Combinatorial and Geometric Structure in the Self-Assembly of Polyhedra

by

Daniel C. L. Johnson

B.S., Rensselaer Polytechnic Institute; Troy, NY, 2009

Sc.M., Brown University; Providence, RI, 2012

A dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy
in The Division of Applied Mathematics at Brown University

PROVIDENCE, RHODE ISLAND

May 2015

This dissertation by Daniel C. L. Johnson is accepted in its present form by The Division of Applied Mathematics as satisfying the dissertation requirement for the degree of Doctor of Philosophy.

Date_____ _____

Govind Menon, Ph.D., Advisor

Recommended to the Graduate Council

Date_____ _____

Basilis Gidas, Ph.D., Reader

Date_____ _____

Miranda Holmes-Cerfon, Ph.D., Reader

Approved by the Graduate Council

Date_____ _____

Peter Weber, Dean of the Graduate School

Vitae

Personal Information

American citizen.

Born November 6^{th}, 1987 in Rochester, New York.

Education

Brown University, Providence, Rhode Island USA

Sc.M., Applied Mathematics, May 2012

Rensselaer Polytechnic Institute, Troy, New York USA

B.S., Mathematics, May 2009

Publications

Dynamics of the exponential integrate-and-fire model with slow currents and adaptation, Victor Barranca, Daniel Johnson, Jennifer Moyher, Joshua Sauppe, Maxim Shkarayev, Gregor Kovacic, and David Cai. Journal of Computational Neuroscience,

2014.

Self-assembly of mesoscale isomers: The role of pathways and degrees of freedom, Shivendra Pandey, Daniel Johnson, Ryan Kaplan, Joseph Klobusicky, Govind Menon, and David Gracias. PLoS ONE, 2014.

Acknowledgements

There are many people that have made this journey possible. I am forever indebted to them for their support, kindness, love, and encouragement. Though I couldn't possibly express my true gratitude to everyone that helped me get to where I am, I would like to extend thanks to the following people:

To Govind Menon, for agreeing to take me as a student three years ago. I have learned a lot and had my mathematical and scientific horizons pushed in directions that I would have never expected. Your guidance made this possible.

To Basilis Gidas, committee member, intramural soccer captain, and friend. Your broad expertise on everything from probability theory to molecular biology provided a vital perspective. Playing on the applied math intramural football team with you in my six years at Brown helped keep me sane and left me with lasting memories.

To Miranda Holmes-Cerfon, for serving on my committee. I've enjoyed our conversations and they have changed how I think about much of the mathematics in this work. Your authority on the subject matter helped me identify ways in which it could be improved.

To others in the Division of Applied Mathematics. The environment that the people here have created has fostered my pursuit of mathematical knowledge. I would like to thank Stu Geman, Matt Harrison, and the rest of the Pattern Theory group for

many conversations, seminars, and ideas. To Carly Klivans, for our conversations on polytopes and geometry. Thanks to all of my friends in the division and particularly Ely, Erich, Joe, Kelly, and Laura who started graduate school with me six years ago. Also Nat, for buying me coffee today.

To my family, for their never ending belief. Mom, thank you for instilling my philosophical outlook that was necessary throughout my education. Dad, thanks for exposing me to numbers, math, and engineering from a young age. To my extended group of parents, Cathy, John, Melissa, Steven, Grandparents, Uncles, and Aunts, for always being there for me. To my siblings Chellie, Hannah, and Yolie; I love you all.

Finally, to Stasya, for being my support and partner in life. I could not have persevered in the tough moments without you by my side. This victory is as much yours as it is mine.

Abstract of "Combinatorial and Geometric Structure in the Self-Assembly of Polyhedra " by Daniel C. L. Johnson, Ph.D., Brown University, May 2015

We consider a discrete attachment model for the self-assembly of polyhedra called the Building Game. The combinatorial configuration space of the model is computed for the Polyhedra in the Platonic, Archimedean, and Catalan solids of up to 30 faces and several novel enumerative results are generated. Further, we extend the Building Game to include geometric information in the form of a polygonal linkage. Using systems of constraints to specify the linkage, the geometric configuration space is given as a solution manifold. A random walk scheme is used to simulate reflected Brownian motion on the geometric configuration space manifolds. The combinatorial and geometric aspects of the Building Game are combined and used in a Markov process model for measuring statistics of the self-assembly process.

Contents

List of Tables

List of Figures

Chapter One

Introduction

Self-assembly is a class of formation process in which a product is constructed without explicit manipulation of its individual parts: many—often identical—parts come together by utilizing the dynamics of their environment to create the finished structure. Such processes can often be encouraged by manipulating broadly controlled parameters of the assembly environment such as temperature, solution content, and assembly subunit selection.

There are many examples of both natural and synthetic self-assembly process that take a wide variety of length scales and serve a plethora of functions. An area of much active research is the self-assembly of RNA, proteins, and viral capsids. These biological processes act on the nano scale and, while it is known that their formation can be aided by a variety of secondary mechanisms, the formation process is not well understood in general. Synthetic examples of molecular self assembly include the formation of supramolecular cages. These cages, composed of many subunit molecules, are large enough to encapsulate smaller molecules and show promise in contributing to a number of medical and scientific fields including drug delivery and nano-scale circuits.

In general, the processes of biological self-assembly are not well understood due to difficulties arising from their small length scale. Conversely, synthetic self-assembly processes often lack the complexity and sophistication of their biological equivalents. In both cases, central themes of self-assembly are similar. Of primary importance is identifying the pathways of formation which consist of a specific order in which unfinished intermediate states are visited before the assembly is completed. It is thought that these pathways are often robust in the sense that the process follows a very small number of pathways to form the end product, despite an overwhelmingly large number of distinct pathways that are possible. It is also of interest to identify mechanisms by which failed or flawed formations can be avoided or minimized. Pos-

sible strategies include the selection of specific precursors, the selection of solution, and the mid-assembly control of experimental parameters.

The goal of this work is to provide a framework in which the interplay between the combinatorial and geometric aspects of self-assembly processes can be analyzed.

1.1 Polyhedra

We focus on the self-assembly of polyhedral structures for a number of reasons. First, there is a rich history of using polyhedra as representative models for molecular structure [26]. Whether it is treating carbon atoms like tetrahedra or the C_{60} Buckminsterfullerene as a truncated icosahedron, it is natural to model the connectivity structure of molecules using polyhedra [7]. One could view each molecular subunit (carbon atom in the case of C_{60}) as a vertex and each subunit to subunit bond as an edge to define the structure as a polyhedron. Sometimes the dual perspective–with each subunit being a polyhedral face and edges existing between bonded subunits–is more useful.

Aside from the clear connection to molecular structure, polyhedra are also mathematical objects with a rich literature and known properties. From the ancient Greeks to the present day, polyhedra have been a cultural fixture for thousands of years as evidenced in the diverse fields of architecture, computer graphics, and board games. As three dimensional structures composed of two-dimensional faces, the construction and assembly of polyhedra is easier and more reliable than other three dimensional alternatives.

There are a few fundamental classes of polyhedra we consider throughout this

work. The Platonic solids, composed of the tetrahedron, cube, octahedron, dodecahedron, and icosahedron are the five polyhedra that are composed of only one type of regular polygon. The tetrahedron is self-dual with the cube-octahedron and dodecahedron-icosahedron pairs dual to each other. The Archimedean solids are defined by relaxing the Platonic conditions to allow its polyhedra to be composed of two or more regular polygons. With a total of thirteen member polyhedra, the Archimedean solids also have the property that each vertex has identical connectivity properties. The Catalan solids are Dual to the Archimedean solids. Also composed of 13 polyhedra, each Catalan solid has only one type of face, but that polygon is not regular [3].

The theory of polytopes provides a d-dimensional generalization of polyhedra. While we restrict our focus to 3-dimensional polytopes, the mathematical structure of polytope theory is at times useful. In particular, a 0-dimensional polytope is a point, a 1-dimensional polytope is an edge, a 2-dimensional polytope is a polygon and a 3-dimensional polytope is a polyhedron. A polyhedron along with all of its 0, 1, and 2-dimensional substructures is called a polytopal complex [32].

1.2 Scientific Motivation

Polyhedra play in important role in many organic and synthetic self-assembly processes. From being a model for molecular subunits to representing an assembled spherical product, polyhedra form a natural construct for describing a variety of these processes. We present a number of scientific applications that motivate the interplay between polyhedra and self-assembly [24].

1.2.1 Self-Folding Polyhedra

In experiment, Pandey et al. have been able to form closed sub-millimeter scale polyhedral structures from the self-folding of flat polyhedral nets [22]. Using photolithography, these nets are cut from a two dimensional sheet with great precision. By adding a specific amount of solder at the net's hinges, surface tension from the melted solder causes the net to fold up into the closed polyhedron. Several different polyhedra have been attempted with varying degrees of success. The geometric structure of each net was shown to have a major influence on its propensity to successfully assemble into the polyhedron. In some cases, two distinct polyhedral isomers can be formed from the same net [23].

1.2.2 Molecular Cages

Self-assembly of molecular cages are the subject of much active research. By isolating individual molecules of interest inside a molecular cage, targeted application of these molecules is theoretically possible. This has enormous implications in the medical industry. For example, if a drug molecule can be encapsulated in such a cage and released when the cage is at a desired location in the body, side effects caused by the drug interacting with other parts of the body can be reduced. Other medical uses are foreseeable given that these molecular cages provide a framework for self-assembling structures on the length scale of proteins. Additionally, by creating a grid-like network of such cages, it may be possible to construct functional electric circuitry on the nano scale [29].

Fujita et. al. have theorized and subsequently synthesized a family of organometallic cages consisting of metallic connector molecules (M) that each connect four bent

ligand molecules (L). With the M molecules as vertices and the L molecules as edges, they can form polyhedral cages. Due to geometric constraints, the number of M and L molecules in a completed cage must satisfy $|M| = k$ and $|L| = 2k$ for $k \in 6, 12, 24, 30, 60$. Each of these five choice of k results in a cage that geometrically resembles a specific Archemedian solid [15].

In one experiment, two different ligands (L_1 and L_2) with slightly different bend angles were used. As the ratio of $L_1 : L_2$ was varied, each ratio only resulted in the formation of one of the possible cages. More specifically, for ratios $L_1 : L_2 < 0.25$ only $M_{12}L_{24}$ formed and for $L_1 : L_2 > 0.25$ only $M_{24}L_{48}$ was formed. Explanation for this steep and curious cutoff remains an open problem that mathematical approaches could shed light on.

In experiments by Liu et al, polyhedral supramolecular cages made from different molecular species were synthesized [17]. Theoretically, a tiling pattern of the molecular species is chemically possible, but it was not observed experimentally. Additional experiments in which copies of two types of molecular species could be combined to form two different polyhedral cages were proposed. It is unknown if both potential polyhedral cages would form or if one would be significantly more favorable over the other. Being able to predict the results of such experiments via mathematical analysis and simulation would be a significant contribution toward optimizing strategies for the self-assembly of supramolecular cages.

1.2.3 Viral Capsid Assembly

A large class of biological viruses have a capsid with icosahedral structure. With clear public health implications, understanding how viruses self-assemble is impor-

tant. This topic has been well studied and a wide variety of mechanisms have been evidenced to contribute, but there still is no general understanding of the process in which a virus' capsid is formed [1]. Viral capsids come in many shapes and sizes, but a significant portion of them are icosahedral in structure. The seminal work of Caspar and Klug identified this connection between polyhedra and viral capsids, and continues to inform research on the process in which they are formed [2]. Understanding the pathways by which a viral capsid forms may be key in preventing or slowing down replication in future medical research.

1.2.4 RNA and Protein Folding

Protein and RNA folding are active fields of research. If we can predict the three dimensional structure of a RNA or protein based on its amino acid sequence, we will know more about its biological function. As many human and animal disorders are caused by proteins folding abnormally, their cures may lie in the understanding of the folding pathway. While we do not directly consider applications relating to RNA and protein folding, success with the above problems may also provide insights for this complex topic [16].

1.3 Mathematical Constructs for Self-Assembly

Discrete geometric models for self-assembly are our primary focus. These models often decompose the self-assembly process into many intermediate steps. Organizing the space of these intermediate configurations as a graph, successful assemblies are paths through this graph. Using the mathematical structure of the graph, a quan-

titative analysis of the process is possible. The following are different mathematical approaches to self assembly. The attachment and folding models employ the graph enumeration strategy, but others are more qualitative.

1.3.1 Attachment

First proposed by Wales, the Building Game is a discrete attachment model that simulates the sequential construction of polyhedral structures [30]. Subsequent work by Zlotnick modified the original model and used it to examine icosahedral viral capsid assembly [33, 6]. Much of our work revolves around the Building Game and it is explained in great depth in the next chapter.

1.3.2 Folding

Another model–we refer to as the Folding Model–was used to model the self-folding of meso-scale metallic polyhedra mentioned in section 1.2.1. In this model, a polyhedron's net is sequentially folded up into the finished polyhedron. At each stage one vertex is closed by a folding move. In analogy with the Building Game, we can define folding intermediates to be partially folded states and a state spaces where intermediates are connected if one can form the other by folding to close an additional vertex. One possible sequence of fold resulting in the octahedron is shown in figure 1.1.

Interestingly, each polyhedron may has multiple nets. In fact, the number of distinct nets grows rapidly with the size of the polyhedron. Certain nets have different folding pathways and measuring the favorability of one of these nets over

another introduces an design problem. If nets that fold into the completed polyhedron most reliably can be identified, the efficiency of the self-assembly process can be optimized [22, 23]

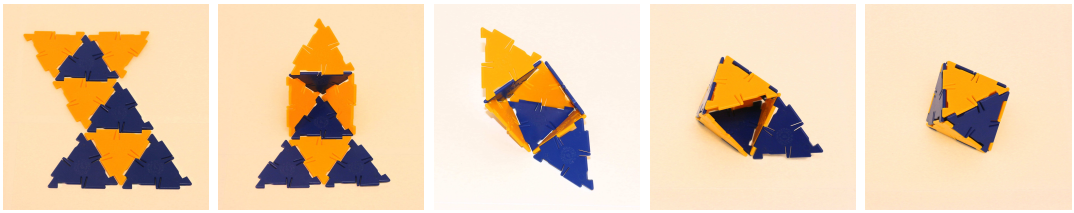

Figure 1.1: A sequence of Folding Model intermediates beginning with an octahedron net and successfully forming an octahedron.

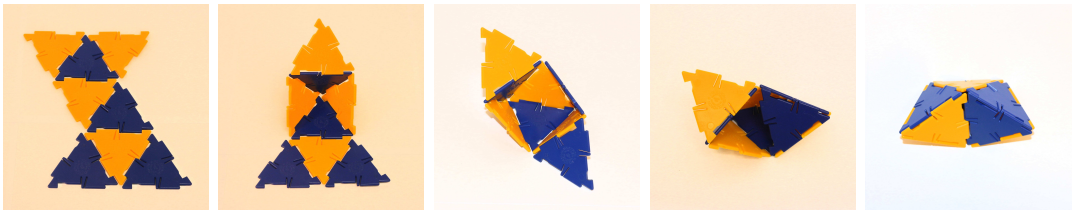

Figure 1.2: A sequence of Folding Model intermediates beginning with an octahedron net and resulting in the boat configuration.

