equation (2.40), and obtaining the periodic stress from equation (2.38).

To solve for such complicated composite domains using the homogenisation method, a standard FE method is used to describe the periodic displacement function, by introducing a shape function **N**.

$$\mathbf{u}=\sum_{i=1}^{\text{total no. of nodes}}\mathbf{N}_{i}\hat{\mathbf{u}}_{i}=\mathbf{N}\hat{\mathbf{u}} \qquad \textbf{(2.41)}$$

where $\hat{\mathbf{u}}$ is the discrete displacement. Substituting equation (2.41) into (2.6) and (2.7), they become:

$$\begin{aligned}\mathbf{e}&=\mathbf{B}\cdot\hat{\mathbf{u}}\\ \boldsymbol{\sigma}&=\mathbf{C}\cdot\mathbf{e}=\mathbf{C}\cdot\mathbf{B}\cdot\hat{\mathbf{u}}\end{aligned} \qquad \textbf{(2.42)}$$

where **B** is the strain operator and **C** the material constitutive matrix (previously represented as *E*)

The FE approximation of the unit cell problem is obtained by introducing the above discretised equations ((2.41) - (2.42)) into (2.30):

$$\hat{\mathbf{v}}_{1}^{T}\int_{Y}\mathbf{B}^{T}\mathbf{C}\mathbf{B}dY\hat{\boldsymbol{\Phi}}^{J}=\hat{\mathbf{v}}_{1}^{T}\int_{Y}\mathbf{B}^{T}\mathbf{c}^{J}dY \qquad \textbf{(2.43)}$$

where vectors $\mathbf{c}^{J}$ are represented by: $\mathbf{c}^{J}=\begin{Bmatrix}C_{11}\\C_{21}\\0\end{Bmatrix},\begin{Bmatrix}C_{21}\\C_{22}\\0\end{Bmatrix},\begin{Bmatrix}0\\0\\C_{33}\end{Bmatrix}$.

The function $\hat{\boldsymbol{\Phi}}$ represents the discretised displacement function (previously known as χ). As $\hat{\mathbf{v}}^{T}$ is an arbitrary test function, it can be removed from both sides.

The above equations resemble the discrete Hooke's Law relationship:

$$\mathbf{K}\hat{\boldsymbol{\Phi}}^{J}=\mathbf{F}^{J},j=1,2,3... \qquad \textbf{(2.44)}$$

where $\begin{cases}\mathbf{K}=\sum_{m=1}^{\text{total no. of element}}\int_{\hat{Y}}\mathbf{B}^{T}\mathbf{C}\mathbf{B}dY\\ \mathbf{F}^{J}=\sum_{m=1}^{\text{total no. of element}}\int_{\hat{Y}}\mathbf{B}^{T}\mathbf{c}^{J}dY\end{cases}$, $\hat{Y}$ is volume of meshed element.

The Hooke's relationship (2.44) is solved three times for a two-dimensional problem (six times for a three-dimensional problem) to determine the discrete displacement $\hat{\boldsymbol{\Phi}}$ with suitable boundary conditions to represent the periodicity of a unit cell. The functional $\hat{\boldsymbol{\Phi}}$ can then be used to evaluate the homogenised elastic constant through the following relation:

$$\mathbf{C}^H = \frac{1}{|\mathrm{Y}|} \sum_{m=1}^{\text{total no. of element}} \int_{\hat{Y}} \left(\mathbf{C} - \mathbf{CB}\hat{\mathbf{\Phi}}\right) dY \tag{2.45}$$

The microscopic properties such as displacement, stress and strain, can then be evaluated as:

$$\begin{aligned} \mathbf{u}^{\varepsilon} &= \mathbf{u}^0 - \varepsilon\hat{\mathbf{\Phi}}\mathbf{e}^0 \\ \mathbf{e}^{\varepsilon} &= \left(\mathbf{I} - \mathbf{B}\hat{\mathbf{\Phi}}\right)\mathbf{e}^0 \\ \boldsymbol{\sigma}^{\varepsilon} &= \left(\mathbf{C} - \mathbf{CB}\hat{\mathbf{\Phi}}\right)\mathbf{e}^0 \end{aligned} \tag{2.46}$$

where $\mathbf{e}^0$ is the initial strain; $\mathbf{e}^0 = \begin{Bmatrix} 1 \\ 1 \\ 0 \end{Bmatrix}, \begin{Bmatrix} 1 \\ 1 \\ 0 \end{Bmatrix}, \begin{Bmatrix} 0 \\ 0 \\ 1 \end{Bmatrix}$.

The FE modelling performed in this work is based on a two-dimensional analysis of a special orthotropic material, which leads to the calculation of four elastic constants.

The numerical homogenisation code developed in this work consists of three main sections. The pre-processor develops the geometrical layout of the unit cell model and constructs the Hooke's relationship to be used to solve for the characteristic displacement $\hat{\mathbf{\Phi}}$ and elastic constants $\mathbf{C}^H$. This section involves setting up the key-point coordinates, nodes and element by meshing the numerical unit cell model. The solver computes the structural responses of the model subjected to unit strain loading, which is done using static analysis inside the FE package. The post-processor then evaluates the stress-strain distribution at element level, by harvesting the numerically computed results, and computes the elastic constants $\mathbf{C}^H$.

ii. Metamodelling

Much of today's engineering analysis is involved in complex computer studies, supplying a vector of design variables / inputs (x) and computing a vector of responses / outputs (y). While computer power is cheap nowadays, a single run of such analysis may take from minutes to hours, it is still unrealistic to rely the complex engineering analysis on many FE / experimental analyses for which every minor adjustment is made. Moreover, this query-response procedure is a trial-and-error technique where the user may never get to know what is the best input corresponding to the desire response.

Many researches have addressed this concern and the basic idea is to develop an approximation of the input-output (or query-response) relationship that is more efficient to run, and gain more insight into the relationship. Therefore the role of learning from the data has become important.

There are two stages in the operation of a learning system / metamodel / approximation model / surrogate model, first learning from data / training the metamodel, followed by predicting for future test samples. It is important to note that the training and testing data must originate from the same underlying statistical distribution. In other words, the training and testing data must be obtained from the same source, though most of the time, the underlying statistical distribution is unknown. Therefore the training data from the query (x) – response (y) procedure, which is used to construct the metamodel, is independent and identically distributed according to the joint probability density function:

$$p(\mathbf{x}, y) = p(\mathbf{x})p(y \mid \mathbf{x}) \tag{2.47}$$

where $p(\mathbf{x})$ is the fixed density for a given set of inputs $\mathbf{x}$, and $p(y|\mathbf{x})$ is the fixed conditional probability density for the corresponding output y.

There are many specific learning tasks; they include classification (separating data into different classes), regression (estimating the input-output relationship), probability density estimation and others. Classical approaches to a learning problem depend on a strict assumption on the fixed number of parameters allowed in a function. The development of the non-parametric models is mainly to overcome the shortage of classical models: having to specify the parametric form of the unknown distributions and dependencies.

Let us assume that the mathematical model explaining the input-output relationship of a simulation is as follows:

$$y = f(\mathbf{x}) + \varepsilon_{\text{measument}} \tag{2.48}$$

where y is the output of the simulation, e.g. the structural property of a honeycomb; and $\mathbf{x}$ is the input, e.g. the length, width and height of the honeycomb, its core material properties etc, and $\varepsilon_{measument}$ is the error obtain during measurement. For simplicity, the derivations below will be based on Multiple-Input-Single-Output (MISO) systems, and the estimation of the metamodel of the actual system can be represented as:

$$y = g(\mathbf{x}) + e = \hat{y} + e_{total} \tag{2.49}$$

where e_{total} is the error of approximation (ε_{approx}) and measurement ($\varepsilon_{measument}$), the approximation error comes from the regression of actual data into metamodels, and the measurement error comes from experiments. The $g(\mathbf{x})$ is the estimated underlying deterministic function of the system; it can be written as $g(\mathbf{x},w)$, where w is the parameter vector of the approximating function.

The measurement of mismatch is quantified by the loss function, $L(y,g(\mathbf{x},w))$, it is usually nonnegative value [15], so a large loss value from the loss function indicates poor approximation. The most commonly used loss function for regression is the squared error (L_2):

$$L\big(y, g(\mathbf{x}, w)\big) = \big(y - g(\mathbf{x}, w)\big)^2 \tag{2.50}$$

In order to obtain the best approximation, assuming that the error is purely random with zero mean and constant variance, the expectation of the loss function must be minimised using the training data, which is also called the risk functional:

$$R(w) = \int \big(y - g(\mathbf{x}, w)\big)^2 \, p(\mathbf{x}, y) \, d\mathbf{x} dy \tag{2.51}$$

The above equation measures the accuracy of the metamodel's predictions of the system's response. It can be explained as measuring the accuracy of approximating the actual function $f(\mathbf{x})$:

$$\begin{aligned} R(w) &= \int \big(y - f(\mathbf{x}) + f(\mathbf{x}) - g(\mathbf{x}, w)\big)^2 \, p(\mathbf{x}, y) \, d\mathbf{x} dy \\ &= \int \big(y - f(\mathbf{x})\big)^2 p(\mathbf{x}, y) \, d\mathbf{x} dy + \int \big(g(\mathbf{x}, w) - f(\mathbf{x})\big)^2 p(\mathbf{x}, y) \, d\mathbf{x} \\ &\quad + 2 \int \big(y - f(\mathbf{x})\big)\big(f(\mathbf{x}) - g(\mathbf{x}, w)\big) p(\mathbf{x}, y) \, d\mathbf{x} dy \end{aligned} \tag{2.52}$$

With error of zero means, the last term on the right hand side in the equation above becomes zero. The first term of the equation does not depend on $g(\mathbf{x},w)$ and by utilising equation (2.48), it can be written as the variance of the noise:

$$\begin{aligned} &\int \big(y - f(\mathbf{x})\big)^2 \, p(\mathbf{x}, y) \, d\mathbf{x} dy \\ &= \int \varepsilon_{measurement}{}^2 \, p(y \mid \mathbf{x}) \, p(\mathbf{x}) \, d\mathbf{x} dy \\ &= \int \Big[\int \varepsilon_{measurement}{}^2 \, p(y \mid \mathbf{x}) \, d\mathbf{x} \Big] p(\mathbf{x}) \, dy \\ &= \int E_e \big(\varepsilon_{measurement}{}^2 \mid \mathbf{x} \big) \, p(\mathbf{x}) \, d\mathbf{x} \end{aligned} \tag{2.53}$$

which simplifies equation (2.52) to become:

$$R(w) = \int E_e\left(\varepsilon_{measurement}{}^2 \mid \mathbf{x}\right) p(\mathbf{x}) d\mathbf{x} + \int \left(g(\mathbf{x}, w) - f(\mathbf{x})\right)^2 p(\mathbf{x}, y) d\mathbf{x} \tag{2.54}$$

Therefore the risk for a regression model, with L_2 loss and zero mean is composed of the noise variance and the accuracy of approximating the function.

As the noise variance does not depend on w, the minimisation of equation (2.54) relies on the second term on the right hand side, which measures the difference between the unknown actual function $f(\mathbf{x})$ and the approximation function $g(\mathbf{x})$.

The second stage of learning is to use the metamodel to predict future data. As the accuracies of the metamodel only measures the training data, it is important that the metamodel can generalise well for other data that is from the same source, which was not presented during the training procedure.

The mean square error (MSE) used in this work to measure the error of the approximation between $f(\mathbf{x})$ and $g(\mathbf{x},w)$ can be written as [15]:

$$\begin{aligned} E_n\left[\left(g(\mathbf{x}, w) - f(\mathbf{x})\right)^2\right] &= E_n\left[\left(g(\mathbf{x}, w) - E_n\left[g(\mathbf{x}, w)\right]\right)^2\right] \\ &\quad + \left(g(\mathbf{x}) - E_n\left[g(\mathbf{x}, w)\right]\right)^2 \end{aligned} \tag{2.55}$$

where $E[\ ]$ represents the expected value of an average over all the training samples of size n. The first term on the right hand side of the above equation represents the variance and is followed by the bias squared, i.e. $\text{mse}(g(\mathbf{x}, w)) = \text{bias}^2(g(\mathbf{x}, w) + \text{var}(g(\mathbf{x}, w))$. When the estimated function is applied on another dataset, the variance indicates how much the approximation function varies from the training data to the new dataset and bias measures the mismatch between the two datasets.

A model with high variance and low bias will give very accurate mapping of the training dataset, but fail to generalise for other data from the same source, this is known as overfitting. On the other hand, a model with low variance and high bias will not be able to fit the data at all, as it only bias towards one value among all values within the dataset, this is known as underfitting. Therefore it is important to find the middle ground between these two. There are several methods to avoid over- / under-fitting of the metamodel, which rely on methods of quantifying the complexity of the metamodels, such as penalisation, early stopping rules, structural risk minimisation, Bayesian inference and many others [15]. They all basically embrace the following concept: 'if two metamodels with equal performance is presented, the one with least complexity will be chosen'. The most widely used method is so far the penalisation method, where a penalisation term is added into the empirical risk to be minimised:

$$R_{pen}(w) = R_{emp}(w) + \lambda\phi\left[g(\mathbf{x}, w)\right] \quad (2.56)$$

where R_{emp} denotes the usual empirical risk (training error) and the penalty $\phi[\ ]$ is a nonnegative functional associated with each possible estimate of $g(\mathbf{x}, w)$, $\lambda > 0$ is a parameter which defines how severely the empirical risk will be penalised by its complexity. As the prior knowledge of the true system is incorporated in the penalty function and it is independent from the training data, it follows that a highly flexible model (in predicting future data) will have penalties close to zero. The penalty function can be strictly defined, such as by ridge penalty and subset selection [15], others may be defined implicitly when choosing the metamodels, such as a linear combination of the basis functions.

The optimal value for λ is usually chosen using a resampling method. The resampling method makes no assumptions on the statistical information of the data or the type of target function that is being estimated. The basic idea is to separate the dataset randomly into learning / training set and validation set. The metamodel is constructed based on the training set, and evaluated using the validation set. There are various strategies on how to partition the data when using the resampling method. The easiest one takes a continuous chunk of data as the training data (say 70% of the whole dataset), while the remaining ones will become the validation data. In the case where the data is expensive to obtain and it is too small to split, an alternative way is to estimate all possible partitionings and average these estimates, known as cross-validation [15]. However, this alternative way requires extremely expensive computational power, therefore more practical solutions were suggested, such as the k-fold cross validation and leave-one-out-cross-validation. The k-fold method is to randomly divide the data into k disjoint subsamples (roughly equal size), take out one of the subsamples and train the metamodel using the remaining samples. The process is repeated for a number of times, and the risks of all the validation subsamples are averaged. The leave-one-out is a special case of k-fold strategy, where every data has to be isolated from the others as the validation set. A careful separation of data is important, as it determines the penalty weight imposed on the empirical risk (training error). As the complexity of the metamodel increases, the empirical risk decreases, while the second term of equation (2.56) – the true risk, which is indicated by the validation set, increases (Figure 2.4).

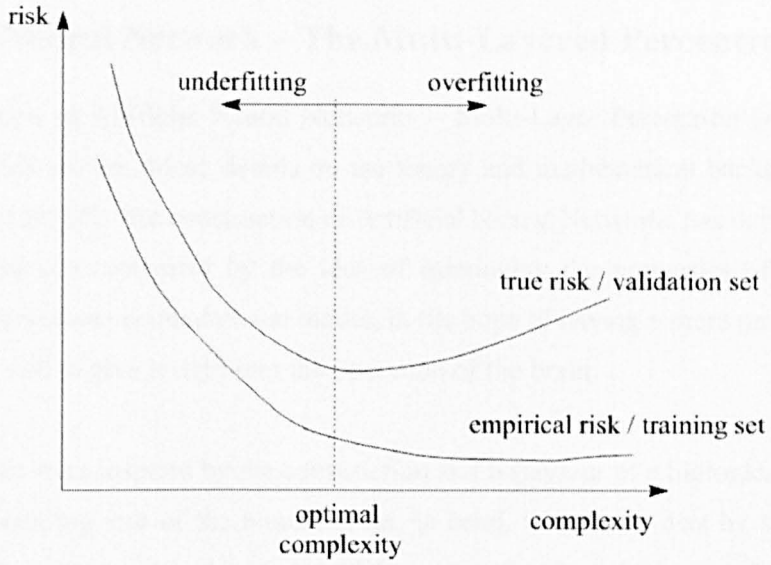


Figure 2.4 Risk versus complexity plot for resampling strategy.

The process of developing a metamodel involves learning from data and implementation of the metamodel in predicting the future data, whereby the generalisation of the metamodel is important. The regularisation / penalisation method is commonly used for choosing which metamodel is generalising well (low risk) without being too complex. The implementation of regularisation becomes difficult when used in flexible metamodels such as Neural Networks (NN). The difficulties involve nonlinear optimisation of the equation (2.56) and the method of resampling. The optimal solution of the nonlinear optimisation problem depends on the initial parameters, which were often initiated randomly, followed by various validation procedures.

iii. A Neural Network – The Multi-Layered Perceptron (MLP)

A brief discussion of Artificial Neural Networks – Multi-Layer Perceptron (ANN – MLP) is carried out in this section. More details on the theory and mathematical background of MLPs are available in [28, 80]. The construction of Artificial Neural Networks has developed since the mid 1940's, and was motivated by the idea of mimicking the properties of a human brain through a theoretical and computational model, in the hope of having a more powerful paradigm for a computer, and to give insight into the operation of the brain.

The ANN studies were inspired by the construction and behaviour of a biological neuron, which is the basic processing unit of the human brain. In brief, the neuron acts by summing stimuli from connected neurons and each element of the summation is weighted by the strength of the connection. If the accumulation of stimulus or activation exceeds a threshold, the neuron 'fires'. In another words, it generates a stimulus, which is passed on into the nerve network system. If the activation is less than the threshold, the neuron remains dormant.

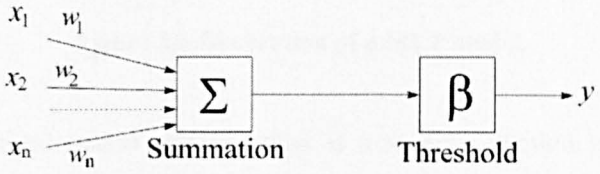


Figure 2.5 The McCulloch-Pitts model.

A mathematical model of a single neuron was developed by McCulloch and Pitts in 1943 [21]. The McCulloch-Pitts (MCP) model (Figure 2.5) constitutes the simplest neural network model. However, the simple single neuron MCP model is insufficient to represent some simple functions, and this led to the construction of a network of MCP neurons. The study of such networks, called perceptrons, began in 1962 [80].

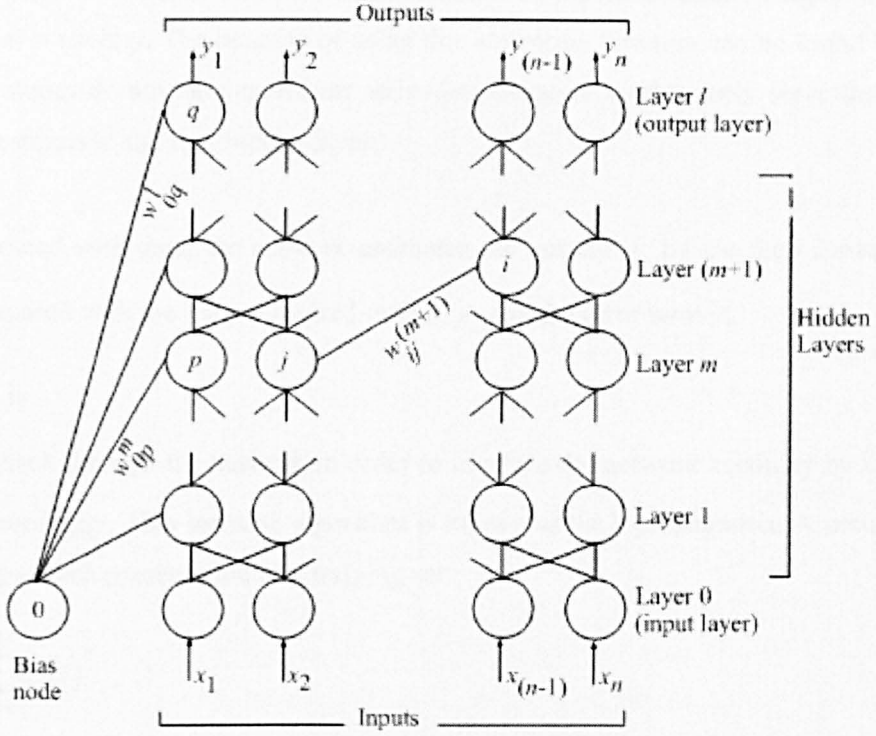


Figure 2.6 Illustration of a MLP model.

The most commonly used neural network now is a modern version of the perceptron - the Multi-Layer Perceptron (MLP) trained with back propagation. The basic back propagation algorithm consists of two passes through the network layers. In the feed forward process, input signals are fed into the input layer nodes, and propagated through the hidden layer nodes, producing an output from the output layer nodes. As seen in Figure 2.6, each node j in layer m is connected to each node i in layer $(m+1)$ by a connection of weight $w_{ij}^{(m+1)}$. For a network with $(l+1)$ layers, layer 0 represents the input layer and layer l is the output layer. The input signal passes through each node in layer $(m+1)$ as follows: a weighted sum is performed at node i of all outputs $x_j^{(m)}$ from layer m, and gives an excitation $z_i^{(m+1)}$ of node i,

$$z_i^{(m+1)} = \sum_{j=0}^{n^{(m)}} w_{ij}^{(m+1)} x_j^{(m)} \tag{2.57}$$

where $n^{(m)}$ is the number of nodes in layer m. The summation index starts at zero to accommodate the bias node, which allows constant offsets. This excitation signal is then passed through a nonlinear activation function to produce an output of the node i in the $(m+1)$ layer, described as

$$x_i^{(m+1)} = f\left(z_i^{(m+1)}\right) = f\left(\sum_{j=1}^{n^{(m)}} w_{ij}^{(m+1)} x_j^{(m)}\right) \tag{2.58}$$

The activation function used in this work is chosen to be the most common sigmoid, hyperbolic function, $f(x) = tanh(x)$. The benefits of using this activation function can be found in [28]. The input layer nodes do not have nonlinear activation functions as they only serve the purpose of distributing inputs to the first hidden layer.

When presented with data, the network estimates the outputs $\hat{y}_i$ by the feed forward process. This is compared with the known desired output y_i and the error term is:

$$\delta_i^{(l)} = y_i - \hat{y}_i \tag{2.59}$$

$\delta_i^{(l)}$ is fed back through the network in order to improve the network accuracy by updating the weights accordingly. This learning algorithm is known as back propagation. A network error J is evaluated at each presentation of a training set:

$$J(t) = \frac{1}{2}\sum_{j=1}^{n^{(l)}} \delta_i(t)^2 \tag{2.60}$$

where $J=J(\theta_1, \theta_2, ..., \theta_n)$ is a function of the network parameters, and the θ_i are the connection weights ordered in some ways. The integer t is the presentation order of the training set.

In the simplest case, a steepest descent algorithm is used to adjust the parameters,

$$\Delta\theta_i = -\eta\frac{\partial J}{\partial \theta_i} = -\eta\nabla_i J \tag{2.61}$$

η determines the step size and is known as the learning coefficient [80, 28]. The weight updates in the hidden layers can be computed by using the chain rule of partial differentiation:

$$w_{ij}^{(m)}(t) = w_{ij}^{(m)}(t-1) + \Delta w_{ij}^{(m)}(t) \tag{2.62}$$

where $\delta_i^{(m)}$ is the error in the output of the i^{th} node in layer m, and $\Delta w_{ij}^{(m)}(t) = \eta\delta_i^{(m)}(t)x_j^{(m-1)}(t)$. The error must be constructed from the known errors $\delta_i^{(l)} = y_i - \hat{y}_i$ at the output layer l, which explains the name 'back propagation'. The weight must be adjusted layer by layer, moving backwards from the output layer.

Little guidance on the learning coefficient can be found, it slows the convergence of parameters if it gets too small; while large values of the learning coefficient will lead to divergence or oscillation. Therefore a momentum term, α, is introduced into the update rule:

$$\Delta w_{ij}^{(m)}(t) = \eta\delta_i^{(m)}(t)x_j^{(m-1)}(t) + \alpha\Delta w_{ij}^{(m)}(t-1) \tag{2.63}$$

The effect of this additional term is to damp out the oscillations in the back-propagation error signal. Once the comparison error reaches an acceptable level over the whole training set, the training procedure ends, and the network is established.

iv. Genetic Programming

Genetic Programming (GP) shares the same concepts as the well-known Genetic Algorithm (GA) but increases the flexibility and complexity by allowing the structure of the solution to undergo adaptation. The structure is typically a hierarchical computer program or mathematical function of dynamically varying size and shape.

GP starts with an initial population of randomly generated individuals, which consists of function and terminal nodes appropriate to the problem domain. The appropriateness of the functions are less strict than the traditional metamodel, as it is just to decide if the function used would be arithmetical, logical etc., one can even use some user-defined functions that are suitable for the problem domain, examples of the function nodes are: '+', '-', '×' and '÷'. Therefore, depending on the problem domain, the GP population individuals may be real, complex, vector, symbolic, multiple valued etc. The terminal set, however, does not take in any argument; it is purely the input variables or randomly generated constants.

As mentioned earlier, each individual is built from repeatedly combining all possible functions and terminals:

$$F = \{f_1, f_2, \ldots, f_M\};\ T = \{a_1, a_2, \ldots, a_N\} \tag{2.64}$$

where F and T represent function sets and terminal sets respectively; M and N are the number of functions and terminals included in the GP. Each function f_i takes in a specific number of arguments – the number also known as the function's arity, while the terminals have null arity.

$$z(f_1, f_2, \ldots, f_M) = \mathbf{Z},\ z(a_1, a_2, \ldots, a_N) = 0 \tag{2.65}$$

There is however one rule to obey: each function must be applicable to any values returned by other functions and any values carried by the terminal nodes.

The creation of the initial random population is a blind search in the problem domain, whereby the birth of each individual is achieved by randomly generating a root node, and its subsequent branches. The root node must be chosen from the function set, as illustrated in Figure 2.7, a root node '+' has arity 2, with each argument being represented by a connection to a subsequent node. It is then randomly combined with other nodes from either the terminal or function set. If a terminal node was chosen, it will stop branching out; else the growth will continue. The individual created from Figure 2.7 represents an expression of (x*y) + z.

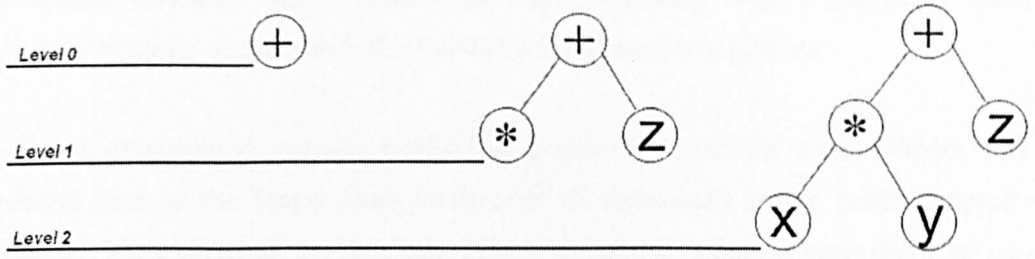


Figure 2.7 Creation of an individual in Genetic Programming (GP).

There are a few growth strategies, namely the full method, grow method and ramped half-and-half [41] The full method of generating the initial population involves creating individuals that grow up until the maximum allowable level. The grow method involves creating individuals that are variably shaped, but the depth can be of any level less than or equal to the maximum allowable depth. The ramped half-and-half is a mixture of both. The grow method is adopted in this work.

To prevent the individuals from growing infinitely, a limit is enforced on the depth of the tree structure and on the total number of nodes a tree can have, after which only terminal nodes are chosen. Some other bloat (i.e. rapid increase of individual tree size) control methods are described in [48, 68].

The driving force of GP, as for GA, is by computing the fitness of the individuals before 'mating' them to produce subsequent generations. Each individual in the population is measured in terms of its performance merit, known as the fitness measure. There are several ways of assigning a fitness measure based on different problem domains [41]. The fitness measure enables the best individuals to be chosen to inherit across the generations, and eliminates the unfit ones. Because the problem here is modelling, the fitness of an individual in this work is assigned by the inverse of its percentage Mean Square Error (MSE).

$$\frac{1}{Fitness} = MSE = \frac{100}{N\sigma_y^2}\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right)^2 \qquad \textbf{(2.66)}$$

where N is the number of data points, σ_y^2 is the variance of the desired output, y_i and $\hat{y}_i$ are the desired and estimated output respectively. Generally, a fitness of 10, in another words an MSE percentage of 0.1, is considered as an excellent fitness.

At each generation, genetic operators are applied to modify the individuals to create offspring. The genetic operators are crossover and mutation. The GP has to go through the population to select the 'good/fit' parents, based on their fitness measure, in order to perform the genetic operations. There are several selection methods, like the GA, this work adopted fitness

proportionate selection, others include Stochastic sampling with replacement, Stochastic sampling with partial replacement, Stochastic universal sampling [20] etc.

The fitness proportionate scheme works like a spin of a roulette wheel (Figure 2.8). The cumulative sum of the fitness (sum_total) over all individuals in the current population is determined. The individuals are then mapped one-to-one into adjacent intervals in the range [0, sum_total]. The size of each individual interval corresponds to its fitness value. A random number is generated in the range of [0, sum total] and the individual whose segment spans the random number is then selected. The process is repeated until the desired number of individuals has been selected [20].

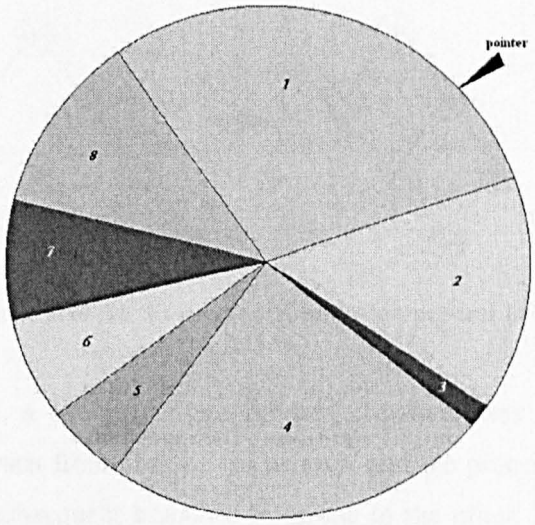


Figure 2.8 Roulette wheel diagram.

The crossover operation in GP brings variation into the population by producing offspring that inherit parts of each parent. As for the mutation, unlike the GA, the GP mutation operation is by randomly substituting a branch of an expression, and often gives more constructive modification [46] than the GA mutation.

There are different crossover and mutation strategies, such as two-parent-one-child, two-parent-two-children crossover, size fair crossover [10], branch mutation, node mutation etc. The genetic operators adopted for this work are two-parent-two-children crossover and a mixture of branch and node mutation.

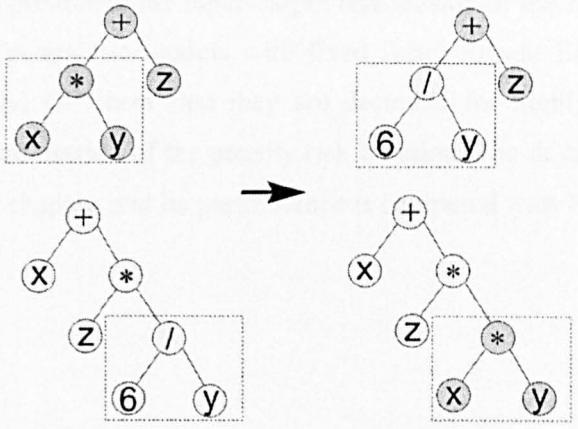


Figure 2.9 An illustration of crossover process in GP.

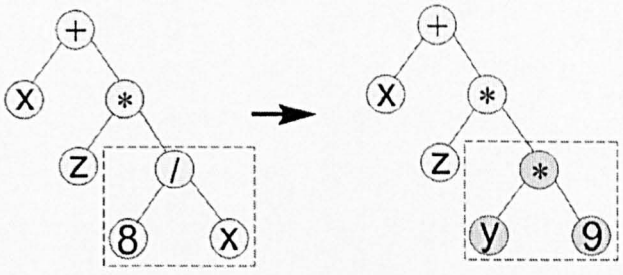


Figure 2.10 An illustration of mutation process in GP.

As shown in Figure 2.9, a two-parent-two-children crossover was chosen for this work. A random node will be chosen from each of the parents, and the process is just to swap the two chosen nodes and their subsequent branches from one to the other. In Figure 2.10, a branch mutation has taken place, whereby a random node is chosen from one parent, the node and its subsequent branches will be replaced or chopped or inserted with another random branch. A node mutation is far simpler; it is just to substitute the randomly chosen node. However, one must take great care of the compatibility of arity between the original and replacement nodes. The mixture of mutation strategies used in this work is namely branch replacement, branch insertion, branch chopping and node mutation, which will be discussed in the following chapter. For methods used by other GPs, one can refer to [21, 44].

So far this chapter has demonstrated the backbone of the homogenisation method and the discretised homogenisation method was used in FEA, the boundary condition (i.e. the unit displacement imposed on the unit cell). Unit cell geometry, meshing and calculation of the homogenised properties will be presented in later chapters. The FE macros for all the unit cells studied in this work are included in Appendix 6.

Though the FE saves the cost of doing experiments, it is still time consuming when it comes to engineering design optimisation. Therefore an approximation model / surrogate model /

metamodel is used to approximate the input-output relationship of the FEA. There are classical approaches, which are parametric models with fixed form. Recent finding on some flexible models, such as NN and GP show that they are desirable for highly complicated systems, despite the nonlinear optimisation of the penalty risk function. The development of the GP will be discussed in the next chapter, and its performance is compared with NN and polynomials.

3. Genetic Programming Development

i. GP configuration

There are many GP implementations available online for download, however these are mostly solution examples of some well known optimisation and search problems, such as the Travelling Salesman, Santa Fe Trail/ Artificial Ant Problem, Lawn Mower Problem etc [41]. A general purpose GP implementation is not as easily found as code for a GA. Though these solution examples come with source code, it is troublesome to alter these case-tailored GP programs. Therefore a decision here was made to develop the GP code from scratch.

Though every computer language (e.g. Pascal, C, FORTRAN, etc) is capable of constructing a GP program, most of the GP implementations found were written in LISP, C++ or Java. Within the GP algorithm, the structures that evolve are hierarchical computer programs based on a symbolic like representation of each function and terminal. Koza [41] chose LISP as it has a symbolic expression (S-expressions) for both computer programs / function (such as +, -, ×, ÷ etc) and data / terminals (such as x, y, z and randomly generated constants). This allows the GP trees to be strings of executables, and to be genetically manipulated separately from the main GP routine, while the code for the GP algorithm handles only the results from these GP trees (i.e. the fitness evaluated from each GP individual).

The hierarchical tree structures of each program that forms an individual in the GP algorithm is similar in structure and purpose to the parse tree that is commonly created when compiling or interpreting the program itself. It is not possible to access this parsing and executing process directly from within most programming languages. Thus, with most programming languages, it is not possible to automatically create code written in that same language, and then execute it directly. LISP is the exception to this rule – a LISP program can generate and execute LISP programs simply and easily – which is why LISP has long been a favourite of 'artificial intelligence' researchers. Other reasons for choosing LISP are listed by J. Koza in [41].

Unfortunately, LISP has become infrequently used over the years; however, the convenience that LISP provides can still be easily achieved by the more recent Object Orientated Programming (OOP) languages. OOP languages available nowadays include C++, Java, Smalltalk etc. The OOP languages create a design paradigm based around data objects that are usually modelled after real-world objects, simulating their state and behaviour. A software object maintains its state by saving it in one or more variables. A variable is an item of data named by an identifier. A software object implements its behaviour with methods, i.e.

subroutines associated with the object [74]. In the case of GP coding, the object is the function/terminal, such as +, -, ×, ÷, constant, *x*, *y*, *z* etc; the state of the object is the arity of the function, while the behaviour of the object is the execution of the functions with arity. For example, *x* and *z* (both are terminals) both maintain their state by carrying numerical / string / integer values, and their behaviour is the value / character; in the mean time, '+' (a function) maintains its state in *x* and *z*, and the execution of '+' behaviour is by 'adding *x* and *z*'. The OOP concepts improve the GP hierarchical structure handling, and allow a similar effectiveness of symbolic expression of GP individuals.

The Genetic Programming (GP) code developed in this work is written in the Java language for the convenience of future deployment on various computer platforms (i.e. platform independence). Though Java shares a lot of features in common with C++, Java is more robust and has better memory management. Other differences between Java and C++ can be found in [19]. Performance comparisons between Java, C++ and LISP can be found in [34].

The GP algorithm can be divided into a sequence of tasks (Figure 3.1).

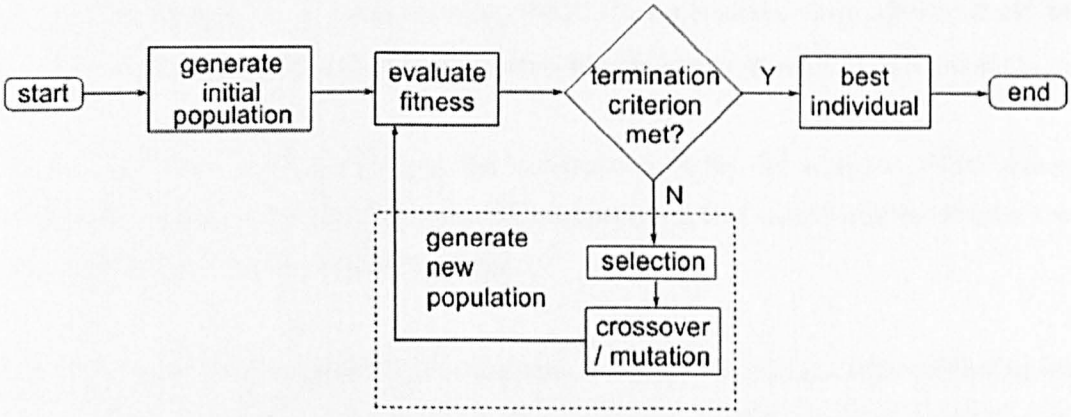


Figure 3.1 The GP flow chart.

The first generation of the population is created by randomly combining the function and terminal nodes. The individuals of the population are restricted to have a maximum of *n* levels of nodes and *m* number of nodes in total.

The GP code was first developed without penalising the larger individuals, and only contained simple crossover and mutation procedures, namely crossing two parent individuals to create two offspring, and randomly substituting a branch of an individual for the mutation.

In the process of crossover, both parents and offspring are assigned their fitness values, only the two fittest individual out of these four will remain for the following generation. Similarly, in the

mutation process, the fitness of the original individual will be compared to its mutated self; the less fit individual will be discarded.

The termination criterion is fixed by the total number of generations, and the best individual from the population of the final generation is printed out.

The initial function set only contained {+, -, ×, ÷, sin, cos, power, log} at that time, and the corresponding arity set was {2, 2, 2, 2, 1, 1, 2, 1}. The terminal set used to test the GP was chosen to represent the inputs for the analytical model of a hexagonal honeycomb with structural properties E_1 and ν_{12} [23], and randomly generated constant values. The results will be discussed in the section (iii).

It is important to note that the operating limits of some functions are different from other operating platforms. The '÷' used here is a protected division. It returns zero rather than infinity when the denominator is close to zero. For the case of 'power' and 'exponential', it returns zero if the base is zero, also when the power term was strictly set to positive values to avoid being used as a denominator in '÷', which creates 'NaN' (Not a Number) error, also to avoid having 0^0; 'log' is a natural log (base e), it returns zero when its input argument is close to zero.

'Infinity' and 'NaN' are prohibited in the construction of the GP code, as 'NaN' cannot be evaluated at successive levels of the tree; the computations that would lead to 'infinite' results such as division by zero are treated as errors.

In the later stages of development, the crossover and mutation process were refined to include more variety. The methods used were branch mutation, branch insertion, branch chopping, mutation at function nodes and mutation at the terminal nodes. The reasons for introducing various mutation strategies is that the GP is sensitive to a high mutation rate, and the original mutation method used in this work causes a drastic alteration of the individual tree [46].

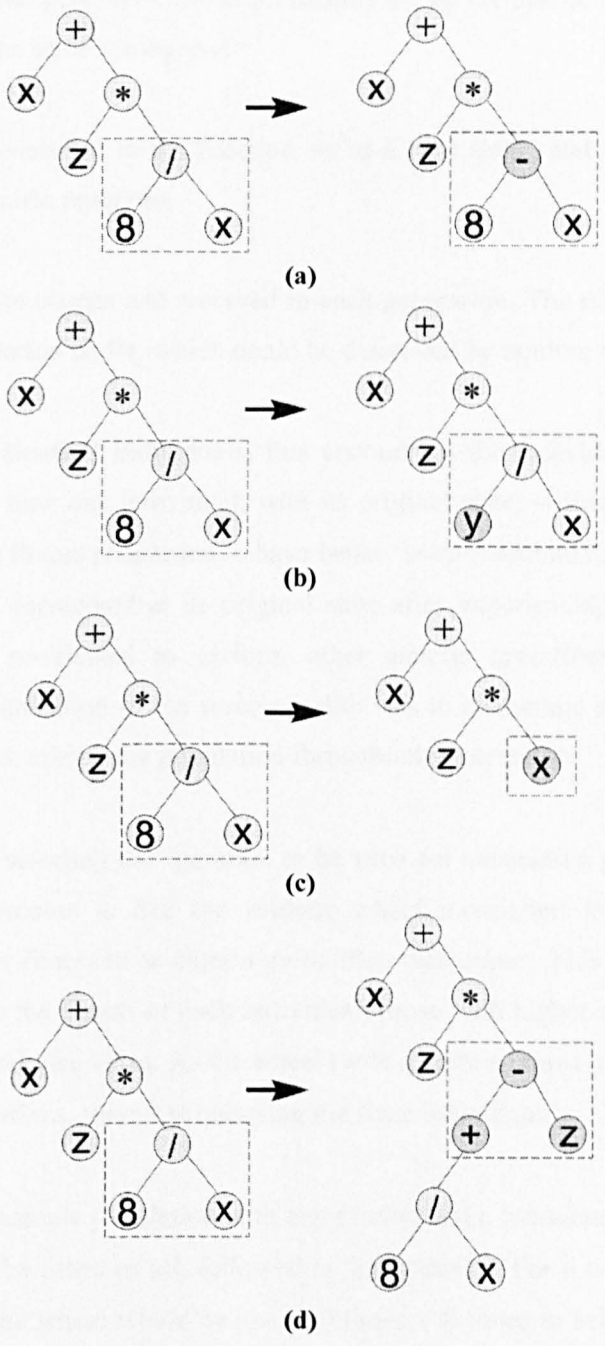


Figure 3.2 (a) mutation at the function node; (b) mutation at the terminal node; (c) branch chopping; (d) branch insertion.

The 'branch mutation' is the original mutation, which only replaces a branch of the individual tree; however, more than a branch can be mutated at a time, which will lead to a drastic mutation. 'Branch insertion' is to randomly create and insert a new branch into an individual tree; it does not chop off the original tree. 'Branch chopping', on the other hand, randomly chops off some branches from an individual tree, to prevent the individual tree from growing too much. 'Mutation at the function nodes' is replacing only the function node, and similarly, 'mutation at the terminal nodes' is replacing only the terminal node. Mutation of the function nodes must be carefully done, as the arity of each function is different. Therefore, only function nodes with the same arity can be used for replacement. Each of these mutation strategies takes a

certain percentage of the general mutation probability set by the user at the beginning of the GP process and remains the same throughout.

More functions were included in the function set at a later stage, and the function set can be customised to suit specific problems.

There is always an elite chosen and reserved in each generation. The elite is chosen to prevent the loss of the best solution so far, which could be destroyed by random crossover or mutation.

The GP also allows identical individuals, this encourages the individual to perform genetic operations with more than one individual, with its original state, within a generation. In other words, it enhances the fitness proportion to have better 'search' around the 'best solution so far'. The individual is not considered at its original state after experiencing crossover / mutation, although it may be re-selected to perform other genetic operations with other different individuals. The multiplication of the reserved elite was to encourage the stronger individuals and their offspring to dominate the population throughout generations.

The method used for selecting the 'parents' to be used for successive generations was fitness proportionate. This process is like the roulette wheel mentioned in Chapter 2, it allows individuals with higher fitness to be chosen more often than others. This is because the wheel is sectioned according to the fitness of each individual; those with higher fitness will get a bigger portion of the wheel and vice versa. As the wheel (with a pointer) turns, the pointer will be more likely to hit the big portions, therefore choosing the fitter individual.

Figure 2.8 shows an example population with eight individuals. Individual 3 has the least fitness, while individual 1 is the fittest of all, followed by individual 4. For a crossover of 4 times and mutation of 2 times, the wheel would be spun 10 times – 8 times to select pairs of parents for crossover, and 2 times for mutations. In each turn, individual 1 and 4 will more likely to be chosen than the others.

ii. GP Extrapolation capability

The extrapolation capability of the GP is studied using a static example. The data involved in this section are generated from the analytical formulation of a hexagonal honeycomb structural properties with respect to the geometrical parameters that determine the hexagonal shape.

