MAP 499 - Relationship between structural behavior and component coordination in bistable lattices

Hung Chu, Regan Stambaugh, Debdeep Bhattacharya

September 27, 2025

Abstract

Bistable lattices are topological models used to demonstrate the behavior of various materials. Our work uses combinatorial and variational approaches to assess the relationship between geometrical constraints and minimal-energy states of various bistable lattices including, but not limited to, triangular and Penrose lattices. This is useful in predicting the asymptotic response of an external energy source on a lattice structure given certain boundary conditions, rendering our results applicable to metamaterial design and understanding crystallization, among others.

Contents

1 Introduction 3

2 Bistability and Double-Well Potential 4
- 2.1 Energy and Force Functions . . . 4
- 2.2 Energy surface . . . 5

3 Minimization Schemes 6
- 3.1 Setting Up a Minimization Problem . . . 6
  - 3.1.1 Computing the Gradient . . . 6
  - 3.1.2 Computing the Hessian Matrix . . . 8
  - 3.1.3 Summary . . . 12
- 3.2 Gradient Descent . . . 13
  - 3.2.1 Overview . . . 13
  - 3.2.2 Demonstration . . . 14
- 3.3 Broyden-Fletcher-Goldfarb-Shanno (BFGS) . . . 16
  - 3.3.1 Overview . . . 16
  - 3.3.2 Demonstration . . . 17
- 3.4 Sequential Least Squares Quadratic Programming (SLSQP) . . . 20
  - 3.4.1 Overview . . . 20
- 3.5 Angle constraint on lattice nodes . . . 20
  - 3.5.1 Demonstration . . . 21
- 3.6 Dynamic Relaxation . . . 22
  - 3.6.1 Overview . . . 22
  - 3.6.2 Demonstration . . . 23

4 Quasistatic solution to boundary value problems 26

5 Conditions on Triangular Lattice Configurations 26

6 3D Printed Bistable Structure 33
- 6.1 Bistable Link . . . 33
- 6.2 Bistable Lattice . . . 35

7 Discussion 38
- 7.1 The Double-well Issue of a Bistable Edge . . . 38
- 7.2 The Restriction on Degrees of Freedom by Connectivity . . . 39
- 7.3 The Effect of Organization . . . 39
- 7.4 Existence of Non-zero Minimal-energy States . . . 41

8 Conclusion 43

1 Introduction

Bistable lattices are mathematical models that are used to demonstrate the behavior of various materials and processes. Structurally, a lattice is a collection of nodes connected by links, often in a uniform manner; common shapes are triangular, square, and hexagonal, as well as the penrose lattice. A lattice becomes bistable when its links have two equilibria, meaning the link is stable in both its short mode (when it has a displacement of 0) and its long mode (when it has a displacement of c). Thus, the lattice itself gains a certain flexibility and has more than one form for which it is stable.

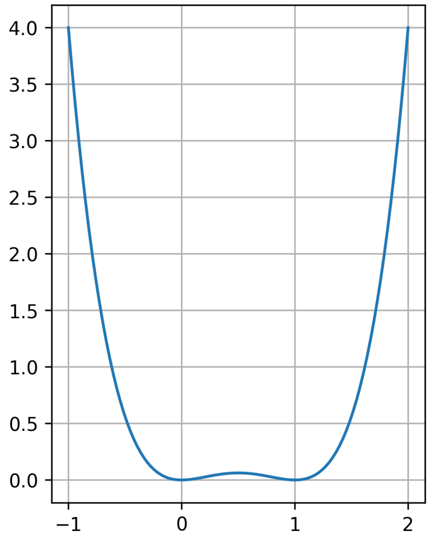
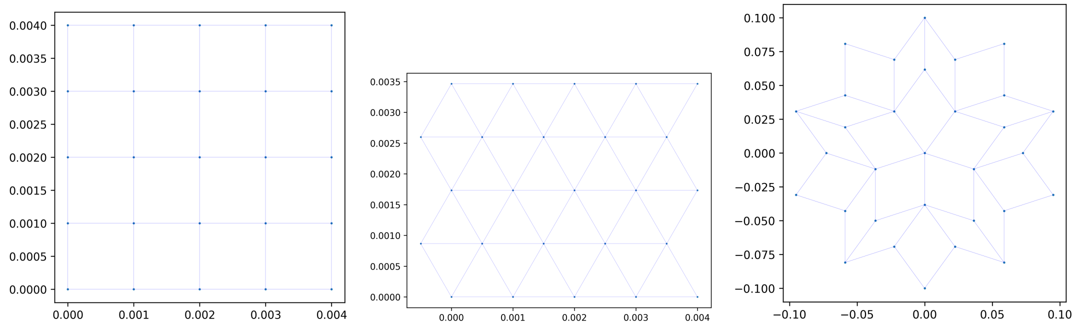


Figure 1: Example energy model of a bistable link.



(a) Square Lattice

(b) Triangular lattice

(c) Penrose Lattice

Figure 2: Example lattices

Bistable structures such as these are notable for their ability to withstand deformations, which makes bistable lattices excellent models for materials on a molecular level. For example, if we let each node of the lattice be a molecule, we can say that the material's solid state is represented when every link is in short mode, and its liquid state is represented when every link is long. Deformations of either of these states model phases of the transition from one state to the other. A similar but more interesting process that can be modeled by a bistable lattice is crystallization, the process by which solids with highly organized atoms form. When matter crystallizes, its atoms first deform and then settle into a stable state and form a crystalline structure. A bistable lattice model mimics this atomic behavior, first deforming and then approaching its closest configuration where all links are at one of their two equilibria. Studying and running simulations of this model can generate possible configurations, as well as exclude impossible ones.

Because of their bistable nature, deformations of a bistable lattice are also reversible, and a structure can return to its original form after a deformation without permanent damage [7]. This makes bistable lattices applicable when thinking of resilient structures, or structures well suited to withstand impact, as the lattice's bistable properties create stronger structures that are more likely to withstand an applied force. Metamaterials, materials manufactured to have properties that do not naturally occur, can be manufactured to behave like a bistable lattice on the molecular level. As a result, metamaterials are less likely to break under force. Studying bistable lattices as models for metamaterials predicts how such materials react to impact, insight useful for those seeking to maximize resiliency when creating and working with metamaterials. More applications of bistable lattices span industries, modeling electric, magnetic, and thermal fields, as well as the impact of mechanical force on the molecular level [7, 3].

It is also important to note that not all applications can be perfectly modeled by one lattice form; for example, in some applications, a bistable Penrose lattice such as that in Figure 2(c) may more accurately model the process being studied than the bistable triangular lattice (Figure 2(b)). It is valuable, too, to consider different lattice forms when examining resilient metamaterials. Not only is it worth examining which lattice form, if any, is the most resistant to force, but one can also investigate whether stronger structures can be created by splicing together triangular, square, and Penrose lattices. Thus, it is critical to develop an understanding of how multiple types of bistable structures work for those who seek to employ these models in their field.

Our study seeks to grow our understanding of these models by finding new ways to evaluate and assess bistable structures. Hooke's law tells us that force is the derivative of energy with respect to time; thus, any constant force added to a lattice increases the total energy of a lattice, which is equal to the sum of the energy stored in each link. In this paper, we first walk through the mathematics required to set up an energy minimization problem and discuss various numerical minimization schemes that allow us to identify a lattice's minimal energy state. We then focus on the geometry of a triangular bistable lattice and the consequences of geometrical constraints on certain minimal energy states (still states), and finally we present 3-dimensional models of a bistable lattice that tangibly visualize our work.

2 Bistability and Double-Well Potential

Because the total energy of a bistable lattice is the sum of the energy of its links, it is important to define the energy of each link. Due to its bistable nature, we say that each link has 0 energy in both its short and long mode. These characteristics can be modeled by a quartic equation [2] and create double-well potential for each link. We assume that each link has identical properties, and thus the sum of these expressions for each link is equal to the lattice's total energy.

Our model is adapted from the quartic presented in Cherkaev's paper [2], with a few notable differences. Our expression takes relative strain on the link as its input rather than length, which not only shifts the graph so that the first minimum is at zero, but neutralizes the difference in impact that a deformation has on a larger lattice versus a smaller one. This allows us to generalize the model to lattices of all sizes. In addition, we define our strain by the displacement of the nodes instead of the length of the link, which is a simpler change to quantify.

2.1 Energy and Force Functions

Define a function,

$$h_c(r) = r^2(r-c)^2,$$

to give the energy of a bistable link at a given strain r such that the link's equilibria occur at $r = 0$ and $r = c$. Then the force on the link is given by

$$h_c'(r) = 2r(r-c)^2 + 2r^2(r-c).$$

For a link between nodes i and j, define the two-point strain as

$$s_{ij} = \frac{|(\mathbf{x}_i + \mathbf{u}_i) - (\mathbf{x}_j + \mathbf{u}_j)| - |\mathbf{x}_i - \mathbf{x}_j|}{|\mathbf{x}_i - \mathbf{x}_j|}.$$

Notice that $|(\mathbf{x}_i + \mathbf{u}_i) - (\mathbf{x}_j + \mathbf{u}_j)|$ represents the length of the link after the ith and jth nodes are displaced by $\mathbf{u}_i$ and $\mathbf{u}_j$ respectively, and $|\mathbf{x}_i - \mathbf{x}_j|$ represents the link's original length at its short equilibrium.

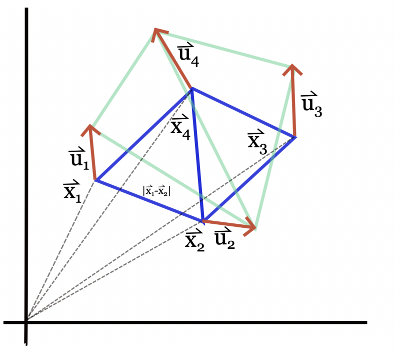


Figure 3: Example of a lattice. Note that $\mathbf{x}_i$ is the vector originating at the origin and ending at the initial position of node i, and the vector $\mathbf{u}_i$ is the vector originating at the initial position of node i and ending at its displaced position.

Now define $\mathbf{u}_i$ as the displacement of the ith node,

$$\mathbf{u}_i = \begin{bmatrix} u_{i1} \\ u_{i2} \end{bmatrix}.$$

If there are n nodes in the lattice, define $\mathbf{U}$ to be the vector

$$\mathbf{U} = \begin{bmatrix} u_{11} \\ u_{12} \\ \vdots \\ u_{n1} \\ u_{n2} \end{bmatrix}$$

which represents the displacement of every node in the lattice. With this vector, we can construct a function $E(\mathbf{U})$ that takes as an input the displacement of every node and yields the energy of the lattice resulting from the strain on each link. If L is the set of all links, then we have

$$E(\mathbf{U}) = \sum_{(i,j)\in L} h(s_{ij})$$

as the total energy in the lattice.

In order to understand and analyze this model, it is important to have a solid comprehension of how lattices work. It is established that the total energy of a lattice is dependent on the displacement of each node, or the elongation of each link. Over time, a deformed lattice will approach a configuration where its total energy is minimized. These are called **minimal energy states** or **minimal energy configurations**, and we would expect the lattice's total energy to be equal to zero. In this case - when a lattice is in a minimal energy state and its total energy is 0 - the configuration is called a **still state** or a **still configuration.** Since a lattice in a still state has 0 energy, all of its links are either in short mode or long mode by necessity. Based on the theoretical overview, we attempt to visualize and assess all possible minimal energy states of a given lattice with the help of computer simulation. In order to do so, we turn this research question into an optimization problem, minimizing the total energy of the lattice with respect to displacement.

2.2 Energy surface

We consider a triangle consisting of 3 bistable links. Assume that two of the three nodes are clamped at its reference position, i.e. one link is fixed at its short mode. The third node is free. Fig. 4 shows

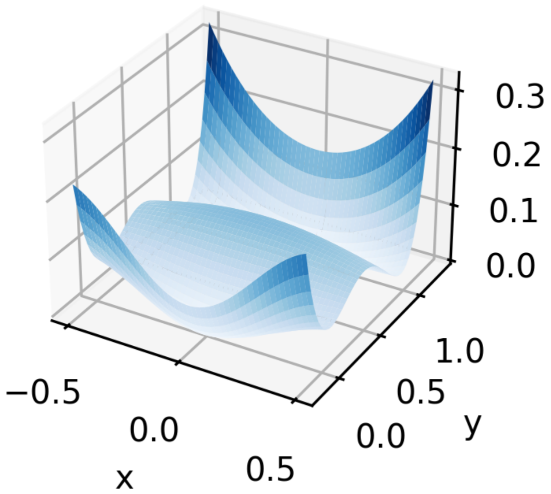


Figure 4: Energy surface as a function of the x and y displacement of the free vertex of a triangle.

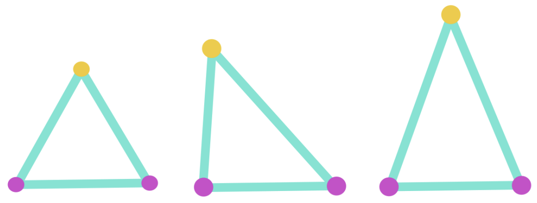


Figure 5: Possible zero-energy configurations of a triangle with three bistable links.

the energy surface as a function of the x and y displacement of the third node, while Figure 5 shows possible zero-energy configurations of this lattice. Notice that Figure 4 has only two visible zero-energy points visualized, but there are two others located outside the range of this plot, representing the middle configuration of 5 and its mirror image.

3 Minimization Schemes

3.1 Setting Up a Minimization Problem

Numerical minimization schemes demand either the energy gradient, the Hessian matrix, or both. Below are detailed computations of each of these.

The minimization problem can be described as

$$\min_{\mathbf{U}\in R^{2\times n}} E(\mathbf{U}) \tag{1}$$

subject to constraints that are described in Section 3.5.

First, we consider the unconstrained problem.

3.1.1 Computing the Gradient

To solve the minimization problem, we first compute $\frac{\partial}{\partial u_{i,1}}E(\mathbf{U})$ and $\frac{\partial}{\partial u_{i,2}}E(\mathbf{U})$. Fix i, and consider first $\frac{\partial}{\partial u_{i,1}}E(\mathbf{U})$. The term $u_{i,1}$ does not appear in some term in the above summation, and thus the derivative of these term with respect to $u_{i,1}$ will be 0. To denote the terms with $u_{i,1}$ (where the derivative is not 0), let Ω_i be the set of nodes that are neighbors of i such that the energy in the lattice dependent on node i is

$$E_i(\mathbf{U}) = \sum_{j\in\Omega_i} h(s_{ij}).$$

The derivative with respect to $u_{i,1}$, given by the chain rule, is

$$\begin{aligned}\frac{\partial}{\partial u_{i,1}} E_i(\mathbf{U}) &= \sum_{j\in\Omega_i} \frac{\partial}{\partial u_{i,1}} h(s_{ij}) \\ &= \sum_{j\in\Omega_i} h'(s_{ij}) \cdot \frac{\partial}{\partial u_{i,1}} s_{ij} \\ &= \sum_{j\in\Omega_i} h'(s_{ij}) \cdot \frac{\partial}{\partial u_{i,1}} \left(\frac{|(\mathbf{x}_i + \mathbf{u}_i) - (\mathbf{x}_j + \mathbf{u}_j)| - |\mathbf{x}_i - \mathbf{x}_j|}{|\mathbf{x}_i - \mathbf{x}_j|} \right).\end{aligned}$$

To derive the vectorized term with respect to a component of one of the vectors, it is necessary to break the vector into its components. Since $|\mathbf{x}_i - \mathbf{x}_j|$ is constant with respect to $u_{i,1}$, the derivation[1] comes to

$$\begin{aligned}&\frac{\partial}{\partial u_{i,1}} \left(\frac{|(\mathbf{x}_i + \mathbf{u}_i) - (\mathbf{x}_j + \mathbf{u}_j)| - |\mathbf{x}_i - \mathbf{x}_j|}{|\mathbf{x}_i - \mathbf{x}_j|} \right) \\ &= \frac{\partial}{\partial u_{i,1}} \left(\frac{|(\mathbf{x}_i + \mathbf{u}_i) - (\mathbf{x}_j + \mathbf{u}_j)|}{|\mathbf{x}_i - \mathbf{x}_j|} \right) + 0 \\ &= \frac{1}{|\mathbf{x}_i - \mathbf{x}_j|} \cdot \frac{\partial}{\partial u_{i,1}} \left(\sqrt{((x_{i,1} + u_{i,1}) - (x_{j,1} + u_{j,1}))^2 + ((x_{i,2} + u_{i,2}) - (x_{j,2} + u_{j,2}))^2} \right) \\ &= \frac{1}{|\mathbf{x}_i - \mathbf{x}_j|} \cdot \frac{2((x_{i,1} + u_{i,1}) - (x_{j,1} + u_{j,1})) \cdot 1}{2\sqrt{((x_{i,1} + u_{i,1}) - (x_{j,1} + u_{j,1}))^2 + ((x_{i,2} + u_{i,2}) - (x_{j,2} + u_{j,2}))^2}} \\ &= \frac{1}{|\mathbf{x}_i - \mathbf{x}_j|} \cdot \left(\frac{(x_{i,1} + u_{i,1}) - (x_{j,1} + u_{j,1})}{|(\mathbf{x}_i + \mathbf{u}_i) - (\mathbf{x}_j + \mathbf{u}_j)|} \right),\end{aligned}$$

and thus

$$\frac{\partial}{\partial u_{i,1}} E_i(\mathbf{U}) = \sum_{j\in\Omega_i} h'(s_{ij}) \cdot \frac{(x_{i,1} + u_{i,1}) - (x_{j,1} + u_{j,1})}{|(\mathbf{x}_i + \mathbf{u}_i) - (\mathbf{x}_j + \mathbf{u}_j)| \cdot |\mathbf{x}_i - \mathbf{x}_j|}.$$

Similarly, the derivative with respect to $u_{i,2}$ comes to

$$\frac{\partial}{\partial u_{i,2}} E_i(\mathbf{U}) = \sum_{j\in\Omega_i} h'(s_{ij}) \cdot \frac{(x_{i,2} + u_{i,2}) - (x_{j,2} + u_{j,2})}{|(\mathbf{x}_i + \mathbf{u}_i) - (\mathbf{x}_j + \mathbf{u}_j)| \cdot |\mathbf{x}_i - \mathbf{x}_j|}.$$

These quantities are used to define the gradient of our energy function:

$$\nabla E(\mathbf{U}) = \begin{bmatrix} \frac{\partial}{\partial u_{1,1}} E(\mathbf{U}) \\ \frac{\partial}{\partial u_{1,2}} E(\mathbf{U}) \\ \frac{\partial}{\partial u_{2,1}} E(\mathbf{U}) \\ \frac{\partial}{\partial u_{2,2}} E(\mathbf{U}) \\ \vdots \\ \frac{\partial}{\partial u_{n,1}} E(\mathbf{U}) \\ \frac{\partial}{\partial u_{n,2}} E(\mathbf{U}) \end{bmatrix} (4).$$

[1] It is important to understand the mathematics of this, as the simplified version of $\partial/\partial u_{i,1} h(s_{ij})$ is taken for granted when computing the second derivatives.

