Modeling Tensegrity Systems via Energy Minimization

by

Adrian Muresan

A thesis submitted to the

School of Computing

in conformity with the requirements for

the degree of Master of Science

Queen's University
Kingston, Ontario, Canada
August 2014

Abstract

Tensegrity systems are a type of structural system relying on a balance of tension and compression forces to maintain structural soundness. These systems have a wide variety of applications ranging from architecture to biological modeling, art, and even space exploration. This thesis provides a flexible modeling platform for tensegrity systems, allowing exploration of a wide range of systems, including fractal and adaptive tensegrity systems. In order to provide the necessary flexibility for scientific exploration, this framework incorporates a hierarchical object definition structure. A hill climbing algorithm is provided for finding minimal potential energy states of these systems. Extensive validation of the presented hill climbing algorithm shows that this algorithm finds global minima in 99% of test cases. This framework employs a clear distinction between object definition, object sampling, and object optimization, to allow for a greater range of uses.

Acknowledgements

I would like to thank the School of Computing for providing me with a wonderful learning environment and extremely helpful professors, administrative, and general support staff which have made my experience at Queen's a rich and fulfilling one. In particular, I want to express my gratitude to my supervisor, Dr. Dorothea Blostein, without whose urging and guidance I would not have been able to assemble this thesis. Thank you very much for the large amounts of time and energy you invested both into our meeting and research discussions and throughout the writing process.

I would also like to thank my friends, and most of all my family, for always being supportive and being there when I needed them.

Contents

List of Figures

List of Tables

Glossary

Symbol		Meaning
O	-	An object in a tensegrity system. Objects are referred to as **atomic** if they have no constituent sub objects, and as **compound** otherwise.
$L(O)$	-	The length of an atomic object O.
$N(O)$	-	The natural length of an atomic object O. This is the length an object would have without the application of compressive or stretching forces.
$d(O)$	-	The difference between $L(O)$ and $N(O)$. If $d(O) > 0$ this is referred to as a stretch, otherwise it is a compression.
$K(O)$	-	The elasticity, or spring constant, of an atomic object O. Inelastic objects are treated as having elasticity 0 for ease of notation.
V_O	-	A state vector of an object O.
$\mu_O(V_O)$	-	μ_O defines a mapping between state vectors V_O and states of the object O
$E_P(V_O)$	-	The potential energy of an object O in the state V_O.
$\mathbb{S}(O)$	-	The state space of the object O.

Chapter 1

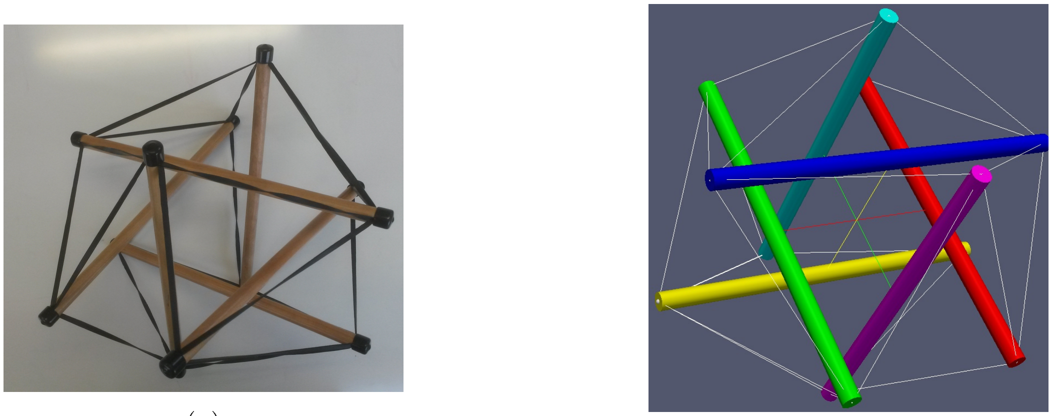
Introduction

A tensegrity system is a physical structure made up of tensioned and compressed objects. Figure 1.1 shows a tensegrity icosahedron in a stable state configuration. Tensegrity systems are used in applications including architecture, art, design, modeling connective tissues, and space exploration, as discussed in Section 1.4. This thesis constructs a flexible simulation platform that supports exploration and experimentation of adaptive and non-adaptive tensegrity systems. Section 1.5 details these contributions.

Definition. *A tensegrity system is said to be in a* ***stable state*** *if in the absence of external forces the system remains at rest in that state.*

1.1 Advantages of Simulation

Simulating physical systems allows us to create and explore a variety of systems cheaply and efficiently. There are many advantages to modeling tensegrity systems using computer simulations. Some of the advantages of using theoretical models to simulate physical systems are discussed by Yucesoy and Huijing [23]. Among these advantages is the ease of designing idealized systems for use in hypothesis testing and generation. Further, simulations allow us to experiment with values which are difficult or even impossible to create experimentally, or



(a)

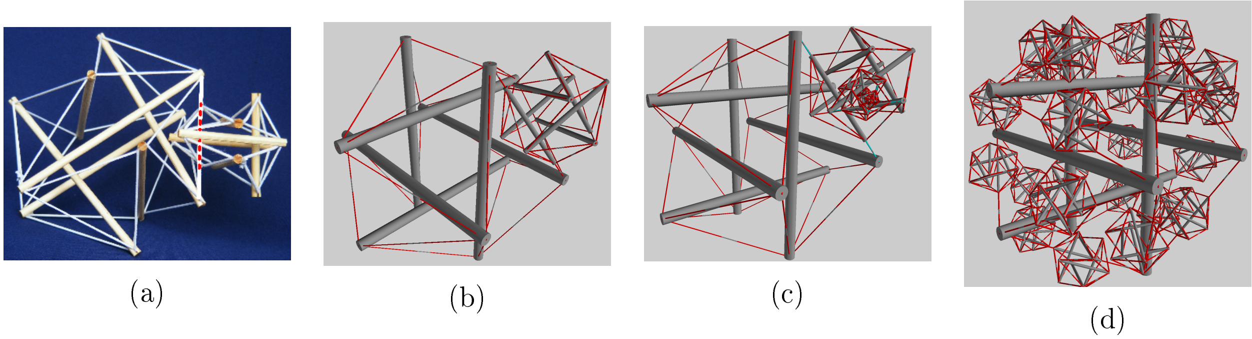
(b)

Figure 1.1: The Tensegrity Icosahedron : Although none of the rigid objects in this systems are in contact, the system as a whole maintains a shape and has a certain structural soundness. (a) shows a physical example of a tensegrity icosahedron, while (b) shows our simulation of it.

which would be difficult or impossible to measure accurately.

Many interesting physical tensegrity models have been built by various artists, engineers and physiology researchers. This is illustrated later in this thesis in figures 1.6, 1.9, and 6.2. Building such models is difficult and time-consuming. Building even the simple tensegrity icosahedron from figure 1.1a can be a challenge.

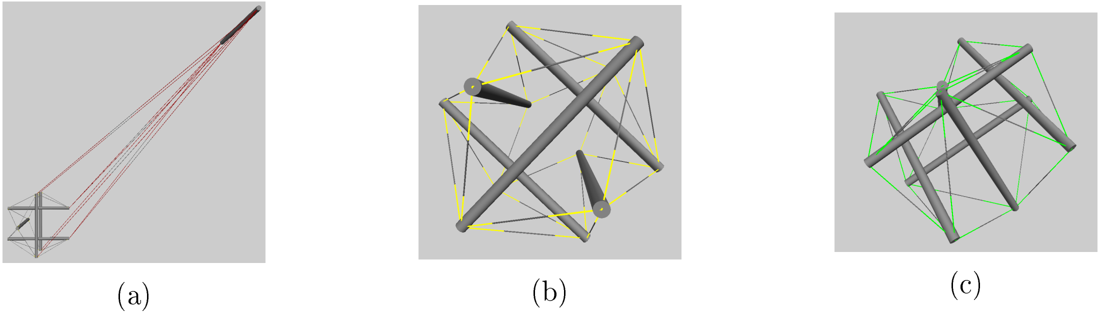
Computer simulation allows easy investigation of the properties of a large variety of complex tensegrity models. For example, researchers are interested in investigating the properties of fractal tensegrity structures [18]. Even a simple fractal tensegrity structure with only one self-similar submodel, such as the one in figure 1.2a, is challenging to build. In contrast, the simulation platform contributed by this thesis easily supports exploration of more complex fractal tensegrities such as figures 1.2b, 1.2c, and 1.2d. Our framework attaches smaller versions of the main icosahedron to the original system by inserting them into an elastic edge (as per the model constructed by Graham Scarr, see figure 1.2a). Scarr and others are investigating various fractal structures that may be useful in biotensegrity modeling; for example, a rod could be replaced by an icosahedral helix (Figure 21 in [18]).



(a) (b) (c) (d)

Figure 1.2: (a) A physical example of a fractal tensegrity icosahedron ©Graham Scarr (http://www.tensegrityinbiology.co.uk/), reproduced with permission. (b) Our simulated model of a fractal tensegrity icosahedron. (c) Our simulated model of a multiple iteration fractal tensegrity icosahedron. (d) Our simulated model of a single iteration fully fractal tensegrity icosahedron.

The simulation platform supports exploration of adaptive tensegrity systems. An **adaptive** tensegrity changes properties dynamically over time in response to tensional and compressive forces in the model. An example of the effects of adaptation is shown in Figure 1.3. Details as to the adaptation rules can be found in chapter 4. The stable states shown in figures 1.3b and 1.3c are produced by a potential-energy minimization algorithm (presented in section 3.2.2) run without, and respectively with, adaptive constraints. This type of simulation allows us to see how tensegrity systems with constituent adaptive objects would react to various loads and displacements. Exploring such behaviors is very difficult with physical models, because adaptive materials are expensive and difficult to work with. Simulation provides an inexpensive and convenient way to study materials with a variety of adaptivity criteria.



(a) (b) (c)

Figure 1.3: Illustration of adaptive tensegrity : (a) A starting state for a tensegrity icosahedron in which a single rod has been displaced far from the rest of the system. This system moves to a stable state in which potential energy of the system is minimized. We simulate this by applying the optimization algorithm described in section 3.1.3. (b) The end result when optimization occurs without adaptation: the system returns to the shape of the standard tensegrity icosahedron. (c) The end result when optimization includes application of the adaptation rules presented in chapter 4 : the system ends in a stable state that evidences a memory of displacement.

1.2 Definitions of Tensegrity

The definition this thesis uses when referring to tensegrity comes from Dr. Chris Barnes.

Definition (http://tensegritychiro.com/)**.** ***Tensegrity*** *is a concept in architecture and biology where a balance of tension and compression results in a maximum efficiency and economy.*

This definition is used because it is concise and captures the essence of this type of structure clearly and succinctly.

Definitions of tensegrity have ranged widely, depending on the uses each author had in mind for the system. For this reason several similar but distinct definitions currently exist. Fuller defines tensegrity as follows.

Definition. *[3]* ***Tensegrity*** *describes a structural-relationship principle in which structural shape is guaranteed by the finitely closed, comprehensively continuous, tensional behaviors of the system and not by the discontinuous and exclusively local compressional member behaviors.*

On his website, Kenneth Snelson, an artist who has worked extensively with tensegrity structures, defines it as follows.

Definition (kennethsnelson.net)**.** ***Tensegrity*** *describes a closed structural system composed of a set of three or more elongate compression struts within a network of tension tendons, the combined parts mutually supportive in such a way that the struts do not touch one another, but press outwardly against nodal points in the tension network to form a firm, triangulated, prestressed, tension and compression unit.*

Snelson adds the stipulation that tensegrity systems form triangulated units to exclude systems which fit the rest of the definition but are flaccid and do not have structural integrity.

1.3 History

First coined in Buckminster Fuller's 1961 paper Tensegrity [2], the term *tensegrity* is an amalgamation of the phrase *tensional integrity*. Although he did not coin the term until 1961, Fuller had been discussing the interplay of tensional and compressional forces in structures from a much earlier date. The starting point for what are now recognized to be tensegrity systems can be seen as early as Fuller's 1947 talk on energetic-synergetic geometry at the Black Mountain College.

While Fuller coined the term, he was by no means the only person to have worked on the concept. Several people worked and even submitted patent applications on similar designs around the same time. A more detailed overview of the history of tensegrity can be found in various publications ([7], [8], and [15]).

Figure 1.1 shows one of the most basic tensegrity structures, the tensegrity sphere, or tensegrity icosahdron. Although none of the rigid objects in this system are in contact, the system as a whole maintains a shape and has a certain structural soundness. This thesis focuses largely on models of the tensegrity sphere (with variations) as will be discussed in chapter 5.

The construction and symmetries of these and other classical tensegrity models are discussed in "A Practical Guide to Tensegrity Design" by Burkhardt. [1]

1.4 Tensegrity Structures in Real World Applications

Tensegrity is used in a wide variety of fields. In this section we highlight some examples of its applications.

1.4.1 Tensegrity in Architecture

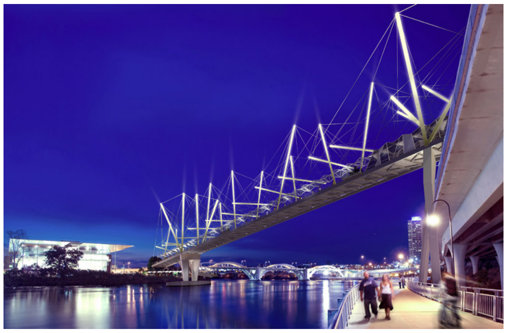


Figure 1.4: The Kurilpa Bridge

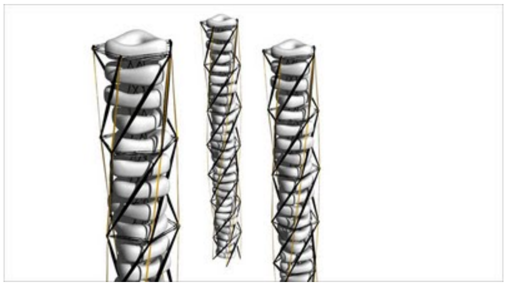


Figure 1.5: The proposed Filamentosa building project

The Kurilpa Bridge (seen in figure 1.4) is a $63 million pedestrian and bicycle bridge over the Brisbane River in Brisbane, Queensland, Australia [6]. This bridge holds the title of the world's largest hybrid tensegrity bridge at 470 meters in length. The bridge structure comprises 18 structural steel bridge decks, 20 structural steel masts and 16 horizontal masts. These elements make up the rigid objects in the tensegrity structure. The tension objects are the complex cabling system comprising 80 main galvanized helical strand cables and 252 superduplex stainless steel cables. The bridge is a hybrid tensegrity system due to the 72 precast concrete deck slabs which sit on the main bridge deck and are secured to the steel structure and each other by in-situ concrete stitch pours.

Another example for tensegrity inspired architecture is the proposed Filamentosa building project (Figure 1.5). Tristan D'estree Strek, one of the architectural minds behind this project, hopes to design and build skyscrapers whose tensegirty based skeletons could adapt to their environments. His work at ORAMBRA [20], the Office for Robotic Architectural Media and The Bureau for Responsive Architecture, aims to use the responsive nature of tensegrity in order to reduce energy uses in buildings. The tensegrity skeleton seen in Figure 1.5 would allow for far lighter, less resource intensive buildings, which are easier to construct and maintain.

More information on the uses of tensegrity in architecture and design can be found in [7].

1.4.2 Tensegrity in Art and Design

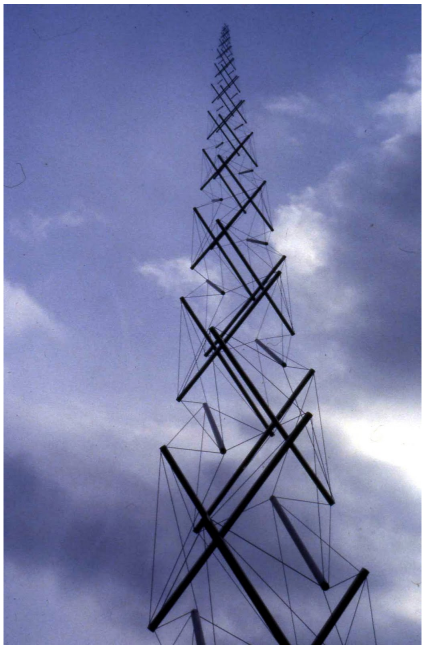


Figure 1.6: The Snelson Needle

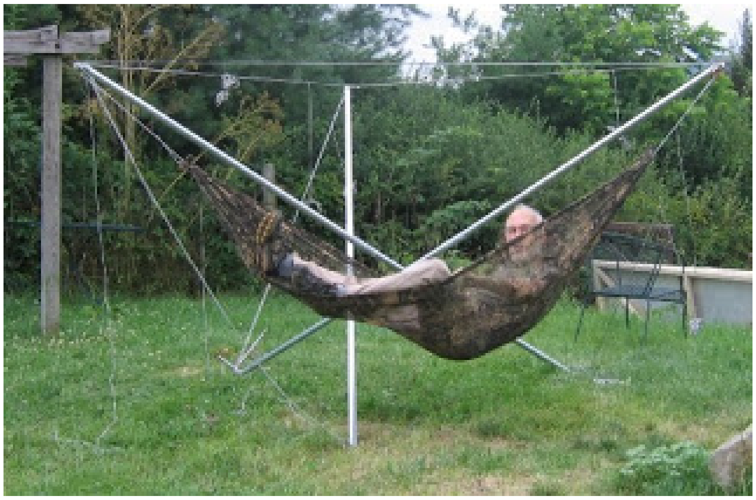


Figure 1.7: A 3-strut tensegrity hammock

Kenneth Snelson, an American sculptor and artist, has long been working with tensegrity structures. He was one of the first to build the systems that Fuller described in his 1947 talk, and has since gone on to make a large variety of tensegrity sculptures. Figure 1.6 is his famous Snelson's Needle (sometimes know as Snelson's Tower), an 18 meter high tensegrity tower connected to the ground in only three points. While swaying with its environment this tower manages to stay upright without having any of the rigid objects connected to each other.

Tensegrity structures have also been adopted to help us relax. The 3 strut tensegrity hammock shown in Figure 1.7 is a great example of everyday uses for tensegrity structures.

