January 2013

Shield Design for Maximum Deformation in Shape-Shifting Surfaces

Daniel Eduardo Perez
University of South Florida, deperez@mail.usf.edu

Shield Design for Maximum Deformation in Shape-Shifting Surfaces

by

Daniel E. Perez

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Mechanical Engineering
Department of Mechanical Engineering
College of Engineering
University of South Florida

Major Professor: Craig Lusk, Ph.D.
Rasim Guldiken, Ph.D.
Luther Palmer, Ph.D.

Date of Approval:
March 18, 2013

Keywords: Compliant Mechanisms, Adaptronics,
Smart Structures, Tiled Mechanisms, Kinematics

ACKNOWLEDGMENTS

This thesis was completed thanks to the support of many people. I would like to express my gratitude primarily to my advisor, Dr. Craig Lusk who was unconditionally helpful during the development of this research. Dr. Lusk's support, knowledge, guidance, and patience were of incredible importance.

I would also like to acknowledge Mr. Ahmad Al-Qasimi, also part of this research team, for his continuous ideas on improvement and support using CAD software.

Furthermore, I also want to thank my thesis committee, Dr. Palmer and Dr. Guldiken for their feedback. I would also like to thank all the faculty and staff of the Mechanical Engineering Department and the College of Engineering at the University of South Florida for their incessant effort on educating real engineers. Last, but not least, to my educators at the Universidad del Norte in Barranquilla for their great contribution in my formation as an engineer.

Dedico este triunfo a mis padres por su incondicional apoyo, paciencia y amor. Mil gracias.

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

This research presents the initial studies and results on shield design for Shape-Shifting Surfaces (SSSs) seeking maximum compression and maximum expansion of a unit-cell. Shape-Shifting Surfaces (SSSs) are multilayered surfaces that are able to change shape while maintaining their integrity as physical barriers. SSSs are composed of polygonal unit-cells, which can change side lengths and corner angles. These changes are made possible by each side and corner consisting of at least two different shields, or layers of material. As the layers undergo relative motion, the unit-cell changes shape. In order for the SSS to retain its effectiveness as a barrier, no gaps can open between different layers. Also, the layers cannot protrude past the boundaries of the unit-cell. Based on these requirements, using equilateral triangle unit-cells and triangular shields, a design space exploration was performed to determine the maximum deformation range of a unit-cell. It was found that the triangular shield that offered maximum expansion and compression ratio is a right triangle with one angle of 37.5 degrees and its adjacent side equal to 61% of the side of the unit-cell. The key contribution of this paper is a first algorithm for systematic SSS shield design. Possible applications for SSSs include protection, by creating body-armor systems; reconfigurable antennas able to broadcast through different frequencies; recreational uses, and biomedical applications.

CHAPTER 1: INTRODUCTION

1.1 Objective

The objective of this thesis is to find the shield surface, made up of overlapping movable triangles, that is most able to change its shape while maintaining its integrity as a physical barrier.

1.2 Motivation

A surface is defined as the material at the top or outer layer of an entity that constitutes a boundary. When referring to a surface, one generally implies that there is something underneath it or that it is a boundary between two volumes. Surfaces can be rigid and flat, like the one of a table or door, or they can be irregular, like the surface of the moon. Liquids that constitute a surface can change its form constantly and can be penetrated easily. Surfaces are often used to keep substances or objects within a constrained area or to keep other objects from entering it. The skin, for instance, is a multilayered surface that protects from external pathogens and keeps essential nutrients within the body. Many times, surfaces made of flexible or elastic materials allow changes in shape that translates into variations in covered area and volume, e.g., plastic bags, clothes, rubber balloons. Other surfaces used in containers are made up of retractable solid layers that expand to increase volume, e.g. retractable cups. Cell phones can increase or reduce their size for ease of use with the push of

one button or simply sliding one of the cover surfaces. The use and applications of movable surfaces is very broad and are applied in different areas like communications, entertainment, fashion, industry, etc. The initial thought, however, that motivated this research was body armor. Soldiers in war-torn areas are constantly exposed to multiple threats that can cause severe permanent injuries or death. A surface that is able to change shape while protecting the physical integrity of soldiers may be a viable solution to the multiple movement constraints and injuries caused by current body armor systems. This could be an initial step in the development of a better system of body armor. The vision while developing this preliminary design is the creation and implementation of a light-weight body-armor system, able to offer maximum protection to soldiers from external threats, with minimized mobility constraints.

1.3 Scope

The scope of this thesis is the description of a computer algorithm that evaluates all the possible dimensions and arrangements of the movable shields that make up the unit-cells of the surface. The program determines which shield shape offers the maximum mobility, i.e. the shield shape that offers maximum compression and maximum expansion without compromising barrier effectiveness. Prototypes with different shapes show the variations in expansion and compression.

1.4 Thesis Overview

Chapter 2 provides the background and the concepts needed to understand of the project. The term “Shape-Shifting Surfaces”, which describes the variable

geometry surfaces, is explained. This chapter includes previous work in similar and related fields as well as the description of a potential application of Shape-Shifting Surfaces.

Chapter 3 describes the methods, theory and terminology used in the design of shields for Shape-Shifting Surfaces.

A detailed overview of the programming considerations, procedures, and the scoring and evaluation method is included in Chapter 4. It describes step-by-step the major and minor algorithms that were implemented to evaluate and score the design variations of Shape-Shifting Surfaces.

Chapter 5 describes the results and provides discussion of them, comparing the outcomes from the programming with the physical prototype. Chapter 6 includes the research contributions and gives recommendations for future work. Finally, chapter 7 gives the thesis conclusions.

CHAPTER 2: BACKGROUND

This chapter describes all the concepts related to SSSs that are needed to have a clear understanding of the subsequent chapters. First, a general background on adaptronics and smart structures and its relation to this thesis is presented; then, more project specific information on compliant mechanisms and results particularly important to this work on Shape-Shifting Surfaces are provided. Finally, a possible direct application of Shape-Shifting Surfaces is described.

2.1 Theoretical Background

The term “Shape-Shifting Surfaces (SSSs)” was defined by Lusk and Montalbano as “surfaces that retain their effectiveness as physical barriers while undergoing changes in shape” [1, 2]. The SSSs achieve their motion through the use of compliant mechanisms. Figure 2.1 shows one of the designs developed by Lusk and Montalbano [2]. This unit-cell in the figure can be deformed into different shapes while maintaining its integrity. The deformations are obtained through compliant segments in each of the links.

2.1.1 Adaptronics and Smart Structures

There are several approaches towards the definition of smart structures and the science that studies and develops them. In a nutshell, we refer to smart structures or materials as those which are capable of sensing and adapting to

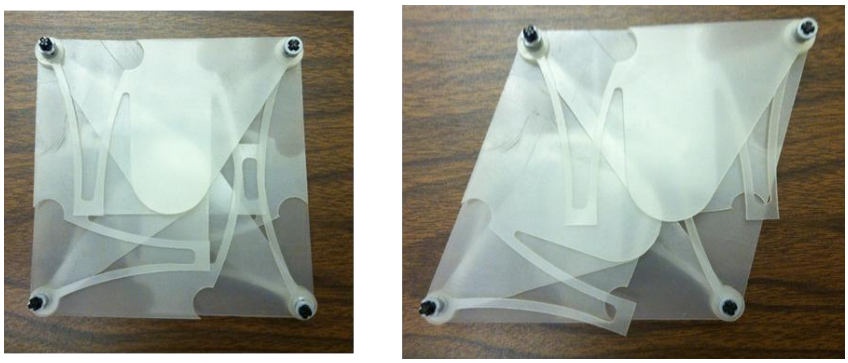

Figure 2.1: Sample of a Shape Shifting Surface unit-cell designed by Lusk and Montalbano.

their surroundings, changing their physical properties such as shape, color, structure, polarization, magnetization, conductivity, etc [3]. The technical term adaptronics (Adaptronik) was originally coined by the VDI Technology Centre in Germany [4], and was described as an interdisciplinary science that deals with the development of what are internationally known as smart materials, smart structures, or intelligent systems. A system that is designed based on adaptronics must include all functional elements of a conventional regulator circuit (sensor, actuator and controlling unit), using at least one in a multifunctional way [4]. This characteristic ensures that the system is adaptable to a variety of external conditions in an autonomous way. The limitations that a regulator circuit has, due to the fact that each function is accomplished by different components, are solved with the use of elements that can perform multiple functions (e.g. multifunctional materials). Most of the materials used in adaptronic systems can be used both as a sensor and as an actuator. New multifunctional materials are based on transducer properties, since they have the special property of converting electrical, magnetic, thermal, or other types of

energy into mechanical energy [4]. Adaptronics includes designing and building adaptive structures using lightweight materials having as one of the main objectives the reduction of the amount of material and energy resources that are necessary for their construction and operation. The ultimate goal of adaptronics is the combination of the greatest amount possible of functions in a single material or structure.

One example of an adaptronic system is the photochromic lenses used in eyeglasses [4]. This material is capable of darkening in a self-adaptable way when exposed to ultraviolet (UV) radiation. These lenses, popularized by Transitions Optical in the 1990's, include all three elements of a regulator circuit: It senses the UV light, darkens or lightens, and self regulates the amount of darkness depending on the received UV radiation. Photochromic lenses show that it is possible to develop materials or structures that can effectively combine the elements of a regulator circuit into a single component [4]. Figure 2.2 shows a general idea of the transition of a conventional system to an adaptronic system. Figure 2.2a) illustrates a system in which the three elements of a regulator circuit perform their task as separate elements; Figure 2.2b) describes a merged system in which the elements of the regulator circuit are all part of a single component.

SSSs can be interpreted as smart structures and as a branch in adaptronics because they are able to adapt to their surroundings depending on the applied forces, changing shape and gaining a variations in performance and functionality. The fact that SSSs include compliant mechanisms makes them have particular

characteristics that allow much more flexibility than rigid structures and are able to transfer force throughout the surface structure, thus changing its geometry [1, 2]. In body armor, SSSs are able to increase the level of protection when compressive

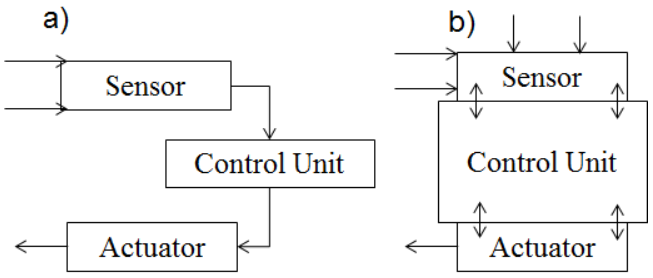

Figure 2.2: a) Conventional regulator system, b) adaptronic system.

forces are applied to it, since compression will allow a greater number of layers to overlap. For future work, compliant segments used in SSSs can include materials and structure design that have intrinsic sensor(s), actuator(s), and control mechanism(s), being able to sense stimulus and respond to it in a prearranged way, in a short period of time, and being able to revert to its original condition as the stimulus is reduced [3].

2.1.2 Compliant Mechanisms

Mechanisms are mechanical devices used to transfer or convert force, motion, or energy. Rigid-body mechanisms are made up of rigid links that move with respect to each other through joints. Compliant mechanisms perform the same functions as regular mechanisms, with the special characteristic that they transfer a significant amount of force, motion, or energy through the deflection of

flexible links rather than through movable rigid links. The flexible members store strain energy as they deform [1, 2, and 5].

"*Compliant mechanisms are often preferred over rigid link mechanisms, since compliant mechanisms do not exhibit frictional losses, joint wear, and tear*". [5] Some of the general advantages are that compliant mechanisms allow an assembly with fewer amounts of parts due to their flexibility. These mechanisms have a fewer movable joints, like pins and sliding joints. Fewer parts means reduced manufacturing and assembly costs, reduced maintenance costs, more durability since the wear and need for lubrication are significantly reduced as well; the weight of the final assemblies is also reduced. Fewer parts also reduce the ratio of defective final products. The advantages of compliant mechanisms can be classified in two groups [5]:

1. Cost reduction:
    - Part-count reduction
    - Reduced assembly time
    - Simplified manufacturing process
2. Improvement of performance:
    - Increased precision
    - Increased reliability
    - Reduced wear
    - Reduced weight
    - Reduced Maintenance [2].

Some drawbacks from the use of compliant mechanisms include the storage of energy in the flexible segments [7]. Flexible segments store energy when they experience deflection within the elastic limit of the material, and forces are required to keep mechanism deflected. Compliant systems with stressed elements are used to balance unstressed segments. Thus, as one segment is loaded, another is unloaded, resulting in balanced energy storage [7, 8].

The behavior of compliant mechanisms cannot be predicted with total accuracy. Due to the fact that compliant mechanisms undergo large non-linear deformations, the well-known equations for small deflections are not adequate to study their behavior. The study of the kinematics of compliant mechanisms can be approached using standard beam equations used for rigid bodies and/or pseudo-rigid body models (PRBM) [5]. The PRBM approach uses similarities between compliant mechanisms and rigid-body mechanisms. Software is also used to analyze and predict the behavior of these mechanisms. These computational approaches usually combine finite element analysis (FEA) to predict the behavior of the mechanisms when force loads and motion are applied. The results of computational analyses are slightly more accurate compared to the PRBM approach but they require more time [2].