3.1.2 Computing the Hessian Matrix

To be able to use further numerical minimization techniques, we need to next find the matrix of second derivatives, or the Hessian matrix. If there are n nodes in the matrix, then the matrix will look like

$$\nabla^2 E(U) = \begin{bmatrix} \frac{\partial^2 E}{\partial u_{1,1}\partial u_{1,1}} & \frac{\partial^2 E}{\partial u_{1,1}\partial u_{2,1}} & \cdots & \frac{\partial^2 E}{\partial u_{1,1}\partial u_{n,1}} & \frac{\partial^2 E}{\partial u_{1,1}\partial u_{1,2}} & \cdots & \frac{\partial^2 E}{\partial u_{1,1}\partial u_{n,2}} \\ \frac{\partial^2 E}{\partial u_{2,1}\partial u_{1,1}} & \frac{\partial^2 E}{\partial u_{2,1}\partial u_{2,1}} & \cdots & \frac{\partial^2 E}{\partial u_{2,1}\partial u_{n,1}} & \frac{\partial^2 E}{\partial u_{2,1}\partial u_{1,2}} & \cdots & \frac{\partial^2 E}{\partial u_{2,1}\partial u_{n,2}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial^2 E}{\partial u_{n,1}\partial u_{1,1}} & \frac{\partial^2 E}{\partial u_{n,1}\partial u_{2,1}} & \cdots & \frac{\partial^2 E}{\partial u_{n,1}\partial u_{n,1}} & \frac{\partial^2 E}{\partial u_{n,1}\partial u_{1,2}} & \cdots & \frac{\partial^2 E}{\partial u_{n,1}\partial u_{n,2}} \\ \frac{\partial^2 E}{\partial u_{1,2}\partial u_{1,1}} & \frac{\partial^2 E}{\partial u_{1,2}\partial u_{2,1}} & \cdots & \frac{\partial^2 E}{\partial u_{1,2}\partial u_{n,1}} & \frac{\partial^2 E}{\partial u_{1,2}\partial u_{1,2}} & \cdots & \frac{\partial^2 E}{\partial u_{1,2}\partial u_{n,2}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial^2 E}{\partial u_{n,2}\partial u_{1,1}} & \frac{\partial^2 E}{\partial u_{n,2}\partial u_{2,1}} & \cdots & \frac{\partial^2 E}{\partial u_{n,2}\partial u_{n,1}} & \frac{\partial^2 E}{\partial u_{n,2}\partial u_{1,2}} & \cdots & \frac{\partial^2 E}{\partial u_{n,2}\partial u_{n,2}} \end{bmatrix} (5).$$

Ultimately, this means we need to find four different families of second derivatives:

$$\frac{\partial^2 E}{\partial u_{i,1}\partial u_{j,1}} \qquad \frac{\partial^2 E}{\partial u_{i,2}\partial u_{j,1}} \qquad \frac{\partial^2 E}{\partial u_{i,1}\partial u_{j,2}} \qquad \frac{\partial^2 E}{\partial u_{i,2}\partial u_{j,2}}$$

For any given pair of nodes i and j, it is necessary to know whether i and j are neighbors or not to find these second derivatives. First, suppose i and j are not neighbors. Then,

$$\begin{aligned} \frac{\partial^2 E}{\partial u_{i,1}\partial u_{j,1}} &= \frac{\partial}{\partial u_{i,1}}\left(\frac{\partial E}{\partial u_{j,1}}\right) \\ &= \frac{\partial}{\partial u_{i,1}}\left(\sum_{p\in\Omega_j} h'(s_{jp}) \cdot \frac{(x_{j,1}+u_{j,1})-(x_{p,1}-u_{p,1})}{|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_p-\mathbf{u}_p)|\cdot|\mathbf{x}_j-\mathbf{x}_p|}\right). \end{aligned}$$

Because $i \notin \Omega_j$, there are no terms in the above sum that contain $u_{i,1}$, and the entire expression is constant with respect to $u_{i,1}$. Thus, if i and j are not neighbors, we have that

$$\begin{aligned} \frac{\partial^2 E}{\partial u_{i,1}\partial u_{j,1}} &= 0 \\ \frac{\partial^2 E}{\partial u_{i,2}\partial u_{j,1}} &= 0 \\ \frac{\partial^2 E}{\partial u_{i,1}\partial u_{j,2}} &= 0 \\ \frac{\partial^2 E}{\partial u_{i,2}\partial u_{j,2}} &= 0. \end{aligned}$$

Now suppose that i and j are neighbors, and consider $\frac{\partial^2 E}{\partial u_{i,1}\partial u_{j,1}}$. In the expression

$$\frac{\partial}{\partial u_{i,1}}\left(\sum_{p\in\Omega_j} h'(s_{jp}) \cdot \frac{(x_{j,1}+u_{j,1})-(x_{p,1}+u_{p,1})}{|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_p+\mathbf{u}_p)|\cdot|\mathbf{x}_j-\mathbf{x}_p|}\right),$$

the variable $u_{i,1}$ appears in only one term of the summation. Every other term in the summation is constant with respect to $u_{i,1}$, so we can focus on the following derivation,

$$\frac{\partial}{\partial u_{i,1}}\left(h'(s_{ji}) \cdot \frac{(x_{j,1}+u_{j,1})-(x_{i,1}+u_{i,1})}{|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_i+\mathbf{u}_i)|\cdot|\mathbf{x}_j-\mathbf{x}_i|}\right).$$

The derivative can be written generally using the product rule[2]:

$$\begin{aligned}
&\frac{\partial}{\partial u_{i,1}}\left(h'(s_{ji})\cdot\frac{(x_{j,1}+u_{j,1})-(x_{i,1}+u_{i,1})}{|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_i+\mathbf{u}_i)|\cdot|\mathbf{x}_j-\mathbf{x}_i|}\right)\\
&=h'(s_{ji})\cdot\frac{\partial}{\partial u_{i,1}}\left(\frac{(x_{j,1}+u_{j,1})-(x_{i,1}+u_{i,1})}{|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_i+\mathbf{u}_i)|\cdot|\mathbf{x}_j-\mathbf{x}_i|}\right)+\frac{\partial}{\partial u_{i,1}}\big(h'(s_{ji})\big)\cdot\frac{(x_{j,1}+u_{j,1})-(x_{i,1}+u_{i,1})}{|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_i+\mathbf{u}_i)|\cdot|\mathbf{x}_j-\mathbf{x}_i|}\\
&=\frac{h'(s_{ji})}{|\mathbf{x}_j-\mathbf{x}_i|}\cdot\frac{\partial}{\partial u_{i,1}}\left(\frac{(x_{j,1}+u_{j,1})-(x_{i,1}+u_{i,1})}{|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_i+\mathbf{u}_i)|}\right)+\frac{\partial}{\partial u_{i,1}}\big(h'(s_{ji})\big)\cdot\frac{(x_{j,1}+u_{j,1})-(x_{i,1}+u_{i,1})}{|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_i+\mathbf{u}_i)|\cdot|\mathbf{x}_j-\mathbf{x}_i|}
\end{aligned}$$

Computing the unknown derivatives within the above expression first requires the chain rule[3] to get

$$\begin{aligned}
\frac{\partial}{\partial u_{i,1}}\big(h'(s_{ji})\big)&=h''(s_{ji})\cdot\frac{\partial}{\partial u_{i,1}}(s_{ji})\\
&=\left(\frac{|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_i+\mathbf{u}_i)|-|\mathbf{x}_j-\mathbf{x}_i|}{|\mathbf{x}_j-\mathbf{x}_i|}\right)\\
&=h''(s_{ji})\left(-\frac{(x_{j,1}+u_{j,1})-(x_{i,1}+u_{i,1})}{|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_i+\mathbf{u}_i)|\cdot|\mathbf{x}_j-\mathbf{x}_i|}\right).
\end{aligned}$$

Next, let $a=(x_{j,1}+u_{j,1})-(x_{i,1}+u_{i,1})$ and $b=|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_i+\mathbf{u}_i)|)$, and use the quotient rule:

$$\begin{aligned}
\frac{\partial}{\partial u_{i,1}}\left(\frac{(x_{j,1}+u_{j,1})-(x_{i,1}+u_{i,1})}{|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_i+\mathbf{u}_i)|}\right)&=\frac{\partial}{\partial u_{i,1}}\left(\frac{a}{b}\right)\\
&=\frac{b\cdot\frac{\partial}{\partial u_{i,1}}(a)-a\cdot\frac{\partial}{\partial u_{i,1}}(b)}{b^2}.
\end{aligned}$$

Note that[4]

$$\begin{aligned}
\frac{\partial}{\partial u_{i,1}}(a)&=\frac{\partial}{\partial u_{i,1}}\big((x_{j,1}+u_{j,1})-(x_{i,1}+u_{i,1})\big)=-1\qquad\text{and}\\
\frac{\partial}{\partial u_{i,1}}(b)&=\frac{\partial}{\partial u_{i,1}}|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_i+\mathbf{u}_i)|=-\frac{(x_{j,1}+u_{j,1})-(x_{i,1}+u_{i,1})}{|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_i+\mathbf{u}_i)|}.
\end{aligned}$$

So,

$$\begin{aligned}
\frac{b\cdot\frac{\partial}{\partial u_{i,1}}(a)-a\cdot\frac{\partial}{\partial u_{i,1}}(b)}{b^2}&=\frac{-|(\mathbf{x}_i+\mathbf{u}_i)-(\mathbf{x}_j+\mathbf{u}_j)|+\big((x_{j,1}+u_{j,1})-(x_{i,1}+u_{i,1})\big)^2/|(\mathbf{x}_i+\mathbf{u}_i)-(\mathbf{x}_j+\mathbf{u}_j)|}{|(\mathbf{x}_i+\mathbf{u}_i)-(\mathbf{x}_j+\mathbf{u}_j)|^2}\\
&=\left(\frac{-1}{|(\mathbf{x}_i+\mathbf{u}_i)-(\mathbf{x}_j+\mathbf{u}_j)|}+\frac{\big((x_{j,1}+u_{j,1})-(x_{i,1}+u_{i,1})\big)^2}{|(\mathbf{x}_i+\mathbf{u}_i)-(\mathbf{x}_j+\mathbf{u}_j)|^3}\right).
\end{aligned}$$

Finally, we can substitute to find $\partial^2E/\partial u_{i,1}\partial u_{j,1}$:

$$\begin{aligned}
&\frac{h'(s_{ji})}{|\mathbf{x}_j-\mathbf{x}_i|}\cdot\frac{\partial}{\partial u_{i,1}}\left(\frac{(x_{j,1}+u_{j,1})-(x_{i,1}+u_{i,1})}{|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_i+\mathbf{u}_i)|}\right)+\frac{\partial}{\partial u_{i,1}}\big(h'(s_{ji})\big)\cdot\frac{(x_{j,1}+u_{j,1})-(x_{i,1}+u_{i,1})}{|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_i+\mathbf{u}_i)|\cdot|\mathbf{x}_j-\mathbf{x}_i|}\\
&=\frac{h'(s_{ji})}{|\mathbf{x}_j-\mathbf{x}_i|}\left(\frac{-1}{|(\mathbf{x}_i+\mathbf{u}_i)-(\mathbf{x}_j+\mathbf{u}_j)|}+\frac{\big((x_{j,1}+u_{j,1})-(x_{i,1}+u_{i,1})\big)^2}{|(\mathbf{x}_i+\mathbf{u}_i)-(\mathbf{x}_j+\mathbf{u}_j)|^3}\right)\\
&\qquad+\left(-h''(s_{ji})\cdot\frac{(x_{j,1}+u_{j,1})-(x_{i,1}+u_{i,1})}{|(\mathbf{x}_i+\mathbf{u}_i)-(\mathbf{x}_j+\mathbf{u}_j)|\cdot|\mathbf{x}_i-\mathbf{x}_j|}\right)\left(\frac{(x_{j,1}+u_{j,1})-(x_{i,1}+u_{i,1})}{|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_i+\mathbf{u}_i)|\cdot|\mathbf{x}_j-\mathbf{x}_i|}\right)\qquad.
\end{aligned}$$

[2] Factoring out $1/|\mathbf{x}_j-\mathbf{x}_i|$ from the derivative in the first term makes the derivation simpler in later steps

[3] Notice that the derivative of s_{ji} with respect to $u_{i,1}$ is identical to $\partial/\partial u_{i,1}(s_{ij})$ except for its sign, since $\partial/\partial u_{i,1}\big((x_{j,1}+u_{j,1})-(x_{i,1}+u_{i,1})\big)=-1$.

[4] The derivation of $\partial/\partial u_{i,1}(b)$ is the same as $\partial/\partial u_{j,1}(s_{ji})$, without constants and coefficients, but negative, since $\partial/\partial u_{i,1}\big((x_{j,1}+u_{j,1})-(x_{i,1}+u_{i,1})\big)=-1$.

Simplification allows us to conclude that, if nodes i and j are neighbors,

$$\frac{\partial^2 E}{\partial u_{i,1}\partial u_{j,1}} = \frac{\big((x_{j,1}+u_{j,1})-(x_{i,1}+u_{i,1})\big)^2}{|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_i+\mathbf{u}_i)|^2|\mathbf{x}_j-\mathbf{x}_i|}\left(\frac{h'(s_{ji})}{|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_i+\mathbf{u}_i)|}-\frac{h''(s_{ji})}{|\mathbf{x}_j-\mathbf{x}_i|}\right)-\frac{h'(s_{ji})}{|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_i+\mathbf{u}_i)||\mathbf{x}_j-\mathbf{x}_i|}.$$

Similarly,

$$\frac{\partial^2 E}{\partial u_{i,2}\partial u_{j,2}} = \frac{\big((x_{j,2}+u_{j,2})-(x_{i,2}+u_{i,2})\big)^2}{|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_i+\mathbf{u}_i)|^2|\mathbf{x}_j-\mathbf{x}_i|}\left(\frac{h'(s_{ji})}{|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_i+\mathbf{u}_i)|}-\frac{h''(s_{ji})}{|\mathbf{x}_j-\mathbf{x}_i|}\right)-\frac{h'(s_{ji})}{|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_i+\mathbf{u}_i)||\mathbf{x}_j-\mathbf{x}_i|}$$

given the same conditions. A similar process can be followed to find $\frac{\partial^2 E}{\partial u_{i,1}\partial u_{j,2}}$, since

$$\frac{\partial^2 E}{\partial u_{i,1}\partial u_{j,2}} = \frac{\partial}{\partial u_{i,1}}\left(\sum_{p\in\Omega_j} h'(s_{jp})\cdot\frac{(x_{j,2}+u_{j,2})-(x_{p,2}+u_{p,2})}{|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_p+\mathbf{u}_p)|\cdot|\mathbf{x}_j-\mathbf{x}_p|}\right).$$

For the same reasons as above, we only need to derive one term within the summation, so once again the product rule gives us

$$\begin{aligned}&\frac{\partial}{\partial u_{i,1}}\left(h'(s_{jp})\cdot\frac{(x_{j,2}+u_{j,2})-(x_{i,2}+u_{i,2})}{|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_i+\mathbf{u}_i)|\cdot|\mathbf{x}_j-\mathbf{x}_i|}\right)\\ &=\frac{h'(s_{ji})}{|\mathbf{x}_j-\mathbf{x}_i|}\cdot\frac{\partial}{\partial u_{i,1}}\left(\frac{(x_{j,2}+u_{j,2})-(x_{i,2}+u_{i,2})}{|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_i+\mathbf{u}_i)|}\right)+\frac{\partial}{\partial u_{i,1}}\big(h'(s_{ji})\big)\cdot\frac{(x_{j,2}+u_{j,2})-(x_{i,2}+u_{i,2})}{|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_i+\mathbf{u}_i)|\cdot|\mathbf{x}_j-\mathbf{x}_i|}\end{aligned}$$

Again, we want to compute the two unknown derivatives in the expression above. We know from earlier that

$$\frac{\partial}{\partial u_{i,1}}\big(h'(s_{ji})\big) = h''(s_{ji})\left(-\frac{(x_{j,1}+u_{j,1})-(x_{i,1}+u_{i,1})}{|(\mathbf{x}_i+\mathbf{u}_i)-(\mathbf{x}_j+\mathbf{u}_j)|\cdot|\mathbf{x}_i-\mathbf{x}_j|}\right),$$

so we just need to solve for $\frac{\partial}{\partial u_{i,1}}\left(\frac{(x_{j,1}+u_{j,1})-(x_{i,1}+u_{i,1})}{|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_i+\mathbf{u}_i)|}\right)$. Since the numerator is a constant coefficient, we can rewrite this as

$$\frac{\partial}{\partial u_{i,1}}\left(\frac{(x_{j,2}+u_{j,2})-(x_{i,2}+u_{i,2})}{|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_i+\mathbf{u}_i)|}\right) = \big((x_{j,2}+u_{j,2})-(x_{i,2}+u_{i,2})\big)\cdot\frac{\partial}{\partial u_{i,1}}\Big(|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_i+\mathbf{u}_i)|\Big)^{-1}$$

and focus on differentiating the vectorized term:

$$\begin{aligned}&\frac{\partial}{\partial u_{i,1}}\Big(|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_i+\mathbf{u}_i)|\Big)^{-1}\\ &=\frac{\partial}{\partial u_{i,1}}\Big(\big((x_{j,1}+u_{j,1})-(x_{i,1}+u_{i,1})\big)^2+\big((x_{j,2}+u_{j,2})-(x_{i,2}+u_{i,2})\big)^2\Big)^{-1/2}\\ &=-\frac{1}{2}\Big(\big((x_{j,1}+u_{j,1})-(x_{i,1}+u_{i,1})\big)^2+\big((x_{j,2}+u_{j,2})-(x_{i,2}+u_{i,2})\big)^2\Big)^{-3/2}(2)\big((x_{j,1}+u_{j,1})-(x_{i,1}+u_{i,1})\big)(-1)\\ &=-\frac{1}{2}\Big(|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_i+\mathbf{u}_i)|^2\Big)^{-3/2}(-2)\big((x_{j,1}+u_{j,1})-(x_{i,1}+u_{i,1})\big)\\ &=\frac{(x_{j,1}+u_{j,1})-(x_{i,1}+u_{i,1})}{|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_i+\mathbf{u}_i)|^3}.\end{aligned}$$