The in plane structural moduli of hexagonal honeycombs were generated from the analytical model in [23].

$$E_1 = E_s\left(\frac{t}{l}\right)^3 \frac{\left(\frac{h}{l}+\sin\theta\right)}{\cos^3\theta} \frac{1}{\left[1+\left(2.4+1.5\upsilon_s+\tan^2\theta+\frac{2(h/l)}{\cos^2\theta}\right)\left(\frac{t}{l}\right)^2\right]} \tag{3.1}$$

$$\nu_{12} = \frac{\sin\theta\left(\frac{h}{l}+\sin\theta\right)}{\cos^2\theta} \frac{1+(1.4+1.5\nu_s)\left(\frac{t}{l}\right)^2}{1+\left(2.4+1.5\nu_s+\tan^2\theta+\frac{2(h/l)}{\cos^2\theta}\right)\left(\frac{t}{l}\right)^2} \tag{3.2}$$

$$E_2 = E_s\left(\frac{t}{l}\right)\frac{\alpha+\sin\theta}{\cos^3\theta} \cdot \frac{1}{\left[1+\left(2.4+1.5\nu_s+\tan^2\theta+\frac{2\alpha}{\cos^2\theta}\right)\left(\frac{t}{l}\right)^2\right]} \tag{3.3}$$

$$\nu_{21} = \frac{\sin\theta(\alpha+\sin\theta)}{\cos^2\theta} \cdot \frac{1+(1.4+1.5\nu_s)(t/l)^2}{1+\left(2.4+1.5\nu_s+\tan^2\theta+\frac{2\alpha}{\cos^2\theta}\right)\left(\frac{t}{l}\right)^2} \tag{3.4}$$

$$G_{12} = \frac{E_s(t/l)^2(\alpha+\sin\theta)}{\alpha^2\cos\theta} \cdot \frac{1}{F}$$

$$F = \left[1+2\alpha+\left(\frac{t}{l}\right)^2\left\{\begin{array}{l}\frac{(2.4+1.5\nu_s)(2+\alpha+\sin\theta)}{\alpha} \\ +\frac{\alpha+\sin\theta}{\alpha^2}\left[(\alpha+\sin\theta)\tan^2\theta+\sin\theta\right]\end{array}\right\}\right] \tag{3.5}$$

where,

$\alpha = \frac{h}{l} = [1,2,\ldots,9], \theta = [-30°,-25°,\ldots,60°], E_s = 1.85\times10^{10}Pa, \nu_s = 0.33, \frac{t}{l} = 0.01$, and subscripts 1 and 2 refer to directions x_1 and x_2, as shown in Figure 3.3.

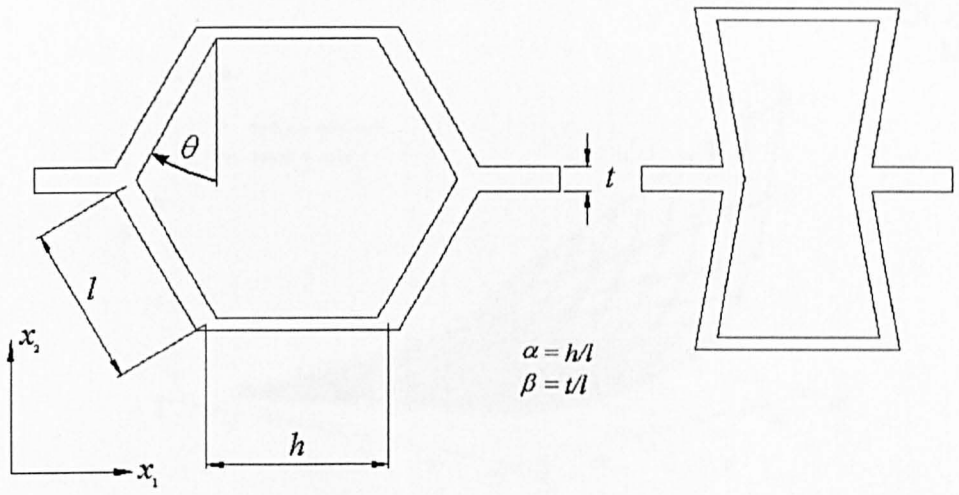


Figure 3.3 Hexagon honeycomb with various θ

Equations (3.1) to (3.5) can be generalised as a function of α and θ:

$$(E_1, \nu_{12}, E_2, \nu_{21}, G_{12}) = f(\alpha, \theta)$$

Meanwhile, the terminal set T and function set F for the GP can be defined as:

$$T = \{\theta, \alpha, \mathbf{R}\}$$

where **R** is the randomly generated real constants during the creation of the GP trees.

$$F = \{+, -, \times, \div, \sin, \cos, \text{power}, \log\}$$

with the associated arity set being {2,2,2,2,1,1,2,1} respectively.

The GP code was first tested using the E_1 data and ν_{12} data, with 10000 generations, 500 individuals used for E_1 and 10000 generations, 100 individuals for ν_{12}. The GP code at this stage only contained branch mutation and two-parent-two-children crossover. The maximum allowable nodes and maximum allowable levels were set as 50 and 10 respectively. As it was at the beginning of the GP development, the GP test run was performed based on the judgement of the researcher, in other words, the number of generations, population size, the crossover rate and the mutation rate were configured heuristically. In this case, the crossover and mutation rates were set as 0.6, 0.5 for E_1 and 0.5, 0.5 for ν_{12}.

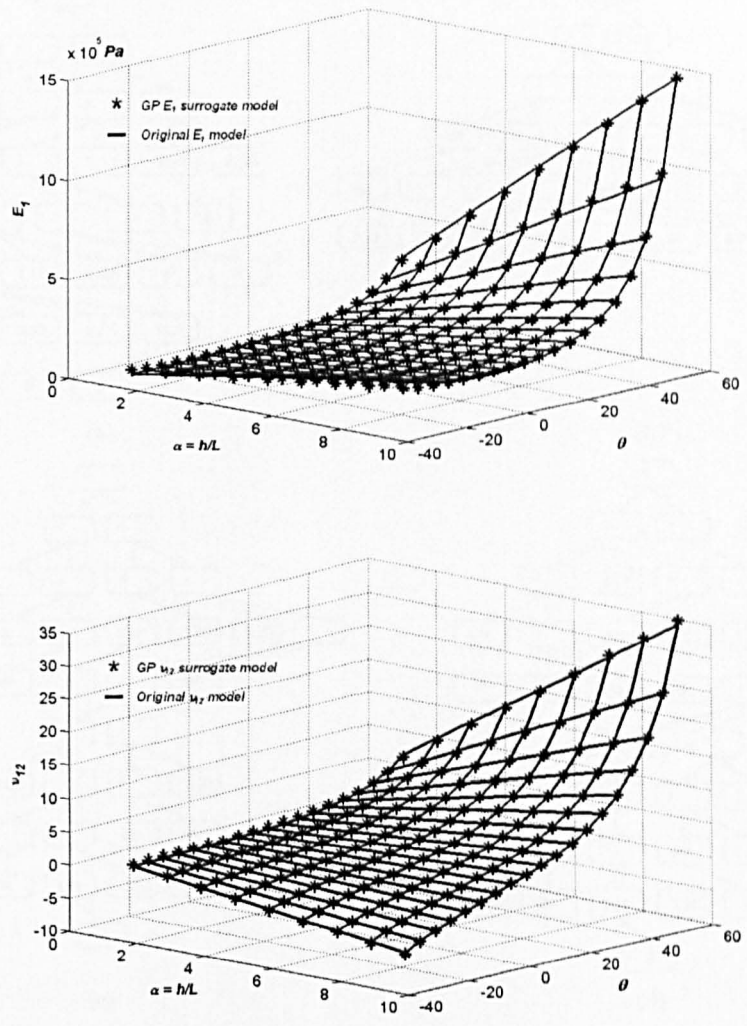


Figure 3.4 Comparison of fitness between GP model and equation (3.1) - (3.2).

The test of the GP code proved to be successful. Figure 3.4 shows the fitness of the GP model compared to the analytical equations (3.1) and (3.2). The metamodels gave a fitness of 6.95 and 17.18 respectively, which corresponds to mean square error of 0.1439 and 0.0582 respectively. Figure 3.5 illustrates the tree representation of the analytical solution of (3.1) and (3.2) in comparison with the GP fitted model; it shows that for data within a predefined range, the GP models are more compact than the analytical ones.

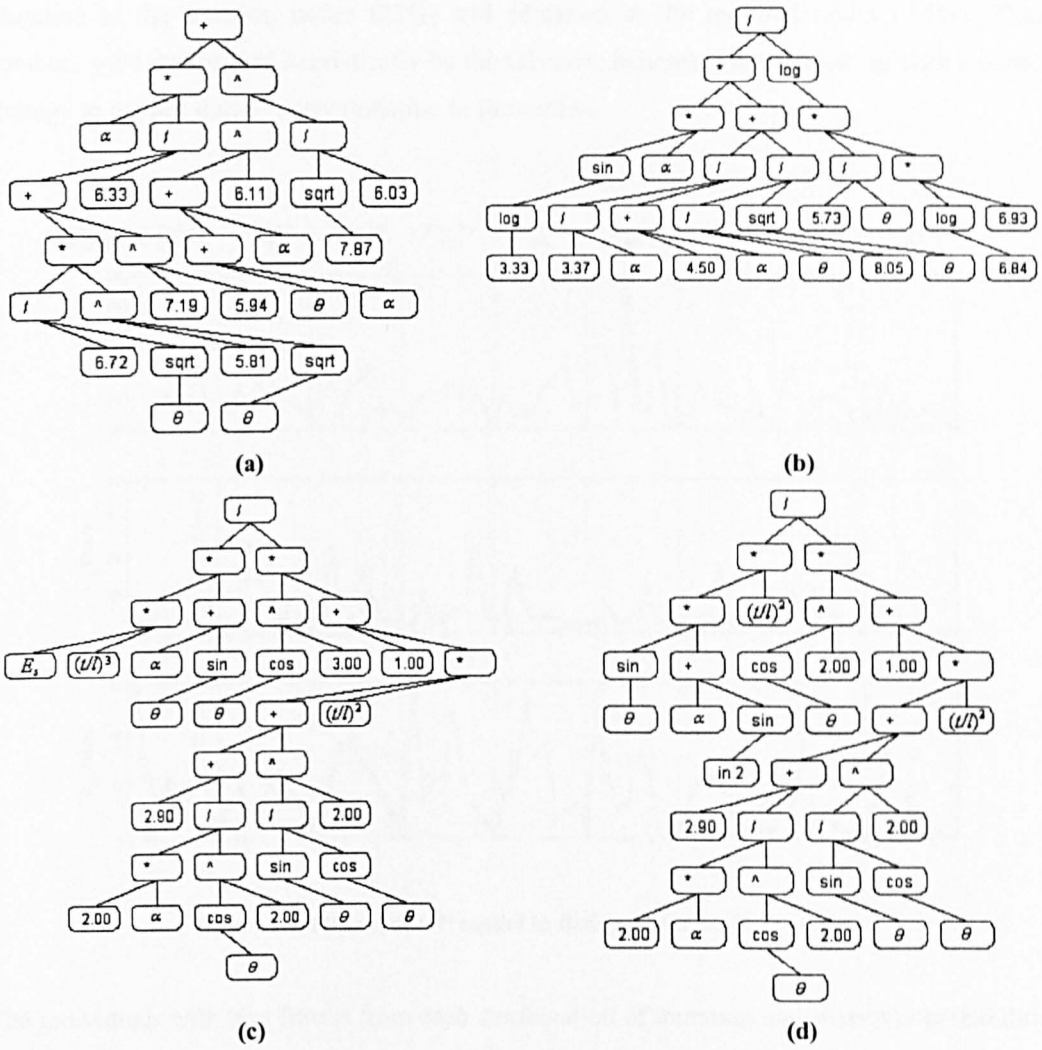


Figure 3.5 (a) GP tree representation of (3.1). (b) GP tree representation of (3.2). (c) Original (3.1) represented in tree structure. (d) Original (3.2) represented in tree structure.

As the GP development has become more mature, subsequent data (E_2, ν_{21}, and G_{12}) were fitted by training with only 2000 generations and 300 individuals, while the crossover and mutation rates were determined by sweeping through all the combinations between 0.1 to 0.9, with each combination being repeated 10 times using 10 different initial conditions, i.e. different random number seeds, for a better generalisation. The mutation method for GP at this stage was modified to contain a greater variety of approaches, which was mentioned in the previous section. Consequently E_2, ν_{21}, and G_{12} had not only the branch mutation, but also branch insertion, branch chopping, function node mutation and terminal node mutation. As mentioned in the previous section, each of these mutation strategies is chosen randomly using a certain percentage of the general mutation probability set by the user at the beginning of GP process, and this percentage remains the same throughout. The percentage set for each of these mutation strategies was: Branch mutation (38.5%), Branch insertion (23.1%), Branch chopping (7.7%),

Mutation at the function nodes (23%) and Mutation at the terminal nodes (7.7%). These numbers were determined heuristically by the GP user, indicating the chances of each mutation strategy to happen during every mutation in generation.

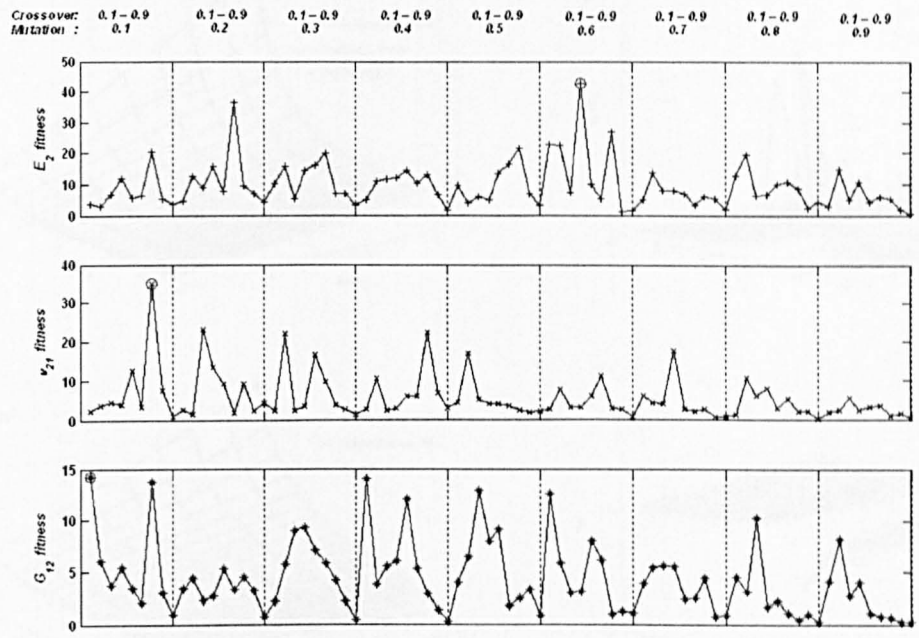


Figure 3.6 Fitness of GP model in fitting equation (3.3) - (3.5).

The individuals with best fitness from each combination of mutation and crossover probabilities were selected and compared with one another. Figure 3.6 shows the best testing fitness of all combinations for E_2, v_{21}, and G_{12}. The GP-created surrogate models for each of these datasets was determined by the highest fitness model, marked by circle in Figure 3.6 and their configurations are listed in Table 3.1.

	Crossover Rate	Mutation Rate	Fitness (%)
E_2	0.4	0.6	42.72
v_{21}	0.7	0.1	35.02
G_{12}	0.1	0.1	14.20

Table 3.1 Fitness of GP model and its genetical operator configurations in fitting equation (3.3) - (3.5).

The output of these GP models compared with the empirical equations (3.1)-(3.5) is illustrated in Figure 3.7. The shaded areas are the area where the hexagonal honeycombs exhibit auxetic behaviour.

Though it seems that there is good agreement between the GP and equations (3.1)-(3.5), the GP surrogate model cannot be used to construct the honeycomb constituent matrix for reasons described below.

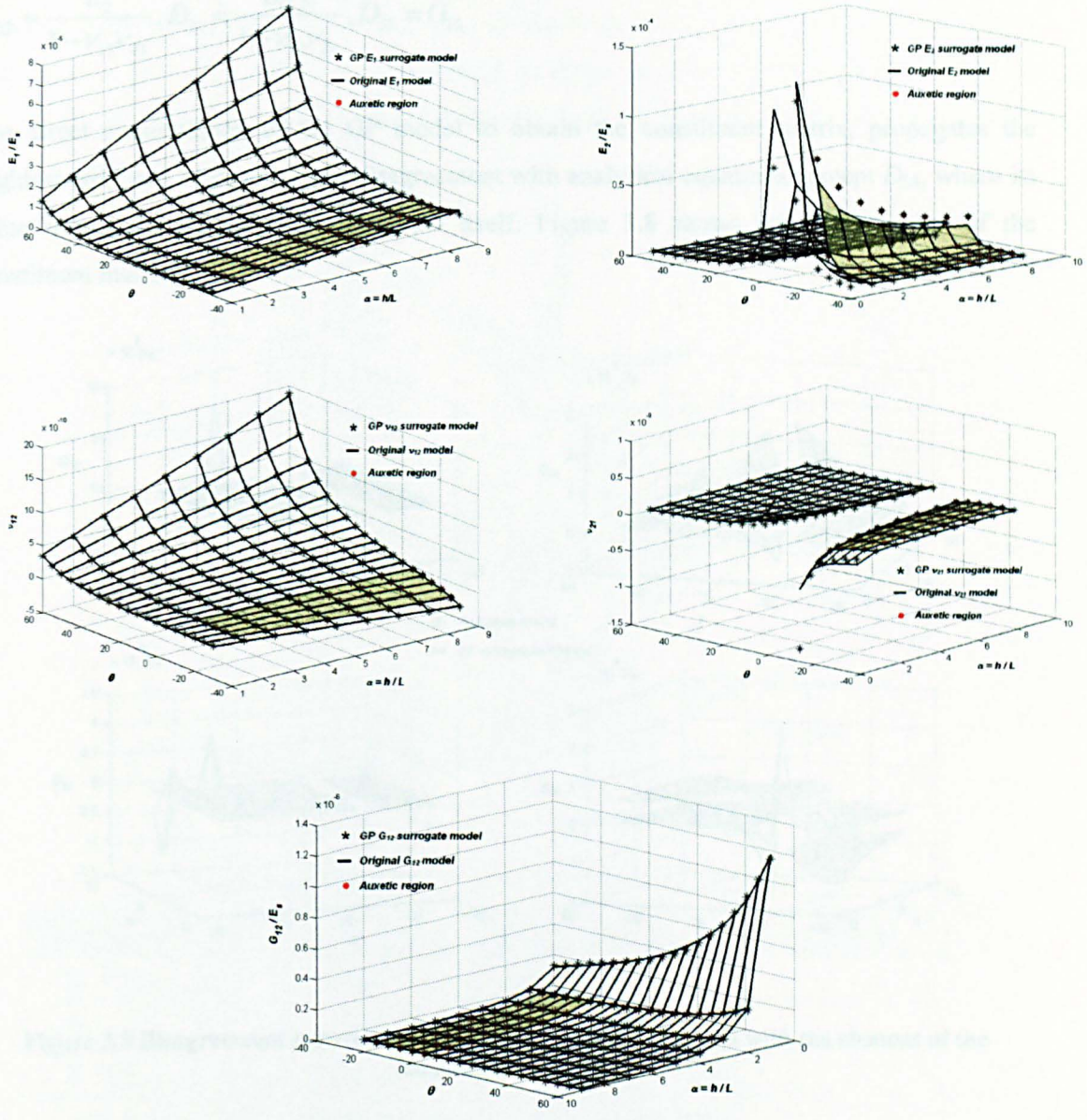


Figure 3.7 Comparison of fitness between GP model and equation (3.3) - (3.5).

The constituent matrix elements are defined as:

$$D_{11} = \frac{E_1}{1-\nu_{12}\nu_{21}}, D_{12} = \frac{E_2\nu_{12}}{1-\nu_{12}\nu_{21}},$$
$$D_{22} = \frac{E_2}{1-\nu_{12}\nu_{21}}, D_{21} = \frac{E_1\nu_{21}}{1-\nu_{12}\nu_{21}}, D_{33} = G_{12} \quad \textbf{(3.6)}$$

The direct manipulation of the GP model to obtain the constituent matrix, propagates the slightest error and leads to a large disagreement with analytical equations, except D_{33}, where its value is actually the original GP model itself. Figure 3.8 shows the disagreement of the constituent matrices.

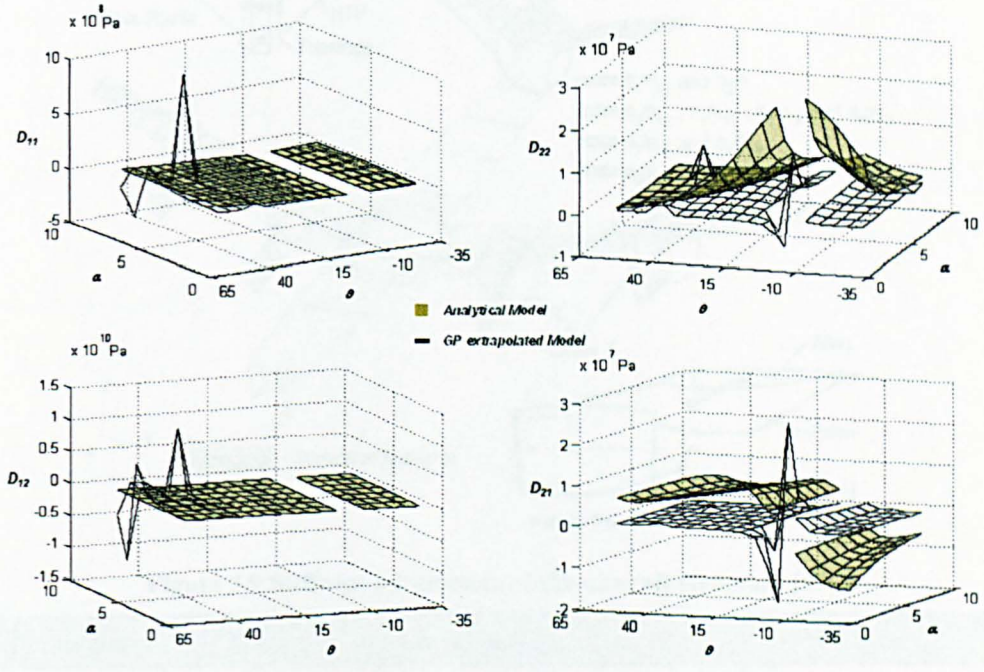


Figure 3.8 Disagreement between direct manipulation of GP model with the element of the constituent matrix.

This phenomenon is common among many regression models, as the extrapolation capability of a surrogate model is error sensitive. Therefore it is suggested that the extrapolation of the GP model should only be used with precautions. This may lead to improper evaluation of the objective functions, which is a function of the constituent matrix, during the optimisation process. It is suggested that instead of manipulating the GP metamodels after it is established, the data representing the element of the constituent matrix (equation (3.6)) must be used for training the GP metamodels.

iii. GP application in damage detection data

The GP development was also applied to data provided by the Los Alamos National Laboratory obtained from a FE damage simulation of a civil aircraft. According to [57], a face-centered cubic design (147 data points) was generated to relate the nine stiffness parameters of the aircraft FE model with the first natural frequency and six elements of its associated modal displacement.

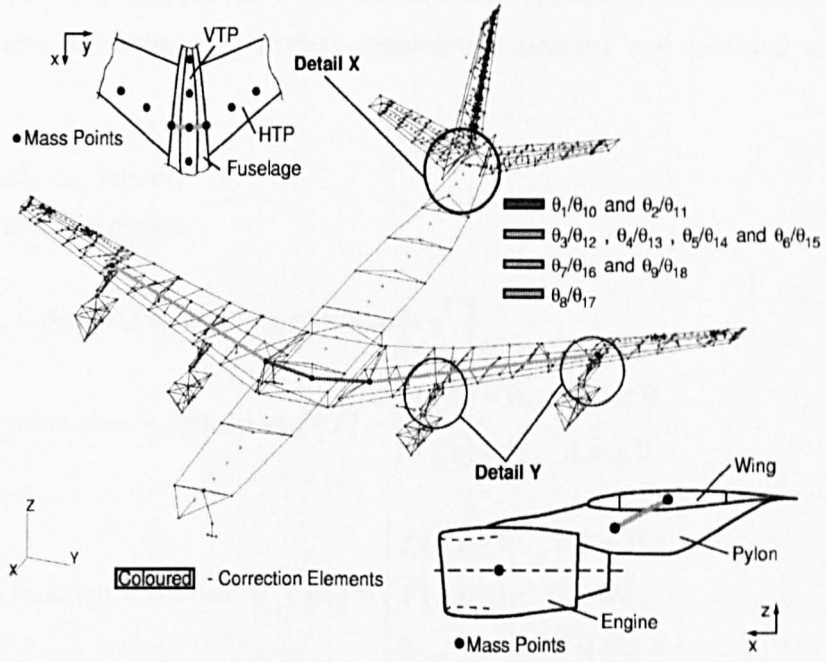


Figure 3.9 Stiffness parameters of the aircraft structure [57].

Parameter	Location	Type
$X_1 = X_{10}$	Fuselage / Wing connection (l/r)	I_{min}
$X_2 = X_{11}$	Fuselage / Wing connection (l/r)	I_{max}
$X_3 = X_{12}$	Wing / Pylon connection at outer engine (l/r)	I_{min}
$X_4 = X_{13}$	Wing / Pylon connection at outer engine (l/r)	I_{max}
$X_5 = X_{14}$	Wing / Pylon connection at inner engine (l/r)	I_{min}
$X_6 = X_{15}$	Wing / Pylon connection at inner engine (l/r)	I_{max}
$X_7 = X_{16}$	Fuselage / HTP connection (l/r)	I_{min}
$X_8 = X_{17}$	Wing (l/r)	E
$X_9 = X_{18}$	Fuselage / HTP connection (l/r)	I_{max}

Table 3.2 Stiffness parameters of the aircraft structure [57].

Figure 3.9 and Table 3.2 illustrate the stiffness parameters of the aircraft structure. The parameters are symmetrical along the aircraft length; therefore, there are only nine parameters [57].

Each of the 9 stiffnesses was 'damaged' by varying their values, which were normalised to −1, 0 and +1 [17]. The damaged stiffnesses will affect the whole modal behaviour of the aircraft. However, only the first natural frequency and the first element of the associated modal displacement were used in this works for illustration.

The GP was applied with the aim of constructing a relationship between the above-mentioned numerical simulated inputs and outputs. The 147 data points were split into training (100), validation (27) and testing (20) sets. The GP terminal set consisted of randomly generated real values and nine normalised parameters representing bending and torsional stiffness of the aircraft.

$$T = \{\mathbf{R}, \text{input1}, \ldots, \text{input9}\} \tag{3.7}$$

The GP functions used were:

$$F = \left\{+, -, \times, \div, \text{power}, \sin, \cos, \log, \text{ramp}, \text{step}, \sqrt{}\right\} \tag{3.8}$$

where the step function is defined as $f(x) = \begin{cases} f(x) = 0, & \text{if } x < 0 \\ f(x) = 1, & \text{if } x \geq 0 \end{cases}$

and the ramp function is defined as $f(x) = \begin{cases} f(x) = 0 & \text{if } x < 0 \\ f(x) = 1 & \text{if } x \geq 1 \\ x & \text{if } 0 \leq x < 1 \end{cases}$

The terminals have zero arity, while the function set has an associated arity set {2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1}.

The data were used to train the GP ten times for each different probability combination of genetic operators (crossover and mutation rate), with ten different initial conditions (i.e. random number seed). A population of 300 and 2000 generations were used throughout. A validation procedure was then performed to find the best initial condition from every combination of the trained models, followed by a testing procedure on the best validated model, in order to check the generalisation of the GP model. It is important to know if the model has successfully obtained the input-output relationship, so that it will give a good prediction of future data. An un-generalised model only fits the model to the data used, which means that the model only works on that particular set of data, and gives poor prediction of future data.

There are some internal settings in the GP code that do not vary throughout the GP parameters tuning process (these values are set based on the heuristic experience of user and do not vary throughout the training procedures):

- Number of generations and populations.
- The population allows identical individuals.
- There is an elite individual chosen at each generation.
- Only a maximum of 100 nodes is allowed in each individual.
- The observation over the GP development shows that the system operates better with a less drastic mutation. Therefore, each different mutation strategies takes a certain percentage of the user defined mutation rate:
  - Branch mutation (38.5%)
  - Branch insertion (23.1%)
  - Branch chopping (7.7%)
  - Mutation at the function nodes (23%)
  - Mutation at the terminal nodes (7.7%)

The best GP surrogate model describing the above dataset was compared with a polynomial (linear and quadratic) and MLP model. All utilising the same training, validation and testing sets.

The MLP model was created using the Neural Network – MLP program. Only one hidden layer was used, however the number of hidden nodes and the number of presentations with different initial condition (i.e. random number seed) are adjusted to find the best network configuration. The network is first trained using the training set, and validated using the validation set, the lowest validation MSE model will be chosen, and will be tested using the testing set for its generalisation.

a. Frequency Surrogate Model

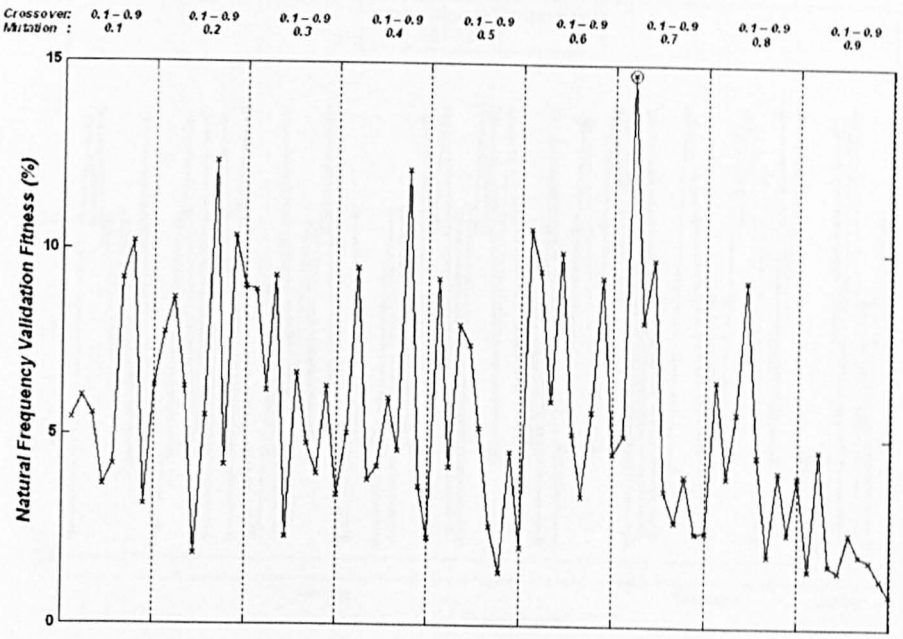
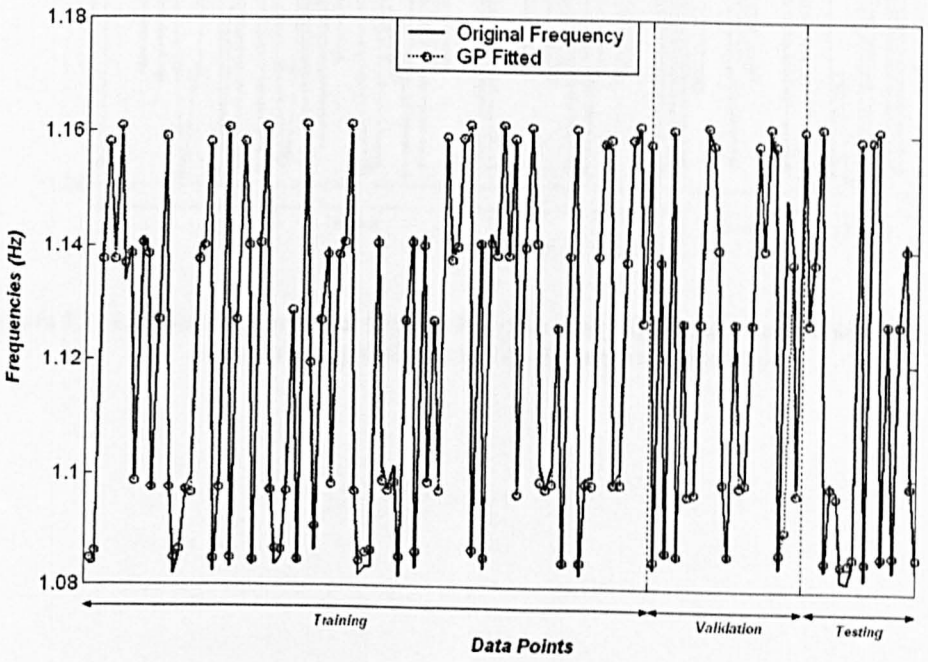
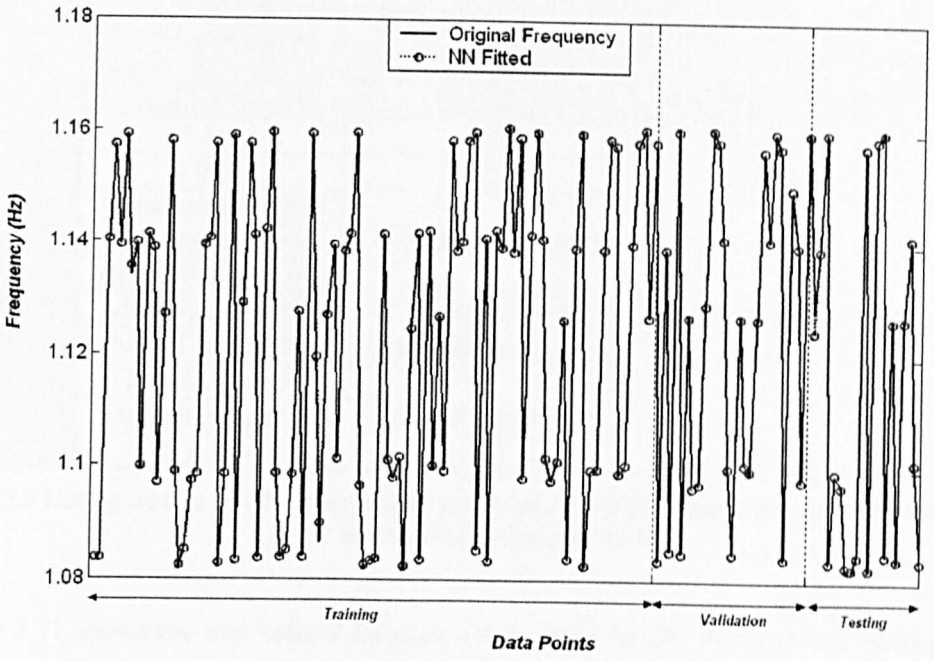


Figure 3.10 Validation Fitnesses from GP - Natural Frequency model.

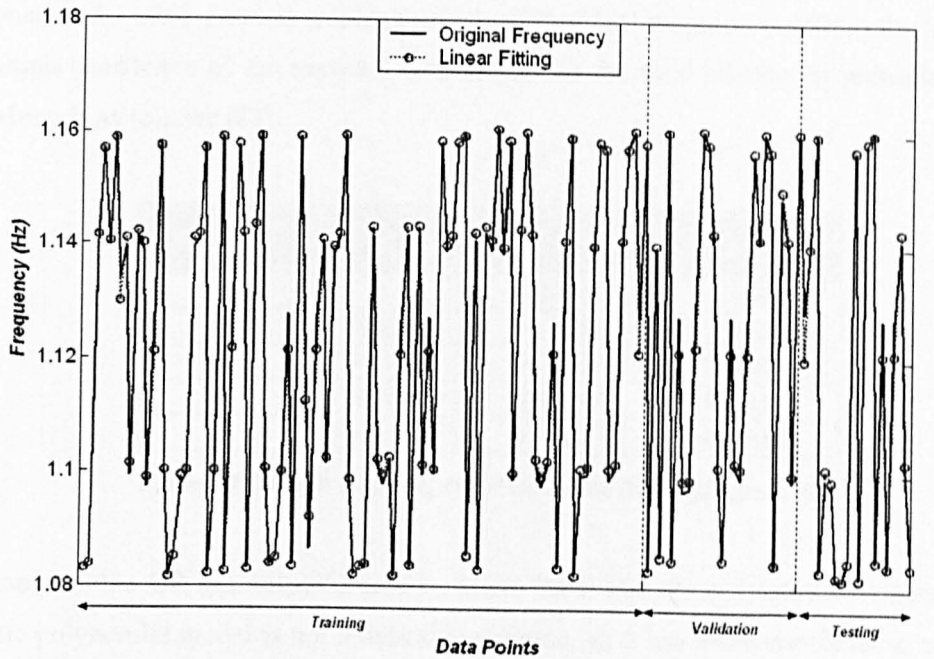
Figure 3.10 illustrates the best validation fitness of the 81 GP parameter combinations applied to the natural frequency data. The overall best combination has mutation rate 0.2, and crossover rate 0.7. This gives a validation fitness of 14.7118 (marked with a circle), and testing fitness of 15.8802.



(a)



(b)



(c)

Figure 3.11 Comparison among (a) Genetic Programming (b) Neural Network and (c) Linear polynomial regression model on the natural frequency.

<table>
<tr><th colspan="5">Frequency (Hz)</th></tr>
<tr><th>Surrogate Model</th><th>GP</th><th>Neural Network</th><th colspan="2">Polynomial</th></tr>
<tr><th></th><th colspan="2"></th><th>Linear</th><th>Quadratic</th></tr>
<tr><td>Fitness</td><td>15.8802</td><td>10</td><td>1.1363</td><td>9.7561</td></tr>
<tr><td>%MSE</td><td>0.0630</td><td>0.1</td><td>0.8800</td><td>0.1025</td></tr>
<tr><td rowspan="9">Configurations</td><td>Generations: 2000</td><td rowspan="3">Hidden nodes: 2</td><td rowspan="9">No. of Coef.: 10</td><td rowspan="9">No. of Coef.: 55</td></tr>
<tr><td>Population: 300</td></tr>
<tr><td>Crossover rate: 0.7</td></tr>
<tr><td>Mutation rate: 0.2</td><td rowspan="2">Learning and momentum regime: 2</td></tr>
<tr><td>No. of nodes: 95</td></tr>
<tr><td>Tree Depth: 22</td><td rowspan="3">No. of weights: 23</td></tr>
<tr><td>Inputs selected:</td></tr>
<tr><td>1, 2, 4, 8</td></tr>
</table>

Table 3.3 Configuration of GP, Neural Network, Linear and Quadratic polynomial models in fitting the natural frequency data.

Figure 3.11 shows the first natural frequency data fitted by GP, NN and the linear polynomial. The configurations of each of the metamodel are listed in Table 3.3. The regime number mentioned in the table above is a default setting of the MLP program, it defines the learning and momentum coefficient of the network with respect to the total number of presentation of the data, which is as follows [81]:

Regime Number	Regime Begins	Learning Coefficient	Momentum Coefficient
1	0	0.30000	0.40000
2	10000	0.15000	0.20000
3	30000	0.03750	0.05000
4	70000	0.00234	0.00313
5	150000	0.00001	0.00001

Table 3.4 Default learning schedule in the MLP program [81].

The fitness of the GP has outperformed both the MLP and the polynomial metamodels. The quadratic polynomial model is not a desirable solution, as it has more coefficients, which leads to cumbersome manipulations. There is also some evidence of over-fitting, as a higher fitness cannot be achieved without increasing the order of the polynomial and hence its number of coefficients.

Also, from Table 3.3, the GP model utilises only 4 input variables instead of all 9 of them. It shows that inputs 1, 2, 4 and 8 have the strongest influence on the natural frequency of the simulation model, which agrees with [57]. The inputs 1, 2, 4 and 8 as indicated in Table 3.2, represent for the aircraft's wing / fuselage connections, wing / pylon connection of the outer turbine engine and the wing stiffness respectively.

Both the GP and polynomial were running on a PC (P4, 3.06GHz and 512MB RAM), while the Neural Network was running on a Sun Grid Engine (10xV880 machine @ 8x900MHz, total 32GB Physical Memory).

The time taken to calculate the results from each of the final metamodels was a fraction of a second. However, the GP and Neural Network take roughly 20 hours for tuning (i.e. choosing the optimum number of hidden nodes, learning and momentum coefficients etc.), in order to achieve an optimal configuration of the model.

b. First Mode Shape Surrogate Model

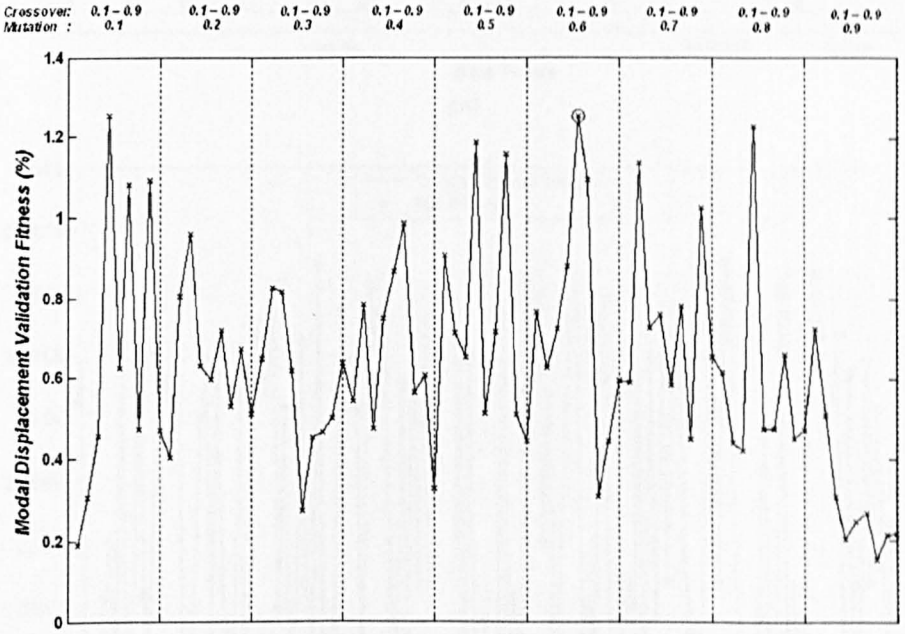
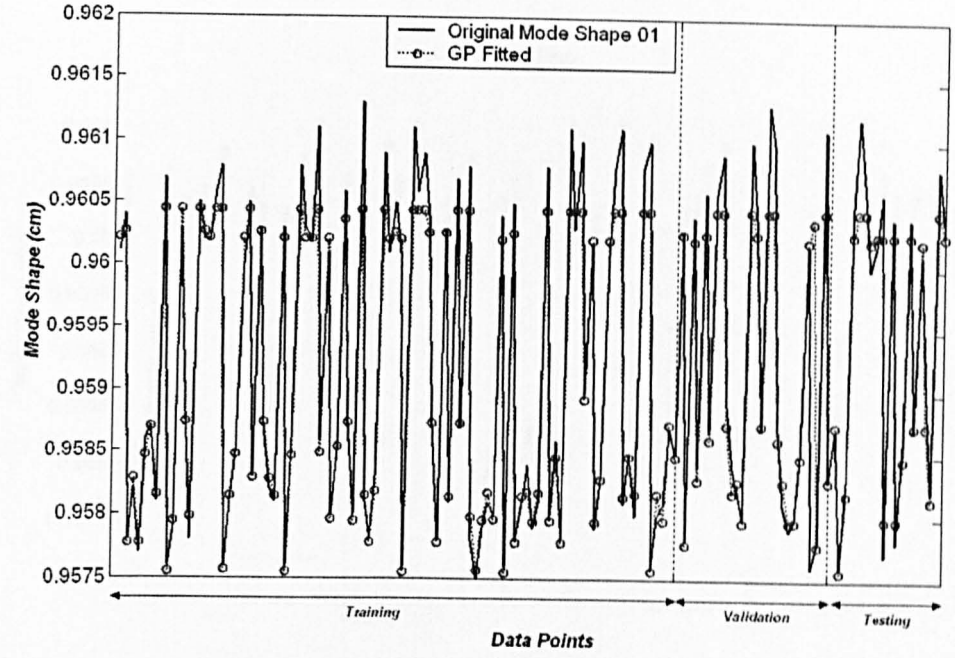
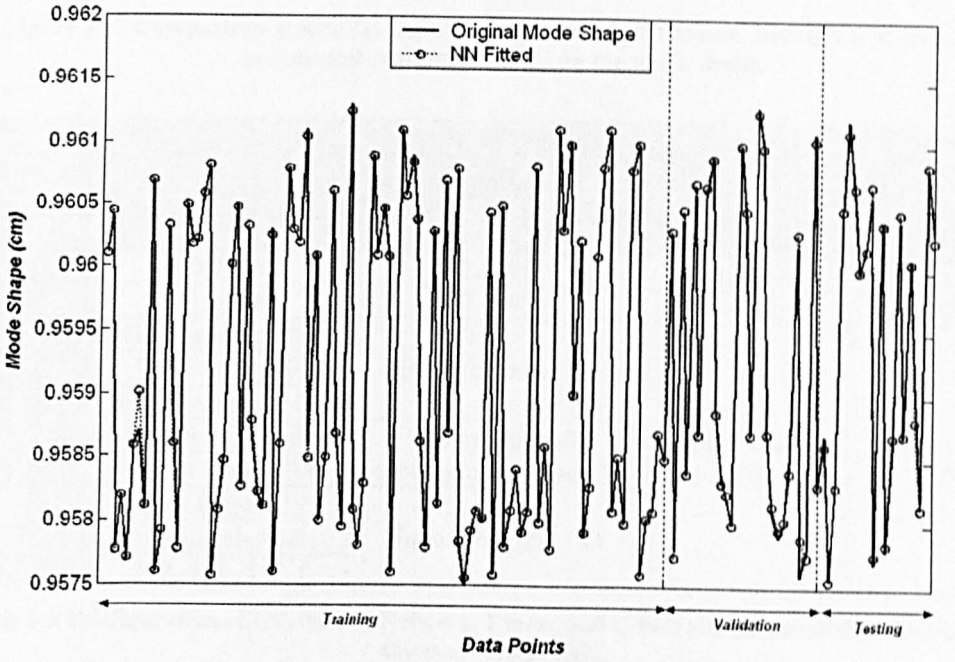


Figure 3.12 Validation Fitnesses from GP – Mode Shape model

Figure 3.12 indicates that the best GP configuration for the mode shape is 0.6 crossover rate and 0.5 mutation rate, which gives a validation fitness of 1.2549 (marked in circle) and testing fitness of 0.9448. The tuning process is similar to that for the natural frequency data.

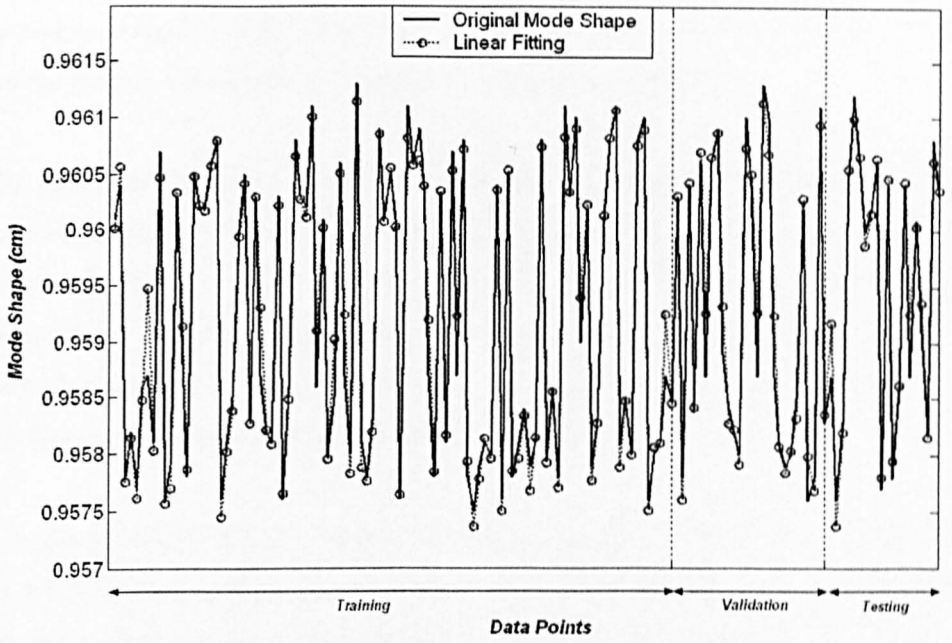


(a)



(b)

Figure 3.13 Comparison among (a) Genetic Programming (b) Neural Network and (c) Linear polynomial regression model on the mode shape.