1.4.3 Tensegrity in Biological Modeling

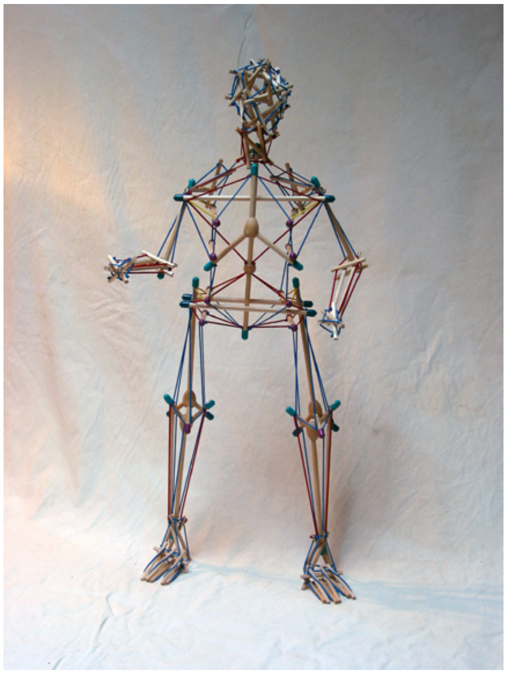


Figure 1.8: A physical tensegrity model of the human body.
©2006, Tom Flemons (http://www.intensiondesigns.com/), reproduced with permission

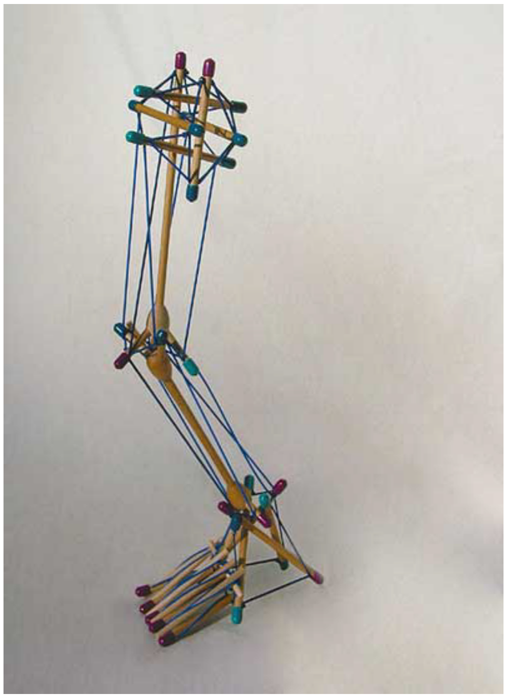


Figure 1.9: A physical tensegrity model of a foot.
©2006, Tom Flemons (http://www.intensiondesigns.com/), reproduced with permission

Tensegrity modeling is used in physiology. As is discussed in [13], [16], [17], and [19] the fascial network can be modeled as the tensional constituent of a tensegrity system within the human body. Bones and hydrostatic pressure are modeled as the compressive elements of this system. Tom Flemons and Graham Scarr have built interesting physical models of tensegrity forces in the human body (see figures 1.8, 1.9, and 6.1).

1.4.4 Tensegrity in Space Exploration

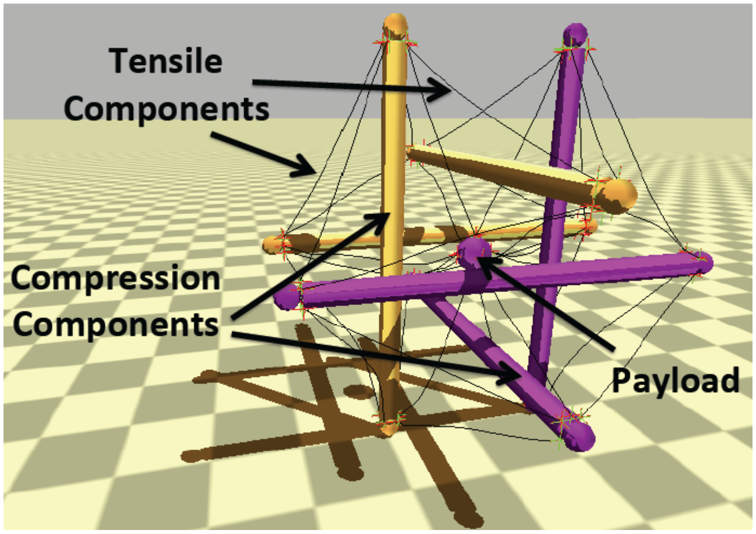


Figure 1.10: A Design for a tensegrity casing for a probe to Titan. ©Vytas SunSpiral [21], reproduced with permision

NASA has been experimenting with using an accentuated tensegrity icosahedron as the casing and transportation system for probes destined to explore the surface of Titan, one of Saturn's moons. Sunspiral et al. point out that, due to the relatively sparse density of these objects, tensegrity casings are cheaper and more easily shipped in space; moreover their structure and design allow them to absorb a great deal of impact force and then return to a stable state [21]. The payload is suspended in the middle of the icosahedron to protect it from impacts. The system they propose (see figure 1.10) has the icosahedron using its six rods as legs, powered by motors built into the elastics, allowing for a more robust and versatile transportation system than the traditional wheeled rover.

1.5 Contributions

This thesis lays the foundation for flexible tensegrity modeling software that can be used in a variety of applications. Though tensegrity modeling software has been used in other applications, at the time of this project there was no publicly available software for such modeling. Two months prior to the publication of this paper NASA released the NTRT, or NASA Tensegrity Robotics Tookit, a collection of C++ and MATLAB software modules for the modeling, simulation, and control of Tensegrity Robots, at http://ti.arc.nasa.gov/tech/asr/intelligent-robotics/tensegrity/ntrt. We are contributing code to NTRT in ongoing and future work. The following contributions are made:

1. A flexible approach simulating a wide variety of tensegrity systems. In particular, a flexibility in defining the types of objects that can be used as constituents of tensegrity systems (chapter 2), and a flexibility in simulation and optimization methods (section 3.1). An object definition includes:
   - A list of fixed parameters that define the attributes of the object.
   - A list of variable parameters that define the state space of the object.
   - A potential energy function that uses the fixed and variable parameters to determine the potential energy of the object in each state.
2. Design and implementation of a hill climbing algorithm to find minimal energy states of simulated tensegrity systems (Chapter 3). Execution of this hill climbing algorithm can optionally include application of adaptation rules as discussed in chapter 4.
3. Experimental results and validation for simulations of selected tensegrity systems. (Chapter 5)

In summary, the main contribution of this thesis is a flexible framework for exploring the properties of tensegrity structures. The exploratory properties of this framework have already yielded an interesting and unexpected insight, namely that the tensegrity icosahedron has an alternative stable state (figure 1.11). Figure 1.11a shows the stable state usually produced by the potential energy minimization algorithm of Section 3.2.2. An alternate stable state produced by the energy minimization algorithm (figure 1.11b) was validated by constructing it physically to demonstrate that it is stable (figure 1.11c). We constructed this model by starting with a physical model in the regular stable state shown in figure 1.1a; transforming this into the state shown in Figure 1.11c required three of the elastics to be unhooked and reattached, but otherwise there was no change to the system or its connectivity. This new stable state was an interesting and exciting discovery for us, and as far as we are aware has not been reported in the literature. The unexpected appearance of this new configuration in the simulation was first thought to be a bug in the algorithm. After no bug could be detected, the unexpected result was verified by constructing the physical model shown in 1.11c. This physical construction was shown to be a stable state that is resilient and capable of withstanding outside forces (figure 1.12).This initial distrust of the results of the algorithm highlights the need for validation of the algorithm. Test suites used for the purposes of validating the algorithm are described in chapter 5.

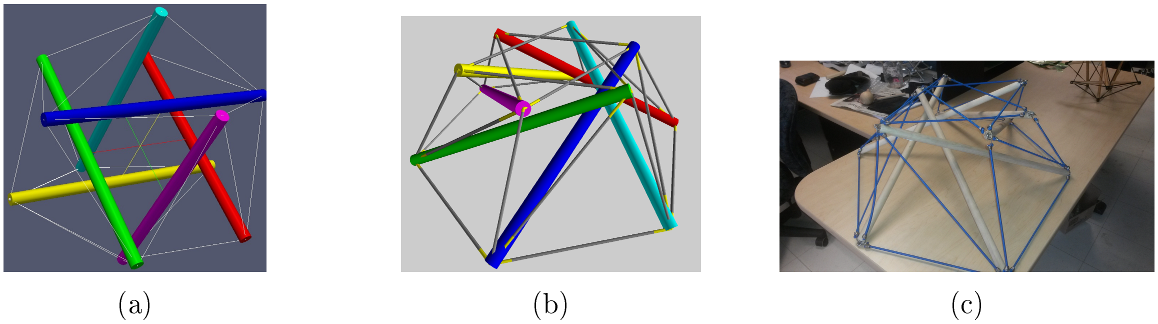

(a) (b) (c)

Figure 1.11: Discovery of an alternate stable state for the tensegrity icosahedron: (a) The expected, and usual, result of the optimization algorithm. (b) A simulated result showing the alternate stable state for the tensegrity icosahedron. (c) A physical model of the alternate stable state for the tensegrity icosahedron.

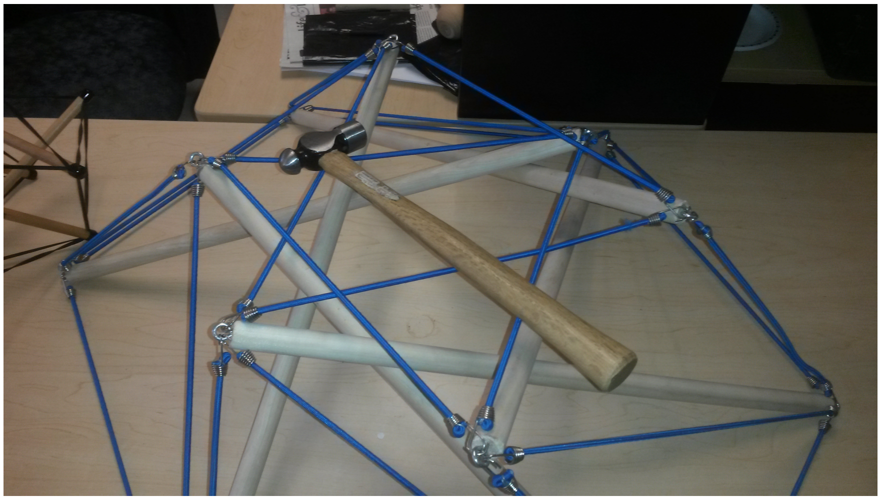


Figure 1.12: The alternate stable state with weight loading.

Chapter 2

Modeling Rod and Elastic Tensegrity Systems

Here we describe our approach to modeling tensegrity systems. Our focus is on modeling the effects of compressive and stretching forces within the tensegrity systems.

Definition. *An **object** is a part of a tensegrity system. **Compound objects** are those objects which are made by combining several other objects, while **atomic objects** are objects which cannot be broken into sub objects. Note that a tensegrity system as a whole is a compound object, though not all compound objects are tensegrity systems. Section 2.1 describes the components that define an object.*

This chapter culminates with a description of how to model a generic rod and elastic tensegrity system (section 2.7). To define this broad class of compound objects we first define some atomic objects that comprise it (elastics in section 2.3 and rods in section 2.4) and present examples of simpler compound objects (rubber band model in section 2.5 and the tensegrity icosahedron in section 2.6). For each object, in sections 2.2 to 2.7, we describe how to define the set of possible states, how to find the potential energy of each state, and how to find the state(s) that minimize potential energy.

2.1 Four Components that Define an Object

In this section we list the components required to define an object.

1. Fixed Parameters:

   These are the parameters of the object that remain constant in every state of the object.

2. State Vector and an Associated Interpretation:

   The state vector describes the possible states that the object can be in. The interpretation defines a mapping from the state vector to the state it represents.

3. Potential Energy Function:

   The potential energy function is denoted $E_P(V_O)$, where O is the object we are modeling and V_O is the current state vector of the object. This function uses the fixed parameters of the object and parameters in the state vector to determine the potential energy stored in the given state of the object.

4. Minimizing Potential Energy:

   This component defines how to find states which have minimal potential energy. In simple systems closed form solutions may be given (sections 2.2, 2.3, 2.4, 2.5). In more complex systems closed form solutions are difficult to derive (sections 2.6 and 2.7); in this case the state space search algorithms presented in chapter 3 can be applied.

2.1.1 Remarks on States and State Vectors

There is a clear distinction between the states of the object and the state vectors of the object. Each state vector can be interpreted to a state of the object but this mapping depends on the interpretation we define. Defining an appropriate interpretation and state vector for a system is challenging.

The ideal state vector provides a bijection between states and state vectors. This provides us with a canonical representation for the states of the object, and makes state space searching heuristics more accurate and efficient. Further this bijection helps us when doing mathematical proofs in regards to finding minimal energy states. An example of such a proof can be seen in section 2.5.4.

Definition. *An* ***isometry*** *is a distance-preserving map: the relative distances between points in the space are kept constant (i.e. rigid motion is permitted). Two states are* ***isomorphic*** *if there exists an isometry mapping one to the other.*

Since we are calculating potential energy in the absence of any external forces, and rigid motion does not change potential energy, any two isomorphic states have the same potential energy level. Because we are only interested in the potential energy of the systems we are studying, any two states which are isometries of each other will be treated as the same state. Ideally, we would like our interpretation of state vectors to map isomorphic states to the same state vector. This reduces the search space and further aids in proof structures. An example of such an ideal state vector is given in section 2.2.2.

2.2 Linear Springs

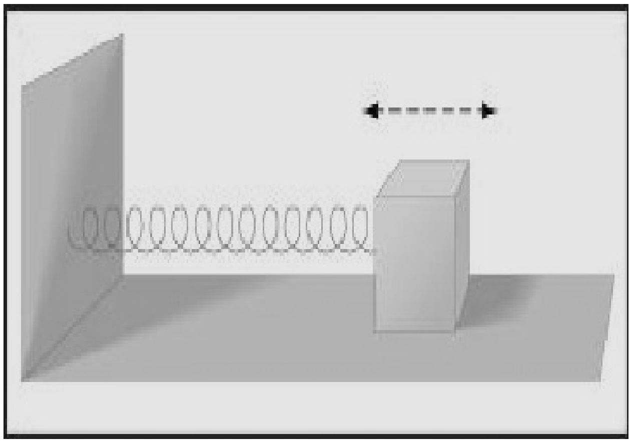


Figure 2.1: A spring.

In this section we preset the modeling of idealized linear springs. A linear spring can be stretched or compressed arbitrarily far and still act linearly as defined by Hooke's law:

$$F = -K(S) \cdot d(S) \tag{2.1}$$

In equation (2.1) $d(S)$ is the change in length of S, the amount by which the spring has been stretched or compressed. This value is negative when the spring is compressed and positive otherwise. This equation is used in Section 2.2.3 to define the potential energy of a state of the spring.

2.2.1 Fixed Parameters of a Spring

A spring S is defined by its spring constant $K(S)$. This value defines how resistant the spring is to being compressed or stretched. The length of the spring is not required as a fixed parameter because the linearity assumption implies that length does not have an effect on the potential energy of the spring.

2.2.2 State Vector of a Spring

The state of a spring S, is defined by the displacement $d(S)$. Thus the state vector is one-dimensional.

We now use springs to give a more concrete example of isomorphic states, as introduced in section 2 definition 2.1.1. The state vector of a spring contains one value: the stretch, or displacement, applied to the spring. Let S be a spring of length 5 in state A with endpoints $[0,0,0]$ and $[0,0,7]$. The state vector $V_A = [2]$ represents the spring S in state A. Let state A' be the rotation of the spring in state A by ninety degrees in the $x - z$ plane. The spring now has endpoints $[0,0,0]$ and $[7,0,0]$, and so the state vector $V_{A'} = [2] = V_A$. In this example the state vector representation chosen preserves the information relating to potential energy and maps isomorphic states to the same state vector.

2.2.3 Potential Energy of a Spring

The potential energy of a linear spring S, with spring constant $K(S)$, in a state defined by a displacement of $d(S)$ (as given in [12]) is

$$E_P(V_S) = \frac{1}{2} \cdot K(S) \cdot d(S)^2. \tag{2.2}$$

2.2.4 Minimizing Potential Energy of a Spring

By inspection, we can see that equation (2.2) is minimized when the absolute value of $d(S)$ is minimized. Hence, the state in which the spring has minimal potential energy is the one in which $d(S) = 0$.

2.3 Elastics

We define elastics as linear springs that do not push when compressed but do exert force when stretched.

Figure 2.2: An unstretched elastic.

2.3.1 Fixed Parameters of an Elastic

An elastic E is defined by its elasticity constant $K(E)$.

2.3.2 State Vector of an Elastic

The state of an elastic E is defined by the change in length (displacement) it has been subjected to. We denote this by $d(E)$.

2.3.3 Potential Energy of an Elastic

Since elastics are a type of specialized spring, we derive from equation (2.2) that the potential energy function of an elastic E is

$$E_P(V_E) = \frac{1}{2} \cdot K(E) \cdot \max(0, d(E))^2. \tag{2.3}$$

2.3.4 Minimizing Potential Energy of an Elastic

By inspection, to minimize the above equation we must make $d(E) \leq 0$.

2.4 Rods

We model a rod as an idealized solid structure which cannot be compressed or stretched.

2.4.1 Fixed Parameters of a Rod

A rod R is defined by its length, denoted $L(R)$.

2.4.2 State Vector of a Rod

Rods have no varying state, so the state vector of a rod is empty.

2.4.3 Potential Energy of a Rod

Since the rod cannot be compressed or stretched, it cannot store any potential energy. The potential energy of the rod is always zero.

2.4.4 Minimizing Potential Energy of a Rod

Because a rod does not store any energy there is nothing to minimize.

This concludes our discussion of three types of atomic objects: linear springs, elastics, and rods. Next we describe the modeling of compound objects: the rubber band model, the tensegrity icosahedron, and the rod and elastic models.

2.5 Rubber Band Model

Here we model a rubber band stretched around a set of fixed points forming a convex hull. Each segment of this convex hull is modeled as an elastic (as defined in section 2.3). The rubber band system is interesting because as we see in section 2.5.4 it has a closed form solution for finding minimal potential energy states, a contrast to the rod and elastic models defined in section 2.7.

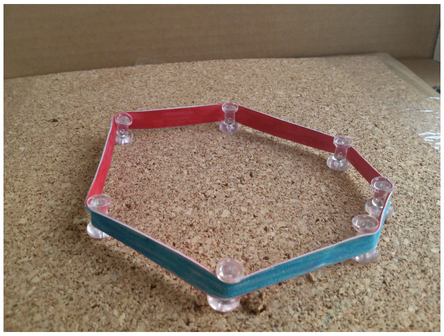


Figure 2.3: A state of the rubber band model.

2.5.1 Fixed Parameters of the Rubber Band Model

The rubber band B has $2n + 2$ fixed parameters:

- n $2D$ points P_0 through P_{n-1} around which the rubber band is stretched, stored as $2n$ values.
- a value $K(B)$ representing the elasticity constant of the band.
- a value $N(B)$ representing the natural length of the band.

Let the edge E_i denote the part of the rubber band which lies between the points P_i and $P_{i+1 \bmod n}$. These edges are modeled as separate elastics.

2.5.2 State Vector of the Rubber Band Model

The states of the model are the various ways in which the rubber band B can be stretched around the n points. Hence a state is defined by how much of the natural length of the band lies in each edge E_i. The natural length given to an edge E_i is denoted by $N(E_i)$.

A state is said to be **legal** if

$$\sum_{i=0}^{n-1} N(E_i) = N(B). \tag{2.4}$$

The following definition of the state vector of the rubber band model has been crafted such that:

- every state vector represents a legal state,
- there is no redundancy of represented states.

The state vector of the rubber band model is an $n-1$ dimensional vector V with real non-negative values.