Now we can make the substitutions necessary to find $\frac{\partial^2 E}{\partial u_{i,1}\partial u_{j,2}}$:

$$\begin{aligned}&\frac{h'(s_{ji})}{|\mathbf{x}_j-\mathbf{x}_i|}\cdot\frac{\partial}{\partial u_{i,1}}\left(\frac{(x_{j,2}+u_{j,2})-(x_{i,2}+u_{i,2})}{|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_i+\mathbf{u}_i)|}\right)+\frac{\partial}{\partial u_{i,1}}\big(h'(s_{ji})\big)\cdot\frac{(x_{j,2}+u_{j,2})-(x_{i,2}+u_{i,2})}{|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_i+\mathbf{u}_i)|\cdot|\mathbf{x}_j-\mathbf{x}_i|}\\ &\quad=\frac{h'(s_{ji})}{|\mathbf{x}_j-\mathbf{x}_i|}\left(\frac{\big((x_{j,2}+u_{j,2})-(x_{i,2}-u_{i,2})\big)\big((x_{j,1}+u_{j,1})-(x_{i,1}+u_{i,1})\big)}{|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_i+\mathbf{u}_i)|^3}\right)\\ &\qquad+\left(-h''(s_{ji})\frac{(x_{j,1}+u_{j,1})-(x_{i,1}+u_{i,1})}{|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_i+\mathbf{u}_i)|\cdot|\mathbf{x}_j-\mathbf{x}_i|}\right)\left(\frac{(x_{j,2}+u_{j,2})-(x_{i,2}+u_{i,2})}{|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_i+\mathbf{u}_i)|\cdot|\mathbf{x}_j-\mathbf{x}_i|}\right).\end{aligned}$$

Simplifying allows us to conclude that

$$\frac{\partial^2 E}{\partial u_{i,1}\partial u_{j,2}} = \frac{\big((x_{j,1}+u_{j,1})-(x_{i,1}+u_{i,1})\big)\big((x_{j,2}+u_{j,2})-(x_{i,2}+u_{i,2})\big)}{|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_i+\mathbf{u}_i)|^2|\mathbf{x}_j-\mathbf{x}_i|}\left(\frac{h'(s_{ji})}{|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_i+\mathbf{u}_i)|}-\frac{h''(s_{ji})}{|\mathbf{x}_j-\mathbf{x}_i|}\right),$$

and similarly,

$$\frac{\partial^2 E}{\partial u_{i,2}\partial u_{j,1}} = \frac{\big((x_{j,1}+u_{j,1})-(x_{i,1}+u_{i,1})\big)\big((x_{j,2}+u_{j,2})-(x_{i,2}+u_{i,2})\big)}{|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_i+\mathbf{u}_i)|^2|\mathbf{x}_j-\mathbf{x}_i|}\left(\frac{h'(s_{ji})}{|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_i+\mathbf{u}_i)|}-\frac{h''(s_{ji})}{|\mathbf{x}_j-\mathbf{x}_i|}\right).$$

Finally, suppose $i = j$, and our second derivatives are

$$\frac{\partial^2 E}{\partial u_{i,1}^2}, \qquad \frac{\partial^2 E}{\partial u_{i,2}\partial u_{i,1}}, \qquad \frac{\partial^2 E}{\partial u_{i,1}\partial u_{i,2}}, \qquad \text{and} \qquad \frac{\partial^2 E}{\partial u_{i,2}^2}.$$

Like before, we'll focus first on $\partial^2 E/\partial u_{i,1}^2$,

$$\begin{aligned}\frac{\partial^2 E}{\partial u_{i,1}^2} &= \frac{\partial}{\partial u_{i,1}}\left(\frac{\partial E}{\partial u_{i,1}}\right)\\ &= \frac{\partial}{\partial u_{i,1}}\left(\sum_{p\in\Omega_i} h'(s_{ip})\frac{(x_{i,1}+u_{i,1})-(x_{p,1}+u_{p,1})}{|(\mathbf{x}_i+\mathbf{u}_i)-(\mathbf{x}_p+\mathbf{u}_p)|\cdot|\mathbf{x}_i-\mathbf{x}_p|}\right)\end{aligned}$$

A crucial difference here from before is that $u_{i,1}$ is contained in every term of the summation. Thus, the second derivative will be a sum itself. So, for every $p \in \Omega_i$, the product rule provides

$$\begin{aligned}&\frac{\partial}{\partial u_{i,1}}\left(h'(s_{ip})\frac{(x_{i,1}+u_{i,1})-(x_{p,1}+u_{p,1})}{|(\mathbf{x}_i+\mathbf{u}_i)-(\mathbf{x}_p+\mathbf{u}_p)|\cdot|\mathbf{x}_i-\mathbf{x}_p|}\right)\\ &= h'(s_{ip})\cdot\frac{\partial}{\partial u_{i,1}}\left(\frac{(x_{i,1}+u_{i,1})-(x_{p,1}+u_{p,1})}{|(\mathbf{x}_i+\mathbf{u}_i)-(\mathbf{x}_p+\mathbf{u}_p)|\cdot|\mathbf{x}_i-\mathbf{x}_p|}\right)+\frac{\partial}{\partial u_{i,1}}h'(s_{ip})\cdot\frac{(x_{i,1}+u_{i,1})-(x_{p,1}+u_{p,1})}{|(\mathbf{x}_i+\mathbf{u}_i)-(\mathbf{x}_p+\mathbf{u}_p)|\cdot|\mathbf{x}_i-\mathbf{x}_p|}\end{aligned}$$

To find the two unknowns in the expression above, we follow similar processes as before. We can find $\partial/\partial u_{i,1}\big(h'(s_{ip})\big)$ using the chain rule -

$$\begin{aligned}\frac{\partial}{\partial u_{i,1}}h'(s_{ip}) &= h''(s_{ip})\cdot\frac{\partial}{\partial u_{i,1}}(s_{ip})\\ &= h''(s_{ip})\left(\frac{(x_{i,1}+u_{i,1})-(x_{p,1}+u_{p,1})}{|(\mathbf{x}_i+\mathbf{u}_i)-(\mathbf{x}_p+\mathbf{u}_p)||\mathbf{x}_i-\mathbf{x}_p|}\right)\end{aligned}$$

- and we can find the other using the quotient rule. As we did before, let $a = (x_{i,1}+u_{i,1})-(x_{p1}+u_{p1})$ and $b = |(\mathbf{x}_i+\mathbf{u}_i)-(\mathbf{x}_p+\mathbf{u}_p)|$ such that

$$\begin{aligned}&\frac{\partial}{\partial u_{i,1}}a = 1\\ &\frac{\partial}{\partial u_{i,1}}b = \frac{(x_{i,1}+u_{i,1})-(x_{p,1}+u_{p,1})}{|(\mathbf{x}_i+\mathbf{u}_i)-(\mathbf{x}_p+\mathbf{u}_p)|}.\end{aligned}$$

By the quotient rule,

$$\begin{aligned}&\frac{\partial}{\partial u_{i,1}}\left(\frac{(x_{i,1}+u_{i,1})-(x_{p,1}+u_{p,1})}{|(\mathbf{x}_i+\mathbf{u}_i)-(\mathbf{x}_p+\mathbf{u}_p)|}\right)\\ &= \frac{1|(\mathbf{x}_i+\mathbf{u}_i)-(\mathbf{x}_p+\mathbf{u}_p)|-\big((x_{i,1}+u_{i,1})-(x_{p,1}+u_{p,1})\big)^2/|(\mathbf{x}_i+\mathbf{u}_i)-(\mathbf{x}_p+\mathbf{u}_p)|}{|(\mathbf{x}_i+\mathbf{u}_i)-(\mathbf{x}_p+\mathbf{u}_p)|^2}.\end{aligned}$$

Simplifying this term yields the following result:

$$\frac{\partial^2 E}{\partial u_{i,1}^2} = \frac{\big((x_{i,1}+u_{i,1})-(x_{p,1}+u_{p,1})\big)^2}{|(\mathbf{x}_i+\mathbf{u}_i)-(\mathbf{x}_p+\mathbf{u}_p)|^2|\mathbf{x}_i-\mathbf{x}_p|}\left(\frac{h''(s_{ip})}{|\mathbf{x}_i-\mathbf{x}_p|}-\frac{h'(s_{ip})}{|(\mathbf{x}_i+\mathbf{u}_i)-(\mathbf{x}_p+\mathbf{u}_p)|}\right)+\frac{h'(s_{ip})}{|(\mathbf{x}_i+\mathbf{u}_i)-(\mathbf{x}_p+\mathbf{u}_p)|}$$

Similarly,

$$\frac{\partial^2 E}{\partial u_{i,2}^2} = \frac{\big((x_{i,2}+u_{i,2})-(x_{p,2}+u_{p,2})\big)^2}{|(\mathbf{x}_i+\mathbf{u}_i)-(\mathbf{x}_p+\mathbf{u}_p)|^2|\mathbf{x}_i-\mathbf{x}_p|}\left(\frac{h''(s_{ip})}{|\mathbf{x}_i-\mathbf{x}_p|}-\frac{h'(s_{ip})}{|(\mathbf{x}_i+\mathbf{u}_i)-(\mathbf{x}_p+\mathbf{u}_p)|}\right)+\frac{h'(s_{ip})}{|(\mathbf{x}_i+\mathbf{u}_i)-(\mathbf{x}_p+\mathbf{u}_p)|}.$$

Following a very similar process, we can find $\partial/\partial u_{i,1}u_{i,2}$ and $\partial/\partial u_{i,2}u_{i,1}$, which end up being equivalent:

$$\begin{aligned}\frac{\partial}{\partial u_{i,1}\partial u_{i,2}} &= \frac{\partial}{\partial u_{i,2}\partial u_{i,1}}\\ &= \frac{\big((x_{i,1}+u_{i,1})-(x_{p,1}+u_{p,1})\big)\big((x_{i,2}+u_{i,2})-(x_{p,2}+u_{p,2})\big)}{|(\mathbf{x}_i+\mathbf{u}_i)-(\mathbf{x}_p+\mathbf{u}_p)|^2|\mathbf{x}_i-\mathbf{x}_p|}\left(\frac{h'(s_{ip})}{|(\mathbf{x}_i+\mathbf{u}_i)-(\mathbf{x}_p+\mathbf{u}_p)|}+\frac{h''(s_{ip})}{|\mathbf{x}_i-\mathbf{x}_p|}\right)\end{aligned}$$

3.1.3 Summary

When nodes i and j are not neighbors,

$$\frac{\partial^2 E}{\partial u_{i,1}\partial u_{j,1}} = \frac{\partial^2 E}{\partial u_{i,2}\partial u_{j,1}} = \frac{\partial^2 E}{\partial u_{i,1}\partial u_{j,2}} = \frac{\partial^2 E}{\partial u_{i,2}\partial u_{j,2}} = 0.$$

When nodes i and j are neighbors,

$$\begin{aligned}\frac{\partial^2 E}{\partial u_{i,1}\partial u_{j,1}} &= \frac{\big((x_{j,1}+u_{j,1})-(x_{i,1}+u_{i,1})\big)^2}{|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_i+\mathbf{u}_i)|^2|\mathbf{x}_j-\mathbf{x}_i|}\left(\frac{h'(s_{ji})}{|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_i+\mathbf{u}_i)|}-\frac{h''(s_{ji})}{|\mathbf{x}_j-\mathbf{x}_i|}\right)-\frac{h'(s_{ji})}{|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_i+\mathbf{u}_i)||\mathbf{x}_j-\mathbf{x}_i|}\\ \frac{\partial^2 E}{\partial u_{i,2}\partial u_{j,2}} &= \frac{\big((x_{j,2}+u_{j,2})-(x_{i,2}+u_{i,2})\big)^2}{|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_i+\mathbf{u}_i)|^2|\mathbf{x}_j-\mathbf{x}_i|}\left(\frac{h'(s_{ji})}{|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_i+\mathbf{u}_i)|}-\frac{h''(s_{ji})}{|\mathbf{x}_j-\mathbf{x}_i|}\right)-\frac{h'(s_{ji})}{|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_i+\mathbf{u}_i)||\mathbf{x}_j-\mathbf{x}_i|}\\ \frac{\partial^2 E}{\partial u_{i,1}\partial u_{j,2}} &= \frac{\big((x_{j,1}+u_{j,1})-(x_{i,1}+u_{i,1})\big)\big((x_{j,2}+u_{j,2})-(x_{i,2}+u_{i,2})\big)}{|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_i+\mathbf{u}_i)|^2|\mathbf{x}_j-\mathbf{x}_i|}\left(\frac{h'(s_{ji})}{|(\mathbf{x}_j+\mathbf{u}_j)-(\mathbf{x}_i+\mathbf{u}_i)|}-\frac{h''(s_{ji})}{|\mathbf{x}_j-\mathbf{x}_i|}\right)\\ &= \frac{\partial^2 E}{\partial u_{i,2}\partial u_{j,1}}\end{aligned}$$

Finally, when $i=j$, we have:

$$\begin{aligned}\frac{\partial^2 E}{\partial u_{i,1}^2} &= \sum_{p\in\Omega_i}\left(\frac{\big((x_{i,1}+u_{i,1})-(x_{p,1}+u_{p,1})\big)^2}{|(\mathbf{x}_i+\mathbf{u}_i)-(\mathbf{x}_p+\mathbf{u}_p)|^2|\mathbf{x}_i-\mathbf{x}_p|}\left(\frac{h''(s_{ip})}{|\mathbf{x}_i-\mathbf{x}_p|}-\frac{h'(s_{ip})}{|(\mathbf{x}_i+\mathbf{u}_i)-(\mathbf{x}_p+\mathbf{u}_p)|}\right)+\frac{h'(s_{ip})}{|(\mathbf{x}_i+\mathbf{u}_i)-(\mathbf{x}_p+\mathbf{u}_p)|}\right)\\ \frac{\partial^2 E}{\partial u_{i,2}^2} &= \sum_{p\in\Omega_i}\left(\frac{\big((x_{i,2}+u_{i,2})-(x_{p,2}+u_{p,2})\big)^2}{|(\mathbf{x}_i+\mathbf{u}_i)-(\mathbf{x}_p+\mathbf{u}_p)|^2|\mathbf{x}_i-\mathbf{x}_p|}\left(\frac{h''(s_{ip})}{|\mathbf{x}_i-\mathbf{x}_p|}-\frac{h'(s_{ip})}{|(\mathbf{x}_i+\mathbf{u}_i)-(\mathbf{x}_p+\mathbf{u}_p)|}\right)+\frac{h'(s_{ip})}{|(\mathbf{x}_i+\mathbf{u}_i)-(\mathbf{x}_p+\mathbf{u}_p)|}\right)\\ \frac{\partial^2 E}{\partial u_{i,1}\partial u_{i,2}} &= \sum_{p\in\Omega_i}\left(\frac{\big((x_{i,1}+u_{i,1})-(x_{p,1}+u_{p,1})\big)\big((x_{i,2}+u_{i,2})-(x_{p,2}+u_{p,2})\big)}{|(\mathbf{x}_i+\mathbf{u}_i)-(\mathbf{x}_p+\mathbf{u}_p)|^2|\mathbf{x}_i-\mathbf{x}_p|}\left(\frac{h'(s_{ip})}{|(\mathbf{x}_i+\mathbf{u}_i)-(\mathbf{x}_p+\mathbf{u}_p)|}+\frac{h''(s_{ip})}{|\mathbf{x}_i-\mathbf{x}_p|}\right)\right)\\ &= \frac{\partial^2 E}{\partial u_{i,2}\partial u_{i,1}}\end{aligned}$$

This can be visualized in a color-coded Hessian.

Assuming i and j are neighbors, any second derivative colored light green is equivalent to $\partial^2 E/\partial u_{i,1}\partial u_{j,1}$. Light pink indicates the second derivative of neighbors can be found with the formula for $\partial^2 E/\partial u_{i,1}\partial u_{j,2}$, and light blue indicates the formula for $\partial^2 E/\partial u_{i,2}\partial u_{j,2}$ should be used. Note that it is impossible to generalize which nodes are neighbors and which are not as it varies from lattice to lattice, so it is important to remember that the matrix could be dense or sparse depending on the lattice's initial

$$\begin{bmatrix}
\frac{\partial^2 E}{\partial u_{1,1}\partial u_{1,1}} & \frac{\partial^2 E}{\partial u_{1,1}\partial u_{2,1}} & \cdots & \frac{\partial^2 E}{\partial u_{1,1}\partial u_{n,1}} & \frac{\partial^2 E}{\partial u_{1,1}\partial u_{1,2}} & \cdots & \frac{\partial^2 E}{\partial u_{1,1}\partial u_{n,2}} \\
\frac{\partial^2 E}{\partial u_{2,1}\partial u_{1,1}} & \frac{\partial^2 E}{\partial u_{2,1}\partial u_{2,1}} & \cdots & \frac{\partial^2 E}{\partial u_{2,1}\partial u_{n,1}} & \frac{\partial^2 E}{\partial u_{2,1}\partial u_{1,2}} & \cdots & \frac{\partial^2 E}{\partial u_{2,1}\partial u_{n,2}} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\frac{\partial^2 E}{\partial u_{n,1}\partial u_{1,1}} & \frac{\partial^2 E}{\partial u_{n,1}\partial u_{2,1}} & \cdots & \frac{\partial^2 E}{\partial u_{n,1}\partial u_{n,1}} & \frac{\partial^2 E}{\partial u_{n,1}\partial u_{1,2}} & \cdots & \frac{\partial^2 E}{\partial u_{n,1}\partial u_{n,2}} \\
\frac{\partial^2 E}{\partial u_{1,2}\partial u_{1,1}} & \frac{\partial^2 E}{\partial u_{1,2}\partial u_{2,1}} & \cdots & \frac{\partial^2 E}{\partial u_{1,2}\partial u_{n,1}} & \frac{\partial^2 E}{\partial u_{1,2}\partial u_{1,2}} & \cdots & \frac{\partial^2 E}{\partial u_{1,2}\partial u_{n,2}} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\frac{\partial^2 E}{\partial u_{n,2}\partial u_{1,1}} & \frac{\partial^2 E}{\partial u_{n,2}\partial u_{2,1}} & \cdots & \frac{\partial^2 E}{\partial u_{n,2}\partial u_{n,1}} & \frac{\partial^2 E}{\partial u_{n,2}\partial u_{1,2}} & \cdots & \frac{\partial^2 E}{\partial u_{n,2}\partial u_{n,2}}
\end{bmatrix}$$

configuration. However, it will always be the case that $i = j$ along the diagonals of each quadrant of the matrix - this is denoted by the darker colors. Dark green corresponds to the formula for $\partial^2 E/\partial u_{i,1}^2$, hot pink corresponds to $\partial^2 E/\partial u_{i,1}\partial u_{i,2}$, and dark blue corresponds to $\partial^2 E/\partial u_{i,2}^2$. A more complete matrix, without dots, would demonstrate the diagonals more clearly.

With the preliminary computations set up, it is now possible to implement minimization schemes to investigate minimal energy configurations. With the help of the SciPy library in the Python programming language [6], we are able to perform the gradient descent, Broyden-Fletcher-Goldfarb-Shanno (BFGS), and Sequential Least Squares Quadratic Programming (SLSQP) minimization methods, as well as visualize the corresponding results. All schemes take the initial displacements of the nodes in the lattice as input, and allow us to observe the condition where the total energy of the entire structure is either minimized or at approximately 0. However, the outputs of the methods are different. The underlying expectation for the minimization schemes is that we will be able to locate the nearest minima for total energy and determine whether or not that point corresponds to a zero-energy state. For these simulations, we fix the critical value c of the energy and force model to be 0.9.