2.1.3 Shape-Shifting Surfaces (SSSs)

Shape-Shifting Surfaces can be described as multi-layered surfaces that undergo variations in shape maintaining the physical barrier characteristic [2]; in other words, allowing changes in geometry while having no gaps and no protrusions. The idea comes from creating better body armor by protecting

vulnerable areas like limb joints. SSSs will allow free movement while keeping a constant protection in all the areas of the body. The shields that make up the unit-cells of each compliant system are connected through compliant flexures, which allow a greater range of movement with reduced amount of parts. The compliant flexures give the unit-cell mobility, while the shield maintains the unit-cell as an effective physical barrier.

2.1.3.1 Kinematic Structure of SSSs

The SSSs are designed to cover a surface, and each unit-cell with any particular geometry must be able to change its shape in response to external applied forces. For this thesis, we will consider SSSs with triangular unit-cell geometry. Each unit-cell has three straight sides with three nodes (shown in Figure 2.3a). To obtain relative movement within the links, each of their sides is represented by a kinematic slider with revolute joints on the corners (shown in Figure 2.3b). The kinematic sliders and revolute joints allow the unit-cell to expand and compress, obtaining different triangular geometries. The relative movement between shields implies overlapping between them. The level of overlapping depends on the shield shape and on the deformation of the unit-cell.

2.1.3.2 Previous Work on Shape Shifting Surfaces

Previous studies on SSSs focused on the compliant portion of the unit-cell. This is the first to focus on shield design. Montalbano introduced bistability into the SSSs design. Bistability gives the surface a variety of positions in which it

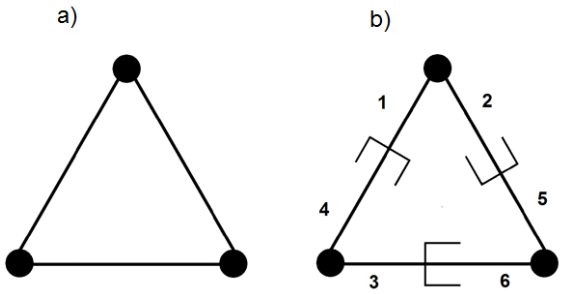

Figure 2.3: Part a): SSS unit-cell with three nodes. Part b): unit-cell with 6 link kinematic representation and same three nodes

remains when shifted to [1]. Stable unit-cell deformations include partial and complete expansion and compression, and shear, resulting in planar shapes of multiple geometries. Pishnery introduced the method to statically balance a specifically-designed compliant mechanism and how to integrate it into a polygonal cell [8].

2.1.3.3 Underactuated Mechanisms and Shape Shifting Surfaces

The design of the compliant mechanisms can consider underactuated systems, which are mechanical systems with more degrees of freedom than control inputs; in other words, if the control input cannot accelerate the system in every direction, the system is underactuated. Such systems have numerous advantages including reduced weight, low cost, and low consumption of energy [9]. The use of underactuated mechanisms has broadened in the past decades with applications to undersea robots, mobile robots, space robots, walking robots, etc. Underactuated mechanisms fit within the design constraints of smart structures, since their study purpose is the design of simple multifunctional mechanisms that adapt to modern design. However, the main obstacle in the

design and implementation of underactuated systems is their control, due to the complexity of the design that makes these systems have no full feedback linearization [10]. On the other hand, fully actuated systems have a number of control methodologies, including feedback linearization and passivity-based adaptive control [10, 11]. For the case of SSSs, single force inputs modify the geometry of the surface by displacing several shields and flexing several compliant segments. Therefore, there is a greater amount of degrees of freedom that can be controlled by one force input.

2.2 Transformational Operators Theory

Transformational operators are used in the algorithm that looks for the shield shape that offers maximum mobility. Transformational operators are made up by translational operators and rotational operators. These operators are not only used for the initial shape of the unit-cell, but also to determine the position of each of its vertices at all times while the algorithm is running. It is important to highlight that using transformational operators optimized the algorithm since the amount of mathematics and lines used in it was significantly reduced. A brief explanation of the mathematics involved in the use of transformational operators is given in this section.

A translation moves a point in space a determined distance through a given vector direction [12]. Figure 2.4 indicates graphically how a vector B_1 is moved through translation by a vector Q, which is the vector that has the translation values. The result is a new vector B_2. The calculation is done as:

$$B_2 = B_1 + Q$$

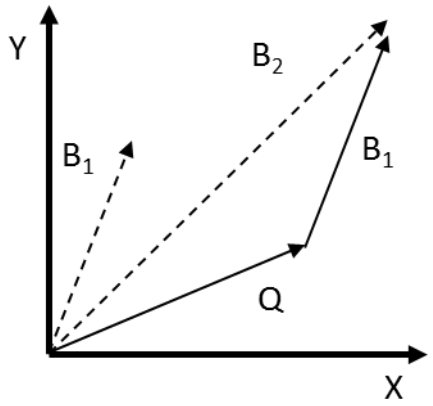

Figure 2.4: Translation operator

To write this operator as a translation matrix, the following notation is used:

$$B_2 = D_Q B_1$$

where D_Q is the matrix

$$D_Q = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and d_x, d_y, and d_z are the translation vector components.

Rotational operators change vectors by means of rotation. The notation for rotating a vector about a determined axis is

$$B_2 = R_z B_1$$

where R_z is the indicator of rotation about the Z-axis. As a matrix, the rotational operator is

$$R_z = \begin{bmatrix} cos\theta & -sin\theta & 0 & 0 \\ sin\theta & cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

In order to rotate a vector about a different axis, a different matrix configuration is used. Since for this research we use rotational operators to rotate about the *Z* axis, we will only provide an example for this case [12]. Figure 2.5 shows the rotation of vector B_1 about the *Z* axis by θ degrees, keeping the same origin.

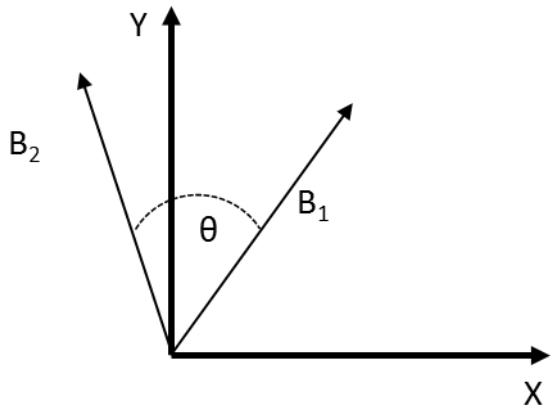

Figure 2.5: The vector B_1 rotated θ degrees about the *Z* axis.

Transformation operators are used to rotate and translate vectors in a single operation. The notation for transformation operators is

$$B_2 = TB_1$$

Where *T* is the transformation matrix, written as

$$T = \begin{bmatrix} cos\theta & -sin\theta & 0 & d_x \\ sin\theta & cos\theta & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The transformation operator *T* performs a rotation of θ degrees about the *Z* axis, and translates the vector using the d_x, d_y, and d_z components [12]. The example in Figure 2.6 shows that the vector B_1 is translated by vector D_Q, and rotated by θ degrees by R_Z using one operational transform.

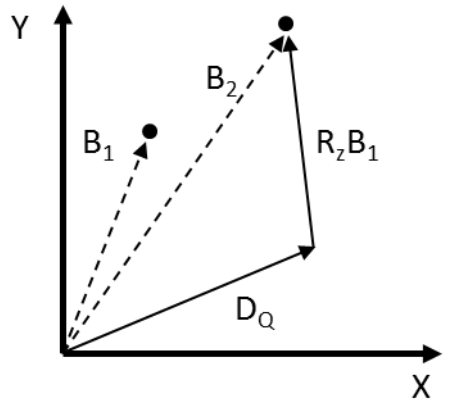

Figure 2.6: Vector B_1 translated by *Q* and rotated by R_z

2.3 Previous Work on Similar Concepts

Claytronics, Programmable Matter, and Digital Clay, are reconfigurable robotic systems that have similar functions and applications as Shape-Shifting Surfaces [2]. Claytronics is the given name for programmable matter whose function is to arrange itself into a programmed shape, to match the external appearance of any object. The individual components that make up claytronics are called catoms [13]. These catoms can move in three dimensions, assemble and disassemble to form the network that shapes the desired figure. Claytronics allows the transformation of matter into any shape, using physical elements and not holograms [14].

Programmable matter looks for the creation of smart materials and structures whose physical properties, for instance shape, optical and acoustic features, and viscosity, can be programmed; alternatively, programmable matter aims to make machines feel more like materials. Current work on programmable matter seeks the creation of materials with intrinsic sensing, actuation, communication, computation, and connection [15].

Digital clay is a digital approach to shape surfaces based on haptic theories. The interface displays shapes specified by the user and the user directed input of shapes [16].

A significant difference between SSSs and the mentioned reconfigurable systems is that SSS, because of the use of compliant mechanisms, are functional without the need of actuation and several passive behaviors can be included in the design without the need of actuators or processors [2].

2.4 Applications Background

SSSs can have a variety of applications in different areas including control surfaces, medicine, and devices such as antennas, sensors, etc. However, as mentioned in Chapter 1, the thought that inspired this research was body armor.

2.4.1 Body-Armor Systems

The continuous dangers to which soldiers are exposed in warzones represent a major threat to their lives. For instance, soldiers in these areas are continuously exposed to improvised explosive devices (IED). To have a better

idea of the effects of these devices, it is estimated that nearly 30% of coalition deaths in Operation Enduring Freedom have been caused by IEDs [17, 18].

It has been proven through experience that body armor has great advantages and has saved many lives, reducing significantly the fatal victims in military and civilian scenarios [19]. The goal of armor systems is to protect the user as much as possible maintaining a balance between the bod’y areas that are covered, weight, and allowed movement. In many situations involving body protection using armor, standard body armor systems provide satisfactory protection to the torso and head involving low- and high-velocity gunshot wounds [20, 21]. Yet several areas of the body remain uncovered allowing the subject to move freely and reducing the weight of the armor. These exposed areas represent a threat since external objects are able to penetrate the body. The basic armor system includes a helmet and a body armor known as Interceptor Multi-Threat Body Armor System (IBA) [22]. To address the issue of exposed body areas and reduce the rates of injury and deaths, different protection systems have been designed that cover other zones of the body. There are different attachments to protect the neck, groin, deltoid, and axillary areas that the IBA system does not cover and are optionally worn by soldiers depending on the unit [22].

It is important to highlight that the added protection also has a cost for the soldier and his physical integrity. Increasing protected areas and reducing vulnerability to external treats also increases the total load of the armor and reduces the mobility, flexibility, and agility of the soldier. For instance, twice as

many soldiers have an increased amount of pain in their musculoskeletal system because of the body armor than because of job tasks and training. A medium-size IBA vest without the optional attachments weights about 28.6 lb. [23]. Carrying this weight during trainings and military operations produce musculoskeletal pain and injuries, which are common cases treated at military medical facilities, most of them occurring in non-battle settings. Reports show that there has been an increase in back, neck, and upper extremity musculoskeletal pain in soldiers located in deployed areas affecting significantly alert readiness and combat performance [22].

2.4.2 Body-Armor Systems and SSSs

A mechanical system that offers body protection through mobile layers able to bend, expand, and compress can be developed using compliant mechanisms and SSSs, allowing subject movement without putting at risk any vulnerable areas. This design can increase the protected regions of the body of people exposed to bullets, explosives, or other external threats. Generally, the limb joints (shoulders, underarms, elbows) are areas with high vulnerability since the existing protection systems are made up of rigid plates unable to expand, contract, or bend. SSSs are a novel concept that is based on compliant mechanisms which experience motion and allow variations in shape when subjected to applied forces. SSSs can be considered as a solution to the risk of exposing areas of the body to external threats; SSSs are able to cover completely these vulnerable areas regardless of the movement of the person wearing the shields. This research focuses on the design of the shields of the

SSSs with a triangular geometry that can be used to create new systems of body armor. The design uses data obtained from the algorithm that determines which combination of triangular geometries can create a more efficient Shape-Shifting Surface that has no gaps and no protrusions.

CHAPTER 3: DESIGN OF SHAPE-SHIFTING SURFACES

This chapter will introduce new concepts related to this research, the geometry of the SSSs, and the assumptions, initial shield shape, and variable parameters which define the design exploration.

3.1 Defining Unit-cells and Link Shapes

The process of the SSSs design includes defining the initial shape of the deformable unit-cell and of the shields, the rigid portions of the links that make up the cell [2]. Because this is the first systematic approach developed to evaluate shield geometries, a simple polygon, the triangle, was considered for both unit-cells and shields. The initial shape of the unit-cell is an equilateral triangle made up of six shields. The shields are arranged in a way that allows them to slide, rotate, and overlap with respect to one another.

3.1.1 Shield Triangles

The shield triangles are right triangles since these offer the maximum range in coverage when assembling the structure without having any unnecessary overlap between the layers. By way of example, Figure 3.1 shows the assembled unit-cell using shield triangles with 30-55-95 angles. The highlighted region in the center indicates an unnecessary overlap when the cell is at its maximum expansion with equilateral triangle geometry. On the other hand, triangles with acute angles do not have the same coverage as right triangles at maximum

expansion. For instance, Figure 3.2 illustrates a unit-cell made up of 30-65-85 degree shields, and shows how the covered area decreases making it necessary to move the shields in order to maintain the integrity of the triangle. The highlighted sections show the overlap between the shields at maximum expansion with this deformation.