To interpret this vector we define a new n dimensional vector V' as follows:

$$V'_i = \begin{cases} V_i & 0 \leq i \leq n-2 \\ 1 & i = n-1 \end{cases} \tag{2.5}$$

The natural length attributed to each edge E_i is:

$$N(E_i) = \frac{V'_i}{\Sigma(V')} \cdot N(B) \tag{2.6}$$

where $\Sigma(V')$ denote the sum of the entires of V'.

Proposition 1. *This construction of a state vector provides a bijection between state vectors and states of the rubber band system under the above interpretation.*

Proof. Let us start by proving the legality of the state that is interpreted from a state vector.

Let V be a given state vector and V' the augmented vector. Then:

$$\begin{aligned}\sum_{i=0}^{n-1} N(E_i) &= \sum_{i=0}^{n-1}\left(\frac{V'_i}{\Sigma(V')}\cdot N(B)\right)\\ &= \frac{N(B)}{\Sigma(V')}\cdot\sum_{i=0}^{n-1}(V'_i)\\ &= \frac{N(B)}{\Sigma(V')}\cdot\Sigma(V')\\ &= N(B).\end{aligned}$$

Hence the state given by this interpretation is indeed legal.

To show uniqueness, let us assume that there are two distinct state vectors V and W with augmented vectors V' and W' which lead to the same state. We will use this assumption to derive a contradiction.

Since $V \neq W$ we have from the definition of the augmented array that $V' \neq W'$. Hence for some $0 \leq i \leq n-1$ we have that $V'_i \neq W'_i$. However, by the assumption that V and W lead to the same state, we have

$$\frac{V'_i}{\Sigma(V')} = \frac{W'_i}{\Sigma(W')}. \tag{2.7}$$

Since $V'_i \neq W'_i$, equation (2.7) implies that $\Sigma(V') \neq \Sigma(W')$ (observation 1). By the assumption that V and W lead to the same state we have that

$$\frac{V'_{n-1}}{\Sigma(V')} = \frac{W'_{n-1}}{\Sigma(W')}.$$

By the construction of the augmented vector, $V'_{n-1} = W'_{n-1} = 1$. Thus

$$\frac{1}{\Sigma(V')} = \frac{1}{\Sigma(W')}.$$

Hence $\Sigma(V') = \Sigma(W')$, but this contradicts observation 1. Thus we conclude that our assumption was incorrect; no two distinct state vectors V and W can lead to the same state. □

Remark 1. Note that this construction has a useful linearity property. If $N(E_i) = c \cdot N(E_j)$ for some constant c, then this is reflected in the augmented array V' as $V'_i = c \cdot V'_j$.

2.5.3 Potential Energy of the Rubber Band Model

We model the sections of the rubber bands as individual elastics and treat the potential energy of the rubber band as the sum of the potential energies of the individual elastics.

Definition. *The **stretch** in an edge E_i is the difference between the length of E_i and $N(E_i)$ and is denoted by $d(E_i)$:*

$$d(E_i) = |P_i - P_{(i+1)\ mod\ n}| - N(E_i). \tag{2.8}$$

Using equation (2.3), the potential energy of the rubber band model is

$$E_P(B) = \frac{1}{2} \cdot K(B) \cdot \sum_{i=0}^{n-1} \max(0, d(E_i))^2 \tag{2.9}$$

2.5.4 Minimizing Potential Energy of the Rubber Band Model

Theorem 1. *If the potential energy of the elastic band B is not 0 then equation (2.9) is minimized when $d(E_i) = d(E_j)$ $\forall 0 \leq i, j \leq n-1$.*

Proof. Note first that if the stretch of any edge E_l is negative, the excess natural length attributed to E_l can be redistributed throughout the system, hence lowering the energy of the rest of the system without raising the potential energy of E_l. This assertion relies on the fact that the potential energy of the model is greater than 0 and as such there must exist an edge with positive stretch. We can now assume without loss of generality that $d(E_l) \geq 0 \ \forall 0 \leq l \leq n-1$.

We will now use proof by contradiction. Assume that there is a state which has minimal energy in which stretch is not evenly distributed. Then there exists a pair i, j with $0 \leq i, j \leq n-1$ such that $d(E_i) \neq d(E_j)$. The combined potential energy of these edges is then:

$$E_P = \frac{1}{2} \cdot K(B) \cdot (d(E_i)^2 + d(E_j)^2)$$

To minimize the above equation we need only minimize $d(E_i)^2 + d(E_j)^2$.

Without loss of generality, assume $d(E_i) < d(E_j)$ and define x such that $d(E_j) = d(E_i) + x$. Note that:

$$\begin{aligned}
d(E_i)^2 + d(E_j)^2 &= d(E_i)^2 + (d(E_i) + x)^2 \\
&= 2 \cdot d(E_i)^2 + 2 \cdot d(E_i) \cdot x + x^2 \\
&> 2 \cdot d(E_i)^2 + 2 \cdot d(E_i) \cdot x + \tfrac{x^2}{2} \\
&= 2 \cdot \left(d(E_i)^2 + d(E_i) \cdot x + \tfrac{x^2}{4}\right) \\
&= 2 \cdot \left(d(E_i) + \tfrac{x}{2}\right)^2
\end{aligned}$$

Therefore we have proven that evening out the stretch between edges E_i and E_j so that $d(E_i) = d(E_j)$ results in a lower combined potential energy. Since evening out the stretches of edges E_i and E_j does not change anything else in the system, this operation lowers the potential energy of the entire system. This shows there cannot be a state which has minimal potential energy in which stretch is not evenly distributed.

□

2.6 The Tensegrity Icosahedron Model

This model consists of 6 rigid rods R_0 through R_5, which are the compression objects in this system. Each rod R_i has two end points R_i^1 and R_i^2 defined in $\mathbb{R}^3$ space. Let the list of these points be denoted by $PointList$.

The model also contains 24 elastic objects which are the tension objects in this system. An elastic denoted $E_{P_1}^{P_2}$ connects elements P_1 and P_2 in the $PointList$. Let $EdgeSet$ be the set of these elastics.

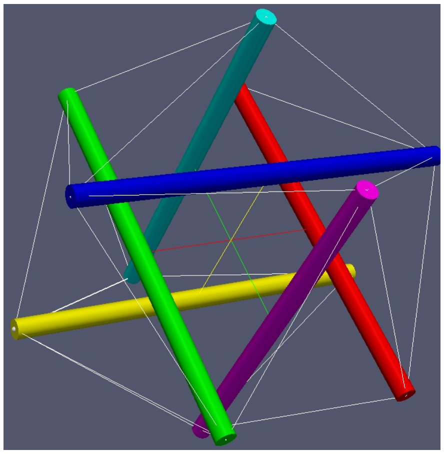


Figure 2.4: The tensegrity icosahedron shown in a state of minimal potential energy

The following notation defines the 24 elastic objects in the tensegrity icosahedron. Letting $0 \leq i \leq 5$, every elastic object $E_{P_1}^{P_2} \in EdgeSet$ belongs to one of the following four categories:

- $E_{R_i^2}^{R^2_{(i+1) \bmod 6}}$,
- $E_{R_i^1}^{R^1_{(i+2) \bmod 6}}$,
- $E_{R_i^1}^{R^2_{(i-2) \bmod 6}}$,
- and $E_{R_i^1}^{R^2_{(i-1) \bmod 6}}$.

This defines the connectivity of elastics seen in figure 2.4.

2.6.1 Fixed Parameters of the 6 Rod Tensegrity Model

Each rod R_i has a length denoted by $L(R_i)$. Further each edge $E_{P_1}^{P_2}$ has a natural length, denoted by $N(E_{P_1}^{P_2})$, and an elasticity value, denoted by $K(E_{P_1}^{P_2})$.

2.6.2 State Vector of the Tensegrity Icosahedron

The states of the tensegrity icosahedron are the various positions in which the rods could be held. The state vector use for this model is described in section 2.7.2.

2.6.3 Potential Energy of the Tensegrity Icosahedron

Since rods are defined to have no potential energy (section 2.4), the potential energy of the system is the sum of the potential energies of each elastic.

The stretch of an elastic $E_{P_1}^{P_2}$, as defined in section 2.3, is

$$d(E_{P_1}^{P_2}) = |P_1 - P_2| - N(E_{P_1}^{P_2}). \tag{2.10}$$

Using equation 2.3, from section 2.3, we have that the potential energy of the system is:

$$E_P = \sum_{E_{P_1}^{P_2} \in EdgeSet} \frac{1}{2} \cdot K(E_{P_1}^{P_2}) \cdot \max(0, d(E_{P_1}^{P_2}))^2 \tag{2.11}$$

2.6.4 Minimizing Potential Energy of the Tensegrity Icosahedron

The systems described up until now have had closed form solutions for minimizing potential energy. Such a solution is not known for the tensegrity icosahedron. Nor is one known for the more general rod and elastic model (section 2.7). Since the tensegrity icosahedron is an example of a rod and elastic model, we will use the same methods to find a state of minimum potential energy. This is discussed in section 2.7.4.

2.7 Rod and Elastic Models

Here we present an approach for modeling tensegrity systems that are made up of varying configurations of rods connected by elastics. Examples of such tensegrity systems include the tensegrity icosahedron discussed in section 2.6 and the system shown in figure 2.5.

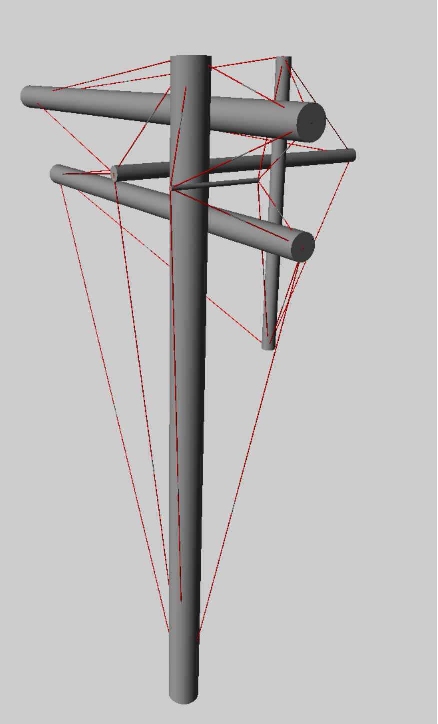


Figure 2.5: A Tensegrity model made entirely of rods and elastics.

2.7.1 Fixed Parameters of the Rod and Elastic Model

This system has the following fixed parameters:

- Number of rods in the system,
- Length of each rod,
- Number of edges in the system,
- Natural length of each edge,
- Elasticity of each edge,
- A connection set describing which edges are connected to which rods.

2.7.2 State Vector of the Rod and Elastic Model

The particular state vector of a rod and elastic system will vary from system to system, but it must represent the current positions of the rods in three space. Ideally these positions could be represented in such a way that there is a bijection between the states of the system and state vectors.

Here we present a representation that applies to any rod and elastic model, but does not have the bijection property mentioned above. The general state vector includes five parameters for each rod: $[x, y, z, \rho, \phi]$. The first three values are the x, y, and z co-ordinates of the center of the rod. The last two values represent the orientation of the rod as two rotations ρ and ϕ. These rotations are applied relative to the z-axis: ρ is the rotation in the $z - y$ plane and ϕ is the rotation in the $x - y$ plane.

The direction vector $D(R)$ of each rod R can be attained by rotating the z-axis, defined by $[0, 0, 1]$, appropriately. Specifically:

$$D(R) = \begin{bmatrix} cos(\phi) & -sin(\phi) & 0 \\ sin(\phi) & cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\rho) & -sin(\rho) \\ 0 & sin(\rho) & cos(\rho) \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{2.12}$$

Hence:

$$D(R) = \begin{bmatrix} sin(\phi) \cdot sin(\rho) \\ -sin(\rho) \cdot cos(\phi) \\ cos(\rho) \end{bmatrix} \tag{2.13}$$

Note that since $D(R)$ was derived by applying rotations to a vector of magnitude 1, $|D(R)| = 1$. Restricting $\rho \in [0, \pi]$ and $\phi \in [0, 2\pi]$ describes all possible orientations of the rod. We chose not to restrict the rotations; though this creates redundant state representations, it leads to a continuous potential energy function, which works better with the hill climbing algorithm described in section 3.2.2.

The state vector of such a model under the above interpretation has $m \cdot 5$ entries denoted V_0 to V_{5m-1}, where m is the number of rods in the system.

Given a system M the state represented by a state vector V using the above interpretation will be denoted as $\mu_M(V)$.

2.7.3 Potential Energy of the Rod and Elastic Model

The potential energy of the system is the sum of the potential energies of each elastic object. The special case of the tensegrity icosahedron was treated in section 2.6.3, this treatment generalizes here.

The stretch of an edge $E_{P_1}^{P_2}$, as defined in definition 2.5.3, is

$$d(E_{P_1}^{P_2}) = |P_1 - P_2| - N(E_{P_1}^{P_2}). \tag{2.14}$$

Letting *EdgeSet* be the list of all edges in the system and using equation 2.3 we have that the potential energy of a system M in state V is:

$$E_P(\mu_M(V)) = \sum_{E_{P_1}^{P_2} \in EdgeSet} \frac{1}{2} \cdot K(E_{P_1}^{P_2}) \cdot \max(0, d(E_{P_1}^{P_2}))^2 \tag{2.15}$$

The above equation is the potential energy as a function of the state of the system. We can also define potential energy as a function of the state vector of the system. In order to do so we number the two end points of a rod. We do this using the direction vector $D(R)$ defined in equation 2.13 and the center of the rod, denoted $C(R)$, defined as

$$C(R) = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{2.16}$$

We define the *first* endpoint of the rod R to be $C(R) - D(R) \cdot \dfrac{L(R)}{2}$ and the *second* to be $C(R) + D(R) \cdot \dfrac{L(R)}{2}$. We denote the j^{th} rod in the system as R_j, for $0 \leq j \leq m-1$.

Let $E_{R_i^j}^{R_k^l} \in EdgeSet$ be an elastic connecting the i^{th} point of the j^{th} rod to the k^{th} point of the l^{th} rod.

Below, V is a state vector as defined in section 2.7.2 and F is a function that computes the value of the n^{th} co-ordinate of the i^{th} point on the j^{th} rod. Note that $i \in \{1, 2\}$, and, for rods in $\mathbb{R}^3$, $n \in \{x, y, z\}$.

$$F(V,i,j,n) = \begin{cases} V_{5\cdot j} + (-1)^{i+1} \cdot \sin(V_{5\cdot j+4}) \cdot \sin(V_{5\cdot j+3}) \cdot \dfrac{L(R_j)}{2} & \text{if } n = x \\ V_{5\cdot j+1} - (-1)^{i+1} \cdot \cos(V_{5\cdot j+4}) \cdot \sin(V_{5\cdot j+3}) \cdot \dfrac{L(R_j)}{2} & \text{if } n = y \\ V_{5\cdot j+2} + (-1)^{i+1} \cdot \cos(V_{5\cdot j+3}) \cdot \dfrac{L(R_j)}{2} & \text{if } n = z \end{cases} \tag{2.17}$$

Using equation 2.17 we can now define the potential energy function of the state via its state vector.

$$E_P(V) = \sum_{E_{R_i^j}^{R_k^l} \in EdgeSet} \frac{1}{2} \cdot K\left(E_{R_i^j}^{R_k^l}\right) \cdot \max\left(0, \sqrt{\sum_{n \in \{x,y,z\}} (F(V,i,j,n) - F(V,k,l,n))^2} - N\left(E_{R_i^j}^{R_k^l}\right)\right)^2 \tag{2.18}$$

2.7.4 Minimizing Potential Energy of the Rod and Elastic Model

A closed form solution is not known for finding the minimal energy states of a rod and elastic model. Instead, a state space searching algorithm can be used. Various algorithms for minimizing the potential energy of rod and elastic models are presented in the next chapter.

Chapter 3

Potential Energy Minimization Algorithms

This chapter contributes a state space search algorithm to find the stable state of a tensegrity system. To define a state space search algorithm we use a sampling algorithm (section 3.2) to define our search space, and a minimization algorithm (section 3.3) to define our path through the search space. Often this process entails conducting state space searches of the system's constituent atomic objects.

Definition. *The **state space** $\mathbb{S}(O)$ of an object or system O is the set of all possible physical positions O can be in. This space is sampled by our sampling algorithm to create a discretized sample state space.*

Each system O has a potential energy function that maps each point in $\mathbb{S}(O)$ to the potential energy of the system in that configuration.

3.1 Flexible Software for Scientific investigation

The main purpose of the tensegrity modeling framework described in this thesis is for scientific investigation. To this end, the implementation of the framework must allow for great variability of use and ease of customization and expansion. As discussed in [9] it is important to realize that with scientific software, the software is not the end product. The end product is the research that can be done with the software.

The following features of our tensegrity modeling framework achieve the ease of customization that is required of scientific exploratory software:

- The hierarchical object definition in chapter 2 allows for easy integration of other objects into the system.
- State files contain all information needed to generate objects. This allows for easy and convenient generation of a variety of models without need to modify the code. An example of a state file is given in appendix E.
- There is a clear separation between object definition, object sampling, and optimization algorithm. The hill climbing optimization algorithm presented in section 3.3 requires a sampling algorithm. The clear separation between these algorithms allows the user to easily replace either the sampling algorithm or optimization algorithm.

3.2 Defining the Discretized Search Space

As per definition 3 in section 3, the state space $\mathbb{S}(O)$ of any system O is infinite and continuous: physical positions are represented by real values. In order to search this space we need to sample it to discretize the space. We construct the discretization by defining a finite set of positions that can be reached from a given position in a single simulation step.

Definition. *The* ***single step space*** *of an object is a set of positions that, in an appropriate metric, are close to the original position.*

Each sampling algorithm takes a single state as input and produces a single step state space as output. This set of sampled states is comprised of states which, given an appropriate metric, are within a given radius of the initial state. For our purposes we want a sampling algorithm to have the following properties.

- Return a relatively small number of samples.

  This property is important as the computation time of our hill climbing algorithm (section 3.3) grows linearly with the size of the sample set.

- Sample evenly about the given state.

  Our hill climbing method is most likely to yield optimal results if we sample evenly in every direction. When more sample points are included in the single step state space, more directions in the sample space are checked by the hill climbing algorithm; this increases the likelihood that the algorithm finds the correct direction.

- Provide the ability to control the radius of the sample space.

  The hill climbing algorithm described in section 3.3 progressively decreases the radius of the single step state space. This allows the algorithm to begin with large steps towards an optimum state, and later refine this result with smaller steps.

In order to define an effective and efficient state space search of a tensegrity system, we define two sampling algorithms that enumerate the ways in which the rods in the system can move. Section 3.2.1 defines a sampling algorithm which moves rods by fixing one endpoint and allowing the other to shift freely, while section 3.2.2 defines a sampling algorithm which allows both endpoints to move simultaneously.

3.2.1 Endpoint Sampling Algorithm

We define the single step state space of the system by moving one endpoint of a rod at a time. This movement defines a sphere on which the endpoint can move; this motion does not affect the spatial positioning, or, by extension, the potential energy, of the rest of the system. More specifically if we let P be the end point of a rod R, then the sample space of P is a sphere centered at the other endpoint of the rod R with radius $L(R)$, as illustrated in figure 3.1. The local potential energy at this point is defined by the tension on the elastics connected to P.