3.2 Gradient Descent

3.2.1 Overview

We implement a first-order iterative minimization algorithm following the idea of gradient descent. In this case, the algorithm is programmed to run for a specified amount of iterations, returning the energy scalar each execution for manual analysis. To begin, a list of all displacement vectors of the nodes in n^2 dimension is obtained:

$$\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_n] = [[u_{1,1}, u_{1,2}], [u_{2,1}, u_{2,2}], ..., [u_{n,1}, u_{n,2}]]$$

To calculate the total energy of the lattice configuration defined with the current U, we pass the displacement vectors into a series of helper functions which return the total energy (scalar) with respect to the strain (scalar) on each link as the starting iteration. We arrive at the strain values based on *Equation (3)*:

$$S = \{\frac{|(x_i + u_i) - (x_j + u_j)| - |x_i - x_j|}{|x_i - x_j|} : i < j \text{ and } j \in \Omega_i\}$$

By passing each element in S into *Equation (1)*, we obtain the energy on each link and subsequently the total energy on the lattice:

$$TotalEnergy = \sum_{s \in S} s^2 (s - c)^2$$

Following the initial iteration, we define the step size h as a modifiable number ($1e-5$ by default), the direction as $\nabla E(U)$, and N as the number of iterations. We then compute the displacement as a new input for total energy calculation:

$$U_1 = U - h \times \nabla E(U)$$

More specifically:

$$(u_{1,1})_1 = h \times \frac{\partial}{\partial u_{1,1}} E(U)$$

$$(u_{1,2})_1 = h \times \frac{\partial}{\partial u_{1,2}} E(U)$$

$$(u_{2,1})_1 = h \times \frac{\partial}{\partial u_{2,1}} E(U)$$

$$(u_{2,2})_1 = h \times \frac{\partial}{\partial u_{2,2}} E(U)$$

$$\ldots$$

$$(u_{n,1})_1 = h \times \frac{\partial}{\partial u_{n,1}} E(U)$$

$$(u_{n,2})_1 = h \times \frac{\partial}{\partial u_{n,2}} E(U)$$

(n being the number of nodes)

The result obtained is routed back to the functions that total the energy in the lattice with the new displacements. The process will be repeated N times, with the input for the m^{th} iteration ($m \in N$) defined as:

$$U_m = U_{m-1} - h \times \nabla E(U_{m-1})$$

Ultimately, we are able to observe the different energy values returned from each run of the algorithm, allowing for the investigation of zero-energy conditions:

$$S_m = \{\frac{|(x_i + u_i) - (x_j + u_j)| - |x_i - x_j|}{|x_i - x_j|} : i < j \text{ and } j \in \Omega_i \text{ and } u_i, u_j \in U_m\}$$

$$TotalEnergy_m = \sum_{s \in S_m} s^2 (s - c)^2$$

As an additional note, our team finds that the results are more optimal as the step size h decreases and the number of iterations N increases.

3.2.2 Demonstration

The research group performs a gradient descent minimization scheme on a 5×5 lattice.

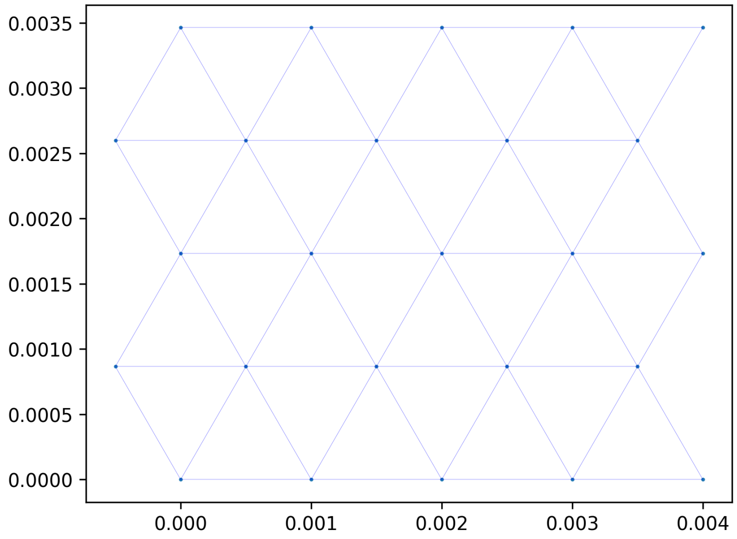


Figure 6: 5×5 lattice set-up.

We define U as a displacement field of a strain magnitude 0.7 on every link.

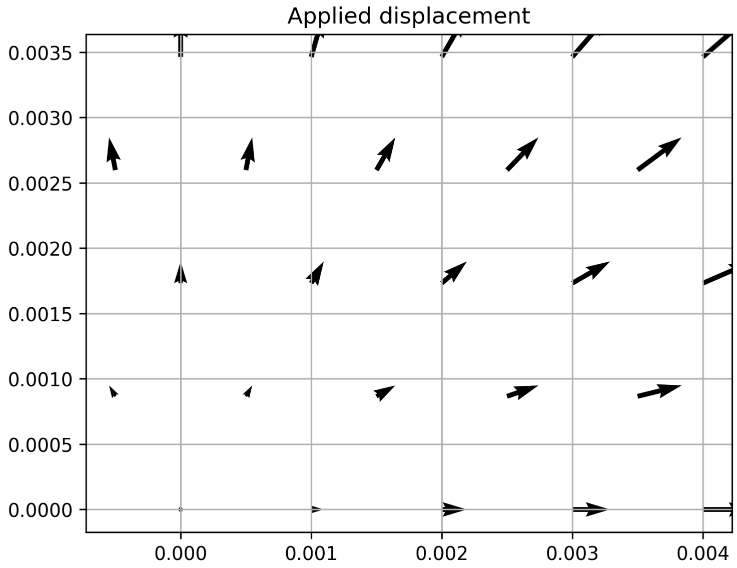


Figure 7: Scaling by 0.7 displacement on lattice.

We begin gradient descent with a stepping size $h = 1e - 8$. The scheme is run for 400 iterations. Results show that the lattice reaches a zero-energy state where all links expand into their long mode (notice that the lengths of the links are now 0.002, in contrast to 0.001 at the start).

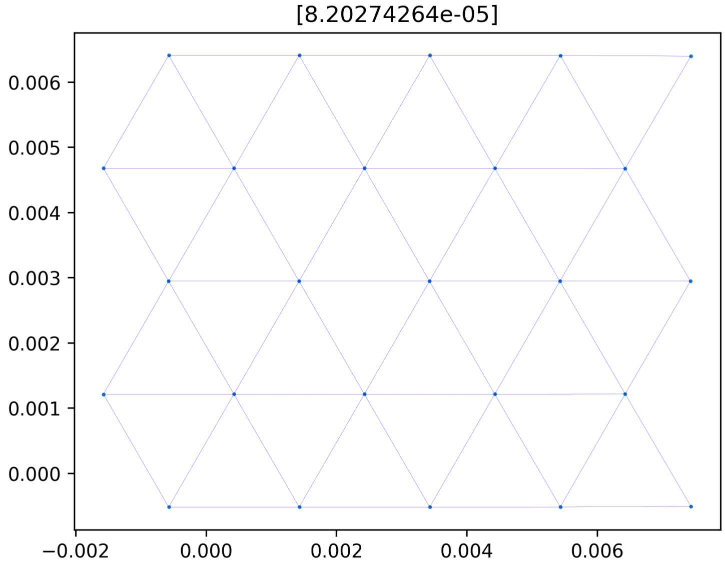


Figure 8: Result of running gradient descent on 0.7-scaling displacement.

We plot the total energy of the intermediate lattices with time (iterations). The model appears to be an exponential decay function. This confirms that the scheme is causing the lattice to converge to a state where energy is minimized.

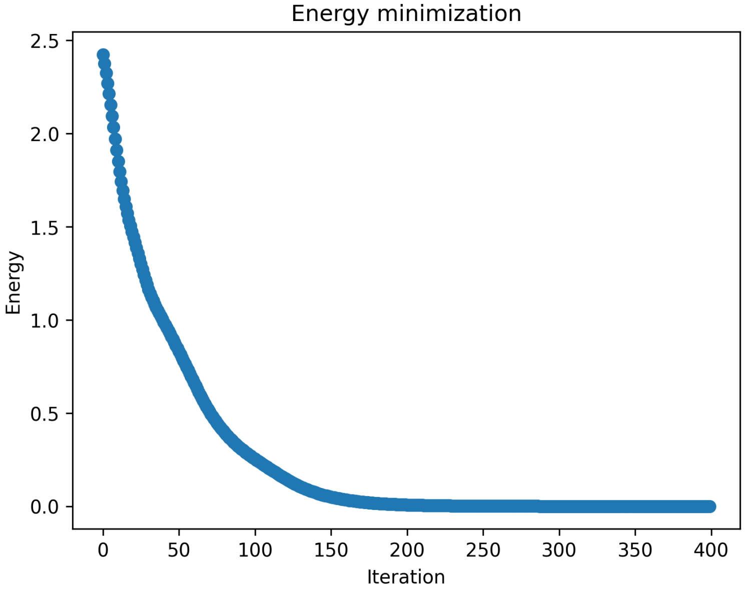


Figure 9: Total energy in time during gradient descent.

However, this minimization method is inaccurate at smaller precision degrees and concludes at the last specified iteration rather than the minimal energy state. Hence, it becomes difficult to pinpoint the exact minimal state in which we are interested.

3.3 Broyden-Fletcher-Goldfarb-Shanno (BFGS)

3.3.1 Overview

To accommodate for the shortcoming from the gradient descent method, the research group utilizes the 'minimize' function from Scipy's 'optimize' library, which by default is the Broyden-Fletcher-Goldfarb-

Shanno (BFGS) algorithm. The BFGS method automatically determines its stepping direction and size by approximating the second order derivative of the objective function gradually, allowing it to find the local minimum more accurately and fast. The following parameters are passed into the function for the most desirable outcome:

- Initial guess U
- Objective function $E(U)$
- Jacobian matrix *Matrix (4)*
- Hessian matrix *Matrix (5)*

We begin by taking the inverse of the Hessian matrix at U $(H(U))$. This process is done automatically by the algorithm. It is worth noting that the BFGS function is capable of approximating an inverse Hessian matrix from the objective function, meaning that it is not required that a Hessian matrix be calculated manually. However, this also means that the scheme will be approximating a second derivative matrix at every iteration, which is computationally expensive. Therefore, by constructing the Hessian prior to initiating the algorithm, we can save significant runtime and yield more accurate results. The inverse matrix $H^{-1}(U)$ is then used to define a step direction of:

$$H^{-1}(U) \times \nabla E(U)$$

This gives us a new value for U:

$$U_1 = U - H^{-1}(U) \times \nabla E(U)$$

At the new value U_1, we then calculate the total energy of the current configuration to assess the algorithm:

$$S = \{\frac{|(x_i + u_i) - (x_j + u_j)| - |x_i - x_j|}{|x_i - x_j|} : i < j \text{ and } j \in \Omega_i \text{ and } u_i, u_j \in U\}$$

$$TotalEnergy = \sum_{s \in S} s^2 (s - c)^2$$

In contrast to gradient descent, BFGS does not require a defined number of iterations, but rather automatically stops at an approximation of the nearest minima. The function will continue to run by updating U at every iteration k:

$$U_k = U_{k-1} - H^{-1}(U_{k-1}) \times \nabla E(U_{k-1})$$

3.3.2 Demonstration

We first define our lattice to be a $5x5$ structure visualized in *Figure 6*. We then define the displacement U to be a Gaussian displacement field with the mean being the midpoint of the lattice, stretching the relative middle link by some variable magnitude, while simultaneously clamping the bottom left most node.

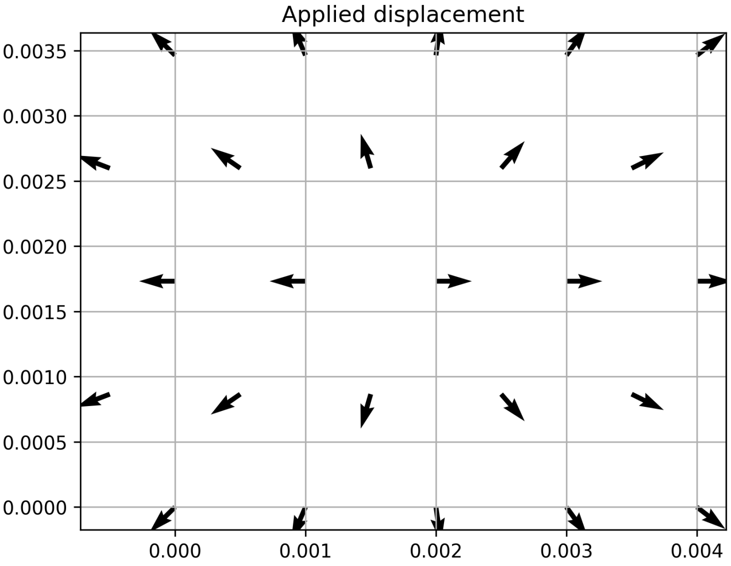


Figure 10: Gaussian displacement on lattice.

We begin running the BFGS optimization with a displacement magnitude of 0.3. This value is chosen to examine the hypothesis that if the strain value is less than 0.5, the lattice will snap back to its original configuration. At 0.3 magnitude, BFGS returns the nearest lattice topology with minimum energy:

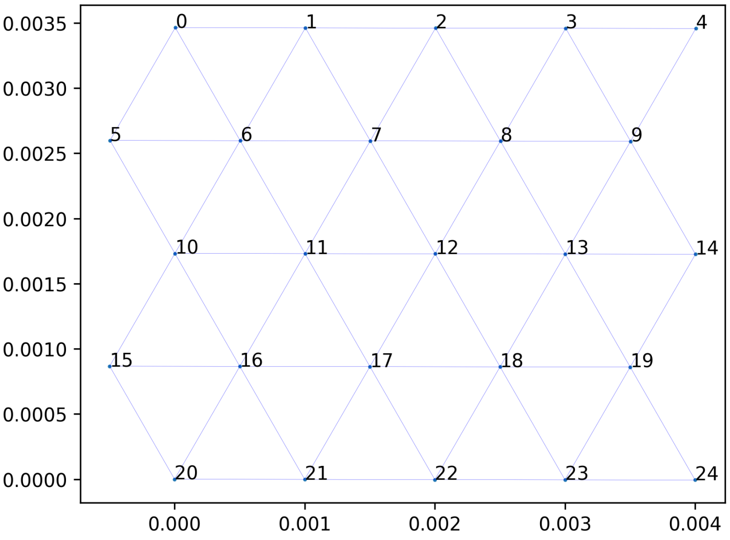


Figure 11: Lattice after BFGS minimization at 0.3 displacement magnitude.

This confirms our prediction of how the lattice would behave. This current configuration is in a zero-energy state. We continue by running the optimization algorithm at 0.5 magnitude:

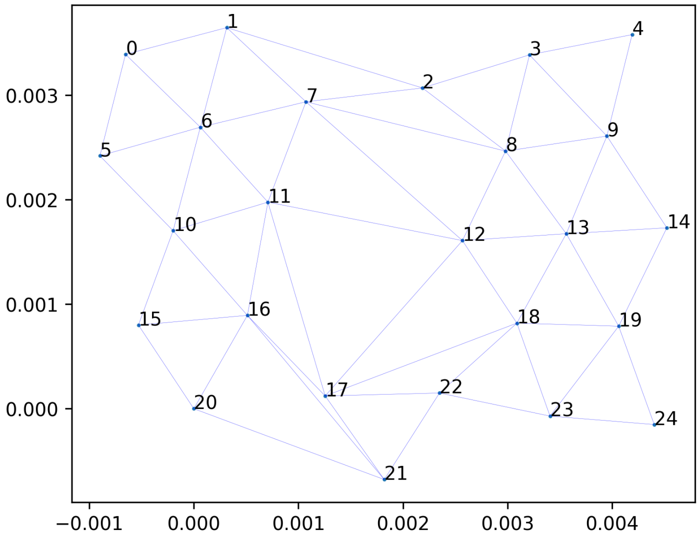


Figure 12: Lattice after BFGS minimization at 0.5 displacement magnitude.

Although this is not a zero-energy configuration, this lattice structure is in a minimal-energy state. We can verify this by setting up a test which perturbs each node by a random and small amount of displacement for a set number of times. As the current lattice is changing, we can observe the correlating changes in the total energy produced. A minimal-energy state implies that any minor perturbation will cause the lattice to produce a higher total energy value. We run the test on the result of the BFGS minimization for 100 iterations and are able to confirm that the resulting lattice is at a minimal-energy state.

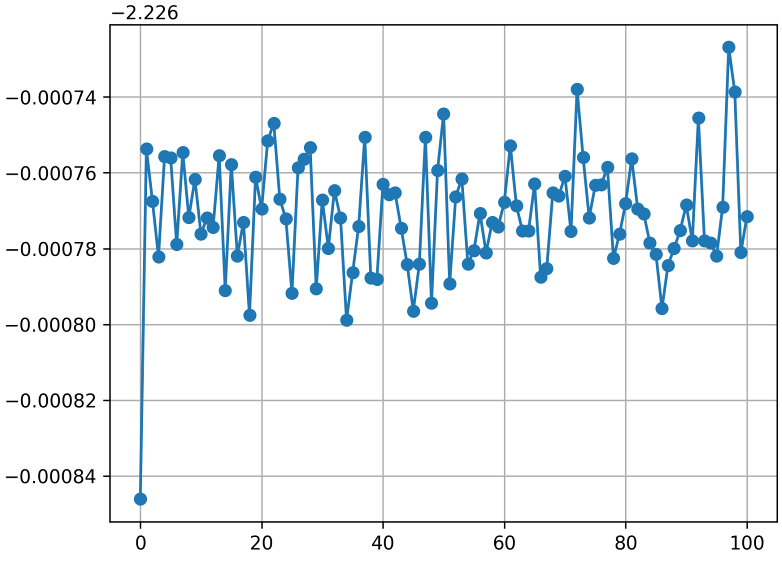


Figure 13: BFGS' product perturbing test result.

However, as we run the minimization scheme again with the displacement magnitude at 0.7, we observe that overlaps occur.

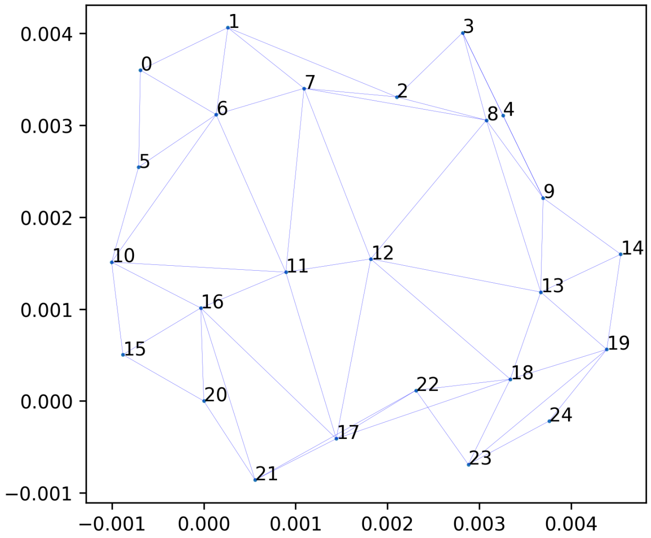


Figure 14: Lattice after BFGS minimization at 0.7 displacement magnitude..