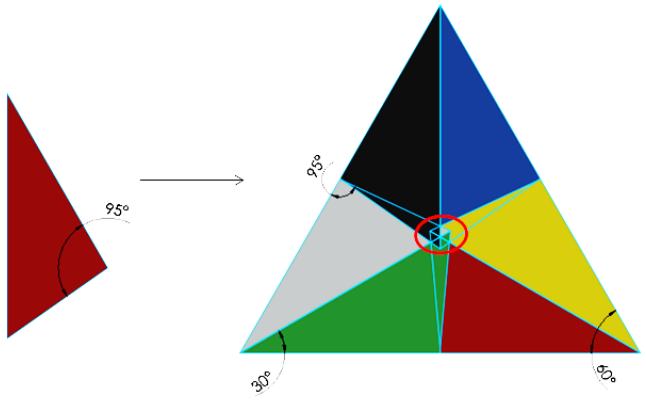

Figure 3.1: Cell configuration with 30-55-95 angle triangular shields

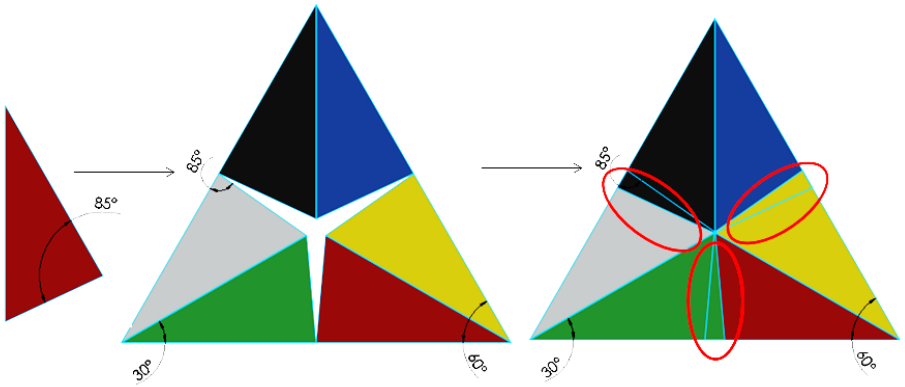

Figure 3.2: Cell configuration with 30-65-85 angle triangular shields

In Figure 3.3, part a) shows the exploded view of the initial configuration of the unit-cell that is considered for this research. Each triangle has initial angular

dimensions of 90-60-30 degrees. Figure 3.3 part b) shows the assembled initial configuration of the unit-cell.

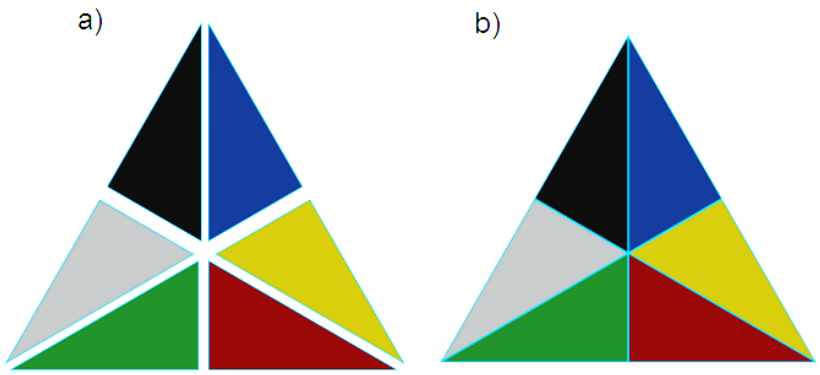

Figure 3.3: a) Exploded view of the initial configuration of the unit-cell. b) assembled shields in initial configuration.

3.2 Gaps and Protrusions

SSSs are made up of overlapped unit-cells with a determined geometry organized in a specific tiling. Our study will include triangles, because they are the most basic polygon with straight sides. The initial goal is to create an algorithm using Matlab software that finds the unit-cell that will allowing maximum compression and maximum expansion with no gaps and no protrusions. Gaps are blank spaces within unit-cell that are not covered by any portion of the shields; protrusions are parts of the shield that extend outside the triangular geometry of the unit-cell. Figure 3.4a) shows an example of a gap within the cell. The circled blank space shows there is no material in that area, thus, the unit-cell is no longer as effective as a physical barrier. In Figure 3.4, part b) illustrates a deformation with a protrusion, which doesn't necessarily mean the unit-cell is no longer an effective physical barrier, but may affect the possibility of tiling the unit-

cells, because the overlaps conflict with other shields from adjacent unit-cells at the same height.

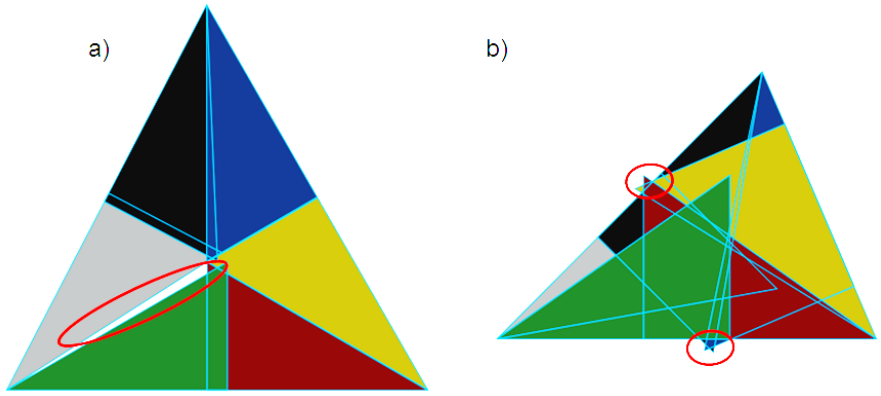

Figure 3.4: a) Example of a gap. b) Example of protrusion

This initial study determined the critical deformations and positions for triangle-based unit-cells as well as the limits of motion for the triangular unit-cell SSSs. In future work, algorithms can be developed in which specific dimensions for a needed structure are used as an input, and the program would output the recommended triangle configuration.

3.3 Shapes of Shields

For this study, we take into consideration the shield portion of the unit-cells. The compliant links that connect them do not take part in the simulation. This means that after an optimal solution is found, other design constraints must be considered to maintain the no gap-no protrusion before the SSSs is assembled. Calculations and simulations of the behavior of the structure should be made as well using pseudo-rigid-body model for the compliant links.

3.3.1 Initial Shape

The initial arrangement of each unit-cell includes three pairs of equal right triangles. Each triangle has two design parameters (one side and one angle) that determine the values for the rest of the sides and angles. The parameters that define the geometry of the shield triangle are one of the angles and its adjacent side. The size and geometry of the shield triangles change with the same rate, having always a cell that is made up of equal triangles. An example of the exploded view of one configuration is shown in Figure 3.5. It shows 6 equal right triangles that will be assembled to create the unit-cell.

The initial geometry of the cell is an equilateral triangle. As illustrated in Figure 3.6a), the shield triangles are arranged in pairs in each vertex of the cell triangle. Each pair is made up of mirrored triangles that use the vertices of the unit-cell triangle as a pivot point to rotate. Figure 3.6b) shows how the shields rotate about a pivot point located in the vertices of the unit-cell.

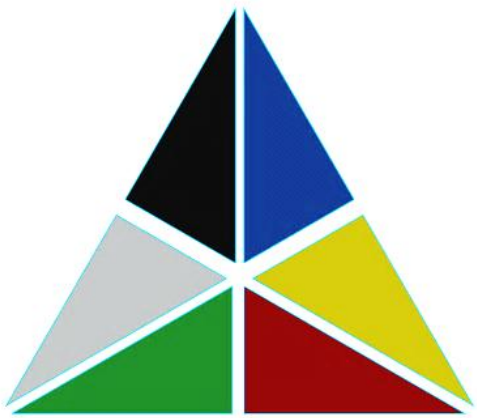

Figure 3.5: Exploded view of initial configuration

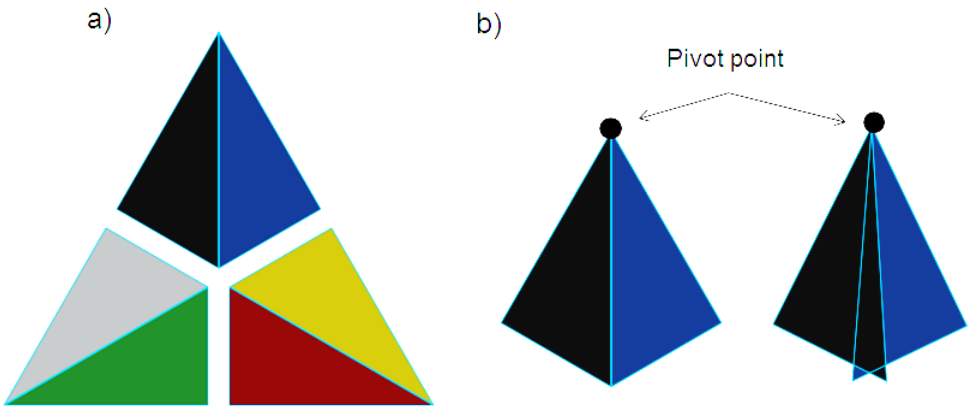

Figure 3.6: a) Exploded view of paired shields. b) Rotation of paired shields about a pivot point

To keep the triangular geometry, the outer sides of the adjacent shield triangles must always be collinear, meaning that the position of each of the shield triangles depends on the position of the other shield triangles. In other words, shield triangles that share the same side of the unit-cell slide with respect to one another, as described in Figure 3.7.

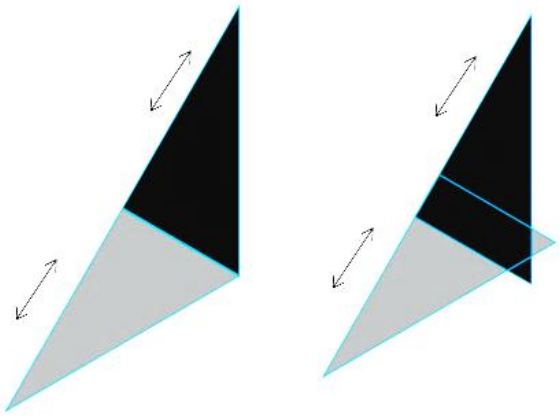

Figure 3.7: Sliding shields on the side of the unit-cell

3.3.2 Variable Parameters

As mentioned earlier in this chapter, the design of the shield triangles depends on two variable parameters and a constant parameter. Taking as an example the triangle in Figure 3.8, the 90° angle remains constant, while the length l_1 and angle α_{13} change. For this particular study, due to easiness of configuration and simplified values, the length of l_1 will have minimum and maximum values of 0.5*u* and 1.0*u* respectively, where *u* stands for length-units. These values were chosen based on single unit length sides of an equilateral triangle, in which the minimum length of a side divided in two segments is 0.5*u* (no overlap), and the maximum is 1.0*u* (complete overlap). In Figure 3.8, part a) illustrates the initial shape of the shield, with the initial values of the two design variables. Figure 3.8b) has an increased value of angle α_{13}; l_1 remains with the same value. Figure 3.8c) maintains the value of the α_{13} angle, and increases the length of side l_1.

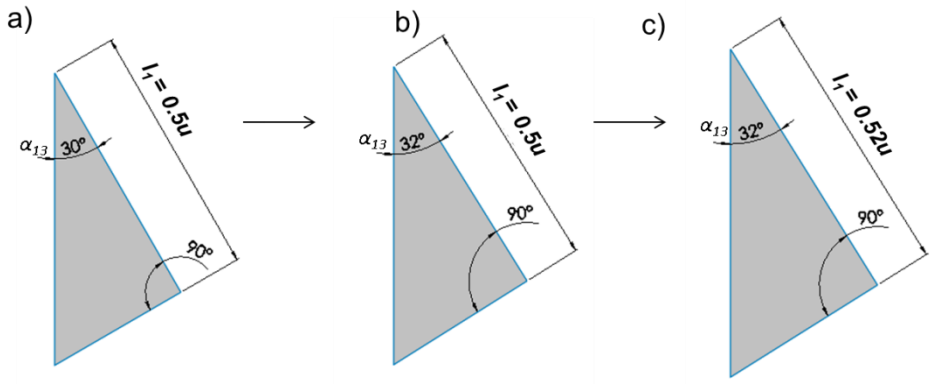

Figure 3.8: Variable parameters in shield shape. Part a) shows initial triangle, Part b) shows increased angle α_{13}, c) shows increased length l_1.

CHAPTER 4: ALGORITHM FOR DESIGN-SPACE EXPLORATION

This chapter shows the development of the algorithm and the series of steps it follows in order to evaluate, classify, and score all the possible arrangements of the shield geometries that can build a unit-cell. Moreover, Chapter 4 describes in detail the algorithm, illustrating the steps followed to reach the objective as well as the scoring method. Finally, Chapter 4 includes the process of designing and manufacturing prototypes using compliant mechanism theory and the obtained data.

4.1 Algorithm Considerations

A Matlab algorithm was designed to model a unit-cell out of pairs of congruent shield triangles. Each pair is replicated two times in different positions, having as a final result three pairs of triangular shields. A number of iterations change each parameter of the shields and tests the shape variations that can be obtained with that particular shield shape. Figure 4.1 shows the outermost loop done by the program in order to classify each of the shield shapes. Each of these steps includes a more complex set of iterations, which are described later on this chapter. The main algorithm is made up by three sub algorithms: the shield definition algorithm, the unit-cell deformation algorithm, and the integrity assessment algorithm.

This main algorithm begins by setting the initial shield shape, with α_{13}= 30° and l_1= 0.5u.It then evaluates the shield through all the possible unit-cell deformations and assigns a score to the shield before moving on to the next shield shape. After evaluating and scoring the final shield shape with α_{13}= 60° and l_1= 1.0u, the algorithm exits the program and displays the results.