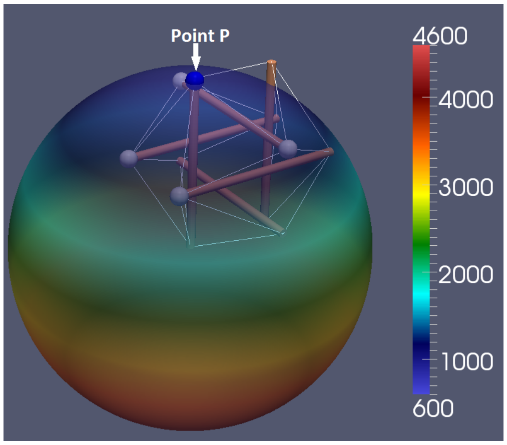


Figure 3.1: The sample space of an endpoint of a rod. The blue point (point P) is being moved and the four white points are connected to the blue point via elastics. The sphere represents the sample space and is colour coded to show the local potential energy, derived from the tension in the elastics connecting P to the four white points.

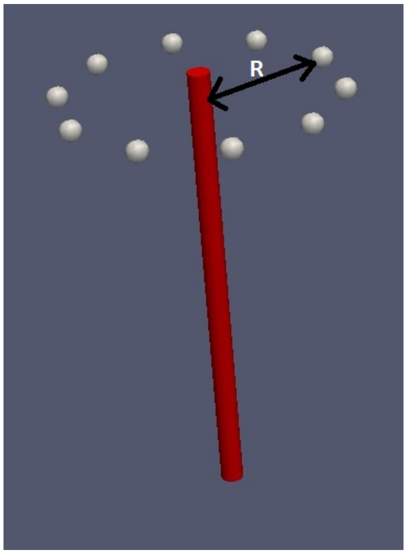


Figure 3.2: The discretized single step sample space of an endpoint.

This sampling algorithm has two parameters, the degree with which the rod is tilted (α), and the number of points to sample (n). The degree with which the rod is tilted defines how close the states in the single step sample space are to the original state. This is our single step space radius.

Below are the steps of the endpoint sampling algorithm, as it is run on a rod R with endpoint P and parameters α and n:

Step 1 Pick an arbitrary direction $\vec{d_0}$ perpendicular to the rod R

Step 2 Compute the location of P', which is the new location of the endpoint P if the rod is rotated by α in the direction $\vec{d_0}$ (See figure 3.3). P' is returned as the first sample point.

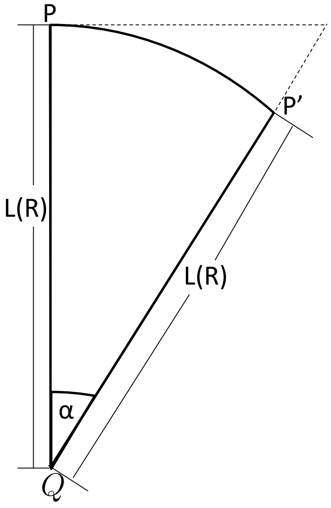


Figure 3.3: Rotating the rod by α.

The position of the first sample point P' is given by the following formula:

$$P' = \frac{P + \frac{\vec{d_0}}{|\vec{d_0}|} \cdot L(R) \cdot \tan(\alpha) - Q}{|P + \frac{\vec{d_0}}{|\vec{d_0}|} \cdot L(R) \cdot \tan(\alpha) - Q|} * L(R) + Q \quad (3.1)$$

In the above equation Q is the opposite endpoint to P on the rod.

Step 3 Generate the remaining $n-1$ sample points by rotating the point P' by an angle of $\frac{2\pi}{n}$ about the rod. Repeat this $n-1$ times. (See figure 3.2)

Experimentally, this sampling method was found to yield good hill climbing (section 3.3) results on the tensegrity icosahedron, but the hill climbing algorithm failed to find globally minimal potential energy states in more general tensegrity systems. For example figure 3.4 shows the result of this algorithm applied to a fractal tensegrity icosahedron.

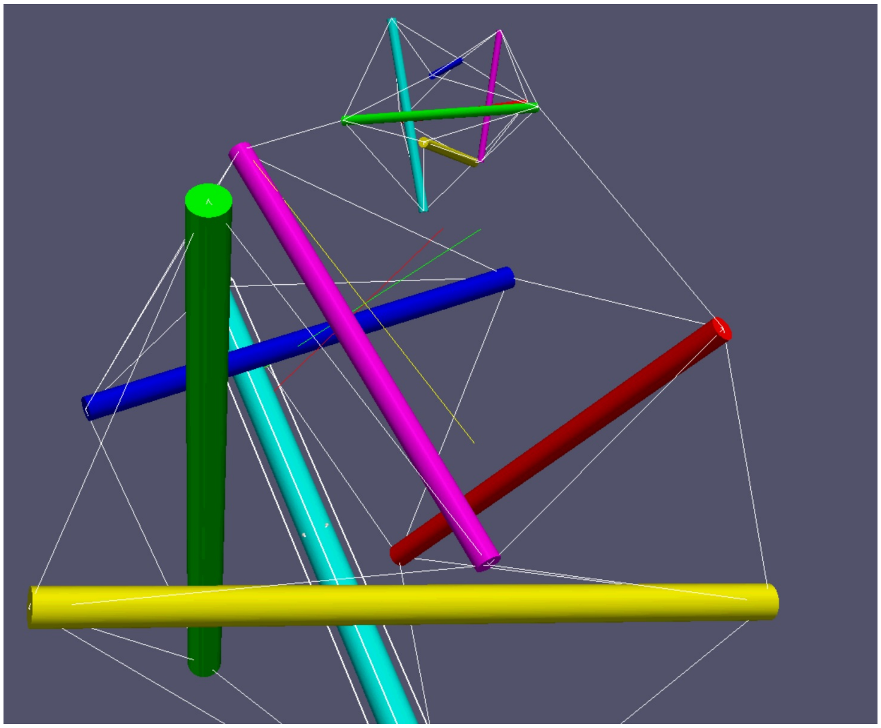


Figure 3.4: The output of hill climbing using the endpoint sampling algorithm on a fractal system. The small green rod above could be moved downward to reduce potential energy, but the endpoint sampling algorithm does not include that as an option.

This failure of the hill climbing algorithm to find a globally optimal state can be seen as stemming from the fact that the sample space generated by moving one rod endpoint at a time does not allow for a rod to move sideways or along its axis. The sampling approach presented in section 3.2.2 fixes this sampling limitation and allows the hill climbing algorithm to find globally optimal solutions for the system in figure 3.4 as well as for most of the systems it was tested on. Details on the testing of these sampling algorithms is presented in chapter 5.

3.2.2 Rod Sampling Algorithm

In order to create a more thorough sampling of the state space we now define a sampling algorithm that considers the ways each rod can move, as opposed to the ways a rod endpoint can move.

The rod sampling algorithm has the following parameters:

- A value n defining coarseness of the sampling. This affects the number of directions of rod displacement and rotation. Specifically the number of samples generated is $(n+1) \cdot (n+3) + 2n$.
- An angle α defining the angle of rod rotation.
- A value t defining the distance of rod translation.

Below are the steps of the rod sampling algorithm, illustrated in figure 3.5:

Step 1 Pick an arbitrary direction perpendicular to the rod. Let this be $\vec{d_0}$.

Step 2 For $i \in [1..n-1]$ rotate $\vec{d_0}$ by $\dfrac{2\pi \cdot i}{n}$, and define $\vec{d_i}$ to be the resultant vector. This step is equivalent to Step 3 in section 3.2.1.

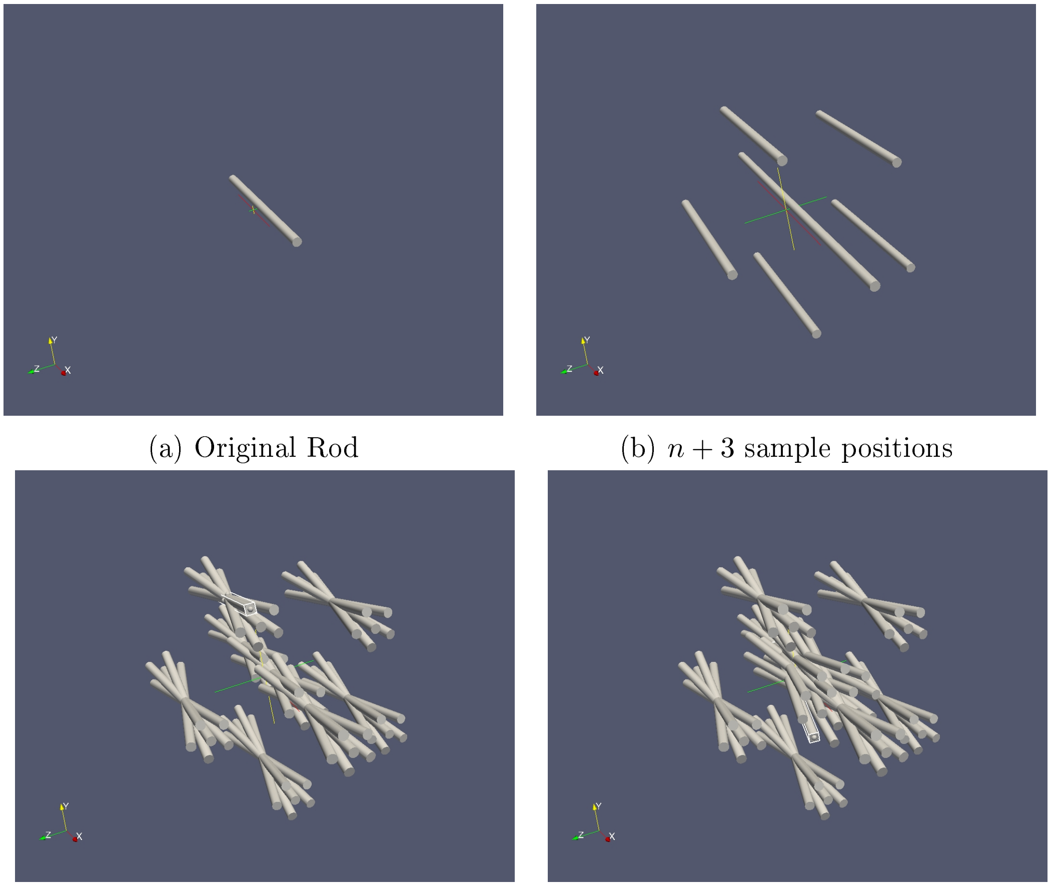
Step 3 Translate the rod by t in each of the following directions (figure 3.5b):

- Positively along the axis of the rod.
- Negatively along the axis of the rod.
- In each of the $\vec{d_i}$ directions.

Step 4 For each $\vec{d_i}$ apply a rotation perpendicular to $\vec{d_i}$ of α degrees to each rod generated in Step 3, with the midpoint of the rod as the center of the rotation (figure 3.5c).

Step 5 For each endpoint of the original rod, rotate the rod by α degrees in the direction of each $\vec{d_i}$ (figure 3.5d).

Remark 2. Note that Step 5 generates all the positions that the endpoint sampling algorithm generates for this rod. Hence the rod sampling algorithm provides a strict superset of the positions provided by the endpoint sampling algorithm.



(a) Original Rod

(b) $n + 3$ sample positions

(c) $(n + 1) \cdot (n + 3)$ sample positions

(d) $(n+1) \cdot (n+3) + 2n$ sample positions

Figure 3.5: (a) A single rod to which the rod sampling algorithm is to be applied. The coarseness parameter n is set to 5 in this example. (b) The first step of the sampling procedure is to create $n+2$ new rod positions by translating the original rod by t along the axis of the rod and in n evenly spaced perpendicular directions (Step 3). (c) The $n+3$ rod positions are then rotated by α about their *midpoints* in each of the n directions resulting in $(n+1) \cdot (n+3)$ sample positions (Step 4). (d) Lastly the original rod is rotated by α about each of its *endpoints* in each of the n directions, adding another $2n$ sample positions (Step 5).

The rod sampling algorithm allows the hill climbing algorithm to find minimal energy states in a vast majority (99%) of test cases (chapter 5). The major concern of this sampling algorithm is that it produces far more samples than the endpoint sampling algorithm, meaning that the hill climbing algorithm takes significantly more execution time. This will be discussed in Chapter 5.

3.3 Hill Climbing Algorithm

In order to find a minimal potential energy state of a system, we apply a hill climbing algorithm to the state space defined by the sampling algorithms given in section 3.2. The hill climbing algorithm presented in this section is not guaranteed to reach a global minima in all cases, but the validation shown in chapter 5 show that it reaches global minima in a large majority of cases. Hill climbing methods are more generally discussed in section 2.6 of [14].

The hill climbing algorithm has the following parameters:

- A set of parameters dictated by the sampling algorithm. This includes variables defining the radius of the single step sample space, and a scaling factor μ for reducing this radius.
- An accuracy value σ for estimation purposes. This value determines how close two numbers must be to be considered equal (i.e. $x = y \iff |x - y| < \sigma$).
- A starting state for the system.

The hill climbing algorithm runs as follows:

Step 1 Set the base state to be the starting state.

Step 2 Use the sampling algorithm to generate a single step state space $\mathscr{S}$ for the base state.

Step 3 If $\mathscr{S}$ is empty terminate the process.

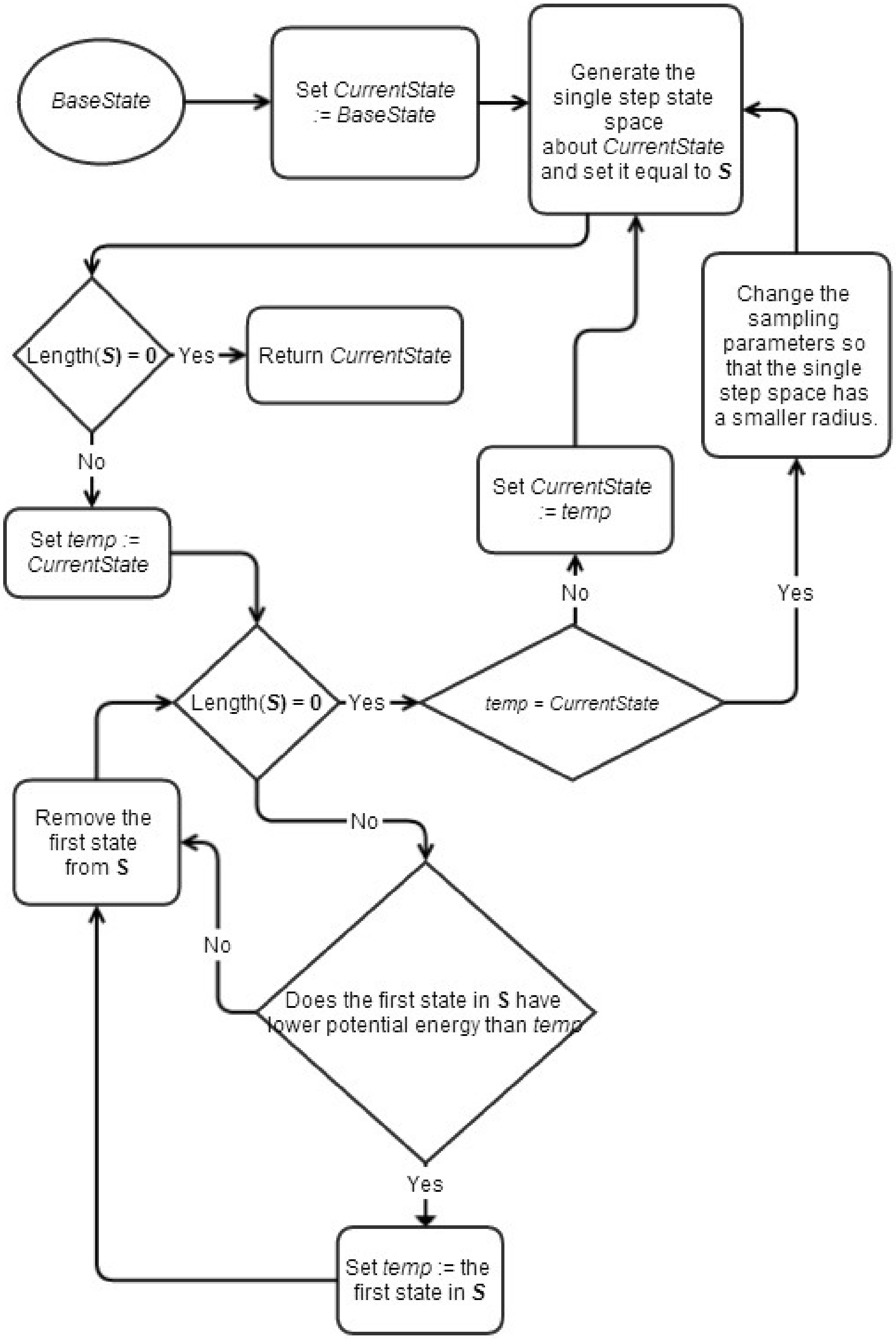
Step 4 For each state $P \in \mathscr{S}$

- If the potential energy at P is lower than the potential energy of the base state by at least the accuracy parameter σ, set the new base state equal to P.

Step 5 If the base state has not changed in this iteration, reduce the radius of the single step space by a factor of μ and restart the algorithm. In the next iteration the tested states are closer to the base state. Note that if the sampling radius is smaller than σ it is considered to be equal to 0; in this case the sampling algorithm returns an empty set, so the hill climbing algorithm terminates in Step 3.

It is important to note that this algorithm does not check for collisions at any point, as such it allows for edges and rods to pass through each other. This feature allowed the alternate state presented in section 1.5 to be found, as this configuration requires 3 elastics to pass through each other. It is also important in the large scale validation suite presented in section 5.2.

The following flowchart provides a summary of the hill climbing algorithm.


BaseState
Set CurrentState := BaseState
Generate the single step state space about CurrentState and set it equal to S
Length(S) = 0
Yes
Return CurrentState
No
Set temp := CurrentState
Length(S) = 0
Yes
temp = CurrentState
No
Set CurrentState := temp
Yes
Change the sampling parameters so that the single step space has a smaller radius.
No
Remove the first state from S
No
Does the first state in S have lower potential energy than temp
Yes
Set temp := the first state in S

Bellow we show that the hill climbing algorithm given here terminates. A general discussion about termination of hill climbing algorithms is given by Johnson and Jacobson [5].

Theorem 2. *The space we are sampling has the requisite properties for the hill climbing algorithm to terminate.*

Proof. Note first that if the algorithm fails to lower the potential energy of the system then a smaller radius is used for the sampling algorithm (Step 5). When this radius approaches zero, within the error margin defined by the approximation variables, the sampling set is empty and the algorithm terminates (Step 3). Therefore if the algorithm fails to lower the potential energy of the system a sufficient number of times it will terminate.

It remains to be proved only that the algorithm cannot continue infinitely reducing the potential energy.