From *Figure 14*, we can see that the link between node 22 and 23 overlaps with that between node 17 and 18. While this, in essence, is not an impossible configuration, this lattice layout is invalid as the research objectives lie within a 2 dimensional space. Therefore, overlap cannot be allowed. One solution will be to implement a constraint on the optimization scheme. Since the BFGS method is designed to solve unconstrained nonlinear optimization problems, the research team looks at a different scheme.

3.4 Sequential Least Squares Quadratic Programming (SLSQP)

3.4.1 Overview

In order to prevent the overlapping of lattice links, we implemented a Lagrange-Newton optimization method built in the 'minimize' function known as Sequential Least Squares Quadratic Programming (SLSQP). Similar to BFGS, this is an iterative optimization method, but for constrained nonlinear problems. SLSQP works by solving a sequence of quadratic programming subproblems, each of which approximates the original nonlinear problem locally using a quadratic model of the objective function and linear models of the constraints. SLSQP handles both equality and inequality constraints and supports variable bounds [5]. This is suitable for the issue in question - running a minimization scheme with a constraint to prevent undesired configurations as described in Section 3.5.

3.5 Angle constraint on lattice nodes

We notice that in a lattice where overlapping between links occurs, there are nodes at which the sum of the angles created by them and their neighboring nodes exceed 360 degrees. Realistically, all of the angles between consecutive edges around a node on a 2-dimensional surface should add up to exactly 360 degrees.

$$\sum_{i=1}^{n} \theta_i = 360^\circ$$

On this basis, we construct an equality constraint for the SLSQP function which holds the sum of the angles in clockwise order around all nodes to 360 degrees. The optimization problem is then defined as:

$$\min_{x} \; E(U)$$
$$\text{subject to} \;\; g(U) = 0,$$

with g(U) defined as:

$$V = X + U = ((x_0 + u_0),\ (x_1 + u_1), ..., (x_n + u_n))$$

$$\text{Let } \vec{v_1}, \vec{v_2} \text{ be two consecutive edges} \in V$$

$$\theta = (-\text{atan2}(\vec{v_1} \times \vec{v_2}, \vec{v_1} \cdot \vec{v_2})) \bmod 2\pi$$

$$\sum_{\text{nodes}} \left(360^\circ - \sum_{i=1}^{n} \theta_i \right)^2$$

3.5.1 Demonstration

We perform SLSQP on a $5x5$ lattice similar to *Figure* 6, with an initial displacement U as a Gaussian similar to that used for the BFGS minimization scheme (see *Figure* 10) at a magnitude of 0.7. With this new minimization, we are successful in eliminating the overlapping seen when running BFGS with the same initial displacement.

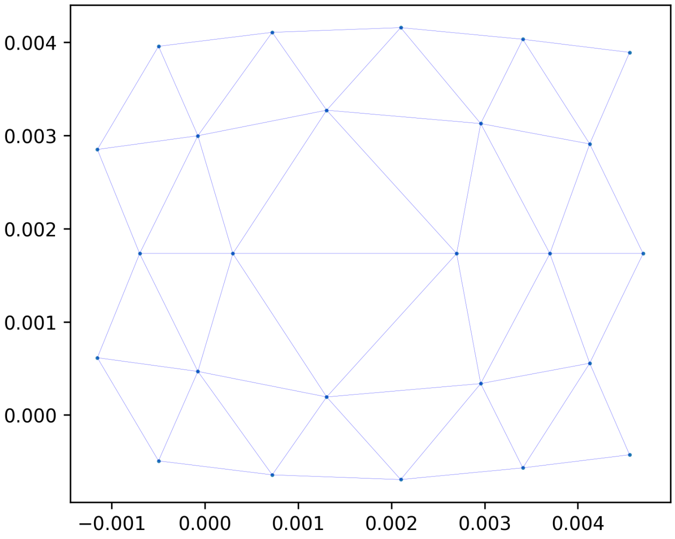


Figure 15: Lattice after SLSQP minimization at 0.7 Gaussian displacement magnitude.

However, this result is not an improvement from that of BFGS, but rather a trade-off. We initially notice that this configuration is almost identical to the starting configuration before the optimization. This means that the current lattice is unlikely to be at a minimal-energy state. We then verify this assumption by running the perturbation test set up above on this result. It then becomes evident that the minimization method has not been successful in yielding the desired result, as the test shows that energy can still be reduced.

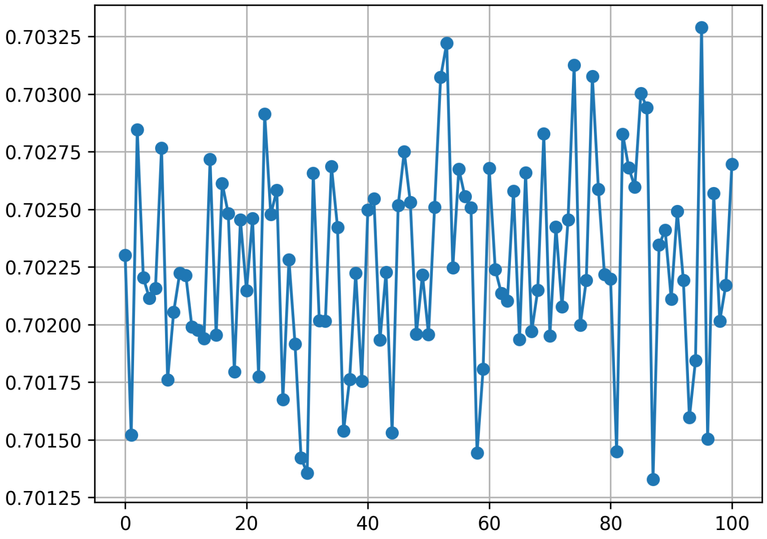


Figure 16: Perturbation test on SLSQP result.

Upon investigation, we hypothesize that the issue lies within the minimization scheme of not only SLSQP, but also any other constrained nonlinear optimization method. These functions, despite being able to work with constraints, are not capable of identifying another local minima when the one being pursued leads to constraint violation. As minimization schemes in general follow a defined stepping direction based on gradients for each initial displacement, they follow only one path to one nearest minimal-energy configuration. If that configuration happens to be invalid, the minimization scheme will stop right before violating the constraint, giving us a lattice that is not in equilibrium. This further shows that not every initial displacement on a lattice will result in a still state. In other words, not every starting configuration has a nearby valid minimal-energy state on a 2 dimensional surface.

3.6 Dynamic Relaxation

3.6.1 Overview

All of the above minimizing techniques seek to find the nearest minimal energy configuration of a lattice given any perturbation. None of them, however, simulate how a lattice settles into these states over time. Dynamic relaxation simulations provide this missing insight. We can relate displacement and force with the equation

$$F = ma(t) = m \cdot \frac{d}{dt}v(t) = m \cdot \frac{d^2}{dt^2}u(t).$$

We established in section 2.1 that the force on node i is equal to the sum of the forceson its connecting links. Since $u_i(t)$ is dependent on both the node i and time t, we can assume that $m = 1$ and write the force on node i as

$$\sum_{j\in\Omega_i} 2s_{ij}(s_{ij}-c)^2 + 2s_{ij}^2(s_{ij}-c) = F_i(t) = \frac{d^2}{dt^2}u_i(t).$$

We also already know $F_i(t)$ to be the derivative of our energy function with respect to strain, from section 2.1, so if $E(s)$ is our energy function with respect to strain, then

$$\frac{d^2}{dt^2}u_i(t) = \frac{d}{ds}\sum_{j\in\Omega_i} E(s_{ij}).$$

Note that the equation is energy-conserving, hence does not achieve equilibrium.

To allow the lattice configuration to dissipate energy, we introduce a damping term in the equation as follows.

$$m\frac{d^2}{dt^2}u(t) = F - \gamma\frac{d}{dt}u(t) \tag{2}$$

where $\gamma > 0$ is a prescribed damping coefficient.

We can use this relationship to simulate how a lattice loses energy after a deformation.

(Scheme details go here).

3.6.2 Demonstration

Using once again a 5x5 lattice similar to Figure 12, we begin with a random starting configuration with displacement at a magnitude of 0.6, shown in Figure 17. The arrows show the force at each node.

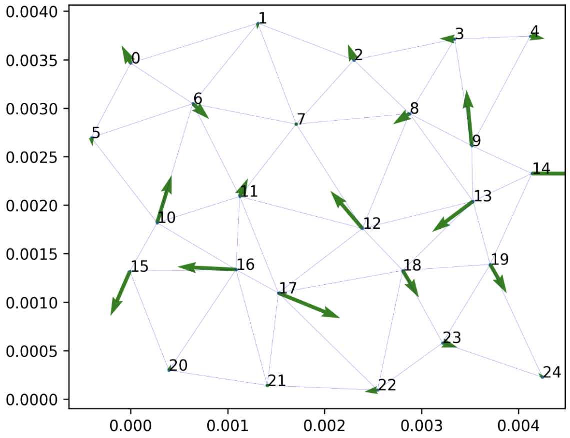


Figure 17: Lattice after random displacement at 0.6 magnitude, with potential energy of 2.3199953757196634

We then run dynamic relaxation for 1000 iterations with a damping value of $5e1$, using the strain-force function. The lattice's change in displacement at various time-steps can be seen in Figures 18, 19, 20, and 21.

We also visualize the lattice's total kinetic energy (Figure 22) and total potential energy (Figure 23), and we can observe that both approach 0 as expected.

The dynamic relaxation method has a much faster runtime than Gradient Descent, and it gives insight into the lattice's behavior as a function of time. However, this method still has several drawbacks; for example, for some initial configurations, the simulation returns a lattice with overlapping

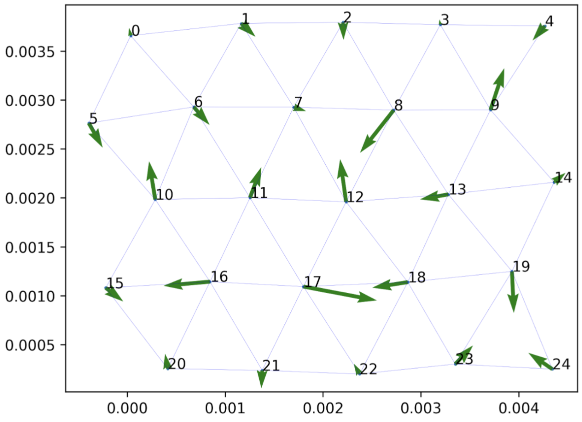


Figure 18: Lattice at iteration 70, with potential energy of 1.025356998857035.

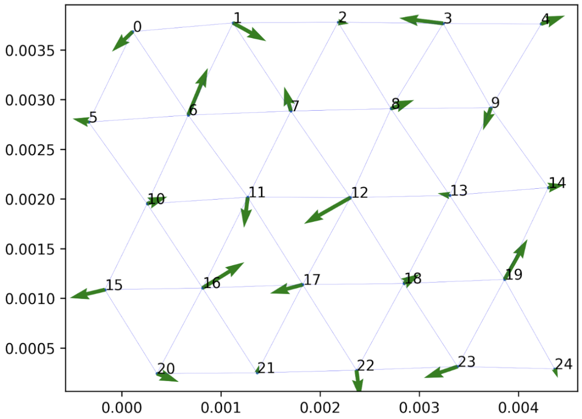


Figure 19: Lattice at iteration 100, with potential energy of 0.6503751849775167.

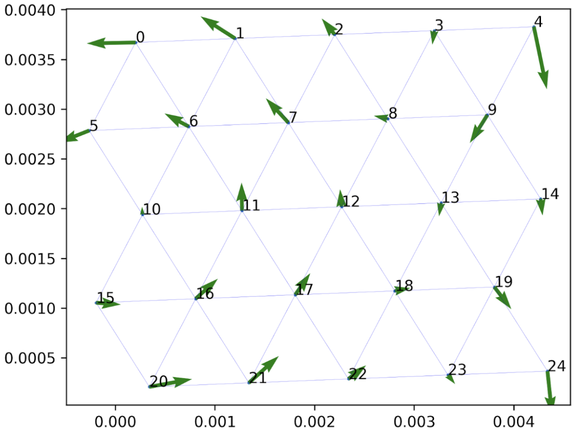


Figure 20: Lattice at iteration 500, with potential energy of 0.10744732291951246

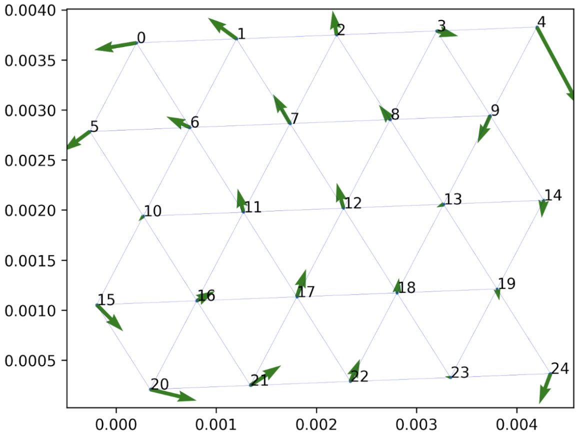


Figure 21: Lattice at iteration 990, with potential energy of $5.224055059038536e-12$.

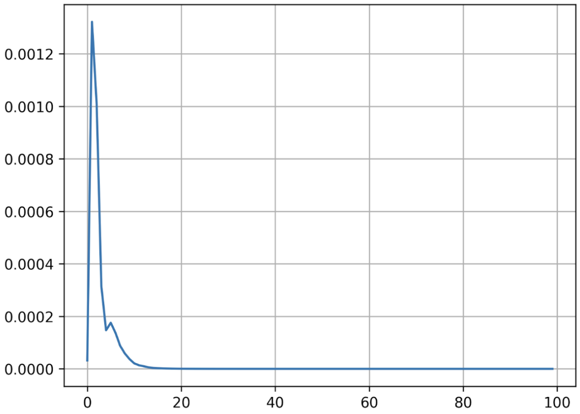


Figure 22: Lattice's total kinetic energy as a function of time.

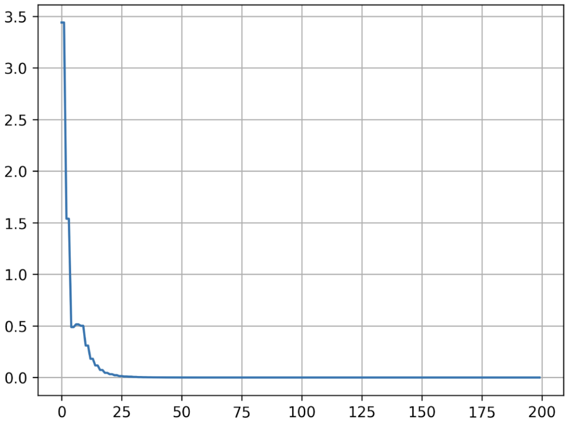
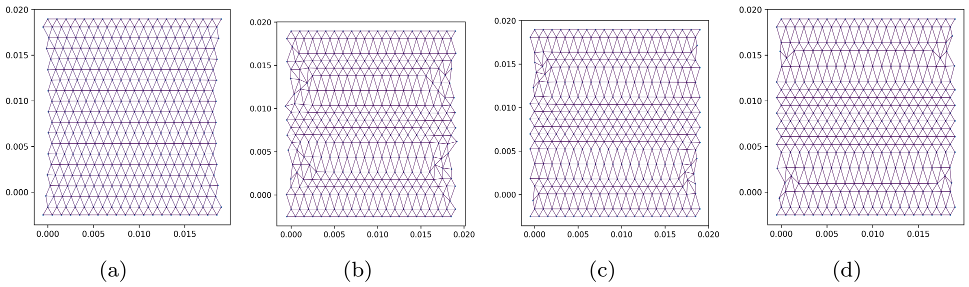


Figure 23: Lattice's total potential energy as a function of time.

links. It is also difficult to determine whether the energy of the lattice will reach zero or approach zero, which could indicate either a non-zero minimal energy state or a non-minimized state. It also appears that the program is not aware of each link's second equilibrium (notice how every link in Figure 21 appears to be in short mode). Similar to our other minimization methods, Dynamic Relaxation has its advantages and imperfections and can be considered as an alternative to Gradient Descent, BFGS, and SLSQP to generate and evaluate minimal energy states.

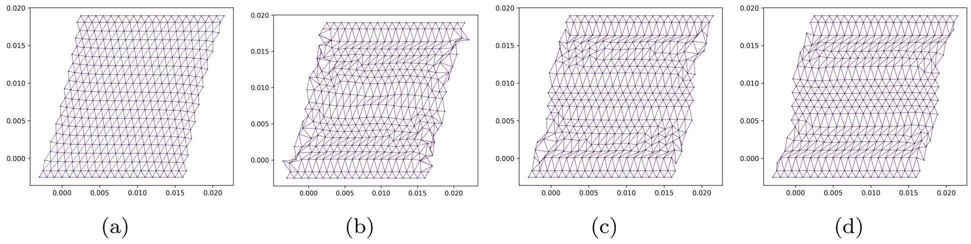
4 Quasistatic solution to boundary value problems

We consider a 20×20 triangular lattice with the $b_c = 0.8$ as the second local minimum for the bond energy. We apply vertical extension by an amount of 2.5× bond length on the top and bottom two layers of the lattice. The solution to the quasistatic problem are shown in Figure 24. Next, we apply an extension and shearing displacement on the top and bottom two layers. The horizontal displacement is also taken to be 2.5× bond length. The equilibrium solution is shown in Figure 25. The equilibrium solutions are compared with the solution to the same lattice consisting of Hookean springs of the same spring constant. It is noteworthy that the solution for the Hookean lattice is unique whereas there are finitely many distinct solutions for the bistable lattice.



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

Figure 24: Solutions to the quasistatic boundary value problem under vertical extension. The solution for the Hookean lattice is shown in Figure 24(a). In Figures 24(b) to 24(d), we show three distinct solutions to the same problem for the bistable lattice.



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

Figure 25: Solution to the quasistatic boundary value problem under shear and extension. The solution for the Hookean lattice is shown in Figure 25(a). In Figures 25(b) to 25(d), we show three distinct solutions to the same problem for the bistable lattice.