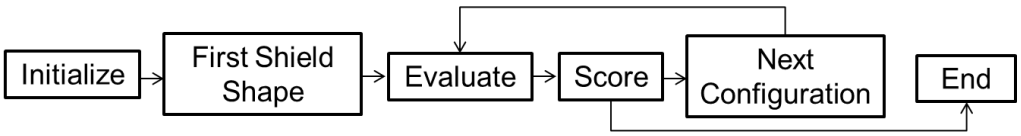

Figure 4.1: General program loop

4.1.1 Locating the Vertices of the Triangular Shields

Before locating the positions of the triangles, the coordinates of the vertices of the shields are set. For the initial unit-cell shape, an equilateral triangle, the top vertex represents the origin of the coordinate system. Figure 4.2 illustrates the positioning of the initial pair of triangles on a Matlab plot, in which Triangle 1 and Triangle 2 represent the basic geometry that determines the shape of the remaining shield triangles. Each triangle is made up of three connecting vectors.

The initial shape of triangles 1 and 2 was determined based on the first shield shape and unit-cell shape described in the previous chapter: shields are right triangles with side length equal to 0.5u, and adjacent angle equal to 30°. The initial unit-cell is an equilateral triangle with side length equal to 1u. With these parameters we are able to determine position the vertices of the two initial shield triangles using geometrical theory.

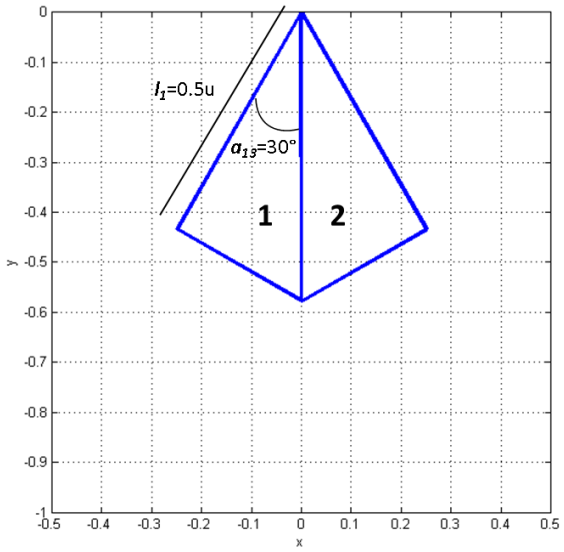

Figure 4.2: The position of the first pair of right triangles on the coordinate system

The vectors used in these triangles are used to determine the shape and location of the other two pairs that make up the unit-cell, and in order to do it, vector theories and transformation operators were used. The advantage offered by transformation operators is that instead of creating new coordinates for each of the vertices, the origin of the initial pair of triangles is translated and rotated with respect to its initial position. .

4.1.2 Transformational Operators Applied to Unit-cell Shape

With the described theory, we are able to show the steps used to complete the unit-cell using transformational operators. Figure 4.3 shows the order in which the shields are positioned for the initial unit-cell shape. Triangles 1 and 2 are positioned first and, through the use of transformational operators, the coordinates of the vectors forming triangles 3 and 4, and then triangles 5 and 6 were obtained.

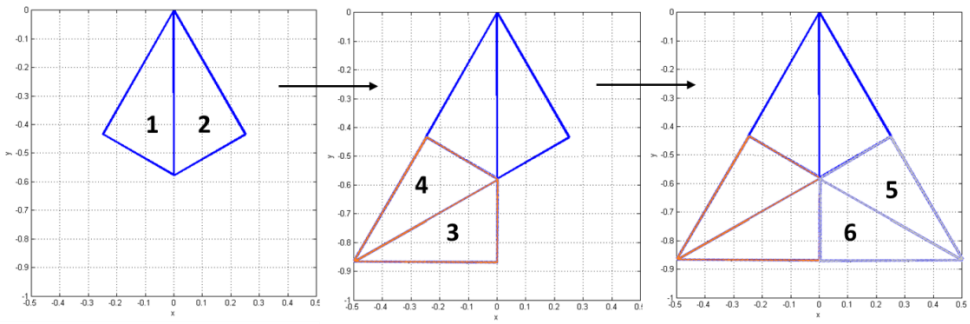

Figure 4.3: Initial locations of shields

4.2 Sub-Algorithms

As previously established, the main program is made up by three sub-algorithms, each one with a specific function. The outermost loop is the Shield Definition algorithm, which focuses on the dimensions of the shields by changing their angle and side length dimensions before each shape is evaluated and scored. The Unit-cell Deformation algorithm changes the shape of the unit-cell, making sure all the possible shapes are evaluated. Finally, the Integrity Assessment algorithm evaluates and scores each one of the Unit-cell deformations.

4.2.1 Shield-Definition Algorithm

The first algorithm focuses on defining the dimensions of the triangular shields. It takes into consideration two of the parameters of the triangle to define all the dimensions. Having right triangles as base geometries, the other two parameters are, as previously mentioned, one of the angles and its adjacent side. The values of the angle α_{13} ($30< \alpha_{13} <60$ degrees) and the side length l_1 ($0.5<l_1<1u$) will determine the degree of overlap between adjacent shields.

Figure 4.4 shows the more detailed description of the Shield-definition algorithm, starting with the initial shape values for l_1 and α_{13}. The inner loop increases the α_{13} values, while the outer loop increases the l_1 values, making sure all the combinations of l_1 and α_{13} are evaluated. This can be considered as the external algorithm, since the other two algorithms are included within the Evaluate & Score blocks.

Once the algorithm goes through all the possible shield shapes, it exits the program and reports the results. The algorithm was designed in a way in which the user can determine the values of Δl_1 and $\Delta\alpha_{13}$. It is important to highlight that making these values smaller would lead to an increase in the amount of iterations; therefore, it would take the program more time to get to the more precise results.

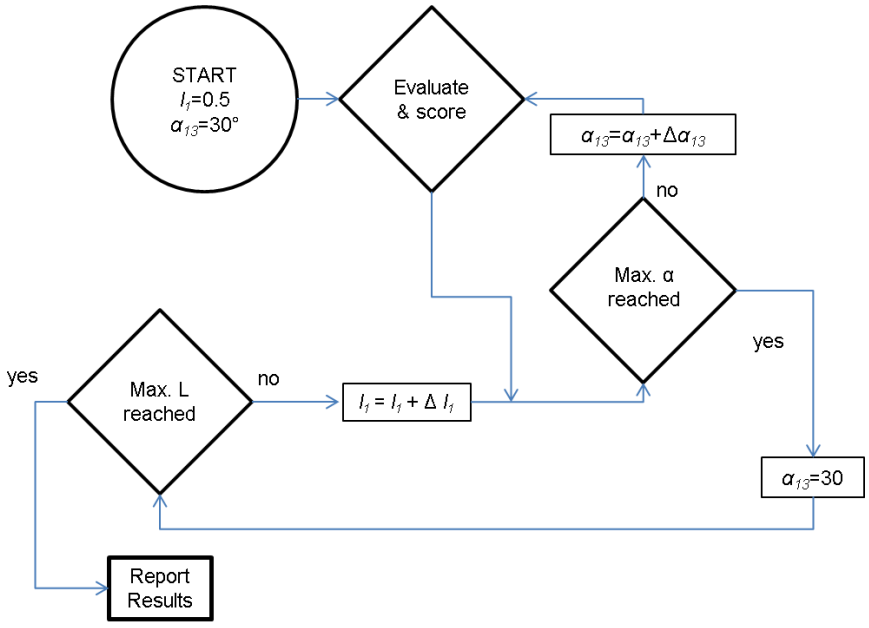

Figure 4.4: Shield definition loop

4.2.2 Unit-cell Deformation

SSSs are designed with a specific initial geometry and with the intention of accommodating deformation in order to comply with the requirements of being a physical barrier. The unit-cell deformation algorithm is the motion approach used to evaluate all the possible triangular positions of the unit-cell with each of the of the shield arrangements, starting with the initial equilateral triangle for each shield shape variation. For this study, the final shape is a triangle, but not necessarily equilateral. Three degrees-of-freedom movement was used to ensure feasible triangular shapes were taken into consideration for the analysis. For the movement, the lower left corner of the unit-cell moved horizontally, the top corner moved horizontally and vertically, and the lower right corner of the unit-cell remained fixed. Figure 4.5 shows some examples of the different shapes that can be obtained using the same shield shape. The side lengths of the unit-cell were named *p*, *q*, and *r*.

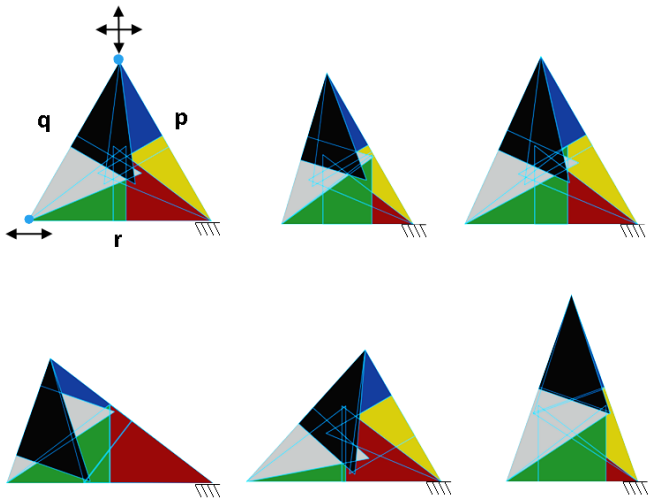

Figure 4.5: Examples of different unit-cell deformation using the same shield shape

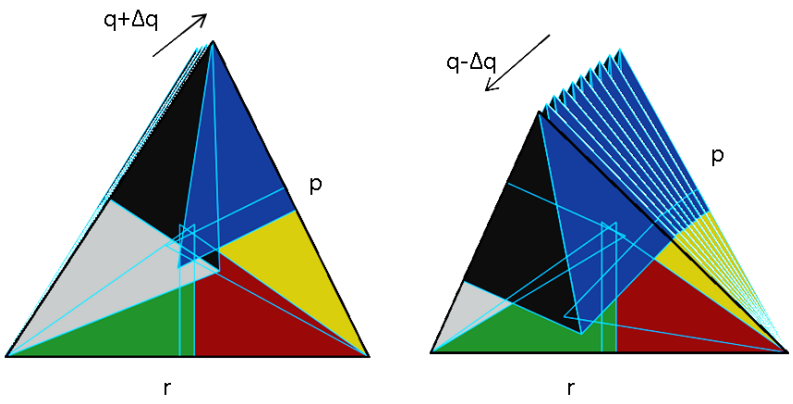

Figure 4.6: Example of a unit-cell step by step deformation

Using the described movement, each of the sides of the unit-cell *p, q*, and *r* increases its value from *1u* to *2u* with an increment ratio of Δ*p*, Δ*q*, and Δ*r* (each equal to 1/100*u*), respectively. After evaluating increasing sides of the unit-cell, the algorithm goes back to the 1*u* value, and decreases down to *0.5u*, using the same tests and increment. Figure 4.6 shows one of the movements performed by the unit-cell, in which the values of *q*, first increase until a gap or protrusion is found, then goes back to the initial value and decreases until a gap or protrusion is found while the values of *p* and *r* remain the same. The algorithm is briefly described for the increasing loop in Figure 4.7. Although the evaluation methodology is not included in this this figure, it explains in a simplified way how all the possible side length combinations were considered. The loop starts by setting the limit values for *p*, *q*, and *r*, (between 0.5*u* and 1*u*) followed by establishing the initial values of each one, which determine the initial unit-cell shape (*p*=1, *q*=1, *r*=1). All three sides begin with a value of 1*u*, and can go up to 2*u* or decrease to 0.5*u*. The increase and decrease *p* loops are nested within the increase and decrease loop of *q*, and the increase and decrease loops of *q* are

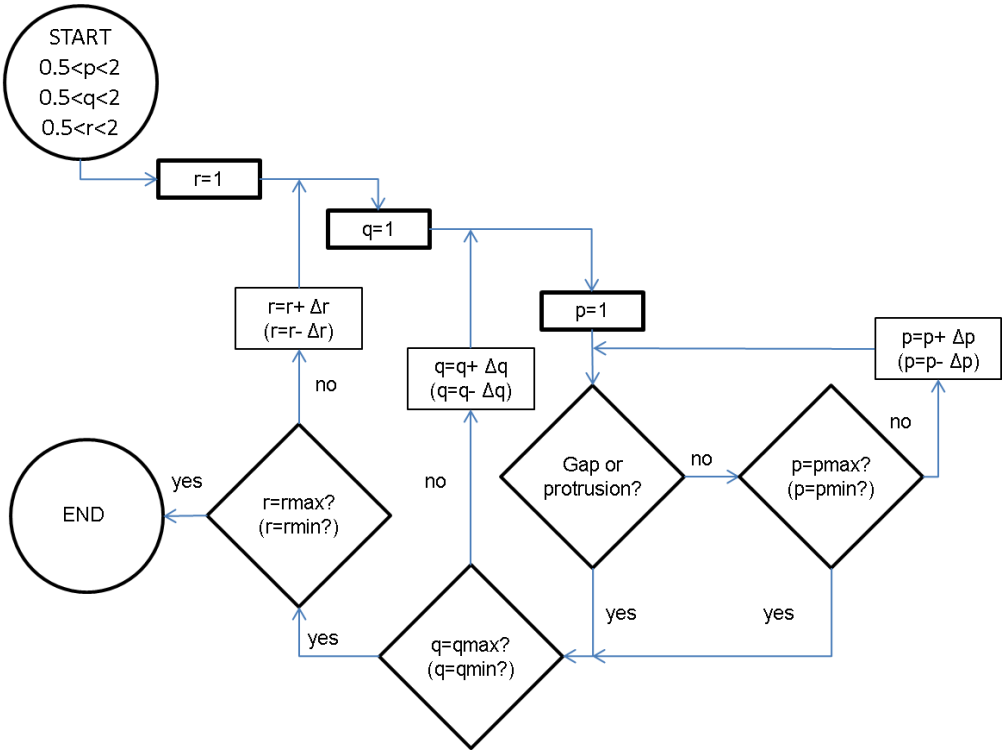

Figure 4.7: Unit-cell deformation loop: the given formulas during the increasing half of each loop are replaced by the parenthetical formulas during the decreasing half of each loop

nested within the increase and decrease loops of *r*, as seen on Figure 4.8. After establishing the initial values, the algorithm enters the *p* loop and increases only the value of *p* until a gap or protrusion is formed; it then resets the *p* value back to *p*=1 and decreases its value until a gap or protrusion is found. The algorithm then increases the value of *q*, and goes into the *p* loop again. This process is repeated until the maximum value of *q* is reached. The *q* value then goes back to *q*=1 and the described process is repeated until the minimum value of *q* is reached. The algorithm then increases the value of *r*, and the process is repeated once again, until the maximum value of *r* is reached, and *r* is set back to *r*=1, to begin the decreasing values of *r*. Figure 4.8 shows the unit cell