(c)

Figure 3.13 Comparison among (a) Genetic Programming (b) Neural Network and (c) Linear polynomial regression model on the mode shape.

Mode Shape (cm)				
Surrogate Model	GP	Neural Network	Polynomial	
			Linear	Quadratic
Fitness	0.9448	1.9608	0.2917	1.4130
%MSE	1.0584	0.5099	3.4282	0.7077
Configurations	Generations: 2000	Hidden nodes: 13	No. of Coef.: 10	No. of Coef.: 55
	Population: 300			
	Crossover rate: 0.6			
	Mutation rate: 0.5	Learning and momentum regime: 2		
	No. of nodes: 100			
	Tree Depth: 19	No. of weights: 144		
	Inputs selected:			
	1, 2, 3, 4, 6, 7, 8			

Table 3.5 Configuration of GP, Neural Network, Linear and Quadratic polynomial models in fitting the mode shape data.

Figure 3.13 shows the performance of all models on the mode shape data. The mode shape data proved to be a challenge for all the methods used to try fitting a metamodel. As shown in Table 3.5, all the surrogate models have become more complicated, except the polynomial, where the number of coefficients was fixed. The MLP model has 13 hidden nodes with 144 weights, while the GP has generated a tree with 19 levels with maximum allowable nodes (100 nodes). It is worthwhile to note that both the polynomial and Neural Network model used all the input variables to perform the metamodelling; while GP only used seven out of nine variables, without much effect on its fitness.

It shows that except for inputs 5 and 9, which denote for the wing / pylon connection of the inner turbine engine and the fuselage / HTP connection from Table 3.2, all other parameters affect the output. The selection of parameters agrees with [57].

So far, the GP has proved to be successful in the aforementioned examples. The performance of GP is compatible with the MLP and polynomial models. Though the GP is still in a rather initial stage of development, where there are still a lot of rooms for improvement. This includes the bloat control to avoid complex and big trees, more effective penalisation strategy, such as using simulated annealing to place a penalisation weight on the size of individuals that will decrease as the generation increases and other selection methods.

The large generation of 10000 used at the beginning of the studies was set by trial and error; the main purpose was just to test if the GP programming coded so far shows a promising performance. Subsequent testing on the honeycomb properties has shown that the GP gives a good convergence without having large generations and populations.

4. Metamodelling for bounded microstructure models

The term 'bounded microstructure' comes from the geometry of the unit cells, whereby the unit cell has an enclosed / bounded domain. In this chapter, the unit cell is a unit square with a void in the middle. Its FE model was developed in ANSYS [6] and the periodic boundary conditions were applied on the enclosed boundaries using homogenisation theory. The in-plane properties of the unit cell were calculated based on the homogenised FE model. These properties were then fed into a Neural Network and Genetic Programming algorithm to create a replacement / surrogate model of the Finite Element model.

i. Homogenisation development & its convergence

The numerical homogenisation code was benchmarked on a domain composed of a periodic unit cell as shown in Figure 4.1. The geometrical configuration of the unit cell was deliberately chosen according to [27], in order to verify the FE homogenisation code developed in this work.

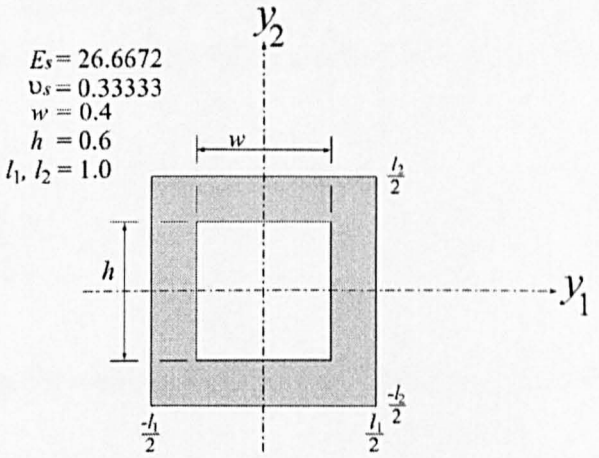


Figure 4.1 Unit cell configuration and its core material properties.

The unit cell model was created using 2-dimensional 8-node elements (PLANE82) in the commercial finite element code ANSYS61 [6]. There are 1908 elements after meshing the full unit cell described above. However by taking advantage of its symmetrical geometry, only a quarter of the cell was simulated.

The periodic boundary condition applied to the cell is by imposing a periodic strain (i.e. displacement / length of the cell border) on its boundary. According to (2.1), a characteristic function (it can be any physical field, such as a displacement function, or stress function) at point **x** in a heterogeneous domain, can be viewed as a minor perturbation of a homogeneous

domain. The minor perturbation is a rapidly changed function within a very small neighbourhood of **x**. Therefore, the displacement function of the unit cell in the y_1 and y_2 directions can be written as:

$$u = e_1 y_1 + u^{\varepsilon} \tag{4.1}$$

$$v = e_2 y_2 + v^{\varepsilon} \tag{4.2}$$

where e_i (i = 1, 2) are the total strains during deformation of the cell in the y_i (i = 1, 2) direction, the term u^{ε} denotes the fast changing periodic displacement of the unit cell and y_1, y_2 is the length of each side of the cell, i.e. l_1, l_2. The above equations can be expanded to become:

$$u = \varepsilon_{11} y_1 + \varepsilon_{12} y_2 + u^{\varepsilon} \tag{4.3}$$

$$v = \varepsilon_{21} y_1 + \varepsilon_{22} y_2 + v^{\varepsilon} \tag{4.4}$$

The strains (ε_{11}, ε_{12}, ε_{21} and ε_{22}) need to be calculated in order to define the finite element boundary conditions at the boundaries of the cell.

By applying a unit displacement to the cell in the y_1 and y_2 directions, the periodic displacements between two opposite walls have the following relationships:

$$\begin{aligned} u\left(-\frac{l_1}{2}, y_2\right) &= u\left(\frac{l_1}{2}, y_2\right) \\ v\left(y_1, -\frac{l_2}{2}\right) &= v\left(y_1, \frac{l_2}{2}\right) \end{aligned} \tag{4.5}$$

In the case of shearing, the periodic displacements between the opposite walls are described by:

$$\begin{aligned} v\left(-\frac{l_1}{2}, y_2\right) &= -v\left(\frac{l_1}{2}, y_2\right) \\ u\left(y_1, -\frac{l_2}{2}\right) &= -u\left(y_1, \frac{l_2}{2}\right) \end{aligned} \tag{4.6}$$

Substituting (4.5) or (4.6) into (4.3) and (4.4) gives the normal strain loading or shear strain loading for the cell boundaries respectively:

$$\varepsilon_{11} = \frac{u\left(-\frac{l_1}{2}, y_2\right) - u\left(\frac{l_1}{2}, y_2\right)}{l_1}$$

$$\varepsilon_{22} = \frac{v\left(y_1, -\frac{l_2}{2}\right) - v\left(y_1, \frac{l_2}{2}\right)}{l_2} \qquad \textbf{(4.7)}$$

$$\varepsilon_{33} = \frac{1}{2}(\varepsilon_{12} + \varepsilon_{21}) = \frac{1}{2}\left(\frac{u\left(y_1, -\frac{l_2}{2}\right) - u\left(y_1, \frac{l_2}{2}\right)}{l_2} + \frac{v\left(-\frac{l_1}{2}, y_2\right) - v\left(\frac{l_1}{2}, y_2\right)}{l_1}\right)$$

where ε_{11} and ε_{22} are the strain in the y_1 and y_2 directions respectively, ε_{33} is the shear strain in plane y_1, y_2.

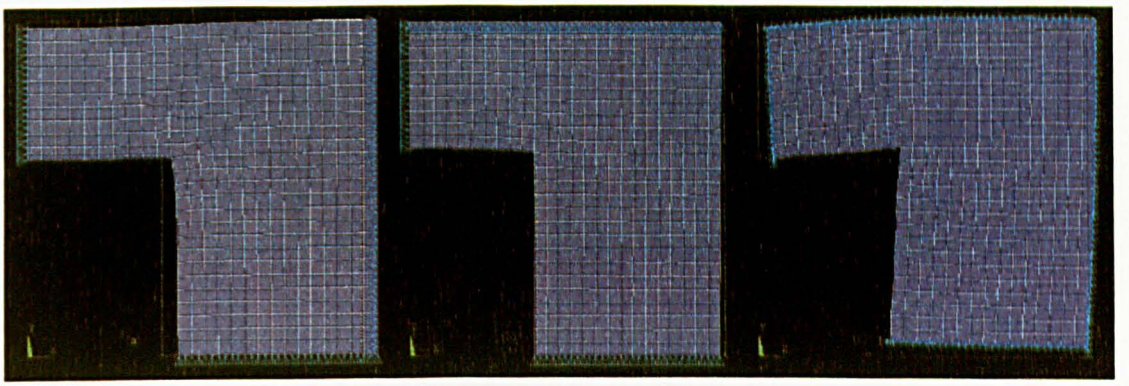


Figure 4.2 (a) Normal loading in y_1 direction; (b) Normal loading in y_2 direction; (c) Shear loading in y_1 y_2 direction.

The deformation of the unit cell under each loading together with their boundary conditions is illustrated in Figure 4.2. A unit displacement was applied on the edge of the cell, while the lines of symmetry were constrained; in the case of shearing, half the unit displacement is applied on every side and the lines of symmetry remain constrained. The resultant displacement of the elements along the edges were recorded for the post processing in order to obtain the elastic moduli (E_1, E_2), the relative Poisson's ratios (ν_{12}, ν_{21}) and in – plane shear modulus (G_{12}). The displacement was divided by the total length of the unit cell to obtain strain. Reaction forces within the model were also divided by the total cross sectional area to obtain the stress. The effective homogenised properties were calculated as the ratio between the stress and strain. These structural properties obtained from the post processing were used to calculate the elements of the stiffness matrix, C:

$$C = \begin{bmatrix} C_{11} & C_{12} & 0 \\ C_{21} & C_{22} & 0 \\ 0 & 0 & C_{33} \end{bmatrix} \tag{4.8}$$

where

$$C_{12} = C_{21} = \frac{E_1 \nu_{21}}{1 - \nu_{21}\nu_{12}},\ C_{11} = \frac{E_1}{1 - \nu_{21}\nu_{12}},\ C_{22} = \frac{E_2}{1 - \nu_{21}\nu_{12}},\ C_{33} = G_{12} \tag{4.9}$$

These results were compared with [27] in Table 4.1. In general, the result proves that the numerical homogenisation code developed so far was successful.

	C_{11}	C_{12}	C_{22}	C_{33}
FE simulation	12.670	3.446	17.767	2.639
Ref. [27]	12.820	3.124	17.407	2.634

Table 4.1 Comparison of the unit square cell structural properties with [27].

A general comparison of the unit cell model with void sizes ranging between 0.1 and 0.9 can be seen in Figure 4.3. The bi-cubic polynomial model from [27] is listed in Appendix 1. The mean square error for each case is: 7.026% for C_{11}, 19.184% for C_{12}, 7.306% for C_{22} and 2.449% for C_{33}. A high error C_{21} is observed and it is suspected to be a compounded error, whereby the error is propagated when formulating (4.9) using the results obtained from each of the FE loading conditions mentioned above.

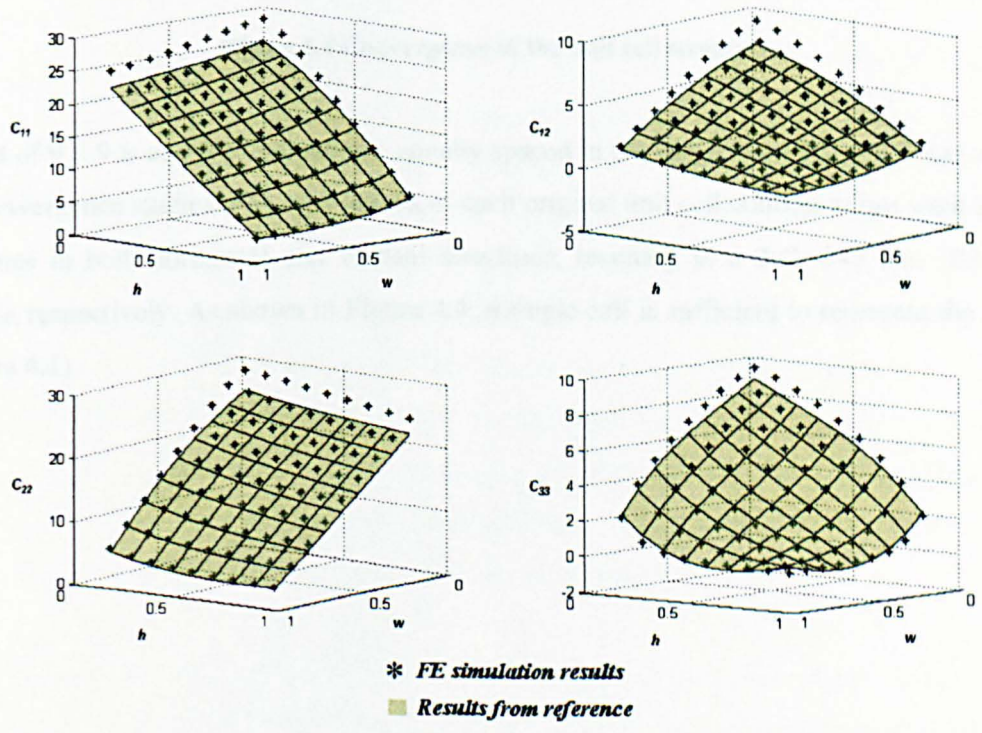


Figure 4.3 Numerical simulation results of void model structural properties in comparison with ref. [27]

a. Convergence study of the homogenisation

A study of the scale aspect of the numerical homogenisation was made to understand the convergence of the heterogeneous multiple cell into the homogenised unit cell. The homogenised model is expected to approach closer to the structural properties of the macrostructure as the unit cell duplicates, i.e. as $\varepsilon \rightarrow 0$.

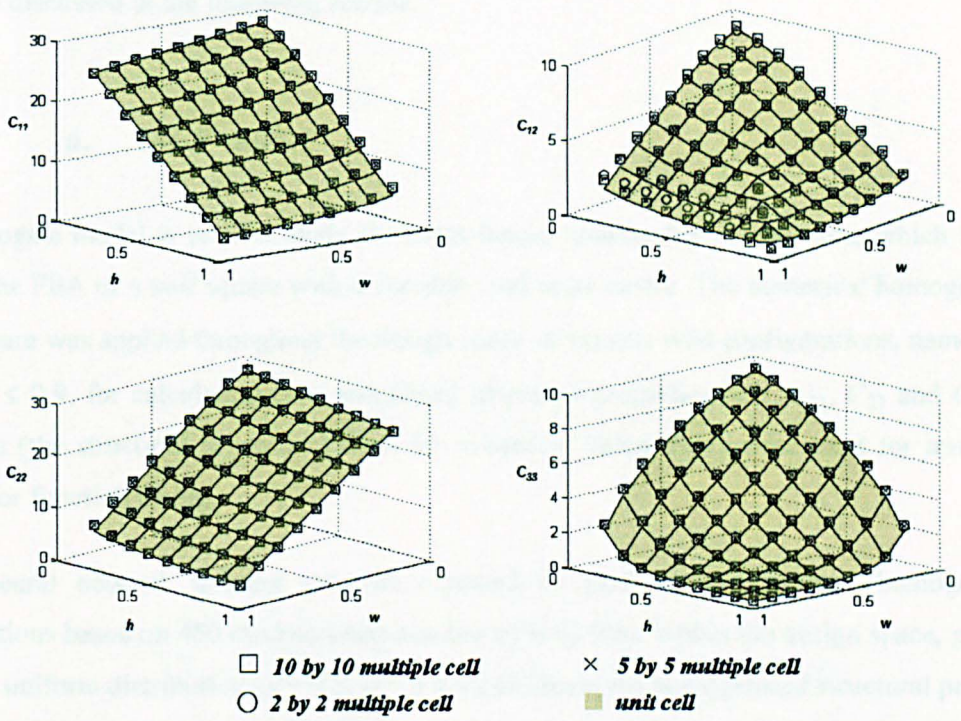


Figure 4.4 Convergence of the unit cell structure.

A grid of 9 × 9 w and h combinations, equally spaced in the range from 0.1 to 0.9 was used for the convergence studies. The replications of each original unit cell configurations were 2, 5 and 10 times in both horizontal and vertical directions, resulting in a 2×2, 5×5 and 10×10 grid domain respectively. As shown in Figure 4.4, a single cell is sufficient to represent the domain (Figure 4.1).

ii. Surrogate models setup

This section is about how to set up a surrogate model using the data gathered from numerical simulations of the unit cell model mentioned in the earlier sections. The surrogate models / metamodels discussed here are Neural Network – Multi-Layer Perceptron (MLP) and Genetic Programming (GP). Their training, validation, testing procedure and performance comparison will be discussed in the following section.

a. MLP model

A surrogate model is used to study the input-output relationship of a system, which is in this case, the FEA of a unit square with a variable void in its centre. The numerical homogenisation procedure was applied throughout the design space of various void configurations, namely $0.1 \leq (w, h) \leq 0.9$, for calculating their associated structural properties (C_{11}, C_{21}, C_{22} and C_{33}). The outputs (the structural properties) from the numerical calculations were used for training the MLP for function fitting.

The neural network training set was obtained by performing numerical homogenisation simulations based on 450 random combinations of void sizes within the design space, generated from a uniform distribution ($\mu = 0.5$, $\sigma = \pm 0.4$), to obtain the homogenised structural properties.

The input – output relationship that needs to be studied by the MLP is the void configuration and its associated homogenised structural properties. The homogenised structural properties are the MLP desired output, while the void sizes are the inputs of the MLP. The system is now a double input - single output MLP system, and there are four such systems, each representing C_{11}, C_{12}, C_{22} and C_{33}. The 450 input – output pairs were trained in the MLP using a Unix Sun Grid Machine. The training process involved tailoring the connection weights and number of hidden nodes in each hidden layer and the number of presentations of data, which defines the learning and momentum coefficients. Only a single hidden layer is used here, as the functions being fitted are rather simple. There were all together 250 MLP network configurations (maximum 25 hidden nodes and number of presentation from 100 to 1000 with an interval of a hundred) each configuration was trained under 10 random initial conditions.

The resultant network was validated by using a separate dataset. The validation set consists of 450 random data points. The validation process is to find the best combination of the network configurations. The validated network would then be tested for its generalization. The testing set consists of 81 void combinations (w and h) that were equally spaced between 0.1 and 0.9, and their corresponding homogenised structural properties.

Table 4.2 shows the best MLP configuration found, they are determined by the lowest validation MSE. Their corresponding testing set MSE values are also listed in the table.

	C_{11}	C_{21}	C_{22}	C_{33}
No. of hidden nodes	1	1	13	8
Learning and momentum regime	2	2	2	2
No. of weights	5	5	53	33
Validation fitness = 1/MSE%	0.769	0.256	0.588	9.090
Validation MSE %	1.3	3.9	1.7	0.11
Testing fitness	0.588	0.071	0.400	0.909
Test set MSE %	1.7	14	2.5	1.1

Table 4.2 The best MLP network configuration for each model: C_{11}, C_{21}, C_{22} and C_{33}.

As shown in Figure 4.5, the MLP surrogate model gives a good agreement compared to the FE model. A performance comparison and discussion about all the metamodels used will be given at the end of the next section.

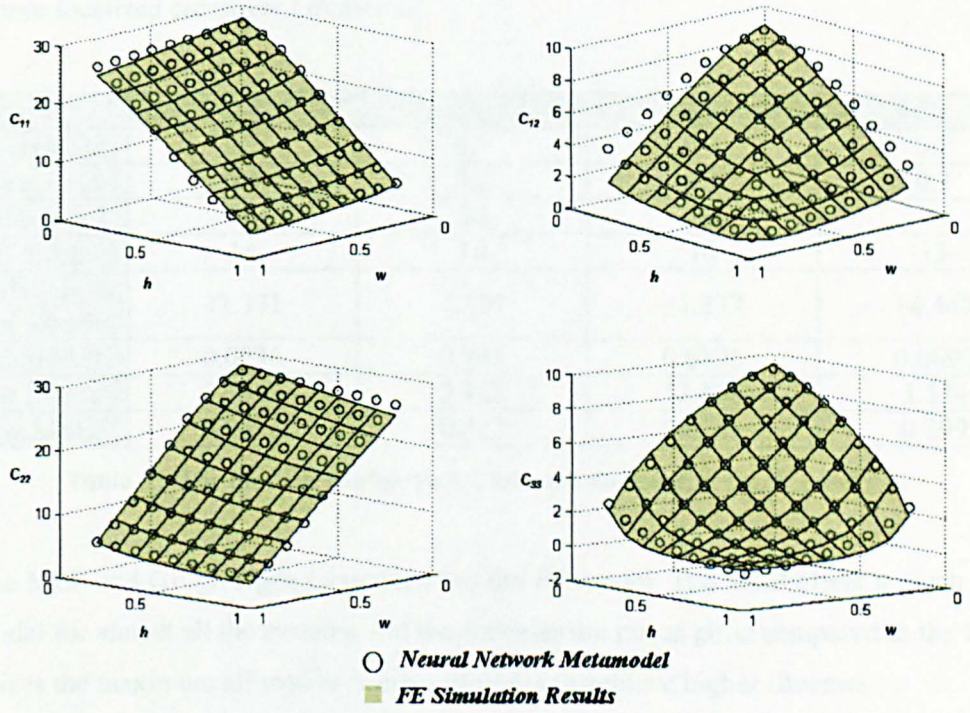


Figure 4.5 Comparison between MLP models with FE models.

b. GP model

The GP metamodels of the unit square FE model were also constructed based on the same set of training, validation and testing set mentioned earlier. The GP metamodels were trained using a normal PC. All the GP metamodels were fixed to have only 1000 generations with 300 individuals, each model being trained 10 times (with different initial condition, i.e. random number seeds) for each different probability combination of genetic operators (i.e. mutation and crossover rate). The trained models were validated to obtain the best GP metamodel (i.e. individual), followed by a testing procedure to obtain the generalisation of the data in the range of $0.1 \leq (w, h) \leq 0.9$.

The best GP configuration for each system together with the validation fitness and the corresponding testing fitness are listed below. The GP program was created in a way that encouraged the individuals to grow in any shape until they reached the maximum allowable nodes, which in this case is 50 nodes for each metamodel. The maximum levels of the individuals are not fixed here, as this helps protect the individuals from being chopped too much over the crossover / mutation process, which allow a finer tune of an individual by performing a much more localized crossover / mutation.

	C_{11}	C_{21}	C_{22}	C_{33}
Crossover Rate	0.7	0.5	0.4	0.2
Mutation Rate	0.4	0.2	0.3	0.3
No. of nodes	50	50	50	50
No. of levels	14	16	16	13
Validation fitness = 1/MSE	42.371	7.107	51.233	14.449
Validation MSE	0.0236	0.141	0.0195	0.0692
Testing fitness	22.963	2.341	13.468	3.334
Testing MSE	0.0435	0.427	0.0743	0.299

Table 4.3 The best GP configuration for each model: C_{11}, C_{21}, C_{22} and C_{33}.

Both the MLP and GP give good agreement to the FE model. The MLP offers a much simpler metamodel for almost all the systems, but the fitnesses are not as good compared to the GP. The GP utilizes the maximum allowable number of nodes to achieve higher fitnesses.

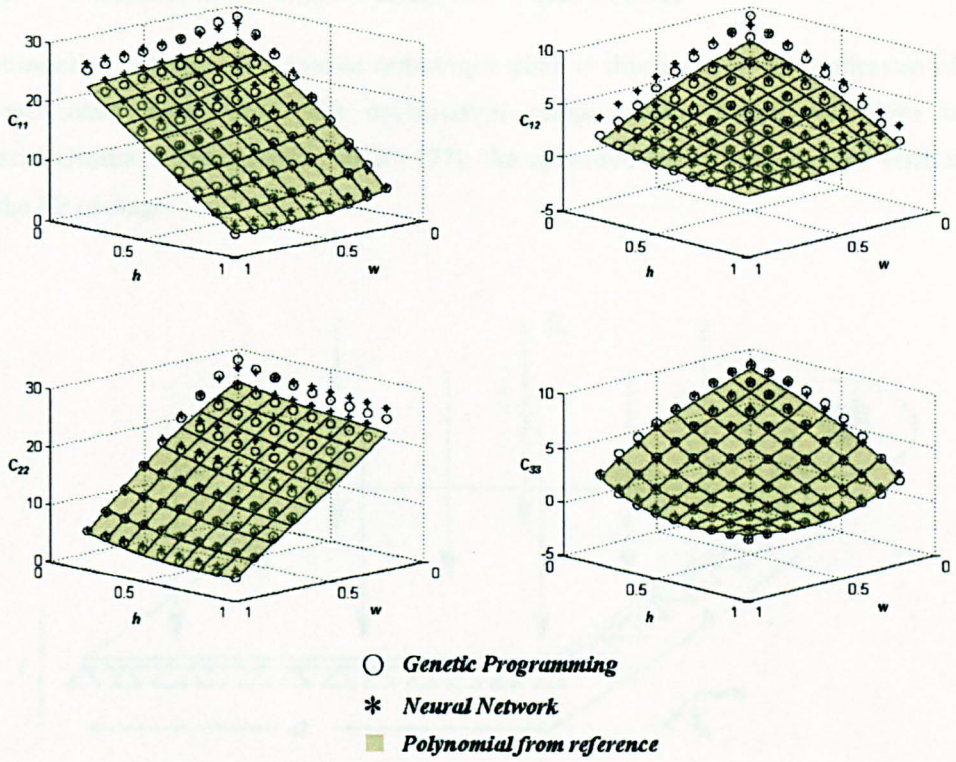


Figure 4.6 Comparison among all metamodels.

Also in Figure 4.6, both of the metamodels agree well with [27]. It is worth mentioning that the bi-cubic polynomial literature models were built using conventional Least Squares (Appendix 2), and the range of void configurations used were [0.0, 1.0], the structural properties of the unit cell were not obtained from the same source as this work. The surrogate models constructed here were only operating in the void range of [0.1, 0.9]. This may bring some disagreement between the MLP / GP models and the literature models. Apart from this issue, the problem with ANSYS compound error with C_{12} calculation has also contributed to the error in function fitting.

iii. Optimisation Case Study for Void Model

An optimisation example of a special orthotropic plate is illustrated as an application of the above-mentioned metamodels. The optimisation computational routine was done using Sequential Quadratic Programming (SQP) [77]; the optimised results were further verified by using the FE package.

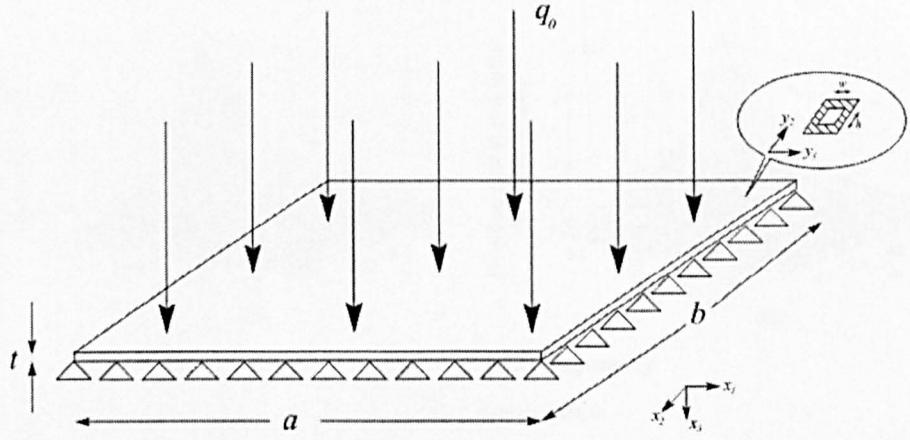


Figure 4.7 Illustration of the optimisation problem.

Figure 4.7 shows the plane stress optimisation example: A square Aluminium plate is simply supported on all sides, it consists of a hundred replications of the void microstructure (Figure 4.1), i.e. $\varepsilon(w)\times\varepsilon(h)=0.1\times0.1=0.01$. As $\varepsilon \to 0$, the plate will become a homogeneous flat panel which is made with infinitely small voids, much like a kitchen scourer.

A uniformly distributed load, q_0, was applied on the top face. The displacement function of the plate, u_z, is described by [79, 47]:

$$u_z = \frac{a^4}{\pi^4}\sum_{m=1}^{\infty}\sum_{n=1}^{\infty}\frac{q_{mn}}{D_{mn}}\sin\left(\frac{\pi m x}{a}\right)\sin\left(\frac{\pi n y}{b}\right), \tag{4.10}$$

$$\text{where } q_{mn} = \begin{cases} \dfrac{16q_0}{\pi^2 mn} & \text{if } m, n = 1, 3, \ldots \\ 0 & \text{if } m, n = 2, 4, \ldots \end{cases}$$

$$a = b = 10\text{m},\ t = 0.2\text{m},\ q_0 = \frac{100\text{N}}{ab(1-wh)},\ 1\times10^{-4}\text{m} \le u_z \le 1\times10^{-3}\text{m},$$

$$D_{mn} = D_{11}m^4 + 2(D_{21} + 2D_{33})(mnR)^2 + D_{22}(nR)^4,$$

$$R = \frac{a}{b},\ D_{ij} = \int_{-t/2}^{t/2} C_{ij}z^2 dz \quad i,j = 1,2,3$$

The constraints of the microstructure geometry were set as: $0.1 \leq (w, h) \leq 0.9$. The optimisation procedures were performed using the built-in command in the MATLAB65 optimisation toolbox (*fmincon*) [77]. The aim is to find the minimum plate displacement of the constrained nonlinear multivariable displacement function (4.10) and the corresponding microstructure configuration based on SQP. All values within the range of microstructure configuration were used as initial guesses (81 initial guesses). Only those that produce feasible solutions from the optimisation were illustrated in Figure 4.8.

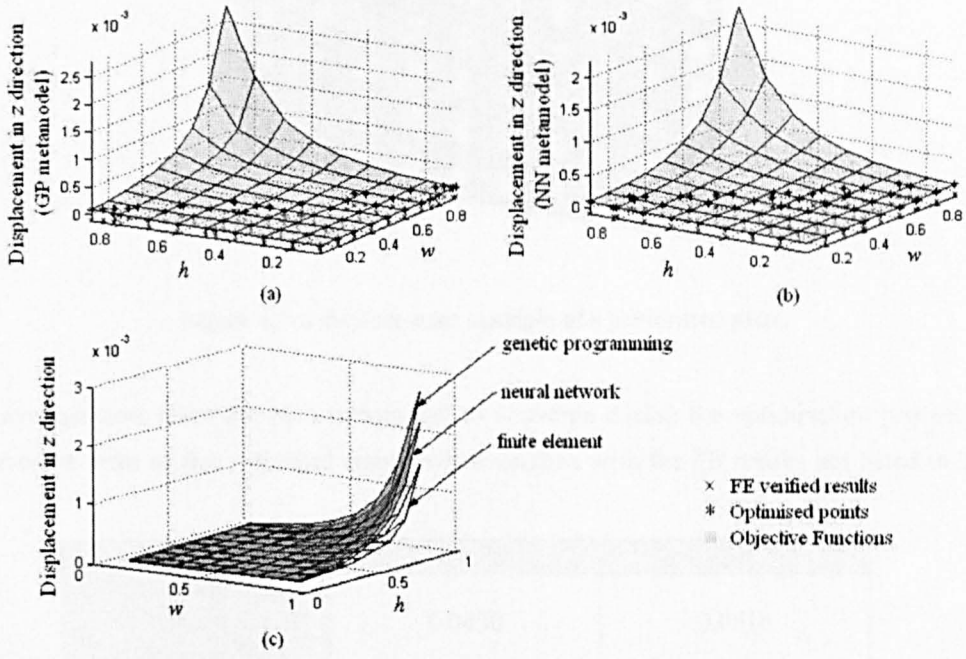


Figure 4.8 Optimised points under the objective functions built using (a) NN; (b) GP and (c) Difference among the objective functions built using different metamodels.

The sub-functions (D_{mn}) used in the objective function (4.10) are calculated based on the surrogate models developed in previous sections. Instead of calculating the sub-function by calling a FE program, the metamodels act as a subroutine of the optimisation process, saving some computational effort. Two optimisation routines were created using two different metamodels (MLP and GP), in order to study the performance differences between them.

Figure 4.8(c) shows the objective functions (4.10) created using the metamodels and the FE model. The FE models were generated using the SHELL63 element [6], with 10 replications of the unit cell in vertical and horizontal directions. The out-of-plane displacements on all four sides of the plate were constrained and the load was applied as an equally distributed force on the top face of the plate. Figure 4.9 shows an example of the FE model. The FE simulated plate displacement shown in Figure 4.8(c) had only 81 points within the range of $0.1 \leq (w, h) \leq 0.9$. It shows the differences between the metamodel, which encapsulate the analytical model, and

the FE model, the analytical model described by (4.10) has a higher estimation compared to the FE model, both the analytical model developed from MLP and GP are agreeing well.

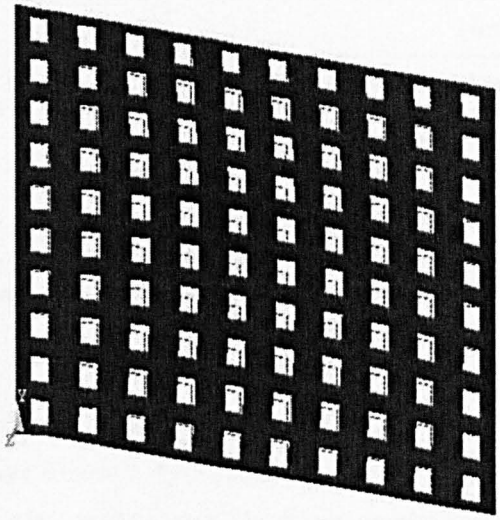


Figure 4.9 A displacement example of a perforated plate.

The average time taken for each metamodel to converge during the optimisation process, and the average error of the optimised results when verified with the FE results are listed in Table 4.4.

	MLP	GP
Average time of convergence (seconds)	0.0430	0.0816
Error when compared to FE (% MSE)	2.7304	2.6427

Table 4.4 Performance of metamodels in optimisation routine.

The reason that the GP runs slower may be that the limits of the mathematical operators used by GP are different between Java and MATLAB. The deployment of the GP metamodels in MATLAB needs to redefine the GP mathematical operators by creating user-defined subroutines in MATLAB, which are called at every evaluation of the relevant mathematical operators within the GP metamodel. These redefined mathematical functions include square roots, natural logs, power and division. This problem may be overcome by revising the GP code, in order to have the Java program handling NaN (Not a Number) and Inf (Infinite) values, alternatively, by performing the optimisation in Java, or creating the GP code in MATLAB environment. These user-defined mathematical functions were redefined as follows:

	Redefined Mathematical functions	Operating Condition
Square roots	$\sqrt{\lvert x \rvert}$;	None
Natural logs	$\ln \lvert x \rvert$;	If $x = 0$, this mathematical function will return 0.
Powers	$\lvert x \rvert^{y}$	If $x = 0$, this mathematical function will return 0.
Division	$\frac{x}{y}$	If $y = 0$, this mathematical function will return 0.

Table 4.5 Redefined mathematical functions used in metamodels developed from Genetic Programming.

The results that successfully converged from the optimisation process were further verified by using the FE package. From Figure 4.8(a), (b) and Table 4.4, both the metamodels shows good agreement with the FE model, and the numerical homogenisation code developed in this work has also represented the homogeneous structure successfully.

5. Metamodelling for cellular microstructure models

Cellular materials have been used for centuries and they are common in nature, for example cork, wood, coral and sponge. The term 'cell' comes from Latin *cella*, which means a small compartment; and the 'cellular' is from the Roman *cellarium*, which is a cluster of cells [23].

Cellular solids are made up of an interconnected network of solid ligaments or plates forming the edges and frames of cells. There are at least three common types of cellular solids. The simplest one is a two-dimensional array of hexagonal polygons clustered to create a plane area like the honeycombs of the bee [23]. Other more common ones are polyhedra packed in three-dimensional space like foams. There are open-cell and closed-cell foams, the latter ones much like a three-dimensional variation of the previously mentioned two-dimensional honeycomb; and the open-cells has planes of honeycomb criss-crossing within the three-dimensional space.

Apart from the naturally occurred ones, man has created more synthetic cellular materials, from simple honeycomb-like panels to foam-like products. The unique characteristics of the cellular solids have given rise to creative applications, such as insulations, energy absorbers and lightweight sandwich panels.

Sandwich panels were invented to improve structural integrity and stiffness in lightweight structures. Classical sandwiches consist of two face sheets at the top and bottom of the sandwich core. The face sheets are generally tougher than the core material, while the core material is normally very light in weight, but has exceptional out-of-plane properties, for example the honeycombs.

The development of sandwich core materials has brought a large advancement in engineering design, as natural materials do not have such excellent weight-to-strength ratio. The sandwich panels are further exploited to achieve a better functionality. One of the possible approaches is to replace the conventional hexagonal honeycomb core by auxetic honeycombs, to develop an auxetic sandwich panel. The advantages of such a substitution include better crashworthiness and shear rigidity of the sandwich panel.

In this chapter the focuses are on the sandwich core (cellular material) design, optimising the shape of the honeycomb core using Differential Evolution (DE), and the effectiveness of metamodels in representing the honeycombs. The sandwich core used here is the centresymmetric hexagonal honeycomb.

i. Homogenisation and metamodel development

Unlike the bounded models, the cellular models have discontinuous unit cells (Figure 5.1). The boundary conditions are similar to the ones used as a homogenisation test case in previous chapters, but are further constrained with constraint equations that describe the motions of ligaments on all sides.

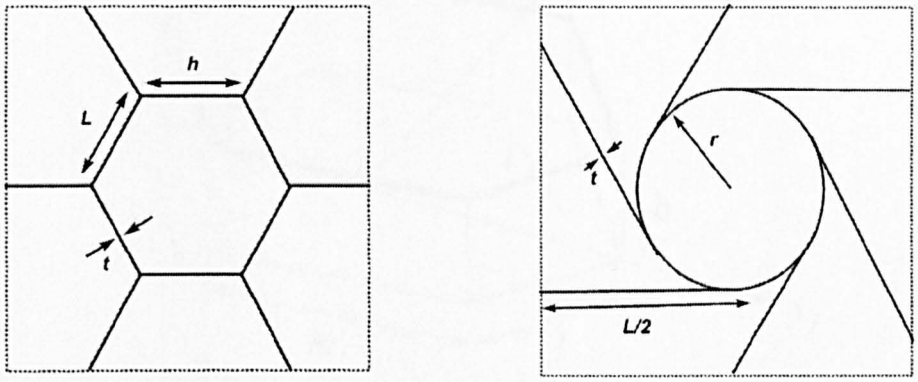


Figure 5.1 (a) Centre-symmetric hexagonal cell in a unit square. (b) Non-centre-symmetric chiral cell in a unit square.

The term centresymmetry means that the unit cell is symmetrical along any line passing through the geometric centre of the cell. A centresymmetric structure can repeat itself with translations along the horizontal and vertical directions. On the contrary, a non-centresymmetric (Figure 5.1(b)) unit cell has symmetry lines that are offset from the centre line, and it duplicates by combined rotations and translation, this will be discussed further in next chapter.

The FE model used here is a centresymmetric hexagonal honeycomb, as shown in Figure 5.2. When the angle, θ, becomes a negative value, which forms a re-entrant hexagonal honeycomb, this re-entrant honeycomb model exhibits an auxetic behaviour (Figure 3.3). The dimensionless parameters defining the model are: $\alpha = h / l$, $\beta = t / l$, $\gamma = b / l$, where b is the gauge thickness of the sandwich core. The symbols of non-dimensional parameters are consistent with the one used in [23].

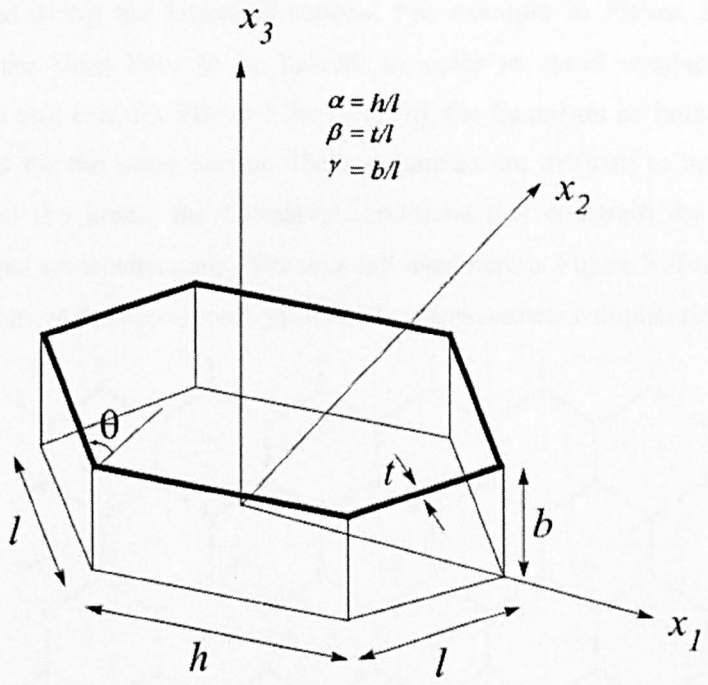


Figure 5.2 Illustration of hexagonal honeycomb.

An auxetic honeycomb experiences expansion when being pulled along one direction, resulting in a volume increment of the structure. This negative Poisson's ratio effect improves the out-of-plane indentation / buckling resistance of an auxetic sandwich panel. The relationship between the unit cell geometry and its auxeticity will be discussed later in the section.

The in-plane auxeticity and the out-of-plane buckling of the honeycomb are studied here using FEA combined with homogenisation techniques. The first step of the process involves finding a unit cell that represents the honeycomb domain properly with boundary conditions describing the neighbourhood with other unit cells. There are no rules on how to choose a unit cell. [27] suggests that the unit cell:

1. should cover the whole range of voids and solids;
2. should fit the periodicity assumptions. (i.e. the unit cell should duplicate in a consistent way);
3. should be defined by a very few parameters.

There are many choices of unit cells available to represent a honeycomb domain (Figure 5.3); all have a good representation of the empty and solid areas of the domain. Not all types of representations duplicate the honeycomb domain in a centresymmetric manner, as can be seen in Figure 5.3(f). However, all unit cells are suitable for homogenisation studies, provided that the proper boundary conditions must be applied to describe the relative behaviours of the contiguous ligaments. It is also important to know that for some of the unit cells, the ligaments

have to be halved along the lateral directions. For example in Figure 5.3(b), the vertical ligaments along the sides have to be halved, in order to avoid overlapping areas during replications of the unit cell; for Figure 5.3(c) and (g), the ligaments on both left and right side have to be halved for the same reason. These examples are difficult to model as the halved ligaments occur at the joints, the boundary conditions that constrain the movement of the ligaments and joints are troublesome. The unit cell used here is Figure 5.3(e), for its simplicity, good representations of the honeycomb joints and centresymmetric duplication manner.

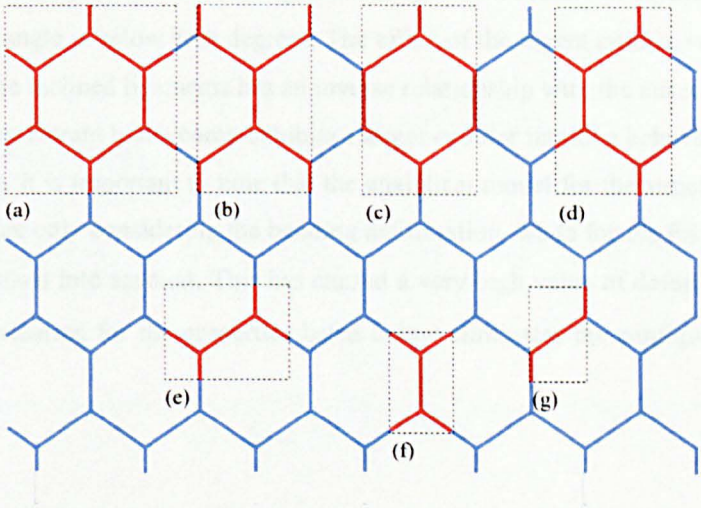


Figure 5.3 Various types of honeycomb unit cells.

The honeycomb geometrical parameters (α, β, γ and θ) were randomly generated based on a uniform distribution between the range of $1 \leq \alpha \leq 6$; $0.05 \leq \beta \leq 0.2$; $-30° \leq \theta \leq 60°$ and $\gamma = 1$.

The FE models built for in-plane studies were based on plane stress assumptions with 2-dimensional 8-node PLANE82 elements [6]. The free ligaments on all sides were constrained by using the constraint equations, describing the movements in relation to the contiguous ligaments. Figure 5.4 gives an example of the boundary condition used in x_2-direction loading.