The Folding Model allows folding sequences that do not result in the initially intended polyhedron and can terminate in other polyhedra and blocked states. For example, many of the intermediates that cannot fold into the octahedron end up folding into a non-convex boat intermediate. This is depicted in figure 1.2. This explicit modeling of misfoldings is an important feature of the Folding Model since it allows for a mathematical treatment of the robustness of successful folding relative to other possible outcomes.

1.3.3 Local Rules

Several previously studied models for self-assembly can be classified as *local rules* based approaches. Such models typically consist of a collection of components that can be combined according to a specified grammar. While this is a basic idea,

the variety of possible component and grammars makes this a widely flexible class of models. Schwartz et al used such a model to describe the assembly of viral capsids [1, 27]. These capsids are composed of proteins that can each assume a number of conformations; by putting a grammar on the ways in which proteins of different conformation can combine, they were able to successfully form a variety of viral capsids in simulation. While this class of models is powerful, we have not actively pursued any such models. Introducing a local rules approach might allow the Building Game to be generalized in a way that allows formation errors as in the Folding Model.

1.3.4 Energy Landscapes

Due to the scale of many of these self-assembly processes, statistical mechanics provides a natural framework for thinking about these models. Each intermediate configuration of self-assembly processes is a local minimum of the potential energy landscape and the Gibbs distribution provides the stationary measure for the process. Dynamics of the process are dictated by energy barriers that must be overcome for the process to transition between potential energy minima.

This idea carries over to our discrete model. Every intermediate is treated as a local minimum and assigned an energy level. Graph connections are given energy barrier heights that must be overcome. The dynamics of discrete process are modeled using a reversible Markov process with transition rates proportional to energy barrier differences. With this flexible mathematical machinery, a wide variety of questions can be asked: from pathway probabilities to formation related stopping times.

Wales et al. introduced a graphical way to represent the structure of a potential

surface possessing many local minima and intermediary transition states [31, 4]. Named a *discontinuity graph*, the tree with leaves representing local minima of the potential surface and other connecting nodes representing transition states that the minima can be reached from. This provides a way of picturing the structure of the a potentially high dimensional energy landscape that looks past simple energy differences. The vertical height of each node is used to represent the potential energy of that state. Horizontally, the tree is organized to partition these local minima into corresponding funnels. If the graph is truncated at a specific energy level, two minima are reachable via transitions to states strictly below this truncation energy if and only if they remain connected in the truncated disconnectivity tree.

While not a quantitative way of assessing a model's pathways, disconnectivity graphs are extremely useful in identifying broad qualitative properties and can be instrumental in gaining insight into a model's dynamics. The use of such graphical and otherwise qualitative techniques should not be overlooked as they can inspire techniques for more quantitative analysis.

1.4 Original Contributions

The Building Game specifies a set of finite intermediate states that may be observed in the successful self-assembly of a polyhedron. We provide a formal mathematical definition for the Building Game and organize the set of allowed intermediates into a graph called the combinatorial configuration space. We have enumerated the combinatorial configuration space for all Platonic, Archimedean, and Catalan solids up to 30 faces. This constitutes the most comprehensive enumeration of the Building Game to date. Graph statistics for these computations are presented in table 3.1.

The relation between the Building Game and the mathematical concept of a shelling is established. The shellable subgraphs of combinatorial configuration spaces are computed with statistics recorded in table 3.2. Finally, the shellable subgraphs are used to calculate the number of shellings for each polyhedra as seen in table 3.3. We believe these to be previously unknown quantities.

Each Building Game intermediate is defined by its combinatorial structure, but by treating it as a linkage in three dimensions its definition is extended in a geometric sense. Concatenating the 3-dimensional vertex locations, each geometric configuration is specified as a point in an n-dimensional space. Since each geometric configuration must satisfy certain combinatorial and geometric properties, we enforce these conditions by using a set of algebraic constraint equations. The solution set of these constraint equations specifies an algebraic variety in the n-dimensional ambient space and the dimension of this variety gives information about the mobility of the intermediate. To examine the geometric configuration space more thoroughly, we consider a reflected Brownian motion on the variety. Boundaries here are added to preclude the geometric configuration from intersecting itself when embedded in three dimensions. A random walk rejection sampling scheme is used to sample paths from the reflected Brownian motion. These samples allow us to address scientific questions pertaining to self-assembly. Firstly, the role of rigidity in the self-assembly intermediates is explored both in terms of degrees of freedom and with more nuanced mobility statistics. Additionally, our framework allows for a natural method for extracting transition rates between intermediate, a task that is difficult to perform experimentally. The combinatorial configuration space for the cube is illustrated in figure 2.6 and different geometric configurations for one of the octahedron intermediates is shown in figure 4.1.

In this work we have shown ways in which combinatorial and geometric structure

of self-assembly intermediates are related. We model the self-assembly process as a Markov process using the Building Game combinatorial configuration space. It is the addition of geometric information that allows us to define transition rates in a physically meaningful way. The combination of these two perspectives allows for a more complete understanding of the underlying processes than either could provide individually.

Chapter Two

The Building Game: Modeling

2.1 The Building Game as a Mathematical Framework for Self-assembly

The Building Game was first introduced by Wales [30] as a model for the formation of closo-boranes. Zlotnick subsequently altered the Building Game and used it as a model for the assembly of polyhedral viral capsids [33]. In the model, a capsid is idealized as a polyhedron with each face treated as a subunit. Assembly proceeds from a single face with a second face attached to the first along an edge. At each subsequent step of the process, an additional face is added along an edge of one of the already added faces. The process ends when all of the faces have been added resulting in a completed polyhedron.

A useful way to think about the Building Game is as a sequential coloring process. Given a polyhedron with each face painted white, choose a face and paint it black. At each subsequent step, choose a white face that is adjacent to a black face and paint it black. Repeat until all faces are black. Here the black faces represent a face being present at a given step of the assembly process.

2.2 Formal Definition

We formalize the Building Game in terms of the group action of the polyhedron's rotation group G acting on subsets of F, the polyhedron's face set. Using the polyhedron's dual graph $\mathfrak{G}(F, E)$ whose nodes are the faces F of the polyhedron and connections E correspond to the pairs of faces sharing an edge we can define the mathematical structures that comprise different Building Game configurations.

Definition 1. *A Building Game* ***state*** $x \subset F$ *is a non-empty subset of the faces* F *of a polyhedron such that the sub-graph of* $\mathfrak{G}$ *restricted to* x, $\mathfrak{G}|_x$, *is a single connected component. As a result, every state can be formed through the Building Game attachment process.*

Schlegel diagrams are two dimensional projections of the three dimensional polyhedron [28]. By coloring different subsets of faces, it is often useful to depict Building Game states with Schlegel diagrams. Figure 2.1 uses Schlegel diagrams to depict all of the states of the tetrahedron. Since every face of the tetrahedron is adjacent to every other face, any non-empty subset of faces is a Building Game state. Thus, there are $\binom{4}{k}$ tetrahedron states that have k faces and $2^4 - 1 = 15$ states in total. As pictured in figure 2.3, some subsets of faces are not states. For instance, the

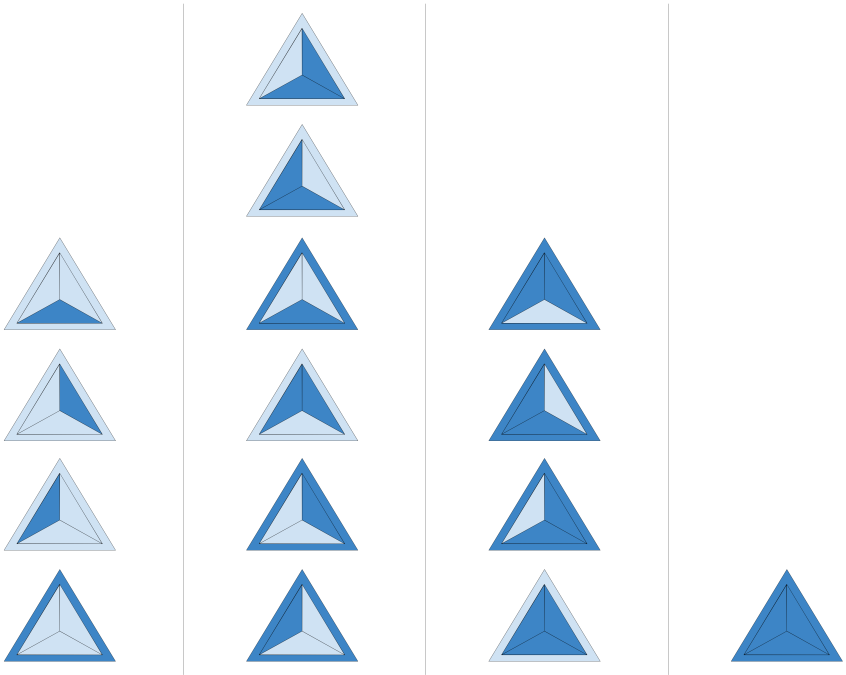

Figure 2.1: The Building Game states of the tetrahedron.

subset of octahedron faces with only two faces that are not adjacent is not a state.

Similarly, the subset of two faces meeting only at a vertex is not a state as they are not connected through edge adjacency in the graph $\mathfrak{G}$. However, when more faces are added to connect these faces in $\mathfrak{G}$ the subset is indeed a state.

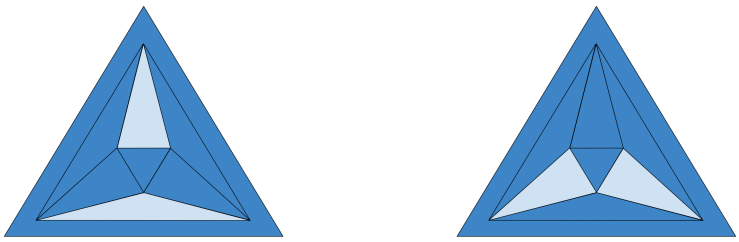

Figure 2.2: Examples of octahedron states.

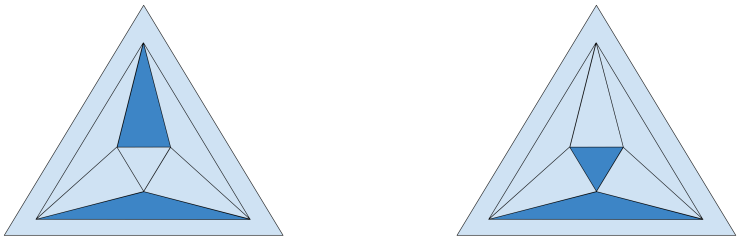

Figure 2.3: Examples of octahedron non-states.

2.2.1 Group Actions

It is easy to see that many states are combinatorially equivalent and are just rotations of each other. As in Endres et al., we group the states into sets that are rotations of each other [6]. However, to do so, we must first build the mathematical infrastructure using group actions.

Definition 2. *A* ***group action*** *of a group G on a set X is a function mapping $G \times X$ to X with $(g, x) \mapsto g.x$ that satisfies: (i) $(gh).x = g.(h.x)$ for $g, h \in G$ and (ii) $e.x = x$ for e the identity element of G [25].*

We typically use a polyhedron's rotation group or one of its subgroups as G and 2^F, the set of subsets of F, as the set X that G acts on. In this case, the action $g.x$ permutes the faces of the polyhedron according to the element g of the rotation group and results in a new subset of faces. Here, we introduce the concepts of orbits and stabilizer subgroups as they play an important role in our later analyses.

Definition 3. *Let G be a group acting on a set X, the **orbit** of an element $x \in X$ is the subset $G.x \doteq \{g.x : g \in G\}$ of X [25].*

We also use the shorthand notation $[x]$ to refer to the orbit $G.x$ since an orbit can be thought of as an equivalence class under the relation $x \sim \hat{x}$ if there is a $g \in G$ such that $x = g.\hat{x}$.

Definition 4. *For a group G acting on a set X, the **stabilizer subgroup** for an element $x \in X$ is the subgroup $G_x \doteq \{g \in G : g.x = x\}$ of G that fixes x [25].*

Now, we introduce three classical results of group actions that relate orbits and stabilizer subgroups and add a corollary that we will use frequently.

Theorem 1 (Orbit-Stabilizer [25])**.** *Let G be a group acting on a set X, then for any $x \in X, |G.x| = [G : G_x]$, the number of left-cosets of G_x.*

Theorem 2 (Lagrange [25])**.** *If G is a finite group and $S \leq G$, then $|S|$ divides $|G|$ and $[G : S] = |G|/|S|$*

Lemma 1 (Burnside [25])**.** *Let G be a finite group acting on a set X, then*

$$|X/G| = \frac{1}{|G|} \sum_{g \in G} |X^g|$$

where $|X/G|$ is the number of orbits and $|X^g| = \{x \in X : g.x = x\}$

Corollary 1. *Let G be a finite group acting on a set X, then for any $x \in X$*

$$|G.x| = \frac{|G|}{|G_x|}.$$

Proof. This result follows trivially from the Orbit-Stabilizer and Lagrange's theorems. □

2.2.2 Building Game Intermediates

With this framework of group actions, we now formally define the principal unit of the Building Game.

Definition 5. *A Building Game* ***intermediate*** $[x] \doteq \{g.x : g \in G\}$ *is the orbit of the state x under the polyhedra's rotation group.*

Since the orbits of a group action form a partition on the set it acts on, each state belongs to a single intermediate and two states x and y are part of the same intermediate if there is a $g \in G$ such that $y = g.x$. In the case of the tetrahedron, as pictured in figure 2.1, there are only four intermediates since any state with the same number of faces can be rotated to reach the others.

Since we have defined the intermediates to be the orbits of states under the polyhedral group, we are naturally also interested in stabilizer subgroups of Building Games states.

Definition 6. *The* ***symmetry number*** r_{x^j} *(or simply r_j) of a state x^j is the order of its stabilizer subgroup $|G_{x^j}|$.*

In figure 2.4 we see three states and their orbit stabilizer subgroups. The first state, with only a single face, has a symmetry number of 3 since any rotation by a multiple of $\frac{2\pi}{3}$ fixes the face. The second has a symmetry number of 2 since only the identity and a rotation by π will fix the faces. The final state, with three faces, cannot be fixed by any rotation other than the identity and thus has symmetry number 1.

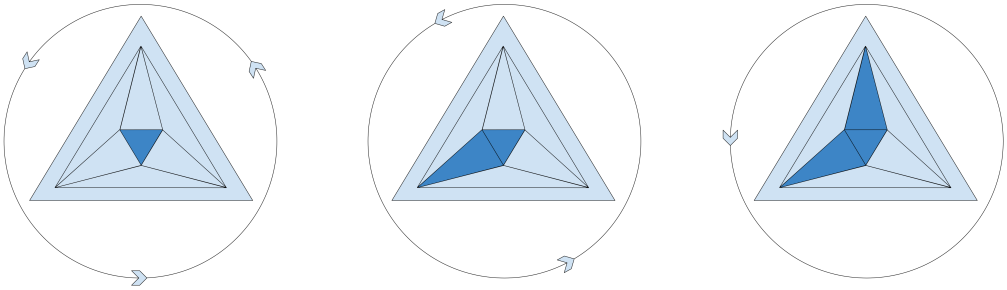

Figure 2.4: The stabilizer subgroups for various octahedron states.