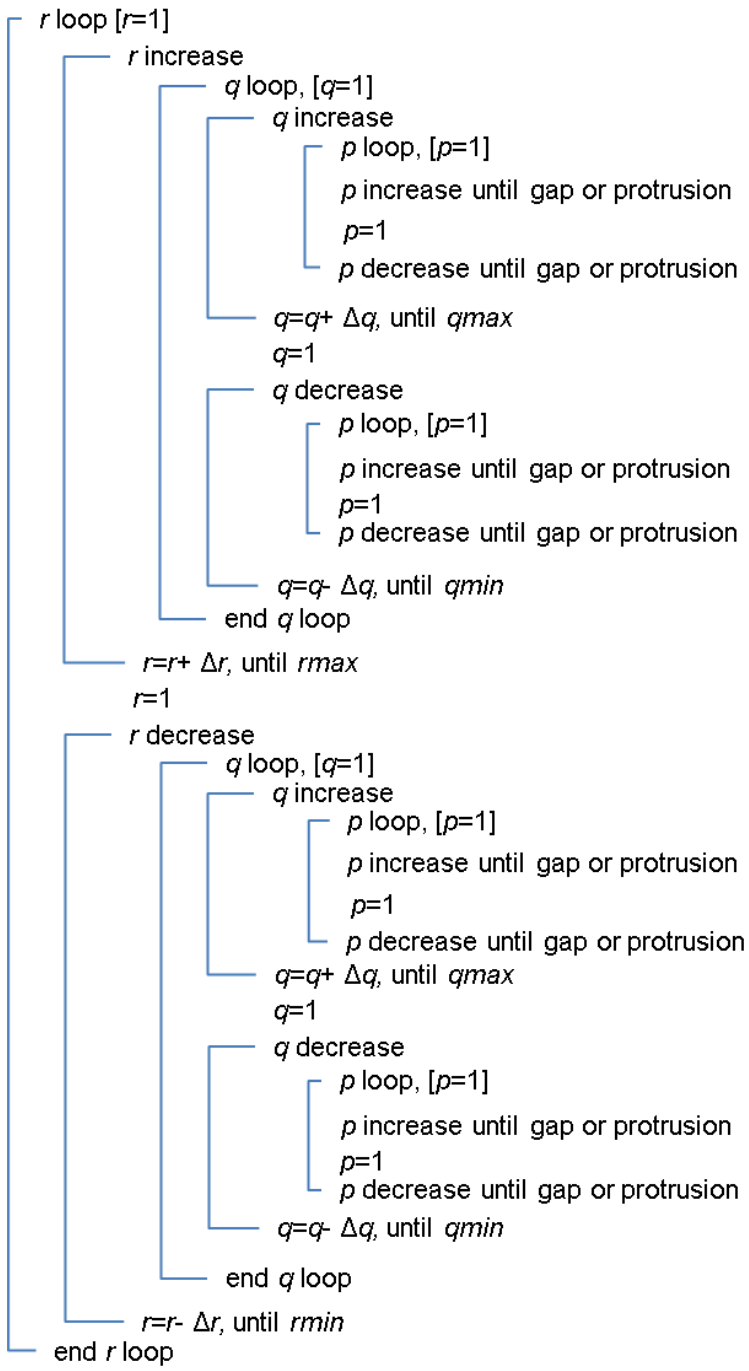

Figure 4.8: Unit cell deformation loop: the *p* loop is nested within the *q* loop; the *q* loop is nested within the *r* loop

deformation algorithm in a simpler format. The algorithm increases *p*, *q*, and *r* by

Δ*p*, Δ*q*, and Δ*r* respectively one at a time until all the combinations are evaluated.

All unit-cell deformation tests start at 1*u* and not at 0.5*u* (minimum evaluated value) because this study focuses on unit-cells with an initial equilateral triangle geometry with sides equal to 1*u*, therefore, all shield shapes must at least comply with this initial condition.

4.2.3 Integrity Assessment

The integrity assessment uses four tests to evaluate if there are any gaps or protrusions for each one of the unit-cell deformations. Deformations that have gaps or protrusions are rejected and not added to the score. There are two gap tests, one for angles and one for sides, and two protrusion tests as well, one for angles and one for protruding vertices. In order to identify gaps, the algorithm goes through a series of tests in which the all the internal angles of the unit-cell are compared to the sum of the angles α_{13} of the pair of shields that make up the corner. In addition, gaps are also identified on the sides of the unit-cell, when the total length of each side is compared to the total length l_1 of the pair of shields that make up each side. Likewise, protrusions are identified in a similar manner, but the internal angles of the unit-cells are compared to the α_{13} angle of the shields, and the unit shield side length is compared to the l_1 length of the shields. These four steps are visualized from Figure 4.9 to Figure 4.12, in which each of the gap and protrusion possibilities are shown. Figure 4.9 shows the possible gaps that are formed when one of the internal angles of the unit-cell has a greater value than the sum of the α_{13} angles of the pair of shields that form that particular corner. From left to right, Figure 4.9displays first the gaps formed at

the top vertex of the unit-cell, followed by the bottom right vertex and the bottom left vertex.

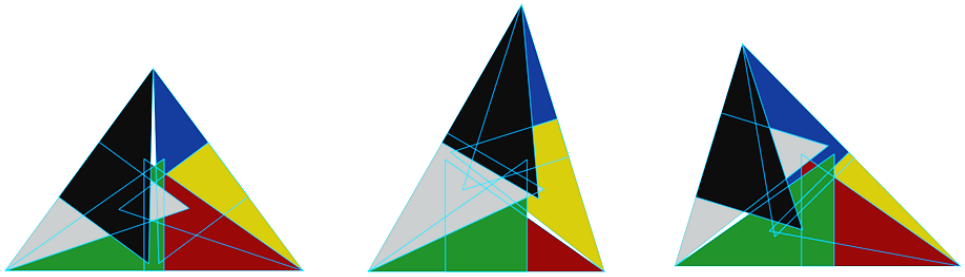

Figure 4.9: Gaps formed by angular deformations

Figure 4.10 shows the gaps formed when then length one of the sides of the unit-cell is greater than the sum of the l_1 lengths of the shields forming that side. The image shows initially a gap on the bottom side of the unit-cell, followed by gaps on the left and right sides respectively.

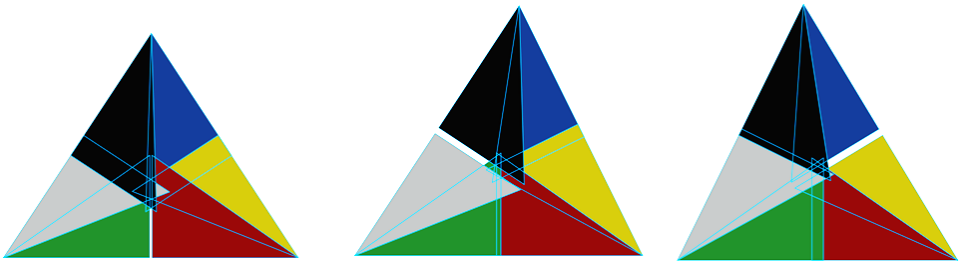

Figure 4.10: Gaps formed by side length deformations

Protrusions were also evaluated through two methods. The first one is shown in Figure 4.11, in which the a_{13} angles of the shields are greater than one of the angles of the unit-cell. From left to right, Figure 4.11 shows protrusions formed by the shields at the bottom corner of the unit-cells, followed by protrusions formed on the bottom left corner and the top corner of the unit-cell.

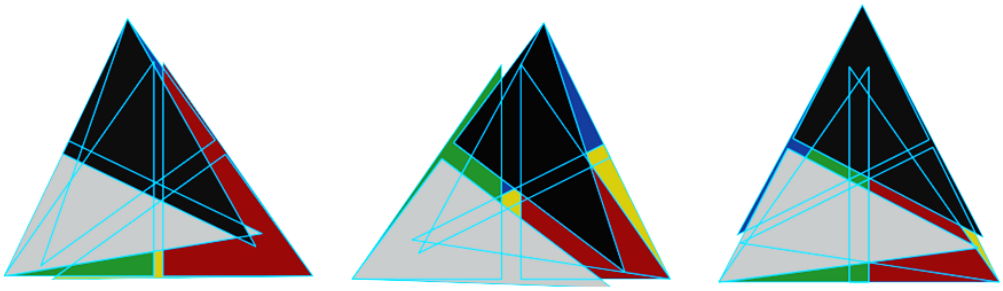

Figure 4.11: Protrusions formed by angular deformations

The final evaluation was made considering protruding shield vertices on the opposite side of the unit-cell. Figure 4.12 shows the three cases in which the corners of the pairs of shields go beyond the opposite side of the unit-cell, changing its triangular geometry. The first SSS on Figure 4.12 shows the shields located on the bottom right protruding on the left side of the unit-cell; the middle SSS shows the shields pivoted at the top corner of the unit-cell protruding on the bottom side of it; the last SSS shows the shields at the bottom left vertex of the unit-cell protruding on its right side.

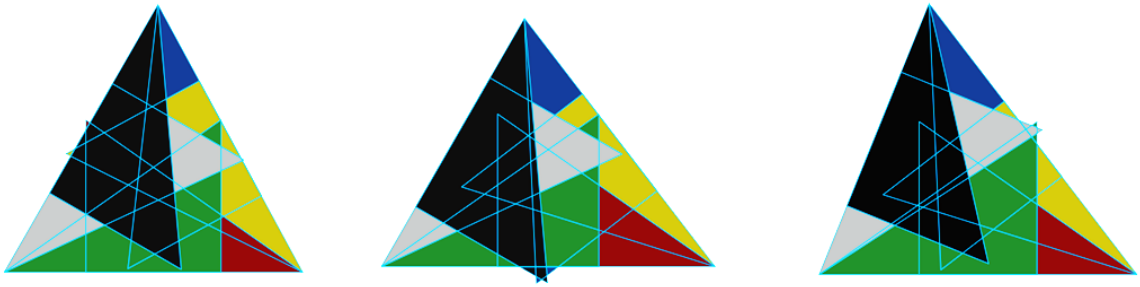

Figure 4.12: Protrusions formed by shield corners that extend outside the unit-cell

The algorithm that was developed to test gaps and protrusions was run two times for each unit-cell deformation, one for increasing values of the sides of

the unit-cell, and one for decreasing values. Figure 4.13 shows a simplified view of the integrity assessment loop. It begins with the previously obtained shield shape values of l_1 and α_{13}, and the unit-cell values for the evaluation, p, q, and r. The algorithm looks for gaps and protrusions for each unit-cell shape and if either one is found the program exits the evaluation loop and goes to the next deformation. If no gaps or protrusions are found, the program adds one to the score of that particular shield shape, and then goes to the next unit-cell deformation. Once all the unit-cell deformations are evaluated for one shield shape, the program records the score value, and moves on to the next shield shape.

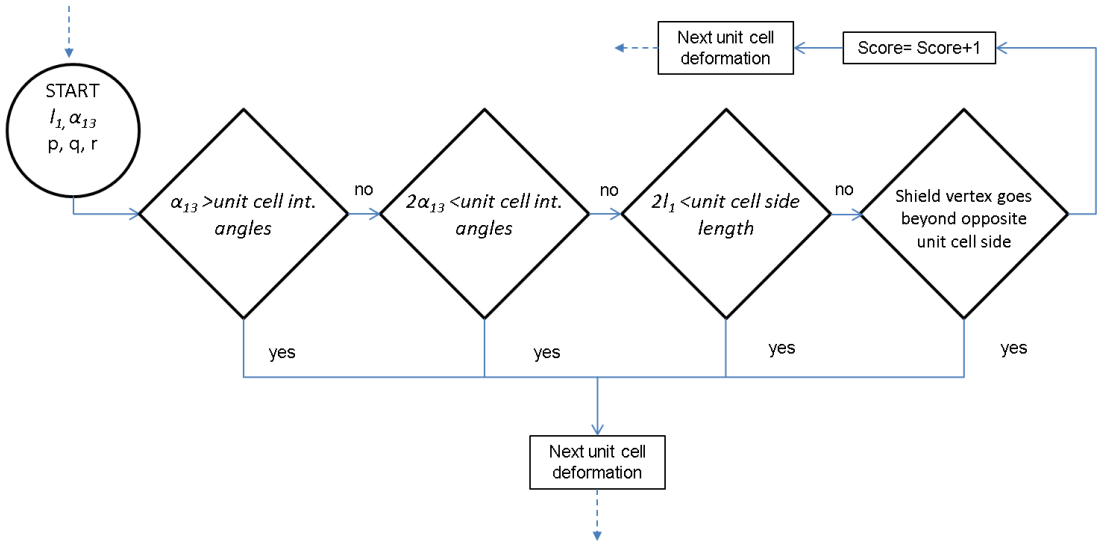

Figure 4.13: Integrity assessment algorithm

4.2.4 Other Evaluation Possibilities

There were other gap and protrusion tests that were not taken into consideration for the algorithm since none of these were physically possible when keeping the unit-cell as a physical barrier. For instance, the integrity

assessment did not look for protrusions formed when the length of l_1 was greater than the length of the side of the unit-cell because before this could happen, the integrity assessment algorithm would have found a different form of gap or protrusion with the established evaluation methods.