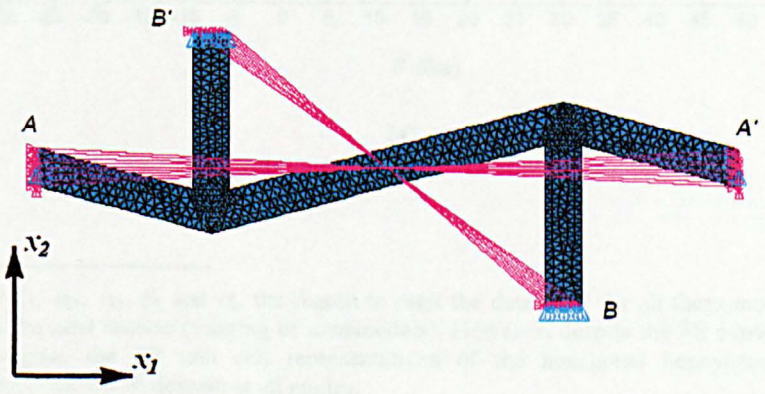


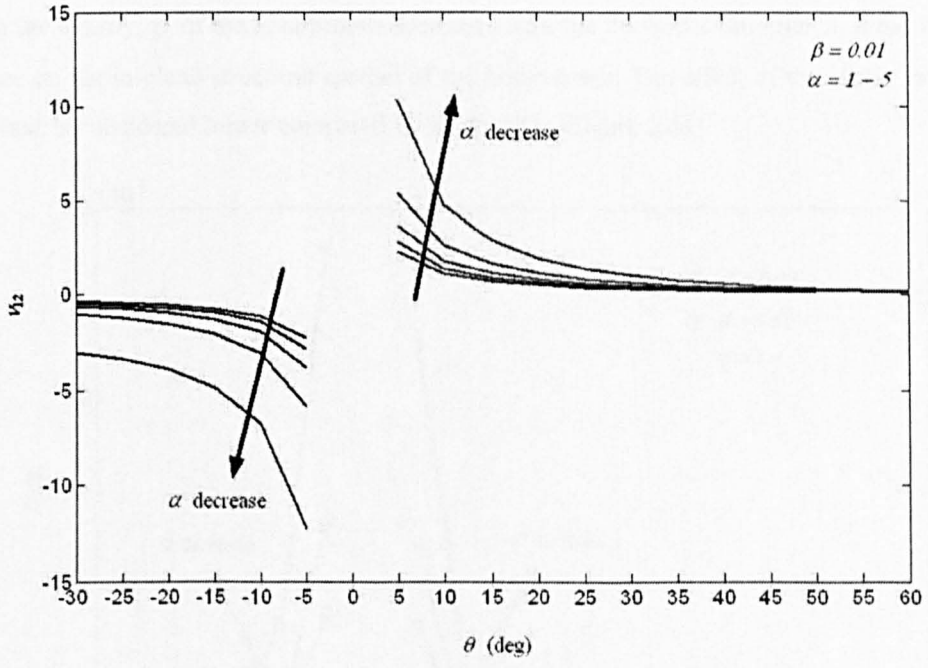
Figure 5.4 Boundary condition for loading in x_2 direction.

The idea of the constraint equations is similar to (4.5), for normal x_1 and x_2 direction loading, the displacements of the contiguous ligaments are:

$$u(\mathrm{A}, x_2) = u(\mathrm{A}', x_2)$$
$$v(x_1, \mathrm{B}) = v(x_1, \mathrm{B}') \qquad \textbf{(5.1)}$$

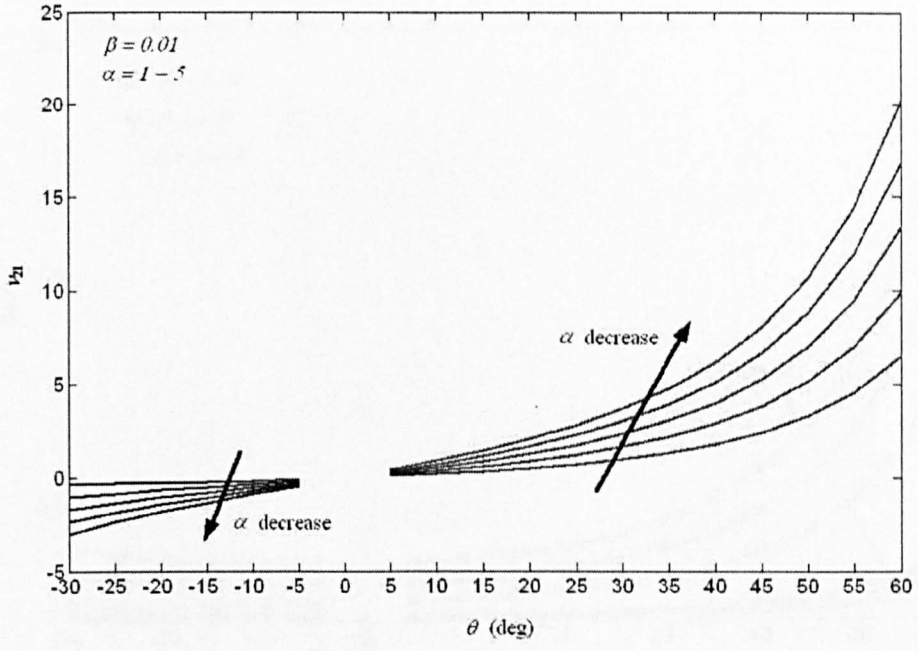
The load was imposed as a unit displacement at the tip of the ligaments.

The figure below shows the effect of the geometrical parameters affecting the in-plane auxetic behaviour of the honeycombs. The hexagonal honeycomb becomes a re-entrant honeycomb once the internal angle is below zero degrees. The effect of the aspect ratio, α, which contributes to the length of the inclined ligaments has an inverse relationship with the auxeticity. Also in the same figure, the re-entrant honeycomb exhibits a larger counter intuitive behaviour, as its aspect ratio gets smaller. It is important to note that the analytical model for the structural moduli and Poisson's ratios are only considering the bending deformation, while for the FE model, the axial stiffness is also taken into account. This has caused a very high value of deformation at $\theta = 0°$. Therefore the evaluation for the properties listed below eliminates the configuration when $\theta = 0°$ *.



(a)

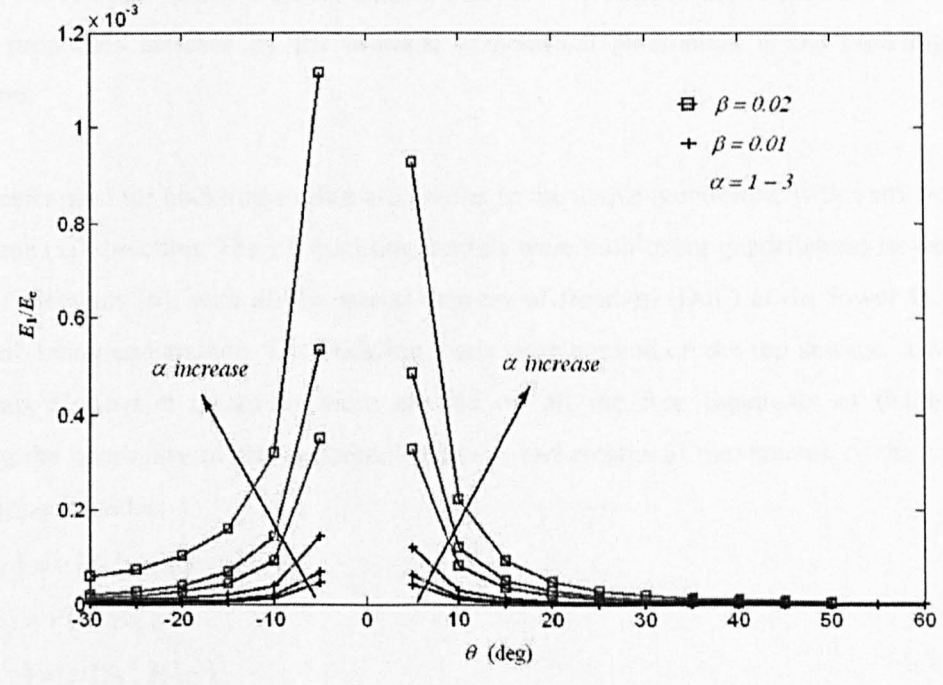
* Similarly, for E_1, ν_{21}, ν_{12}, E_2 and σ_c, the reason to omit the data at 0° for all these models was due to coherence with the next section (training of metamodels). However, despite the FE convergence limit at this difficult region, the FE unit cell representations of the hexagonal honeycomb give correct represention the honeycomb domain at all angles.



(b)

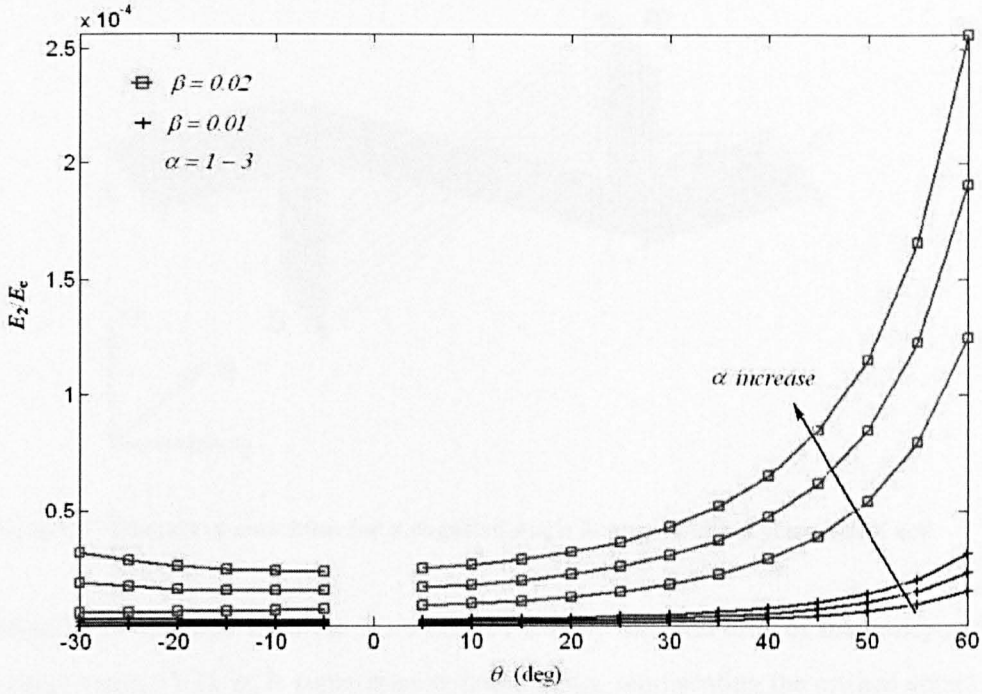
Figure 5.5 Effect of honeycomb geometrical parameters to Poisson's ratios (a) x_1-direction loading; and (b) x_2-direction loading.

Though the density, β, of the honeycomb does not affect the Poisson's ratio much, it has a large influence on the in-plane structural moduli of the honeycomb. The effect of the aspect ratio, α, in this case is considered minor compared to the density. (Figure 5.6)



(a)

Figure 5.6 Effect of honeycomb geometrical parameters to Structural Moduli (a) x_1-direction loading; and (b) x_2-direction loading. (cont.)



(b)

Figure 5.6 Effect of honeycomb geometrical parameters to Structural Moduli (a)) x_1-direction loading; and (b) x_2-direction loading.

The normal hexagonal honeycomb or the re-entrant honeycombs, having long and slender ligaments caused by large aspect ratios; or the thick and fat walls by larger densities have variations in the out of plane property as much as the variations in the in-plane. One of the out of plane properties affected by the in-plane geometrical parameters is the buckling of the honeycomb.

The unit cells used for buckling studies are similar to the above mentioned, with extrusion in the out-of-plane (x_3) direction. The FE buckling models were built using quadrilateral isoparametric SHELL93 elements [6], with all the lateral degrees of freedom (DoF) at the lower face of the honeycomb being constrained. The buckling loads were applied on the top surface, towards the honeycomb. Constraint equations were applied on all the free ligaments of the unit cell, describing the continuity of the horizontal, vertical and rotational movements of the ligaments on each opposite sides:

$$\begin{aligned} u(\mathrm{A},y,z) &= u(\mathrm{A}',y,z) \\ v(x,\mathrm{B},z) &= v(x,\mathrm{B}',z) \\ \omega(\mathrm{A},\mathrm{B},z) &= \omega(\mathrm{A}',\mathrm{B}',z) \end{aligned} \qquad \textbf{(5.2)}$$

where u, v and ω are x_1-direction displacement, x_2-direction displacement and rotation in x_1x_2-plane respectively. An example of the hexagonal honeycomb unit cell with all the boundary conditions is illustrated in Figure 5.7.

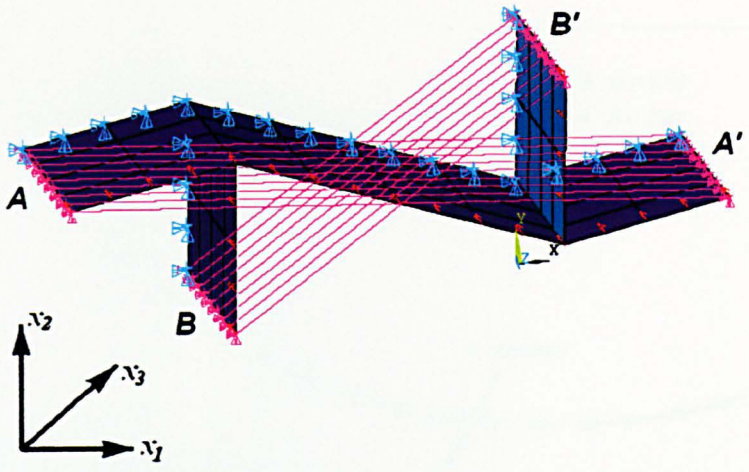


Figure 5.7 Boundary condition for a negative-angle hexagonal honeycomb unit cell.

The buckling force obtained from the FEA was divided by the total area of the honeycomb to get the critical stress, (5.3). σ_c is sometimes written as σ_{crit}, representing the critical stress. The total area of Figure 5.3(e) is a half of a hexagon area. The buckling stress was also normalised by the Young's modulus of the core material.

$$\sigma_c = \frac{F_{buckle}}{A_{total}} = \frac{F_{buckle}}{(hl\cos\theta + l^2\sin\theta\cos\theta)} \tag{5.3}$$

The FE results were compared to the empirical results obtained from theoretical formulae [23]. Equation (3.1) - (3.4) were used to verify the FE in-plane results; and (5.4) was used to verify the FE buckling analysis [23].

$$\frac{\sigma_c}{E_c} = \frac{2\left(\frac{1}{\alpha}+2\right)\beta^3}{\left(1-\nu_c^2\right)(\alpha+\sin\theta)\cos\theta} \tag{5.4}$$

The buckling resistance for the re-entrant honeycomb is higher compared to the normal honeycomb (Figure 5.8). Given a finite domain, when a pressure is applied towards the depth of a block of material domain, say a soap sponge, it will cause the sponge to expand in the horizontal direction. For any normal material, this horizontal expansion will cause one side, x_1 direction for example, to expand; while another side will contract. Hence the material collapses / buckles under pressure. On the other hand, if the material exhibits auxeticity, this horizontal expansion will cause all sides to expand. This reaction acts as a resistance towards the buckling pressure applied, which elevates the buckling resistance of the auxetic materials.

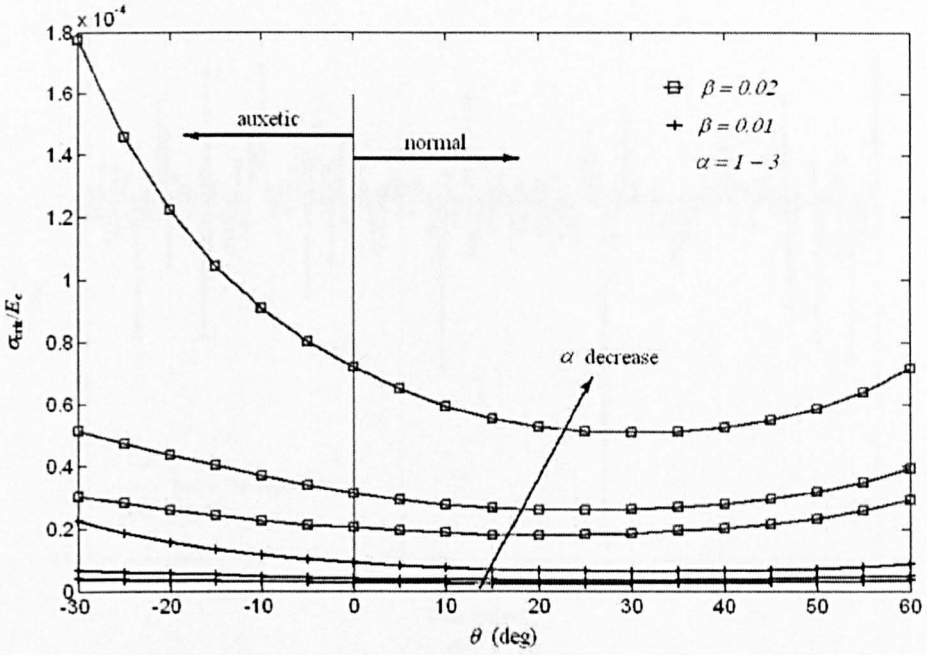
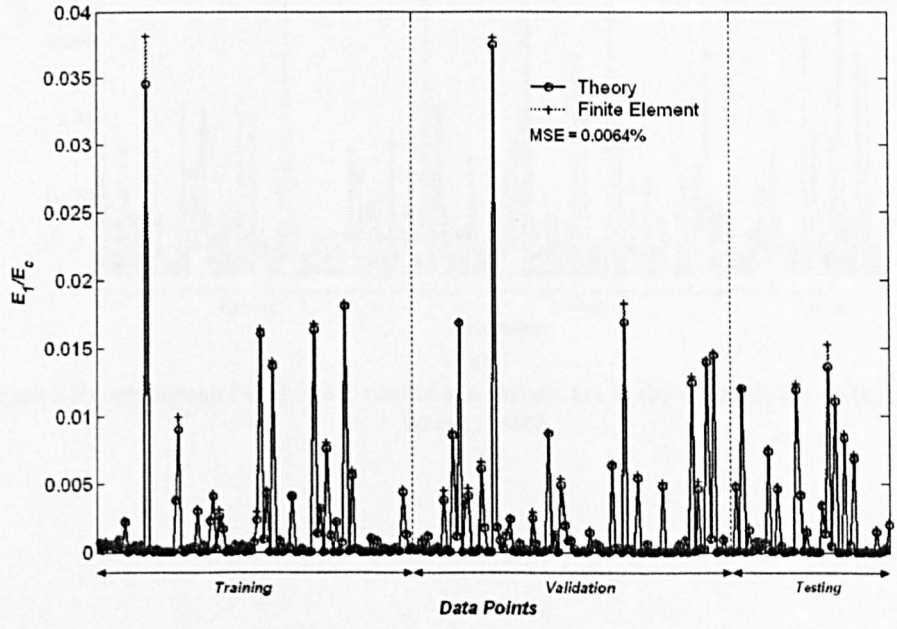


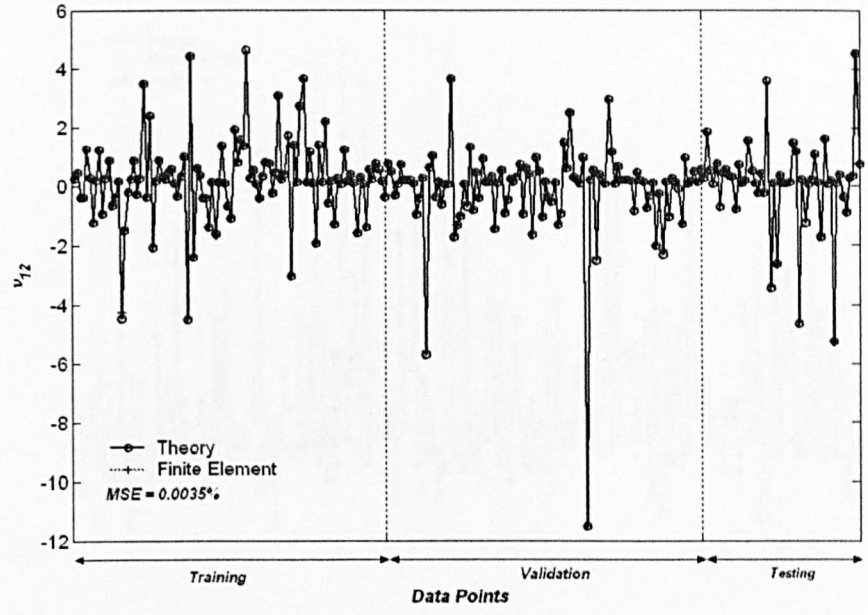
Figure 5.8 Buckling resistivity of auxetic and normal honeycombs.

The comparisons between the FE model and theoretical formulae are illustrated in Figure 5.9, each shows that the FE model gives good representation to the honeycomb domain, that the MSE between the FE and theory are minimal.

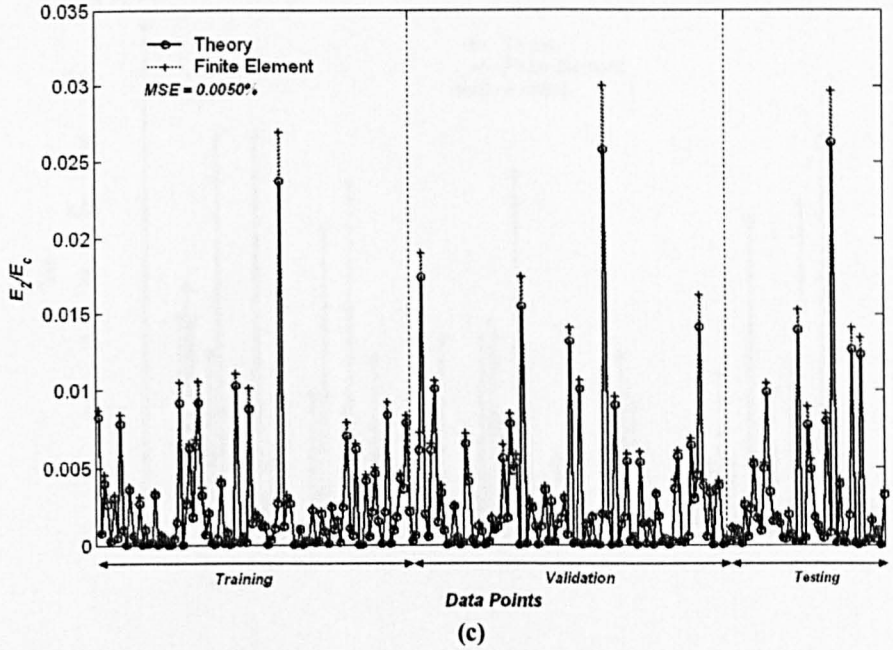


(a)

Figure 5.9 Comparison between FE results and theory. (a) E_1 (b) ν_{12} (c) E_2 (d) ν_{21} (e) buckling stress. (cont.)

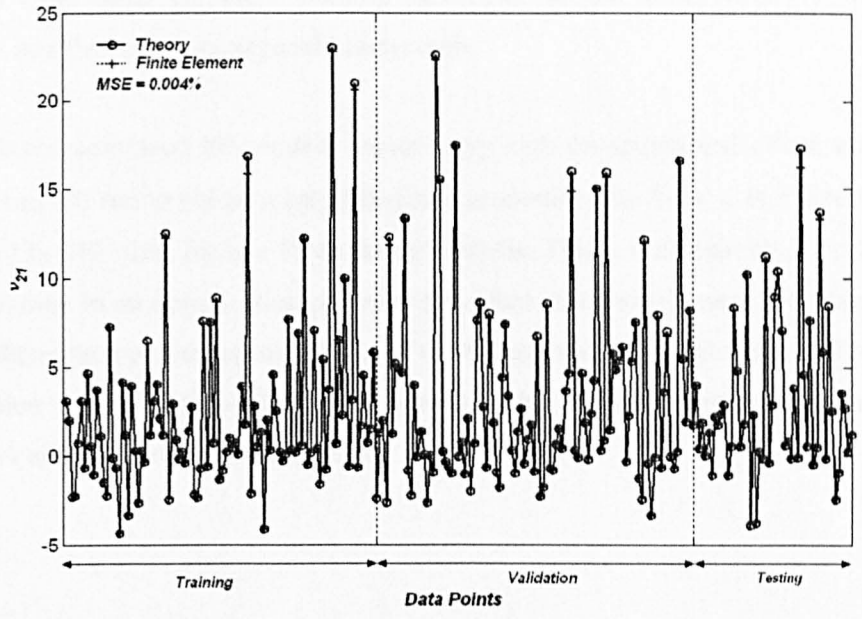


(b)

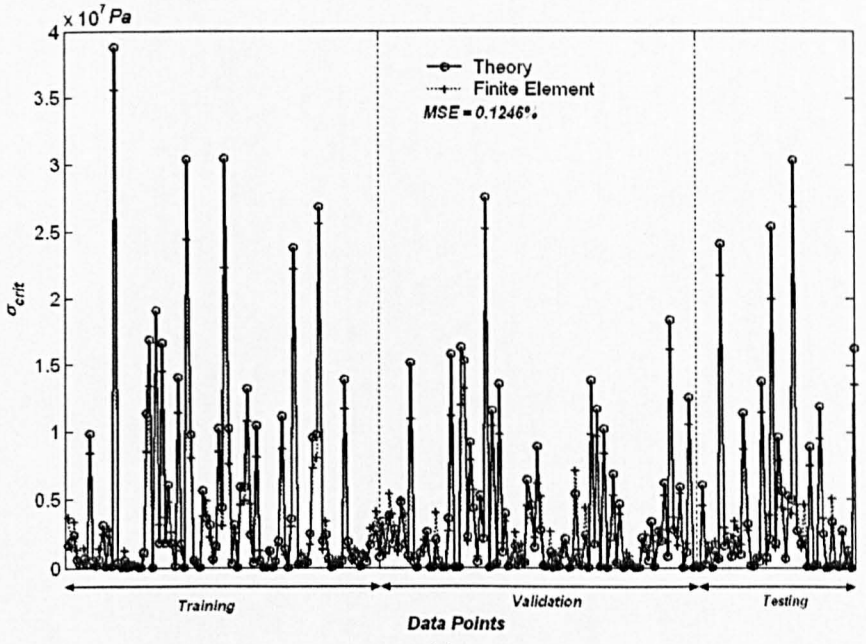


(c)

Figure 5.9 Comparison between FE results and theory. (a) E_1 (b) ν_{12} (c) E_2 (d) ν_{21} (e) buckling stress. (cont.)



(d)



(e)

Figure 5.9 Comparison between FE results and theory. (a) E_1 (b) ν_{12} (c) E_2 (d) ν_{21} (e) buckling stress.

Also, from Figure 5.5 - Figure 5.6 and Figure 5.8, the FE models show a reasonable relationship between the geometrical configurations and the structural properties, and agree well with the theory. The in-plane structural moduli showed a larger dependency towards the density (β) than the aspect ratio (α); the in-plane Poisson's ratios showed a strong relationship with the aspect ratio and the density did not affect much the results; while the buckling depending very much on both the density and the aspect ratio of the honeycomb. It is also worth reminding that the

angle (θ) contributes the most exciting behaviour of the honeycomb by introducing the auxeticity into the regular hexagonal honeycomb.

All the above-mentioned FE models require very high computational effort; an average time taken for one FE run to obtain a set of in-plane properties (E_1, E_2, ν_{12}, ν_{21}) is roughly 20s CPU time, and 12s CPU time for one FE buckling analysis. This is not desirable for making the FEA as a subroutine in an optimisation process. Therefore metamodels were introduced here in the hope of obtaining a proper representation of the honeycomb design domain, and reduce the time consumption of the computation for the relationship between the honeycomb geometrical parameters and its mechanical properties.

ii. Metamodels for hexagonal honeycomb

There were 250 FE generated models based on the uniformly distributed random geometrical parameters, α, β and θ, and these data were used to create metamodels from the MLP and GP. The range of each geometrical parameter was: $1 \leq \alpha \leq 6$; $0.05 \leq \beta \leq 0.2$; $-30° \leq \theta \leq 60°$ and $\gamma = 1$. The data were separated into training, validation and testing sets, each having 100, 100 and 50 points respectively. The MLP and GP surrogate models were then used toformulate a relationship between the honeycomb properties and its geometrical parameters with the same training procedures mentioned in the earlier chapters. Polynomials were also used to fit these models.

As mentioned in Chapter 3, equations (3.1) - (3.4) (the in-plane properties derived from theoretical formulas) contains several regions of NaN (Not a Number) caused by some combinations of mathematical functions that depend on the geometrical parameters. Though the FEA managed to calculate the properties of these difficult geometrical configurations, the errors are reasonably high. Similarly, for the buckling analysis, the theoretical formula gives extreme buckling resistance for some of the honeycomb geometry configurations. These areas are difficult to model with the FEA, and difficult to fit by the metamodels. This may be overcome by having more data points scattered around the difficult region to obtain a better 'zoom in' of these areas; however it is expensive to generate more FE models, especially at these regions.

The properties correspond to x_1 loading and buckling were difficult to fit, due to the reason mentioned above. The best configurations for every surrogate model representing each property are listed in the table below.

Surrogate Model	GP	MLP	Linear Polynomial	Quadratic Polynomial
E_1				
%MSE	6.33	72	60.51	56.36
Configurations	Generations: 1000	Hidden nodes: 4	No. of coef.: 10	No. of coef.: 55
	Population: 300	Learning and momentum regime: 5		
	Crossover rate: 0.1			
	Mutation rate: 0.4			
	No. of nodes: 50	No. of weights: 21		
	Tree Depth: 11			
ν_{12}				
%MSE	2.02	56	90.28	68.52
Configurations	Generations: 1000	Hidden nodes: 3	No. of coef.: 10	No. of coef.: 55
	Population: 300	Learning and momentum regime: 3		
	Crossover rate: 0.1			
	Mutation rate: 0.3			
	No. of nodes: 49	No. of weights: 16		
	Tree Depth: 13			

E_2				
%MSE	0.61	2.90	45.65	13.88
Configurations	Generations: 1000	Hidden nodes: 10	No. of coef.: 10	No. of coef.: 55
	Population: 300	Learning and momentum regime: 5		
	Crossover rate: 0.2			
	Mutation rate: 0.3			
	No. of nodes: 50	No. of weights: 51		
	Tree Depth: 12			
ν_{21}				
%MSE	0.30	1.1	18.35	4.86
Configurations	Generations: 1000	Hidden nodes: 3	No. of coef.: 10	No. of coef.: 55
	Population: 300	Learning and momentum regime: 3		
	Crossover rate: 0.1			
	Mutation rate: 0.4			
	No. of nodes: 50	No. of weights: 21		
	Tree Depth: 12			
σ_c				
%MSE	2.05	0.84	40.65	13.66
Configurations	Generations: 1000	Hidden nodes: 17	No. of coef.: 10	No. of coef.: 55
	Population: 300	Learning and momentum regime: 4		
	Crossover rate: 0.4			
	Mutation rate: 0.5			
	No. of nodes: 50	No. of weights: 86		
	Tree Depth: 17			

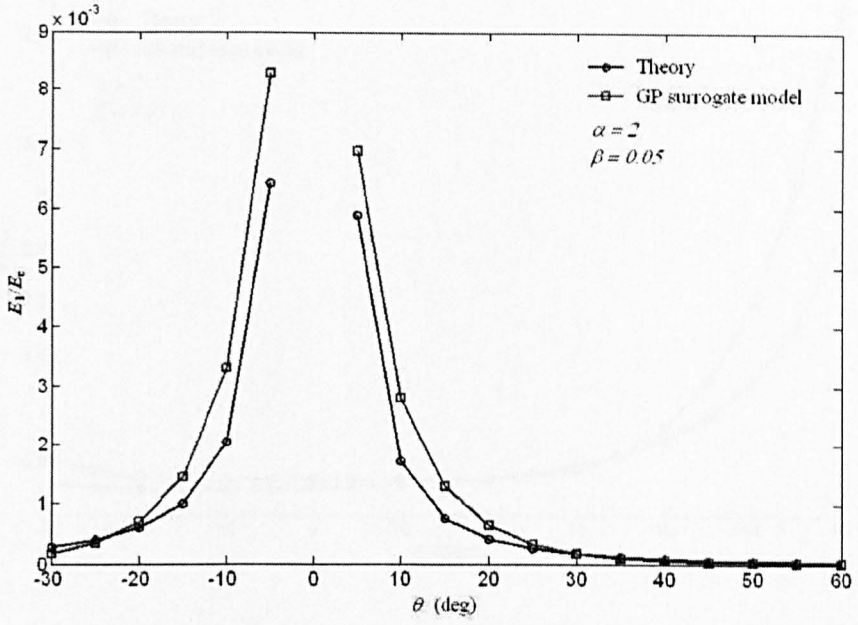
Table 5.1 Metamodel configurations for hexagonal honeycomb buckling stress.

Figure 5.10 also illustrates examples of the performance of the surrogate models in comparison with the theoretical models. In general, the polynomials failed to fit the data. And the MLP also failed to represent E_1 and ν_{12}. In the case for E_2 and σ_c, the MLP has too many weights to estimate for a given 250 data points.

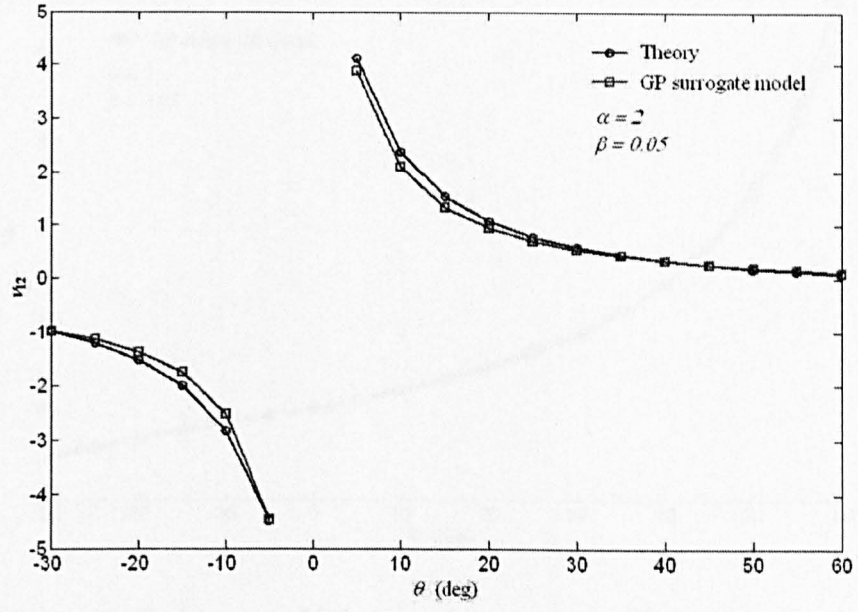
The GP too struggled to fit the models, though in general its performance is better compared to others. It maximises the use of allowable nodes, and all models have a big expression up until maximum of 17 levels, which is the σ_c model. The complications of finding the optimal weights in MLP is not encountered in GP, as the constants are generated randomly and no optimisation scheme has yet developed for these constants in this stage.

However, the GP doesn't seem to predict the buckling stress as well as the MLP, the MSE error is nearly doubled the MLP model. The MLP model as can be seen in Figure 5.10(e), managed to capture the trend of the buckling model, despite its vast number of weights, but fails to fit the difficult regions where high buckling stress occur.

Therefore, mixtures of metamodels are chosen to perform the shape optimisation routine. The GP models are used to predict the in-plane properties, and the MLP model is used to predict the buckling stress of the honeycomb.

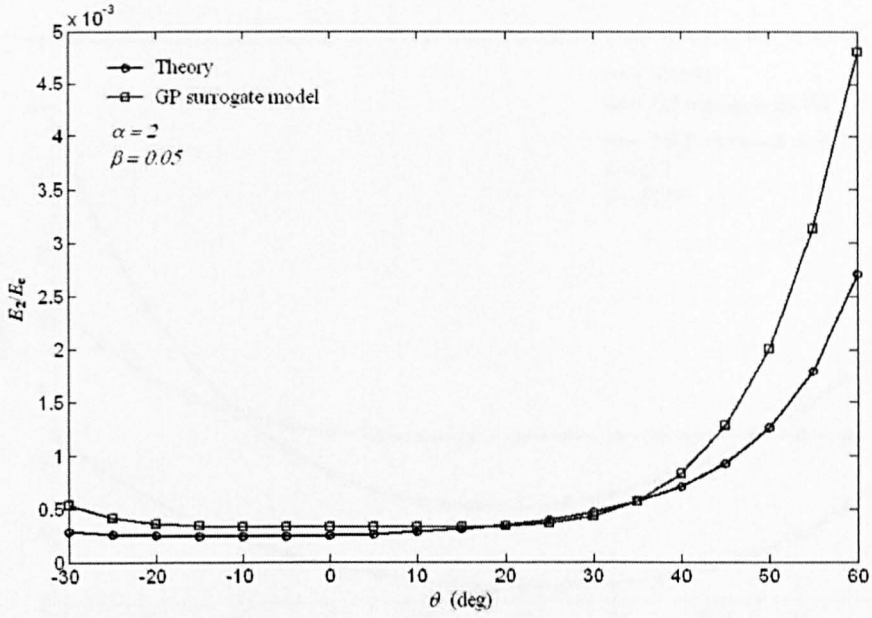


(a)

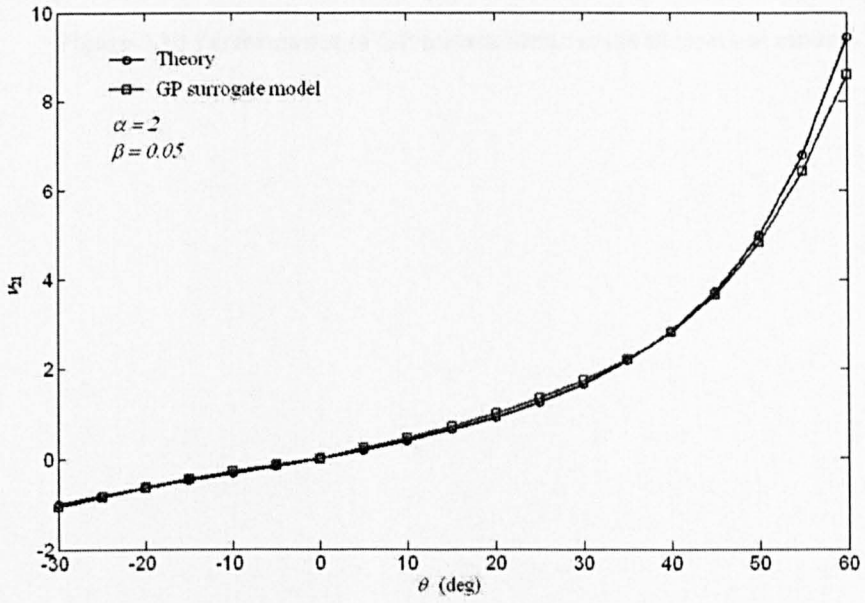


(b)

Figure 5.10 Performance of GP metamodels versus theoretical models. (cont.)

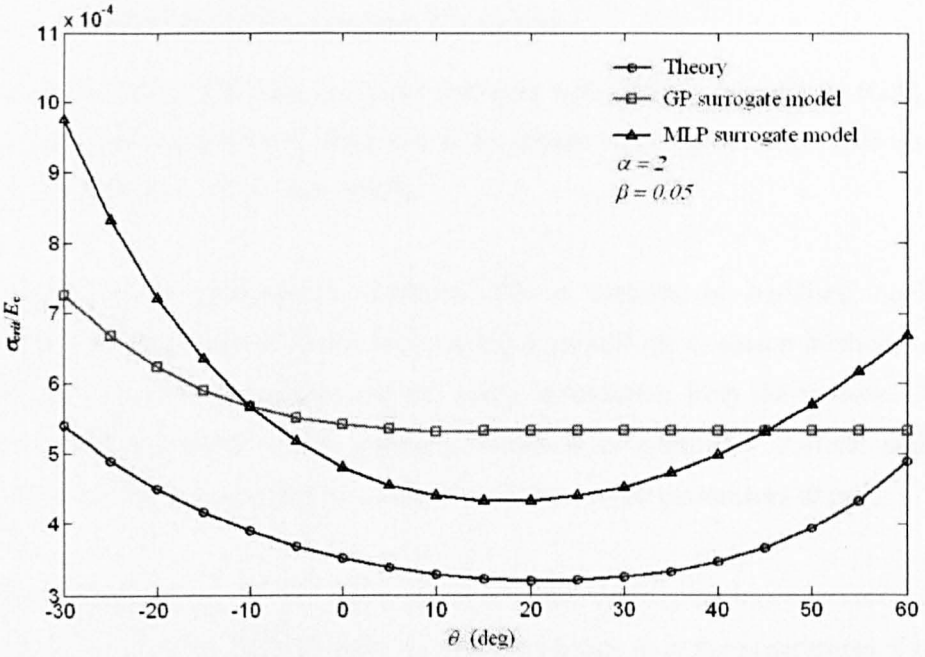


(c)



(d)

Figure 5.10 Performance of GP metamodels versus theoretical models. (cont.)



(e)

Figure 5.10 Performance of GP metamodels versus theoretical models.

iii. Differential Evolution Concept

Differential Evolution (DE) shares similar concepts with Genetic Algorithms (GA), except that it is for real value optimisation, while GA is for binary value optimisation. DE was developed by Storn and Price [71] in the late 1990's.

Unlike conventional optimisation methods, DE is capable of handling nonlinear, non-differentiable and multimodal functions, by using a parallel direct search method, i.e. searching in multi-directions simultaneously and not using information from derivatives. Direct search involves a search strategy which generates variation of parameter vectors, and a decision making process which is whether to accept the newly derived parameters or not.

For each generation, G, in DE, there is an m-dimensional population vector containing n-individuals. The number of individuals, n, does not change over the generations. Each of the m-th dimensional population represents the mth-parameter that is of interest to the user.

$$\mathbf{x}_{i,G}^{j} = \begin{Bmatrix} x_{1,G}^{1}, x_{2,G}^{1}, \cdots, x_{n,G}^{1} \\ x_{1,G}^{2}, x_{2,G}^{2}, \cdots, x_{n,G}^{2} \\ \vdots \\ x_{1,G}^{m}, x_{2,G}^{m}, \ldots, x_{n,G}^{m} \end{Bmatrix}, \qquad \textbf{(5.5)}$$

$$i = 1, 2, \cdots, n$$

$$j = 1, 2, \cdots, m$$

Like all other inherently parallel search techniques, the initial populations are generated randomly and should give a good coverage of the entire parameter space. A Gaussian distribution is used here.

For each m-dimensional target vector at current generation, $\mathbf{x}_{i,G} = \left(x_{i,G}^{1}, x_{i,G}^{2}, \cdots, x_{i,G}^{m}\right)$, a mutated vector is generated:

$$\mathbf{v}_{i,G+1} = \mathbf{x}_{r_1,G} + F \cdot \left(\mathbf{x}_{r_2,G} - \mathbf{x}_{r_3,G}\right) \qquad \textbf{(5.6)}$$

where $\mathbf{x}_{r_1,G}, \mathbf{x}_{r_2,G}, \mathbf{x}_{r_3,G}$ are m-dimensional vectors randomly chosen from the population, n, within the current generation G. The total number in the population, n, must be more than 4 in order to have the mutation take effect. F is a real constant between the range of 0 and 2, and F must be greater than 0. It gives weight to the differential variation $\left(\mathbf{x}_{r_2,G} - \mathbf{x}_{r_3,G}\right)$.

A crossover process is introduced to increase the diversity of the perturbed parameter vectors. The crossover produces an m-dimensional trial vector:

$$\mathbf{u}_{i,G+1} = \left(u^1_{i,G+1}, u^2_{i,G+1}, \cdots, u^m_{i,G+1}\right) \tag{5.7}$$

The elements of the trial vector are chosen based on:

$$u^j_{i,G+1} = \begin{cases} v^j_{i,G+1} & \text{if } \left(randb(j) \le CR\right) \text{ or } j = rnbr(i) \\ x^j_{i,G} & \text{if } \left(randb(j) > CR\right) \text{ and } j \ne rnbr(i) \end{cases}, \quad j = 1, 2, \cdots, m \tag{5.8}$$

Equations (5.7)-(5.8) states that the trial vector consists of the elements from both mutated vectors and original target vectors ($x^j_{i,G}$). The choice is made based on the *randb(j)*: for each element of the trial vector ($u^j_{i,G+1}$), a random number *randb* is generated, it should be between 0 and 1. If the *randb* is greater than the user defined crossover ratio, $CR \in [0, 1]$, the element $u^j_{i,G+1}$ will be taken from the original target vector at current generation; else, it will be taken from the mutated vector. Higher CR will have more chances of taking the elements from original target vector and vice versa. The *rnbr(i)* is randomly generated between 1 and m, to ensure that at least one element of the trial vector is taken from the mutated vector.

The trial vector $\mathbf{u}_{i,G+1}$ will then be compared with the target vector $\mathbf{x}_{i,G}$ in order to decide if it is fit enough to become a member of the next generation. If the trial vector gives smaller error than the target vector, $\mathbf{x}_{i,G+1}$ is replaced by $\mathbf{u}_{i,G+1}$; otherwise $\mathbf{x}_{i,G}$ will remain in the population.

The above-mentioned procedures are repeated until all the individuals of the current generation have been chosen as a target vector, then the algorithm restarts with the next generation. The algorithm is terminated when the fitness or error converges to the predefined threshold, or the maximum number of generation is reached.

iv. Shape Optimisation Case Study for Honeycomb

The shape optimisation proposed here takes the desired material properties as the target vector, and the DE algorithm determines the corresponding geometrical parameters of the hexagonal honeycomb.

Some of the structural properties of the honeycomb may be achieved by several different geometrical parameters. For example, hexagon honeycombs with $\alpha = 1$, $\beta = 0.04$, $-25° < \theta < -20°$; $\alpha = 1$, $\beta = 0.05$, $-10° < \theta < -5°$ and $\alpha = 1$, $\beta = 0.05$, $50° < \theta < 55°$, all gives a relative / normalised critical buckling stress close to 9.3×10^{-4}. Therefore constraints are introduced to limit the search region into the auxetic region ($\theta < 0°$) and non-auxetic region ($\theta \geq 0°$), in order to avoid non-unique optimisation solution.

The objective function of this shape optimisation is the summation of the MSE from each structural property:

$$MSE = \sum_{i=1}^{a} \frac{(x_i - \hat{x}_i)^2}{x_i^2} \tag{5.9}$$

where a is the length of the target vector, in other words, it is the number of desired structural properties; x_i and $\hat{x}_i$ are the target structural property and the DE estimated structural property respectively. In this case, there are five elements (i.e. $a = 5$) in the target vector, i.e. E_1, E_2, ν_{12}, ν_{21}, σ_{crit}.

According to [71], the population is best chosen to be 5 to 10 times the number of dimensions, and a high crossover ratio can speed the convergence rate, while scaling factor around 0.5 is also recommended. The parameters used in this work were set heuristically, there were 20 generations with 30 populations, each running for 100 times. The scale factor is 0.6 and crossover ratio is 0.8.