Let M be an arbitrary starting configuration of a tensegrity system. This starting configuration has a finite potential energy. Let $E_P(M) = c$. Each step in the algorithm that reduces potential energy does so by at least σ; the user provides this accuracy value as input to the algorithm. Since the user must request finite accuracy, σ must be greater than zero. Therefore the maximum number of energy decreasing steps in the algorithm is $\frac{c}{\sigma}$. Since this is a finite number, the algorithm cannot continue infinitely reducing the potential energy and so it must terminate. □

3.3.1 Complexity Analysis

Bellow we analyze the hill climbing algorithm presented in section 3.3, and ascertain the complexity of the algorithm.

The first step of the algorithm is to set a pointer and as such takes $O(1)$ time. The second step is to create a sampling of the base state. The complexity of this step depends on the sampling algorithm, using the rod sampling algorithm we generate $(n+1)\cdot(n+3)+2n$, or $O(n^2)$, sample positions for each rod, where n is the resolution of the sampling algorithm. Therefore step 2 of the algorithm takes $O(m \cdot n^2)$ time, where m is the number of rods in the system. Step 3 checks if the sample set created in step 2 is empty, and therefore has complexity $O(1)$. Step 4 checks the potential energy in each sample state created in step 2. It does this by summing the potential energy of each edge in the system in each sampled configuration. Letting the number of edges in the system be denoted e we have that step 4 runs in $O(m \cdot n^2 \cdot e)$ time. Lastly step 5 makes one comparison and a multiplication operation, and is therefore of complexity $O(1)$. Overall each iteration of steps 1-5 takes $O(m \cdot n^2 \cdot e)$.

Each iteration of steps 1-5 either reduces the potential energy of the system, or reduces the radius of the sample space. Therefore the hill climbing algorithm runs steps 1-5 at most $\frac{c}{\sigma} + \frac{r}{\mu}$ times, where c is the starting potential energy of the system, σ is the minimal amount by which each step reduces the potential energy of the system, r is the starting radius of the sample space, and μ is the amount by which the radius is reduced each time the sampling radius is reduced. the final complexity of the algorithm is

$$O\left(\left(\frac{c}{\sigma} + \frac{r}{\mu}\right) \cdot m \cdot n^2 \cdot e\right). \tag{3.2}$$

Chapter 4

Simulating Adaptation within Tensegrity Systems

Up to this point in this thesis we have described tensegrity systems in which the rigidity of rigid objects and the elasticity of elastic objects has been fixed. In order to be able to explore a larger and more varied space of systems, we now define a means to create systems where these properties change as the system moves through its state space (section 4.1). Our implementation is described in section 4.2, and examples of are given in section 4.3.

Definition. *The **potential energy manifold** $\mathbb{M}(O)$ of a system O is a weighting of its state space $\mathbb{S}(O)$ by the potential energy of the system at each point.*

An example of part of a potential energy manifold can be seen in figure 4.1. In this figure color coding is used to represent the potential energy on a surface in three space. Because potential energy manifolds are generically space with more than three dimensions, any visualizations of them will need to be projections into a drawable space.

Definition. *A **fixed** system is a system in which the potential energy manifold cannot change during the simulation. Conversely, an **adaptive** system is one in which the potential energy manifold can change.*

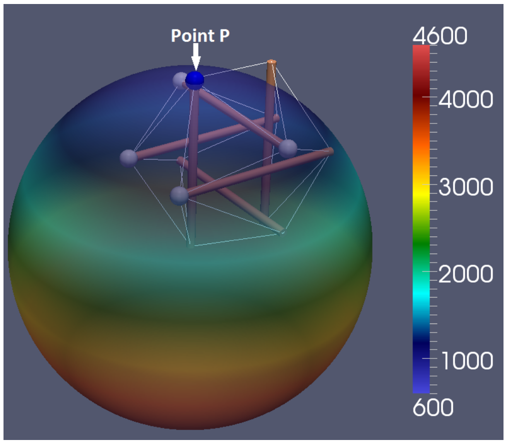


Figure 4.1: The sample space of an endpoint of a rod. The blue point (point P) is being moved and the four white points are connected to the blue point via elastics. The sphere represents the sample space and is colour coded to show the local potential energy, derived from the tension in the elastics connecting P to the four white points.

4.1 Adaptation Rules for Elastics

In this section we define the adaptation rules that allow the elasticity and lengths of elastics to change. These rules are based on work done by fellow M.Sc. candidate Vyacheslav I. Jdanov [4]. The set of the adaptation rules can be changed as required by the application.

For each elastic E we define an **ideal displacement** $I(E)$:

$$I(E) = N(E) \cdot K(E), \tag{4.1}$$

where $N(E)$ and $K(E)$ are the natural length and elasticity of the elastic E respectively. We then define the **ideal displacement difference** $ID(E)$ value as:

$$ID(E) = \max\left(-1, \min\left(1, \frac{L(E) - I(E)}{I(E)}\right)\right) \tag{4.2}$$

where $L(E)$ is the length that elastic E has in the current state of the object. The $ID(E)$ value is bounded in the range $[-1, 1]$, so that the system does not adapt too drastically.

Informal experimentation showed that allowing $ID(E)$ to be larger than $|1|$ could lead lead to oscillation in the elasticity of the elastics. Bounding $ID(E)$ in ranges greater than $[-10, 10]$ led the systems to oscillate between overstretched and under-stretched at each iteration. In the extreme case, if $ID(E)$ was unbounded, the algorithm would adapt the elastics to be stable in whatever configuration the system was currently in.

At each adaptation cycle the elasticity of E is set to

$$max(0, K(E) + ID(E) \cdot K(E) \cdot \epsilon) \tag{4.3}$$

where ϵ is an external variable that dictates the extent of the adaption. The new elasticity is bounded below by zero as negative elasticities are not allowed.

Similarly the natural length of E is set to:

$$max(0, N(E) + ID(E) \cdot N(E) \cdot \epsilon) \tag{4.4}$$

with ϵ as above. The new natural length is similarly bounded below by zero.

4.2 Adaptive Hill Climbing Algorithm

One of the advantages of the flexible implementation described in section 3.1 is that it makes the integration of adaptation rules a simple matter. As described in chapter 3, our optimization procedure is broken up into a hill climbing algorithm and a sampling algorithm. We add an adaptation cycle after each optimization step in the hill climbing algorithm - Step 4 in the algorithm provided in Section 3.3.

The adaptive hill climbing algorithm is:

Step 1 Set the base state to be the starting state.

Step 2 Use the sampling algorithm to generate a single step state space $\mathscr{S}$ for the base state.

Step 3 If $\mathscr{S}$ is empty terminate the process.

Step 4 For each state $P \in \mathscr{S}$

- If the potential energy at P is lower than the potential energy of the base state by at least the accuracy parameter σ, set the new base state equal to P.

Step 4a For each elastic E, apply the adaption rules desired by the user. This thesis uses the rules defined in section 4.1.

Step 5 If the base state has not changed in this iteration, reduce the radius of the single step space by a factor of μ and restart the algorithm. In the next iteration the tested states are closer to the base state. Note that if the sampling radius is smaller than σ it is considered to be equal to 0; in this case the sampling algorithm returns an empty set, so the hill climbing algorithm terminates in Step 3.

It is important to note that the way we integrate adaptation rules into our system affects the end result of the adaptation. In particular, the adaptive hill climbing algorithm defined above has the property that the size of the steps in the hill climbing algorithm will determine the extent of adaptation that will occur before a stable state is found. An example of this will be given in section 4.3.

In section 3.3 the proof of theorem 2, which states that the hill climbing algorithm will terminate, relied on the fact that at each hill climbing step the potential energy in the system is less than at the previous step. With the addition of adaptive rules (Step 4a) this is no longer true, and thus the proof of theorem 2 requires modification. We leave this modification as future work. Experimental results have, however, shown termination in all cases.

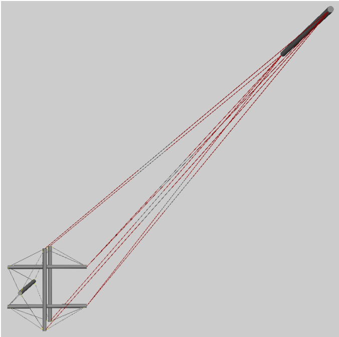


Figure 4.2: A starting state for a tensegirty icosahedron in which a single rod has been pulled far from the rest of the system

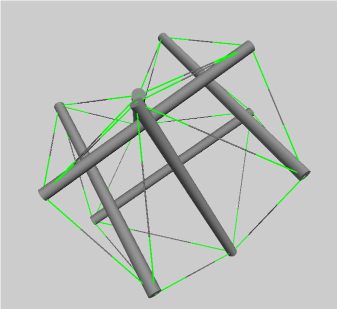


Figure 4.3: The stable state resulting from the application of the adaptive hill climbing algorithm (section 4.2) on the system in figure 4.2

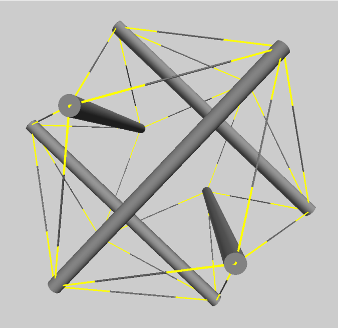


Figure 4.4: The stable state resulting from the application of the non-adaptive hill climbing algorithm (section 3.3) on the system in figure 4.2

4.3 Examples of Adaptive Results Using Our Algorithm

Our implementation color codes highly overstretched elastics as red, moderately overstretched as yellow, and ideally stretched as green. Figure 4.2 shows a tensegrity icosahedron in an unstable state. Here we can see that all the edges are overstretched, but after the system is optimized with adaptation (figure 4.3) we see that the edges are green, indicating that the system is now ideally stretched. This means that for every elastic E in the system, $I(E) = N(E) \cdot K(E) = L(E)$ as described in equation 4.1. However, without adaptation, the resultant state after optimization (figure 4.4) still has the edges as yellow, this coloring is due to the fact that this state, while stable is not ideally stretched.

Figure 4.5 shows the effect that changing the step size in hill climbing has on the final stable state of the system. Figure 4.5b shows the final stable state when the hill climbing algorithm is run with a single step sample radius ten times smaller than that used to obtain the stable state shown in figure 4.5c. The stable state in figure 4.5b has a lower potential energy (by 9%) and is more symmetric than the stable state shown in figure 4.5c. These differences result from the difference in size of the single step space radius. Specifically if the single step state space has a large radius as in figure 4.5c, then there are fewer adaptation cycles (Step 4a of the adaptive hill climbing algorithm presented in section 4.2) before the system reaches a near equilibrium state. This means that the elastics have less time to adapt to their initial configuration. Conversely, if the single step sample space has a small radius as in figure 4.5b, the hill climbing algorithm will make many smaller steps towards equilibrium allowing for more adaptation cycles while the system is still close to its starting configuration.

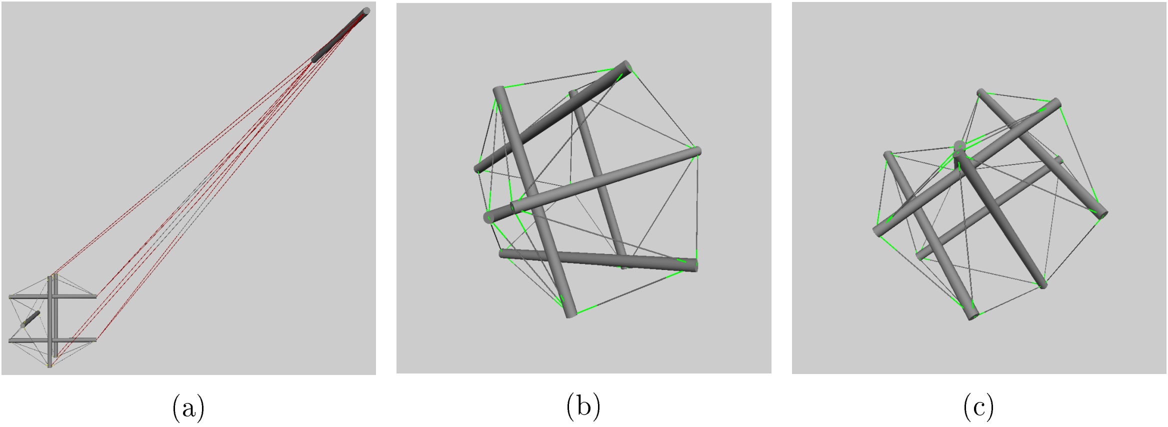


Figure 4.5: (a) The starting configuration of a tensegrity icosahedron in an unstable state. (b) The adaptive hill climbing algorithm applied to the initial state (a) with a small single step space radius. (c) The adaptive hill climbing algorithm applied to the initial state (a) with a single step space radius ten times larger than the one used in (b).

Chapter 5

Algorithm Validation

In this chapter we discuss the tests used to validate our implementation of the potential energy minimization algorithm proposed in chapter 3. These tests involved generating a base state of a tensegrity system, and then running our hill climbing algorithm (section 3.3) while using one of the sampling algorithms provided in section 3.2. A small scale test suite (section 5.1) demonstrates that the rod sampling algorithm (section 3.2.2) is more robust than the endpoint sampling algorithm (section 3.2.1). The large scale test suite in section 5.2 provides extensive validation of the rod sampling algorithm.

Our implementation was written in $Python2.7$. The hill climbing algorithm and sampling algorithms use only the standard *math*, *sys*, *copy*, and *import* python libraries. A graphical interface using $VPython$ and $wxPython$ was written to display the results of these optimizations. This display component does not affect the results of our optimizations as it is separate from the optimization package and only displays the results after the fact.

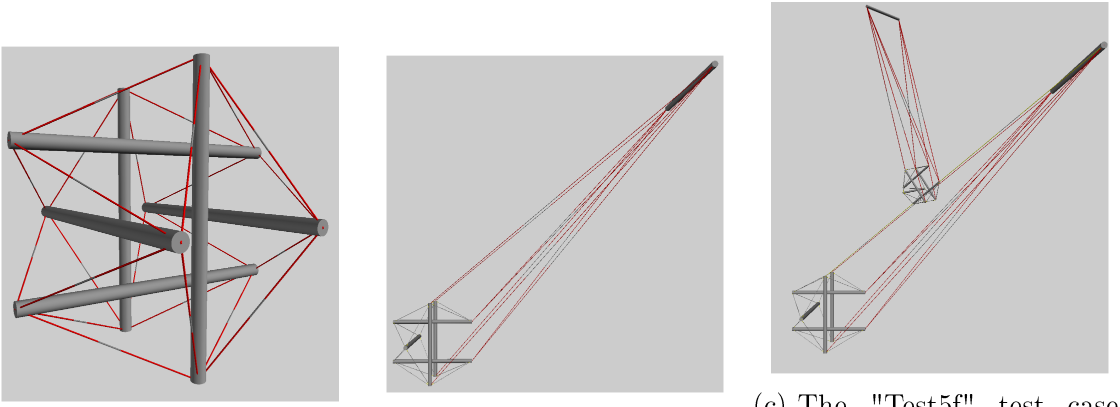
5.1 Small Scale Tests

The end point and rod sampling algorithms described in section 3.2 were applied to a test suite of six systems and fractal versions of these systems. These systems are variations on the tensegrity icosahedron using rods of various lengths. These tests were run both with and without adaptation. The results are presented in tabular form in appendices: appendix A and appendix B provide the detailed results of non-adaptive, and respectively adaptive, tests. Below we provide a detailed discussion of one of these result tables.

Table 5.1 Small scale validation results using the endpoint sampling algorithm on fractal adaptive systems. This is a copy of table B.2 in appendix B.

Test	Time	Final EP	Visual Inspection	Steps	Rate of Convergence:			
					99%	90%	75%	50%
Default	21.24	11707.57	Y	181	1%	1%	1%	1%
Test1f	11008.58	7650.38	N	99063	0%	0%	0%	0%
Test2f	12173.88	8093.04	N	93953	0%	0%	0%	0%
Test3f	1241.19	2403.76	N	15459	0%	0%	0%	0%
Test4f	1222.6	558.32	Y	18777	2%	0%	0%	0%
Test5f	180.24	777.77	N	2659	1%	0%	0%	0%

The "Default" test, figure 5.1a, is a system set up to be stable from initialization, and should take no steps to optimize (in non-adaptive cases). The other tests are variations on the same connection set, with different rod lengths and starting positions. The base states are made into fractals by inserting a smaller version of themselves into one of their edges (see figure 5.1c).



(a) The "Default" test case.

(b) The "Test5" test case

(c) The "Test5f" test case, a fractal version of the "Test5" test case

Figure 5.1: Examples of starting test cases.

The "Time" column in table 5.1 is a measure of how long the algorithm took to run in seconds. This was measured using *Python*2.7's built in *time* module. "Final EP" gives the final potential energy of the system.

The ”Visual Inspection” column in table 5.1 records whether or not the systems looked as though they reached a correct stable state. A Y means that the sample passed the visual inspection, and a N means that it failed. The tests were designed to have an optimal state that approximated an icosahedron and any other state was judged as having failed this visual inspection. This visual inspection was conducted via the author looking at the output of the algorithm in a 3D rendered GUI. This method of validation is of course limited as small variances cannot be detected by the human eye, and as any novel and unexpected results can not be verified in this fashion. Due to the limitations of this form of validation, visual inspection was used only as a cursory criterion and a stronger validation methodology was applied in the large scale test suite (section 5.2). While visual inspection cannot ascertain that a sistem is in an optimal configuration, there are scenarios where it can ascertain that the system is in a suboptimal configuration. Table 5.1 has four tests which were deemed to be suboptimal by visual inspection (see figure 5.2). This poor performance is caused by the limitations of the endpoint sampling algorithm; this performance improved by using the rod sampling algorithm (figure 5.3 shows the output on the rod sampling algorithm on the starting state used in figure 5.2).

The "Steps" column in table 5.1 records the number of steps each test case required to reach a stable state. It was found during these experiments that the majority of the minimization occurred in the initial steps. To quantify this, the "Rate of Convergence" column tells us what percentage of the steps were required to reach what percentage of the energy minimization. In table 5.1, for example, "Test5f" using the endpoint sampling algorithm with adaptation took less than 0.5% of the total steps for optimization to reach 90% of the energy minimization that occurred, and by 1% of the algorithm 99% of the

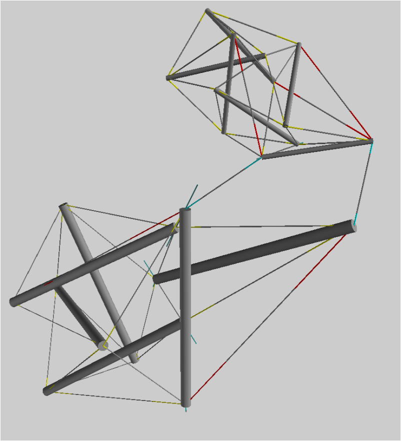


Figure 5.2: Result of the endpoint sampling algorithm on "Test5f". This failed the visual inspection.

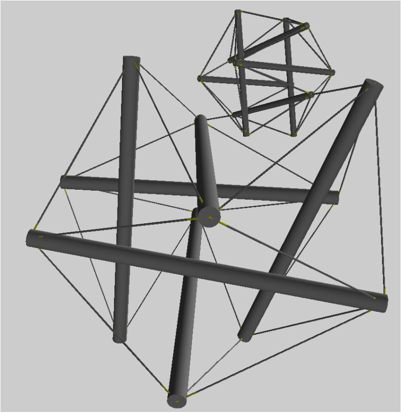


Figure 5.3: Result of the rod sampling algorithm on "Test5f". This passed the visual inspection.

minimization had occurred. In general, most tests reached 99% of their total optimization within the first 30% of the steps required for the optimization.