5 Conditions on Triangular Lattice Configurations

Following the computer simulations, we observe that the movement of the lattice is possibly subject to restriction by the coordination of the edges, or in other words, the length and connectivity of the links. For instance, all bistable lattices theoretically have many possible still and minimal energy configurations; we would like to say that a lattice with N links has 2^N still states alone, since every link has two modes. However, this hypothesis is limited by the lattice's unique geometry, which creates conditions on lattice behavior. The following section will focus on triangular bistable lattices, drawing from Cherkaev's 2010 paper describing necessary geometric conditions of a triangular lattice [2]. Consider a hexagonal section of a triangular lattice as in Figure 26.

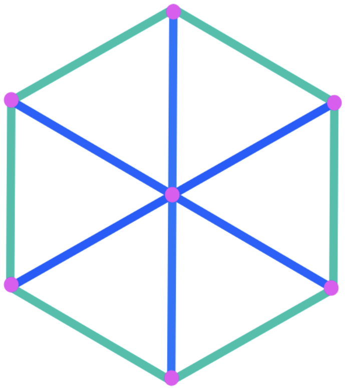


Figure 26: Simple hexagonal subsection of a triangular lattice.

Cherkaev states that the sum of the elongations of the **spokes** of the hexagon (the blue links) must equal the sum of the elongations of the **rims** (the green links) at all times. If a lattice is **compatible**, it follows Cherkaev's geometrical conditions. We can extrapolate these conditions to apply to any triangular lattice - if there exists a hexagonal subsection of the lattice, a deformation of a rim link must be balanced with a deformation of a spoke link, and visa versa. If these conditions are not fulfilled, the lattice breaks or exits the 2-dimensional plane.

This creates conditions on possible still states of the lattice - there must be an equal amount of elongated rims and spokes. With this knowledge, it is possible to count all still states in a triangular lattice consisting of a simple hexagon, like that in Figure 26. There exist seven possible variations of still states:

1. All links are in short mode;
2. All links are in long mode;
3. One rim and one spoke are in long mode while the rest are in short mode;
4. Two rims and two spokes are in long mode while the rest are in short mode;
5. Three rims and three spokes are in long mode while the rest are in short mode;
6. Two rims and two spokes are in short mode while the rest are in long mode; and
7. One rim and one spoke are in short mode while the rest are in long mode.

Though there is just one possible compatible configuration for variations 1 and 2, there are multiple compatible configurations possible with variations 3-7. We can calculate these by multiplying the total possible rim combinations with the total possible spoke combinations[5]:

3. $({}^6C_1)^2 = 6^2 = 36$;
4. $({}^6C_2)^2 = 15^2 = 225$;
5. $({}^6C_3)^2 = 20^2 = 400$;
6. $({}^6C_2)^2 = 15^2 = 225$; and
7. $({}^6C_1)^2 = 6^2 = 36$.

[5] For example, for variation 3, we select one out of 6 rims and one out of 6 spokes, giving us ${}^6C_1 \cdot {}^6 C_1$ or $\left({}^6C_1\right)^2$.

Summing up, we have that the total number of still states for a simple hexagon triangular lattice is

$$1 + 1 + 36 + 225 + 400 + 225 + 36 = 924.$$

This being a rather large number gives us a sense of the sheer magnitude of the total number of still states for a lattice as simple as Figure 26. A larger lattice containing more hexagons would have more links in the lattice, which means means more possible links to manipulate, giving rise to many more possible still state configurations.

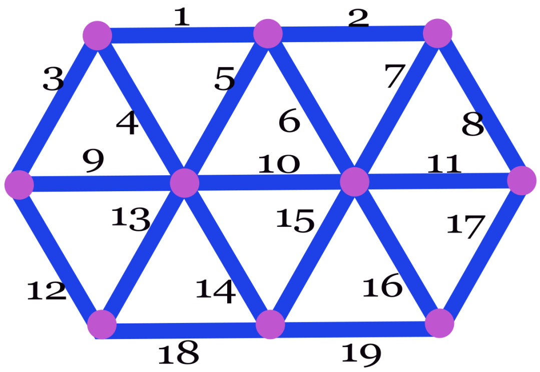


Figure 27: Triangular lattice of two hexagons.

Consider, for example, the lattice in Figure 27, comprised of two overlapping hexagons that have 5 **shared links** - specifically, links 5, 6, 10, 14, and 15. We will refer to all other links in the lattice as **free links**. To sum up the total number of still states for this lattice, we assume that the leftmost hexagon (we will call it Hexagon 1) can be in any of its 924 configurations, and we determine which configurations of the right hexagon (Hexagon 2) are compatible.

Given any configuration of Hexagon 1, the five links shared between Hexagons 1 and 2 mean that Hexagon 2's still states are restricted. Using a counting argument, we will determine how many still states Hexagon 2 has given a configuration of its shared links. We can categorize configurations of the shared links in terms of the 2 shared rims and 3 shared spokes of Hexagon 2:

1. All 2 rims and all 3 spokes short;
2. 2 rims and 2 spokes short, 1 spoke long;
3. 2 rims and 1 spoke short, 2 spokes long;
4. 2 rims short, 3 spokes long;
5. All 2 rims and all 3 spokes long;
6. 2 rims and 2 spokes long, 1 spoke short;
7. 2 rims and 1 spoke long, 2 spokes short;
8. 2 rims long, 3 spokes short;
9. 1 rim and 3 spokes short, 1 rim long;
10. 1 rim and 2 spokes short, 1 rim and 1 spoke long;
11. 1 rim and 1 spoke short, 1 rim and 2 spokes long; and
12. 1 rim short, 1 rim and 3 spokes long.

In essence, for each of the 924 configurations of Hexagon 1, the five shared links will be locked in one of these 12 configurations. Now, given any of these 12 configurations of the shared links, we can count how many out of Hexagon 2's 924 combinations are compatible by counting configurations of the free links[6]:

1. $\left({}^4C_0 \cdot {}^3C_0\right) + \left({}^4C_1 \cdot {}^3C_1\right) + \left({}^4C_2 \cdot {}^3C_2\right) + \left({}^4C_3 \cdot {}^3C_3\right) = 35$
2. $\left({}^4C_1\right) + \left({}^4C_2 \cdot {}^3C_1\right) + \left({}^4C_3 \cdot {}^3C_2\right) + \left({}^4C_4 \cdot {}^3C_2\right) = 35$
3. $\left({}^4C_2\right) + \left({}^4C_3 \cdot {}^3C_1\right) + \left({}^4C_4 \cdot {}^3C_2\right) = 21$
4. $\left({}^4C_3\right) + \left({}^4C_4 \cdot {}^3C_1\right) = 7$
5. $\left({}^4C_1\right) + \left({}^4C_2 \cdot {}^3C_1\right) + \left({}^4C_3 \cdot {}^3C_2\right) + \left({}^4C_4 \cdot {}^3C_3\right) = 35$
6. $\left({}^4C_0 \cdot {}^3C_0\right) + \left({}^4C_1 \cdot {}^3C_1\right) + \left({}^4C_2 \cdot {}^3C_2\right) + \left({}^4C_3 \cdot {}^3C_3\right) = 35$
7. $\left({}^3C_1\right) + \left({}^4C_1 \cdot {}^3C_2\right) + \left({}^4C_2 \cdot {}^3C_3\right) = 21$
8. $\left({}^3C_2\right) + \left({}^4C_1 \cdot {}^3C_3\right) = 7$
9. $\left({}^3C_1\right) + \left({}^4C_1 \cdot {}^3C_2\right) + \left({}^4C_2 \cdot {}^3C_3\right) = 21$
10. $\left({}^4C_0 \cdot {}^3C_0\right) + \left({}^4C_1 \cdot {}^3C_1\right) + \left({}^4C_2 \cdot {}^3C_2\right) + \left({}^4C_3 \cdot {}^3C_3\right) = 35$
11. $\left({}^4C_1\right) + \left({}^4C_2 \cdot {}^3C_1\right) + \left({}^4C_3 \cdot {}^3C_2\right) + \left({}^4C_4 \cdot {}^3C_3\right) = 35$
12. $\left({}^4C_2\right) + \left({}^4C_3 \cdot {}^3C_1\right) + \left({}^4C_4 \cdot {}^3C_2\right) = 21$

Finally, to sum up all still states of the lattice in Figure 27, we need to determine which configurations of Hexagon 2 occur with each of the 924 configurations of Hexagon 1. As previously established, Hexagon 1 could have all links short, all links long, only 1 rim and 1 spoke long, 2 rims and 2 spokes long, 3 rims and 3 spokes long, 4 rims and 4 spokes long, or 5 rims and 5 spokes long. If all links of Hexagon 1 are short, then all shared links are short, and our lattice falls into the first category in the list above. Thus, there exist 35 still states where all links in Hexagon 1 are short. We can say the same if all links in Hexagon 1 are long.

There are 36 configurations where one rim and one spoke of Hexagon 1 are long and everything else is short, each of which influences the shared links differently. Of these 36 configurations[7],

- ${}^4C_1 \cdot {}^3C_1 = 12$ do not elongate the shared links, falling into Category 1;
- $2 \cdot \left({}^3C_1\right) = 6$ have only link 6 or 15 elongated, falling into Category 2 in the list above;
- $2 \cdot \left({}^4C_1\right) = 8$ have only link 5 or 14 elongated, falling into Category 9;
- ${}^4C_1 = 4$ have only link 10 elongated, falling into Category 2;
- $2 \cdot 2 \cdot \left({}^4C_0 \cdot {}^3C_0\right) = 4$ have link 5 or 14 AND link 6 or 15 elongated, falling into Category 10; and
- $2 \cdot \left({}^4C_0 \cdot {}^3C_3\right) = 2$ have link 10 and either link 6 or 15 elongated, falling into Category 3.

[6] For example, for category 1, we have all 5 shared links short, and we need to determine how the four free rims and three free spokes can behave. We can choose no rims or spokes to be long (${}^4C_0 \cdot {}^3C_0$), we can have one, two, or three of each long. We cannot have four of each long, because three spokes are short by hypothesis.

[7] Each bullet point represents a possible combination of elongated shared links. By hypothesis, we have one rim and one spoke of Hexagon 1 long, so for each bullet point we determine how many free rims and/or spokes must be elongated to meet this hypothesis. For example, for the first bullet point, none of the shared links are elongated, so we must elongate one out of the four free rims and one out of the three free spokes of Hexagon 1. Each bullet point is then categorized according to the shared links' configuration's impact on Hexagon 2. This logic is maintained when supposing that two rims and two spokes of Hexagon 1, as well as three rims and three spokes of Hexagon 1, are elongated.

For each of these configurations, we can multiply by the number of Hexagon 2's configurations that are compatible:

$$(12 \cdot 35) + (6 \cdot 35) + (8 \cdot 21) + (4 \cdot 35) + (4 \cdot 35) + (2 \cdot 21) = 1120.$$

Thus, there exist 1,120 still states of the lattice when Hexagon 1 has one rim and one spoke elongated. Because of the inherent symmetry in the hexagon, we find that this is the same result as when Hexagon 1 has five rims and five spokes elongated.

There are 225 configurations where two rims and two spokes of Hexagon 1 are long and everything else is short. Of these 225 configurations,

- ${}^4C_2 \cdot^3 C_2 = 18$ do not elongate the shared links, falling into Category 1;
- $2 \cdot \left({}^4C_1 \cdot^3 C_2\right) = 24$ have only link 6 or 15 elongated, falling into Category 2;
- ${}^3C_2 = 3$ have both links 6 and 15 elongated, falling into Category 3;
- $2 \cdot \left({}^4C_2 \cdot^3 C_1\right) = 36$ have only either link 5 or 14 elongated, falling into Category 9;
- ${}^4C_2 = 6$ have both links 5 and 14 elongated, falling into Category 8;
- ${}^4C_2 \cdot^3 C_1 = 18$ have only link 10 elongated, falling into Category 2;
- $2 \cdot \left({}^4C_2\right) = 12$ have either link 5 or 14 AND link 10 elongated, falling into Category 10;
- $2 \cdot 2 \cdot \left({}^4C_1 \cdot^3 C_1\right) = 48$ have either link 6 or 15 and either link 5 or 14 elongated, falling into Category 10;
- $2 \cdot \left({}^4C_1 \cdot^3 C_1\right) = 24$ have either link 6 or 15 and link 10 elongated, falling into Category 3;
- $2 \cdot \left({}^3C_1\right) = 6$ have links 6, 15, and either 5 or 14 elongated, falling into Category 11;
- ${}^3C_1 = 3$ have links 6, 15, and 10 elongated, falling into Category 4;
- $2 \cdot \left({}^4C_1\right) = 8$ have links 5, 14, and either 6 or 15 elongated, falling into Category 7;
- $2 \cdot 2 \cdot \left({}^4C_1\right) = 16$ have either link 6 or 15, either link 5 or 14, and link 10 elongated, falling into Category 11;
- ${}^4C_0 \cdot^3 C_0 = 1$ has links 5, 6, 14, and 15 elongated, falling into Category 6; and
- $2 \cdot \left({}^4C_0 \cdot^3 C_0\right) = 2$ have links 5, 6, 10, and either 5 or 14 elongated, falling into Category 12.

Similar to before, we can multiply each of these values by the number of Hexagon 2's configurations for which they are compatible:

$$\begin{aligned}&(18 \cdot 35) + (24 \cdot 35) + (3 \cdot 21) + (36 \cdot 21) + (6 \cdot 7) + (18 \cdot 35) + (12 \cdot 35) + (48 \cdot 35) + (24 \cdot 21)\\ &+ (6 \cdot 35) + (3 \cdot 7) + (8 \cdot 21) + (16 \cdot 35) + (1 \cdot 35) + (2 \cdot 21)\\ &= 6601\end{aligned}$$

Thus, there exist 6,601 still states of the lattice when Hexagon 1 has two rims and two spokes elongated. Just like in the previous case, symmetry gives us that there also exists 6,601 still states of lattice when Hexagon 1 has four rims and four spokes elongated.

There are 400 configurations where three rims and three spokes of Hexagon 1 are long and everything else is short. Of these 400 configurations,

- ${}^4C_3 \cdot^3 C_3 = 4$ do not elongate the shared links, falling into Category 1;
- $2 \cdot \left({}^4C_2 \cdot^3 C_3\right) = 12$ have only link 6 or 15 elongated, falling into Category 2;
- ${}^4C_1 \cdot^3 C_3 = 4$ have both links 6 and 15 elongated, falling into Category 3;

- $2 \cdot \left({}^4C_3 \cdot{}^3 C_2\right) = 24$ have only link 5 or 14 long, falling into Category 9;
- ${}^4C_3 \cdot{}^3 C_1 = 12$ have both links 5 and 14 long, falling into Category 8;
- ${}^4C_3 \cdot{}^3 C_2 = 12$ have only link 10 elongated, falling into Category 2;
- $2 \cdot \left({}^4C_3 \cdot{}^3 C_1\right) = 24$ have link 10 and either link 5 or 14 long, falling into Category 10;
- ${}^4C_3 = 4$ have links 5, 10, and 14 elongated, falling into Category 7;
- $2 \cdot 2 \cdot \left({}^4C_2 \cdot{}^3 C_2\right) = 72$ have either link 6 or 15 and either link 5 or 14 elongated, falling into Category 10;
- $2 \cdot \left({}^4C_2 \cdot{}^3 C_2\right) = 36$ have link 10 and either link 6 or 15 elongated, falling into Category 3;
- $2 \cdot \left({}^4C_1 \cdot{}^3 C_2\right) = 24$ have links 6, 15, and either link 5 or 14 elongated, falling into Category 11;
- ${}^4C_1 \cdot{}^3 C_2 = 12$ have links 6, 15, and 10 elongated, falling into Category 4;
- $2 \cdot \left({}^4C_2 \cdot{}^3 C_1\right) = 36$ have links 5, 14, and either link 6 or 15 elongated, falling into Category 7;
- $2 \cdot 2 \cdot \left({}^4C_2 \cdot{}^3 C_1\right) = 72$ have link 10, either link 6 or 15, and either link 5 or 14 elongated, falling into Category 11;
- ${}^4C_1 \cdot{}^3 C_1 = 12$ have links 5, 6, 14, and 15 elongated, falling into Category 6;
- $2 \cdot \left({}^4C_1 \cdot{}^3 C_1\right) = 24$ have links 6, 10, 15, and either link 5 or 14 elongated, falling into Category 12;
- $2 \cdot \left({}^4C_2\right) = 12$ have links 5, 10, 14, and either link 6 or 10 elongated, falling into Category 6; and
- ${}^4C_1 = 4$ have links 5, 6, 10, 14, and 15 elongated, falling into Category 5.

Once again, we can multiply each of these values by the number of configurations of Hexagon 2 that are compatible:

$$\begin{aligned}
&(4 \cdot 35) + (12 \cdot 35) + (4 \cdot 21) + (24 \cdot 21) + (12 \cdot 7) + (12 \cdot 35) + (24 \cdot 35) + (4 \cdot 21) \\
&+ (72 \cdot 35) + (36 \cdot 21) + (24 \cdot 35) + (12 \cdot 7) + (36 \cdot 21) + (72 \cdot 35) + (12 \cdot 35) + (24 \cdot 21) \\
&+ (12 \cdot 35) + (4 \cdot 35) \\
&= 11536
\end{aligned}$$

Thus, there exist 11,536 still states of the lattice where three rims and three spokes of Hexagon 1 are elongated.

For each of the 924 possible still configurations of Hexagon 1, we have found how many ways Hexagon 2 can also be still. To get the total number of still states in the lattice, we sum these up:

$$(2 \cdot 35) + (2 \cdot 1120) + (2 \cdot 6601) + 11536 = 27048$$

We can conclude that there are 27,048 still configurations of the lattice in Figure 27. This is a fraction of the early estimate of $2^{19} = 524288$ still states for a lattice with 19 links, which gives an idea of the impact overlapping hexagons have on one another. Future research could look at counting the still states of larger lattices using similar counting arguments.

Larger lattices not only contain many overlapping hexagons, but often their configurations create larger hexagons that also must follow Cherkaev's compatibility conditions. For example, consider an expanded version of our simple hexagon-shaped triangular lattice, depicted in Figure 28. The lattice contains 42 links, and within the large green hexagon are 7 smaller hexagons (example given in yellow). Each hexagon is subject to Cherkaev's conditions. We can prove that if all of the smaller hexagons are in compatible still states - that is, that it meets Cherkaev's conditions and each link is either in long or short mode - then the large hexagon is in a compatible still state.

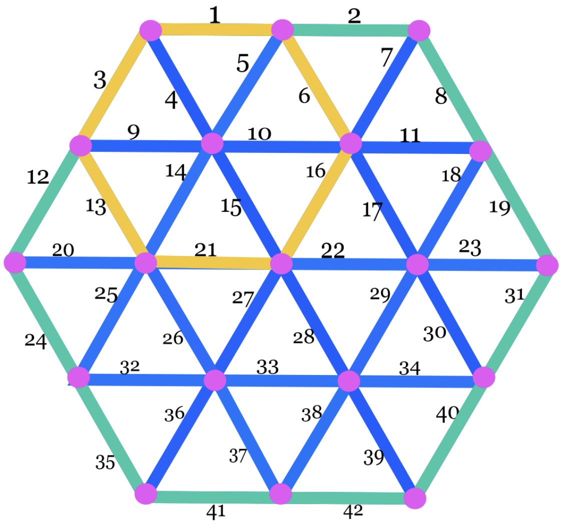


Figure 28: Large hexagonal subsection of a triangular lattice.