A penalty was also introduced for geometrical parameters that fall outside the design domain, it can be seen that the penalty increases quadratically as the data deviates further away from the feasible region [42]:

$$\text{Penalty}(x_i) = \begin{cases} 10^3 \left(\dfrac{x_{\min} - x_i}{x_{\min}} \right)^2, & \text{for } x_i < x_{\min} \\ 0, & \text{for } x_{\min} \leq x_i \leq x_{\max} \\ 10^3 \left(\dfrac{x_{\max} - x_i}{x_{\max}} \right)^2, & \text{for } x_i > x_{\max} \end{cases} \tag{5.10}$$

The penalty was added into the MSE values, the constant of 1000 was found to be efficient by trial and error.

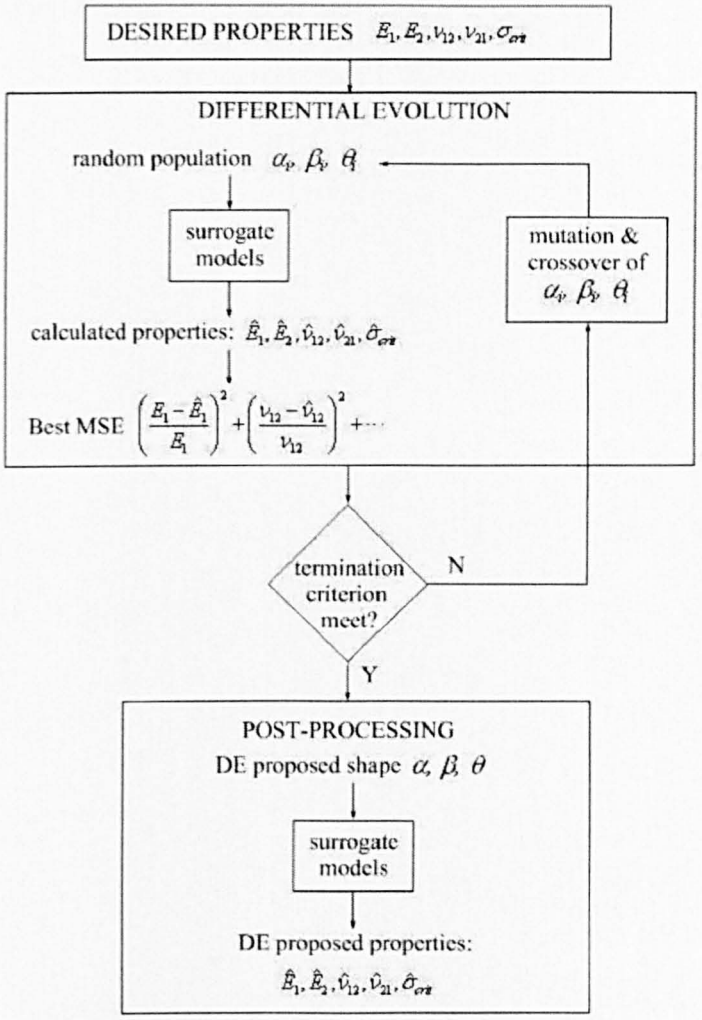


Figure 5.11 The shape optimisation flowchart.

As shown in Figure 5.11, the shape optimisation starts with the desired structural properties, which were assigned by the user as: E_1 = 1.0e8Pa, E_2 = 3.0e7Pa, ν_{12} = -2, ν_{21} = -0.5, σ_{crit} = 5.0e7Pa, which roughly corresponds to the geometrical parameters $\alpha = 2$; $\beta = 0.006$; $\theta = -16°$ and core material properties were E_s = 7e10Pa, ν_s = 0.33. These structural values were generated by inserting the arbitrary geometrical parameters (α,β,θ) into the theoretical formulae that determine the structural properties of the honeycomb.

The shape optimisation was performed by taking these structural properties as the target vector. The DE optimisation first generates a random population that contains all the geometrical parameters within the design range, each of these geometrical parameters were fed into the surrogate models (GP for in-plane properties, MLP for critical buckling stress) to evaluate for its corresponding DE estimated structural properties. These estimated structural properties were

compared to the desired ones, and thereby calculating its percentage of MSE. The penalty is applied here for those geometrical parameters that fall outside the range. Mutation and crossover are performed within the population in the current generation. The performance measure among these offspring and the original geometrical parameters then takes place, those that out-perform their competitors will remain in the population for the next generation.

As the termination criterion, such as the number of generations or number of runs is reached, the optimisation process will terminate with the suggested geometrical parameters that give the lowest MSE. A simple post processing was then used to evaluate the corresponding structural properties of the honeycomb that has these suggested geometrical parameters. The purpose is to see if the surrogate models, despite their errors in representing the FEA data, can still give good estimation of the geometrical parameters.

Figure 5.12 shows an example of the final structure of the honeycomb. The result of the DE shape optimisation has a MSE of 2.081%. The geometrical parameters suggested by the surrogate models are listed in Table 5.2.

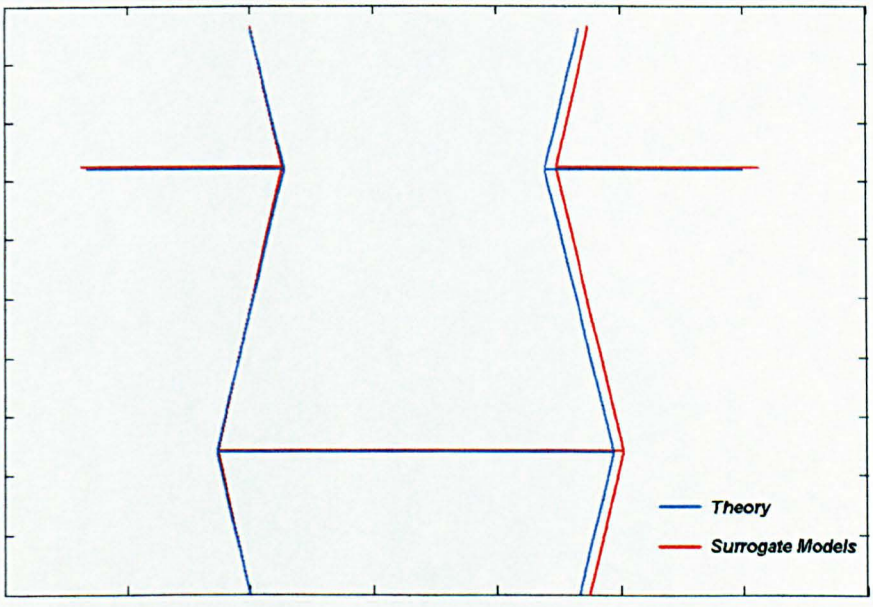


Figure 5.12 Shape optimisation of hexagonal honeycomb.

	Desired Value	Surrogate Model
α	2	1.642
β	0.006	0.006
θ	-16°	-15.4°
E_1	1e8 Pa	1.55e8 Pa
E_2	3e7 Pa	2.97e7 Pa
ν_{12}	-2	-2.007
ν_{21}	-0.5	-0.351
σ_{crit}	5e7 Pa	8.94e7 Pa
MSE	N/A	2.081%

Table 5.2 Shape optimised from Theoretical model and surrogate model compared to the desired values.

The table above shows that the surrogate models have given a very good estimation of the honeycomb geometry, except α, which leads to the error for the critical buckling stress and ν_{21}, both sensitive towards the aspect ratio of the honeycomb.

It is worth mentioning again that an accurate surrogate model is important, as the error of the surrogate model, which is inherent from the FEA, will propagate and be magnified through the optimisation process.

6. Chiral Honeycomb

The chiral structural honeycombs were first shown in [54, 82]. The term 'chiral' in physics means that it is impossible to superimpose the structure on its mirror image [31], or it cannot be mapped to its mirror images by rotation and translation alone. The word 'chiral' is derived from the Greek χειρ (cheir), meaning the 'hand', which is the most common chiral structure. There are many familiar objects that are chiral, such as our left and right hands, b to d or p (d and p are the same, except for orientation), helix structures etc [32].

Figure 6.1(a) shows a conceptual layout of the chiral honeycomb unit cell, the mirror image of the original layout is presented in Figure 6.1(b), which is not identical to the original layout. It is impossible to replicate Figure 6.1(b) from Figure 6.1(a) by simple rotation or translation alone. However, by making Figure 6.1(a) infinite in all directions, the pattern becomes achiral (i.e. able to create an identical mirror image), but without a line of symmetry or centre of symmetry, an example is illustrated in Figure 6.1(c) [32].

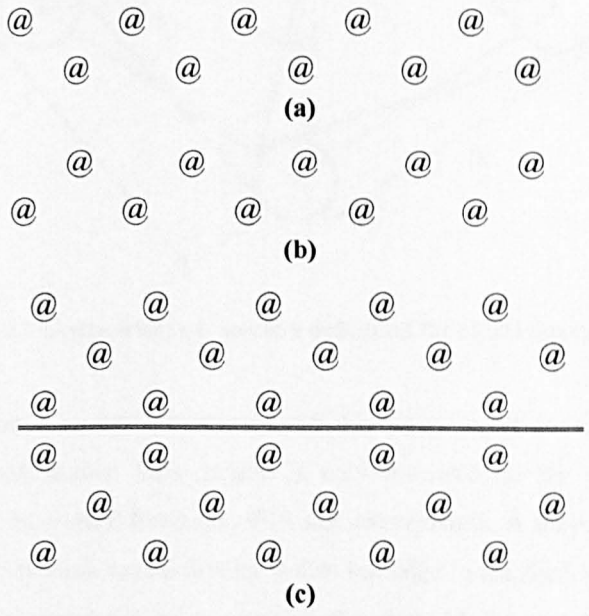


Figure 6.1 A basic layout of (a) chiral honeycomb (b) its mirror image and (c) more replications of the chiral honeycomb (above the double line) with its mirror image (below the double lines).

Though this infinite honeycomb domain no longer maintains its chirality (unable to superimpose the structure on its mirror image), the name 'chiral honeycomb' is retained to celebrate the unique pattern (without a line of symmetry) of such a honeycomb. Unlike the cellular structure in the previous chapter, the chiral honeycomb is a non-centresymmetric cellular structure; it does not have symmetry lines along the centre axes. The replications of the chiral honeycomb will be discussed in further details in the next few sections.

The parameters that determine the chiral honeycomb are shown in Figure 6.2. The circular elements / nodes of equal radius r form the basis of the honeycomb. The distance between the centres of these nodes is indicated as R. The angle between the horizontal axis and the centre distance of two nodes is denoted by θ. The tangent ligament connecting the nodes has length L. The intersection angle between the tangent lines and the centre distance of the nodes is φ. The ligaments and nodes have a uniform wall thickness, t. In order to use a consistent nomenclature with the Cellular Material Theory (CMT) [23], the cell aspect ratio, density and gauge thickness are defined as $\alpha\,(L/r)$, $\beta\,(t/r)$ and $\gamma\,(h/L)$ respectively.

This chiral structure has some outstanding in – plane and out of plane properties compared to other auxetic structures, e.g. the re-entrant hexagonal honeycomb. According to [54], the chiral honeycomb with $\theta = 30°$ has an in-plane Poisson's ratio of nearly –1, and exhibits an in-plane isotropic behaviour.

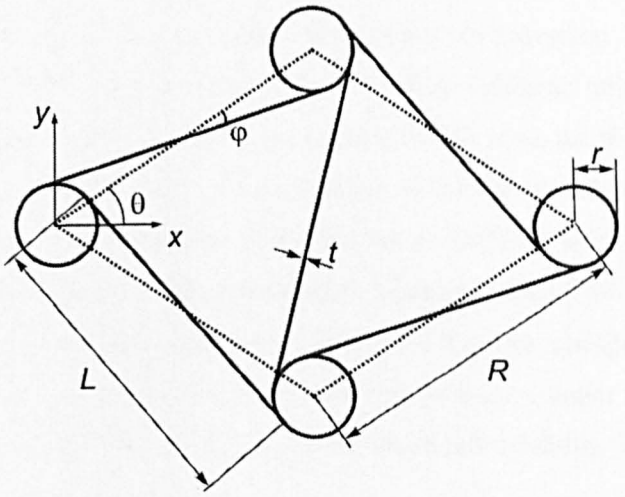


Figure 6.2 Geometrical parameters definition for chiral honeycomb.

This chapter is dedicated to the investigation of this novel structure, and to looking at the possible sandwich applications. This section is only focussed on the out-of-plane property characterisation using theoretical formulae, FEA and experiments. A shape optimisation routine based on the chiral honeycomb modal density is also included in the final section to demonstrate the effectiveness of the metamodels in representing the chiral FE simulated properties.

i. Homogenisation Development and Simulations

There are not as many varieties of unit cells for chiral honeycombs compared to the centresymmetric hexagonal honeycomb, as the chiral honeycomb does not have a centresymmetric duplication manner. However, it is not impossible to choose a unit cell that can represent the chiral domain properly. Figure 6.3 shows four types of unit cell configurations. By looking at the apparent area enclosed by the thin blue dotted lines, cell (a), (c) and (d) can tessellate in a normal horizontal and vertical directions, without overlapping any part of the unit cell. In fact, the ratio between cell (c) and cell (d) with cell (a) is 1/3 and 1/6 respectively. For cell (b), though the tessellation manner remains non-centresymmetric, it is a reduced model of the cell (a), due to its chiral symmetry. The ratio between cell (a) and (b) is 2/3.

Another exciting feature of the chiral honeycomb is that different parts of the unit cell have their own strength, for example, the cylinders have better compression resistance; while the ligaments are more flexible under shear [50]. Therefore different unit cells were used for different analysis. Figure 6.3(b) and (c) were adopted in this work for their simplicity. Cell (c) was used to model out of plane shear modulus (G_{xz}), as the shear loading is mostly carried by the ligaments. A proper representation of the ligaments (without splitting them in half) can provide a better understanding of the deformation characteristics of the ligaments. Cell (b) is used to model the out-of-plane compressive strength (σ_{xz}). As cylinders carry most of the compression, a FE representation of a full circle will provide a better insight into the stress distribution in the cylinder. For the out of plane structural modulus, E_z, both models show similar results.

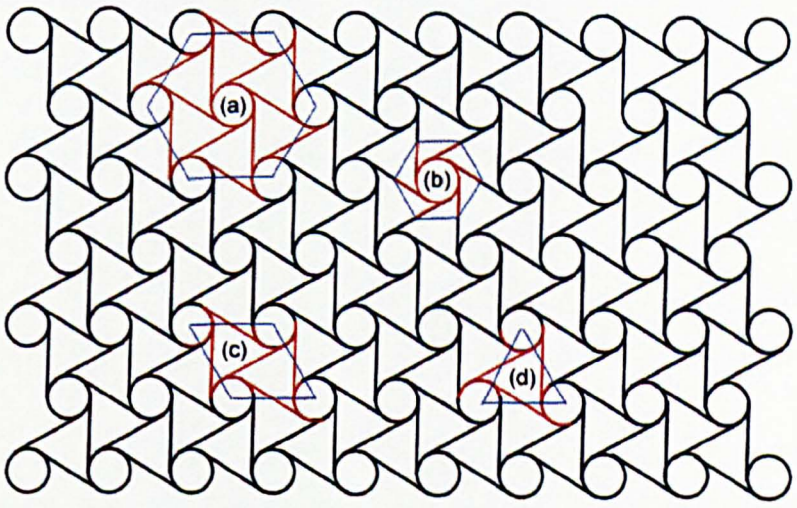


Figure 6.3 Possible unit cells that represent the chiral honeycomb domain.

The cross sectional areas of both Figure 6.3(b) and (c) have slight differences. However, the boundary condition for the chiral honeycomb is the same as before, whereby the loads were applied at the top of the structure, and constraints were applied at the bottom of the structure.

The free ends of the structures were linked by constraint equations describing their relative movements. The element used for all chiral models is shell93 [6]. The constraints and cross sectional area used for different studies will be discussed in more detail in the following sections.

ii. Analytical derivations

The chiral honeycomb is still a very new structure in auxetic material / structural studies, therefore there is not much theoretical documentation available, and it is mentioned in this chapter for the reader to gain some understanding towards the chiral honeycomb structure. The reader is also suggested to refer to the originals of the derivations from [65, 50, 64, 69, 53]. FE model setups for E_z, G_{xz} and σ_c are included here. The results obtained from the analytical derivations and the FEA will be compared with the experiments in the next section.

a. Elastic Modulus

The relationship between the Young's modulus of the core material (E_c) to the out of plane elastic modulus (E_z) of the chiral honeycomb, can be described as the ratio between the load bearing and the total cross sectional area of the structure. This relationship is also equivalent to the relative structural density (ρ/ρ_c) of the honeycomb [23].

$$\frac{E_z}{E_c} = \frac{\rho}{\rho_c} = \frac{\text{Total load bearing area}}{\text{Total cross section area}} \tag{6.1}$$

As the FE unit cell was built from Figure 6.3(c), the calculations of the chiral honeycomb areas mentioned above are also based on the same model. In Figure 6.4, the area covered by dotted lines indicates the total cross sectional area, which is made of one whole circle with two triangles [50].

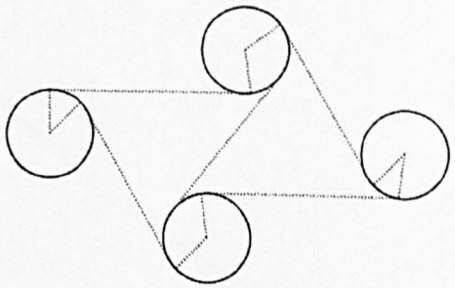


Figure 6.4 The total load bearing area and total cross section area of the chiral honeycomb unit cell.

$$\mathrm{A}_{\text{total\_cross\_section}} = 2\times \mathrm{A}_{\text{triangle}} + \mathrm{A}_{\text{circle}} = 2\times\frac{1}{2}\times L^2\cos\theta + \pi r^2 \tag{6.2}$$

Some concerns regarding the derivation of the area of the triangles are illustrated in (Appendix 2) [49, 50].

The load bearing area is represented by the ligaments and the circumference of the circle marked by the dotted lines. The four ligaments' areas at the edge of the unit cell are halved as the thickness is shared among adjacent unit cells. It is worth noticing that the above statement does not indicate that the ligaments were halved in the FE models.

$$A_{tot\_load\_bearing} = 2\pi rt + Lt + \frac{4Lt}{2} = (2\pi r + 3L)t \tag{6.3}$$

As $\theta = 30°$, applying the dimensionless parameters into (6.2) and (6.3), the empirical formula for E_z can be written as:

$$\frac{E_z}{E_c} = \frac{\rho}{\rho_c} = \frac{2\beta(2\pi + 3\alpha)}{\sqrt{3}\alpha^2 + 2\pi} \tag{6.4}$$

This calculation is based on low density structures, with $t << l$ or $\beta << 1$, in this case, $\beta = 0.22$, measured from the experimental specimen. Equation (6.4) shows some similarities with the theoretical formulae for hexagonal honeycombs in [23], as the structural modulus and the relative density (ρ/ρ_c) of the honeycomb scales linearly with the density β. There is also an inverse relationship between the aspect ratio α and the relative density (ρ/ρ_c)/β, Figure 6.5 gives an example obtained from a FE model. Though with these similarities, it is important to remember that the definitions of these dimensionless parameters (α, β, γ) are not similar to the hexagonal honeycombs. The similarities between the two honeycomb models only showed the consistency of equation (6.4).

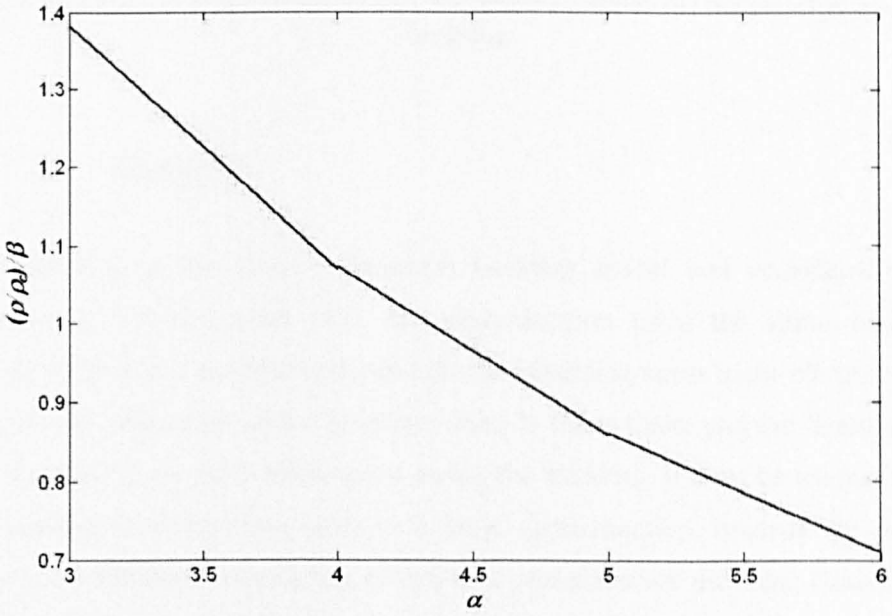


Figure 6.5 An inverse relationship between the relative density of the chiral honeycomb and its aspect ratio.

The transverse structural modulus of the chiral honeycomb was also estimated using FEA. The FE model was built using 8-node 6 DoF shell elements, shell93 [6]. An example of the unit cell geometrical parameters were modelled based on the experimental specimen, where $\alpha = 4.44$, $\beta = 0.22$, $\gamma = 1$ and $\theta = 30°$, and consists of 1625 elements. Loads were applied on the top surface

as a uniform displacement field, all movements in the lower surface were constrained and constraint equations were also applied on the free ends to constrain the out of plane rotation (Figure 6.6). The transverse structural modulus was calculated by averaging the reaction forces over the cross-sectional area of the unit cell, and dividing by the imposed strain.

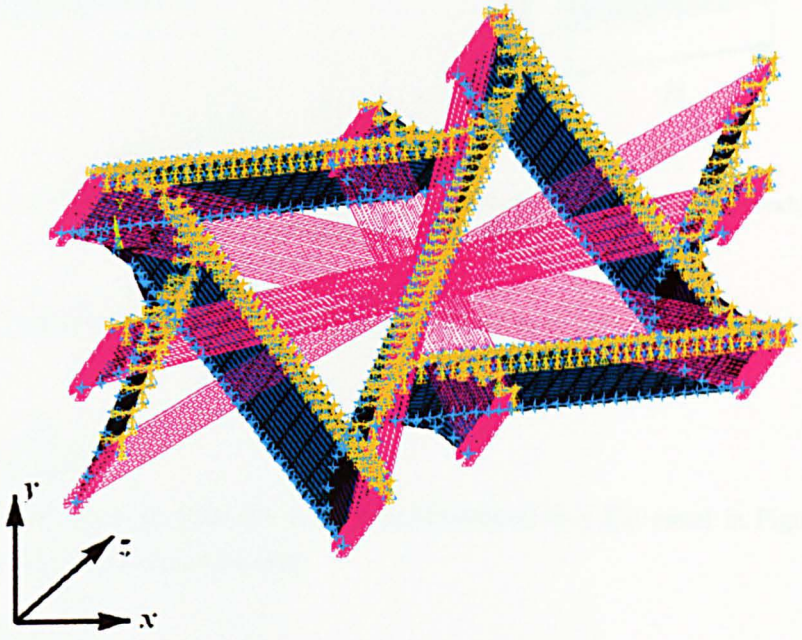
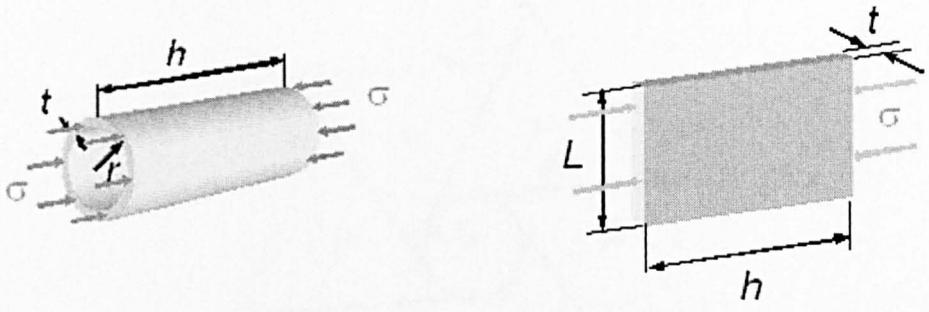


Figure 6.6 Chiral honeycomb FE model with boundary condition for the out of plane structural modulus.

b. Buckling

The approximation of the chiral honeycomb buckling model was considered as a linear superposition of buckling loads [64], the approximation takes the chiral unit cell as a combination of cylinders and ligaments, and the total buckling stress in the chiral honeycomb is equivalent to the summation of the buckling stress in the cylinder and the ligament, it is also assumed that there is no shear deformation during the buckling. It must be emphasized that the linear superposition of buckling mode is a basic approximation, because the mode shapes associated to the fundamental buckling of cylinders and plates are different. However, for long aspect ratio cylinders (i.e. tending to become slender hollow beams), the eigenmodes associated to fundamental buckling in both structural elements tend to converge. It must be noted however, that the approximaton put in place provides a good convergence with the experimental results, with the cylinder contributes the most to the compressive strength.



(a) (b)

Figure 6.7 (a) A cylinder under compression; (b) A ligament of the chiral honeycomb, viewed as a panel, under compression.

The critical elastic buckling stress for a cylinder with radius r (Figure 6.7(a)), and wall thickness of t is [84]:

$$\sigma_{cyl} = \frac{1}{\sqrt{3}} \frac{E_c}{\sqrt{1-\nu_c^2}} \left(\frac{t}{r}\right) \tag{6.5}$$

And the critical buckling stress for a ligament (illustrated as a flat panel in Figure 6.7(b)) of height L and wall thickness of t is [84]:

$$\sigma_{lig} = K \frac{E_c}{1-\nu_c^2} \left(\frac{t}{L}\right)^2 \tag{6.6}$$

where the constant K is dependent on the boundary conditions at the edge of the ligaments and the aspect ratio (h/L, from Figure 6.7(b)) of the ligament. For large aspect ratio ($h/L \geq 3$), typical values of K can range from 3.29 (the ligament edge is simply supported. In this case, the K value is also the same for h/L = 1, 2) to 5.73 (fully clamped edge) [84].

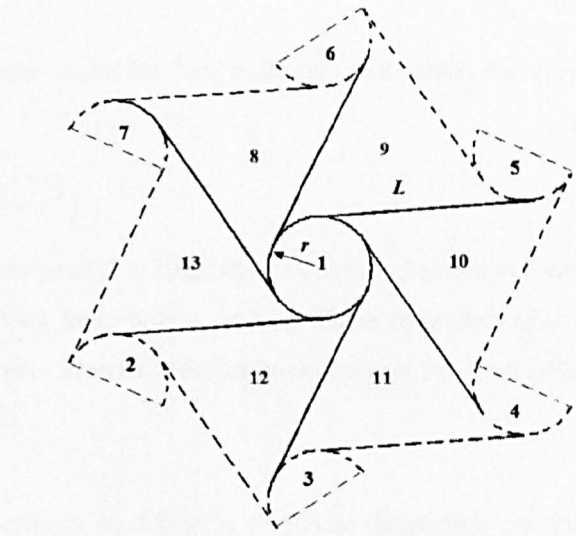
Combining (6.5) and (6.6), the elastic buckling force of the chiral honeycomb becomes:

$$F_{crit} = 2\pi r t \sigma_{cyl} + 6Lt\sigma_{lig} \tag{6.7}$$

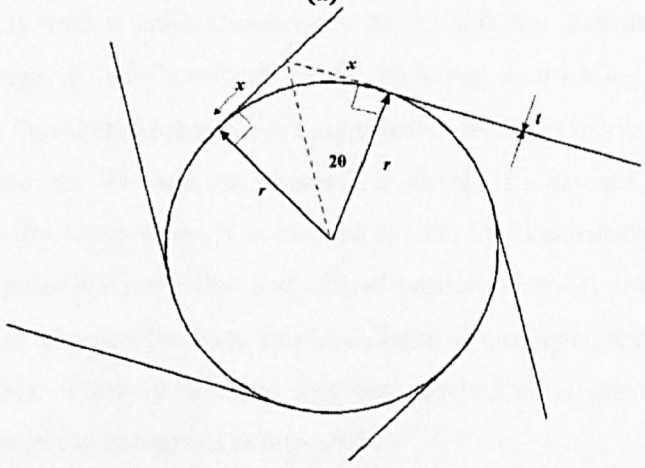
Therefore the critical buckling stress for the chiral honeycomb can be expressed as:

$$\sigma_{crit} = \frac{F_{crit}}{A_{total}} \tag{6.8}$$

where the A_{total} is the total cross sectional area and it is different from the previous model. The total area used here is from the full model (Figure 6.3(a)), and it is illustrated below [64]:



(a)



(b)

Figure 6.8 (a) Geometry of the chiral honeycomb used for the FE buckling analysis [64], (b) Magnified angles in the centre circle [64].

Figure 6.8(a) shows that the area of the hexagon consists of one full circle, 6 triangles and 6 semicircles. And due to its chiral symmetry, element 2 is contiguous to element 5, similarly elements 3 – 6 and 4 – 7. Six ligaments connect the centre circle to the outer ring of semicircles, forming an angle of 2θ ($\theta = 30°$) between two radii that joins two adjacent tangent points in the centre circle (Figure 6.8(b)). The total area of the unit cell is the summation of the area of the centre circle and all the external semicircles (element 1 – 7) and the area of each triangle (element 8 – 13) [64]:

$$A_{total} = \pi r^2 + 6\left(\frac{1}{2}\pi r^2\right) + 6\left(\frac{1}{2}L^2 \sin(2\theta)\right) = 4\pi r^2 + 3L^2 \sin(2\theta) \qquad \textbf{(6.9)}$$

Substituting (6.5) - (6.7), (6.9) into (6.8), the critical buckling stress of the chiral honeycomb becomes:

$$\frac{\sigma_{crit}}{E_c} = \frac{4\pi\beta^2}{\left(8\pi + 3\sqrt{3}\alpha^2\right)\sqrt{3\left(1-\nu_c^2\right)}} + \frac{12K\beta^3}{\alpha\left(1-\nu_c^2\right)\left(8\pi + 3\sqrt{3}\alpha^2\right)} \qquad \textbf{(6.10)}$$

Equation (6.10) is only valid for long cylinders that satisfy the condition $Z > 2.85$ [69], where Z is defined as:

$$Z = \left(\frac{h}{r}\right)^2 \left(\frac{r}{t}\right) \sqrt{\left(1-\nu_c^2\right)} \tag{6.11}$$

While long cylinders have $Z > 2.85$, short cylinders have lower values. The buckling modes are also different for long and short cylinders. Short cylinders give ring / symmetrical buckling, medium cylinders give chessboard-like buckling and the long cylinders buckle like any normal slender column [84].

The value of the critical buckling in (6.10) is dependent on the square of β. And only the ligaments contribute with a cubic dependency of β. This has indicated that in general, the cylinders' performances are dominant when applying a buckling force. However, the contribution of the ligaments becomes more significant when β is increased. Figure 6.9 shows a comparison between the FE and the theoretical model. In general, the theoretical model underestimates the buckling when β is beyond 0.125. The contribution of the cylinder and ligament was computed from the first and second portion of (6.10) respectively. On the other hand, the hexagonal honeycombs have elastic collapse stress dependent only on the cube of β [23, 50]. It is again, worth mentioning that the dimensionless geometrical parameters are different from those of the hexagonal honeycombs.

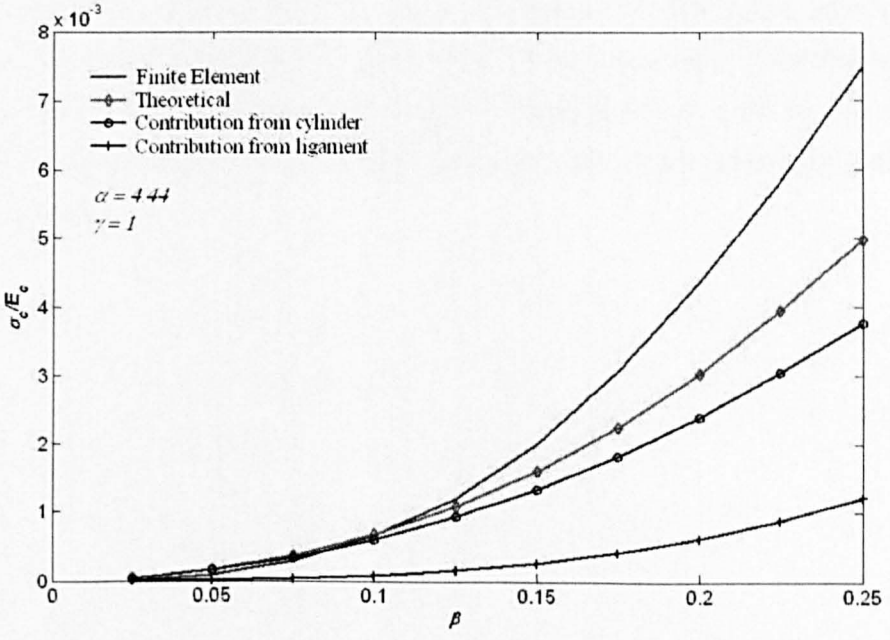


Figure 6.9 Comparison between theoretical, FE model for the buckling of the chiral honeycomb, include the contribution from the cylinder and the ligament.

An eigenvalue buckling analysis was carried out on the FE chiral honeycomb unit cell. The model is shown in the figure below, it is a reduced model from Figure 6.8, the ligaments were only modelled for half of their length.

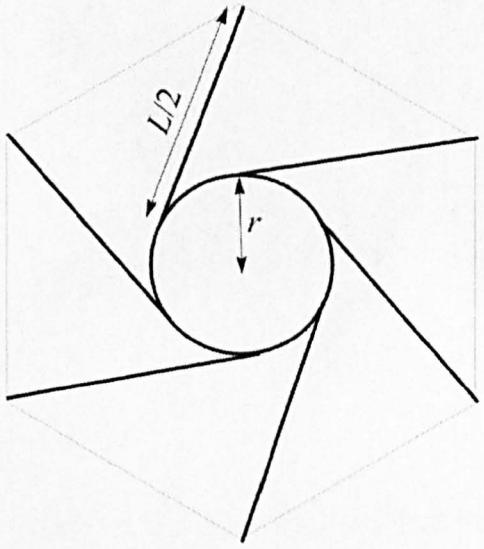


Figure 6.10 Unit cell used for FE modelling of the compressive strength.

This model was chosen based on the assumption that the global buckling produces symmetric and homogeneous deformation. According to [64], when considering the whole honeycomb, it is reasonable to assume that the ligaments are stiffeners, and that they would minimise the effect of local imperfections (e.g. eccentricity) in the cylinders alone (the gauge thickness, $\gamma = 1$). Furthermore the neighbouring effect of adjacent cells would stabilise the buckling behaviour of a single cylinder. As the cylinders carry most of the buckling forces, a full picture of the stress flow around the cylinder and half the length of the ligaments should capture the buckling mode of the whole domain.

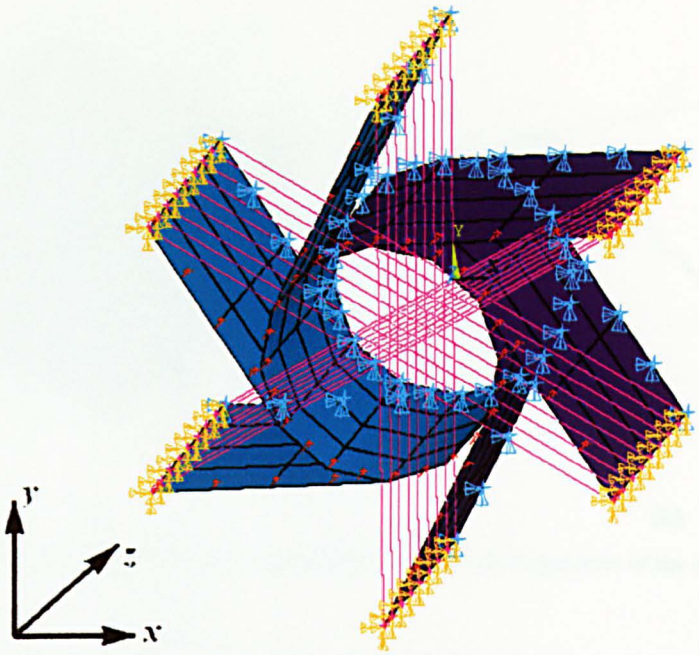


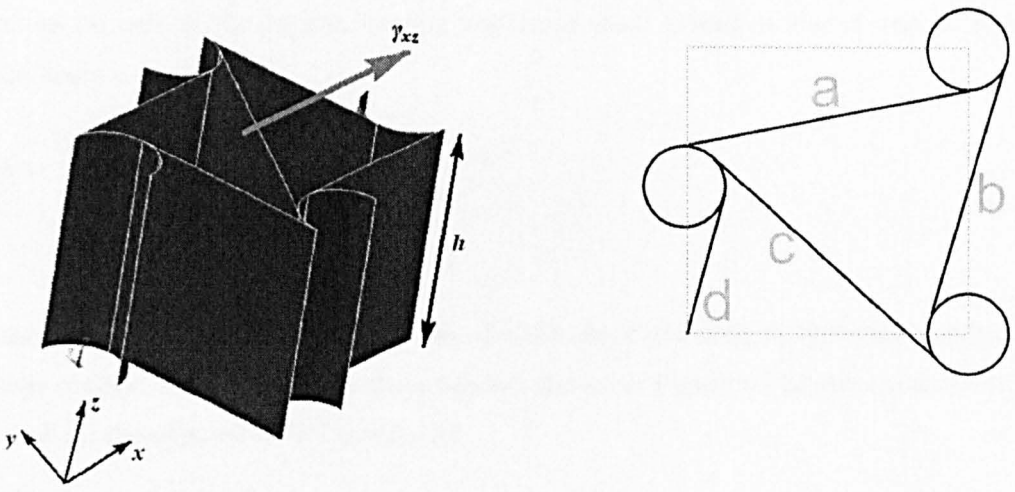
Figure 6.11 Boundary conditions of the chiral honeycomb used for the FE buckling analysis

A unit force was uniformly distributed on the top surface. A periodic boundary condition was applied as a series of constraint equations, describing the relative out-of-plane rotation of the contiguous ligaments. The in-plane rotations of the ligaments were prohibited. The bottom surface is simply supported (only constraining the translational DoFs).

The buckling force obtained from the FE buckling analysis was divided by the total area of the unit cell - in this case, the area of the hexagon - for the critical buckling stress.

c. Shear Modulus

The computation of the shear modulus, G_{xz}, for the chiral honeycomb is relatively sophisticated comparing to the previous two properties. As until today, there are still limits to where the analytical solution can be stretched, the theoretical analysis for the chiral honeycomb out-of-plane shear modulus is still determined by upper and lower bounds. The exact solution of the shear modulus relies on the numerical method, and is later verified by experiment results.



(a) (b)

Figure 6.12 (a) Shear stress imposed in *xz* plane. (b) Ligaments of the chiral cell.

The upper bound of the chiral honeycomb is calculated using the minimum potential energy [23, 50]. As shown in Figure 6.12(a), a shear stress τ_{xz} causing a deformation γ_{xz} is acting on the face of the unit cell in the x-direction. The bending is neglected, as almost all the elastic strain energy is stored in cell walls during the shear displacements. It is also assumed that the cylinders do not carry any shear or bending during the deformation. Therefore the shear strain in wall a, b and c (Figure 6.12(b)) can be written as:

$$\begin{aligned} \gamma_a &= \gamma_{xz}\cos(\theta-\varphi) \\ \gamma_b &= \gamma_{xz}\cos(\theta+\varphi) \\ \gamma_c &= \gamma_{xz}\sin\varphi \end{aligned} \tag{6.12}$$

The expression of the minimum potential energy has the form:

$$\frac{1}{2}G_{xz}\gamma_{xz}^2 V \le \frac{1}{2}\sum_i \left(G_c\gamma_i^2 V_i\right) \tag{6.13}$$

where G_c is the shear modulus of the core material, γ_i is the shear strain in each of the wall of volume V_i. Substituting (6.12) into (6.13), and with some manipulations (Appendix 3), the equation above, with non-dimensional terms, becomes:

$$\frac{G_{xz}}{G_c} \le \frac{\alpha\beta}{\alpha^2\cos\theta+\pi}\left[\cos^2(\theta-\varphi)+\cos^2(\theta+\varphi)+\sin^2\varphi\right] \tag{6.14}$$

Similarly, for the upper bound of G_{yz}:

$$\frac{G_{yz}}{G_c} \le \frac{\alpha\beta}{\alpha^2\cos\theta+\pi}\left[\sin^2(\theta-\varphi)+\sin^2(\theta+\varphi)+\cos^2\varphi\right] \tag{6.15}$$

Both (6.14) and (6.15) suggest that the transverse shear moduli in the *xz* and *yz* planes equal each other:

$$\frac{G_{xz}}{G_c}=\frac{G_{yz}}{G_c}=\frac{G_z}{G_c}\le\frac{3}{2}\frac{\alpha\beta}{\alpha^2\cos\theta+\pi} \tag{6.16}$$

In the case of the lower bound of the out of plane shear modulus, the minimum complementary energy method is used. With the shear loading shown in Figure 6.12a, the components of shear force F for the unit cell are (Figure 6.13):

$$\begin{aligned}\tan(\theta-\beta)&=\frac{F_{ay}}{F_{ax}}\\ \tan(\theta+\beta)&=\frac{F_{by}}{F_{bx}}\\ \tan\beta&=\frac{F_{cx}}{F_{cy}}\end{aligned} \tag{6.17}$$

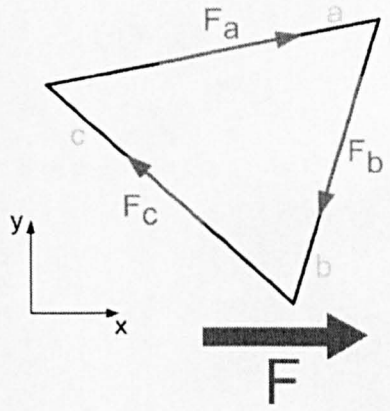
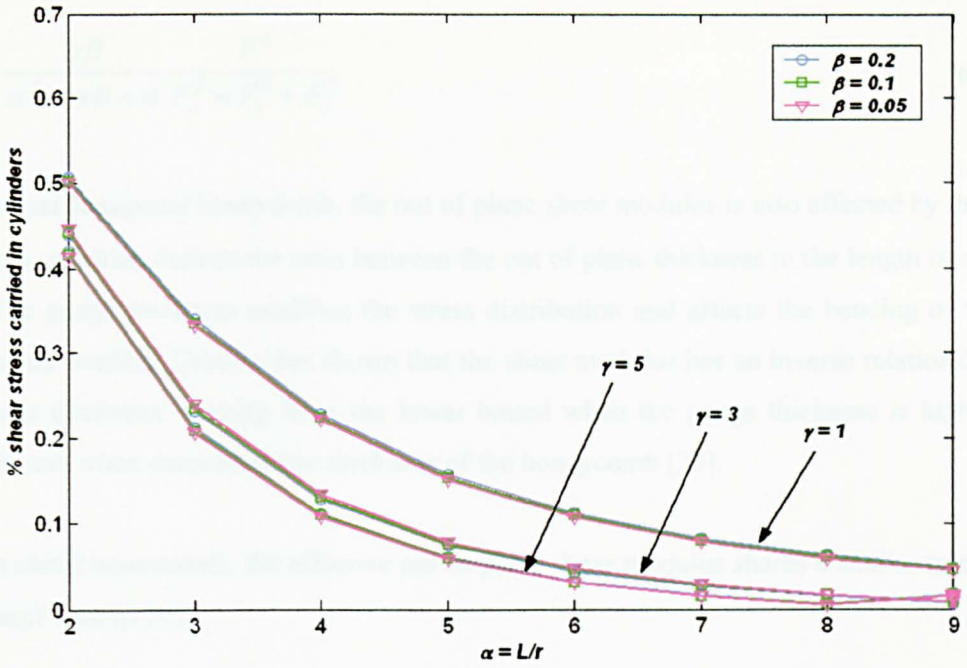


Figure 6.13 Equilibrium of shear forces in the ligaments of the chiral unit cell.

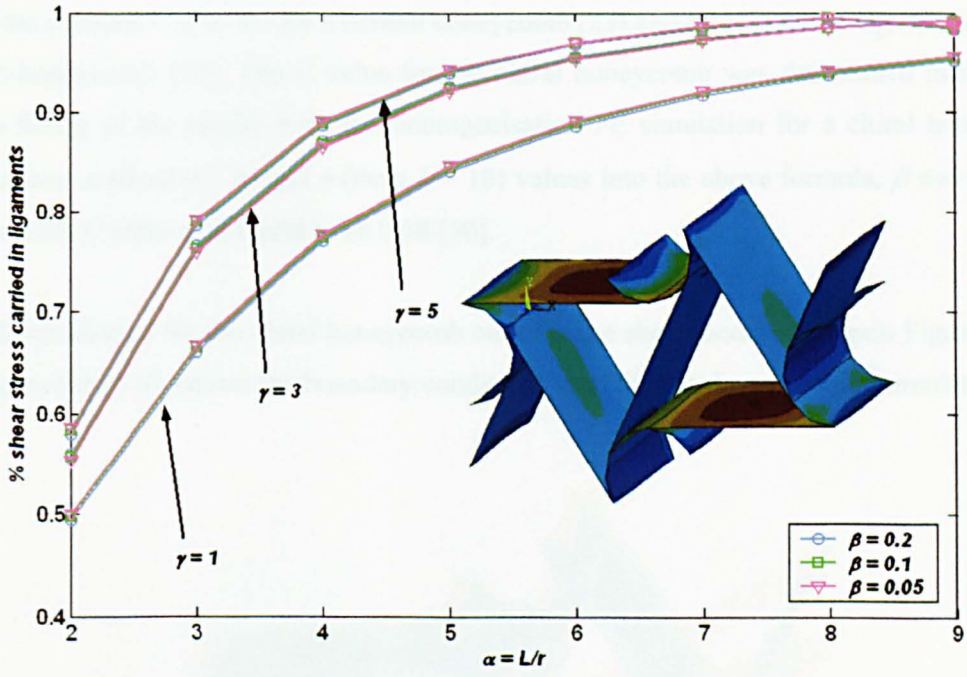
The minimum complementary energy theorem can be expressed as an inequality:

$$\frac{1}{2}\frac{\tau_{xz}}{G_{xz}}V\le\frac{1}{2}\sum_i\left(\frac{\tau_i}{G_c}V_i\right) \tag{6.18}$$

where τ_i is the shear strain in each of the walls of volume V_i. According to the assumption stated before, that the shear loading was mainly carried by the walls (verified in Figure 6.14), i.e. the G_{xz} can be developed based on the four walls *a*, *b*, *c* and *d* (Figure 6.12(b)). Also the symmetry of the cell indicates that walls *c* and *d* each carry the same load, therefore (6.18) is applied only to walls *a*, *b* and *c*. The summation of $\tau_i V_i$ is the resultant shear force F.