4.3 Scoring

The main objective of this research is designing a shield that offers the greatest mobility to the unit-cell, utilizing triangles. The algorithm evaluated all possible unit-cell deformations with each shield shape arrangement and reported the one with the highest score. A score was assigned to each shield shape depending on the number of positions that fulfilled the physical barrier requirements. If no gaps or no protrusions were found in any given unit-cell position, a unit was added to the score of the shield shape. When finalizing the evaluation for a shield arrangement, the program compares the score with previous results from other shield shapes. If the current score is higher than the previous one, the program displays the score, the l_1 length and the α_{13} angle.

4.4 Algorithm Output

To make sure the algorithm was following the desired path and evaluation method, a visual output was programmed to display the steps of the unit-cell deformations with each shield shape. Figure 4.14 and Figure 4.15 illustrate some examples of the visual output in which the lengths of *p*, *q* and *r* and the current values of l_1 and α_{13} are shown; *CS* stands for *current score* (score of the evaluated shield shape), and *BS* for *best score* (best score up to that point of the entire test). Figure 4.14 shows the initial equilateral triangle unit-cell position for

shield shapes with l_1=0.61u and α_{13}=37.25° followed by the unit-cell deformation with p=0.86u, q=1u, and r=1u, just before reaching a protrusion. The length p decreases from 1u to 0.86u with a step of 1/100u so when it reaches the protrusion, the current score is 14. The process continues in Figure 4.15, where the increase of p is evaluated until it reaches a length of 1.22u, just before a gap is identified in the unit cell.

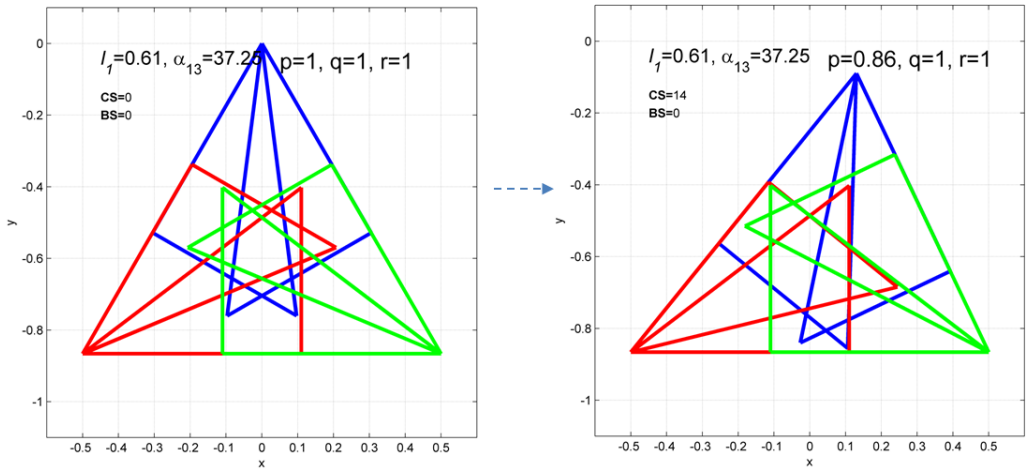

Figure 4.14: Matlab algorithm output figure for decreasing p length

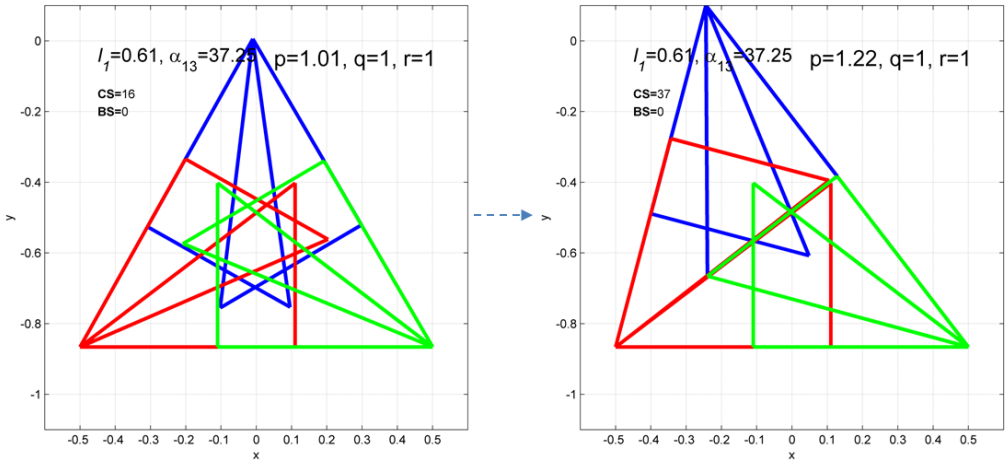

Figure 4.15: Matlab algorithm output figure for increasing p length

4.5 Algorithm Optimization

After performing initial trials with the described method and considering that the algorithm was to evaluate *150,000* shield shapes and *3,375,000* unit-cell deformations for each shield shape, it became necessary to develop an optimization method to reduce the running time of the program without affecting the final result. The initial optimization disregarded arrangements that did not increase the score of the shield shape: once a unit-cell reached a deformation with a gap or a protrusion with lengths for p, q, and r, the increment of the loop variable was terminated and the algorithm moved to the next outer loop. However, after running some tests, it was noticed that some of the successful unit-cell deformations were path-dependent, meaning that some deformations were reached by following determined deformation paths in order to maintain mechanism integrity at all times. Because of this, the initial optimization method was improved to ensure all the possible deformations were included in the score of each shield shape. The improved optimization consists of allowing a set number of failed tests before terminating the loop, allowing path-dependent points to be counted but still limiting the amount of evaluated positions that do not add value to the total score of the shield shape. As a result, total amount of unit-cell positions for each shield arrangement is reduced significantly.

CHAPTER 5: RESULTS AND DISCUSSION

This section shows the outcomes of the algorithm, and the results. It also includes the process of designing and manufacturing prototypes using compliant mechanisms and the obtained data.

5.1 Results

The total amount of tested shields was 150,000 with 3,375,000 tested positions for each shield. The shield shape that offered the maximum motion range for the unit-cell has a length l_1 of 0.61u and an angle α_{13} of 37.25°. The score for this shield shape was 33,048 successful positions, which is a considerable small value compared to the total amount of evaluated positions.

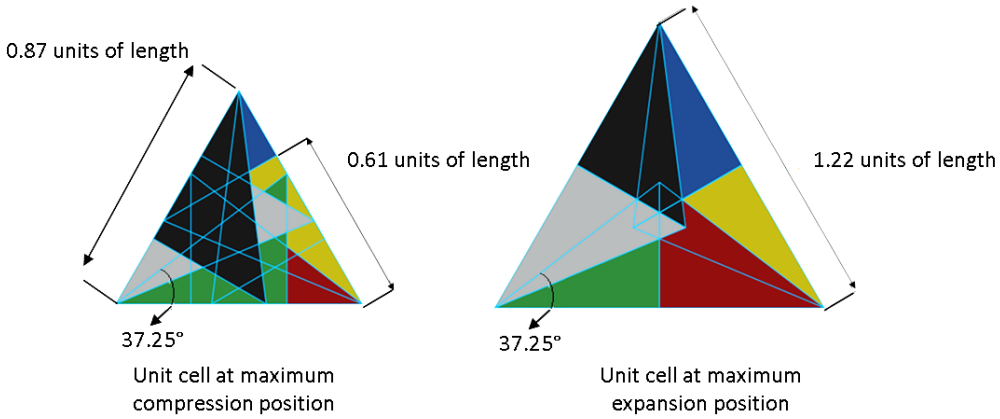

Figure 5.1: SSS with maximum compression and maximum expansion

The area ratio between maximum expansion and maximum compression is 1.93, which shows that the area can be almost doubled using a same shield shape.Figure 5.1 shows the arrangement at maximum compression and maximum expansion.

Figure 5.2, Figure 5.3, and Figure 5.4 show the 3-D plot of the unit-cell values of *p*, *q*, and *r* that can be obtained with the shield shape that allows maximum mobility. This means that all the *p*, *q*, and *r* values within the boundaries of this figure are successful unit-cell deformations that can be obtained only with a unit-cell formed by shields with α_{13} and l_1 values of 37.25° and 0.61*u* respectively. The total amount of points that make up the figure is then 33,048, which is the total score of the shield shape that offers maximum mobility. The figure is a hexahedron with triangular bipyramid geometry and shows three axes of symmetry of 120° each (visible on Figure 5.3). As expected, the symmetry indicates that if the values of *p*, *q*, and *r* are interchanged, the outcome will be the same as in the original values. The multiple views allow a better interpretation of the geometry of the figure. The figures also show the contrast between the explored space and the successful results, since the *p*, *q*, and *r* vertices range between the minimum and maximum values of the side lengths of the unit-cells that were evaluated on the algorithm. Hence, the obtained volume represents not only the possible values for a SSS, but also a visual comparison between the explored values and the successful values. As we see, the successful results are significantly less than the total explored values, which

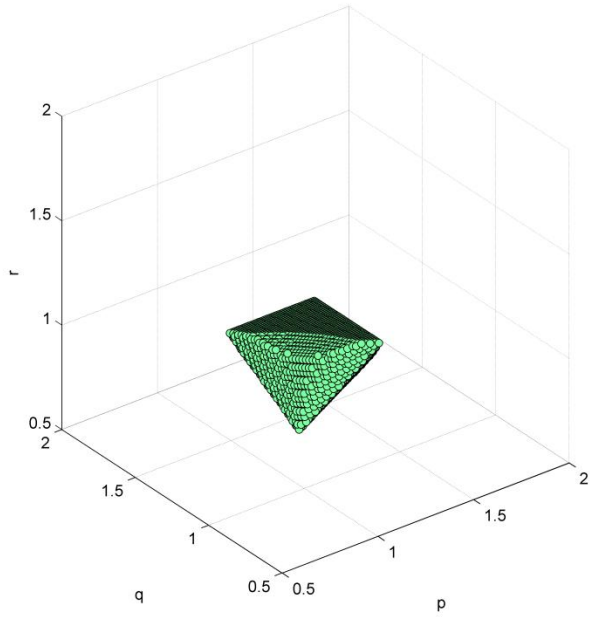

Figure 5.2: Possible values for *p*, *q*, and *r* with the maximum mobility shield, first view

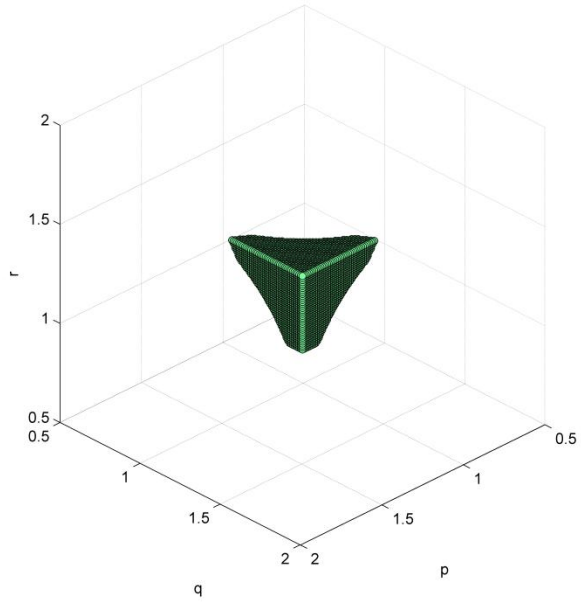

Figure 5.3: Possible values for *p*, *q*, and *r* with the maximum mobility shield, second view

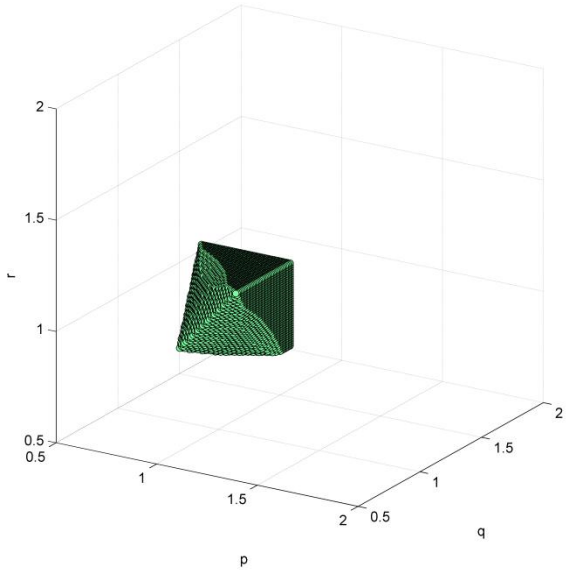

Figure 5.4: Possible values for *p*, *q*, and *r* with the maximum mobility shield, third view

indicates that even though the algorithm was optimized to disregard unit-cell deformations that did not add any value to the score, it can still be redesigned to be more efficient by reducing the amount of evaluated shield shapes. It was initially decided to have a more broad evaluation and make sure all the possibilities were tested because this is the first systematic approach it was necessary to make sure that all the possible unit-cell deformations as well as different shield shapes were tested.