Theorem 3. *If the states x and $\hat{x}$ are members of the same intermediate $[x]$, they have the same symmetry number. Thus, we extend the notion of a symmetry number to be a property of an intermediate.*

Proof. By Corollary 1 and since $G.x \doteq [x] = [\hat{x}] \doteq G.\hat{x}$, the result follows.

$$
\begin{aligned}
r_x &= |G_x| && (2.1)\\
&= \frac{|G|}{|G.x|} && (2.2)\\
&= \frac{|G|}{|G.\hat{x}|} && (2.3)\\
&= |G_{\hat{x}}| && (2.4)\\
&= r_{\hat{x}} && (2.5)
\end{aligned}
$$

□

Since the Building Game is at its core an attachment model, we are interested in which intermediates can be formed from others by attaching a face to a particular intermediate.

Definition 7. *Two distinct intermediates* $[x]$ *and* $[y]$ *are* ***connected*** *if there exist states* $x \in [x]$ *and* $y \in [y]$ *such that one of the following holds:*

- *there is a face,* f*, in the state* y *such that* $y = x \cup \{f\}$
- *there is a face,* f*, in the state* x *such that* $x = y \cup \{f\}$

Lemma 2. *If intermediates* $[x]$ *and* $[y]$ *are connected, then for every state* $x \in [x]$ *there is a state* $y \in [y]$ *such that* $\exists f \in y : y = x \cup \{f\}$ *or* $\exists f \in x : x = y \cup \{f\}$.

Proof. Without loss of generality, assume $|x| < |y|$. Since $[x]$ and $[y]$ are connected, let $\hat{x} \in [x]$, $\hat{y} \in [y]$, and $\hat{f} \in \hat{y}$, be such that $\hat{y} = \hat{x} \cup \{\hat{f}\}$. Then, for any $x \in [x]$, pick $g \in G$ such that $x = g.\hat{x}$. By choosing $y = g.\hat{y}$ and $\{f\} = g.\{\hat{f}\}$, we have

$$\begin{aligned} y &= g.\hat{y} && (2.6)\\ &= g.(\hat{x} \cup \{\hat{f}\}) && (2.7)\\ &= g.\hat{x} \cup g.\{\hat{f}\} && (2.8)\\ &= x \cup \{f\} && (2.9)\\ & && (2.10) \end{aligned}$$

and our result is shown. □

Definition 8. *A Building Game* ***pathway*** *is a sequence of intermediates* $[x^{p_1}]$, $[x^{p_2}]$, $\ldots$, $[x^{p_N}]$ *such that* $[x^{p_i}]$ *is connected to* $[x^{p_{i+1}}]$, $|x^{p_i}| = i$, *and* $x^{p_N} = F$.

Figure 2.5 shows a Building Game pathway for the octahedron using Schlegel diagrams. The pathway has 8 intermediates since there must be exactly one intermediate x^{p_i} satisfying $h\left(x^{p_i}\right) = i$ for each $i = 1, 2, \ldots, 8$.

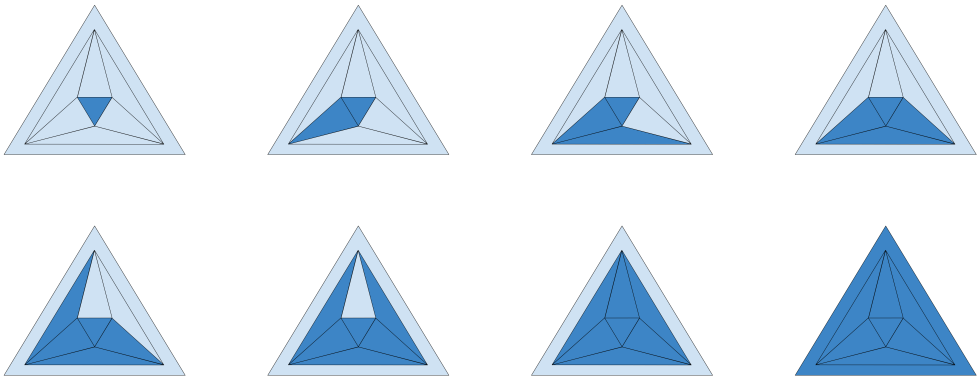

Figure 2.5: One Building Game pathway for the Octahedron.

Definition 9. *The Building Game* ***combinatorial configuration space*** *for a polyhedron is a graph in which the nodes are the polyhedron's intermediates and a graph edge exists between two intermediates if and only if they are connected.*

When the intermediates are partitioned by the number of faces they possess, it is natural to arrange the combinatorial configuration space into columns or tiers according to this partition. Figure 2.6 shows the Building Game state space for the cube. As seen, each column has intermediates with the same number of faces and connections thus exist with intermediates that are either in the tier directly above or below them. We can also see that there are three distinct pathways contained in the state space.

Interestingly, it is not the case that the addition or removal of each face of an intermediate results in a distinct intermediate.

Definition 10. *For two connected intermediates* $[x^j]$ *and* $[x^k]$*, the set of different*

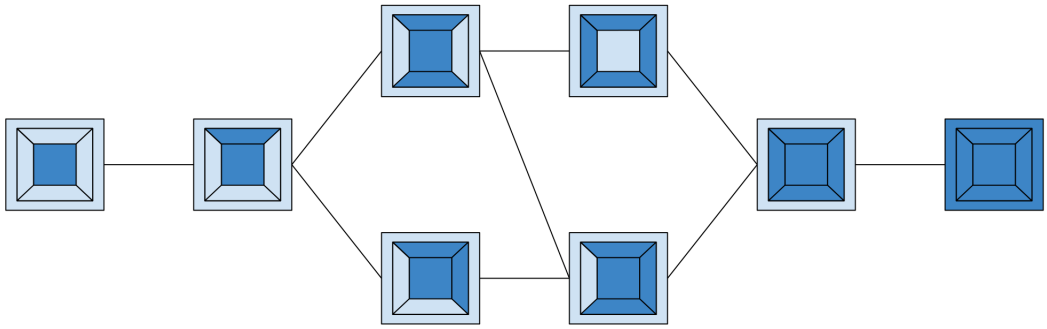

Figure 2.6: The Building Game combinatorial configuration space of the cube.

faces

$$F_{jk} \doteq \left\{ f \notin x^j : x^j \cup \{f\} \in [x^k] \right\} \cup \left\{ f \in x^j : x^j \setminus \{f\} \in [x^k] \right\}$$

that can be added or removed from x^j *to get an element of* $[x^k]$ *is called the* ***degeneracy set*** *and the number of such faces* $S_{jk} \doteq |F_{jk}|$ *is called the* ***degeneracy number****.*

The act of attaching an additional face is referred to as a forward step in the Building Game and the removal of a face is a backward step. As such, degeneracies are sometimes referred to as forward or backward degeneracies. It is important to note that in general the degeneracy number is not symmetric, i.e. $S_{jk} \neq S_{kj}$ for some connections $[x^j] \leftrightarrow [x^k]$ in the state space. Figure 2.7 depicts the forward and backward degeneracy numbers for a particular connection. As illustrated, there are four faces that can be added to the first intermediate to form the second. However, removing any of the two faces of the second intermediate will result in the first. Thus the forward degeneracy number is four and the backward degeneracy number is two.

Finally, we extend the notion of a pathway to to be a sequence of intermediates of arbitrary lengths, such that the sequence begins at an intermediate with a single face and ends with the completed polyhedron and each pair of successive intermediates

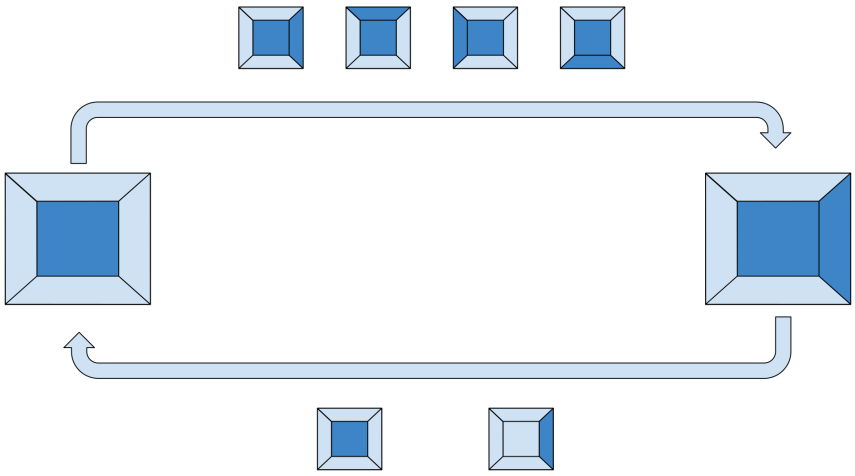

Figure 2.7: Degeneracies between two connected cube intermediates.

are connected.

Definition 11. *A Building Game* ***reversible pathway*** *is a sequence of intermediates* $[x^{p_1}], [x^{p_2}], \ldots, [x^{p_N}]$ *such that* $[x^{p_i}]$ *is connected to* $[x^{p_{i+1}}]$, $|x^{p_0}| = 1$, *and* $|x^{p_N}| = |F|$.

2.2.3 Group Theoretic Results

Lemma 3. *Let* $g \in G$ *and* $x, y \subset F$. *Then* $g.(x \cup y) = g.x \cup g.y$

Proof. Let $f \in g.(x \cup y)$. Then $g^{-1}.\{f\} \in x \cup y$. Without loss of generality, assume $g^{-1}.\{f\} \in x$. It follows that $f \in g.x$ and $f \in g.x \cup g.y$ as well. Therefore, $g.(x \cup y) \subset g.x \cup g.y$.

Conversely, suppose $\hat{f} \in g.x \cup g.y$ and without loss pick $\hat{f}$ to be in $g.x$. It follows that $g^{-1}.\{\hat{f}\} \in x, x \cup y$ and subsequently $\hat{f} \in g.(x \cup y)$. From this we have the reverse relation $g.x \cup g.y \subset g.(x \cup y)$ which proves our equality. □

Theorem 4. *Let* $[x^j]$ *and* $[x^k]$ *be two Building Game intermediates connected in*

the combinatorial configuration space. If r_j and r_k are the intermediates' symmetry numbers and S_{jk} and S_{kj} are the forward and backward degeneracy numbers, then $r_k S_{jk} = r_j S_{kj}$.

Proof. We prove the result by arguing the following three equalities.

$$r_k S_{jk} = |\{(f,g) \in F \times G : x^j \cup \{f\} = g.x^k\}| \quad = \sum_{f \in F} |\{g \in G : x^j \cup \{f\} = g.x^k\}| \tag{2.11}$$

$$= |\{(f,g) \in F \times G : x^k \setminus \{f\} = g.x^j\}| \quad = \sum_{f \in F} |\{g \in G : x^k \setminus \{f\} = g.x^j\}| \tag{2.12}$$

$$= r_j S_{kj} \tag{2.13}$$

First, equality 2.11 is shown as follows.

$$|\{(f,g) \in F \times G : x^j \cup \{f\} = g.x^k\}| = \sum_{f \in F} |\{g \in G : x^j \cup \{f\} = g.x^k\}| \tag{2.14}$$

$$= \sum_{f \in F^{jk}} |\{g \in G : x^j \cup \{f\} = g.x^k\}| \tag{2.15}$$

$$= \sum_{f \in F^{jk}} |\{g \in G : h^{(f)}.x^k = g.x^k\}| \tag{2.16}$$

$$= \sum_{f \in F^{jk}} |\{g \in G : g^{-1}h^{(f)}.x^k = x^k\}| \tag{2.17}$$

$$= \sum_{f \in F^{jk}} |\{\tilde{g} \in G : \tilde{g}.x^k = x^k\}| \tag{2.18}$$

$$= \sum_{f \in F^{jk}} |G_y| \tag{2.19}$$

$$= r_k S_{jk} \tag{2.20}$$

This sum in 2.15 is reduced to be only over $f \in F^{jk}$ since the inner condition means that $x^j \cup \{f\} \in [x^k]$ and thus $f \in F^{jk}$. Since $x^j \cup \{f\} \in [x^k]$, in 2.16 there is a $h^{(f)} \in G$ such that $x^j \cup \{f\} = h^{(f)}.x^k$. Here the (f) superscript just denotes dependence on the particular f in the sum. Equation 2.18 reindexes G using $\tilde{g} = g^{-1}h^{(f)}$ since this still represents all elements in G.

Arguing similarly, we prove equality 2.13.

$$|\{(f,g) \in F \times G : x^k \setminus \{f\} = g.x^j\}| = \sum_{f \in F} |\{g \in G : x^k \setminus \{f\} = g.x^j\}| \tag{2.21}$$

$$= \sum_{f \in F^{kj}} |\{g \in G : x^k \setminus \{f\} = g.x^j\}| \tag{2.22}$$

$$= \sum_{f \in F^{kj}} |\{g \in G : h^{(f)}.x^j = g.x^j\}| \tag{2.23}$$

$$= \sum_{f \in F^{kj}} |\{g \in G : g^{-1}h^{(f)}.x^j = x^j\}| \tag{2.24}$$

$$= \sum_{f \in F^{kj}} |\{\tilde{g} \in G : \tilde{g}.x^j = x^j\}| \tag{2.25}$$

$$= \sum_{f \in F^{kj}} |G_{x^j}| \tag{2.26}$$

$$= r_j S_{kj} \tag{2.27}$$

It only remains to prove equality 2.12.

$$|\{(f,g) \in F \times G : x^j \cup \{f\} = g.x^k\}| = \sum_{g \in G} |\{f \in F : x^j \cup \{f\} = g.x^k\}| \tag{2.28}$$

$$= \sum_{g \in G} |\{f \in F : g^{-1}.x^j \cup g^{-1}.\{f\} = x^k\}| \tag{2.29}$$

$$= \sum_{g \in G} |\{f \in F : g^{-1}.x^j = x^k \setminus g^{-1}.\{f\}\}| \tag{2.30}$$

$$= \sum_{g \in G} |\{\hat{f} \in F : g^{-1}.x^j = x^k \setminus \{\hat{f}\}\}| \tag{2.31}$$

$$= |\{(\hat{f}, g) \in F \times G : x^k \setminus \{\hat{f}\} = g^{-1}.x^j\}| \tag{2.32}$$

$$= \sum_{\hat{f} \in F} |\{g \in G : x^k \setminus \{\hat{f}\} = g^{-1}.x^j\}| \tag{2.33}$$

$$= \sum_{\hat{f} \in F} |\{\hat{g} \in G : x^k \setminus \{\hat{f}\} = \hat{g}.x^j\}| \tag{2.34}$$

$$= |\{(f, g) \in F \times G : x^k \setminus \{f\} = g.x^j\}| \tag{2.35}$$

□

2.3 Stochastic Modeling Results

Since the Building Game is a sequential process with several choices at each step, it is natural to consider it as a stochastic process. If we define a Building Game process that allows faces to be sequentially added or removed in a reversible way, the process consists of transitions from intermediate to intermediate along connections in the combinatorial configuration space. By specifying a distribution on these transitions, it will induce a stationary distribution on the state space and provide a framework for compute relevant statistics such as expected formation times.