(a)



(b)

Figure 6.14 Percentage of shear carried in (a) cylinder (b) ligaments of a chiral honeycomb unit cell.

With some manipulation from (6.18), the lower bound on G_{xz} can be written as:

$$\frac{G_{xz}}{G_c} \geq \frac{\alpha\beta}{\alpha^2 \cos\theta + \pi} \frac{F^2}{F_a^2 + F_b^2 + F_c^2} \tag{6.19}$$

where F_a, F_b and F_c represents the forces in wall a, b and c, and F can be determined from the equilibrium of internal shear stresses (Appendix 3).

Similarly, for the case of G_{yz}:

$$\frac{G_{yz}}{G_c} \geq \frac{\alpha\beta}{\alpha^2 \cos\theta + \pi} \frac{F^2}{F_a^2 + F_b^2 + F_c^2} \tag{6.20}$$

In a general hexagonal honeycomb, the out of plane shear modulus is also affected by the gauge thickness, γ, which defines the ratio between the out of plane thickness to the length of cell wall [23]. The gauge thickness modifies the stress distribution and affects the bending of the wall under shear loading. Grediac has shown that the shear modulus has an inverse relationship with the gauge thickness, varying from the lower bound when the gauge thickness is high, to the upper bound when decreasing the thickness of the honeycomb [23].

For the chiral honeycomb, the effective out of plane shear modulus shares a similar trend to the hexagonal ones (6.21),

$$G_{eff} = G_{low} + \frac{C}{\gamma}\left(G_{up} - G_{low}\right) \tag{6.21}$$

where the constant C is 0.787 for a normal honeycomb [23] and 1.342 for a hexagonal re-entrant auxetic honeycomb [50]. The C value for the chiral honeycomb was determined using Least Square fitting of the results from the homogenisation FE simulation for a chiral honeycomb with various α (from 2 – 6) and γ (from 1 – 10) values into the above formula, β was fixed to 0.05, and the C value was found to be 0.58 [50].

The FE model used for the chiral honeycomb out of plane shear modulus is again Figure 6.3(c). The figure below illustrates the boundary conditions used for the shear modulus simulation.

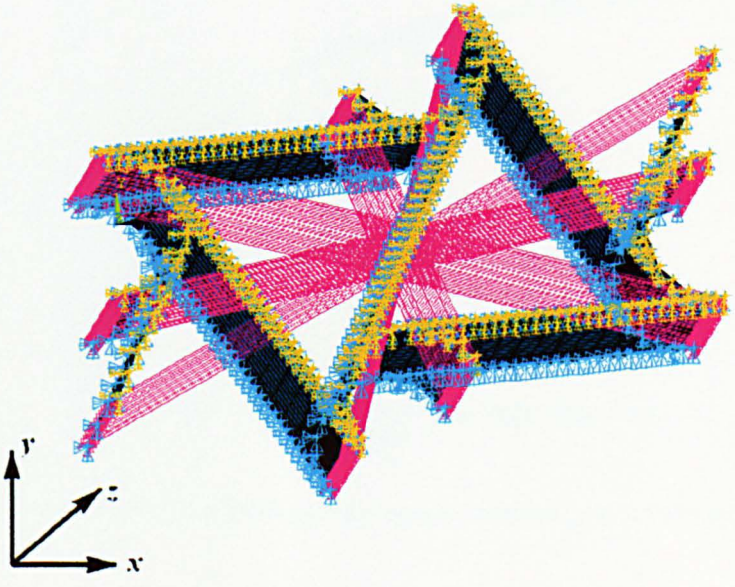


Figure 6.15 Boundary condition for chiral out of plane shear modulus simulation.

The boundary conditions are similar to those mentioned for the out of plane elastic modulus simulation, except that the top plane of the honeycomb is now a displacement field in the x-direction. The shear modulus was calculated from the reaction shear force averaged by the total area, and divided by the imposed strain.

iii. Experimental setup for chiral honeycomb

Some experiments were done to verify the chiral honeycomb analytical and numerical formulations. These included tensile tests, compression tests and three-point bending tests to determine the core material properties that made the chiral honeycomb, compressive strength, structural modulus and shear modulus of the chiral honeycomb respectively.

a. Compression test

The sample for the compression test was manufactured by the Exeter Advanced Technologies Enterprises, using the Selective Laser Sintered (SLS) Rapid Prototyping technology. The material used for manufacturing the compression test sample is the Nylon 6 DuraForm PA powder [35] and the core material properties (i.e. Nylon 6 DuraForm PA powder) were determined from three standard tensile specimens (Figure 6.16) manufactured and tested according to the ATSM D638 standard [9].

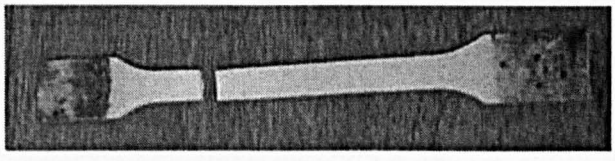


Figure 6.16 A tensile test specimen.

Figure 6.17 shows the stress-strain curve from one of the Nylon 6 DuraForm PA powder tensile test samples, giving Young's modulus of 15GPa, and yield strength of about 47MPa at 10% of strain. The tensile tests were carried out using a MAYES tensile machine, with strain rate of 0.1 mm/s.

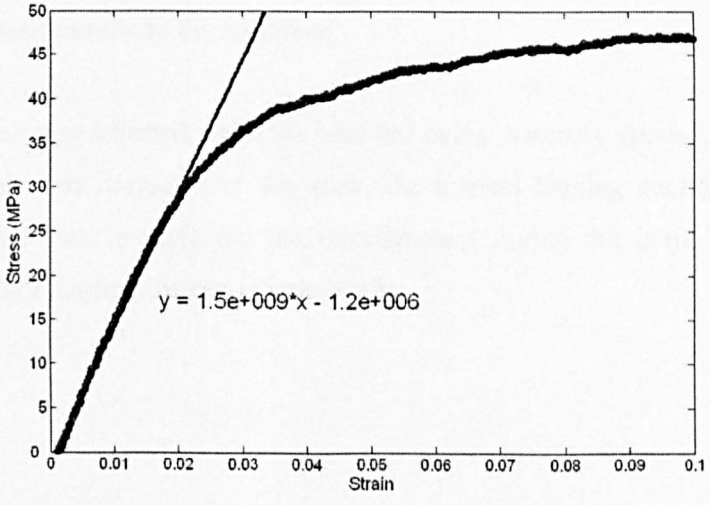


Figure 6.17 Stress strain curve for HP-PA powder sample obtained from ASTM D638 tensile test.

The overall size of the chiral honeycomb sample was 210mm × 200mm × 20mm, it was split into four equal size samples, each having size around 100mm × 105mm × 20mm, providing a total area of around 10500mm^2. The configurations of the chiral honeycomb test sample were α = 4.4, β = 0.22, γ = 1, L = 20mm and r = 4.5mm (Figure 6.18).

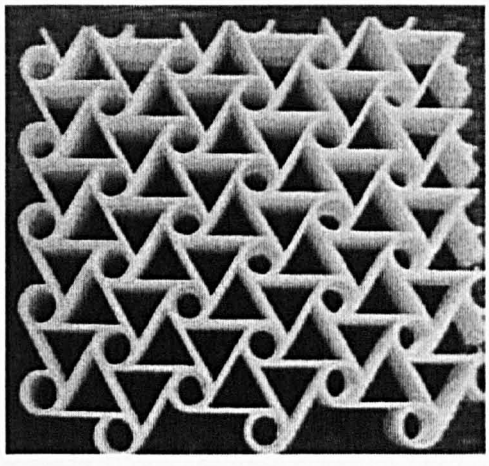


Figure 6.18 Chiral honeycomb compression test sample.

The compressive test was done according to the ASTM C365-00 standard (Flatwise Tensile Strength of Composites) for non-metallic core materials [8, 64]. The test was carried out using the Instron 801 machine. The loading strain rate was 0.008 mm/s, with which the specimen will reach its maximum load in around 3 – 6 min, and the displacements were collected at every 0.2s interval. The load was applied from the movement of the machine head to an alignment-plate, which is sitting on top of the specimen (Figure 6.19). It is to help getting a uniform load distribution to the specimen. The plate must be aligned at the centre of the machine head, and great care must be taken to ensure that the plate is sitting in parallel to the specimen, so that load is transferred perpendicularly to the specimen.

The alignment plate was adjusted while the load has being manually applied, any misalignment of the plate-sample was corrected at this time, the manual loading continued until all gaps visibly closed. This was to minimise the misalignment during the initial movement of the machine head, before starting the test (Figure 6.19).

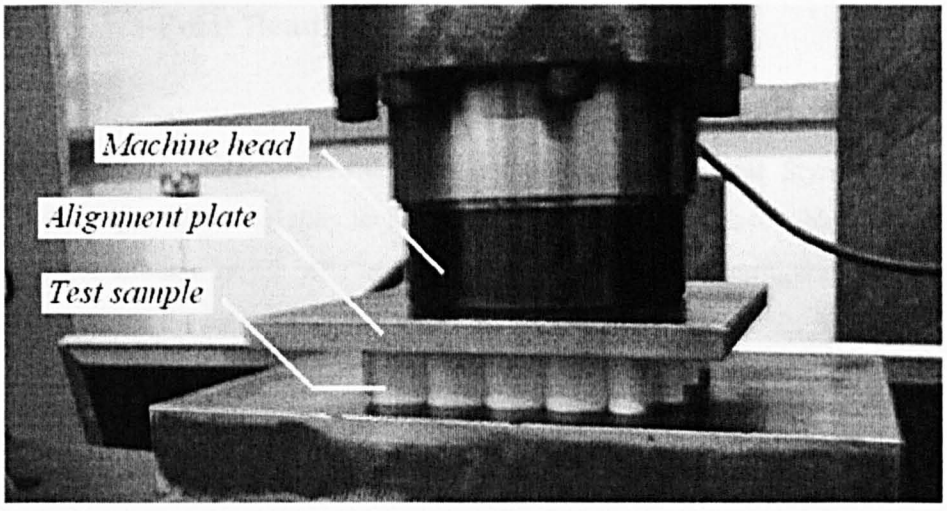


Figure 6.19 Compressive test apparatus set up.

The sample was tested for 50% strain; a force-displacement curve was plotted to determine the compressive strength (maximum load) and its compressive elastic modulus (the slope). Figure 6.20 shows the force-displacement curve of the sample up to 50% load, the maximum compressive strength was 87 kN, and the slope was 80.7 kN/mm.

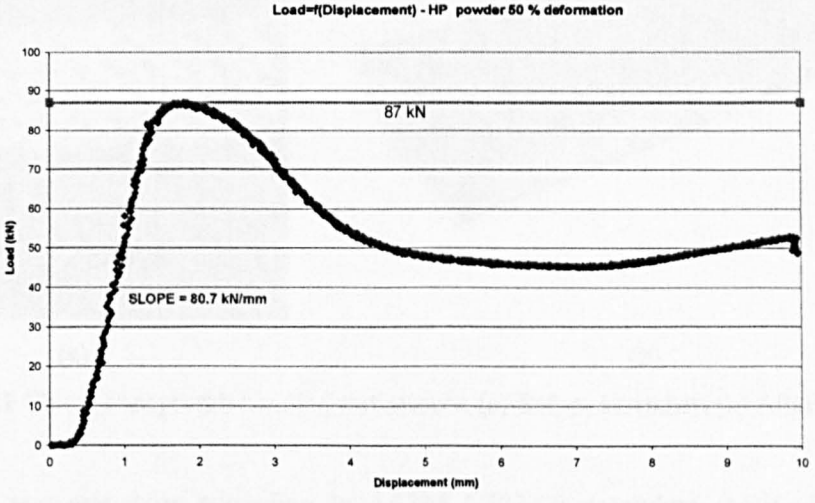


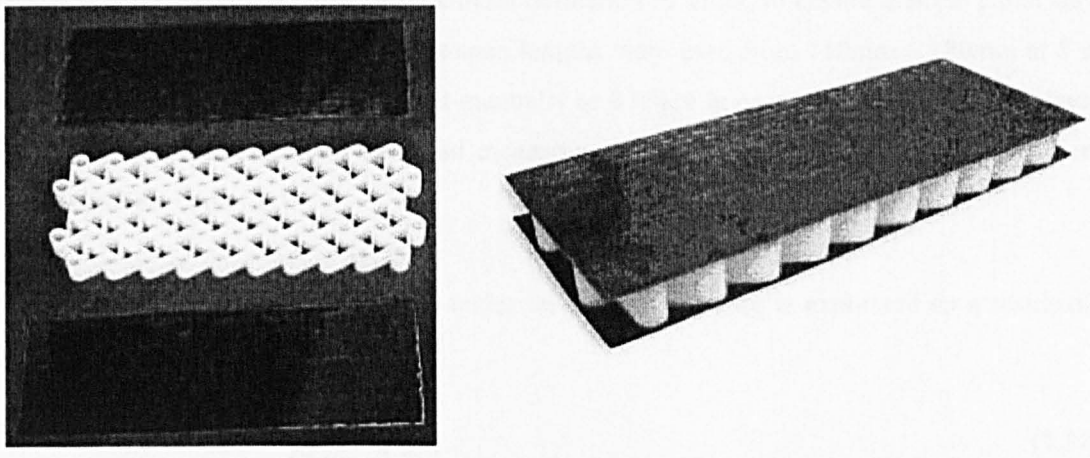
Figure 6.20 Chiral honeycomb under compression with strain up to 50%.

At the initial state, the curve is constant until the alignment plate starts to press on the whole specimen, when the load starts to transfer. The linear relationship in the force-displacement curve indicates the elastic compressive strength. It continues until the maximum load is reached and followed by compressive failure. The compressive failure of the sample is indicated by the decrease of force until around 45kN. A densification process starts after 8mm compression deformation.

b. 3-Point Bending test

The chiral honeycomb sample shown in Figure 6.21 was used for the bending test. The sandwich core was manufactured with ABS (Acrylonitrile Butadiene Styrene) plastic using Rapid Prototyping stereolithography technology. The ABS plastics have a Young's modulus of 900 MPa, yield strength of 30 MPa and Poisson's ratio 0.3. The configurations of the chiral honeycomb test sample were $\alpha = 4.4$, $\beta = 0.22$, $\gamma = 1$, $L = 20$mm and $R = 21.5$mm.

The faces were made of unidirectional prepreg ACG MTM28 carbon fibre. The face skin material has a tensile modulus of 121GPa, flexural modulus of 124GPa, flexural strength 1.7GPa and interlaminar shear strength 80MPa. These properties were normalised with volume fraction 0.6 at room temperature [50]. The face skins were attached to the chiral honeycomb core using EvoStik impact adhesive [50]. The overall dimension of the sandwich panel was 85.5mm × 225mm, with total thickness of 20.8mm.



(a) **(b)**

Figure 6.21 Chiral honeycomb bending test sample. (a) Before assembly, (b) After assembly.

The bending test was done according to ASTM C393-00 procedure (Method for Flexural Properties of Sandwich Cores) [7, 53], though the experiment is much easier to set up, the shear information will require some post processing of the experimental data, the post-processing steps will be mentioned later in the section.

The three point bending test was carried out using an Instron 8501 machine. The specimen is loaded in the middle of the plate at a rate of 0.005 mm/s, and the plate was simply supported at two ends, equidistant from the center (Figure 6.22). Two extensometers were placed at the supporting edges to identify any possible local crushing.

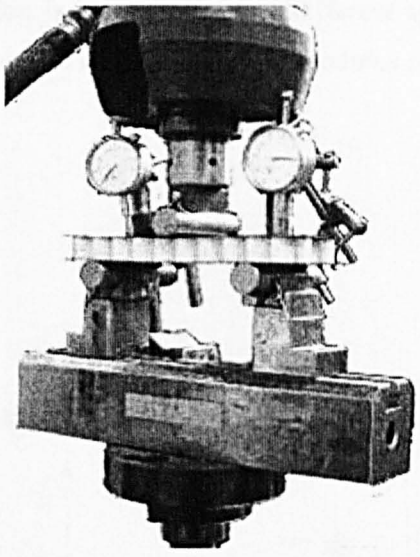


Figure 6.22 Machine set up for bending test [53].

The tests were carried out up to a maximum deflection of 1mm, to ensure that the panel stayed within its elastic limits. Eight different span lengths were used from 150mm – 180mm at 5 mm intervals. The specimen was preloaded manually to 0.04kN in order to account for any slack in the system. The displacement and load measurements were recorded every 1 second interval using LabView code [53].

The deflection of a sandwich sample under three-point bending is expressed as a resultant of bending and shearing effects:

$$\delta_{total} = \delta_{bending} + \delta_{shear} = \frac{FL^3}{48D} + \frac{FL}{4AG} \qquad \textbf{(6.22)}$$

where δ_{total} is the centre deflection of the sandwich, F is a point load applied at the centre of the sandwich, L is the span length between the two simply supported ends, D is the flexural rigidity, G is the shear modulus of the sandwich core and constant $A = bd^2/c$ (Figure 6.23), in this case, A = 1779.084 mm^2.

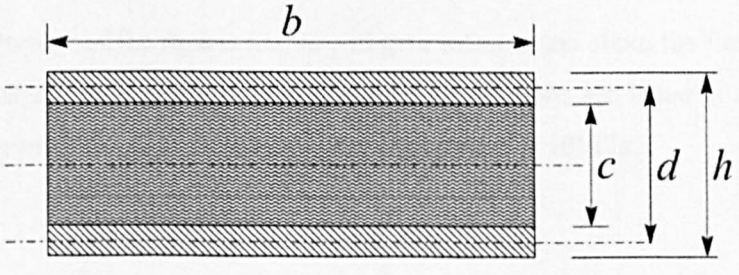


Figure 6.23 Cross sectional area of the sandwich panel.

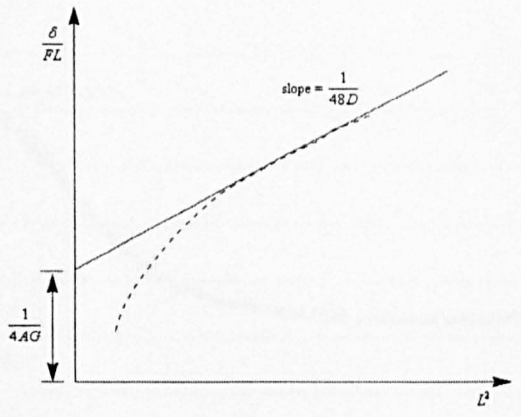
According to [4] the deflection is measured at two different spans and the resulting pair of equations can be solved for flexural rigidity and shear modulus of the sandwich core. Therefore rearranging (6.22), one gets

$$\frac{\delta}{FL} = \frac{L^2}{48D} + \frac{1}{4AG} \tag{6.23}$$

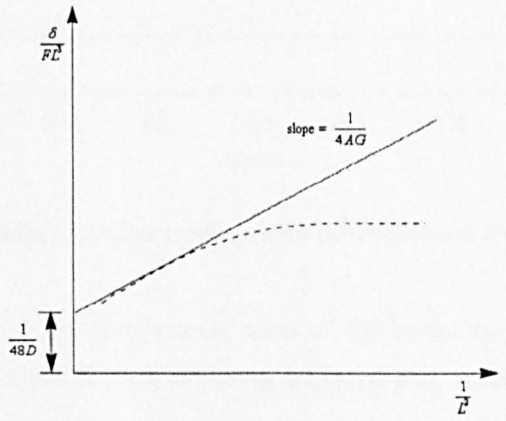
and

$$\frac{\delta}{FL^3} = \frac{1}{48D} + \frac{1}{4AG} \cdot \frac{1}{L^2} \tag{6.24}$$

which can be illustrated as straight lines in the graphs below:



(a)



(b)

Figure 6.24 Shear stiffness and flexural rigidity curves [4].

The slope and intercept of the figures above will give information about the flexural rigidity and the shear modulus of the bending sample. Detailed calculations are listed in Appendix 4. The out of plane shear modulus determined from the experiment is 10MPa.

iv. Experimental results vs theory and simulations

All the FE and analytical models mentioned earlier for, E_z, G_{xz} and σ_{crit} were verified with experimental results.

a. Elastic Modulus

For the elastic modulus, the experimental result was computed as the slope of the stress-strain curve from the compression test (Figure 6.25).

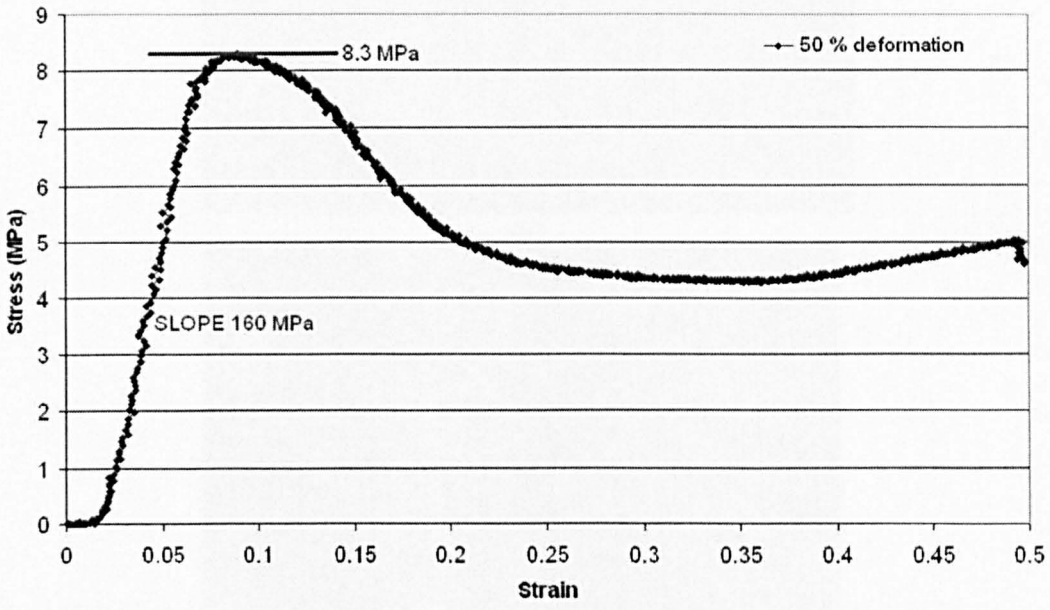
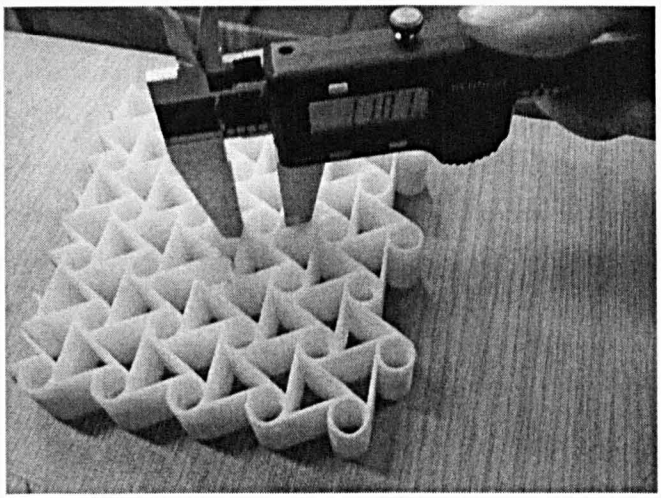


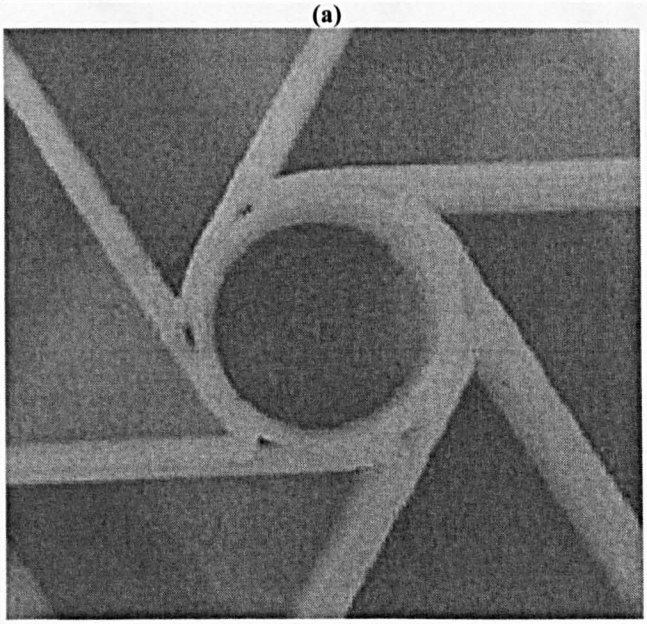
Figure 6.25 Structural modulus of the chiral honeycomb panel obtained from the compressive test.

It is considered here that the cylinders carried most of the loads, the parameter defining the cylinder thickness and its radius, $\beta = t/r$, is playing a crucial part. After measuring the internal and external diameter of the cylinders in the test specimen, the mean thickness of cylinders were found to be 1.5 mm, whereby the thickness of the ligaments were 'tapered' to close the connection between ligaments and cylinders. This can be seen in a coarser test sample that made by using stereolithography method (Figure 6.26(b)). Although the laser sintering technique has already overcome the imperfections / gaps manufactured using stereolithography (Figure 6.26(b)), the behaviour of the ligaments and cylinders are still the same as the stereolithography sample, such as the cylinders are still unable to rotate to create enough auxetic effects and the behaviour of the cylinders and ligaments are still very independent from each other under loading conditions.

Therefore the actual thickness of the cylinders was taken as 0.5 mm instead of full thickness of 1mm, which led to $\beta_{cylinder}$ as 0.11. This value of β is used in the analytical and simulation calculation of the chiral honeycomb elastic modulus. A rough measurement of a cylinder, using a caliper, is also shown in the figure below.



(a)



(b)

Figure 6.26 (a) Measurement of a cylinder of the chiral honeycomb sample, manufactured using laser sintering technology; (b) imperfection of the chiral honeycomb manufactured using stereolithography.

The other configurations were $\alpha = L/r$ = 20mm / 4.5mm = 4.44, $\gamma = b/L$ = 20mm/20mm = 1 and θ = 30°. Table 6.1 shows a comparison between the analytical, simulation and experiment which shows exellent results.

Analytical Equation (6.4) [MPa]	Finite Element [MPa]	Experiment in Figure 6.17 [MPa]
160	159	160

Table 6.1 Chiral honeycomb elastic modulus result comparisons between analytical, simulation and experiment.

As mentioned in Appendix 2, the analytical equations neglected the gaps between the cylinders and ligaments; it is still assumed that the cylinder is free to rotate, due to the torque produced while the ligaments are bent. In Figure 6.27, the finite elements model has also permitted rotation of the cylinders, though the curvatures of the cylinders were not perfectly fitted (i.e. looked more like a polygon while fitted by the shell elements). The manufacturing of this novel structure is still a very challenging task to be able to maximise the capabilities of the structure.

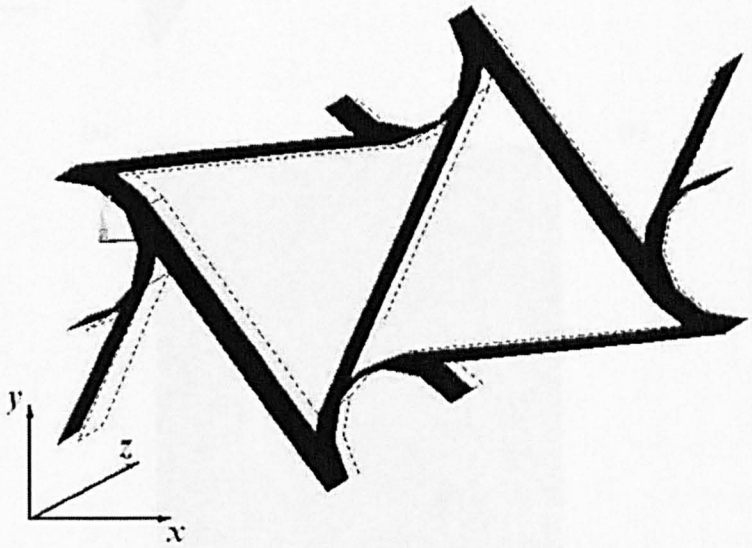


Figure 6.27 Displacement of the chiral honeycomb unit cell under loading in *z*-direction.

b. Compressive Strength

The compression strength of the chiral honeycomb is taken from the same test as above, and the result is more straightforward compared to the elastic modulus (Figure 6.20). Table 6.2 shows the comparison between theory, simulation and experiment.

Analytical (6.10) [MPa]	Finite Element [MPa]	Experiment [MPa]
8.9	8.5	8.3

Table 6.2 Chiral honeycomb collapse stress result comparisons between analytical bounds, simulations and experiment.

In Figure 6.28(a), the FE model shows a global buckling mode of the chiral honeycomb unit cell. While most of the ligaments already buckled under the loading, the displacement of the cylinder is behaved in a ring-type buckling, and is more rigid (Figure 6.28(b)).

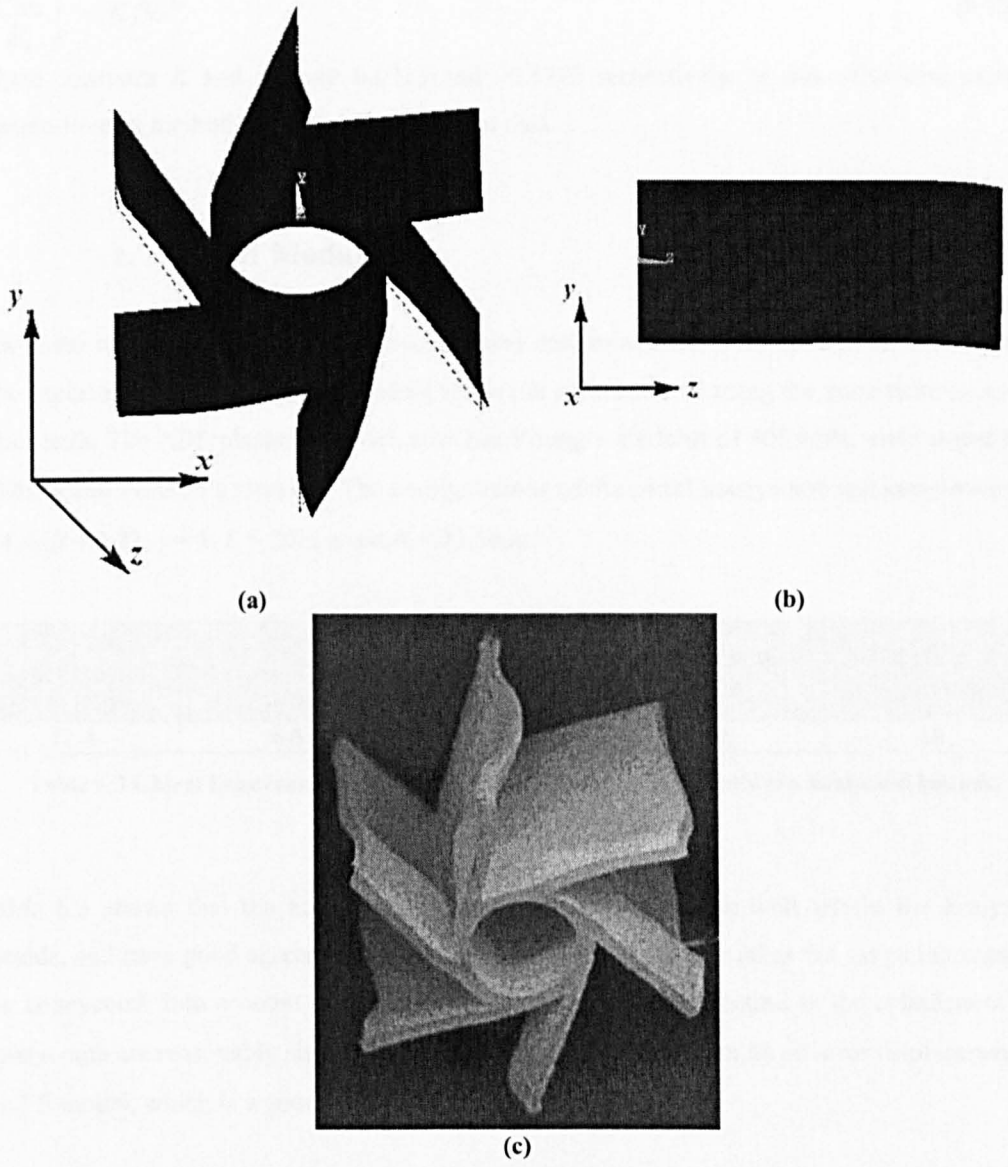


Figure 6.28 FE model and experimental sample for buckling analysis.

The overall mode shape agrees well with the compression test sample Figure 6.28(c), which was cut out from the test sample after an applied 20% compressive strain. The good agreement between the analytical, numerical and experimental shows that the assumption made earlier, where the cylinders contribute significantly to the compressive strength is valid. However, as shown in Figure 6.9 the contribution of ligaments will become more dominant when the thickness or β is higher.

It is known that the honeycomb collapse stress has a parametric form of relationship as described below [23]:

$$\left(\frac{\sigma_{crit}}{E_c}\right)^{\frac{1}{3}} = K\beta\alpha^m \tag{6.25}$$

where constants K and m were 0.0215 and –0.6288 respectively. It was calculated using a pseudo-inverse method from the FE simulation data.

c. Shear Modulus

The shear modulus of the chiral honeycomb was determined from the three-point bending test. The experimental sample for the bending test is not manufactured using the same material as the other tests. The ABS plastic sandwich core has Young's modulus of 900 MPa, yield strength of 30 MPa and Poisson's ratio 0.3. The configurations of the chiral honeycomb test sample were α = 4.4, β = 0.22, γ = 1, L = 20mm and R = 21.5mm.

Analytical Upper Bound (6.14) [MPa]	Analytical Lower Bound (6.19) [MPa]	Equation (6.21) [MPa]	Finite Element [MPa]	Experiment [MPa]
12.4	6.6	11	9.8	10

Table 6.3 Chiral honeycomb shear modulus result comparisons between analytical bounds, simulations and experiment.

Table 6.3 shows that the numerical and experimental results lie well within the analytical bounds, and have good agreement with the analytical equation that takes the gauge thickness of the honeycomb into account. The values are closer to the upper bound as the cylinders of the honeycomb are reasonably short (γ = 1). Figure 6.29 shows an example of shear displacement in the FE model, which is a good estimate of the experimental values.

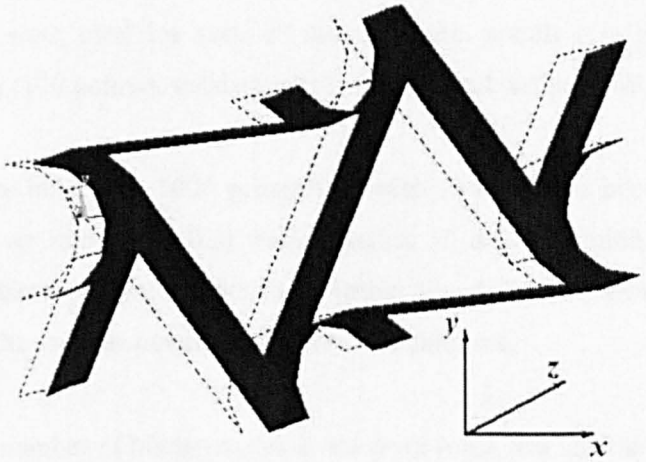


Figure 6.29 Shear displacement of a chiral honeycomb unit cell.

v. Optimisation Case study

In general, all the chiral models described previously were validated with satisfactory results. Surrogate models were again built to replace the time-consuming numerical models to overcome the bottleneck of an optimization process.

The surrogate models were built using both Neural Network and Genetic Programming to compare their effectiveness and accuracies in this optimisation case study.

The optimisation process in this section is a shape optimisation of the chiral honeycomb. For a given modal density of the chiral honeycomb, the optimisation process will compute the corresponding geometrical properties of the chiral honeycomb. As the modal density of the honeycomb is related to the out of plane shear modulus (G_{xz}) and elastic modulus (E_z)(aka apparent density in equation (6.1)), the surrogate models describe here are the GP and NN fitted G_{xz} and E_z models. Theory regarding modal density, which is a topic discussed under the Statistical Energy Analysis (SEA) studies of honeycombs can be found in [62].

a. Surrogate model setup for the chiral shear and structural modulus

The surrogate models for the chiral honeycomb out of plane shear properties were built by first obtaining data from the FE models, which were built from randomly generated geometrical parameters ($1 \leq \alpha \leq 6$, $0.05 \leq \beta \leq 0.2$, $1 \leq \gamma \leq 10$) from a uniform distribution. The geometrical parameters and the FE generated G_{xz} and E_z were then used for creating the surrogate models, a total of 400 points were used for each of the surrogate models (G_{xz} and E_z), which were allocated for training (170 points), validation (130 points) and testing (100 points) sets.

The GP models were built with 1000 generations each containing a population of 300. Each mutation and crossover rate (0.1 – 0.9) was trained at 10 different initial conditions (random numbers), and validated to avoid under-/over-fitting the data. The setup for each types of mutation method is the same as mentioned in previous chapters.

For NN models, the number of hidden nodes in the layer (only one layer is used) and number of presentations (affecting the momentum and learning coefficients) were the variables. Each different setting was also trained with 10 different initial conditions and then validated.

The best configuration of GP and NN surrogate model is listed in Table 6.4.

Surrogate Model	GP	MLP
E_z		
%MSE	0.0037	0.12
Configurations	Generations: 1000	Hidden nodes: 20
	Population: 300	Learning and momentum regime: 3
	Crossover rate: 0.2	
	Mutation rate: 0.3	
	No. of nodes: 50	No. of weights: 101
	Tree Depth: 16	
G_{xz}		
%MSE	0.0894	0.46
Configurations	Generations: 1000	Hidden nodes: 22
	Population: 300	Learning and momentum regime: 3
	Crossover rate: 0.6	
	Mutation rate: 0.2	
	No. of nodes: 49	No. of weights: 111
	Tree Depth: 22	

Table 6.4 Best configurations of GP and NN in fitting the G_{xz} and E_z data.

b. Objective function of chiral honeycomb shape optimisation

The optimisation process is to get the geometrical parameters of the chiral honeycomb, based on the desired modal density. As the modal density requires the knowledge of the chiral honeycomb relative density (i.e. $\frac{E_z}{E_c} \times \rho_c$) and out-of-plane shear property (i.e. G_{xz} and G_{yz}), these surrogate models developed in the section before will be computed during the optimisation process to find out the correct chiral geometry corresponding to these properties, therefore its modal density.

The theory regarding modal density can be found in [62]. It is one of the three properties (modal density, coupling loss and damping) that used in Statistical Energy Analysis (SEA), which determines the high frequency dynamic characteristics of a complex structure. The SEA method is useful for quantifying the transmissions and absorptions of the acoustic waves, handling vehicle internal noise and many other vibro-acoustic applications. At a high operating frequency, the modes of a structure can overlap, which made it difficult to count the resonant modes; the modal density allows the engineer to determine the vibration energy of the structure by giving the number of resonant modes per Hertz.

The honeycomb sandwich plates are much stiffer than normal uniform plates with similar mass; therefore the honeycomb plates have low modal density and are ideal for making typical spacecraft platforms and side panels. The theoretical expression and experimental measurement techniques of the modal density for honeycomb sandwich plates have been studied by [16, 56] and many other researchers.

The modal density, which is the objective function of the shape optimisation in this work, of the chiral honeycomb sandwich plate (Figure 6.30) with different configurations were computed using the equation below [56, 16]:

$$n(f)=\frac{\pi mAf}{gB}\left\{1+\frac{m\omega^2+2g^2B\left(1-\upsilon_c^2\right)}{\sqrt{m^2\varpi^4+4m\varpi^2g^2B\left(1-m^2\right)}}\right\} \tag{6.26}$$

where $g=\dfrac{\sqrt{G_{xz}G_{yz}}}{h_2}\left(\dfrac{1}{E_1h_1}+\dfrac{1}{E_3h_3}\right)$ is the core stiffness parameter, the shear modulus G_{xz} and G_{yz} is the same due to the isotropic behaviour of the chiral honeycomb, and will be computed from the surrogate model of G_{xz}. The constant A is the surface area of the plate, υ_c = 0.33 is the Poisson's ratio of the constitutive material, E_1 and E_3 are the structural moduli of the top and bottom face sheet, $B=\left(h_2+\dfrac{h_1+h_3}{2}\right)^2\left(\dfrac{E_1h_1E_3h_3}{E_1h_1+E_3h_3}\right)$ is the face sheets longitudinal stiffness parameter, f and ω are frequencies in Hz and rad/s respectively, $m=\rho_1h_1+\rho_2h_2+\rho_3h_3$ is the total mass per unit area of the sandwich plate, where ρ_1, ρ_2 and ρ_3 represent the density of the top face sheet, composite core and bottom face sheet individually, and $\rho_2=\dfrac{E_z}{E_c}\times\rho_c$. The data used for the sandwich core and faces are listed in Table 6.5, where the density of the chiral honeycomb, ρ_2, will be computed during the optimisation process using the surrogate model of the normalised elastic modulus, E_z.

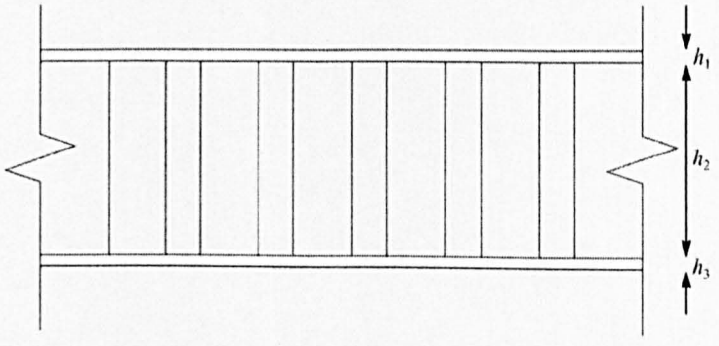


Figure 6.30 A chiral honeycomb composite model.

Chiral Honeycomb	
Area	0.23 m × 0.088 m
Thickness	2×10^{-2} m
Apparent Density	To be computed from the surrogate models
G_{xz} , G_{yz}	To be computed from the surrogate models
Face Sheets (CFRP layer)	
Thickness	2×10^{-3} m @ side
Young's Modulus	121×10^{9} Nm^{-2}
Density	1380 kgm^{-3}

Table 6.5 Sandwich panel data used for the computation of modal density.

The shape optimisation were again computed using the Differential Evolution (DE) routine. The scaling factor and crossover ratio was determined to be 0.9 and 0.7 respectively. The objective statement is, for a given frequency of 2 kHz, find a chiral honeycomb, which has a modal density of about 0.3 modes / kHz. The radius of the chiral honeycomb was fixed to be 4.5mm. The core material density (ρ_c) and the Young's modulus (E_c) of the chiral honeycomb is set to be 590 kg/m^3 and 5×10^6 N/m^2 respectively.

The shape converged with the above scaling factor and crossover ratio is shown in the figure below:

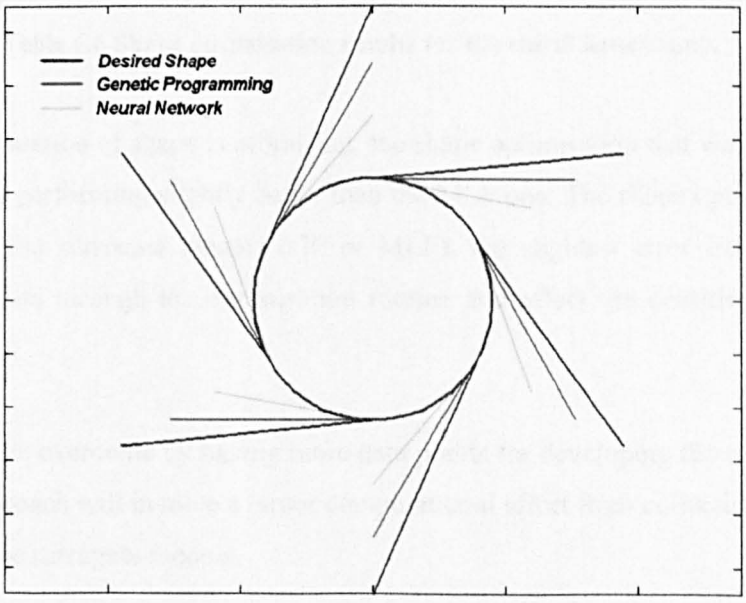


Figure 6.31 Final geometrical configuration of the chiral honeycomb obtained from the DE shape optimisation.

From the figure above, it shown that the new configurations have different ligament lengths (geometrical parameters are listed in Table 6.6). The ligaments were not appeared as tangents of the cylinders. This is due to the limitation from the numerical FE model, as it interpolates the curved surface by a polygon spline. This does not affect the FE calculation much, as the accuracy of the numerical models only increases slightly with a very fine polygon mesh on the curved cylinder surface.

The MSE for the shape optimisation was calculated as:

$$MSE = \frac{\left(n(f) - \hat{n}(f)\right)^2}{n(f)^2} \tag{6.27}$$

The MSE of the optimisation, based on the MLP and GP surrogate model, were 8.4999×10^{-9} and 6.7744×10^{-10} respectively. The corresponding relative density, shear modulus, geometrical parameters and the modal density are listed in the table below.

	Target Value	GP	MLP
α	4.44	3.3912	2.3131
β	0.22	0.1906	0.1887
γ	1	1.1102	2.2776
G_{xz} (N/m^2)	1.4392×10^5	1.4883×10^5	1.6420×10^5
ρ (kg/m^3)	125.60	139.30	182.19
$n(f)$(no. of modes / kHz)	0.3043	0.3043	0.3043
MSE %	n/a	6.8×10^{-10}	8.5×10^{-9}

Table 6.6 Shape optimisation results for the chiral honeycomb.

In general, the estimation of shape is promising; the shape optimisation that was built on the GP surrogate model is performing slightly better than the MLP one. The shape optimisation routine relies heavily on the surrogate models (GP or MLP); the slightest error from the surrogate model can propagate through the optimisation routine and affect the sensitivity of the shape optimisation.