These results show that the end point sampling algorithm worked as well as the rod sampling algorithm on the regular tensegrity icosahedron (compare table A.1 to A.3, and also table B.1 to B.3), but on fractal system the rod sampling algorithm reached lower potential energy levels (compare table A.2 to A.4, and also table B.2 to B.4). Moreover, visual inspection of the results shows that the rod sampling algorithm always returned states that appear to be stable, whereas the endpoint sampling algorithm did not do so for fractal configurations. This prompted us to thoroughly test the rod sampling algorithm using a large scale validation (section 5.2).

5.2 Large Scale Test Suite

One of the major drawbacks of a hill climbing algorithm is that it can get stuck in suboptimal configurations. One of the purposes of this large scale validation suite is to determine whether local minima are a frequent problem for our hill climbing algorithm. A test suite of 100 different tensegrity icosahera was created. These were created by randomizing the length of the rods in the "Default" base case (figure 5.1a). Each of these test cases was then started from 1000 different starting positions and allowed to optimize (see figures 5.4 and 5.5). Fractal tensegrity icosahedra were also tested in this fashion, as described in section 5.2.2. Note that this form of validation requires that rods and elastics can pass through each other. otherwise suboptimal states could be constructed based on the starting positions of the rods.

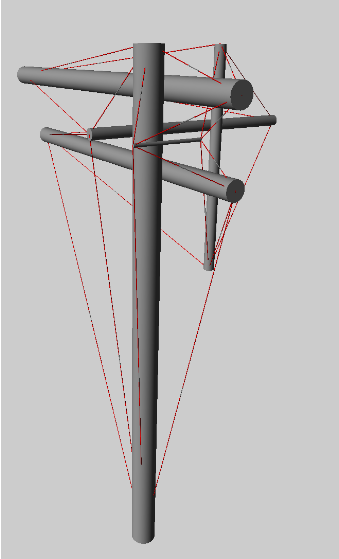


Figure 5.4: A base case used in the large scale test suite.

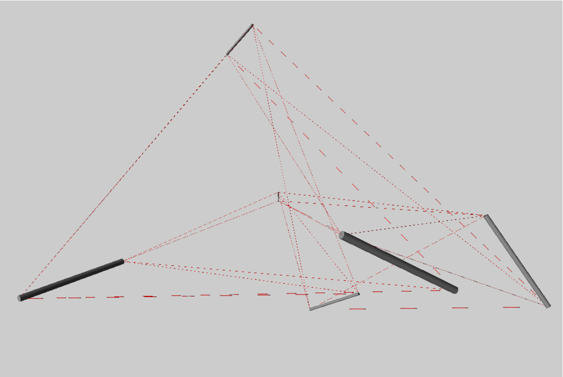


Figure 5.5: A randomly chosen starting position of the base case in figure 5.4

The aim of this suite was to test how often these different starting positions ended at comparable energy levels. We judged two systems to have roughly the same energy level if their potential energies were within 10^{-5}% of each other.

These tests were run without adaptation. Adaptation was avoided because the final potential energy level in an adaptive system is inherently dependent on its starting position; thus it cannot be used for comparisons across the various test cases.

5.2.1 Tensegrity Icosahedron: $100,000$ Tests

The full results of this suite are presented in appendix C. Table 5.2 below provides a sample of the results.

Table 5.2: Validation Results on Non-Fractal Systems (extract of table C.1)

Test	Trials	Average Steps	Average Final EP	Average Time	Average Dev	Average Dev %	Max Dev	Max Dev %
BaseState10	1000	545	21377.57	81	4.61E-06	2.16E-08%	1.98E-05	9.25E-08%
BaseState11	1000	707	35671.42	98	1.14E-05	3.20E-08%	6.06E-05	1.70E-07%
BaseState12	1000	611	36298.89	93	9.50E-06	2.62E-08%	5.26E-05	1.45E-07%

The name of the base case is recorded in the "Test" column, while the number of trials completed on that base case is stored in the "Trials" column. The "Average Time" column records the average execution time in seconds per trial. This value was generated using the *time* module, built into $Python2.7$. Each trial ends with a computation of the deviation between its final potential energy and the average potential energy for the base case, shown in the "Average Final EP" column. The "Average Dev" and "Average Dev %" column show the average of these deviations as raw numbers and as a percent of the final potential energy respectively. The largest of these deviations is recorded in the "Max Dev" and Max Dev %" columns.

Of the $100,000$ tests run $99,999$, or 99.999%, of the results had a deviation within the $10^{-5}\%$ tolerance margin. The one remaining test had a deviation of $10^{-2}\%$. The global average deviation over these $100,000$ tests was $4.60 \cdot 10^{-7}\%$.

These results suggest that regardless of starting position, the hill climbing algorithm applied to tensegrity icosahedra rarely gets stuck in suboptimal states, but instead reaches a global minimum. This test suite was repeated using fractal systems, as discussed in the following section.

5.2.2 Fractal Tensegrity Icosahedron: $87,189$ Tests

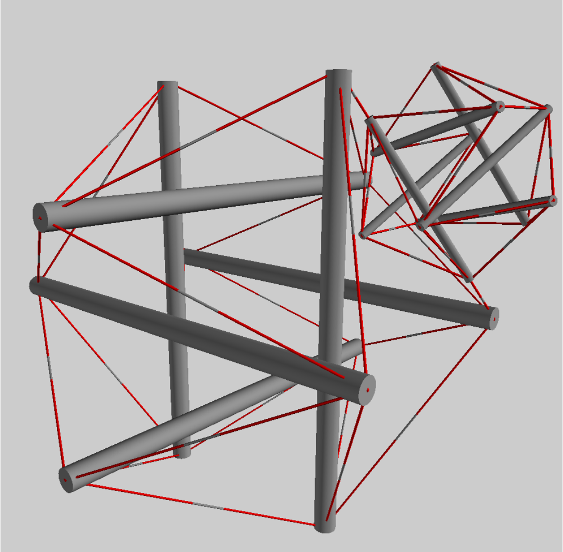


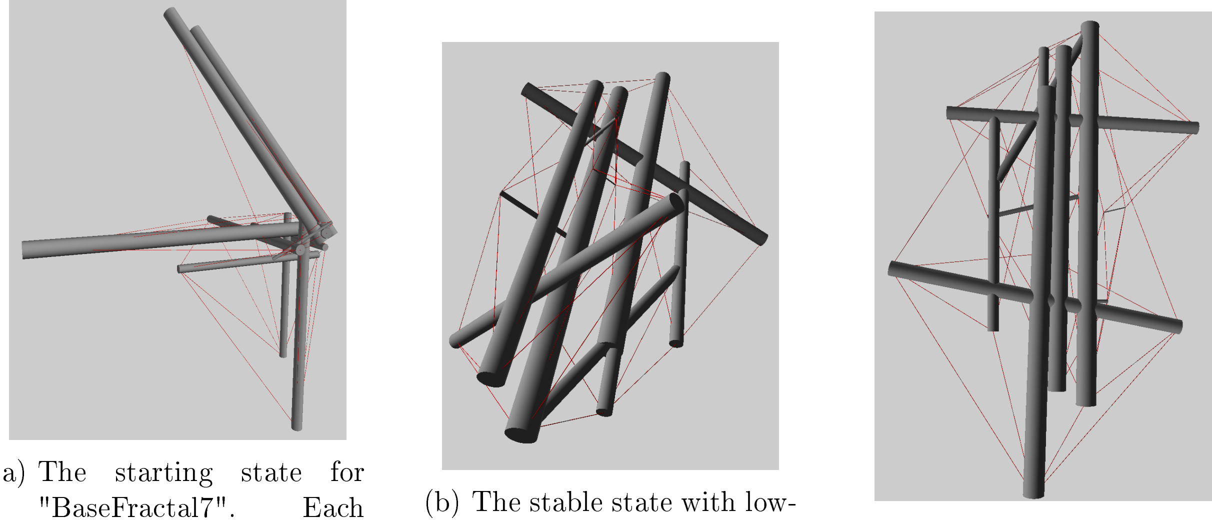
Figure 5.6: The starting case for the fractal icosahedron test suite.

Full results can be found in appendix D. The results of this suite are tabulated using the same notation as for the regular icosahedron test suite described in section 5.2.1.

This large scale test suite uses the fractal base shown in figure 5.6. From this starting case, we randomized the length of the rods to create the 100 base cases used in this test suite. Due to limited availability of server time, not all of these tests were run in the intended 1000 various starting positions; the number of trials run is shown in the $Trials$ column of table D.1.

Of the $87,189$ tests run $87,181$, or 99.990%, of them fell within the goal of having deviations smaller than $10^{-5}\%$ from the mean final potential energy.

One of the tests which had a higher deviation was "BaseFractal7", shown in figure 5.7. This base case was run 723 times; two of the trials (i.e. 0.27%), fell outside of the acceptable deviation range.



(a) The starting state for "BaseFractal7". Each trial randomized the positions of the rods in this state then ran the hill climbing algorithm to find a stable state.

(b) The stable state with lowest potential energy, as found by 721 of the 723 trials run on randomizations on (a).

(c) The results of the trial which had a high deviation from the usual potential energy atained for "BaseFractal7"

Figure 5.7: An example of a result with higher than desired deviation of potential energy.

Over 99.99% of the trials did not get caught in local minima. This shows that our hill climbing algorithm, using the rod sampling algorithm, does not frequently get stuck in suboptimal positions when applied to tensegrity icosahedron and fractal versions of the tensegrity icosahedron. In future work simulated annealing [11] could be investigated to further improve these results.

Chapter 6

Conclusions

This thesis has laid out a flexible simulation framework for modeling and exploring tensegrity systems.

6.1 Summary of Contributions

This thesis has made the following contributions:

1. A flexible framework for defining (chapter 2) and modeling (chapter 3) a wide variety tensegrity systems. Four components define an object (section 2.1):

   - A list of fixed parameters that define the attributes of the object.
   - A list of variable parameters that define the state space of the object.
   - A potential energy function that uses the fixed and variable parameters to determine the potential energy of the object in each state.

   The flexibility built into the framework allows for easy customization of the simulation, and allows for scientific exploratory research (section 3.1). This ease of customization is brought about by:

- State files contain all information needed to generate objects. This allows for easy and convenient generation of a variety of models without need to modify the code. An example of a state file is given in appendix E.
- There is a clear separation between object definition, object sampling, and optimization algorithm. The hill climbing optimization algorithm presented in section 3.3 requires a sampling algorithm. The clear separation between these algorithms allows the user to easily replace either the sampling algorithm or optimization algorithm.

This exploration lead to the discovery of an alternate stable state for the tensegrity icosahedron (figure 1.11b).

2. An optimization algorithm for finding the stable states of tensegrity systems (section 3.3), as well as two competing sampling algorithms for the optimization algorithm (section 3.2). The flexibility of the framework is highlighted by our ability to easily incorporate adaptivity (chapter 4) to our optimization methods.

3. An extensive validation suite (described and summarized in chapter 5) showing that the hill climbing algorithm sampling the state space using the rod sampling algorithm described in section 3.2.2 finds an optimal minimized energy state in over 99% of test cases.

6.2 Future Work

This section discusses some directions for expanding this body of work. Complexity can be added to the system via more types of atomic objects (Section 6.2.1), and by modeling hydrostatic pressures (section 6.2.2).

6.2.1 Expanding the Set of Objects

Currently our implementation of the framework only uses atomic objects which are straight one dimensional rods and one dimensional elastics. The addition of arbitrarily shaped rods as well as two and three dimensional elastics and rigid objects would allow for greater flexibility in exploration and testing tensegrity systems, such as those shown in figures 6.1 and 6.2.

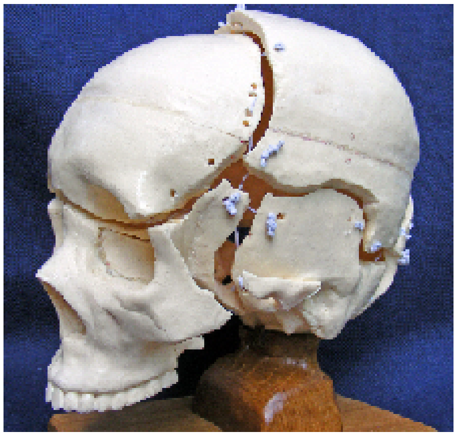


Figure 6.1: An example of a tensegrity model using 3D rigid objects. ©Dr. G. Scarr, reproduced with permission. (http://www.tensegrityinbiology.co.uk/)

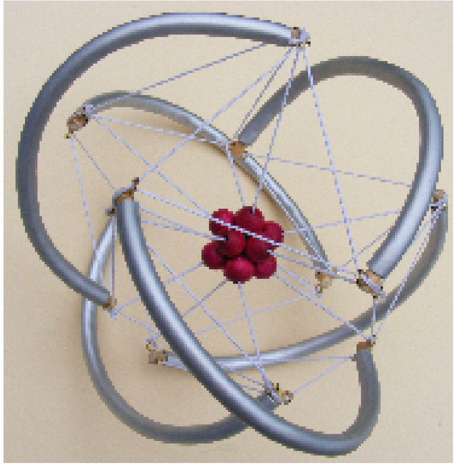


Figure 6.2: An example of a tensegrity model using curved rigid struts. ©Dr. G. Scarr, reproduced with permission. (http://www.tensegrityinbiology.co.uk/)

To define these new types of atomic objects we must define the four components set out in section 2.1: the fixed parameters, the state vector along with an associated interpretation, the potential energy of the object as a function of its state vector, and an approach for minimizing potential energy. In addition, the state file encoding (Appendix E) must be extended to include the new types of objects.

6.2.2 Modeling Hydrostatic Pressures

Hydrostatic pressures play an important role in biological systems [10].There is need to add the modeling of hydrostatic forces to existing biotensegrity models [22].

Here we outline a method for extending our tensegrity framework to model bags of hydrostatic pressure:

- Triangulate the surface of the desired hydrostatic bag.

- Define a compound object made up of springs that are connected as per the edges in the above triangulation. We use springs instead of elastics so that edges push back when compressed.

- Define the potential energy function of this object provides a great penalty for deviations between its current volume and a predefined desired volume for the hydrostatic bag. By making this energy penalty large enough we can enforce a ”constant volume” constraint.

Bibliography

[1] Robert William Burkhardt. *A Practical Guide to Tensegrity Design.* Cambridge, MA, 2008.

[2] Buckminster Fuller. Tensegrity. *Portfolio and Art News Annual*, vol 4, 1961.

[3] Buckminster Fuller and Edgar J. Applewhite. *Synergetics.* Macmillan Publishing Co. Inc., 1979.

[4] Vyacheslav I Jdanov. A tensegrity based structure optimization framework. Master's thesis, Queen's University, expected completion August 2014.

[5] A.W. Johnson and S.H. Jacobson. On the convergence of generalized hill climbing algorithms. *Discrete Applied Mathematics*, 119:37–57, 2002.

[6] Chirstopher Frederick Jones. Cox rayner + arup complete worlds largest tensegrity bridge in brisbane. *World Architecture News*, October 2009.

[7] Valentin Gomez Jouregui. Tensegrity structures and their application to architecture. Master's thesis, Queen's University Belfast, 2004.

[8] Sergi Hernandez Juan and Josep M. Mirats Tur. Tensegrity frameworks and static analysis review. *Mechanism and Machine Theory*, vol 43:859 – 881, July 2008.

[9] Diane Kelly. Industrial scientific software and a set of interviews on software development. In *Proceedings of the 2013 Conference of the Center for Advanced Studies on*

Collaborative Research, CASCON '13, pages 299 – 310, Riverton, NJ, USA, 2013. IBM Corp.

[10] William M. Kier. The diversity of hydrostatic skeletons. *The Journal of Experimental Biology*, vol 215:1247–1257, 2012.

[11] Scott Kirkpatrick, Daniel C. Gelatt, and Mario P. Vecchi. Optimization by simulated annealing. *Science*, vol 220:671 – 680.

[12] Lev Davidovich Landau, Arnol'd Markovich Kosevich, and Evgenii Mikhailovich Lifshitz. *Theory of Elasticity.* Oxford [England] and Burlington, MA and Butterworth-Heinemann, 3rd english edition, 1986.

[13] Stephen M. Levin. The tensegrity-truss as a model for spine mechanics and biotensegrity. *Journal of Mechanics in Medicine and Biology*, vol 2:375 – 388.

[14] Zbigniew Michalewicz and David B. Fogel. *How to Solve It and Modern Heuristics.* Springer books, 2^{nd} edition, 2004.

[15] Rene Motro. Tensegrity systems and the state of the art. *Journal of Space Structures*, vol 7:75 – 83, 1992.

[16] Thomas W. Myers. *Anatomy Trains.* Churchill Livingstone, 3 edition, 2014.

[17] Graham Scarr. A model of the cranial vault as a tensegrity structure and its significance to normal and abnormal cranial development. *International Journal of Osteopathic Medicine*, vol 11:80 – 89, September 2008.

[18] Graham Scarr. Simple geometry in complex organisms. *Journal of Bodywork and Movement Therapies*, 14:424–444, 2010.

[19] Robert Schleip, Thomas W. Findley, Leon Chaitow, and Peter A. Huijing, editors. *Fascia: The Tensional Network of the Human Body.* Churchill Livingstone Elsevier, 2012.

[20] Tristan d'Estree Streck. Using actuated tensegrity structures to produce a responsive architecture. *ACADIA22 and Connecting Crossroads of Digital Discourse*, pages 84 – 93, October 2003.

[21] Vytas SunSpiral, George Gorospe, Jonathan Bruce, Atil Iscen, George Korbel, Sophie Milam, Adrian Agogino, and David Atkinson. Tensegrity based probes for planetary exploration and entry, descent and landing (edl) and surface mobility analysis. *International Journal of Planetary Probes*, June 2013.

[22] Randel L. Swanson II. Biotensegrity: A unifying theory of biological architecture with applications to osteopathic practice, education, and research – a review and analysis. *The Journal of the American Osteopathic Association*, vol 113:34 – 52, January 2013.

[23] Can A. Yucesoy and Peter A. Huijing. Mathematical modeling. In Robert Schleip, Thomas W. Findley, Leon Chaitow, and Peter A. Huijing, editors, *Fascia: The Tensional Network of the Human Body*, pages 503–509. Churchill Livingstone Elsevier, 2012.

Appendix A

Small Scale Validation with Non-Adaptive Systems

The format and meaning of these tables are described in section 5.1.