We define the Markov process X_t by the transition rate matrix Q from equation 2.36, with the heuristic that the rate of transition to an intermediate $[x^k]$ from

an intermediate $[x^j]$ should be proportional to the number of faces that can be added or removed from $[x^j]$ to reach $[x^k]$. For this reason, we include the degeneracy number S_{jk} as a factor in the transition rate matrix. Furthermore, we model the process using an energetic interpretation in which each intermediate has an energy and to transition between intermediates, an energy barrier $E_{jk} = E_{kj}$ must be overcome.

$$Q_{jk} = \begin{cases} S_{jk} e^{-\beta\left(E_{jk} - E_j\right)} & \text{if } [x^j] \leftrightarrow [x^k] \\ -z_j & \text{if } j = k \\ 0 & \text{else} \end{cases} \tag{2.36}$$

Here, $z_j \doteq \sum_{\ell:\ell\neq j} S_{j\ell} e^{-\beta\left(E_{j\ell} - E_j\right)}$ is the rate at which the process leaves x^j.

Both [13] and [6] adopt a similar approach, but handle degeneracies differently and use a different energy function.

2.3.1 Stationary Distribution

Theorem 5. *Let D be a diagonal matrix with positive entries and C be a symmetric matrix with $C_{jk} > 0$ for connected intermediates and $C_{jk} = 0$ for unconnected intermediates. If a transition rate matrix can be factorized as $Q = DC$, then X_t has the unique stationary distribution $\pi = \operatorname{diag}(D^{-1})$.*

Proof. First, we show Q and π satisfy detailed balance.

$$\pi_j Q_{jk} = \left(\frac{1}{D_{jj}}\right)(D_{jj} C_{jk}) \tag{2.37}$$

$$= C_{jk} \tag{2.38}$$

$$= C_{kj} \tag{2.39}$$

$$= \left(\frac{1}{D_{kk}}\right)(D_{kk}C_{kj}) \tag{2.40}$$

$$= \pi_k Q_{kj} \tag{2.41}$$

Now, since the combinatorial state space is connected and $Q_{jk}, Q_{kj} > 0$ for all connected intermediates $[x^j]$ and $[x^k]$, the process is clearly aperiodic and positive recurrent. Thus, π is the unique stationary distribution of the Markov process [21]. □

Theorem 6. *The Markov process X_t defined by the transition rate matrix Q in equation 2.36 admits the unique stationary distribution $\frac{1}{Zr_j}e^{-\beta E_j}$ where $Z \doteq \sum_\ell \frac{1}{r_\ell}e^{-\beta E_\ell}$ is the partition function.*

Proof. We take $C_{jk} \doteq \frac{S_{jk}}{Zr_j}e^{-\beta E_{jk}}$ and notice that it is symmetric by theorem 4. With $D_{jj} \doteq Zr_j e^{\beta E_j}$ we have our factorization

$$Q_{jk} = S_{jk}e^{-\beta\left(E_{jk}-E_j\right)} \tag{2.42}$$

$$= \left(Zr_j e^{\beta E_j}\right)\left(\frac{S_{jk}}{Zr_j}e^{-\beta E_{jk}}\right) \tag{2.43}$$

$$= D_{jj}C_{jk}. \tag{2.44}$$

Thus, by theorem 5, $\pi_j = \frac{1}{D_{jj}} = \frac{1}{Zr_j}e^{-\beta E_j}$. □

2.3.2 Hitting Times

While the stationary distribution is an important piece in understanding the nature of our Markov process, other statistics which describe the dynamics are also useful.

For instance, we may be interested in the expected time it will take the process to travel from an intermediate $[x^j]$ to a specific subset A of the combinatorial configuration space. Equation 2.45 provides a mathematical definition for such a stopping time.

$$\tau_j^A \doteq \inf\left\{t \geq 0 : X_t \in A, X_0 = x^j\right\} \tag{2.45}$$

Often, we choose A to be the intermediate of the fully completed polyhedron. This means the the stopping time will represent the expected formation time from a given intermediate.

To assist in our calculations of hitting times, it is helpful to use X_t's discrete time jump process Y_n. The jump process is simply a record of the sequence of intermediates that X_t travels through. The discrete time Markov transition matrix associated with Y_n is defined using Q.

$$P_{jk} = \begin{cases} -Q_{jk}/Q_{jj} & \text{if } j \neq k \\ 0 & \text{if } j = k \end{cases} \tag{2.46}$$

With this definition, the continuous time process X_t has the alternative interpretation where the intermediate X_t changes according to the jump process Y_n and stays at each intermediate $[x^j]$ for an exponentially distributed amount of time with parameter $-Q_{jj}$ [21].

The expected hitting time $E\left[\tau_j^A\right]$ can be computed as the solution to se set of

linear equations: one for each intermediate [21]. In the case of $j \notin A$:

$$1 = -\sum_k Q_{jk} E\left[\tau_k^A\right], \tag{2.47}$$

and clearly for $j \in A$ we trivially have a stopping time of zero

$$E\left[\tau_j^A\right] = 0. \tag{2.48}$$

Putting together the two cases, we write the solution as a linear system with τ^A the vector of stopping times with each possible initial intermediate

$$\left(\operatorname{diag}\left(\mathbb{1}_A\right) - \operatorname{diag}\left(\mathbb{1}_{A^c}\right) Q\right) E\left[\tau^A\right] = \mathbb{1}_{A^c}. \tag{2.49}$$

Thus, we have

$$E\left[\tau^A\right] = \left[\left(\operatorname{diag}\left(\mathbb{1}_A\right) - \operatorname{diag}\left(\mathbb{1}_{A^c}\right) Q\right)\right]^{-1} \mathbb{1}_{A^c} \tag{2.50}$$

which contains the particular value $E[\tau_j^A]$ that we are interested in.

Further, we can also compute the exact distribution of the stopping times τ^A in a similar manner [21]. First, we define the CDF as

$$\psi_j^A(t) \doteq P\left(\tau_j^A \leq t\right) \tag{2.51}$$

$$\tag{2.52}$$

which gives rise to the system of differential equations and boundary conditions

$$\frac{d\psi^A}{dt} = \operatorname{diag}\left(\mathbb{1}_{A^c}\right) Q \psi^A \tag{2.53}$$

$$\psi^A(0) = \mathbb{1}_A \tag{2.54}$$

$$\psi_j^A(t) = 0 \ \forall j \in A. \tag{2.55}$$

This gives rise to the solution

$$\psi^A(t) = e^{\mathrm{diag}(\mathbb{1}_{A^c})Qt}\mathbb{1}_A \tag{2.56}$$

for the CDF of the stopping time τ^A, but we can also compute the PDF explicitly for $t > 0$.

$$p\left(\tau^A = t\right) = \frac{d\psi^A}{dt} \tag{2.57}$$

$$= \mathrm{diag}\left(\mathbb{1}_{A^c}\right) Q\psi^A \tag{2.58}$$

Another hitting time statistic we may be interested in is, given an initial intermediate, what is the probability we will hit a subset of intermediates A before some other disjoint subset of intermediates B.

$$\rho_j^{A,B} \doteq P(\tau_j^A < \tau^B) \tag{2.59}$$

Trivially, we have

$$\rho_j^{A,B} = 1 \text{ if } j \in A \tag{2.60}$$

$$\rho_j^{A,B} = 0 \text{ if } j \in B \tag{2.61}$$

but must compute the case of $j \in (A \cup B)^C$.

$$\rho_j^{A,B} = P(\tau_j^A < \tau_j^B) \tag{2.62}$$

$$= \sum_k P(\tau_k^A < \tau_k^B) P(Y_1 = k) \tag{2.63}$$

$$= \frac{1}{Z_j} \sum_{k \neq j} \rho_k^{A,B} Q_{jk} \tag{2.64}$$

$$0 = Q\rho^{A,B} \tag{2.65}$$

Again, putting each of these cases together, we get the linear system

$$\left(\text{diag}\left(\mathbb{1}_A\right) + \text{diag}\left(\mathbb{1}_B\right) + \text{diag}\left(\mathbb{1}_{(A\cup B)^c}\right) Q\right) \rho^{A,B} = \mathbb{1}_A \tag{2.66}$$

and our solution is

$$\rho^{A,B} = \left[\text{diag}\left(\mathbb{1}_A\right) + \text{diag}\left(\mathbb{1}_B\right) + \text{diag}\left(\mathbb{1}_{(A\cup B)^c}\right) Q\right]^{-1} \mathbb{1}_A. \tag{2.67}$$

Using this result, an insightful choice for A and B would be $A = \{x_k\}$ and $B = x_{|F|}$. Then, the value of $\rho_1^{A,B}$ would correspond to the probability that a particular intermediate x^k is in a reversible pathway between the intermediate with a single face and the completed polyhedron.

Chapter Three

The Building Game: Enumeration

3.1 Known Enumerative Results

When treating the Building Game for a polyhedron as a stochastic process, we must first compute the combinatorial configuration space exactly. This is a computational enumeration problem, but the results of this enumeration are also of mathematical interest outside its stochastic use.

Attachment models like the Building Game have been the topic of much research in combinatorial mathematics. A well known example of this is the study of polyominoes. An n-omino is a configuration of n attached squares in a 2-dimensional lattice. As seen in figure 3.1, the classic video game Tetris uses each of the seven rotationally unique tetrominoes ($n = 4$) as game pieces. The enumeration of n-ominoes has been extensively studied and there are many enumeration results, both explicit and asymptotic. As in our case, the general goal is to enumerate the polyominoes that are unique when acted on by some group (rotation, reflection, etc) or under certain constraints which are often topological.

The Building Game contains a wealth of combinatorial enumeration problems. Some have been addressed in the work of Endres et al. [6] in which they computed the number of intermediates for each of the Platonic solids. Additionally, the entire combinatorial configuration space for the dodecahedron is illustrated. Before [6], David Wilson enumerated the number of intermediates composed of each number of faces for the icosahedron [10]. This computation seems to have occurred independent of the scientific context of self-assembly as Wilson's enumerations are referred to as the "Number of one-sided triangular n-ominoes (or triominoes) on the icosahedron." He also includes variants such as the "Number of triangular n-ominoes on the icosahedron" which enlarges the equivalence classes to identify intermediate that

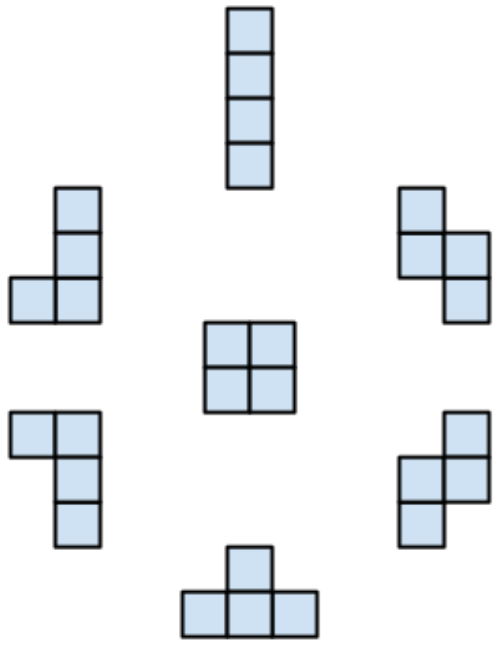

Figure 3.1: The seven tetrominoes used in Tetris.

are reflections of each other.

3.2 New Enumerative Results

Previous enumeration results for the Building Game have focused on the number of intermediates of the Platonic solids. We extend these results to also count the number of Building Game connections and pathways in the combinatorial configuration space. Additionally, we consider the Platonic, Archimedean, and Catalan solids classes with these results presented in table 3.1.

As we consider polyhedra with more and more faces, there is a combinatorial explosion in the number of intermediates in combinatorial configuration space. This was noted in [6], where the dodecahedron has 73 intermediates and the icosahedron has $2,649$ [6]. We have enumerated the intermediates for polyhedra of up to 30 faces

in our three classes of polyhedra. The 30-faced rhombic triacontahdron has the most intermediates of all polyhedra in our computation with $2,423,212$.

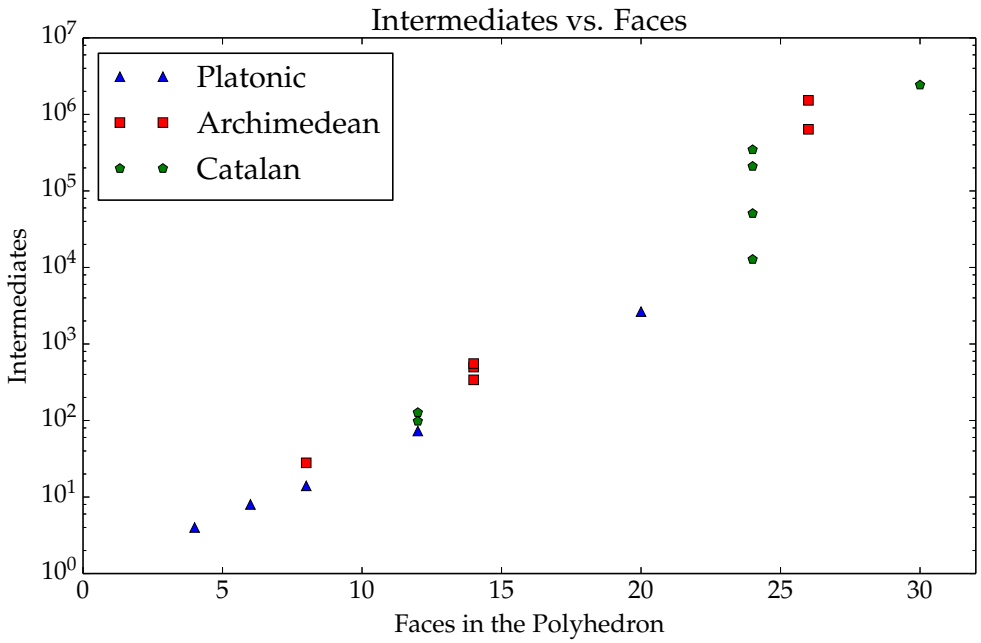

Figure 3.2: The relation between number of faces and intermediates in the Building Game combinatorial configuration space.

3.2.1 Shellability

The theory of polytopes provides a mathematical method for constructing a polytope called a *shelling*. The process, similar to the Building Game, involves sequentially adding facets of the polytope until all of the facets are added. However, the rules as to which facets may be added at each step of the process are more restrictive. Ziegler provides the following definitions of a polytopal complex, a shelling, and a shellable polytopal complex [32]. Recall that a polytopal complex is

Definition 12. *A (geometric)* ***polytopal complex*** $\mathcal{C}$ *in* $\mathbb{R}^d$ *is a collection of polytopes in* $\mathbb{R}^d$ *such that*

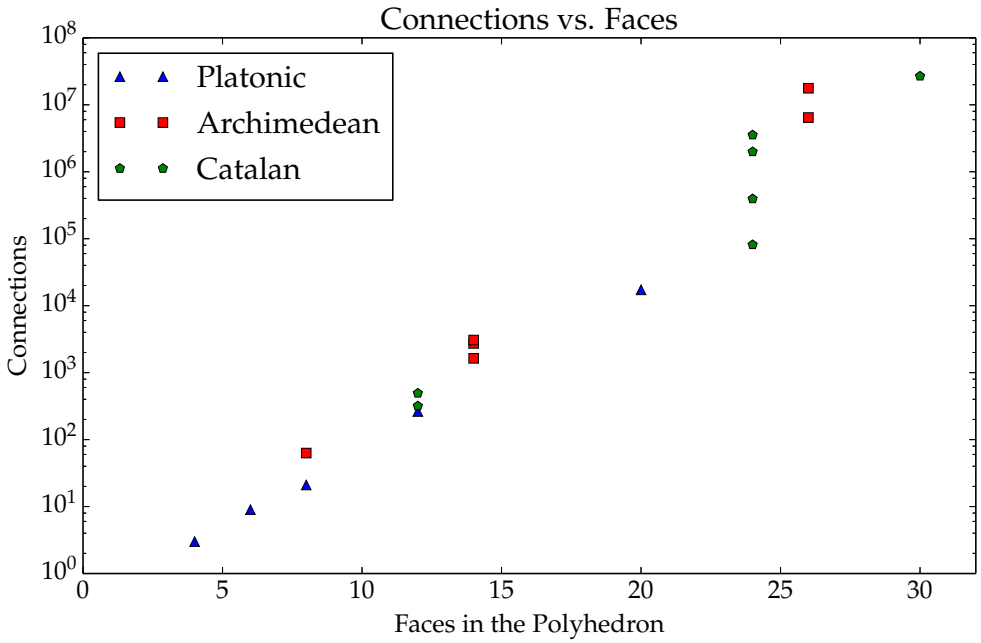

Figure 3.3: The relation between number of faces and connections in the Building Game combinatorial configuration space.