This problem can be overcome by having more data points for developing the surrogate models. However, this approach will involve a larger computational effort from collecting FE simulation data for training the surrogate models.

7. Conclusion

The work done in the thesis so far has developed a novel metamodelling technique (GP) to improve the effectiveness of honeycomb simulations based on a FEA approach. The complex simulation models were successfully represented by a reduced order metamodel, which was built by using either NN or GP. The NN or GP built metamodels were approximating the relationship between the honeycomb geometrical parameters and its corresponding material properties; they were also used to perform interpolation / some minor extrapolation of the material properties up to certain accuracies, in order to avoid regenerating a new FE model for every minor change of the honeycomb structure.

In terms of the FE modelling of the microstructures, either bounded (Chapter 4) or cellular models (Chapter 5 and 6), are built using the homogenisation method. This method allows the user to generalise the heterogeneous domain into a more homogeneous domain, by including all the contents of the heterogeneous domain into a unit cell according to its proportions.

All the material models used in this study contain only hollow and solid materials. However, a strategic distribution of the hollows and solids must be considered when forming a unit cell, in order to maintain the periodicity of the unit cell and make it easy to perform numerical simulations. For a bounded model mentioned in Chapter 4, the ratio between the hollow and solid is low and it is easy to manipulate the numerical model by choosing the most obvious unit cell. While for the cellular solids, the ratios between hollow and solid are much higher. Particularly for the auxetic cellular solids, the existence of the hollows in fact promotes the change in shape of the auxetic materials during deformations. Furthermore as the solids are thinner, the free-ends need to be constrained with constraint equations, in order to guide the movement of the cellular solid unit cell when perturbation is introduced during the numerical simulations. The coding of these numerical models for bounded and cellular solids, using the homogenisation technique, has been successfully validated by experiments and analytical solutions in Chapter 4. The finding in Chapter 4 has been published in 2004 (Please refer to the publications list [I]).

One of the auxetic honeycombs studied here was chiral honeycomb, the experimental validations, analytical derivations [IV - IX] and comparisons between these with the metamodels were also reported in [IV - IX], and these results shows excellent agreements, which proves that the assumptions made so far were correct.

For the metamodel built in this work – based on Genetic Programming, show its effectiveness and its compatibility to the MLP in some of the tests mentioned in Chapter 3. The GP is still in its very primary stage, and more detailed works are required to further refine the program. These will be discussed in the next chapter. The GP extrapolation capability is also lower than the MLP; however, it is not recommended that either of these surrogate models is reliable for extrapolation.

From the studies of the aircraft wing damage detection (Chapter 3), the GP has not only shown its performance at such early stage of coding (without many mutation / crossover strategies). It has also shown the capability of choosing the most sensitive parameters (instead of using all) to create the metamodel. Traditional data fitting requires data screening to pick for the few most influential parameters to reduce the order / complexity [57]. The GP's capability to pick the most influential parameters has provided a very convenient way of analysing data. This finding has been reported in the author's publications. (Please refer to the publications list [II, VI])

It is worth mentioning that the 10000 generations used in Chapter 3 was during the testing stage of the GP code and the GP configurations were determined heuristically to examine the performance of GP in fitting some theoretical formulas of the hexagonal honeycomb within a predefined domain.

For the damage detection of the aircraft mentioned in Chapter 3, the first natural frequency of the aircraft was found to be sensitive particularly to the 1, 2, 4 and 8th parameter, which denotes for the aircraft's wing / fuselage connections, the wing / pylon connection of the outer turbine engine and the wing stiffness itself. While for the first element of the mode shape that corresponds to the natural frequency, it shows sensitivities to almost all parameters, except the 5th and 9th parameter, which denotes for the wing / pylon connection of the inner turbine engine and the fuselage / HTP connection. Both these selections of parameters agree with [57], which states that the most influential parameters for the natural frequency were the 1st and 8th parameter; and for the first element of the modal displacement affected mostly by the 1st, 3rd, 4th, 6th and 8th parameters.

When the coding of the GP became more complete (more methods of mutation were included), it was put to perform some metamodelling, such as to replace the numerical models in Chapter 4 – 6. It has shown some good achievements. However, as pointed out earlier, the GP is not reliable for any extrapolation; it has to be used strictly within the design domain. Also, it is inappropriate to perform mathematical operations among the GP surrogate models. The GP models, though appeared to be some conventional mathematical functions, it does not follow the rule of elementary / advanced manipulations of the conventional functions.

The applications of using the GP surrogate models in the design of a honeycomb structure have also shown good agreement when compared among other surrogate models. It has shown some successful applications when being cascaded as a sub-function of the optimisation process. It is worth mentioning that this cascading of surrogate models (either NN or GP or polynomials) will propagate (and possibly magnify) the error of the surrogate model in representing the actual model, through the optimisation process in this case. The less complicated the optimisation in cascading the metamodels, the less error propagation would be magnified over the optimisation procedures.

Since the GP in this work was developed in the Java language, the interface between the GP metamodel to the objective function (written in Matlab) has slowed down the optimisation process significantly. The reason this occurred was due to the difference in function handling. Matlab is tailored to handle various numerical calculations; while for Java, the mathematical functions included are rather elementary. Manipulations of these functions will lead to error if NaN and Inf were produced, which will halt the GP training. Hard limits were introduced in Java, during the training of the GP, to avoid such problems, as shown in Table 4.5. However, the GP metamodel will create conflicts if these hard limits were not introduced in Matlab during the optimisation, and these hard limits of the Java mathematical functions were introduced as little sub-routines of the optimisation process. These little sub-routines will be called each time the GP metamodel uses them.

The speed of training the GP using Java is also an issue. Though Java is powerful in terms of its object orientated programming nature and its independence of computer platforms, it is memory hungry and relies heavily on the capacity and performance of the computer. The basic requirement up to date to run Java programs is 512MB and 3GHz of RAM, the more advanced the computer, the faster the computation of GP.

While performing the shape optimisation of the auxetic honeycombs, it is noticed that the slightest change of shape of the material unit cell model can have a large effect on its overall structural performance. The metamodelling techniques were introduced in this work to obtain an understanding of the geometrical parameters of the unit cell to its corresponding structural behaviour. This was done by using a reduced order replacement / surrogate models to represent a random Gaussian distribution of input – output matching of the numerical simulations. The resultant surrogate models provide the relationship between the unit cell geometry and its corresponding structural behaviour within a predefined design range, which also saved some computational effort to re-generate the unit cell model for every change of the geometry. These

surrogate models were inserted into the shape optimisation routine in order to perform the shape optimisation based on the criteria set by the user.

The shape optimisation of the honeycombs that is studied in this work, has suggested a new approach for material optimisation. The shape optimisation was targeted at only one unit cell at a time, where the designer already knows which type of unit cell to be used. The geometry of the unit cell was optimised based on the desired performance predefined by the designer. The shape optimisation routine was carried out using the differential evolution method (DE). The basic idea of this DE shape optimisation is in fact an inverse of the surrogate models, in other words, it is to get the geometrical parameters based on the desired structural properties, by utilising the aforementioned NN or GP surrogate models.

The shape optimisation performed in both the hexagonal re-entrant honeycomb and the chiral honeycomb shows successful results. These optimisation procedures require constant updates of the properties of the sandwich core, which relies on the geometry of the honeycombs. Therefore, instead of calling for a FE routine during each step of the optimisation, a surrogate model was inserted as a subroutine of the optimisation procedure. Also in Chapter 5, the hexagonal honeycomb has some transitional properties between auxetic and normal when changing from negative angles to positive angles, the solutions lying in this region are very sensitive to the accuracies of the metamodels. Error will occur when the metamodels gives positive Poisson's ratio for negative angles.

In Chapter 6, the shape optimisation was in fact embedded within a few different surrogate models (some developed using GP, others using MLP). The chiral honeycomb surrogate models in Chapter 6 were more complicated, the error of the metamodel in representing the original geometry – property has propagated throughout the optimisation process. Also, the optimisation problem proposed for the chiral honeycomb has a lot of local minima, as there are a few possible chiral geometry configurations that can give a close approximation of the desired modal density. Therefore hard constraints must be imposed and several trials and different scaling / crossover ratios (for the Differential Evolutions) are also suggested. The choice of the best model will depend on the designer, as which one is more feasible to be manufactured etc.

8. Future work

i. Chiral honeycomb

The investigation of the honeycombs involved in this work has led to some new understanding of the structural behaviour of the auxetic honeycombs, typically the chiral honeycombs. Analytical, computational and experimental studies has validated that the assumptions were correct, and the mechanisms that are responsible for their unusual properties.

However, the manufacturing of the chiral honeycomb is still an obstacle to exploit a full capability of the chiral honeycomb. Though the gaps presented in Figure 6.26(b), which was manufactured using stereolithography technique, were overcome by using the laser sintering technique, the cylinders in the chiral honeycomb must be allowed to rotate and warp the ligaments in order to exhibit higher deformation.

On top of the outstanding out-of-plane performance of the chiral honeycombs, they are also a good candidate for manufacturing smart panels for damage detection, where the sensors can be located inside the cylinders, as the cylinders do not collapse easily compared to a flat panel. However the wiring of sensors, optimised location of sensors and the weight of sensors affecting the dynamical behaviour of the chiral sandwich panel is still a challenging task.

The study of damage detection in the sandwich panel, particularly skin cracks and the core-skin debonding, would require the measurement of strain or acceleration imposed to the sandwich panel. While the damage of the panel, in general requires the knowledge of the total strain of the panel, the acceleration measurements at different locations are useful for impact testing. Though the strain gauge provides direct information, it is troublesome to mount the gauge inside the cylinders. Furthermore, it is too difficult to ensure an excellent contact between the strain gauges with the cylinder wall, with such a small opening.

On the other hand, the accelerometer is easier to install and does not require advanced workmanship. However, they have a disadvantage in terms of their weight. As the kinetic energy is indeed proportional to the mass of the structure, a sensor with relatively higher weight than the sandwich panel would modify the dynamic characteristics of the panel. The size of the accelerometers is still a problem, while the cylinders are the most resistant part in the chiral honeycomb, the accelerometer must be small enough to be inserted into the chiral honeycomb cylinder. Also, the cylinders do not provide a flat surface to glue the accelerometers, which also brings a difficulty to use the accelerometers, though it could be glued on the ligament walls as a

less desirable alternative. The short range of operating frequencies that is provided by the accelerometers is also a challenge, as a sensor with high frequency with wide range is needed.

Another alternative choice of sensor would be a PVDF (polyvinylidene fluoride) film. The PVDF film can be bent into complex shapes, which can accommodate any surfaces. But these PVDF sensors have some disadvantage such as they are more delicate than the conventional sensors and can be damaged mechanically, which will cause unreliable measurements.

A modal density experiment and simulation were performed on the chiral honeycomb core itself, without any cylinders being filled (to represent the sensors' weight). The result and procedure of the modal density experiment is described in Appendix 5. Meanwhile the optimisation of chiral honeycomb sandwich panel remains numerical and the modal density of a chiral honeycomb sandwich was modelled based on the theoretical studies of [16, 43, 56].

Apart from all the problem of choosing the correct sensors, the problem of wiring of sensors inside the chiral honeycomb sandwich panel remains unsolved. The wire can pass around the honeycomb itself easily without the top and bottom sandwich faces. A solution to this is to bore a hole in the honeycomb to pass the wires. This approach is undesirable as it decreases the mechanical properties of not only the honeycomb, but also the whole sandwich, as the hole will be a stress concentration place during loading. An alternative way to this wiring problem would be to create some grooves at the panels as shown in Figure 8.1. It was suggested in [49] that this layout could be manufactured using Rapid Prototyping (RP) technology, such as stereolithography or laser sintering, as it would manufacture the chiral honeycomb with grooves as one piece, rather than drilling holes to the completed product, which initiates cracks. However the suggestion would involve precision manufacturing, and though the RP technology brought down the manufacturing cost of some products, precision manufacturing using RP technology would require strict workmanship, which may not turn out to be cost effective after all.

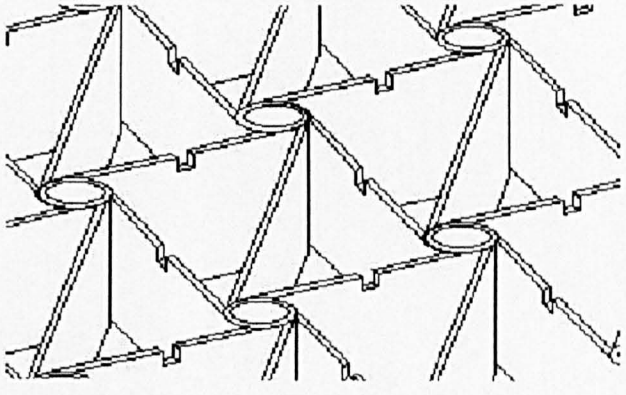


Figure 8.1 Suggestions of cutting grooves on the ligaments for sensor wiring [49].

The optimisation of sensor locations in the chiral honeycomb is another topic to study, as it is to minimise the cost of installing excessive sensors, minimise the weight added on to the panel that would change the dynamical behaviour of the sandwich, while still manage to capture good results from the damage detection tests. While the experiment and manufacturing of the chiral honeycomb for sensor installation still remains difficult, the optimisation of sensor location may lead to some interesting findings.

ii. Genetic programming

As mentioned earlier, the Genetic Programming (GP) developed in this studies still needs some improvements. These include the optimisation of the randomly assigned constant in the individuals, evolution strategies such as the mutation / crossover and selection method with their effects on the population search direction, more functions included and more effective bloat controls.

The optimisation for each randomly assigned constant in the individuals was proposed to maximise the possibility to search for the best individual for shorter generations and population. Simulated annealing may be a choice for refining these constants, as it can vary the optimisation steps according to the generations. When the GP is at its initial generations, the refining step would be bigger, and slowly reduce to smaller steps as the GP evolves towards the maximum allowed generations. This may waste more computer effort compared to the GP without such optimisation; however, the trade off of computer effort should be studied before any decision is made.

More initiation of population, selection methods and evolution methods should also be included to suit for different purposes. The initiation must give adequate diversity into the population, in order to perform healthy genetic operations, examples of initiation method include the grow method, the full method and a mixture of both etc. The selection method is also important, as it usually affects the convergence of an evolutionary program, examples of selection method include fitness proportionate, which was used in this work, tournament selection, demetic grouping etc. Tournament selection allows the user to vary the selection pressure by varying the size of the tournament. And the demetic grouping is also an alternative selection to avoid premature convergence [21]. More variety of the genetic operators must also be included, in order to obtain less destructive crossover and mutation processes. Some studies on the methods used for crossover and mutation can be found in [41, 11].

As also mentioned in the previous chapter, the GP metamodels experience performance degrade when being used in Matlab. This may be overcome by improving the capabilities of GP-Java in handling NaN and Inf.

iii. Auxetic material design using metamodels with shape optimisation

Auxetic materials are exhibiting some very interesting properties as a result of its counter-intuitive deformation, it will be better if one can create more auxetic structures, instead of replying on the limited choice of naturally occurred auxetic materials.

Ideally, a shape optimisation, which can evolve any sensible shape, based on the desired properties of the engineer is very exciting, for example shapes that exhibits auxetic behaviour. However, this will involve having a very large database of all possible auxetic structures and their corresponding properties to training the surrogate models (i.e. large database of metamodels for every different auxetic structure), and able to create an effective classification (As there may be regions where different geometrical configurations of structures having similar performances.). This will result in a huge computational power and ineffective training.

Another problem is the interpolation between structures in order to create a new structure. Imagine teaching a child to differentiate two types of fruit, orange and apple, it is straightforward to classify them into two different groups. The orange having rough skin and orange in colour, while the apple has smooth skin and red in colour (assuming that there is only one colour for all apples). When the child was asked to name a fruit, which has a rough skin and red in colour, a strawberry may come out as an answer. While the metamodel may learn much slower than a child, also a surprising 'strawberry' will not appear as an answer when it was asked to interpolate among these different structures. In the mean time, the search for new auxetic materials / structures remains as a blind search, which much relies on the human's creativity and curiosity.

Reference

1. Alderson, A., *A triumph of lateral thought*, Chemistry and Industry, 1999, pp. 385 – 391.
2. Allaire, G., *Two-scale convergence and homogenisation of periodic structures*, Commissariat à l'Energie Atomique, School of Homogenisation, 1993, France.
3. Allaire, G., *Shape optimisation by the homogenisation method*, Applied Mathematical Science, vol. 146, Springer, 2002, New York.
4. Allen, G. H., *Analysis and design of structural sandwich panels*, Pergamon Press, Oxford, 1969, pp. 257 - 259.
5. Angeline, P. J. and Fogel, D. B., *An evolutionary program for the identification of dynamical systems*, Proceedings of the International Society of Optical Engineering (SPIE), vol. 3077, 1997, pp. 409 - 417.
6. ANSYS Inc., *ANSYS: Reference 6.1: Element reference*, ANSYS Inc., 2002, Canonsburg, PA.
7. ASTM standard C393-00, *Test method for flexural properties of sandwich cores*.
8. ASTM standard C 365-00, *Standard test method for flatwise compression properties*.
9. ASTM standard D 638–99, *Standard test method for compressive properties of rigid plastics*.
10. Banzhaf, W. and Langdon, W. B., *Some consideration on the reason for bloat*, Genetic Programming and Evolvable Machines, vol. 3, 2002, pp. 81-91.
11. Banzhaf, W., Nordin, P., Keller, R. E. and Francone, F. D., *Genetic programming – An introduction on the automatic evolution of computer programs and its applications*, Morgan Kaufmann Publishers, Inc., 1998, California.
12. Bendsøe, M. P. and Kikuchi, N., *Generating optimal topologies in structural design using a homogenisation method*, Computer Methods in Applied Mechanics and Engineering, 71, 1988, pp. 197 – 224.
13. Blain, S., *The flatwise compressive behaviour of hexagonal chiral honeycomb*, MEng. Thesis, University of Sheffield, 2004.
14. Cherkaev, A., *Variational methods for structural optimisation*, Springer, 2000, New York.
15. Cherkassky, V. and Mulier, R., *Learning from data – concepts, theory and methods*, John Wiley and sons, Inc., 1998, Canada.
16. Clarkson, B. L. and Ranky, M. F., *Modal density of honeycomb plates*, Journal of Sound and Vibration, 91(1), 1983, pp. 103 – 118.
17. Cundy, A. L., *Use of response surface metamodels in damage identification of dynamic structures*, MSc. Thesis in Engineering Science and Mechanics, Virginia Polytechnic Institute and State University, 2003, USA.
18. Evans, K. E., Nkansah, M. A., Hutchinson, I. J. and Rogers, S. C., *Molecular network design*, Nature, 353, 1991, pp. 124.
19. Flanagan, D., *JAVA in a nutshell a desktop quick reference, 4th. Edition*, O'Reilly, 2002, Sebastopol, USA.
20. Fleming, P., *MSc. Lecture notes on optimisation and search*, University of Sheffield, Automatic Control and Systems Engineering Department, 2001.
21. Fraser, A. P., *Genetic programming in C++ (A manual in progress for gpc++, a public domain genetic programming system)*, University of Salford, Cybernetics Research Institute, 1994.
22. Gervy, V., *Experimental and numerical analysis of the transverse shear in hexagonal chiral honeycombs*, University of Sheffield, Master of Engineering thesis, 2005.
23. Gibson, L. J. and Ashby, M. F., *Cellular solids – structure and properties, 2nd. edition*, Cambridge Press, 1997, Cambridge.
24. Guedes, J. M. and Kikuchi, N., *Preprocessing and postprocessing for materials based on the homogenisation method with adaptive finite element method*, Computer Methods in Applied Mechanics and Engineering, 83, 1993, pp. 143 – 198.
25. Hadjigeorciou, E. P. and Stavroulakis, G. E., *The use of auxetic materials in smart structures*, Computational methods in science and technology, 10(2), 2004, pp. 147 – 160.
26. Hassan, M. R. and Scarpa, F., *Shape memory alloys auxetic honeycombs: design and properties*, Auxetic Materials Network: Abstract of Papers, Poster exhibition in the

AUXETNET first annual young researchers forum, University of Sheffield, 2004, pp. 19 – 21.
27. Hassani, B. and Hinton, E., *Homogenisation and structural topology optimisation*, Springer, 1999, UK.
28. Haykin, S., *Neural networks a comprehensive foundation*, Macmillan College Publishing Co. Inc., 1994, New York, USA, pp. 138-235.
29. Herber, K., *Chiral honeycomb design for sandwich structure*, University of Sheffield, Master of Engineering thesis, 2005.
30. Holmes, H. M., *Introduction to perturbation methods*, Springer-Verlag, 1995, New York.
31. http://dictionary.reference.com/search?q=chiral
32. http://en.wikipedia.org/wiki/Chirality_(mathematics)
33. http://mathworld.wolfram.com/DivergenceTheorem.html
34. http://www.flownet.com/gat/papers/lisp-java.pdf
35. http://www.x-at.co.uk/pddtm.htm
36. Hull, D. and Clyne, T., *An introduction to composite materials*, 1996, New York, pp. 60-77.
37. Kervokian, J. and Cole, J. D., *Multiple scale and singular perturbation methods*, Springer-Verlag, 1996, New York.
38. Kikuchi, N., *Homogenisation method and its application to material design for composites - An analysis of thermal elastoplasticity*, International Conference on Computer Assisted Materials Design and Process Simulation, ISIJ, 1993, Tokyo, pp. 434 – 439.
39. Kikuchi, N., Chung, K. Y., Torigaki, T. and Taylor, J. E., *Adaptive finite element methods for shape optimization of linearly elastic structures*, Computer Methods in Applied Mechanics and Engineering, 57, 1986, pp. 67 – 89.
40. Koish, M. and Kabe, K., *Development of homogenisation analysis system based on general-purpose FEM code*, JSME International Journal, Series A, 38 (2), 1995, pp. 171 – 176.
41. Koza, J. R., *Genetic programming on the programming of computers by means of natural selection, 7th. Printing*, MIT Press, 1992, pp. 73-164, Massachusetts, USA.
42. Kyprianou, A., Giacomin, J., Worden, K., Heidrich, M. and Böcking, J., *Differential evolution based identification of automotive hydraulic engine mount model parameters*, Proceedings of the IMechE, Journal of Automobile Engineering, Proceedings part D, vol. 214(D3), 2000, pp. 249 – 264.
43. Lew, T. L., Spadoni, A., Scarpa, F. and Ruzzene, M., *Chiral hexagonal cellular sandwich structure: A vibroacoustic assessment*, Proceedings of SPIE, Smart Structures and Materials 2004: Damping and Isolation, 2004, San Diego, USA.
44. Lew, T. L., Spencer, A. B., Scarpa, F., Worden, K., Rutherford A. and Hemez, F., *Identification of response surface method using genetic programming*, Proceedings of ISMA 2004, 2004, Belgium, pp. 3287-3299,.
45. Li, Y., Xin, X., Kikuchi, N. and Saitou, K., *Optimal shape and location of piezoelectric materials for topology optimisation of flextensional actuators*, GECCO, 2001, San Francisco.
46. Luke, S. and Spector, L., *A comparison of crossover and mutation in genetic programming*, Proceedings of the Second Annual Conference on Genetic Programming (GP-97), Morgan Kaufmann, 1997.
47. Marsden, W. and Irving, D. J., *NAFEMS: How to – analyse composite*, NAFEMS Ltd., 2002, Glasgow.
48. Panait, L. and Luke, S., *Alternative bloat control methods*, GECCO 2004, LNCS 3103, 2004, pp. 630-641.
49. Paulhac, A., *New chiral honeycomb concept*, BEng. final year report, University of Sheffield, 2003.
50. Paulhac, A., Scarpa, F., Lew, T. L., Perrot, D. and Yates, J. R., *The linear elastic transverse properties of hexagonal chiral honeycombs*, submitted to International Journal of Mechanical Sciences, 2005.
51. Pavliotis, G. A., *Homogenization theory for partial differential equation*, http://www.ma.ic.ac.uk/~pavl/homogenization.html, 2004.
52. Périn, F. and Lévy, T., *Application of the homogenisation method to a suspension of fibres*, International Journal of Engineering Sciences, 38(8), 1994, pp. 1253 – 1269.

53. Perrot, D., *Experimental testing of a chiral honeycomb to determine out of plane properties*, MEng. Thesis, University of Sheffield, 2003.
54. Prall, D. and Lakes, R., *Properties of a chiral honeycomb with Poisson's ratio -1*, International Journal of Mechanical Sciences, 39, 1996, pp. 305-314.
55. Ravirala, N., Alderson A., Alderson, K. L. and Davis, P. J., *Auxetic polyamide fibres*, Auxetic Materials Network: Abstract of Papers, Poster exhibition in the AUXETNET first annual young researchers forum, University of Sheffield, 2004, pp. 25 – 27.
56. Renji, K., *Experimental modal densities of honeycomb sandwich panels at high frequencies*, Journal of Sound and Vibration, 237(1), 2000, pp. 67 – 79.
57. Rutherford, A. C. and Hemez, F. M., *Bayesian model selection for a finite element model of a large civil aircraft*, IMAC-XXII, January, 2004, Dearborn, MI.
58. Rutherford, A., *Use of response surface metamodels in damage identification of dynamics structures*, MSc. Thesis, Virginia Polytechnic Institute and State University, 2002,.
59. Ruzzene, M. Scarpa, F, and Soranna, F., *Wave beaming effects in two-dimensional cellular structures*, Smart Materials and Structures, 12, 2003, pp. 1-10.
60. Sanchez-Panlencia, E., *Non homogeneous and vibration theory, lecture notes in physics vol. 127*, Springer-Verlag, 1980, Berlin, pp. 1 – 61.
61. Sanchez-Panlencia, E., *Homogenisation technique for composite media, lecture notes in physics vol. 272*, Springer-Verlag, 1987, Berlin, pp. 1 – 82.
62. Sarradj, E., *Energy-based vibroacoustics: SEA and beyond*, http://www.akustik-forschung.de/pdf/sarradj_SEA_DAGA2004.pdf.
63. Scarpa, F. and Tomlin, P. J., *On the transverse shear modulus of negative Poisson's ratio honeycomb structures*. Fatigue and Fracture in Engineering Materials and Structures, 23, 2000, pp. 717-720.
64. Scarpa, F., Blain, S., Lew, T. L., Perrot, D., Ruzzene, M. and Yates, J. R., *Elastic buckling of hexagonal chiral cell honeycombs*, accepted in Journal of Composite A, 2005.
65. Scarpa, F., Burriesci, G., Smith, F. C. and Chambers, B., *Mechanical and electromagnetic behaviour of auxetic honeycomb structures*, The Aeronautical Journal, March 2003, pp 175-183.
66. Sigmund, O. and Torquato, S., *Design of smart composite materials using topology optimisation*, Smart Material Structure, 8, 1999, pp. 365 – 379.
67. Simkins, V. R., Alderson, K. L., Alderson, A. and Davies, P. J., *Auxetic fibres anchoring composite materials*, Abstract of Papers – Auxetic Materials Network, 2004, Bolton, pp. 28 – 31.
68. Soule, T. and Heckendorn, R. B., *An analysis of the causes of code growth in genetic programming*, Genetic Programming and Evolvable Machines, vol. 3, 2002, pp. 283-309.
69. Spadoni, A., Ruzzene, M. and Scarpa, F., *Global and local linear buckling behaviour of chiral cellular structure*, Physica Status Solidi B, 242(3), 2005, pp. 695 – 709.
70. Srinivasan, A.V. and McFarland, D.M., *Smart structures: analysis and design*, Cambridge University Press, 2001, New York.
71. Storn, R. and Price, K., *Differential evolution – A simple and efficient heuristic for global optimisation and continuous spaces*, Journal of Global Optimisation, 11, 1997, pp. 341 – 359.
72. Stott, P. J., Mitchell, R., Alderson, K. and Alderson, A., *Auxetic materials – an introduction*, http://www.azom.com/Details.asp?ArticleID=167, January 2002.
73. Stott, P. J., Mitchell, R., Alderson, K. and Alderson, A., *Auxetic materials – applications*, http://www.azom.com/Details.asp?ArticleID=168, January 2002.
74. Sun Microsystems Inc., *The Java Tutorial, 3rd. Edition*, Sun Microsystems Inc., 1995 – 2005, http://java.sun.com/docs/books/tutorial/information/download.html
75. Swan, C. C. and Cakmak, A. S., *Homogenisation and effective elastoplasticity models for periodic composites*, Communications in Numerical Methods in Engineering, 10, 1994, pp. 257 – 265.
76. Swan, C. C. and Kosaka, I., *Homogenisation-based analysis and design of composites*, Computers and Structures, 64(1-4), 1997, pp. 603 – 621.
77. The Mathworks Inc., *Optimisation toolbox for use with MATLAB: User guide version 2*, The Mathworks Inc., 1999, MA, pp. 4.32 – 4.44.

78. Watson, A. H. and Parmee, I. C., *System Identification Using Genetic Programming*, Proceedings of 2nd International Conference on Adaptive Computing in Engineering Design and Control (PEDC), 1996, pp. 248-255.
79. Whitney, J. M., *Structural analysis of laminated anistropic plates*, Technomic Publishing Co. Ltd., 1987, USA, pp. 87 – 125.
80. Worden, K. and Tomlinson, G. R., *Nonlinearity in structural dynamics detection - identification and modelling*, IOP Publishing Ltd., 2001, UK, pp. 566 – 600.
81. Worden, K., *MLP – Multi-Layer Perceptron version 3.4 – A user's manual*.
82. Yang, W., Li, Z. M., Shi, W., Xie, B. H., Yang, M. B., 2004, *Review on Auxetic Materials*, Journal of Mateiral Science, 39, pp. 3269 – 3279.
83. Yoo, J. H., Kikuchi, N. and Volakis, J. L., *Structural optimisation in magnetic devices by the homogenisation method*, IEEE Trasaction on Magnetics, 36(3), 2000, pp. 574-580.
84. Young, W.C., *Roark's formulas for stress and strain, 6th. Edition*, McGraw-Hill, 1989, London.
85. Zhang, L., Jack, L. B. and Nandi, A. K., *Fault detection using genetic programming*, University of Liverpool, Department of Electrical Engineering and Electronics, 2004.

Appendix 1
Bi-cubic polynomial for void model

The polynomial listed below is taken from [27], which was obtained from least square fitting. The bi-cubic polynomial was used in Chapter 4 as a comparison between NN and GP.

i. Bi-cubic complete polynomial

$$C_{11}^{H}(w,h) = 1.03470 - 0.19507w - 0.44185h + 0.17949w^2 - 0.83233wh - 0.46685h^2 - 0.08561w^3 + 0.19546w^2h + 0.68235wh^2 - 0.04131h^3$$

$$C_{12}^{H}(w,h) = 0.31345 - 0.12080w - 0.12080h - 0.17639w^2 - 0.26954wh - 0.17639h^2 + 0.00835w^3 + 0.27454w^2h + 0.27454wh^2 + 0.00835h^3$$

$$C_{22}^{H}(w,h) = 1.03470 - 0.44815w - 0.19507h - 0.46682w^2 - 0.83233wh + 0.17949h^2 - 0.04131w^3 + 0.68235w^2h + 0.19546wh^2 - 0.08561h^3$$

$$C_{33}^{H}(w,h) = 0.37940 - 0.17684w - 0.17684h + 0.12103w^2 - 1.03828wh + 0.12103h^2 - 0.24676w^3 + 0.65449w^2h + 0.65449wh^2 - 0.24676h^3$$

ii. Bi-quartic complete polynomial

$$C_{11}^{H}(w,h) = 1.00382 - 0.01512w - 0.00878h - 0.04857w^2 - 1.77977wh - 2.21837h^2 - 0.08208w^3 + 1.46200w^2h + 1.26972wh^2 + 2.71675h^3 + 0.04045w^4 - 0.16884w^3h - 1.01329w^2h^2 + 0.28394wh^3 - 1.45002h^4$$

$$C_{12}^{H}(w,h) = 0.29676 + 0.05085w + 0.05085h - 0.73751w^2 - 0.75721wh - 0.73751h^2 + 0.77797w^3 + 0.77154w^2h + 0.77154wh^2 + 0.77797h^3 - 0.37826w^4 - 0.02620w^3h - 0.45770w^2h^2 - 0.02620wh^3 - 0.37826h^4$$

$$C_{22}^{H}(w,h) = 1.00382 - 0.00878w - 0.01512h - 2.21837w^2 - 1.77977wh - 0.04857h^2 + 2.71675w^3 + 1.26972w^2h + 1.46200wh^2 - 0.08208h^3 - 1.45002w^4 + 0.28394w^3h - 1.01329w^2h^2 - 0.16884wh^3 + 0.04045h^4$$

$$C_{33}^{H}(w,h) = 0.33684 + 0.10190w + 0.10190h - 0.11496w^2 - 2.93878wh - 0.11496h^2 - 0.36793w^3 + 2.5907w^2h + 2.59067wh^2 - 0.367793h^3 + 0.08564w^4 - 0.10022w^3h - 1.78586w^2h^2 - 0.10022wh^3 + 0.08564h^4$$

Appendix 2
Chiral triangle calculations.

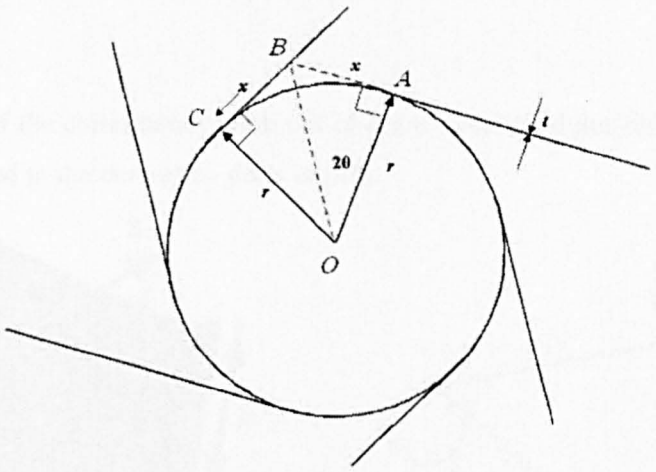


Figure A 2. 1 Gap appeared at the joints between chiral ligaments and the cylinder.

Considering the figure above, the ratio between the radius r and segment x is given by:

$$\frac{r}{x} = \tan(2\theta)$$

where $\theta = 30°$. The circular segment $A\hat{B}C$ is calculated as

$$A_{A\hat{B}C} = \frac{r^2}{\tan(2\theta)}$$

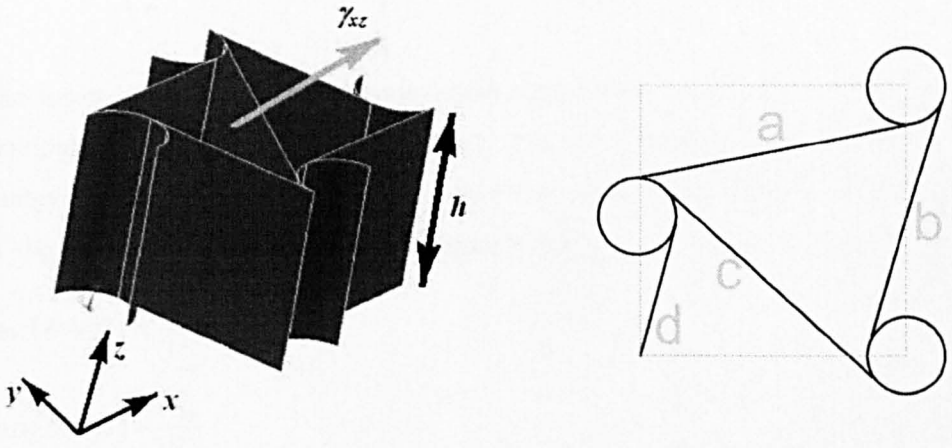
From the equation above, the area of the polygon $OA\hat{B}C$ can be written as:

$$A_{OA\hat{B}C} = r^2\left[2\theta + \frac{1}{\tan(2\theta)}\right]$$

The area $A\hat{B}C$ only covers roughly 27% of the total area $OA\hat{B}C$. When computing the area of the chiral unit cell, it is therefore reasonable to neglect its contribution [64].

Appendix 3
Chiral honeycomb out-of-plane shear modulus – Analytical derivations.

The upper bound of the chiral honeycomb out-of-plane shear modulus detailed derivation is as below, and is refered to the derivation done in [49]:



(a) (b)

Figure A 3.1 (a) Shear stress imposed in *xz* plane. (b) Ligaments of the chiral cell

The upper bound of the chiral honeycomb is calculated using the minimum potential energy [23, 50]. As shown in Figure A 3.1(a), a shear stress τ_{xz} causing a deformation γ_{xz} is acting on the face of the unit cell in the *x*-direction. The bending is neglected, as almost all the elastic strain energy is stored in cell walls during the shear displacements. It is also assumed that the cylinders do not carry any shear or bending during the deformation. Therefore the shear strain in wall *a*, *b* and *c* (Figure A 3.1 (b)) can be written as:

$$\begin{aligned} \gamma_a &= \gamma_{xz}\cos(\theta-\varphi) \\ \gamma_b &= \gamma_{xz}\cos(\theta+\varphi) \\ \gamma_c &= \gamma_{xz}\sin\varphi \end{aligned} \qquad \textbf{(A 3.1)}$$

The expression of the minimum potential energy has the form:

$$\frac{1}{2}G_{xz}\gamma_{xz}^2 V \le \frac{1}{2}\sum_i \left(G_c\gamma_i^2 V_i\right) \qquad \textbf{(A 3.2)}$$

which lead to

$$\frac{1}{2}G_{xz}\gamma_{xz}{}^2 V \le \frac{1}{2}G_s\left(\gamma_a{}^2 V_a + \gamma_b{}^2 V_b + \gamma_c{}^2 V_c\right) \qquad \textbf{(A 3.3)}$$

Substituting (A 3.1) into (A 3.3), yields

$$\left(\frac{1}{2}G_{xz}\gamma_{xz}^{\ 2}\right)2b\left(\frac{1}{2}L^2\cos(\theta)+\frac{\pi r^2}{2}\right)\le\frac{1}{2}G_s\begin{pmatrix}\gamma_{xz}^{\ 2}\cos^2(\theta-\beta)bLt\\+\gamma_{xz}^{\ 2}\cos^2(\theta+\beta)bLt\\+\gamma_{xz}^{\ 2}\sin^2(\beta)bLt\end{pmatrix}$$

$$\Rightarrow\frac{G_{xz}}{G_s}\le\frac{bLt}{2\left(\frac{1}{2}L^2\cos(\theta)+\frac{\pi r^2}{2}\right)b}\left(\cos^2(\theta-\beta)+\cos^2(\theta+\beta)+\sin^2(\beta)\right) \quad \textbf{(A 3.4)}$$

The lower bound of the chiral honeycomb out-of-plane shear modulus involves some tedious manipulation of equations. The reference is also taken from [49]. The minimum complementary energy method is used here. With the shear loading shown in Figure A 3.1(a), the components of shear force F for the unit cell are (Figure A 3.2):

$$\tan(\theta-\beta)=\frac{F_{ay}}{F_{ax}}$$

$$\tan(\theta+\beta)=\frac{F_{by}}{F_{bx}} \quad \textbf{(A 3.5)}$$

$$\tan\beta=\frac{F_{cx}}{F_{cy}}$$

The minimum complementary energy theorem can be expressed as an inequality:

$$\frac{1}{2}\frac{\tau_{xz}}{G_{xz}}V\le\frac{1}{2}\sum_i\left(\frac{\tau_i}{G_c}V_i\right) \quad \textbf{(A 3.6)}$$

where τ_i is the shear strain in each of the wall of volume V_i. According to the assumption stated before, that the shear loading was mainly carried by the walls (verified in Figure 6.14), i.e. the G_{xz} can be developed based on the four walls *a*, *b*, *c* and *d* (Figure A 3.1(b)). Also the symmetry of the cell indicates that walls *c* and *d* each carry the same load, therefore (A 3.6) is applied only to walls *a*, *b* and *c*. The summation of τ_iV_i is the resultant shear force F.

Therefore the equation above becomes:

$$\frac{1}{2}\frac{\tau_{xz}^2}{G_{xz}}V\le\frac{1}{2\times G_s}\left(\tau_a^2V_a+\tau_b^2V_b+\tau_c^2V_c\right) \quad \textbf{(A 3.7)}$$

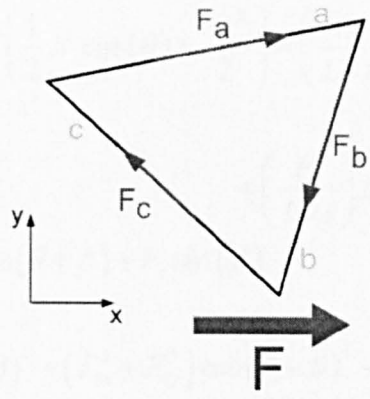


Figure A 3.2 Flow of forces in the chiral honeycomb cell wall as a result of shear in *xz* direction.

From the figure above, a set of relation can be derived:

$$\begin{cases} F_{ax} + F_{bx} + F_{cx} = F \\ F_{ay} + F_{by} + F_{cy} = 0 \end{cases} \quad \textbf{(A 3.8)}$$

where each of these component can be seen as:

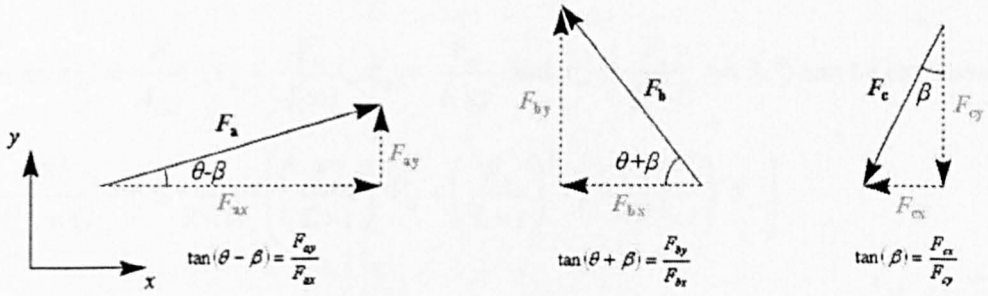


Figure A 3.3 Force component of the chiral honeycomb cell wall, during shear at *xz* direction.

Therefore, (A 3.8) can be written as:

$$\begin{cases} F_{ax} + F_{bx} + F_{cx} = F \\ F_{ax} \times \tan(\theta - \beta) + F_{bx}.\tan(\theta + \beta) + F_{cx} \times \dfrac{1}{\tan(\beta)} = 0 \end{cases} \quad \textbf{(A 3.9)}$$

To solve (A 3.9), an equilibrium of external stress is formulated as below:

$$\tau_{xz} \times 2\left(\frac{1}{2}L^2\cos(\theta) + \frac{\pi r^2}{2}\right) = \tau_a \times L \times t \times \cos(\theta - \beta) + \tau_b \times L \times t \times \cos(\theta + \beta) + \tau_c \times L \times t \times \sin(\beta)$$

$$\Rightarrow \frac{F}{2\left(\frac{1}{2}L^2\cos(\theta)+\frac{\pi r^2}{2}\right)}\times 2\left(\frac{1}{2}L^2\cos(\theta)+\frac{\pi r^2}{2}\right)=\left(\frac{F_a}{L\times t}\right)L\times t\times\cos(\theta-\beta)$$

$$+\left(\frac{F_b}{L\times t}\right)L\times t\times\cos(\theta+\beta)+\left(\frac{F_c}{L\times t}\right)\times\sin(\beta)$$

$$\Rightarrow F=F_a\cos(\theta-\beta)+F_b\cos(\theta+\beta)+F_c\sin(\beta)$$

$$\Rightarrow F^2=\left(F_{ax}^2+F_{ay}^2\right)\cos(\theta-\beta)^2+\left(F_{bx}^2+F_{by}^2\right)\cos(\theta+\beta)^2+\left(F_{cx}^2+F_{cy}^2\right)\sin(\beta)^2$$

with the triangulation of the force components shown in Figure A 3.3, the above equilibrium stress can be simplified as:

$$F^2=F_{ax}^2\left(\cos(\theta-\beta)^2+\sin(\theta-\beta)^2\right)+F_{bx}^2\left(\cos(\theta+\beta)^2+\sin(\theta+\beta)^2\right)+F_{cx}^2\left(\sin(\beta)^2+\cos(\beta)^2\right)$$

$$\Rightarrow F^2=F_{ax}^2+F_{bx}^2+F_{cx}^2 \qquad \textbf{(A 3.10)}$$

Also as $\tau_{xz}=\frac{F}{A_{cell}}$, $\tau_a=\frac{F_a}{L\times t}$, $\tau_a=\frac{F_a}{L\times t}$ and $\tau_c=\frac{F_c}{L\times t}$, (A 3.7) can be expressed as

$$\frac{1}{2}\frac{F^2}{A_{cell}^2\times G_{xz}}V\le\frac{1}{2\times G_s}\left(\left(\frac{F_a}{L\times t}\right)^2V_a+\left(\frac{F_b}{L\times t}\right)^2V_b+\left(\frac{F_c}{L\times t}\right)^2V_c\right)$$

$$\Rightarrow\frac{G_{xz}}{G_s}\ge C\times\frac{F^2}{F_a^2+F_b^2+F_c^2},\ \text{where}\ C=\frac{Lt}{\left(2\times\left(\frac{1}{2}L^2\cos(\theta)+\frac{\pi r^2}{2}\right)\right)}\ \text{and}\ \begin{cases}\mathrm{F_a}=\sqrt{F_{ax}^2+F_{ay}^2}\\ \mathrm{F_b}=\sqrt{F_{bx}^2+F_{by}^2}\\ \mathrm{F_c}=\sqrt{F_{cx}^2+F_{cy}^2}\end{cases}$$

$$\Rightarrow\frac{G_{xz}}{G_s}\ge C\times$$

$$\frac{F^2}{\left(F_{ax}+F_{bx}+F_{cx}\right)^2-2\left(F_{ax}F_{bx}+F_{ax}F_{cx}+F_{bx}F_{cx}\right)+\left(F_{ay}+F_{by}+F_{cy}\right)^2-2\left(F_{ay}F_{by}+F_{ay}F_{cy}+F_{by}F_{cy}\right)}$$

$$\Rightarrow\frac{G_{xz}}{G_s}\ge C\times\frac{F^2}{F_{ax}^2+F_{bx}^2+F_{cx}^2+F_{ay}^2+F_{by}^2+F_{cy}^2}$$

$$\Rightarrow\frac{G_{xz}}{G_s}\ge C\times\frac{F^2}{F^2-2\left(F_{ax}F_{bx}+F_{ay}F_{by}+F_{ax}F_{cx}+F_{ay}F_{cy}+F_{bx}F_{cx}+F_{by}F_{cy}\right)}$$

where each components of F can be solved from the formulae described in (A 3.9) and (A 3.10).