Table A.1 Results using the endpoint sampling algorithm on non-adaptive systems. Sample reading: "Test3" used 4% of the optimization steps to reach 50% of the energy minimization, by 21% of the optimization steps, the system had reached 99% of the minimization. For full details on these tables see section 5.1

Test	Time	Final EP	Visual Inspection	Steps	Rate of Convergence:			
					99%	90%	75%	50%
Default	4.84	2772.18	Y	1	100%	100%	100%	100%
Test1	15.13	2656.31	Y	333	21%	11%	5%	2%
Test2	14.9	2656.31	Y	317	21%	12%	5%	2%
Test3	10.44	121.22	Y	177	21%	11%	7%	4%
Test4	9.49	835.59	Y	151	23%	9%	5%	1%
Test5	9.99	121.22	Y	166	8%	52%	4%	2%

Table A.2 Results using the endpoint sampling algorithm on fractal non-adaptive systems

Test	Time	Final EP	Visual Inspection	Steps	Rate of Convergence:			
					99%	90%	75%	50%
Default	30.5	3173.43	Y	319	19%	6%	4%	3%
Test1f	126.61	3226.44	N	1862	6%	2%	1%	0%
Test2f	9262.33	3251.07	N	74593	0%	0%	0%	0%
Test3f	48.31	256.04	N	595	10%	5%	2%	1%
Test4f	65.57	975.92	N	860	18%	2%	1%	0%
Test5f	55.64	505.95	N	702	3%	1%	1%	1%

Table A.3 Results using the rod sampling algorithm on non-adaptive systems

Test	Time	Final EP	Visual Inspection	Steps	Rate of Convergence:			
					99%	90%	75%	50%
Default	9.69	2772.18	Y	1	100%	100%	100%	100%
Test1	55.92	2656.31	Y	554	27%	14%	6%	2%
Test2	58.51	2656.31	Y	586	22%	14%	7%	2%
Test3	28.99	121.22	Y	232	37%	24%	14%	7%
Test4	20.21	835.59	Y	128	20%	2%	2%	2%
Test5	22.71	121.22	Y	154	22%	12%	8%	5%

Table A.4 Results using the rod sampling algorithm on fractal non-adaptive systems

Test	Time	Final EP	Visual Inspection	Steps	Rate of Convergence:			
					99%	90%	75%	50%
Default	184.38	3169.6	Y	986	7%	0%	0%	0%
Test1f	585.39	3035.2	Y	3354	5%	2%	1%	0%
Test2f	487.27	3035.2	Y	2895	5%	3%	1%	0%
Test3f	181.93	136.82	Y	976	12%	6%	4%	2%
Test4f	112.33	966.77	Y	498	15%	1%	1%	0%
Test5f	181.11	119.65	Y	900	5%	2%	1%	1%

Appendix B

Small Scale Validation with Adaptive Systems

The format and meaning of these tables are described in section 5.1.

Table B.1 Results using the endpoint sampling algorithm on adaptive systems

Test	Time	Final EP	Visual Inspection	Steps	Rate of Convergence:			
					99%	90%	75%	50%
Default	5.08	10315.43	Y	1	100%	100%	100%	100%
Test1	282.92	5890.59	Y	8147	1%	1%	0%	0%
Test2	24.56	6657.1	Y	590	16%	7%	6%	4%
Test3	11.09	1094.18	Y	183	15%	10%	9%	7%
Test4	9.75	461.71	Y	143	18%	10%	6%	3%
Test5	10.59	326.83	Y	173	8%	5%	4%	3%

Table B.2 Results using the endpoint sampling algorithm on fractal adaptive systems

Test	Time	Final EP	Visual Inspection	Steps	Rate of Convergence:			
					99%	90%	75%	50%
Default	21.24	11707.57	Y	181	1%	1%	1%	1%
Test1f	11008.58	7650.38	N	99063	0%	0%	0%	0%
Test2f	12173.88	8093.04	N	93953	0%	0%	0%	0%
Test3f	1241.19	2403.76	N	15459	0%	0%	0%	0%
Test4f	1222.6	558.32	Y	18777	2%	0%	0%	0%
Test5f	180.24	777.77	N	2659	1%	0%	0%	0%

Table B.3 Results using the rod sampling algorithm on adaptive systems

Test	Time	Final EP	Visual Inspection	Steps	Rate of Convergence:			
					99%	90%	75%	50%
Default	1.39	10315.42	Y	1	100%	100%	100%	100%
Test1	10.48	6178.1	Y	740	21%	13%	9%	6%
Test2	10.52	6360.68	Y	743	16%	12%	9%	7%
Test3	4.59	1215.11	Y	261	33%	26%	20%	16%
Test4	3.37	448.77	Y	159	20%	8%	4%	2%
Test5	4.07	488.57	Y	211	16%	12%	9%	8%

Table B.4 Results Using the Rod Sampling Algorithm on Fractal Adaptive Systems

Test	Time	Final EP	Visual Inspection	Steps	Rate of Convergence:			
					99%	90%	75%	50%
Default	120.83	11748.33	Y	599	0%	0%	0%	0%
Test1f	612.63	7058.1	Y	3524	7%	3%	2%	1%
Test2f	714.66	8041.77	Y	4133	5%	3%	2%	1%
Test3f	222.13	1353.27	Y	1208	11%	7%	5%	4%
Test4f	181.54	523.65	Y	962	14%	6%	1%	0%
Test5f	137.19	504.66	Y	704	6%	3%	2%	2%

Appendix C

Large Scale Validation for Tensegrity Icosahedron

The tables in appendix C and D provide the results of the large scale validation tests described in section 5.2. For an explanation of the entries in these tables see section 5.2.

Table C.1: Validation Results on Non-Fractal Systems

Test	Number of Trials	Average Steps	Average Final EP	Average Time	Average Dev	Average Dev %	Max Dev	Max Dev %
BaseState0	1000	1880	26577.51	173	2.81E-05	1.06E-07%	1.05E-04	3.96E-07%
BaseState1	1000	2050	24330.58	274	1.48E-05	6.10E-08%	6.67E-05	2.74E-07%
BaseState10	1000	545	21377.57	81	4.61E-06	2.16E-08%	1.98E-05	9.25E-08%
BaseState11	1000	707	35671.42	98	1.14E-05	3.20E-08%	6.06E-05	1.70E-07%
BaseState12	1000	611	36298.89	93	9.50E-06	2.62E-08%	5.26E-05	1.45E-07%
BaseState13	1000	714	42896.95	105	1.27E-05	2.96E-08%	5.15E-05	1.20E-07%
BaseState14	1000	753	32528.06	109	9.80E-06	3.01E-08%	4.32E-05	1.33E-07%
BaseState15	1000	597	16978.87	89	2.70E-06	1.59E-08%	3.22E-05	1.90E-07%
BaseState16	1000	570	24361.19	85	6.56E-06	2.69E-08%	2.82E-05	1.16E-07%
BaseState17	1000	744	55123.24	109	1.56E-05	2.82E-08%	6.47E-05	1.17E-07%

Validation Results on Non-Fractal Systems - continued

Test	Number of Trials	Average Steps	Average Final EP	Average Time	Average Dev	Average Dev %	Max Dev	Max Dev %
BaseState18	1000	581	36651.26	89	8.10E-06	2.21E-08%	3.77E-05	1.03E-07%
BaseState19	1000	688	26463.51	100	8.36E-06	3.16E-08%	5.26E-05	1.99E-07%
BaseState2	1000	1042	32305.32	146	9.05E-06	2.80E-08%	4.27E-05	1.32E-07%
BaseState20	1000	551	29042.57	84	6.46E-06	2.23E-08%	3.56E-05	1.22E-07%
BaseState21	1000	499	24650.45	76	5.16E-06	2.09E-08%	2.80E-05	1.14E-07%
BaseState22	1000	725	25107.08	102	7.07E-06	2.82E-08%	3.04E-05	1.21E-07%
BaseState23	1000	878	47250.13	118	2.38E-05	5.04E-08%	1.06E-04	2.25E-07%
BaseState24	1000	1441	23609.89	159	2.11E-05	8.94E-08%	8.23E-05	3.49E-07%
BaseState25	1000	1348	34413.78	150	2.74E-05	7.96E-08%	1.18E-04	3.43E-07%
BaseState26	1000	853	27999.54	117	8.02E-06	2.86E-08%	5.33E-05	1.91E-07%
BaseState27	1000	675	34915.52	96	8.44E-06	2.42E-08%	4.74E-05	1.36E-07%
BaseState28	1000	1211	23489.88	149	6.01E-06	2.56E-08%	4.23E-05	1.80E-07%
BaseState29	1000	1164	22769.68	144	7.38E-06	3.24E-08%	4.73E-05	2.08E-07%
BaseState3	1000	624	45100.10	93	8.23E-06	1.82E-08%	4.59E-05	1.02E-07%
BaseState30	1000	564	29132.59	80	7.43E-06	2.55E-08%	3.44E-05	1.18E-07%
BaseState31	1000	742	18202.35	106	4.82E-06	2.65E-08%	1.85E-05	1.02E-07%
BaseState32	1000	804	30235.32	110	1.01E-05	3.33E-08%	4.43E-05	1.47E-07%
BaseState33	1000	764	45277.11	105	1.51E-05	3.33E-08%	7.45E-05	1.65E-07%
BaseState34	1000	1427	31154.72	158	1.36E-05	4.35E-08%	6.50E-05	2.09E-07%
BaseState35	1000	877	36128.96	111	1.71E-05	4.72E-08%	8.17E-05	2.26E-07%
BaseState36	1000	1017	27848.24	126	2.12E-05	7.61E-08%	8.87E-05	3.19E-07%
BaseState37	1000	707	26440.39	96	5.86E-06	2.22E-08%	3.75E-05	1.42E-07%
BaseState38	1000	738	59875.50	102	1.55E-05	2.59E-08%	6.05E-05	1.01E-07%
BaseState39	1000	725	23794.05	103	5.48E-06	2.30E-08%	3.60E-05	1.51E-07%
BaseState4	1000	1221	33720.88	172	3.31E-05	9.83E-08%	1.29E-04	3.82E-07%

Validation Results on Non-Fractal Systems - continued

Test	Number	Average	Average	Average	Average	Average Dev	Max	Max
	of Trials	Steps	Final EP	Time	Dev	%	Dev	Dev %
BaseState40	1000	1332	39046.10	154	4.32E-05	1.11E-07%	1.58E-04	4.06E-07%
BaseState41	1000	915	45048.19	128	1.74E-05	3.87E-08%	7.33E-05	1.63E-07%
BaseState42	1000	1074	52068.12	130	2.06E-05	3.95E-08%	1.18E-04	2.27E-07%
BaseState43	1000	330	6152.54	55	7.32E-07	1.19E-08%	3.43E-06	5.57E-08%
BaseState44	1000	516	22486.22	78	4.63E-06	2.06E-08%	2.41E-05	1.07E-07%
BaseState45	1000	1081	33526.71	138	2.04E-05	6.09E-08%	8.52E-05	2.54E-07%
BaseState46	1000	652	41187.12	91	1.15E-05	2.79E-08%	4.98E-05	1.21E-07%
BaseState47	1000	525	25007.88	76	5.45E-06	2.18E-08%	2.61E-05	1.04E-07%
BaseState48	1000	851	40968.81	114	1.98E-05	4.84E-08%	9.20E-05	2.25E-07%
BaseState49	1000	826	47069.61	118	1.76E-05	3.74E-08%	6.73E-05	1.43E-07%
BaseState5	1000	673	30231.13	102	1.03E-05	3.40E-08%	4.36E-05	1.44E-07%
BaseState50	1000	1074	16036.84	135	6.07E-06	3.78E-08%	2.94E-05	1.84E-07%
BaseState51	1000	990	51960.79	134	2.58E-05	4.97E-08%	1.18E-04	2.27E-07%
BaseState52	1000	1513	30834.82	162	4.25E-05	1.38E-07%	1.67E-04	5.40E-07%
BaseState53	1000	811	38401.50	113	1.07E-05	2.77E-08%	5.18E-05	1.35E-07%
BaseState54	1000	1278	25349.70	153	9.28E-06	3.66E-08%	4.73E-05	1.87E-07%
BaseState55	1000	1044	28616.64	133	1.16E-05	4.05E-08%	5.71E-05	1.99E-07%
BaseState56	1000	782	44173.01	112	1.45E-05	3.28E-08%	7.17E-05	1.62E-07%
BaseState57	1000	310	6483.36	53	8.50E-07	1.31E-08%	3.82E-06	5.90E-08%
BaseState58	1000	1264	47732.87	146	3.51E-05	7.35E-08%	1.52E-04	3.18E-07%
BaseState59	1000	502	17653.65	76	4.89E-06	2.77E-08%	2.98E-05	1.69E-07%
BaseState6	1000	988	46041.42	143	1.80E-05	3.90E-08%	1.06E-04	2.31E-07%
BaseState60	1000	550	31015.94	77	8.28E-06	2.67E-08%	3.83E-05	1.24E-07%
BaseState61	1000	480	13259.16	44	3.06E-06	2.31E-08%	1.39E-05	1.05E-07%
BaseState62	1000	591	16270.32	53	3.69E-06	2.27E-08%	1.86E-05	1.14E-07%

Validation Results on Non-Fractal Systems - continued

Test	Number of Trials	Average Steps	Average Final EP	Average Time	Average Dev	Average Dev %	Max Dev	Max Dev %
BaseState63	1000	647	41703.98	56	1.05E-05	2.53E-08%	5.57E-05	1.33E-07%
BaseState64	1000	621	21604.24	55	5.90E-06	2.73E-08%	2.47E-05	1.14E-07%
BaseState65	1000	786	24613.30	65	5.32E-06	2.16E-08%	4.11E-05	1.67E-07%
BaseState66	1000	736	39451.21	62	1.21E-05	3.06E-08%	7.56E-05	1.92E-07%
BaseState67	1000	446	9464.27	42	2.18E-06	2.31E-08%	1.05E-05	1.11E-07%
BaseState68	1000	462	20448.40	44	3.23E-06	1.58E-08%	1.40E-05	6.85E-08%
BaseState69	1000	671	35896.46	58	1.18E-05	3.30E-08%	6.47E-05	1.80E-07%
BaseState7	1000	1132	20082.79	119	5.83E-06	2.90E-08%	2.71E-05	1.35E-07%
BaseState70	1000	585	22078.62	67	2.91E-06	1.32E-08%	1.97E-05	8.91E-08%
BaseState71	1000	755	26076.10	81	8.60E-06	3.30E-08%	4.47E-05	1.72E-07%
BaseState72	1000	1033	32379.99	108	1.58E-05	4.89E-08%	6.39E-05	1.97E-07%
BaseState73	1000	602	23867.98	66	7.34E-06	3.08E-08%	4.30E-05	1.80E-07%
BaseState74	1000	1980	28831.21	148	5.52E-05	1.92E-07%	1.85E-04	6.43E-07%
BaseState75	1000	2813	34540.44	226	8.65E-05	2.51E-07%	3.60E-04	1.04E-06%
BaseState76	1000	784	21143.66	71	4.92E-06	2.33E-08%	2.01E-05	9.50E-08%
BaseState77	1000	516	22818.94	68	4.31E-06	1.89E-08%	2.66E-05	1.17E-07%
BaseState78	1000	1057	35780.55	96	2.44E-05	6.83E-08%	1.01E-04	2.83E-07%
BaseState79	1000	1706	26899.11	141	2.75E-05	1.02E-07%	1.22E-04	4.55E-07%
BaseState8	1000	840	40623.94	120	1.75E-05	4.31E-08%	7.89E-05	1.94E-07%
BaseState80	1000	1293	35603.11	126	1.82E-05	5.11E-08%	8.79E-05	2.47E-07%
BaseState81	1000	482	22783.34	66	3.90E-06	1.71E-08%	1.94E-05	8.49E-08%
BaseState82	1000	706	15224.74	86	3.68E-06	2.41E-08%	1.68E-05	1.10E-07%
BaseState83	1000	1014	33741.11	99	1.42E-02	4.20E-05%	7.08E+00	2.10E-02%
BaseState84	1000	823	36746.26	88	1.94E-05	5.28E-08%	8.34E-05	2.27E-07%
BaseState85	1000	650	36293.59	57	1.00E-05	2.77E-08%	4.73E-05	1.30E-07%

Validation Results on Non-Fractal Systems - continued

Test	Number of Trials	Average Steps	Average Final EP	Average Time	Average Dev	Average Dev %	Max Dev	Max Dev %
BaseState86	1000	899	42851.81	92	1.74E-05	4.06E-08%	7.08E-05	1.65E-07%
BaseState87	1000	867	35914.97	78	1.11E-05	3.09E-08%	5.02E-05	1.40E-07%
BaseState88	1000	651	44660.97	66	9.76E-06	2.19E-08%	7.54E-05	1.69E-07%
BaseState89	1000	440	20963.68	52	3.00E-06	1.43E-08%	1.25E-05	5.95E-08%
BaseState9	1000	430	18363.56	67	2.81E-06	1.53E-08%	1.49E-05	8.11E-08%
BaseState90	1000	777	22766.52	87	8.03E-06	3.53E-08%	4.28E-05	1.88E-07%
BaseState91	1000	688	41812.45	79	1.19E-05	2.84E-08%	4.61E-05	1.10E-07%
BaseState92	1000	631	42370.68	74	8.05E-06	1.90E-08%	3.83E-05	9.04E-08%
BaseState93	1000	792	30119.26	72	9.74E-06	3.23E-08%	5.41E-05	1.80E-07%
BaseState94	1000	1122	24416.70	107	8.19E-06	3.35E-08%	6.87E-05	2.81E-07%
BaseState95	1000	1380	40295.31	124	2.91E-05	7.23E-08%	9.84E-05	2.44E-07%
BaseState96	1000	797	28321.81	83	8.01E-06	2.83E-08%	5.57E-05	1.97E-07%
BaseState97	1000	869	25325.37	73	1.16E-05	4.56E-08%	4.54E-05	1.79E-07%
BaseState98	1000	546	21108.11	65	6.01E-06	2.85E-08%	3.12E-05	1.48E-07%
BaseState99	1000	757	50529.46	76	1.37E-05	2.71E-08%	7.89E-05	1.56E-07%

Appendix D

Large Scale Validation for Fractal Tensegrity Icosahedron

The tables in appendix C and D provide the results of the large scale validation tests described in section 5.2. For an explanation of the entries in these tables see section 5.2.