(i) the empty set is in $\mathcal{C}$*,*

(ii) for any $P \in \mathcal{C}$ *all faces of* P *are in* $\mathcal{C}$*,*

(iii) the intersection of any two polytopes in $\mathcal{C}$ *is a face of both..*

If all facets are of the same dimension then $\mathcal{C}$ *is* ***pure****.*

Definition 13. *Let* $\mathcal{C}$ *be a pure* k*-dimensional polytopal complex* $\mathcal{C}$*. A* ***shelling*** *of* $\mathcal{C}$ *is a linear ordering* $F_1, F_2, \ldots, F_s$ *of the facets of* $\mathcal{C}$ *such that either* $\mathcal{C}$ *is* 0*-dimensional (and thus the facets are points), or it satisfies the following conditions:*

(i) The boundary complex of the first facet F_1 *has a shelling.*

(ii) For $1 < j \leq s$ *the intersection of the facet* F_j *with the previous facets is non-empty and is a beginning segment of a shelling of the* $(k-1)$*-dimensional*

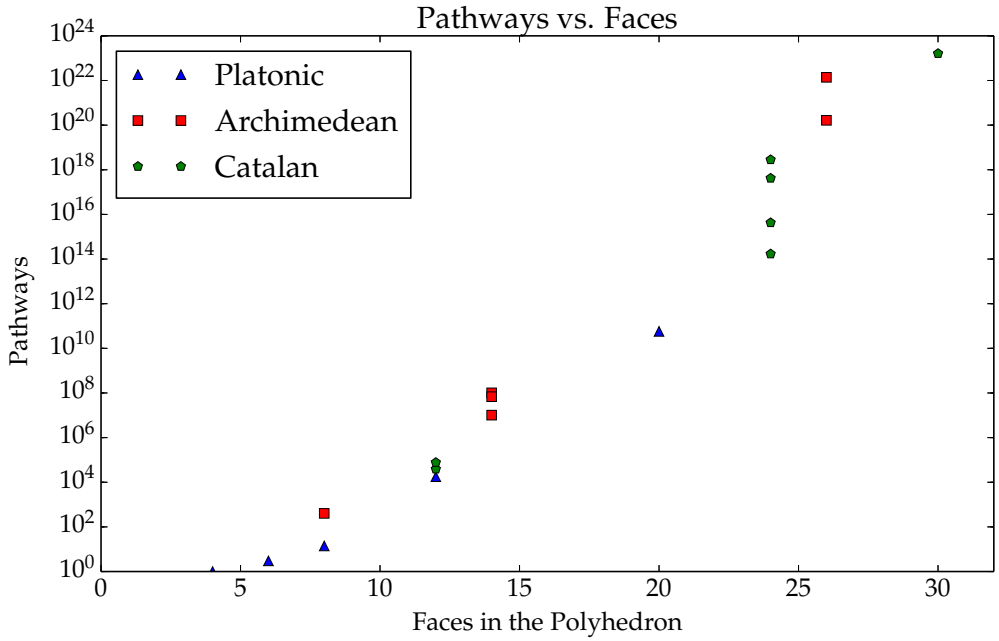

Figure 3.4: The relation between number of faces and paths in the Building Game combinatorial configuration space.

boundary complex of F_j, that is,

$$F_j \cap \left(\bigcup_{i=1}^{j-1} F_i \right) = G_1 \cup G_2 \cup \cdots \cup G_r$$

for some shelling $G_1, G_2, \ldots, G_r, \ldots, G_t$ of $\mathcal{C}(\partial F_j)$, and $1 \leq r \leq t$. (In particular, this requires that $F_j \cap (\bigcup_{i=1}^{j-1} F_i)$ has a shelling, so it has to be a pure $(k-1)$-dimensional, and connected for $k > 1$.)

Definition 14. *A polytopal complex is **shellable** if it is pure and has a shelling.*

Since we are only concerned with 3-dimensional polyhedra, the corresponding polytopal complex simply consists of the faces, edges, and vertices of the polyhedron. The facets of the complex are just the faces.

Definition 15. *A Building Game intermediate x is called a **shellable intermediate** if for every state $x \in [x]$ there is a linear ordering $f_1, f_2, \ldots, f_{|x|}$ on the faces*

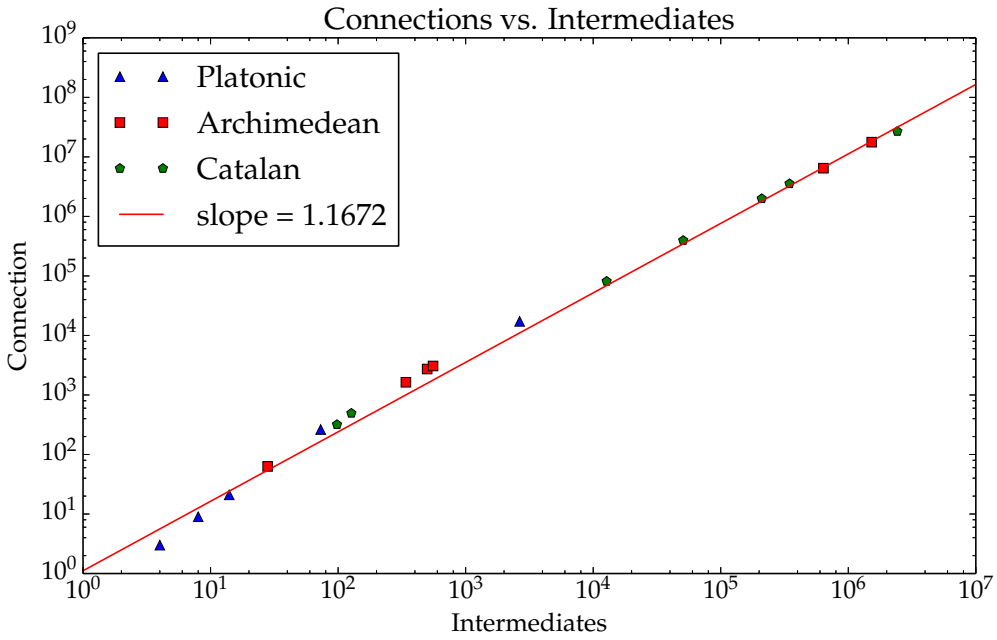

Figure 3.5: The relation between number of intermediates and the number of connections in polyhedra.

of x such that this ordering is a beginning of a shelling $f_1, f_2, \ldots, f_{|x|}, \ldots f_{|F|}$ of the polytopal complex $F \cup E \cup V$.

Lemma 4. *The polytopal complex consisting of the edges and vertices of a polygonal face is shellable.*

Proof. We see that condition (i) from definition 13 is satisfied since the boundary complex of each edge is 0-dimensional.

Condition (ii) is also satisfied since each newly attached edge and the existing partial shelling is a subset of the attached edge's vertices. Since the vertices of the attached edge are 0-dimensional, any linear order of its vertices is a shelling. □

Definition 16. *A Building Game connection is said to be a **shellable connection** if the two connected intermediates are both shellable. A **shellable pathway** is a Building Game pathway composed entirely of shellable intermediates and connections.*

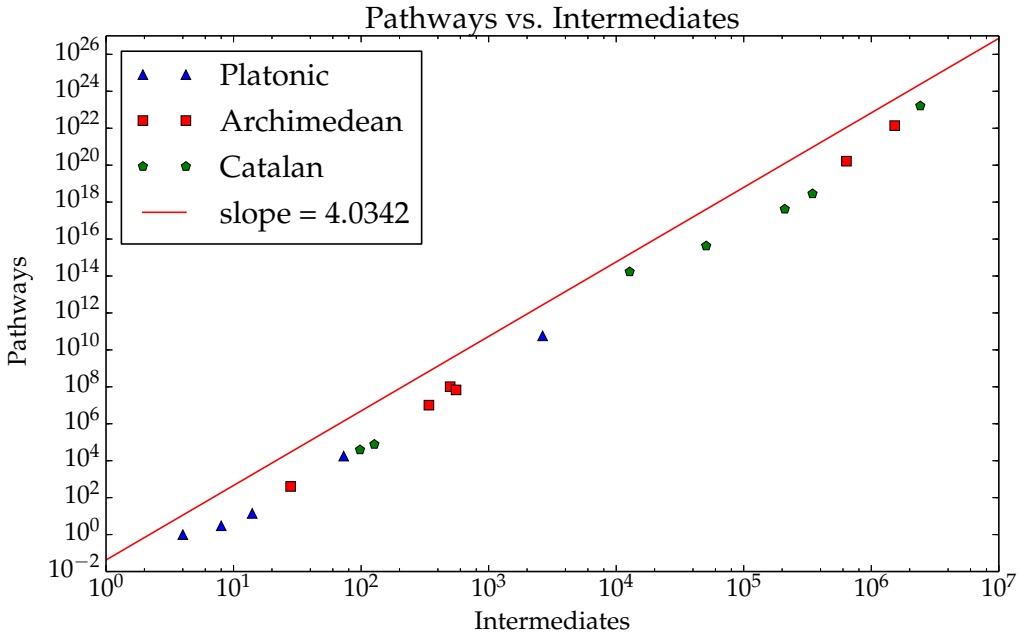

Figure 3.6: The relation between number of intermediates and the number of pathways in polyhedra.

Table 3.2 details the shellability statistics for the Platonic, Archimedean, and Catalan solids classes. Because of the added shellability restriction, these count statistics are lower than in the general case, but the combinatorial growth as the number of faces increases is similar.

3.2.2 Shelling Enumeration

To our knowledge, the enumeration of the number of shellings of the polyhedra in the Platonic, Archimedean, and Catalan solid classes remains an open problem. Here we present these enumerations for the polyhedra of up to 30 faces. Since the concept of a shelling is similar to to that of a Building Game pathway, we use the structure of each computed combinatorial configuration space to derive an efficient method for counting the number of shellings of a polyhedron.

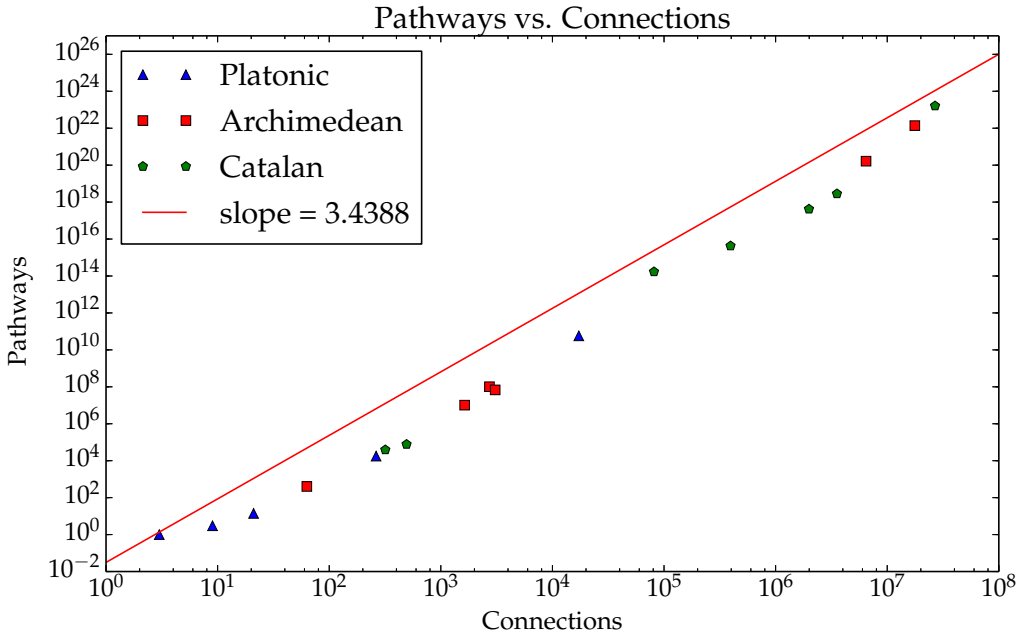

Figure 3.7: The relation between number of connections and the number of pathways in polyhedra.