First, suppose all of the smaller hexagons are in compatible still states. This assumption gives us the following equations (note that 1, for example, does not mean the numerical value of 1, but rather the elongation of link 1):

$$\begin{gathered}
1+3+6+13+16+21=4+5+9+10+14+15\\
2+5+8+15+18+22=6+7+10+11+16+17\\
9+12+15+24+27+32=13+14+20+21+25+26\\
10+14+17+26+29+33=15+16+21+22+27+28\\
11+16+19+28+31+34=17+18+22+23+29+30\\
21+25+28+35+38+41=26+27+32+33+36+37\\
22+27+30+37+40+42=28+29+33+34+38+39.
\end{gathered}$$

Note also that because each hexagon is in its still state, every link is either short or long. We assume that every link must be elongated the same length - call it l - to be in long mode, so each term in every equality above has a value of either 0 or l. Now observe that when we sum all seven equalities and cancel terms that appear on both sides of the equality,

$$\begin{array}{l}
\quad 1+3+6+13+16+21=4+5+9+10+14+15\\
+\big(2+5+8+15+18+22=6+7+10+11+16+17\big)\\
\hline
\quad 1+2+3+8+13+18+21+22=4+7+9+2(10)+11+14+17\\
+\big(9+12+15+24+27+32=13+14+20+21+25+26\big)\\
\hline
\quad 1+2+3+8+12+15+18+22+24+27+32=4+7+2(10)+11+2(14)+17+20+25+26\\
+\big(10+14+17+26+29+33=15+16+21+22+27+28\big)\\
\hline
\quad 1+2+3+8+12+18+24+29+32+33=4+7+10+11+14+16+20+21+25+28\\
+\big(11+16+19+28+31+34=17+18+22+23+29+30\big)\\
\hline
\quad 1+2+3+8+12+19+24+31+32+33+34=4+7+10+14+17+20+21+22+23+25+30\\
+\big(21+25+28+35+38+41=26+27+32+33+36+37\big)\\
\hline
\quad 1+2+3+8+12+19+24+28+31+34+35+38+41=4+7+10+14+17+20+22+23+26\\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad+27+30+36+37\\
+\big(22+27+30+37+40+42=28+29+33+34+38+39\big)\\
\hline
\quad 1+2+3+8+12+19+24+31+35+40+41+42=4+7+10+14+17+20+23+26+29+33+36+39,
\end{array}$$

we get one equality with twelve terms on each side. On the left, each term represents a link that makes up the rim of the large hexagon. On the right, six terms represent links that make up the spokes of the large hexagon (respectively, 4, 7, 20, 23, 36, and 39), and the other six terms (10, 14, 17, 26, 29, and 33) make up the rim of the center hexagon. We know by hypothesis that the total elongation of the center hexagon's rims is equal to the total elongation of its spokes; so by substitution, we get

$$\begin{aligned}&1+2+3+8+12+19+24+31+35+40+41+42\\&\quad=4+7+15+16+20+21+22+23+27+28+36+39,\end{aligned}$$

and the large hexagon is compatible according to Cherkaev's conditions. Recall that each term in this equality has a value of 0 or l, so the links that make up the large hexagon have 0 energy. We can therefore conclude that if each of the smaller hexagons are in compatible still states, then so is the large hexagon.

It is important to note that the converse of this statement is **NOT** true - a large compatible hexagon does imply that all small hexagons are balanced, but if the large hexagon is in a still state, it does not imply that all the small hexagons are likewise in still configurations. For example, if the large hexagon is in a compatible still state, we can have links 5 and 6 equally elongated to a length between 0 and l without violating any geometric conditions. Similarly, the links that make up the rim of the center hexagon could be at any elongation between 0 and l, as long as their sum is equal to the elongation of the spokes. This will have interesting implications in future efforts to sum up the total number of still states in this lattice.

While all of the above pertains strictly to triangular lattices, it seems likely that similar conditions can be found for square lattices, Penrose lattices, and others. Efforts to find such configurations and sum the still states of these lattices will be crucial in broadening our understanding of bistable lattice behavior.

6 3D Printed Bistable Structure

6.1 Bistable Link

Our team sets out to prove whether this condition on lattice behavior exists. Using 3D printers, we are able to create a bistable structure, which is then used to create a bistable lattice. The design of a bistable edge is below:

Figure 29: Bistable link design

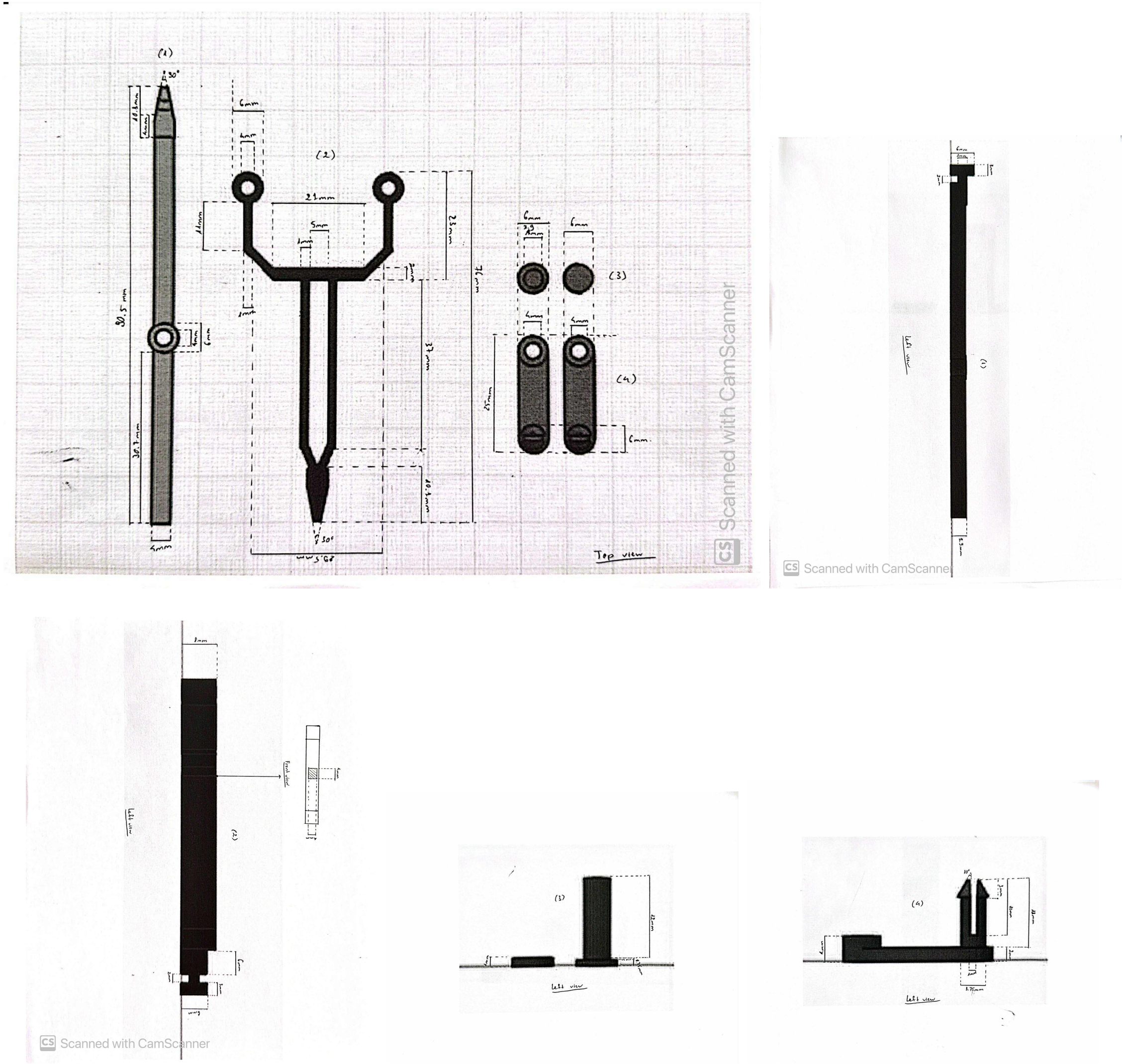


Figure 30: 3D-printed bistable link in short mode

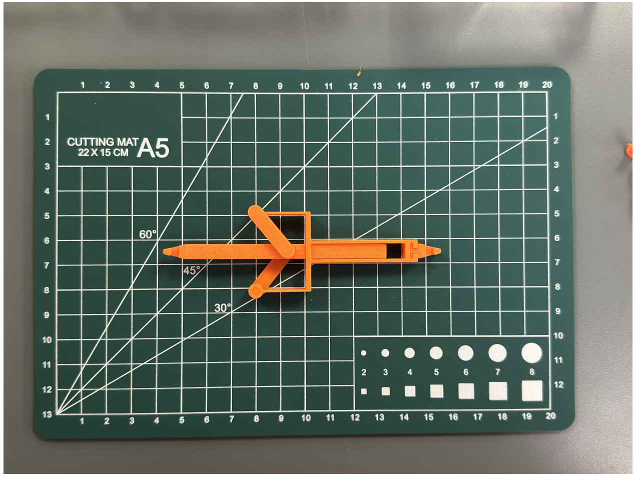
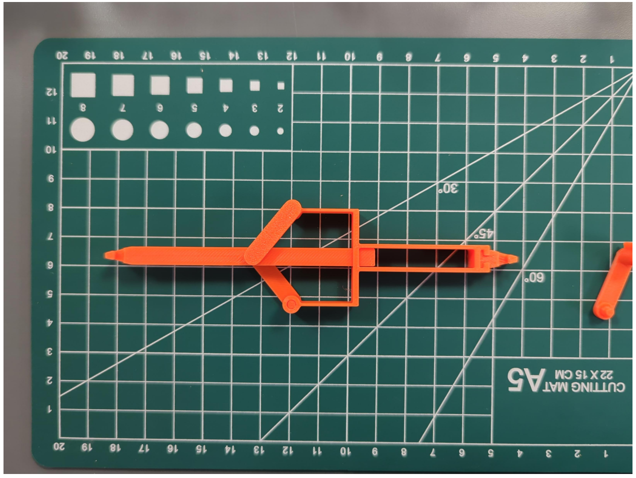


Figure 31: 3D-printed bistable link in long mode



6.2 Bistable Lattice

With the bistable links printed, we construct a triangular lattice in a hexagonal shape:

Figure 32: A lattice constructed by 3D-printed bistable links

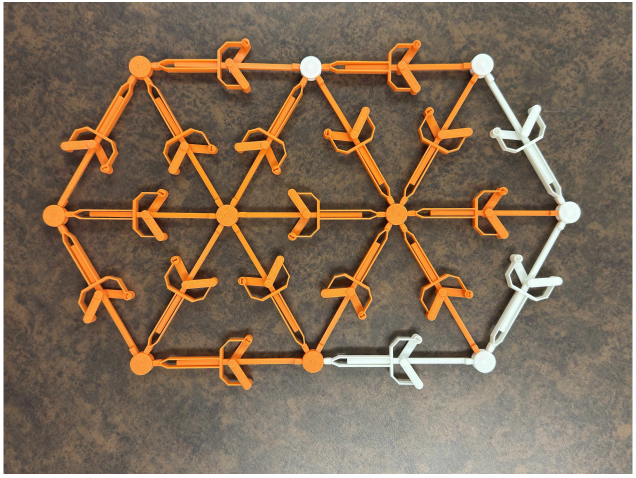
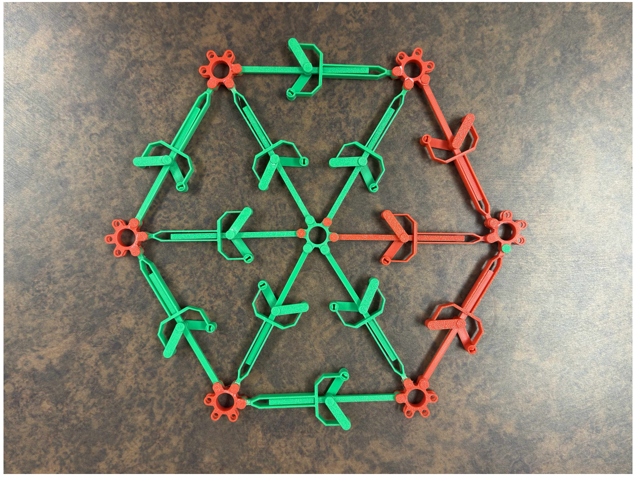
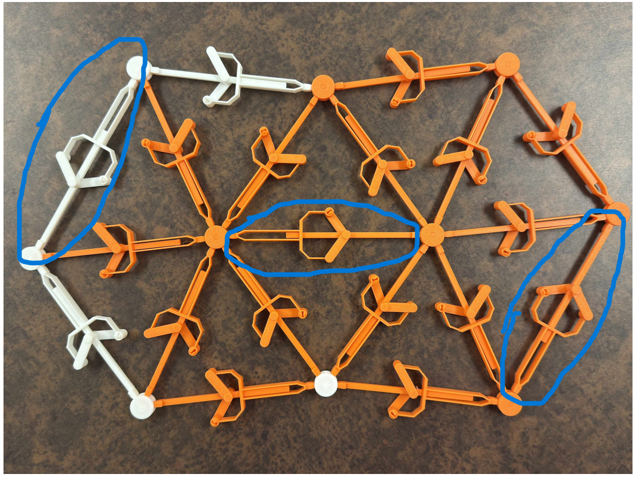


Figure 33: An alternative lattice constructed by 3D-printed bistable links



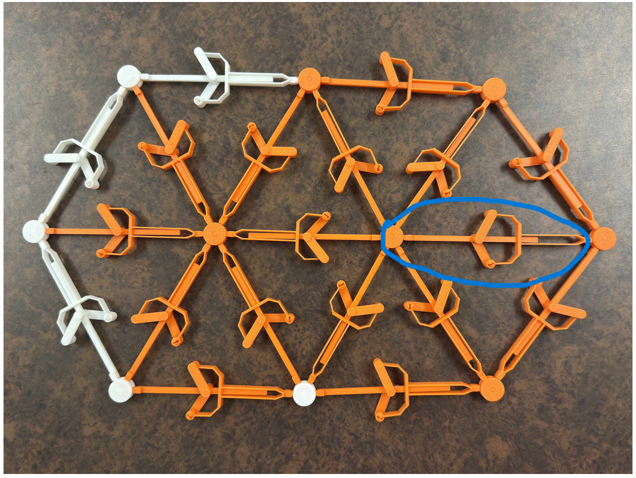
Notice that the two lattices have two different node structure. These are useful for the investigation of still states in a lattice. We begin the experiment with the first lattice by extending the very middle edge to its long mode of the first lattice.

Figure 34: 3D-printed lattice after displacing the middle edge



We can observe that the middle edge acts as a shared spoke for two hexagons. By Cherkaev's geometric condition, the total elongation of the rims must equal to that of the spokes in each hexagon. Therefore, as the mutual spoke is extended, each hexagon extends one of their rims in order to maintain total energy at 0. This configuration is what we consider as a valid still state. However, the mechanical flaw of this lattice manifests in some configurations.

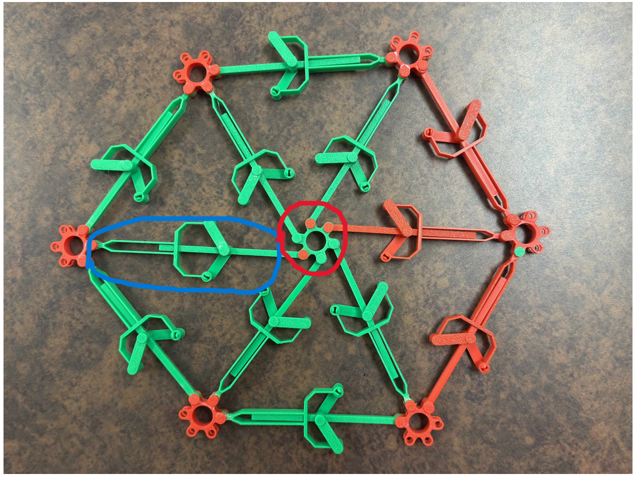
Figure 35: 3D-printed lattice after displacing the right-most spoke



This configuration, while apparently stable, is not a still state. By observation, the edges neighboring the extended one are pulled out by a small amount, meaning this current lattice has a potential energy value larger than 0. Hence, this cannot be considered a still state. We hypothesize that the distance between the long and short mode of each edge is not significant enough to force another edge to change when one violates Cherkaev's condition. Nevertheless, extending the distance will simultaneously widen the edge. Therefore, we opt for the second lattice design (*Figure 33*).

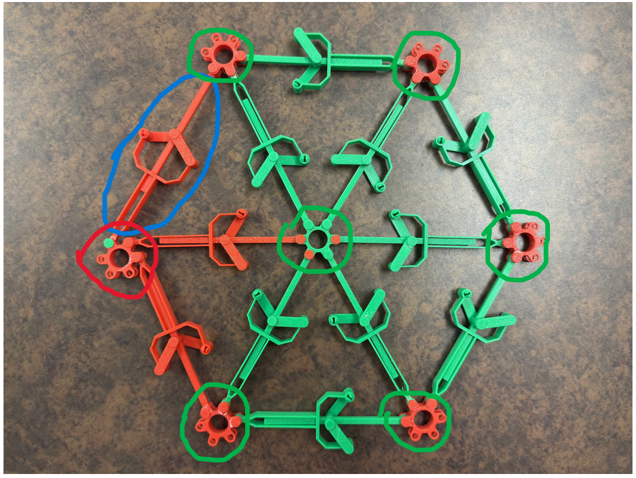
This lattice, instead of automatically attempt to correct itself to a still state in response to displacement such as the prior lattice, will indicate whether a configuration is in a valid equilibrium or not by the rotation of the nodes. We test this out by extending one spoke of the hexagon.

Figure 36: 3D-printed lattice after displacing the right-most spoke



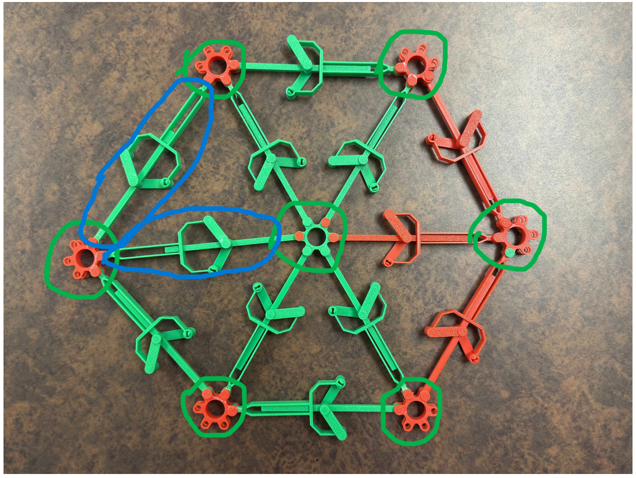
From the picture, the middle node is rotated, indicating that this is not a valid zero-energy configuration. We also attempt to extend a rim edge.