Appendix 4
Chiral shear modulus experimental post-processing

As mentioned in Chapter 6, where the out-of-plane shear modulus of the chiral honeycomb depends on the following relationship:

$$\delta_{total} = \delta_{bending} + \delta_{shear} = \frac{FL^3}{48D} + \frac{FL}{4AG}$$

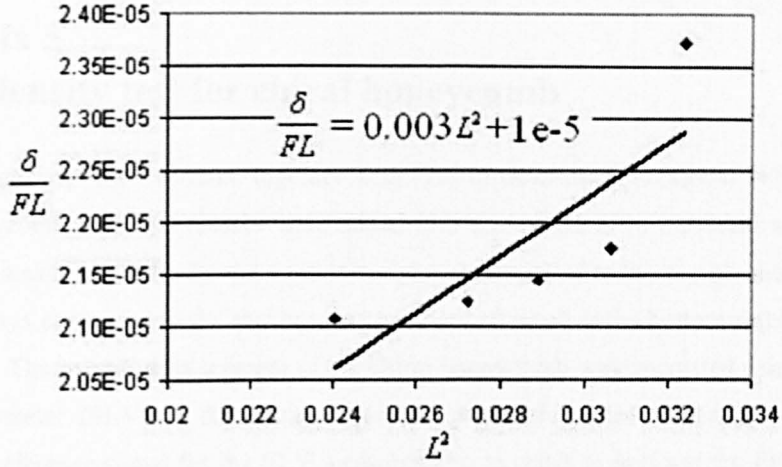
and that by rearranging the equation, the shear modulus can be obtained from the slope of the following plots:

$$\begin{cases} \dfrac{\delta}{FL} = \dfrac{L^2}{48D} + \dfrac{1}{4AG} \\ \dfrac{\delta}{FL^3} = \dfrac{1}{48D} + \dfrac{1}{4AG} \cdot \dfrac{1}{L^2} \end{cases}$$

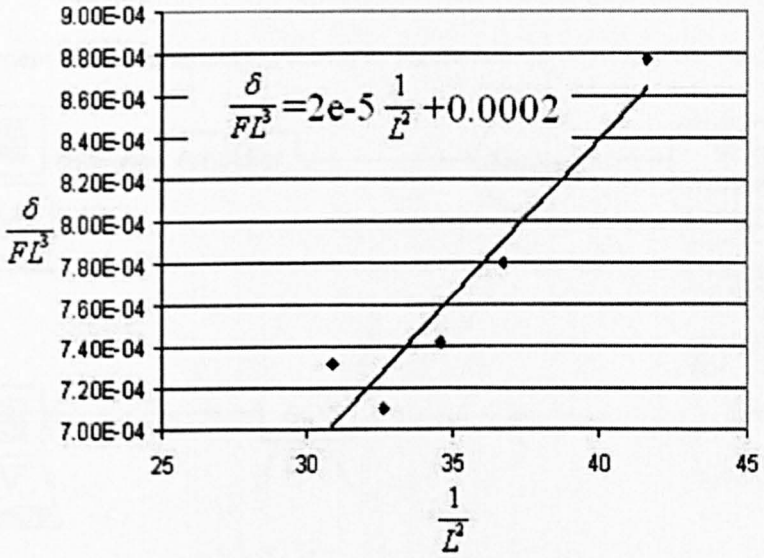
The data is too large to be included in this work, explanation of the post-processing procedures, which helps to calculate the effective out-of-plane shear modulus will be described in words.

The displacement (up till 1mm) collected in the three-point bending test was plotted in a load (kN) versus displacement (mm) graph in order to obtained the best fit straight line and its corresponding slope. The displacements collected from each of the span lengths, from 150 mm – 180 mm, were plotted in order to find the slope of the graph, there are 7 slopes gathered here. These slopes represent $\frac{F}{\delta}$, inverting the values of the slopes will give $\frac{\delta}{F}$.

Then, with the inversed slopes, further divide it to become $\frac{\delta}{FL}$ and $\frac{\delta}{FL^3}$. Plot these two results against L^2 and $1/L^2$ respectively, as shown in the figure below:



(a)



(b)

Figure A 4. 1 Plot of $\frac{\delta}{FL}$ vs. L^2 and plot of $\frac{\delta}{FL^3}$ versus $1/L^2$.

The slopes and intercepts of Figure A 4. 1 are listed in the table below, where $A = bd^2/c$, as mentioned in Chapter 6, is 1779.084 mm^2.

	From Figure A 4. 1(a)		From Figure A 4. 1(b)	
Slope	$\frac{1}{48D} = 2.69\times10^{-4}$	D = 77.32 Nm	$\frac{1}{4AG} = 1.50\times10^{-5}$	G = 9.54 MPa
Intercept	$\frac{1}{4AG} = 1.41\times10^{-5}$	G = 10.1 MPa	$\frac{1}{48D} = 2.39\times10^{-4}$	D = 87.2 Nm

Table A 4. 1 Slopes and intercepts obtained from Figure A 4. 1, in order to compute for the shear modulus (G) and flexural rigidity (D) of the chiral honeycomb.

Appendix 5
Modal density test for chiral honeycomb

The modal density test was done together with [29] to determine the modal behaviour of the chiral honeycomb core. No masses were added into the cylinders to represent sensor weight. The sample was hanging freely and a random signal was applied to the sample though a shaker, the sample was connected to the shaker using a stinger attached at the bottom right corner of the honeycomb. The overall displacement of the chiral honeycomb was measured using a Scanning Laser Vibrometer (SLV). A force transducer was attached between the shaker and sample, acting as a reference signal for the SLV measurement, in order to perform the FFT in the SLV program.

The arrangement of the experiment is shown in Figure A 6. 1.

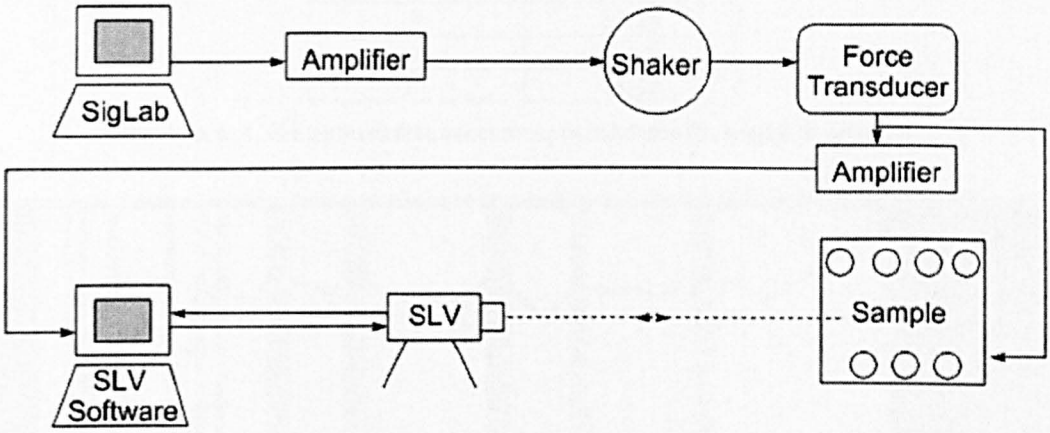


Figure A 6. 1 Instrument setup for the experiment.

The test was done with a mean frequency of 100Hz. The random signals were generated using SigLab from a computer.

The displacement was measured using the SLV, pointing at 250 points selected on the chiral honeycomb, in order to map the chiral honeycomb surface. Each point was scanned 10 times throughout the experiment and an average was taken from the 10 measurements, in order to obtain a good result. A Hanning window was chosen for sampling the displacement, sample time is 8s.

The specimen is made by Nylon 6 DuraForm PA powder, using laser sintering technology. The density of this powder is 950kgm^{-3}. The dimension of the test sample used in this experiment is

100 mm × 100mm with depth of 20mm. The mass of the test sample is 43g without the stinger attached at the bottom right, 43.86g with the stinger attached.

The result is shown in the table below, the peaks were obtained from the H1 window as shown in Figure A 6. 2. A sharp spike at 50Hz was due to the interference of the ambient wiring connection.

Max. Frequency 100 Hz	
Peak No.	Frequency [Hz]
1	3.25
2	5.125
3	8.625
4	15.25
5	18.875
6	21
7	27.5
8	36.75
9	53.75
10	63.5
11	78.25

Table A 6. 1 The natural frequencies captured from the modal density test.

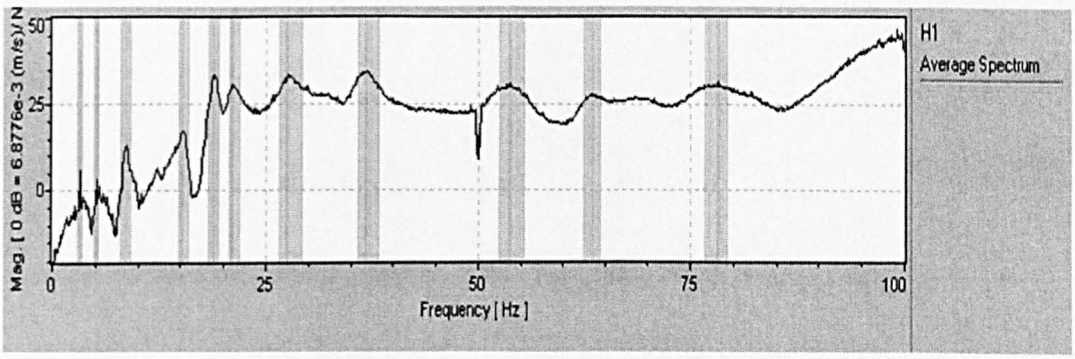


Figure A 6. 2 Frequency response function of the modal density test for chiral honeycomb.

The result shows some success in using the SLV to determine the modal density of the chiral honeycomb core, which can benchmark for future exploration of how the sensor weight affects the dynamics behaviour of the chiral honeycomb.

The mode shapes captured from the experiment are in animation form, some snapshots of the mode shape at 3.25Hz are shown below.

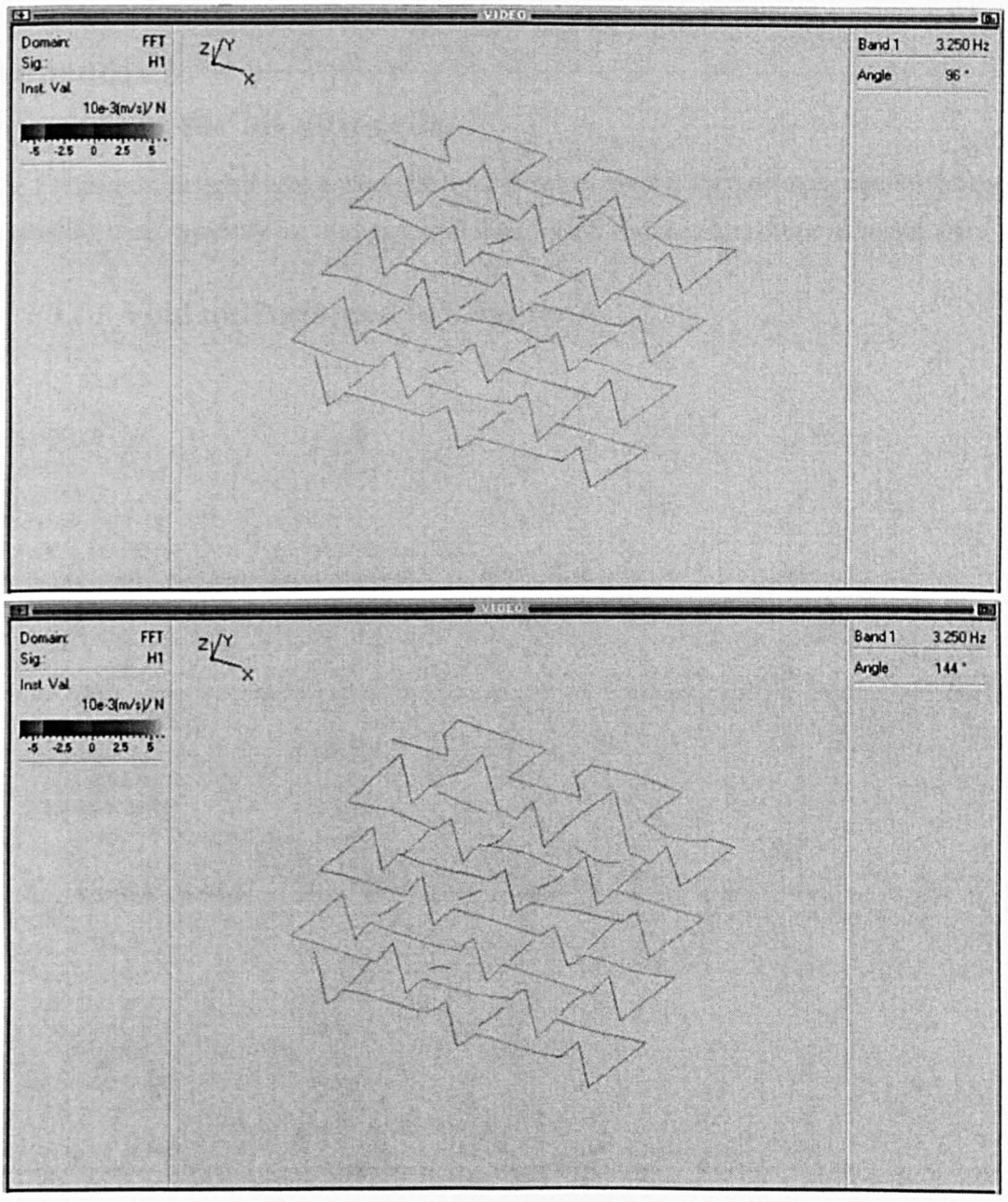


Figure A 6. 3 Some snapshots of the chiral honeycomb sample at 3.25Hz.

Appendix 6
FE macros for all unit cells.

The FE macros included here are not the full macros as most of the boundary conditions are repeatedly used, typically for the hexagonal honeycomb and the chiral honeycomb model.

i. Void unit cell used in Chapter 4

```
Es=7e10
nus=0.33333
stepi=10
stepj=10
/prep7
!*hght=0.1
!*wdth=0.1
!**ask,wdth,width of the hole,0.1
!**ask,hght,height of the hole,0.9
!*stepi=10
!*stepj=10
!*Es=26.6672
!*Es=7e10
!*nus=0.33333
!* q0=100
*do,i,0,stepi,1
      k,,i,0
      *do,j,1,stepj,1
            k,,i,j
      *enddo
*enddo
a_1=1
b_1=stepj+2
c_1=b_1+1
d_1=2
a1=a_1
b1=b_1
c1=c_1
d1=d_1
a01=a1
b01=b1
c01=c1
d01=d1
*do,i,1,stepi
  *do,j,1,stepj
      a,a01,b01,c01,d01
      a01=d01
      b01=c01
      d01=a1+1+j
      c01=c1+j
  *enddo
      a1=b1
      d1=c1
      b1=b_1-1+a1
      c1=b_1+a1
      a01=a1
      b01=b1
      c01=c1
      d01=d1
*enddo
```

```
*do,i,1,stepi,1
      *do,j,1,stepj,1
            k,,(i-1)+(1-wdth)/2,(j-1)+(1-hght)/2
            k,,(i-1)+(1+wdth)/2,(j-1)+(1-hght)/2
            k,,(i-1)+(1-wdth)/2,(j-1)+(1+hght)/2
            k,,(i-1)+(1+wdth)/2,(j-1)+(1+hght)/2
      *enddo
*enddo
e1=(stepi+1)*(stepj+1)+1
f1=e1+1
g1=f1+2
h1=f1+1
e01=e1
f01=f1
g01=g1
h01=h1
*do,i,1,stepi*stepj
      a,e01,f01,g01,h01
      e01=h01+2
      f01=e01+1
      g01=f01+2
      h01=f01+1
*enddo
*do,j,1,stepi*stepj
      asba,j,(stepi*stepj)+j
*enddo
nummrg,all
numcmp,all
aglue,all
ET,1,SHELL63
!*keyopt,1,1,0
!*keyopt,1,2,0
!*keyopt,1,3,2
!*keyopt,1,5,0
!*keyopt,1,6,2
r,1,0.2
mp,ex,1,Es
mp,nuxy,1,nus
smrtsize,4
type,1
mat,1
amesh,all
nummrg,all
numcmp,all
nsel,s,loc,x,0
nsel,a,loc,x,10
nsel,a,loc,y,0
nsel,a,loc,y,10
d,all,uz,0.0
allsel,all
*get,tot_ele,elem,0,count,
!*f,all,fz,-100/tot_ele
f,all,fz,-100/(100-wdth*hght*100)
!*sfa,all,1,pres,-100/(100-wdth*hght*100)
/solu
solve
/post1
set,1,1
etable,uz,u,z
!*pretab,uz
esort,etab,uz,1,0
*get,uz_min,sort,0,min
*get,uz_max,sort,0,max
*get,loc_min,sort,0,imin
```

```
*get,loc_max,sort,0,imax
uz_ans(count,1)=wdth
uz_ans(count,2)=hght
uz_ans(count,3)=uz_min
uz_ans(count,4)=uz_max
uz_ans(count,5)=loc_min
uz_ans(count,6)=loc_max
*sta
fini
```

ii. Honeycomb unit cell used in Chapter 5

```
*afun,deg
beta=0.41
alph = 1
theta = -30
Es = 7.0e9
nus = 0.33
L = 1
*dim,E_hexa,array,50,5
*afun,deg
th=theta
a=alpha
l=L
h=l*a
t=beta*l
disp=l*0.01
c1=t/cos(th)
c2=-t/2*tan(th)+c1/2
ltr=sqrt((t/2)**2+((c1/2)+c2)**2)
*set,wdth,2*l*cos(th)
*set,hgth,((l*sin(th))+h)
mc=1.6                          !*mesh refining factor
nt=6
nl=nint((nt/(beta*mc)))
na=nint((a*nl))
nc=nint(((ltr*nt)/t))
nr1=nint((nt*((c1/t)/(mc*2))))
nr2=nint((nt/(mc*2)))
/prep7
k,1,(l/2)*cos(th),l/2*sin(th)
k,2,(l/2)*cos(th),(l/2)*sin(th)-c1/2
k,3,(l/2)*cos(th),(l/2)*sin(th)+c1/2
k,4,l*cos(th),l*sin(th)
k,5,l*cos(th),l*sin(th)-c1/2
k,6,l*cos(th)-t/2,l*sin(th)+c2
k,7,l*cos(th)+t/2,l*sin(th)+c2
k,8,l*cos(th),l*sin(th)+(h/2)
k,9,l*cos(th)-t/2,l*sin(th)+(h/2)
k,10,l*cos(th)+t/2,l*sin(th)+(h/2)
k,11,2*l*cos(th),0
k,12,2*l*cos(th),c1/2
k,13,2*l*cos(th)-t/2,-c2
k,14,2*l*cos(th)+t/2,-c2
k,15,(5/2)*l*cos(th),(l/2)*sin(th)
k,16,(5/2)*l*cos(th),(l/2)*sin(th)+c1/2
k,17,(5/2)*l*cos(th),(l/2)*sin(th)-c1/2
k,18,2*l*cos(th),-h/2
k,19,2*l*cos(th)-t/2,-h/2
k,20,2*l*cos(th)+t/2,-h/2
l,1,2
l,1,3
l,2,5
l,3,6
```

```
l,7,10
l,6,9
l,8,10
l,8,9
l,7,12
l,5,13
l,12,16
l,14,17
l,15,17
l,15,16
l,14,20
l,13,19
l,18,19
l,18,20
l,6,5   !*triangles
l,5,7
l,7,6
l,12,13
l,13,14
l,14,12
mp,ex,1,Es
mp,nuxy,1,nus
mp,dens,1,2400
et,1,plane82
keyopt,1,3,0
keyopt,1,1,0       !*ELement coordinate system global
lesize,1,,,nr1     !*lines perpendicular to beams
lesize,2,,,nr1
lesize,7,,,nr2
lesize,8,,,nr2
lesize,13,,,nr1
lesize,14,,,nr1
lesize,17,,,nr2
lesize,18,,,nr2
lesize,19,,,nc !*triangles
lesize,20,,,nc
lesize,21,,,nt
lesize,22,,,nc
lesize,23,,,nt
lesize,24,,,nc
lesize,3,,,nl/2
lesize,4,,,nl/2
lesize,5,,,na/2
lesize,6,,,na/2
lesize,9,,,nl
lesize,10,,,nl
lesize,11,,,nl/2
lesize,12,,,nl/2
lesize,15,,,na/2
lesize,16,,,na/2
al,1,2,3,4,19   !*l areas
al,9,10,20,22
al,11,12,24,14,13
al,5,6,7,8,21   !*alpha areas
al,15,16,17,18,23
al,19,20,21   !*triangular areas
al,22,23,24
mshape,1,2D   !*triangular elements
asel,s,,,1,5
amesh,all
asel,all
mshape,1,2d
asel,s,,,6,7
amesh,all
```

```
asel,all
nummrg,all
numcmp,all
allsel,all
dk,1,ux,disp/2
dk,15,ux,-disp/2
dk,1,uy,0
dk,15,uy,0
ksel,s,kp,,1
nslk
*get,n1,node,,num,min
allsel,all
ksel,s,kp,,15
nslk
*get,n15,node,,num,min
allsel,all
lsel,s,line,,1
nsll,s,1
nsel,u,node,,n1
*get,num5,node,0,count
*dim,mat3,array,num5,2
*do,count5,1,num5,1
  *get,mat3(count5,1),node,,num,min
  nsel,u,node,,mat3(count5,1)
*enddo
allsel,all
*dim,place3,array,num5,1
lsel,s,line,,2
nsll,s,1
*do,count6,1,num5,1
  place3(count6,1)=NY(mat3(count6,1))
  mat3(count6,2)=NODE(0,l*sin(th)-place3(count6,1),0)
  ce,(count6+1),0,mat3(count6,1),uy,1,mat3(count6,2),uy,1
  ce,(count6+num5+1),0,mat3(count6,1),ux,1,mat3(count6,2),ux,1,n1,ux,-
2
*enddo
allsel,all

!!!A LARGE AMOUNT OF PERIODIC CONSTRAINT EQUATIONS WERE OMITTED HERE!!!

sbctran
fini
/solu
solve
/post1
set,first
lsel,s,line,,1,2
nsll,s,1
fsum
*get,fx_1,fsum,0,item,fx
*get,fy_1,fsum,0,item,fy
allsel,all
lsel,s,line,,13,14
nsll,s,1
fsum
*get,fx_2,fsum,0,item,fx
*get,fy_2,fsum,0,item,fy
allsel,all
ksel,s,kp,,8
nslk,s
*get,n8,node,,num,min
allsel,all
uy_1=UY(n8)
```

```
fx_av=(fx_2-fx_1)/2
s_X=fx_av/hgth
str_x=disp/wdth
str_y=2*uy_1/hgth
nu12r=str_y/str_x
E1r=s_x/str_x
E_hexa(i,1)=alpha
E_hexa(i,2)=beta
E_hexa(i,3)=th
E_hexa(i,4)=E1r
E_hexa(i,5)=nu12r
======================Buckling================================
/prep7           !* Parameters of the panel
*afun,deg
teta = theta
*SET,gam,1
*SET,t,beta*L
*SET,h,L*alpha
*SET,b,gam*L
k,1,0,(h/2)+L*sin(teta)
k,2,(L*cos(teta))/2,(h/2)+L*sin(teta)/2
k,3,-L*cos(teta),0
k,4,-L*cos(teta),h/2
kgen,2,1,,,,h/2
kgen,2,4,,,-L*cos(teta)/2,L*sin(teta)/2
l,1,5
l,1,2
l,1,4
l,4,3
l,4,6
!* Extrusion of plane cell
lgen,2,all,,,,,b,,0,0
l,1,7
l,2,9
l,5,8
l,4,10
l,3,11
l,6,12
al,12,2,11,7
al,11,1,13,6
al,11,3,14,8
al,14,5,16,10
al,14,9,15,4
nummrg,all,1.e-12
numcmp,all
ET,1,SHELL93
R,1,t,,,,,,
MP,DENS,1,724  !density
MP,EX,1,Es
MP,PRXY,1,nus
!* MESH
type,1
real,1
amesh,all
area_c=2*(2*h*L*sin(teta)+2*sin(teta)*cos(teta)*L**2)
nsel,s,loc,z,0
d,all,ux,0
d,all,uy,0
d,all,uz,0
allsel,all
lsel,s,line,,16,,,,1
nsll,s,1
*get,n_cal01,node,0,count
*dim,ary_01,array,n_cal01,2
```

```
*do,count1,1,n_cal01,1
      *get,ary_01(count1,1),node,,num,min
      nsel,u,node,,ary_01(count1,1)
*enddo
*dim,locat1,array,n_cal01,1
lsel,s,line,,12,,,,1
nsll,s,1
*do,count2,1,n_cal01,1
      locat1(count2,1) = nz(ary_01(count2,1))
      ary_01(count2,2)=node((L*cos(teta))/2,(h/2)+L*sin(teta)/2,locat1
(count2,1))
      ce,count2,0,ary_01(count2,1),ux,1,ary_01(count2,2),ux,-1
      ce,n_cal01+count2,0,ary_01(count2,1),uy,1,ary_01(count2,2),uy,-1
      ce,2*n_cal01+count2,0,ary_01(count2,1),rotz,1,ary_01(count2,2),r
otz,-1
*enddo
allsel,all
ce_no = n_cal01*100

!!!A LARGE AMOUNT OF PERIODIC CONSTRAINT EQUATIONS WERE OMITTED HERE!!!

/solu
antype,static
pstres,on
allsel,all
solve
fini
/solu
antype,buckle
bucopt,subsp,10
outpr,nsol,all
save
solve
fini
/post1
set,1,3
*get,f_buck,active,0,set,freq
sig_c=abs(f_buck)/(area_c*1.e6)
!* Gibson & Ashby results
c1=2/(1-nus**2)
c2=(2+(1/alpha))/(cos(teta)*(alpha+sin(teta)))
sig_ga=Es*c1*c2*beta**3/1.e6
E_hexa(i,1) = alpha
E_hexa(i,2) = beta
E_hexa(i,3) = theta
E_hexa(i,4) = f_buck
E_hexa(i,5) = sig_c
E_hexa(i,6) = sig_ga
fini
```

iii. Chiral honeycomb unit cell used in Chapter 6

```
!* Parameter of macro: Chiri2b.mac
!* R        distance between the axes of the circle [m]
!* radius   circles radius [m]
!* teta           degree between le line that joint the circle axes
[gradi ...centigradi]
!* t        wall thick [m]
!* thick    panel thick [m]
!* span
*afun,rad
pi=acos(-1)
*afun,deg
```

```
!*A=1
!*B=0.05
!*C=10
!*R=10/1000
beta = atan(2/A)
!*radius=4.5/1000
R = (2*radius)/sin(beta)
L = A*radius
teta=30
!*thick=1*R
thick=A*C*radius
!*t=0.05*radius
t=B*radius
/prep7
ffi=asin(radius/(R*sin(teta)))!* tangent degree between the central
cirlces
k,1,0,0,0
k,2,R*cos(teta),R*sin(teta),0
k,3,R*cos(teta),-R*sin(teta),0
k,4,2*R*cos(teta),0,0
!*beta=asin(2*radius/R)
beta = atan(2/A)
gamma=teta-beta   !* upper tangent degree of the first circles -90
k,5,radius*cos(90+gamma),radius*sin(90+gamma),0
k,6,radius*cos(90+gamma-2*teta),radius*sin(90+gamma-2*teta),0
k,7,R*cos(teta)+radius*cos(-90+gamma),R*sin(teta)+radius*sin(-
90+gamma),0
k,8,R*cos(teta)+radius*cos(-ffi),R*sin(teta)+radius*sin(-ffi),0
k,9,R*cos(teta)+radius*cos(gamma+90-
2*teta),R*sin(teta)+radius*sin(gamma+90-2*teta),0
k,10,R*cos(teta)+radius*cos(90+gamma),-
R*sin(teta)+radius*sin(90+gamma),0
k,11,R*cos(teta)+radius*cos(180-ffi),-R*sin(teta)+radius*sin(180-
ffi),0
k,12,R*cos(teta)+radius*cos(gamma+270-2*teta),-
R*sin(teta)+radius*sin(gamma+270-2*teta),0
k,13,2*R*cos(teta)+radius*cos(-90+gamma),radius*sin(-90+gamma),0
k,14,2*R*cos(teta)+radius*cos(-90+gamma-2*teta),radius*sin(-90+gamma-
2*teta),0
k,15,radius*cos(-90+gamma),radius*sin(-90+gamma),0
k,16,R*cos(teta)+radius*cos(gamma-90-
2*teta),R*sin(teta)+radius*sin(gamma-90-2*teta),0
k,17,R*cos(teta)+radius*cos(gamma+90-2*teta),-
R*sin(teta)+radius*sin(gamma+90-2*teta),0
k,18,2*R*cos(teta)+radius*cos(90+gamma),radius*sin(90+gamma),0
k,19,radius*cos(-ffi),radius*sin(-ffi),0
!*k,20,radius*cos(180-ffi),-2*R*sin(teta)+radius*sin(180-ffi),0
k,21,2*R*cos(teta)+radius*cos(180-ffi),radius*sin(180-ffi),0
k,20,0.5*(radius*cos(-ffi)+radius*cos(180-ffi)),0.5*(radius*sin(-
ffi)+radius*sin(180-ffi)-2*R*sin(teta)),0
mdx=0.5*(radius*cos(-ffi)+radius*cos(180-ffi))+2*R*cos(teta)
mdy=0.5*(radius*sin(-ffi)+radius*sin(180-ffi)-
2*R*sin(teta))+2*R*sin(teta)
k,22,mdx,mdy,0
l,5,7              !* line 1
l,8,11                  !* line 2
l,6,12                  !* line 3
l,9,14                  !* line 4
l,10,13                 !* line 5
l,19,20                 !* line 6
l,21,22                 !* line 7
larc,15,19,1,radius     !* line 8
larc,19,6,1,radius      !* line 9
larc,6,5,1,radius !* line 10
```

```
larc,16,7,2,radius      !* line 11
larc,7,8,2,radius !* line 12
larc,8,9,2,radius !* line 13
larc,17,10,3,radius     !* line 14
larc,10,11,3,radius     !* line 15
larc,11,12,3,radius     !* line 16
larc,13,14,4,radius     !* line 17
larc,14,21,4,radius     !* line 18
larc,21,18,4,radius     !* line 19
angolo10=2*teta                         !* 10
angolo9=abs(90+gamma-2*teta+ffi)        !* 9
angolo8=180-angolo10-angolo9            !* 8
angolo17=angolo10               !* 17=10
angolo19=angolo8                !* 19=8
angolo18=angolo9                !* 18=9
angolo15=abs(-90+gamma+ffi)             !* 15
angolo16=abs(-ffi-(gamma+90-2*teta))        !* 16
angolo14=180-angolo15-angolo16              !* 14
angolo13=abs(gamma-2*teta-(-90-ffi))        !* 13=16
angolo12=abs(gamma-(90-ffi))            !* 12=15
angolo11=180-angolo13-angolo12              !* 11=12
lunarc8=(angolo8*2*pi/360)*radius   !* lenght 8
lunarc9=(angolo9*2*pi/360)*radius   !* lenght 9
lunarc10=(angolo10*2*pi/360)*radius !* lenght 10
lunarc11=(angolo11*2*pi/360)*radius !* lenght 11
lunarc12=(angolo12*2*pi/360)*radius !* lenght 12
lunarc13=(angolo13*2*pi/360)*radius !* lenght 13
lunarc14=(angolo14*2*pi/360)*radius !* lenght 14
lunarc15=(angolo15*2*pi/360)*radius !* lenght 15
lunarc16=(angolo16*2*pi/360)*radius !* lenght 16
lunarc17=(angolo17*2*pi/360)*radius !* lenght 17
lunarc18=(angolo18*2*pi/360)*radius !* lenght 18
lunarc19=(angolo19*2*pi/360)*radius !* lenght 19
*if,teta,ge,50,then
      lunbase=lunarc9
*elseif,teta,le,20,then
      lunbase=lunarc10
*else
      lunbase=lunarc10/2
*endif
ndiv_9=nint(lunarc9/lunbase)  !* middle arc of 1st semicircle on left
& all the same
lesize,9,,,ndiv_9
lesize,18,,,ndiv_9
lesize,13,,,ndiv_9
lesize,16,,,ndiv_9
*get,len_2,line,2,leng
ndiv_2=(nint(len_2/lunbase))/2 !* line from top circle to bottom
circle & all the same
lesize,2,,,ndiv_2
!*lesize,6,,,ndiv_2
*get,len_6,line,6,leng  !* two extended ligaments from circle 1 & 4
ndiv_6=(nint(len_6/lunbase))
lesize,6,,,ndiv_6
lesize,7,,,ndiv_6
*get,len_1,line,1,leng
ndiv_1=(nint(len_1/lunbase))/2!* line from left circle to bottom
circle & all the same
lesize,1,,,ndiv_1
lesize,3,,,ndiv_1
lesize,4,,,ndiv_1
lesize,5,,,ndiv_1
```

```
ndiv_8=nint(lunarc8/lunbase)!* 1st arc of 1st semicircle on left & all
the same
lesize,8,,,ndiv_8
lesize,19,,,ndiv_8
lesize,12,,,ndiv_8
lesize,15,,,ndiv_8
ndiv_10=nint(lunarc10/lunbase)!* last arc of 1st semicircle on left &
all the same
lesize,10,,,ndiv_10
lesize,15,,,ndiv_10
lesize,11,,,ndiv_10
lesize,12,,,ndiv_10
lesize,14,,,ndiv_10
lesize,17,,,ndiv_10
!* MESH
lgen,2,all,,,,,thick,,0,0
l,5,23              !*line 39
l,6,27              !*line 40
l,19,33             !*line 41
l,15,37             !*line 42
l,20,34             !*line 43
l,9,29              !*line 44
l,8,25              !*line 45
l,7,24              !*line 46
l,16,38             !*line 47
l,12,28             !*line 48
l,11,26             !*line 49
l,10,31             !*line 50
l,17,39             !*line 51
l,13,32             !*line 52
l,14,30             !*line 53
l,21,35             !*line 54
l,18,40             !*line 55
l,22,36             !*line 56
al,43,6,41,25
al,42,8,41,27
al,41,9,40,28
al,40,10,39,29
al,39,1,46,20
al,47,11,46,30
al,46,12,45,31
al,45,13,44,32
al,40,3,48,22
al,49,2,45,21
al,48,35,49,16
al,49,34,50,15
al,50,33,51,14
al,44,4,53,23
al,50,5,52,24
al,52,36,53,17
al,53,37,54,18
al,54,38,55,19
al,54,26,56,7
*get,len_39,line,39,leng
ndiv_39=(nint(len_39/lunbase))/3
lesize,39,,,ndiv_39
lesize,40,,,ndiv_39
lesize,41,,,ndiv_39
lesize,42,,,ndiv_39
lesize,43,,,ndiv_39
lesize,44,,,ndiv_39
lesize,45,,,ndiv_39
lesize,46,,,ndiv_39
lesize,47,,,ndiv_39
```

```
lesize,48,,,ndiv_39
lesize,49,,,ndiv_39
lesize,50,,,ndiv_39
lesize,51,,,ndiv_39
lesize,52,,,ndiv_39
lesize,53,,,ndiv_39
lesize,54,,,ndiv_39
lesize,55,,,ndiv_39
lesize,56,,,ndiv_39
bb=6*t
Area=b*t
Izz=bb*t**3/12
!*Es=7.0e10 !* [Pa]
!*nus=0.33  !* Material: ALUMINIUM
et,1,shell93      !* Element type: SHELL63
r,1,t
!*et,1,beam3
!*r,1,Area,Izz
mp,ex,1,Es
mp,nuxy,1,nus
amesh,all
nummrg,all,1.e-12
numcmp,all
!*-----------------------------E z--------------------------
!*disp=0.01*thick
!*nsel,s,loc,z,0
!*d,all,all,0
!*allsel,all
!*nsel,s,loc,z,thick
!*d,all,uz,-disp
!*allsel,all
!*lsel,s,line,,39
!*lsel,a,line,,42,44,1
!*lsel,a,line,,47,48
!*lsel,a,line,,51,52
!*lsel,a,line,,55,56
!*nsll,s,1
!*d,all,rotx,0
!*d,all,roty,0
!*allsel,all
!*lsel,s,line,,39,,,,1
!*nsll,s,1
!**get,n_cal01,node,0,count
!**dim,ary_01,array,n_cal01,2
!**do,count1,1,n_cal01,1
!*    *get,ary_01(count1,1),node,,num,min
!*    nsel,u,node,,ary_01(count1,1)
!**enddo
!**dim,locat1,array,n_cal01,1
!*lsel,s,line,,52,,,,1
!*nsll,s,1
!**do,count2,1,n_cal01,1
!*    locat1(count2,1) = nz(ary_01(count2,1))
!*
      ary_01(count2,2)=node(radius*cos(90+gamma),radius*sin(90+gamma),
locat1(count2,1))
!*    ce,count2,0,ary_01(count2,1),ux,1,ary_01(count2,2),ux,-1
!*    ce,n_cal01+count2,0,ary_01(count2,1),uy,1,ary_01(count2,2),uy,-1
!*ce,2*n_cal01+count2,0,ary_01(count2,1),rotz,1,ary_01(count2,2),rotz,
-1
!**enddo
!*allsel,all
!*ce_no = n_cal01*100
```

```
!!!A LARGE AMOUNT OF PERIODIC CONSTRAINT EQUATIONS WERE OMITTED HERE!!!

!*fini
!*/solu
!*solve
!*fini
!*/post1
!*set,1,1
!*are_tot=4*R*sin(teta)*R*cos(teta)
!*nsel,s,loc,z,thick
!*fsum
!**get,f_z,fsum,0,item,fz
!*sz=abs(f_z)/are_tot
!*epz=disp/thick
!*Ez= sz/epz
!*Ez = Ez/Es
!*allsel,all
!*Gz_crl(j_cal,1)=thick
!*Gz_crl(j_cal,2)=t
!*Gz_crl(j_cal,3)=L
!*Gz_crl(j_cal,4)=radius
!*Gz_crl(j_cal,5)=Ez
!*-------------------------shear xz-------------------------
disp=0.01*thick
nsel,s,loc,z,0
d,all,all,0
allsel,all
nsel,s,loc,z,thick
d,all,ux,disp
d,all,uy,0
d,all,uz,0
allsel,all
lsel,s,line,,39,,,,1
nsll,s,1
*get,n_cal01,node,0,count
*dim,ary_01,array,n_cal01,2
*do,count1,1,n_cal01,1
      *get,ary_01(count1,1),node,,num,min
      nsel,u,node,,ary_01(count1,1)
*enddo
*dim,locat1,array,n_cal01,1
lsel,s,line,,52,,,,1
nsll,s,1
*do,count2,1,n_cal01,1
      locat1(count2,1) = nz(ary_01(count2,1))
      ary_01(count2,2)=node(radius*cos(90+gamma),radius*sin(90+gamma),
locat1(count2,1))
      ce,count2,0,ary_01(count2,1),ux,1,ary_01(count2,2),ux,-1
      ce,n_cal01+count2,0,ary_01(count2,1),uy,1,ary_01(count2,2),uy,-1
      ce,2*n_cal01+count2,0,ary_01(count2,1),rotz,1,ary_01(count2,2),r
otz,-1
*enddo
allsel,all
ce_no = n_cal01*100

!!!A LARGE AMOUNT OF PERIODIC CONSTRAINT EQUATIONS WERE OMITTED HERE!!!

lsel,s,line,,39
lsel,a,line,,42,44,1
lsel,a,line,,47,48
lsel,a,line,,51,52
lsel,a,line,,55,56
nsll,s,1
d,all,rotx,0
```

```
d,all,roty,0
allsel,all
fini
/solu
solve
fini
/post1
set,1,1
are_tot=4*R*sin(teta)*R*cos(teta)
nsel,s,loc,z,thick
fsum
*get,f_xz,fsum,0,item,fx
sxz=abs(f_xz)/are_tot
epxz=disp/thick
Gxz=sxz/epxz
G_c = Es/(2*(1+nus))
Gxz_c=Gxz/G_c
allsel,all
Gz_crl(j_cal,1)=thick
Gz_crl(j_cal,2)=t
Gz_crl(j_cal,3)=L
Gz_crl(j_cal,4)=radius
Gz_crl(j_cal,5)=Gxz_c
fini
!*----------------------------compression-----------------
/prep7      !* Parameters of the panel
*afun,deg
!**SET,L,0.02
!*A = 4.44
!*B = 0.22
!*C = 1
!*radius = 4.5/1000
L = A*radius
teta = theta
!**SET,thick,0.02
thick = A*C*radius
t = B*radius
beta = atan(radius/(L/2))
R = (2*radius)/sin(beta)
ffi=asin(radius/(R*sin(teta)))            !* tangent degree between
the central cirlces
k,1,0,0,0
beta = atan(2/A)
gamma = teta-beta  !* upper tangent degree of the first circles -90
k,2,radius*cos(90+gamma),radius*sin(90+gamma),0
k,3,radius*cos(90+gamma-2*teta),radius*sin(90+gamma-2*teta),0
k,4,radius*cos(-ffi),radius*sin(-ffi),0
k,5,radius*cos(-90+gamma),radius*sin(-90+gamma),0
k,6,radius*cos(-90+gamma-2*teta),radius*sin(-90+gamma-2*teta),0
k,7,radius*cos(180-ffi),radius*sin(-180-ffi),0
k,8,0.5*R*cos(teta),0.5*R*sin(teta),0
k,9,0.5*R*cos(teta),-0.5*R*sin(teta),0
k,10,0,-R*sin(teta),0
k,11,-0.5*R*cos(teta),-0.5*R*sin(teta),0
k,12,-0.5*R*cos(teta),0.5*R*sin(teta),0
k,13,0,R*sin(teta),0
l,2,8   !* line 1
l,3,9   !* line 2
l,4,10   !* line 3
l,5,11   !* line 4
l,6,12   !* line 5
l,7,13   !* line 6
larc,2,3,1,radius !* line 7
larc,3,4,1,radius !* line 8
```

```
larc,4,5,1,radius !* line 9
larc,5,6,1,radius !* line 10
larc,6,7,1,radius !* line 11
larc,7,2,1,radius !* line 12
!* Extrusion of plane cell
lgen,2,all,,,,,thick,,0,0
l,2,14   !* line 25
l,3,16   !* line 26
l,4,18   !* line 27
l,5,20   !* line 28
l,6,22   !* line 29
l,7,24   !* line 30
l,8,15   !* line 31
l,9,17   !* line 32
l,10,19   !* line 33
l,11,21   !* line 34
l,12,23   !* line 35
l,13,25   !* line 36
al,7,25,19,26  !* area 1
al,8,26,20,27  !* area 2
al,9,27,21,28  !* area 3
al,10,28,22,29  !* area 4
al,11,29,23,30  !* area 5
al,12,30,24,25  !* area 6
al,1,25,13,31  !* area 7
al,2,26,14,32  !* area 8
al,3,27,15,33  !* area 9
al,4,28,16,34  !* area 10
al,5,29,17,35  !* area 11
al,6,30,18,36  !* area 12
nummrg,all,1.e-12
numcmp,all
ET,1,SHELL93
R,1,t, , , , , ,
!*r,2,tcyl, , , , , ,
mp,dens,1,950
mp,ex,1,Es
mp,nuxy,1,nus
ESIZE,0.001,0,  !element edge length size
MSHAPE,0,2D
MSHKEY,1    !mapped meshing 1
!* MESH
type,1
real,1
amesh,1
amesh,2
amesh,3
amesh,4
amesh,5
amesh,6
type,1
real,1
amesh,7
amesh,8
amesh,9
amesh,10
amesh,11
amesh,12
!*amesh,all
nummrg,all,1.e-12
numcmp,all
*get,minx,node,0,mnloc,x
*get,maxx,node,0,mxloc,x
*get,miny,node,0,mnloc,y
```

```
*get,maxy,node,0,mxloc,y
side_x=maxx-minx
side_y=maxy-miny
area_cell=2.6*maxy*maxy
nsel,s,loc,z,0.0
d,all,uz,0.0
d,all,ux,0.0
d,all,uy,0.0
allsel,all

!!!A LARGE AMOUNT OF BOUNDARY CONDITIONS WERE OMITTED HERE!!!

lsel,s,line,,31,,,,1
nsll,s,1
*get,n_cal01,node,0,count
*dim,ary_01,array,n_cal01,2
*do,count1,1,n_cal01,1
      *get,ary_01(count1,1),node,,num,min
      nsel,u,node,,ary_01(count1,1)
*enddo
*dim,locat1,array,n_cal01,1
lsel,s,line,,34,,,,1
nsll,s,1
*do,count2,1,n_cal01,1
      locat1(count2,1) = nz(ary_01(count2,1))
      ary_01(count2,2)=node(-0.5*R*cos(teta),-
0.5*R*sin(teta),locat1(count2,1))
      ce,count2,0,ary_01(count2,1),ux,1,ary_01(count2,2),ux,-1
      ce,n_cal01+count2,0,ary_01(count2,1),uy,1,ary_01(count2,2),uy,-1
      ce,2*n_cal01+count2,0,ary_01(count2,1),rotz,1,ary_01(count2,2),r
otz,-1
*enddo
allsel,all
ce_no = n_cal01*100

!!!A LARGE AMOUNT OF PERIODIC CONSTRAINT EQUATIONS WERE OMITTED HERE!!!

!*-----------------------compressive strength-------------------------
*SET,F_z,1.0
nsel,s,loc,z,thick
*get,node_1,node,0,count
f,all,fz,-F_z/node_1
allsel,all
fini
/solu
antype,static
pstres,on
allsel,all
solve
fini
/solu
antype,buckle
bucopt,subsp,3
outpr,nsol,all
save
solve
fini
/post1
set,1,3
*get,f_buck,active,0,set,freq
sigma_cell=f_buck/(area_cell*1E6)
coef_epsilon=sigma_cell/Es
Gz_crl(j_cal,1)=thick
Gz_crl(j_cal,2)=t
```

```
Gz_crl(j_cal,3)=L
Gz_crl(j_cal,4)=radius
Gz_crl(j_cal,5)=sigma_cell
Gz_crl(j_cal,6)=coef_epsilon
fini
```