Theorem 7. *The total number of shellings for a polyhedron is*

$$\#(shellings) = \sum_{\substack{shellable\\ pathways:\\ p_1,p_2,\ldots,p_{|F|}}} |[x^{p_1}]| \prod_{j}^{|F|-1} S_{p_j p_{(j+1)}}. \tag{3.1}$$

Proof. We use the function that returns whether adding f_k to the existing sub-shelling $f_1, \ldots, f_{k-1}$ is itself a sub-shelling and call it $sh(\{f_1, \ldots, f_{k-1}\}, f_k)$. Further, a Building Game shellable connection between two intermediates is denoted $[x^{p_i}] \xrightarrow{sh} [x^{p_{i+1}}]$. Using this notation, we compute the result directly:

$$\begin{aligned}
\#(shellings) &= \sum_{f_1 \in F} \mathbb{1}_{sh(\emptyset, f_1)} \cdots \sum_{f_k \in F \setminus \{f_1,\ldots,f_{k-1}\}} \mathbb{1}_{sh(\{f_1,\ldots,f_{k-1}\}, f_k)} \\
&\qquad \cdots \sum_{f_{|F|} \in F \setminus \{f_1,f_2,\ldots,f_{|F|-1}\}} \mathbb{1}_{sh(\{f_1,f_2,\ldots,f_{|F|-1}\}, f_{|F|})} \\
&= \sum_{f_1 \in F} \mathbb{1}_{sh(\emptyset, f_1)} \cdots \sum_{f_k \in F \setminus \{f_1,\ldots,f_{k-1}\}} \mathbb{1}_{sh(\{f_1,\ldots,f_{k-1}\}, f_k)}
\end{aligned}$$

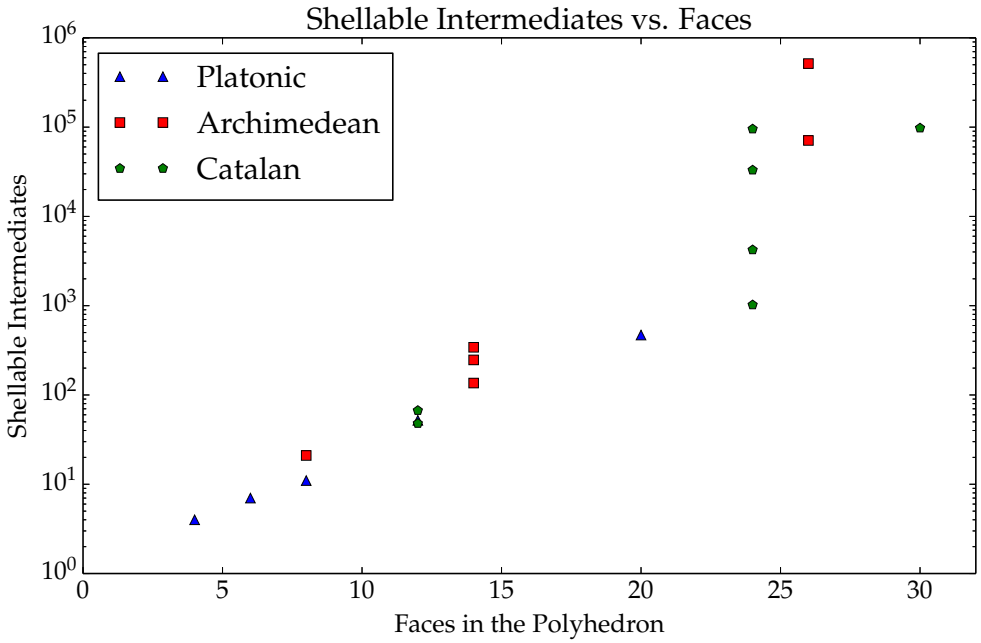

Figure 3.8: The relation between number of faces and shellable intermediates in the Building Game combinatorial configuration space.

$$
\begin{aligned}
&\cdots \sum_{x^{p_{|F|}}:[x^{p_{|F|-1}}]\xrightarrow{sh}[x^{p_{|F|}}]} S_{p_{|F|-1}p_{|F|}} \\
&= \sum_{f_1\in F} \sum_{x^{p_2}:[x^{p_1}]\xrightarrow{sh}[x^{p_2}]} S_{p_1p_2} \cdots \sum_{x^{p_{|F|}}:[x^{p_{|F|-1}}]\xrightarrow{sh}[x^{p_{|F|}}]} S_{p_{|F|-1}p_{|F|}} \\
&= \sum_{[x^{p_1}]:|x^{p_1}|=1} |[x^{p_1}]| \sum_{x^{p_2}:[x^{p_1}]\xrightarrow{sh}[x^{p_2}]} S_{p_1p_2} \cdots \sum_{x^{p_{|F|}}:[x^{p_{|F|-1}}]\xrightarrow{sh}[x^{p_{|F|}}]} S_{p_{|F|-1}p_{|F|}} \\
&= \sum_{\substack{\text{shellable}\\ \text{pathways:}\\ p_1,p_2,\ldots,p_{|F|}}} |[x^{p_1}]| \prod_{j}^{|F|-1} S_{p_jp_{(j+1)}}.
\end{aligned}
$$

□

Using dynamic programming we can compute the number of shellings explicitly without having to explicitly consider each pathway individually. Define the quantity

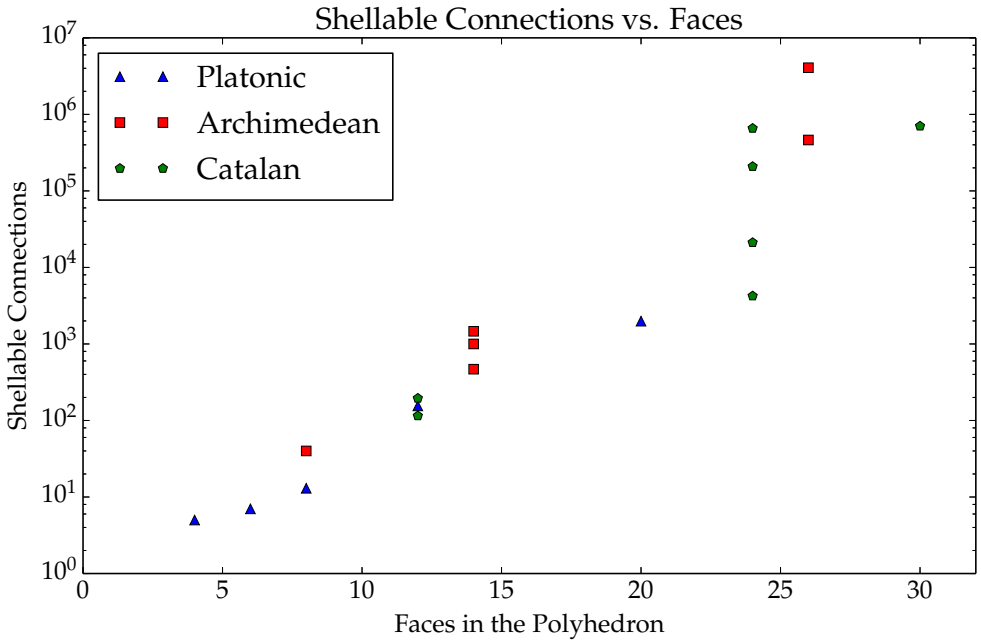

Figure 3.9: The relation between number of faces and shellable connections in the Building Game combinatorial configuration space.

a_{ik} as

$$a_{i,k} \doteq \sum_{\substack{\text{shellable subpaths:} \\ p_1,\ldots,p_k \\ x^{p_k} \in [x^i]}} |G.x^{p_1}| \prod_{j=1}^{k-1} S_{p_j p_{j+1}} \tag{3.2}$$

where a subpath is the first k intermediates in some valid pathway. Using this definition, we first note that the number of shellings, which is the quantity of interest, is equal to $a_{N,|F|}$ where $x^N = F$. Now, we set up the following recursion that will be the basis for our computation.

$$a_{i,k} \doteq \sum_{\substack{\text{shellable subpaths:} \\ p_1,\ldots,p_k \\ x^{p_k} \in [x^i]}} |G.x^{p_1}| \prod_{j=1}^{k-1} S_{p_j p_{j+1}} \tag{3.3}$$

$$= \sum_{[x^\ell]:[x^\ell] \xrightarrow{sh} [x^i]} \sum_{\substack{\text{shellable subpaths:} \\ p_1,\ldots,p_{k-1} \\ x^{p_{k-1}} \in [x^\ell]}} |G.x^{p_1}| S_{\ell i} \prod_{j=1}^{k-2} S_{p_j p_{j+1}} \tag{3.4}$$

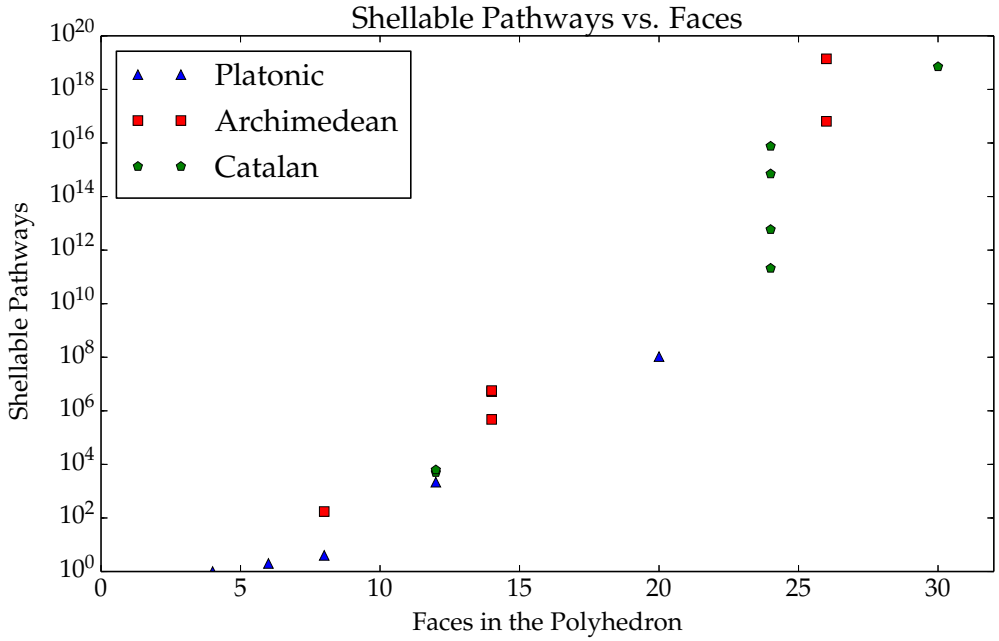

Figure 3.10: The relation between number of faces and shellable paths in the Building Game combinatorial configuration space.

$$= \sum_{[x^\ell]:[x^\ell]\xrightarrow{sh}[x^i]} S_{\ell i} \sum_{\substack{\text{shellable subpaths:} \\ p_1,\ldots,p_{k-1} \\ x^{p_{k-1}}\in[x^\ell]}} |G.x^{p_1}| \prod_{j=1}^{k-2} S_{p_j p_{j+1}} \tag{3.5}$$

$$= \sum_{[x^\ell]:[x^\ell]\xrightarrow{sh}[x^i]} S_{\ell i} a_{\ell,k-1} \tag{3.6}$$

Thus, using the base cases $a_{j,1} = |G.x^j|$ for each single faced intermediate $[x^j]$, we can use this relation to recursively solve for the number of shellings $a_{N,|F|}$.

3.2.3 Bounds and Asymptotics

There is a clear relation between the number of faces in a polyhedron and then number of intermediates it has. However, that relationship also greatly depends on the polyhedral symmetry group. For instance, if the polyhedron has a small

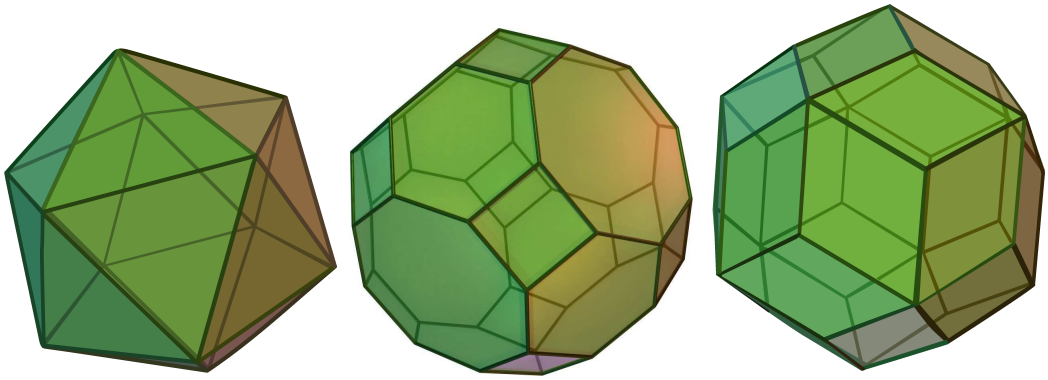

Figure 3.11: These are the largest polyhedron from each of the polyhedral classes that we compute the Building Game for. The icosahedron of the Platonic solids has 20 faces. The Truncated Cuboctahedron of the Archimedean solids has 26 faces. The Rhombic Triacontahedron of the Catalan solids has 30 faces.

number of faces and a trivial rotation group consisting only of the identity, every edge-connected subset of the polyhedron's faces will be a distinct intermediate. In aggregate, this may mean that the polyhedron has more intermediates than another polyhedron with more faces, yet a larger symmetry group.

An upper bound on the number of intermediates is possible using the theory of group actions. Consider the set of all subsets 2^F of a polyhedron with rotation group G. Trivially $|2^F/G|$ is an upper bound on the number of intermediates since it simply relaxes the connectivity requirement for a subset to be a Building Game state. Using Burnside's lemma, we see that

$$|2^F/G| = \frac{1}{|G|}\sum_{g\in G}|(2^F)^g| \tag{3.7}$$

$$> \frac{|(2^F)^e|}{|G|} \tag{3.8}$$

$$= \frac{|2^F|}{|G|} \tag{3.9}$$

$$= \frac{2^{|F|}}{|G|} \tag{3.10}$$

which is not a particularly good bound in practice. The precise value of $|2^F/G|$ is

Polyhedra Name	$\|F\|$	Intermediates	Connections	Pathways
Tetrahedron	4	4	3	1
Cube	6	8	9	3
Octahedron	8	14	21	14
Dodecahedron	12	73	263	17,696
Icosahedron	20	2,649	17,241	57,396,146,640
Truncated Tetrahedron	8	28	63	402
Cuboctahedron	14	340	1,634	10,170,968
Truncated Cube	14	499	2,729	101,443,338
Truncated Octahedron	14	555	3,069	68,106,377
Rhombicuboctahedron	26	638,850	6,459,801	164,068,345,221,515,292,308
Truncated Cuboctahedron	26	1,525,658	17,672,374	13,837,219,462,483,379,105,902
Triakis Tetrahedron	12	98	318	38,938
Rhombic Dodecahedron	12	127	493	76,936
Triakis Octahedron	24	12,748	81,296	169,402,670,046,670
Tetrakis Hexahedron	24	50,767	394,377	4,253,948,297,210,346
Deltoidal Icositetrahedron	24	209,675	1,989,548	418,663,242,727,526,726
Pentagonal Icositetrahedron	24	345,938	3,544,987	2,828,128,000,716,774,492
Rhombic Triacontahedron	30	2,423,212	26,823,095	161,598,744,916,797,017,978,128

Table 3.1: Building Game combinatorial configuration space enumerative results for the Platonic, Archimedean, and Catalan solids.

calculable with minimal computer assistance, but details of this computation are omitted due to its relative uselessness. For the cube, the bound is fairly tight, only including the two non-intermediates corresponding to the empty subset of faces, and the non-connected subset consisting of the top and bottom faces. Thus the cube has the bound $|2^F/G| = 10 \geq 8$. In the case of the tetrahedron, the only over-counted subset of faces is the empty one and the bound is $|2^F/G| = 5 \geq 4$. However, in the case of the icosahedron we have $|2^F/G| \geq \frac{2^{20}}{60} \approx 17476.3 \gg 2649$. Here we use the approximate bound $\frac{2^{|F|}}{|G|}$ which is the largely dominant term in the sum from equation 3.7.

We can get a similar bound on the number of intermediates with a particular number of faces,

$$|\{x \in 2^F : |x| = k\}/G| = \frac{1}{|G|} \sum_{g \in G} |\{x \in 2^F : |x| = k\}^g| \tag{3.11}$$

$$> \frac{|\{x \in 2^F : |x| = k\}^e|}{|G|} \tag{3.12}$$

Polyhedra Name	$\|F\|$	Shellable Intermediates	Shellable Connections	Shellable Pathways
Tetrahedron	4	4	5	1
Cube	6	7	7	2
Octahedron	8	11	13	4
Dodecahedron	12	52	155	2,166
Icosahedron	20	469	1,985	105,999,738
Truncated Tetrahedron	8	21	40	174
Cuboctahedron	14	136	468	477,776
Truncated Cube	14	247	1,000	5,232,294
Truncated Octahedron	14	342	1,464	5,704,138
Rhombicuboctahedron	26	70,887	462,721	64,308,526,503,247,584
Truncated Cuboctahedron	26	515,335	4,070,813	13,890,723,216,176,694,816
Triakis Tetrahedron	12	48	115	5,012
Rhombic Dodecahedron	12	67	195	6,258
Triakis Octahedron	24	1,021	4,237	210,459,770,300
Tetrakis Hexahedron	24	4,224	21,125	5,894,431,702,846
Deltoidal Icositetrahedron	24	33,046	208,317	703,619,122,996,096
Pentagonal Icositetrahedron	24	95,326	657,013	7,572,459,719,248,765
Rhombic Triacontahedron	30	97,741	702,219	7,057,239,571,753,327,764

Table 3.2: Building Game enumerative shellability results for the Platonic, Archimedean, and Catalan solids.

$$= \frac{|\{x \in 2^F : |x| = k\}|}{|G|} \tag{3.13}$$

$$= \frac{\binom{|F|}{k}}{|G|} \tag{3.14}$$

but again, this is not particularly useful, especially for intermediates with around $\frac{1}{2}|F|$ faces.