The normalized scores in Figure 5.5 show in a graphical way the percentage of distribution of results, where the 1 represents the score of 33,048 obtained by the best shield shape. We can see that the shield shapes that provided a higher amount of deformations for the SSS are between 35 and 37 degrees for the values of α_{13} and between 0.57 and 0.67 for the length of l_1. These results show that there are other shield shapes that can have similar

results to the one with the highest mobility. Within this range, different designs of unit-cells can be made depending on the application and need of mobility.

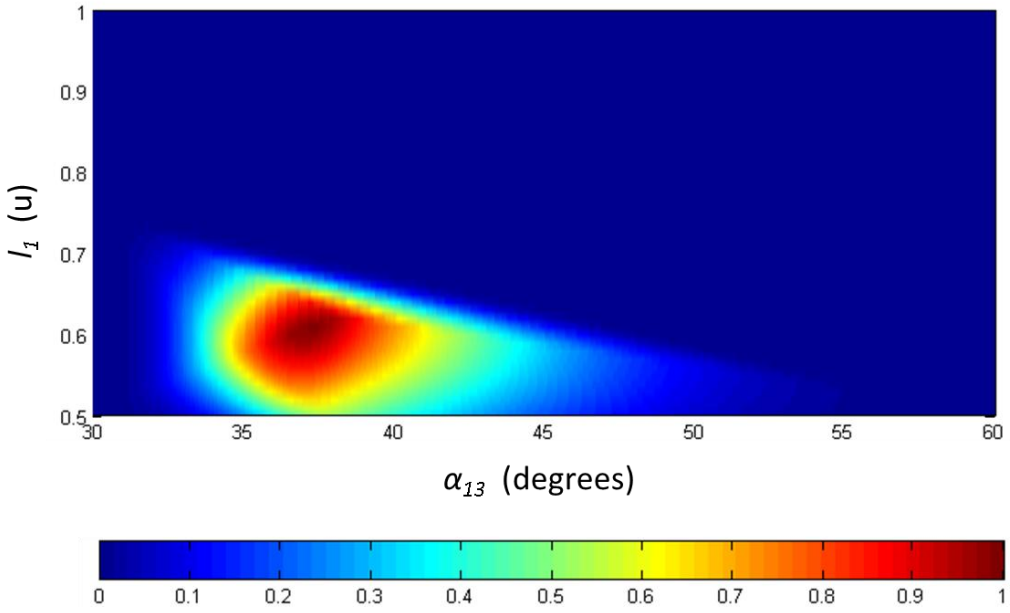

Figure 5.5: Normalized shield scores, length of l_1 vs. angle α_{13}

5.2 Prototype

To build a prototype, a sheet of polypropylene was cut with an 85 *watt* laser cutter that uses CNC control. For the manufacturing process it was necessary to create a virtual model of the design using Solidworks. The file was saved in the AutoCAD .dfx format for it to be read by the CNC software used by the laser cutter.

The use of the laser offers advantages over other manufacturing processes (e.g. milling machine) because of the accuracy and the reduced amount of time that it takes to fabricate the parts. The laser cutter however, cannot manufacture

micro sized elements; these types of shields would need the use of nano-manufacturing technologies due to the elevated degree of dimensional accuracy required on the shield shapes and compliant segments [8].

The compliant segments for each link were taken from previous work done by Lusk and Montalbano [1, 2]. Figure 5.6 shows a link sample drawn on CAD. The compliant segments at this level of assembly do not comply with the SSSs characteristics (no gaps, no protrusions), since the current research was focused on the shield design. The compliant segments were slightly modified to generate a straight line relative motion between adjacent links located on the same side of the unit-cell. Figure 5.7 shows different deformations the assembled prototype. The deformations show the compliant segments protruding out of the triangular geometry, but the shield portion remains within the geometry, maintaining the mechanism integrity.

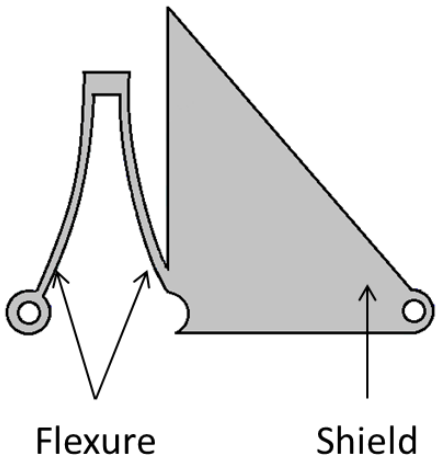

Figure 5.6: Link sample

Figure 5.7: Assembled prototype

CHAPTER 6: CONTRIBUTIONS AND RECOMMENDATIONS

The objective of this chapter is to highlight the most significant contributions of this paper in the field of study as well as recommendations for future work.

6.1 Contributions to Systematic SSSs Design and Future Work

The developed algorithm parameterizes feasible shields for a given unit-cell; considers all possible unit-cell deformations; and scores shields based on deformations that maintain integrity. With this algorithm, the shield that offered maximum deformation capability was determined. This algorithm can be used as foundational work for future research on SSSs with more complex geometries.

6.1.1 More Complex Geometries

Triangles are the simplest polygon geometry, and every polygon with more than 3 sides can be broken down into triangles. Circles can also be separated into very small triangles, in which the circumference is made up by very small segments that can be unnoticeable for a determined application.

The shield shape that offered most mobility had unnecessary overlap at its maximum expansion point. Another approach for a similar study could be removing the edges that overlap at maximum expansion (creating an irregular quadrilateral) and recreating the same algorithm to see how the removal of material will affect the unit-cell deformations. Figure 6.1shows the shield shape that could have this new approach. To complete it, new parameters and must be

taken into consideration since there is an additional angle and an additional side length in the figure. New variables and constants will also come into play when determining the length and angular position of the new side.

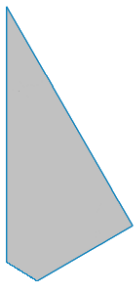

Figure 6.1: Possible geometry for future work

6.1.2 Algorithm Optimization

Although the developed algorithm was optimized to deliver results in a faster way, there are still some unnecessary evaluations that were included to make sure all the possibilities were evaluated, especially for the path-dependent deformations. Improved algorithms can include classifying the failure reasons for the unit-cell deformations; that is, being able to show which of the integrity evaluation tests (gap or protrusion) is responsible for each failure.

6.1.3 Tiling

Other analyses can also include the tiling of SSSs and studying their behavior when shifting geometry. This would require additional extensions to the existing algorithm since interactions between unit-cells must be considered. The covered area, surface geometry, and amount of unit-cells of the tiling can be determined based on the application. The type of tiling might also require the use of different shapes of unit-cells.

6.1.4 Shape-Shifting Sufraces in 3-D

Another systematic approach can be applied to Shape-Shifting Surfaces in three dimensions. This analysis has a significant increased complexity since the additional dimension adds more degrees of freedom to the SSS.

6.2 Applications

As stated earlier in this thesis, one of the initial motivation areas is the development of better body armor systems that can protect users in a better way by increasing covered areas without limiting movement. A surface that can change its shape and increase area coverage by nearly twice will result can be the initial step in developing this type of body armor. Amor made up of Shape-Shifting Surfaces may not only protect areas that are normally uncovered (limb joints, neck), but may also reduce the fatigue to the user caused by having limited movement.

6.3 Recommendations

Before developing a new algorithm that focuses on more complex structures or tiling, it is important to design flexure components that are able to satisfy the characteristic of no gaps and no protrusions. A similar approach to the one used in this research can be used when designing the integrity assessment. Compliant structures will require the use of compliant mechanisms theories and equations (i.e. pseudo rigid body model). A similar methodology for identifying gaps and protrusions can be used with the integration of the compliant segments.

Future algorithms can be optimized as well by reducing the amount of evaluated shield shapes and unit-cell deformations. This can significantly reduce the amount of time it takes for the algorithm to run, and more significant digits can be added to the change ratio between shield shapes and unit-cell deformations.

CHAPTER 7: CONCLUSIONS

This chapter includes a final summary of the research, results, and applications. Previous works on Shape-Shifting Surfaces include research on multi-stable mechanisms [1] and statically balanced SSSs made up by multiple tiled cells, which means that there is a near absence of forces within the mechanism [8]. This research offers a first systematic focus on shield shape design, integrating computer programming with SSSs design and its purpose was to finding a triangular shield shape able to offer maximum mobility, i.e. maximum compression and maximum expansion for triangular unit-cells.

Chapter 2 described related material in previous work done on Shape-Shifting Surfaces and similar areas. SSSs were classified as smart structures since they are able to adapt to their surroundings, changing shape with externally applied forces. This chapter also described the potential application of SSSs, body armor systems.

Chapter 3 introduced the concepts that were needed to understand the developed process to reach the objective as well as the shield shapes that were studied. The variable shield parameters α_{13} and l_1 were defined and the assembly and deformation of the unit-cells were explained.

Chapter 4 focused on the development of the algorithm. The algorithm considerations included a general loop in which each of the shield shapes was

evaluated on every possible unit-cell deformation. The algorithm was optimized by reducing the amount of evaluated deformations disregarding shapes that did not add value to the scoring of the shield shape.

The main algorithm was made up by three sub algorithms: the first one and outer loop controlled the shield variables; the second one determined the unit-cell deformations; the third one and inner most loop evaluated each unit-cell looking for gaps and protrusions and scored the shield shapes.

Chapter 5 showed that the shield shape with α_{13} equal to 37.25° and l_1 equal to 0.61u offered a the maximum mobility for the triangular unit-cells. These results were also shown through different graphical interpretations. This chapter showed and described the steps and considerations made to obtain the prototype using polypropylene and a laser cutter.

Finally, Chapter 6 focused on the contributions made to the SSSs study, this research being the first systematic approach to shield design and initiating an algorithm based design methodology for future research.

LIST OF REFERENCES

[1] Montalbano, P., 2012, "Multistable Shape-Shifting Surfaces (MSSSs)". *Graduate School Theses and Dissertations.* http://scholarcommons.usf.edu/etd/4169.

[2] Lusk, C., and Montalbano, P., 2011, "Design Concepts for Shape-Shifting Surfaces," in *Proceedings of the ASME 2011 International Design Engineering Technical Conferences,* Washington, DC, Aug 29-31, 2011, DETC2011-47402.

[3] Suleman, A. 2001, “Smart Structures: applications and related technologies”, Springer, New York.

[4] Janocha, H., 2007, “Adaptronics and Smart Structures: Basics, Materials, Design, and Applications”, Springer, New York.

[5] Howell, L.L., 2001, "Compliant Mechanisms," Wiley, New York.

[6] Frecker, M., Mehta, V., and Lesieutre, G., 2009, “Stress Relief in Contact-Aided Compliant Cellular Mechanisms”, *Journal of Mechanical Design*, Trans. ASME, Vol. 131, Sept. pp. 091009-1 to 091009-11.

[7] Hoetmer, Karin, Herder, Just L and Kim, Charles. *“*A Building Block Approach for the Design of Statically Balanced Compliant Mechanisms”. *International Design Engineering Technical Conference* San Diego, California, USA,2009. Vols. DETC2009-87451.

[8] Pishnery, J., (2012) "A Statically Balanced Shape-Shifting Surface". *Graduate School Theses and Dissertations.* http://scholarcommons.usf.edu/etd/4204.

[9] Bin, H., Zhen, L., Feng, L.,2010. A Survey on Underactuated Mechanisms. *Second International Conference on Advanced Computer Control (ICACC).*pp. 551 to 555

[10] Spong, M. 1998. Underactuated Mechanical Systems. *Control Problems in Robotics and Automation*, 230.

[11] Spong, M., Lewis, F., and Abdallah, C. *Robot Control: Dynamics, Motion Planning, and Analysis.* IEEE Press, 1992.

[12] Craig, John J.(2005) “Introduction to Robotics Mechanics and Control”, Pearson Prentice Hall, NJ.

[13] Goldstein,S. and Mowry, T., “Claytronics: A scalable basis for future robots,” *In RoboSphere 2004,* November, 2004.

[14] Pelletier, D. (2012). Shape-Shifting Robots: Never Forget Nanotech; Think Claytronics. www.positivefuturist.com. July 14.

[15] Rus, D. (2010) “Programmable matter with self-reconfiguring robots,” *7th ACM international conference on Computing frontiers,* ACM, New York, NY.

[16] Wayne J. Book and Haihong Zhu, “Control concepts for digital clay” IFAC, Atlanta GA, 2002.

[17] Selinger, J., Gooyers, C., Stevenson, J., Costigan, R., and Chafefensen, G., 2010, “Case Study: A Novel Biomechanical Approach for Evaluating Extended Body Armor Systems”, *Military Medicine, International Journal of AMSUS*. Vol. 175, Nov. pp. 871-875.

[18] Iraq Coalition Casualty Count: Available at http://www.icasualties.org; accessed March 2009.

[19] Kosashvili Y, Hiss J, Davidovic N, et al. Influence of personal armor on distribution of entry wounds: lessons learned from urban-setting warfare fatalities. J Trauma. 2005;58(6):1236–40.