Table D.1: Validation Results on Fractal Systems

Test	Number of Trials	Average Steps	Average Final EP	Average Time	Average Dev	Average Dev %	Max Dev	Max Dev %
BaseStateFractal0	1001	5495	70376.72	1438	2.55E-04	3.62E-07	1.08E-03	1.54E-06
BaseStateFractal1	1001	4783	37097.71	1215	1.07E-04	2.87E-07	7.27E-04	1.96E-06
BaseStateFractal10	1001	1912	57921.69	503	3.03E-05	5.22E-08	1.27E-04	2.19E-07
BaseStateFractal11	1001	6537	52206.01	1707	1.10E-04	2.10E-07	3.53E-04	6.76E-07
BaseStateFractal12	1001	4451	74916.36	1165	2.09E-04	2.79E-07	1.01E-03	1.35E-06
BaseStateFractal13	1001	4316	66078.34	1102	2.14E-04	3.24E-07	6.13E-04	9.28E-07
BaseStateFractal14	1001	4099	41108.85	1096	1.23E-04	3.00E-07	4.26E-04	1.04E-06
BaseStateFractal15	1001	2337	33449.70	625	4.65E-05	1.39E-07	2.01E-04	6.01E-07
BaseStateFractal16	1001	5376	63446.01	1374	9.82E-05	1.55E-07	3.28E-04	5.18E-07
BaseStateFractal17	1001	3445	80869.58	883	1.50E-04	1.85E-07	5.98E-04	7.40E-07

Validation Results on Fractal Systems - continued

Test	Number of Trials	Average Steps	Average Final EP	Average Time	Average Dev	Average Dev %	Max Dev	Max Dev %
BaseStateFractal18	1001	7399	57952.35	1925	4.97E-02	8.58E-05	6.65E+00	1.15E-02
BaseStateFractal19	1001	4560	53644.48	1178	8.50E-05	1.58E-07	3.65E-04	6.81E-07
BaseStateFractal2	1001	5557	51183.29	1437	2.01E-04	3.92E-07	8.25E-04	1.61E-06
BaseStateFractal20	1001	2461	55392.51	629	5.10E-05	9.21E-08	2.71E-04	4.90E-07
BaseStateFractal21	1001	4733	39102.36	1224	1.57E-04	4.02E-07	7.64E-04	1.95E-06
BaseStateFractal22	1001	7504	82884.95	1949	4.64E-04	5.60E-07	1.84E-03	2.22E-06
BaseStateFractal23	1001	4523	36632.17	1138	5.66E-05	1.55E-07	1.08E-03	2.96E-06
BaseStateFractal24	1001	4959	54269.12	1266	1.67E-04	3.07E-07	5.17E-04	9.53E-07
BaseStateFractal25	1001	3371	52805.19	900	6.33E-05	1.20E-07	2.42E-04	4.59E-07
BaseStateFractal26	1001	4247	62431.49	1114	1.54E-04	2.47E-07	9.38E-04	1.50E-06
BaseStateFractal27	1001	3441	51648.18	911	5.60E-05	1.09E-07	2.51E-04	4.86E-07
BaseStateFractal28	928	12917	65632.65	3339	5.45E-04	8.31E-07	2.72E-03	4.15E-06
BaseStateFractal29	1001	3712	43022.57	958	5.59E-05	1.30E-07	2.65E-04	6.15E-07
BaseStateFractal3	1001	2743	38333.67	715	3.49E-05	9.12E-08	1.25E-04	3.25E-07
BaseStateFractal30	1001	3631	72303.60	956	1.38E-04	1.91E-07	2.19E-03	3.02E-06
BaseStateFractal31	1001	5015	71086.72	1276	2.89E-04	4.06E-07	1.50E-03	2.11E-06
BaseStateFractal32	1001	2706	17065.87	731	7.84E-05	4.60E-07	3.15E-04	1.85E-06
BaseStateFractal33	1001	2475	43863.54	649	4.32E-05	9.85E-08	2.46E-04	5.60E-07
BaseStateFractal34	1001	2464	37200.21	652	5.27E-05	1.42E-07	1.90E-04	5.11E-07
BaseStateFractal35	1001	4048	56154.90	1063	1.30E-04	2.32E-07	7.60E-04	1.35E-06
BaseStateFractal36	1001	8231	48794.67	2119	1.67E-04	3.43E-07	6.82E-04	1.40E-06
BaseStateFractal37	867	14112	60374.10	3582	1.72E-04	2.84E-07	6.86E-04	1.14E-06
BaseStateFractal38	1001	6799	51718.55	1769	1.47E-04	2.84E-07	6.10E-04	1.18E-06
BaseStateFractal39	1001	6296	91268.91	1631	4.52E-04	4.95E-07	1.64E-03	1.80E-06
BaseStateFractal4	960	12612	37558.08	3234	7.30E-04	1.94E-06	2.72E-03	7.23E-06

Validation Results on Fractal Systems - continued

Test	Number of Trials	Average Steps	Average Final EP	Average Time	Average Dev	Average Dev %	Max Dev	Max Dev %
BaseStateFractal40	1001	5461	63593.25	1443	3.02E-04	4.75E-07	1.23E-03	1.93E-06
BaseStateFractal41	1001	7875	63437.71	2047	2.13E-04	3.35E-07	7.05E-04	1.11E-06
BaseStateFractal42	1001	4456	40730.68	1187	7.73E-05	1.90E-07	3.17E-04	7.79E-07
BaseStateFractal43	1001	5533	45941.42	1438	1.44E-04	3.14E-07	5.85E-04	1.27E-06
BaseStateFractal44	1001	6019	57473.70	1554	1.66E-04	2.89E-07	1.00E-03	1.75E-06
BaseStateFractal45	1001	4216	55017.02	1087	1.82E-04	3.31E-07	6.71E-04	1.22E-06
BaseStateFractal46	1001	5224	55136.91	1372	9.60E-05	1.74E-07	4.02E-04	7.30E-07
BaseStateFractal47	806	10654	71633.62	2755	2.37E-02	3.31E-05	8.04E+00	1.12E-02
BaseStateFractal48	1001	2986	62650.51	772	7.90E-05	1.26E-07	3.38E-04	5.40E-07
BaseStateFractal49	1001	3822	78532.60	985	1.75E-04	2.23E-07	5.39E-04	6.86E-07
BaseStateFractal5	1001	5941	76856.73	1549	1.91E-04	2.49E-07	7.12E-04	9.26E-07
BaseStateFractal50	1001	3793	78251.41	968	1.68E-04	2.15E-07	5.79E-04	7.40E-07
BaseStateFractal51	1001	6457	73655.22	1692	2.56E-04	3.48E-07	7.97E-04	1.08E-06
BaseStateFractal52	505	15506	57380.24	4035	1.84E-04	3.20E-07	6.74E-04	1.17E-06
BaseStateFractal53	1001	7229	44906.27	1924	1.11E-04	2.48E-07	4.54E-04	1.01E-06
BaseStateFractal54	748	10397	70611.64	2676	3.94E-04	5.58E-07	7.54E-03	1.07E-05
BaseStateFractal55	1001	3786	39322.89	1002	6.47E-05	1.65E-07	2.82E-04	7.16E-07
BaseStateFractal56	1001	4863	58980.65	1215	6.25E-05	1.06E-07	2.81E-04	4.77E-07
BaseStateFractal57	1001	2731	60493.23	716	5.27E-05	8.72E-08	2.02E-04	3.34E-07
BaseStateFractal58	1001	1374	25677.16	364	9.44E-06	3.68E-08	4.01E-05	1.56E-07
BaseStateFractal59	932	7933	64814.22	2028	1.80E-04	2.78E-07	7.90E-04	1.22E-06
BaseStateFractal6	617	19711	95194.98	5031	5.45E-04	5.72E-07	2.57E-03	2.70E-06
BaseStateFractal60	1001	3732	42766.37	959	4.68E-05	1.09E-07	1.95E-04	4.57E-07
BaseStateFractal61	1001	5568	57198.96	1461	1.20E-04	2.09E-07	6.92E-04	1.21E-06
BaseStateFractal62	1001	2734	49783.30	699	4.61E-05	9.26E-08	1.92E-04	3.85E-07

Validation Results on Fractal Systems - continued

Test	Number of Trials	Average Steps	Average Final EP	Average Time	Average Dev	Average Dev %	Max Dev	Max Dev %
BaseStateFractal63	1001	4692	57496.70	1227	1.59E-04	2.77E-07	5.25E-04	9.13E-07
BaseStateFractal64	870	7252	70203.28	1913	4.37E-04	6.22E-07	1.56E-03	2.22E-06
BaseStateFractal65	1001	6179	71061.32	1619	2.15E-04	3.03E-07	1.29E-03	1.81E-06
BaseStateFractal66	1001	3948	49857.22	1018	4.98E-05	9.99E-08	1.92E-04	3.86E-07
BaseStateFractal67	955	6140	77611.42	1623	3.97E-04	5.11E-07	2.35E-03	3.03E-06
BaseStateFractal68	1001	4085	67515.09	1069	1.30E-04	1.93E-07	5.26E-04	7.79E-07
BaseStateFractal69	1001	4367	76614.92	1151	2.22E-04	2.89E-07	8.92E-04	1.16E-06
BaseStateFractal7	723	16493	68967.84	4293	1.43E-02	2.08E-05	3.18E+00	4.62E-03
BaseStateFractal70	946	5539	46717.60	1409	6.31E-05	1.35E-07	2.22E-04	4.75E-07
BaseStateFractal71	455	10981	46715.56	2815	8.71E-04	1.87E-06	2.99E-03	6.40E-06
BaseStateFractal72	1001	4739	52077.16	1224	1.46E-04	2.81E-07	6.23E-04	1.20E-06
BaseStateFractal73	1001	1781	37795.36	468	1.36E-05	3.60E-08	5.34E-05	1.41E-07
BaseStateFractal74	824	5468	67022.16	1462	2.92E-04	4.36E-07	1.11E-03	1.66E-06
BaseStateFractal75	878	5245	63327.76	1342	2.44E-04	3.85E-07	1.61E-03	2.55E-06
BaseStateFractal76	954	4868	46291.18	1229	5.77E-05	1.25E-07	2.38E-04	5.13E-07
BaseStateFractal77	1001	3507	45002.52	928	8.72E-05	1.94E-07	3.50E-04	7.77E-07
BaseStateFractal78	1001	4002	58680.72	1025	8.65E-05	1.47E-07	3.57E-04	6.09E-07
BaseStateFractal79	578	7560	29611.76	1947	1.23E-04	4.14E-07	7.98E-04	2.69E-06
BaseStateFractal8	1001	3482	30002.22	875	3.02E-05	1.01E-07	1.39E-04	4.63E-07
BaseStateFractal80	1001	2734	64619.76	714	5.89E-05	9.12E-08	1.99E-04	3.08E-07
BaseStateFractal81	649	6161	45722.79	1556	1.25E-04	2.74E-07	6.63E-04	1.45E-06
BaseStateFractal82	1001	3602	68832.69	928	8.75E-05	1.27E-07	4.43E-04	6.44E-07
BaseStateFractal83	242	15456	37187.14	4050	8.04E-04	2.16E-06	3.53E-03	9.49E-06
BaseStateFractal84	1001	3400	59242.30	891	8.16E-05	1.38E-07	3.41E-04	5.75E-07
BaseStateFractal85	718	5025	47514.35	1278	1.06E-04	2.22E-07	3.87E-04	8.15E-07

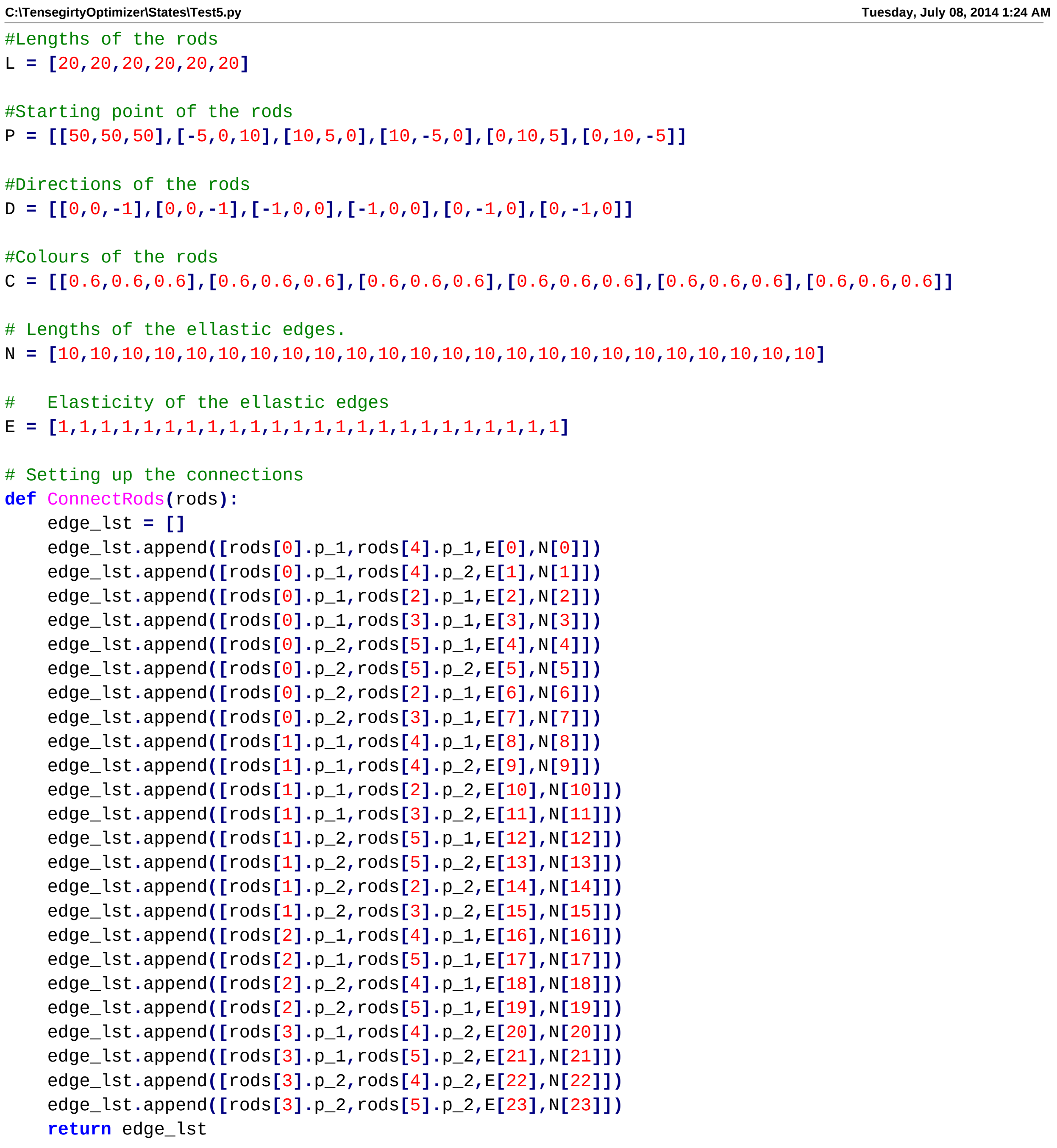
Validation Results on Fractal Systems - continued

Test	Number of Trials	Average Steps	Average Final EP	Average Time	Average Dev	Average Dev %	Max Dev	Max Dev %
BaseStateFractal86	375	8588	57918.92	2273	2.31E-04	3.98E-07	9.68E-04	1.67E-06
BaseStateFractal87	891	3652	48145.09	935	6.06E-05	1.26E-07	2.48E-04	5.16E-07
BaseStateFractal88	574	5006	63272.30	1307	1.72E-04	2.72E-07	8.76E-04	1.38E-06
BaseStateFractal89	553	5391	63498.25	1343	1.53E-04	2.41E-07	6.06E-04	9.55E-07
BaseStateFractal9	1001	7526	64229.57	1929	4.21E-04	6.55E-07	1.32E-03	2.06E-06
BaseStateFractal90	483	4319	80405.17	1111	1.35E-04	1.67E-07	5.73E-04	7.12E-07
BaseStateFractal91	376	5281	55739.43	1335	9.25E-05	1.66E-07	5.45E-04	9.78E-07
BaseStateFractal92	883	1970	32787.19	511	3.40E-05	1.04E-07	1.44E-04	4.39E-07
BaseStateFractal93	486	3447	80738.26	873	1.25E-04	1.54E-07	4.54E-04	5.63E-07
BaseStateFractal94	286	5036	72859.21	1302	3.12E-04	4.29E-07	1.77E-03	2.42E-06
BaseStateFractal95	166	8208	45933.07	2140	2.95E-04	6.42E-07	1.51E-03	3.28E-06
BaseStateFractal96	525	2364	32124.86	598	4.27E-05	1.33E-07	1.43E-04	4.44E-07
BaseStateFractal97	144	6649	66926.20	1663	2.98E-04	4.45E-07	1.78E-03	2.66E-06
BaseStateFractal98	127	7529	50732.52	1896	3.25E-04	6.41E-07	1.50E-03	2.95E-06
BaseStateFractal99	99	5431	49955.53	1292	4.80E-05	9.61E-08	1.91E-04	3.83E-07

Appendix E

State File

The next page shows a sample of a state file. This is a *.py* file that our implementation uses as input to generate a rod and elastic model. This file also contains some display information, such as rod colours.



C:\TensegirtyOptimizer\States\Test5.py Tuesday, July 08, 2014 1:24 AM

```
#Lengths of the rods
L = [20,20,20,20,20,20]

#Starting point of the rods
P = [[50,50,50],[-5,0,10],[10,5,0],[10,-5,0],[0,10,5],[0,10,-5]]

#Directions of the rods
D = [[0,0,-1],[0,0,-1],[-1,0,0],[-1,0,0],[0,-1,0],[0,-1,0]]

#Colours of the rods
C = [[0.6,0.6,0.6],[0.6,0.6,0.6],[0.6,0.6,0.6],[0.6,0.6,0.6],[0.6,0.6,0.6],[0.6,0.6,0.6]]

# Lengths of the ellastic edges.
N = [10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10]

#   Elasticity of the ellastic edges
E = [1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1]

# Setting up the connections
def ConnectRods(rods):
    edge_lst = []
    edge_lst.append([rods[0].p_1,rods[4].p_1,E[0],N[0]])
    edge_lst.append([rods[0].p_1,rods[4].p_2,E[1],N[1]])
    edge_lst.append([rods[0].p_1,rods[2].p_1,E[2],N[2]])
    edge_lst.append([rods[0].p_1,rods[3].p_1,E[3],N[3]])
    edge_lst.append([rods[0].p_2,rods[5].p_1,E[4],N[4]])
    edge_lst.append([rods[0].p_2,rods[5].p_2,E[5],N[5]])
    edge_lst.append([rods[0].p_2,rods[2].p_1,E[6],N[6]])
    edge_lst.append([rods[0].p_2,rods[3].p_1,E[7],N[7]])
    edge_lst.append([rods[1].p_1,rods[4].p_1,E[8],N[8]])
    edge_lst.append([rods[1].p_1,rods[4].p_2,E[9],N[9]])
    edge_lst.append([rods[1].p_1,rods[2].p_2,E[10],N[10]])
    edge_lst.append([rods[1].p_1,rods[3].p_2,E[11],N[11]])
    edge_lst.append([rods[1].p_2,rods[5].p_1,E[12],N[12]])
    edge_lst.append([rods[1].p_2,rods[5].p_2,E[13],N[13]])
    edge_lst.append([rods[1].p_2,rods[2].p_2,E[14],N[14]])
    edge_lst.append([rods[1].p_2,rods[3].p_2,E[15],N[15]])
    edge_lst.append([rods[2].p_1,rods[4].p_1,E[16],N[16]])
    edge_lst.append([rods[2].p_1,rods[5].p_1,E[17],N[17]])
    edge_lst.append([rods[2].p_2,rods[4].p_1,E[18],N[18]])
    edge_lst.append([rods[2].p_2,rods[5].p_1,E[19],N[19]])
    edge_lst.append([rods[3].p_1,rods[4].p_2,E[20],N[20]])
    edge_lst.append([rods[3].p_1,rods[5].p_2,E[21],N[21]])
    edge_lst.append([rods[3].p_2,rods[4].p_2,E[22],N[22]])
    edge_lst.append([rods[3].p_2,rods[5].p_2,E[23],N[23]])
    return edge_lst
```