Since the Building Game is similar in spirit to polyomino enumeration, one might try to assimilate some the techniques used for polyominoes. For example, through fairly simple arguments, one can show that $s_m s_n \leq s_{m+n}$ where s_m is the number of unique polyominoes with m subunits [14]. This leads to the bound $s_m \leq (const)^m$. Unfortunately, this approach is cannot be applied to the Building Game as there is a fundamental difference between the two growth models. In the polyomino case, there is no limit to the number of subunits that can be considered. Importantly, this is not the case for the Building Game since an intermediate can only have $|F|$ faces

Polyhedra Name	$\|F\|$	Shellings
Tetrahedron	4	24
Cube	6	480
Octahedron	8	4,224
Dodecahedron	12	19,041,600
Icosahedron	20	1,417,229,099,520
Truncated Tetrahedron	8	9,216
Cuboctahedron	14	113,055,744
Truncated Cube	14	654,801,408
Truncated Octahedron	14	937,087,104
Rhombicuboctahedron	26	4,728,400,467,971,102,208
Truncated Cuboctahedron	26	688,499,026,944,479,645,952
Triakis Tetrahedron	12	587,040
Rhombic Dodecahedron	12	5,836,800
Triakis Octahedron	24	66,063,419,534,592
Tetrakis Hexahedron	24	1,389,323,257,015,296
Deltoidal Icositetrahedron	24	125,987,819,253,281,472
Pentagonal Icositetrahedron	24	1,144,572,832,023,047,616
Rhombic Triacontahedron	30	15,574,782,555,813,226,074,240

Table 3.3: Number of Shellings for the Platonic, Archimedean, and Catalan solids of up to 30 faces.

at most. Thus any such recurrence relation for the Building Game will result in a good upper bound for the intermediates with a small number of faces at best.

The formulation of meaningful bounds for the number of Building Game intermediates with k faces remains an open problem, especially for $k \sim \frac{1}{2}|F|$. At the root of the problem is the difficulty in mathematically describing the subsets of F are edge connected. Future approaches may incorporate enumeration results for connected sub-graphs or Hamiltonian paths since these topics explicitly acknowledge connectedness properties.

From looking at the statistics on number of faces $|F|$ of a polyhedron and the number of intermediates in its combinatorial configuration space, it is natural to want to make statements about the asymptotic growth of the combinatorial configuration space's size. Unfortunately, when formed in this way, the problem is ill-posed. To

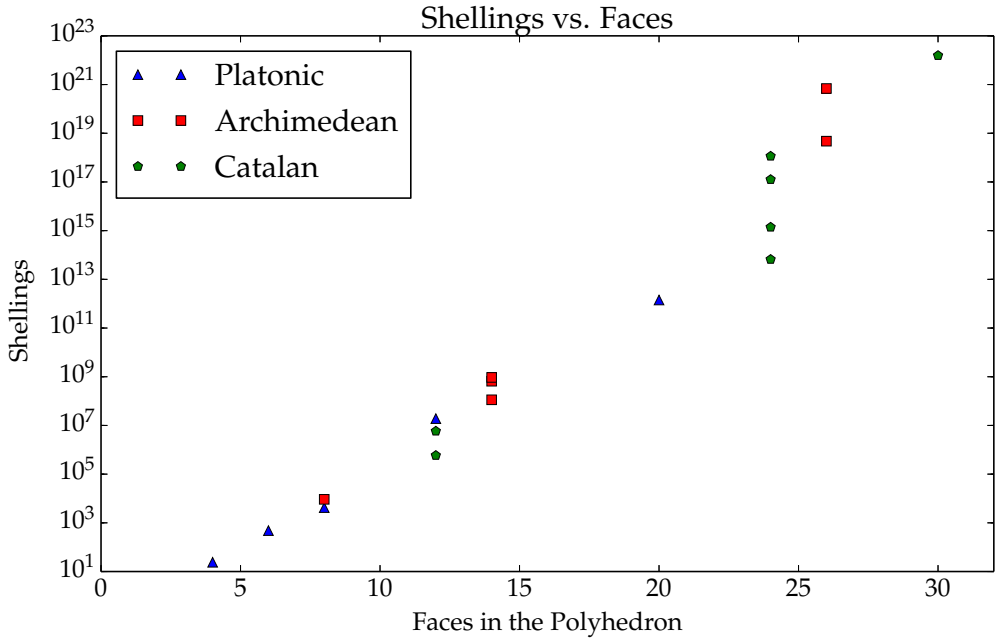

Figure 3.12: The relation between number of faces and the number of shellings a polyhedron has.

discuss asymptotics, we must first specify an infinite class of polyhedra. The Platonic, Archimedean, and Catalan Solid classes that we've worked with thus far are all finite though, so other choices must be considered. One option is to take an existing polyhedron in one of these classes and create an infinite family by describing finer and finer tilings on top of the polyhedron's faces. If designed carefully each member of the tiled polyhedron family will have the same symmetry group, even as the number of faces grows.

3.3 Computational Methods

We use a brute force method to compute the combinatorial configuration space and enumerate the intermediates for a particular polyhedron. Computation begins by first enumerating the intermediates with a single face. The results of this enumeration

are then used to compute the intermediates with two faces. This process proceeds iteratively until all intermediates are accounted for. Figure 3.13 outlines the detailed algorithm for this computation.

At each stage of our algorithm, we know the set of intermediates that have k faces, which we call A_k. This information is then used to compute the set of faces with $k+1$ faces, A_{k+1}. Since all intermediates in A_{k+1} must be formed by adding a single face to an intermediate from A_k, we take each intermediate $[x] \in A_k$ and try adding each face to x that is allowable under the Building Game rules. This means we look at every face $f \in F \setminus x$ and look for another face $\hat{f} \in x$ such that f and $\hat{f}$ share an edge. For every such face f, we look at the new $(k+1)$-faced state $y \doteq x \cup \{f\}$. Since we know that $[y]$ is a Building Game intermediate, it must be represented in A_{k+1}, however before adding y to A_{k+1}, we must verify that there is no $\hat{y}$ already in A_{k+1} such that $y \in [\hat{y}]$.

The act of comparing two states y and $\hat{y}$ to check if they are members of the same intermediate is the task where the majority of computational time is spent. The brute force method of checking if $y \sim \hat{y}$ involves checking if $g.y = \hat{y}$ for each $g \in G$. If a g is found that makes this equality hold, then $[y] = [\hat{y}]$ and the computation terminates and we know that $[y]$ is not a new intermediate. In this case, nothing is added to A_{k+1} but a connection $x \leftrightarrow \hat{y}$ is added in the combinatorial configuration space. Furthermore, by tracking how many time a particular connection $x \leftrightarrow \hat{y}$ is found, the forward degeneracy numbers can be computed quickly.

$A_1 \leftarrow \{\{f\} : f \in F\}$
for $\{f\} \in A_1$ **do**
  if $\exists \{\hat{f}\} \in A_1 \setminus \{f\} : [\{\hat{f}\}\ ] = [\{f\}]$ **then**
    $A_1 \leftarrow A_1 \setminus \{\{f\}\}$
  end if
end for
$A_2, \ldots, A_{|F|} \leftarrow \{\}$
for $k = 0, \ldots, |F| - 1$ **do**
  for $x \in A_k$ **do**
    for $f \in F \setminus x$ such that $\exists \hat{f} \in x$ with f and $\hat{f}$ sharing an edge. **do**
      NewIntermediate $\leftarrow True$
      for $\hat{y} \in A_{k+1}$ **do**
        if $y \in [\hat{y}]$ **then**
          NewIntermediate $\leftarrow False$
          Add connection $[x] \leftrightarrow [\hat{y}]$ to combinatorial configuration space.
        end if
      end for
      if NewIntermediate $= True$ **then**
        $A_{k+1} \leftarrow A_{k+1} \cup \{y\}$
        Add connection $[x] \leftrightarrow [y]$ to combinatorial configuration space.
      end if
    end for
  end for
end for

Figure 3.13: Algorithm for iteratively enumerating the Building Game combinatorial configuration space.

3.3.1 Efficient Congruence Testing

Similar to the approach of [22], we implement a hash function $\mathfrak{h}$ that maps each state to an integer with the property that $[y] = [\hat{y}]$ if and only if $\mathfrak{h}(y) = \mathfrak{h}(\hat{y})$. By designing such a function, checking for congruence between two states just amounts to a comparison of the states' hash values.

Given an enumeration of the polyhedron's faces, each state may be represented by a binary vector giving each face's inclusion or exclusion in the state. The hash we use is the base ten integer corresponding to the minimal binary representation of

the state taken over all possible rotations

$$\mathfrak{h}(y) = \min_{g \in G} \text{int}(g.y). \tag{3.15}$$

While this does require considering each possible rotation for every proposal state, this computation must only be performed once.

3.3.2 Data Structures

Before the computation of the combinatorial configuration space, we hard-code an enumeration $f_1, \ldots, f_{|F|}$ of the faces of our polyhedron. Then, any state x is represented by a binary vector of length $|F|$ with a one in the kth entry if $f_k \in x$ and zero otherwise. With this convention, each rotation $g \in G$ corresponds to a permutation of the indices of x. Each such permutation in the group is precomputed and then applied as necessary when performing a rotational comparison of two states. To track the connectivity structure of the polyhedron, an adjacency list on faces is stored as a two dimensional array. For each face number, the adjacency list specifies the index of adjacent faces.

Each group of k-faced intermediates A_k is stored as a hash table using the previously described hash function. This allows for order one look-up of intermediates already in the A_k that match the hash of a new proposal intermediate.

3.3.3 Run Time

The computing time required to enumerate the combinatorial configuration space is heavily dependent on the number of intermediates that are found. Since we do not have a tight bound on the number of intermediates as a function of simple statistics of the polyhedron (faces, edges, etc.), it is difficult to provide a meaningful estimates on the time required to compute the combinatorial configuration space without explicit knowledge of its size a priori.

That said, we can express an upper bound on the number of state to state comparisons that are required as a function of the intermediate sets $A_1, \ldots, A_{|F|}$.

$$\#\text{comparisons} \leq \sum_{k=1}^{|F|-1} \sum_{x \in A_k} \sum_{f \in F \setminus x} \sum_{\hat{y} \in A_{k+1}} 1 \tag{3.16}$$

$$\leq |F| \sum_{k=1}^{|F|-1} |A_k||A_{k+1}| \tag{3.17}$$

3.3.4 Implementation

All computation was carried out on a desktop computer running 64-bit Ubuntu 14.04 LTS with 15.6 GiB memory and a quad-core 3.20GHz Intel processor. To compute the Building Game combinatorial configurations space, we developed C++ code with the help of the Boost graph library for storing the connectivity structure. Computation of the combinatorial configuration space took less than a minute for most polyhedra, though–as expected–this time grew dramatically with the number of faces. Our largest case, the 30-faced Rhombic Triacontahedron took on the order

of 10 minutes to compute.

Additional computation–especially for pathway and shelling statistics–was completed using Python and the mpmath module for arbitrary precision arithmetic [11].

Chapter Four

Constraint Models and Embedding Intermediates in Space

While models such as the Building Game treat assembly intermediates as idealized structures, the intermediates of the physical application we are modeling may face unpredictable forces. To realistically model the ways in which our intermediate might flex and move under these forces, we use a constraint model in which individual faces of an intermediate are rigid, but with the edges at which two faces meet modeled as a hinge. As seen in figure 4.1, this approach allows for a variety of geometric configurations while the combinatorial connectivity properties of the intermediate remain unchanged. This geometric framework provides a physically motivated way of addressing questions that strictly combinatorial models cannot adequately answer.

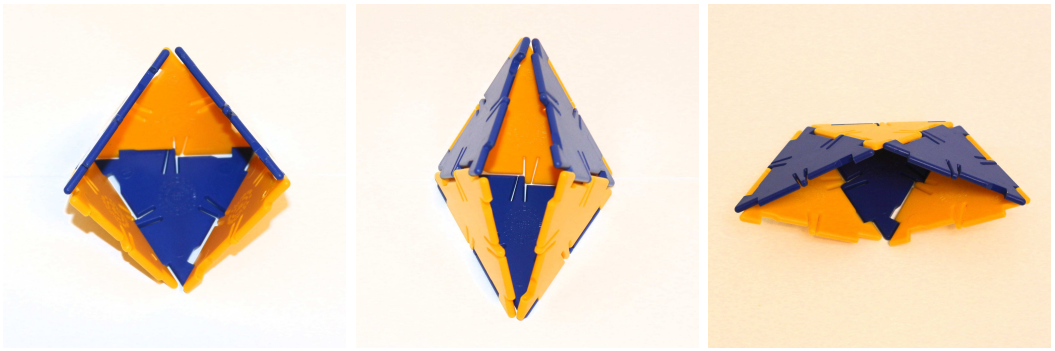

Figure 4.1: Different geometric configurations of an octahedron intermediate.

Definition 17. *The* ***canonical configuration*** *of a Building Game intermediate* $[x]$ *is the configuration that is the* 3*-dimensional embedding of the polyhedron restricted to the faces in* x

For example, in the case of the cube, the canonical configuration would have all of the faces of the intermediate meeting at right angles just as they do in the cube itself, however this is just one possible geometric configuration.

4.1 Geometric Configuration Space

Using a system of constraint functions, we can mathematically define the space of different geometric configurations that a given intermediate can take. Since we are treating each intermediate as collection of rigid polygons attached to each other along hinged edges, there are two basic types of constraint we must enforce. First, we must ensure that the each individual face is rigid and has the correct geometric shape. Additionally, each edge to edge connection must remain fixed with only hinge-like motion allowed. By defining this set of constraints as a single constraint function $c : \mathbb{R}^n \to \mathbb{R}^{n-m}$, the solution set $\Omega \doteq \{z \in \mathbb{R}^n : c(z) = 0\}$ is the set of points where all of the constraints are simultaneously satisfied. This constraint system implicitly specifies all of the ways in which an intermediate has freedom to move if it is able to move at all. If c is composed of polynomial functions, then the solution set is an algebraic variety. If this variety has no singularities, then it is also a manifold.

Definition 18. *The* ***geometric configuration space*** *of a Building Game intermediate is an algebraic variety corresponding to a system of polynomial constraints that enforces the rigidity of individual faces and hinged motion along connected edges.*

To mathematically describe a particular configuration, we must specify the locations of the vertices of each face. Thus, for an intermediate $[x]$, its Building Game configuration space can be represented as a subset of the ambient space $\mathbb{R}^{3\times N_x}$ where $N_x \doteq \sum_{f\in x} s_f$ where s_f is the number of sides (and vertices) of face f. Using this representation, we must then identify the corresponding constraint equations that give rise to the combinatorial configuration space as a function of points in ambient space.