[20] Mabry RL, Holcomb JB, Baker AM, et al. United States Army Rangers in Somalia: an analysis of combat casualties on an urban battlefield. J Trauma. 2000;49(3):515–28, discussion 28e9.

[21] Peleg K, Rivkind A, Aharonson-Daniel L. Does body armor protect from firearm injuries? J Am Coll Surg. 2006;202(4): 643–8.

[22] Konitzer, L., Brininger,T., and Reed, M., 2008 “Association between Back, Neck, and Upper Extremity Musculoskeletal Pain and the Individual Body Armor”. *Journal of Hand Therapy,* Special Issue # 88, Apr.-Jun. p.p. 143-149.

[23] GlobalSecurity.org. Advanced combat helmet (ACH) Available at http://www.globalsecurity.org/military/systems/ground/ach.htm. Accessed Apr 25, 2007.

APPENDICES

Appendix 1: Matlab Main Code

This is the Matlab code developed for this thesis. This code includes only the main algorithm and does not include the developed external functions.

```
% clc
clf
clear all

a12=90;
syms l1_Best a13_Best;
BestScore=0;
CurrentScore=0;
NewScore=0;
Ch_rt=(1/100);%change rate for increasing or decreasing sides

Iscore=zeros((1-0.5)/Ch_rt+1,(60-30)/Ch_rt+1);
i=0;


for l1 =[0.5:Ch_rt:1],%Length of side of shield
    i=i+1;
    j=0;
    for a13 =[35:Ch_rt:60],%Internal angle a13 of shield
        j=j+1;
        if CurrentScore>BestScore;
            BestScore=CurrentScore;
            VolumeScore=BestScore;
            disp(['Best lenth 1=',num2str(l1),' units'])
            disp(['Best alpha 13 angle=',num2str(a13),'°'])
            disp(['Volume Score=',num2str(VolumeScore)])

        end
        CurrentScore=0;
        flag_6=1;
        flag_7=1;
        r_dec=1;
        r_inc=2;
        while flag_6==1;%flag that exits the while loop and goes no
next a13 angle
            NewScore2=CurrentScore;
            if flag_7==1%flag that detremines if Lr increases or
decreases
                Lr=1-(r_dec-1)*Ch_rt;%decrease Lr
            else
                Lr=1+(r_inc-1)*Ch_rt;%increase Lr
end


            flag_4=1;
            flag_5=1;
```

Appendix 1 (Continued)

```
            q_dec=1;
            q_inc=2;
            while flag_4==1;%flag that exits the while loop and goes no
next Lr length
                NewScore=CurrentScore;
                if flag_5==1;%flag that determines if Lq increases or
decreases

                    Lq=1-(q_dec-1)*(Ch_rt);%decrease side Lq
                else
                    Lq=1+(q_inc-1)*(Ch_rt);%increase side Lq
                end

                k=1;%counter that increases Lp
                w=1;%counter that decreases Lp

                flag_2=1;
                flag_3=1;

                while flag_2==1;%flag that exits the while loop and
goes no next Lq length

                    if flag_3==1;%flag that determines if Lp increases
or decreases

                        %THE FOLLOWING LINE DEFINES THE MOVING VERTICES
OF THE UNIT CELL FROM THE FUNCTIONS "Angles"
                        [Angle1_4 Angle2_5 Angle_1 Angle_2 a3x a3y
a3x_34 a3y_34 a3y_56 a3x_56 Lp angle_q
angle_q_comp]=Angles(k,Lq,Lr,Ch_rt);%this line defines the decreasing
Lp
                    elseif flag_3==0;
                        w=w+1;
                        [Angle1_4 Angle2_5 Angle_1 Angle_2 a3x a3y
a3x_34 a3y_34 a3y_56 a3x_56 Lp angle_q
angle_q_comp]=Angles_2(w,Lq,Lr,Ch_rt);%this line defines the increasing
Lp

                    end


                    %THE FOLLOWING LINE DEFINES INTERNAL ANGLES OF THE
UNIT CELLS
                    [GF a23]=intAngles3(l1,a13,a12);



                    %THE FOLLOWING LINE DEFINES THE INITIAL ORIGINAL
TRIANGLES
                    [Gx Gy fx fy Gx2 Gy2 fx2
fy2]=getTriangles(l1,GF,a12);
```

Appendix 1 (Continued)

```
                    %Defining vectors for G and f
                      Gv=[Gx;Gy;0;1];
                      fv=[fx;fy;0;1];

                      Gv2=[Gx2;Gy2;0;1];
                      fv2=[fx2;fy2;0;1];

                      %Transform Equation for triangle 1
                      RotAng1=[cosd(Angle_1) -sind(Angle_1) 0
a3x;sind(Angle_1) cosd(Angle_1) 0 a3y;0 0 1 0;0 0 0 1];
                      %Transform Equation for triangle 2
                      RotAng2=[cosd(Angle_2) sind(Angle_2) 0 a3x;-
sind(Angle_2) cosd(Angle_2) 0 a3y;0 0 1 0;0 0 0 1];
                      %Transform Equation for 3 (rotation and translation
of G and f):
                      RotAng3=[cosd(240) sind(240) 0 a3x_34;-sind(240)
cosd(240) 0 a3y_34;0 0 1 0;0 0 0 1];
                      %Transform Equation for 4 (rotation and translation
of G and f)
                      RotAng4=[cosd(240-Angle_1) sind(240-Angle_1) 0
a3x_34;-sind(240-Angle_1) cosd(240-Angle_1) 0 a3y_34;0 0 1 0;0 0 0 1];
                      %Transform Equation for 5 (rotation and translation
of G and f):
                      RotAng5=[cosd(120+Angle_2) sind(120+Angle_2) 0
a3x_56;-sind(120+Angle_2) cosd(120+Angle_2) 0 a3y_56;0 0 1 0;0 0 0 1];
                      %Transform Equation for 6 (rotation and translation
of G and f):
                      RotAng6=[cosd(120) sind(120) 0 a3x_56;-sind(120)
cosd(120) 0 a3y_56;0 0 1 0;0 0 0 1];

                      %Triangle 1
                      %THIS LINE DEFINES THE VECTORS FOR TRIANGLES 1 AND
2
                      [Gx1 Gy1 fx1 fy1 Gx2 Gy2 fx2
fy2]=rotTriang12(RotAng1,RotAng2,Gv,fv,Gv2,fv2);

                      %Triangles 3 and 4
                      [Gx3 Gy3 fx3 fy3 Gx4 Gy4 fx4
fy4]=rotTriang34(RotAng3,RotAng4,Gv,fv,Gv2,fv2);

                      %Triangles 5 and 6
                      [Gx5 Gy5 fx5 fy5 Gx6 Gy6 fx6
fy6]=rotTriang56(RotAng5,RotAng6,Gv,fv,Gv2,fv2);

                      %The following Function determines the angles of
the major
                      %triangle; the angles should not exceed 60°
                      [Angle_12 Angle_34
Angle_56]=getAxisAngles(Gx1,Gx2,Gx3,Gx4,Gx5,Gx6,Gy1,Gy2,Gy3,Gy4,Gy5,Gy6
,a3x,a3y,a3x_34,a3y_34,a3x_56,a3y_56);
```

Appendix 1 (Continued)

```
%The following equations use the form 0=(y2-y1)/(x2-x1)*(x-x1)+(y1-y)
to
                    %build the line equation, where x1,y1 and x2,y2 are
the axes of the mayor
                    %triangle, while x and y are coordinates of the f
point of the unit cell
                    %triangles. If the f point of the unit cell
triangles goes above the line,
                    %a protrusion is formed

                    [eq_1_2_5 eq_1_2_6 eq_1_3_3
eq_1_3_4]=getLine(a3x,a3x_34,a3x_56,a3y,a3y_34,a3y_56,fx3,fx4,fx5,fx6,f
y3,fy4,fy5,fy6);

                    %Length of the sides of the unit cell triangle;
r=bottom
                    %side. q=left side. r=right side
                    p=sqrt((a3x_56-a3x)^2+(a3y_56-a3y)^2);
                    q=sqrt((a3x_34-a3x)^2+(a3y_34-a3y)^2);
                    r=sqrt((a3x_56-a3x_34)^2+(a3y_56-a3y_34)^2);



plot([a3x,Gx1,fx1,a3x,Gx2,fx2,a3x],[a3y,Gy1,fy1,a3y,Gy2,fy2,a3y],'b',
'LineWidth',3),axis([-.6 .6 -1.1 0.1])
                    axis square
                    text(0.05,-0.05,['p=',num2str(p),',
q=',num2str(q),', r=',num2str(r)],'FontSize',18)
                    text(-0.45,-0.05,['{\itl1}=',num2str(l1),',
\alpha=',num2str(a13)], 'Fontsize',16)
                    text(-0.45,-
0.15,['{\bfCS=}',num2str(CurrentScore)])
                    text(-0.45, -0.20,['{\bfBS=}',num2str(BestScore)])
                    xlabel ('x')
                    ylabel ('y')
                    grid on
                    hold on


plot([a3x_34,Gx3,fx3,a3x_34,Gx4,fx4,a3x_34],[a3y_34,Gy3,fy3,a3y_34,Gy4,
fy4,a3y_34], 'r', 'LineWidth',3)


plot([a3x_56,Gx5,fx5,a3x_56,Gx6,fx6,a3x_56],[a3y_56,Gy5,fy5,a3y_56,Gy6,
fy6,a3y_56], 'g', 'LineWidth',3)
                    hold off



                    pause(0.001)

                    if (flag_3==1);
```

Appendix 1 (Continued)

```
%                         CurrentScore=CurrentScore+1;
                        if (Angle_12>a13*2)|| (Angle_34>a13*2) ||
(Angle_56>a13*2);% if internal angles of unit cells add up more than
external angle, gap will be formed;
                            flag_3=0;
%                             CurrentScore=CurrentScore-1;
                        elseif (Angle_12<a13)|| (Angle_34<a13) ||
(Angle_56<a13);%if internal angles of unit cells are greater than
external major triangle angles, protrusion will be formed
                            flag_3=0;
%                             CurrentScore=CurrentScore-1;
                        elseif (Lp>l1*2)|| (Lq>l1*2) || (Lr>l1*2);%if
distance l1 in internal triangles is less than the lenght of sides of
external triangle, gaps are formed
                            flag_3=0;
%                             CurrentScore=CurrentScore-1;
                        elseif (eq_1_2_5<0 || eq_1_2_6<0 || eq_1_3_3<0
|| eq_1_3_4<0);%if the f point of the minor triangles 3,4,5, and 6 go
above the lines that unite top and bottom axes, a protrusion is formed
                            flag_3=0;
%                             CurrentScore=CurrentScore-1;
                        elseif (fy1<a3y_34)||(fy2<a3y_34);%if the f
point from triangles 1 and 2 go below the horizontal base, a protrusion
is formed
                            flag_3=0;
%                             CurrentScore=CurrentScore-1;
                        else
                            CurrentScore=CurrentScore+1;
                        end

                    elseif (flag_3==0);
%                         CurrentScore=CurrentScore+1;
                        if (Angle_12>=a13*2)|| (Angle_34>=a13*2) ||
(Angle_56>=a13*2);% if internal angles of unit cells add up more than
external angle, gap will be formed;
                            flag_2=0;
%                             CurrentScore=CurrentScore-1;
                        elseif (Angle_12<=a13)|| (Angle_34<=a13) ||
(Angle_56<=a13);%if internal angles of unit cells are greater than
external major triangle angles, protrusion will be formed
                            flag_2=0;
%                             CurrentScore=CurrentScore-1;
                        elseif (Lp>l1*2)|| (Lq>l1*2) || (Lr>l1*2);%if
distance l1 in internal triangles is less than the lenght of sides of
external triangle, gaps are formed
                            flag_2=0;
%                             CurrentScore=CurrentScore-1;

                        elseif (eq_1_2_5<0 || eq_1_2_6<0 || eq_1_3_3<0
|| eq_1_3_4<0);%if the f point of the minor triangles 3,4,5, and 6 go
above the lines that unite top and bottom axes, a protrusion is formed
                            flag_2=0;
%                             CurrentScore=CurrentScore-1;
```

Appendix 1 (Continued)

```
                        elseif (fy1<=a3y_34)||(fy2<=a3y_34);%if the f
point from triangles 1 and 2 go below the horizontal base, a protrusion
is formed
                            flag_2=0;
%                             CurrentScore=CurrentScore-1;
                        else
                            CurrentScore=CurrentScore+1;
                        end

                    end %Ends if loop for flag_3

                    k=k+1;%increase counter k by 1
                end %ends while loop for flag_2

                if flag_5==1, q_dec=q_dec+1;
                else q_inc=q_inc+1;
                end

                if flag_5==1;
                    if NewScore==CurrentScore, flag_5=0; end
                else
                    if NewScore==CurrentScore, flag_4=0;end
                end


            end %ends while loop for flag_4

            if flag_7==1, r_dec=r_dec+1;
            else r_inc=r_inc+1;
            end

            if flag_7==1;
                if NewScore2==CurrentScore, flag_7=0;end
            else
                if NewScore2==CurrentScore, flag_6=0;end
            end

        end %ends the while that initiates decrease of Lr--> flag_6=1
        Iscore(i,j)=CurrentScore;
    end %ends for loop of a13
end %ends for loop of l1

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%.......GRAPHIC RESUTLS.......%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%

%PLOTING THE CONTOUR
figure
contour([30:Ch_rt:60],[0.5:Ch_rt:1],Iscore)%plots contour
```