Figure 37: 3D-printed lattice after displacing the right-most spoke



We notice that a the left-most node is rotated, confirming the rim-spoke condition on lattice equilibrium. Ultimately, we extend one spoke and one rim edge.

Figure 38: 3D-printed lattice after displacing the right-most spoke



After confirming that no node is rotated, we conclude that this is a valid still state, again, confirming the established condition.

7 Discussion

7.1 The Double-well Issue of a Bistable Edge

Although bistable elements are known for having two distinct stable equilibria, a singular bistable edge often behaves similarly to a conventional elastic material when considered in isolation. In a typical energy model, the bistability arises from a double-well potential: the energy landscape has two minima (representing the short and long modes) separated by a local maximum. This maximum corresponds to an unstable equilibrium, effectively dividing the energy profile into two elastic regimes. When the applied strain on a bistable link lies between the first critical point (near zero) and the central maximum, the system will tend to relax back into the short stable state. In contrast, if the strain exceeds the central maximum and approaches the second critical point, the link will instead snap into the long (stretched) stable state. In either regime, the behavior is locally elastic—meaning small perturbations produce proportional restoring forces—but globally, the system remains nonlinear and discontinuous due to the presence of this unstable transition point.

This non-convexity of the energy landscape introduces significant challenges for mathematical optimization. Standard minimization routines, such as those used in the computational simulations, are typically gradient-based and local in nature. As a result, they are often "blind" to the existence of alternative stable configurations separated by an energy barrier. For example, if a minimizer starts from an initial strain that lies below the energy maximum, it will converge to the short mode without detecting the possibility of a long mode on the other side. Conversely, if the initial condition lies above the maximum, the optimizer will settle into the long mode, entirely disregarding the short configuration. This characteristic of bistable elements—appearing elastic within each stable basin but exhibiting global nonlinearity—complicates efforts to find global energy minima or to predict system behavior under varying loads. It underscores the need for more sophisticated optimization techniques or initialization strategies that can account for the full shape of the energy landscape, rather than merely following the local gradient.

Therefore, when considering the entire lattice composed of many bistable links, this local behavior becomes a global challenge: the minimization process frequently produces impractical or invalid configurations. The optimizer may converge to a state that appears locally stable for each individual link but is inconsistent with the expected structural or mechanical properties of the full system. It is, again, "blind" towards the other critical point because of the energy barrier. These configurations may violate global compatibility, symmetry, or boundary conditions, leading to results that are meaningless or structurally impossible on a 2 dimensional plane. This emphasizes the importance of global insight or constraint-guided methods when attempting to design or simulate bistable lattices which we have yet been able to account for in our minimization problem.

7.2 The Restriction on Degrees of Freedom by Connectivity

In discrete lattice systems, the connectivity of a node—defined by the number of edges it shares with neighboring nodes—plays a critical role in determining both the system's mechanical flexibility and the number of admissible still states (configurations with total energy value at 0). Higher connectivity imposes more geometric constraints, directly reducing the degrees of freedom of each node and edge. This is a direct consequence of constraint counting principles such as Maxwell's rule, which relates the number of constraints (edges) to the number of degrees of freedom in a mechanical network [1].

Mechanically, a node can only move within the limits imposed by its attached edges. As connectivity increases, the movement of any single node necessarily causes disturbance of multiple connected edges, which in turn affects neighboring nodes. Rather than simply adapting to this motion, neighboring nodes typically exert restorative (counteracting) forces to maintain their own equilibrium, thereby suppressing local movement and reinforcing rigidity.

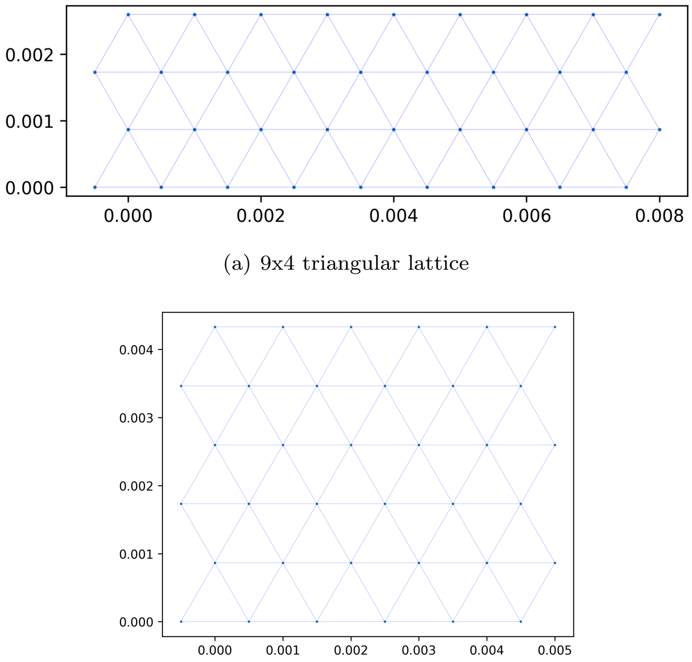
In contrast, under-constrained lattices (with low connectivity) provide greater freedom for individual node motion. This relaxed structure allows the lattice to accommodate deformations without significant energy cost. As a result, such systems exhibit increased mechanical flexibility and can allow for the lattice's transition into a larger number of still states, especially in bistable lattices, where each link may rest in two different equilibrium positions [4].

This effect becomes particularly pronounced in multi-stable systems in general, bistable in specific. In highly connected regions, nodes act as mechanical anchors, enforcing geometric compatibility across adjacent links. This compatibility filters out combinations of bistable configurations that, while locally stable, are globally incompatible in terms of geometry. Furthermore, if one edge flips to a different equilibrium, the resulting internal stresses from neighboring elements may be strong enough to prevent relaxation, leading to geometrically compatible but energetically unfavorable states. Ultimately, this suppresses configurational diversity, and the number of valid still states decreases sharply with increasing connectivity.

In one instance we provide above where we attempt to count the number of still states in a hexagonal subsection of a triangular lattice, this inverse relationship between connectivity and number of still states manifest in the rim-spoke rule. The rule states that for a hexagonal lattice, the total elongation of the rims (edges on the rim of the hexagon) is equal to the total elongation of the spokes (edges inside the hexagon) [2]. Therefore, the number of possible zero-energy configurations are limited, exponentially smaller than the theoretical 2^n value. This rule derives from the restrictions each edges put on another, in this case, caused by the heavy connectivity of a triangular lattice.

7.3 The Effect of Organization

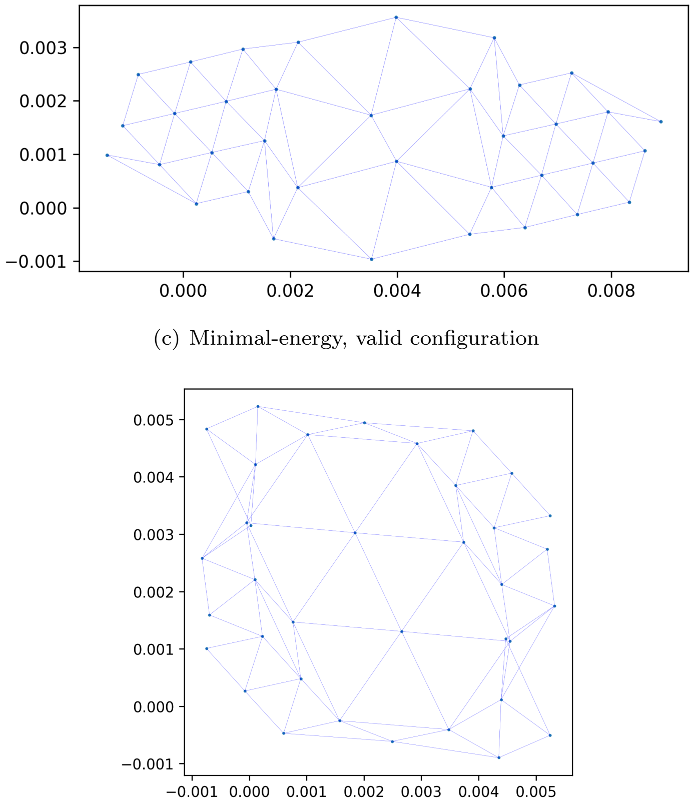
We have also found that different lattice overall shapes show distinctive behavior under the same displacement. To demonstrate, we generate two triangular lattices consisting of 36 nodes: one is a 9×4 lattice, the other is a 6×6 lattice.



(a) 9x4 triangular lattice

(b) 6x6 triangular lattice

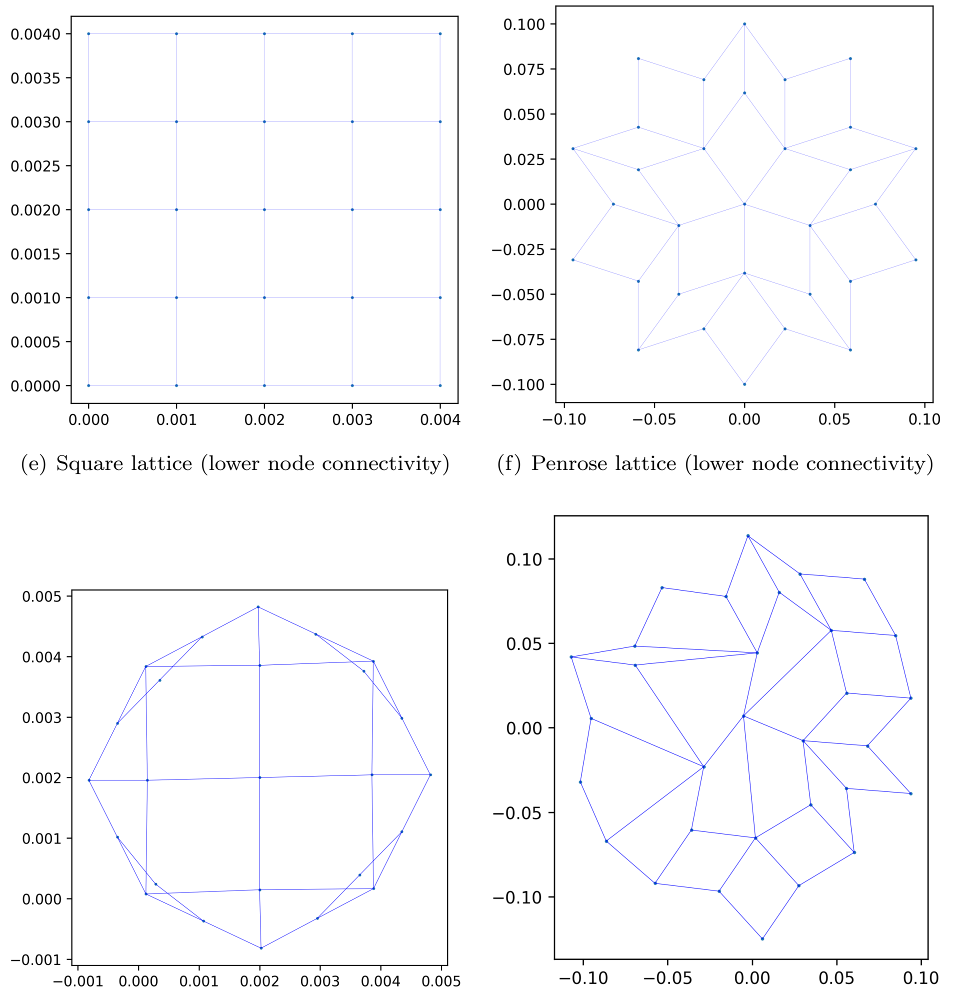
Both lattices are then subject to an initial displacement of a Gaussian displacement at 0.7 magnitude. We run the BFGS minimizer on the two structures, obtaining the following results:



(c) Minimal-energy, valid configuration

(d) Minimal-energy, invalid configuration

We hypothesize that the difference in results (despite both being a triangular lattice) is caused by how the nodes are organized in a lattice. Observing the two lattices, we notice that the layout of the nodes contributes to the difference in the number of nodes inside the structure and those on the rim of the lattice between the two structures. This, in turn, changes the number of interconnected hexagons (sub-sectional hexagons which share edges and nodes with another), which ultimately affects the connectivity of the lattice. It can also be observed that the 9 × 4 lattice has fewer edges than the 6 × 6 lattice. Nevertheless, we also observe more deviations with different lattices under the same displacement:



(e) Square lattice (lower node connectivity)

(f) Penrose lattice (lower node connectivity)

(g) Square lattice after Gaussian displacement at magnitude 0.7 (invalid)

(h) Penrose lattice after Gaussian displacement at magnitude 0.7 (valid)

These results show that we cannot expect lattices with more still states to produce a minimal-energy configuration from an initial displacement with which a more restricted lattice can. This underscores that the geometric layout of a lattice is also an active determinant of its mechanical response to different displacements.

7.4 Existence of Non-zero Minimal-energy States

Throughout our research, one of our by-products worth looking into are minimal-energy states that are non-zero. These are the results from the minimization schemes, whose total energy is at a local minima, but does not equal 0. This indicates that these configurations are at equilibrium, however, still possesses energy. Here are some examples:

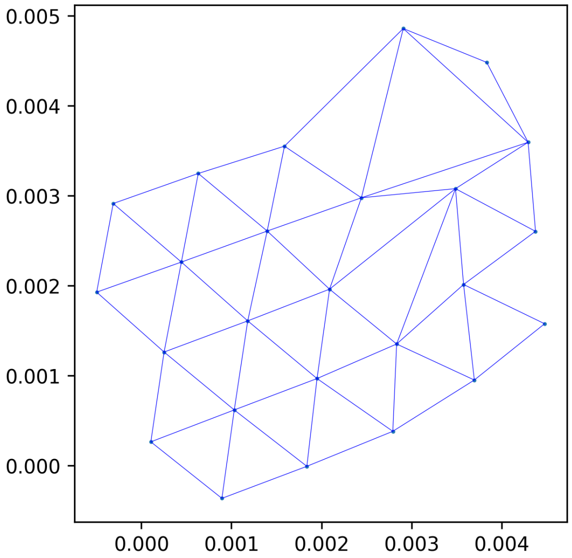


Figure 39: Total energy: 0.0535732.

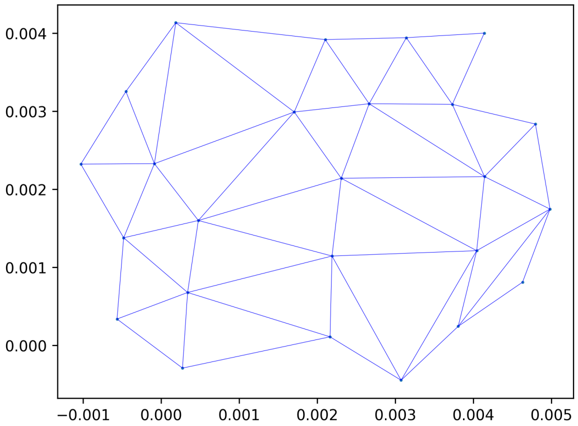


Figure 40: Total energy: 0.12184345.

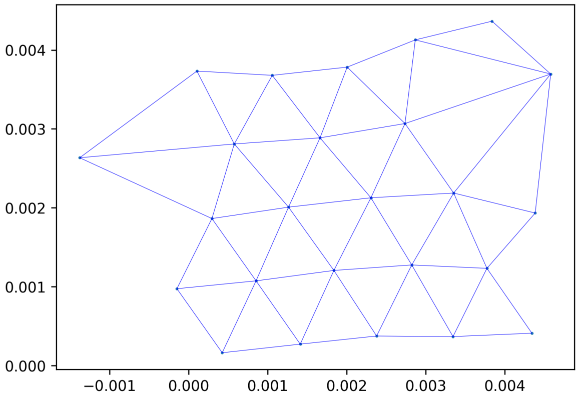


Figure 41: Total energy: 0.090107329.

Although these configurations are not the focus point of the research, their existence is unexpected and intriguing. The team hopes that there will be further investigation into this in future research.

8 Conclusion

Our research revealed many different characteristics of bistable lattices that provide context for future research. Numerical simulations confirmed that when a force deforms a lattice and increase its energy, the structure will then settle into its nearest minimal-energy state, minimizing the total energy of the lattice. These simulations also provide evidence that there exist minimal energy states where the total energy of the lattice is non-zero. We developed counting arguments to quantify a triangular lattice's total still states based on the lattice's geometric constraints, and for triangular lattices with identical, overlapping hexagonal subsections, we have shown that stillness and compatibility on the micro scale necessarily results in the the same at the macro scale.

From these results, many future research questions emerge. The existence of non-zero minimal energy states warrants further investigation; our numerical minimization techniques suggest this to be the result of a deformed lattice being closest to an invalid 0-energy configuration, but more evidence is needed to draw this conclusion. It is also worthwhile to develop more generalized counting arguments that would allow us to sum all still states of any lattice. This could involve generalizing our counting argument to apply to any triangular lattice and developing similar arguments for square and Penrose lattices, or alternatively one could develop a technique based on the effects of a lattice's organization and connectivity. It could also be advantageous investigate whether we can draw more conclusions about the lattice at certain deformations from the Hessian for our energy function and its eigenvalues.

All of these research directions would help to further our understanding of bistable lattice behavior, not only helping the mathematicians who work with them but also those in any industry that benefits from understanding the model. Therefore, investigations into lattice behavior for real-world applications should be framework-specific, taking into account the connectivity, structural geometry, and contextual displacement (force applied) in an effort to stay grounded in its applied context.

References

[1] C. R. Calladine. Buckminster fuller's ”tensegrity” structures and clerk maxwell's rules for the construction of stiff frames. *International Journal of Solids and Structures*, 14(2):161–172, 1978.

[2] A. Cherkaev, A. Kouznetsov, and A. Panchenko. Still states of bistable lattices, compatibility, and phase transition. *Springer Nature*, 22:421–444, 2010.

[3] P. Ducarme, M. van Hecke, and J. T. Overvelde. Exotic mechanical properties enabled by countersnapping instabilities. *Proceedings of the National Academy of Sciences of the United States of America*, 122(16), 2025.

[4] C. L. Kane and T. C. Lubensky. Topological boundary modes in isostatic lattices. *Nature Physics*, 10(1):39–45, 2014. doi: 10.1038/nphys2835.

[5] D. Kraft. A software package for sequential quadratic programming. Technical Report DFVLR-FB 88-28, DLR German Aerospace Center, Institute for Flight Mechanics, Köln, Germany, 1988.

[6] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

[7] C. Zhang, X. Yin, R. Chen, K. Ju, Y. Hao, T. Wu, J. Sun, H. Yang, and Y. Xu. A review on reprogrammable bistable structures. *Smart Materials and Structures*, 33, 2024.