It is worth noting that while we represent each face with $3 \times s_f$ coordinates, only

Constraint Type	Constraint Equation
edge length	$c_{edge}^{j,k}(z) = \left\|v^{j,k} - v^{j,k-1}\right\|^2 - (\ell_{j,k})^2$
angle	$c_{ang}^{j,k}(z) = (v^{j,k-1} - v^{j,k}) \cdot (v^{j,k+1} - v^{j,k}) - \ell^{j,k}\ell^{j,k+1}\cos(\theta^{j,k})$
2D face	$c_{2D}^{j,k}(z) = v^{1,k} + \ell^{j,k,1}R(v^{j,0} - v^{j,1}) - v^{j,k}$
vertex identification	$c_{ident}^{j_1,k_1,j_1,k_1,d}(z) = v_d^{j_1,k_1} - v_d^{j_2,k_2}$

Table 4.1: The different types of polynomial constraint equations used to describe the geometric configuration space.

6 are required to specify a face's position and orientation if they are chosen carefully. With this in mind, we will typically use a function of 6 of a face's vertex coordinates to constrain each of the face's remaining vertex coordinates.

Notationally, we refer to the kth vertex of the jth face of x as $v^{jk} = \left(v_x^{jk}, v_y^{jk}, v_z^{jk}\right)$. Since the context of a solution variety requires a function $c : \mathbb{R}^n \to \mathbb{R}^{n-m}$, we flatten the matrix of vertex coordinates into a vector of length $n = 3N_x$.

$$z = \begin{bmatrix} v^{1,1} \\ \vdots \\ v^{1,s_{f_1}} \\ \vdots \\ v^{|x|,1} \\ \vdots \\ v^{|x|,s_{f_{|x|}}} \end{bmatrix} \in \mathbb{R}^n \tag{4.1}$$

There are four fundamental types of constraint equations: edge length constraints, angle constraints, and $2D$ face constraint to enforce the rigid structure of each face as well as vertex identification constraints to enforce the hinged connections. The equations for these constraints are outlined in table 4.1 with an extended discussion appearing in section 4.2

4.1.1 Special Case of Triangular Faces

In the case where all of the faces of the polyhedron we consider are triangles (tetrahedron, octahedron, icosahedron, etc.) we notice that by simple enforcing that each edge of each triangle has a specified length, the triangle will be rigid. This means that we do not have to use the angle and 2D face constraints. Further, rather than explicitly using vertex identification constraints, we can either treat them as length constraints with zero length between identified vertices or we can simply reindex the vertices so that identified vertices are actually treated as a single vertex. With either choice, in the triangular case, we may only deal with length constraints if we wish. This will be a useful property in the next chapter.

4.2 Degrees of Freedom

Roughly speaking, the degrees of freedom of a system are the different independent motions the system is able to exercise. Since the concept of degrees of freedom exists in many diverse scientific fields, such as mechanical engineering and statistical physics, many different formal definitions of degrees of freedom are used in the literature [5]. McCarthy defines degrees of freedom of a mechanical system as follows [18].

> We derive formulas for the number of parameters needed to specify the configuration of a mechanism, in terms of the number of links and joints and the freedom of movement allowed at each joint. This number is the *degrees of freedom* or *mobility* of the mechanism. Changing the values of these parameters changes the configuration of the mechanism. Thus,

> if we view the set of all configuration available to a mechanism as a manifold, then the mobility of the mechanism is the dimension of this manifold.

Since our geometric configuration space is an algebraic variety and not a manifold in general, this definition must be modified to make degrees of freedom a statistic of each individual configuration rather than a global statistic of the geometric configuration space.

Definition 19. *The number of* ***degrees of freedom*** *of a Building Game intermediate at configuration z is the dimension of the geometric configuration space at z. If z is a singularity of the algebraic variety, then the degrees of freedom is undefined.*

In most cases we consider, rigid rotation and rigid translation will preserve the value of the constraint function since they do not move the vertices relative to one another. In other words, if $c(z) = 0$ then we also have $c(Rz + T) = 0$ where $R \in \mathbb{R}^n \times n$ rotates each vertex in the configuration by some $\hat{R} \in SO(3)$ and translates each vertex by $\hat{T} \in \mathbb{R}^3$. With this definition, R is the block diagonal matrix with each of the $n/3$ block being $\hat{R} \in SO(3)$ and $T \in \mathbb{R}^n$ is composed of $n/3$ copies of $\hat{T} \in \mathbb{R}^3$ stacked upon each other. These rigid body rotations account for 6 of the configuration's degrees of freedom: 3 rotational degrees of freedom and 3 translational and are called the *trivial degrees of freedom.* Thus, the degrees of freedom we are most interested in are those that do represent movement of the vertices and faces relative to each other. The *internal degrees of freedom* a Building Game intermediate at configuration z are the degrees of freedom that are not trivial.

4.2.1 Computing Degrees of Freedom

For a generic solution variety, we can find the dimension of the space at a point z by looking at the Jacobian matrix $C(z) \in \mathbb{R}^{(n-m)\times n}$ of the constraint function c. Since the number of degrees of freedom is defined to be the dimension of that space, we look at the rank of the Jacobian. Since this rank quantifies the number of independent constraints given by c at z, the number of degrees of freedom is given by the following.

$$DoF \doteq n - \text{rank}\left((C(z)\right) \tag{4.2}$$

Since the rank can take values between 0 and $\min\{n, n-m\} = n-m$, there can be anywhere from 0 degrees of freedom, when there are functionally no constraints on z, and m degrees of freedom in the case where all $n-m$ constraint equations are independent and $C(z)$ is of full rank.

In the typical case in which the constraint equations are invariant under three-dimensional rotation and translation, there will be six trivial degrees of freedom. The number of internal degrees of freedom would then be $n-\text{rank}\left((C(z)\right)-6$. If there are zero internal degrees of freedom at a configuration z, we say that the configuration is *rigid*.

To actually compute the number of degrees of freedom for configurations of Building Game intermediates, we must first find an explicit form for the the constraint function c and its Jacobian matrix C.

Edge length constraints enforce that the lengths of the edges of each face in an intermediate cannot change. If the kth edge is defined to be that between the

$(k-1)$st and kth vertices, we use the following function to constrain its lengths to a known value ℓ^{jk}.

$$c_{edge}^{j,k}(z) = \left|v^{j,k} - v^{j,k-1}\right|^2 - (\ell_{j,k})^2 \tag{4.3}$$

$$= \left(v_1^{j,k} - v_1^{j,k-1}\right)^2 + \left(v_2^{j,k} - v_2^{j,k-1}\right)^2 + \left(v_3^{j,k} - v_3^{j,k-1}\right)^2 - (\ell^{j,k})^2 \tag{4.4}$$

This uses the notational convention that $v^{j,0} \doteq v^{j,s_j}$. For reasons that will be explained, we only explicitly enforce the lengths of two edges ($k = 1, 2$) per face, so there are a total of $2|x|$ edge length constraints.

Angle constraints ensure that each face's polygonal angles are conserved. Using the dot product formula for angles, we can write this constraint as a polynomial.

$$c_{ang}^{j,k}(z) = (v^{j,k-1} - v^{j,k}) \cdot (v^{j,k+1} - v^{j,k}) - \ell^{j,k}\ell^{j,k+1}\cos(\theta^{j,k}) \tag{4.5}$$

Here, $\theta^{j,k}$ is the angle at $v^{j,k}$ between kth and $(k+1)st$ edges. This angle is a constant that is known a priori as it is one of the polygonal angles appearing in a rigid subunit. In practice, we only enforce that the $k = 1$ angle constraint for each face.

Now, between the two edge constraints and one angle constraints, we have described 3 constraints as a function of 9 vertex coordinates per face. Thus for any choice of the 6 coordinates, $v_1^{j,1}, v_2^{j,1}, v_3^{j,1}, v_1^{j,2}, v_2^{j,2}, v_1^{j,0}$, the remaining 3 coordinates, $v_3^{j,2}, v_2^{j,0}, v_3^{j,0}$, are specified by the 3 constraint equations. Similarly, since each face's location and rotational orientation can be defined by this choice of 6 coordinates, we have enough information to specify the coordinates of the remaining vertices.

Since the positions of the first three vertices dictate the locations of the remain-

ing vertices, the $2D$ face constraints use a map from the known vertex coordinates to the yet unknown locations. Using a template for what the ideal polygonal structure for each face should be, we use a rotation matrix to specify these remaining vertices. If this template has vertices $\hat{v}^{j,0}, \hat{v}^{j,1}, \hat{v}^{j,2}, \ldots, \hat{v}^{j,k}, \ldots$ and the locations for $v^{j,0}, v^{j,1}$, and $v^{j,2}$ are known, we can identify the location of $v^{j,k}$ for $k > 2$. Using this template, we can define the following length and angle constants.

$$\ell^{j,k_1,k_2} \doteq |\hat{v}^{j,k_1} - \hat{v}^{j,k_2}| \tag{4.6}$$

$$\phi^{j,k_1,k_2,k_3} \doteq \cos^{-1}\left(\frac{\left(\hat{v}^{j,k_1} - \hat{v}^{j,k_2}\right) \cdot \left(\hat{v}^{j,k_3} - \hat{v}^{j,k_2}\right)}{\left(\ell^{j,k_1,k_2}\right)\left(\ell^{j,k_3,k_2}\right)}\right) \tag{4.7}$$

Our basic strategy is to first place a point $\bar{v}^{j,k}$ in the span of $v^{j,0} - v^{j,1}$ at a distance of $|\bar{v}^{j,k} - v^{j,1}| = \ell^{j,k,1}$. The choice $\bar{v}^{j,k} = v^{j,1} + \frac{\ell^{j,k,1}}{\ell^{j,0,1}}(v^{j,0} - v^{j,1})$ will work, since

$$|\bar{v}^{j,k} - v^{j,1}| = |\frac{\ell^{j,k,1}}{\ell^{j,0,1}}(v^{j,0} - v^{j,1})| \tag{4.8}$$

$$= \frac{\ell^{j,k,1}}{\ell^{j,0,1}}|v^{j,0} - v^{j,1}| \tag{4.9}$$

$$= \ell^{j,k,1}. \tag{4.10}$$

Then, a rotation matrix is used to rotate $\bar{v}^{j,k}$ by the correct angle into its position $v^{j,k}$. The rotation matrix is centered at $v^{j,1}$ and its axis of rotation is defined by $u = \frac{1}{\ell^{j,0,1}\ell^{j,2,1}}(v^{j,0} - v^{j,1}) \times (v^{j,2} - v^{j,1})$. Similarly, the angle of rotation $\phi^{j,0,1,k}$ is the angle created by the two line segments in the template $(\hat{v}^{j,0}, \hat{v}^{j,1})$ and $(\hat{v}^{j,2}, \hat{v}^{j,1})$. Thus, using $R = R(\phi^{j,0,1,k}, u)$ and our equation for $v^{j,k}$ is

$$v^{j,k} = v^{1,k} + R(\bar{v}^{j,k} - v^{j,1}) \tag{4.11}$$

$$= v^{1,k} + \ell^{j,k,1}R(v^{j,0} - v^{j,1}) \tag{4.12}$$

Since R is polynomial in $v^{j,0}, v^{j,1}, v^{j,2}$, we get the following polynomial 2D face constraint for each $k > 2$.

$$c_{2D}^{j,k}(z) \doteq v^{1,k} + \ell^{j,k,1} R(v^{j,0} - v^{j,1}) - v^{j,k} \tag{4.13}$$

The final constraint type, vertex identification, is used to enforce that the connection between edges of two faces has the mobility of a hinge. To do this, we simply need to ensure that that corresponding vertices on each edge share identical locations. This results in the relatively simple constraints:

$$c_{ident}^{j_1,k_1,j_1,k_1,d}(z) \doteq v_d^{j_1,k_1} - v_d^{j_2,k_2} \tag{4.14}$$

$$\tag{4.15}$$

where v^{j_1,k_1} and v^{j_2,k_2} are corresponding vertices from the faces j_1 and j_2 meeting at a hinged edge. If there are $|E_x|$ of these hinged connections in a Building Game state x, there must be $6|E_x|$ corresponding vertex identification constraints.

With all four constraint types explicitly defined, the aggregate constraint function for x will have $n-m = 2|F_x|+|F_x|+(N_x-3|F_x|)+6|E_x| = N_x+6|E_x|$ total constraints.

Below are the partial derivatives of the constraint functions we specified above.

$$\frac{\partial c_{edge}^{j,k}}{\partial z_i} = \begin{cases} 2\left(v_d^{j,k} - v_d^{j,k-1}\right) & \text{if } z_i = v_d^{j,k} \\ -2\left(v_d^{j,k} - v_d^{j,k-1}\right) & \text{if } z_i = v_d^{j,k-1} \\ 0 & \text{else} \end{cases} \tag{4.16}$$

$$\frac{\partial c_{ang}^{j,k}}{\partial z_i} = \begin{cases} v_d^{j,k+1} - v_d^{j,k} & \text{if } z_i = v_d^{j,k-1} \\ 2v_d^{j,k} - v_d^{j,k-1} - v_d^{j,k+1} & \text{if } z_i = v_d^{j,k} \\ v_d^{j,k-1} - v_d^{j,k} & \text{if } z_i = v_d^{j,k+1} \\ 0 & \text{else} \end{cases} \tag{4.17}$$

$$\frac{\partial c_{2D}^{j,k}}{\partial z_i} = \begin{cases} & \text{if } z_i = v_d^{j,0} \\ & \text{if } z_i = v_d^{j,1} \\ & \text{if } z_i = v_d^{j,2} \\ -1 & \text{if } z_i = v_d^{j,k} \\ 0 & \text{else} \end{cases} \tag{4.18}$$

$$\frac{\partial c_{ident}^{j_1,k_1,j_2,k_2,d}}{\partial z_i} = \begin{cases} 1 & \text{if } z_i = v_d^{j_1,k_1} \\ -1 & \text{if } z_i = v_d^{j_2,k_2} \\ 0 & \text{else} \end{cases} \tag{4.19}$$

Since the number of degrees of freedom is defined as a local property of a specific configuration z, different configurations of the same Building Game intermediate could theoretically have differing numbers of degrees of freedom. However, we have yet to observe a Building Game intermediate that has this property. A contrived case in which a linkage of squares leads to a configuration space with differing degrees of freedom is described in section 4.2.2. When we report the number of degrees of freedom that a Building Game intermediate has, we use its canonical configuration in computation.

In practice, we use NumPy's numpy.linalg.svd module to take the singular value decomposition of the compute Jacobian matrix in Python. The rank of the Jacobian

Figure 4.2: A linkage of 6 squares with configurations of different degrees of freedom.

is then the number of effectively non-zero singular values where a small cutoff is used to determine whether a small singular value is indeed zero or non-zero.

4.2.2 A Configuration Space that is no a Manifold

Consider the linkage of six squares arranged in a 2×3 lattice depicted in figure 4.2. This linkage has three lines of hinges: 2 across and 1 down. As shown, the two horizontal hinges can be manipulated independently with each of the vertical hinges maintaining at an angle of π degrees. This seems to indicate two internal degrees of freedom. Alternatively, when folded along the vertical crease, the horizontal hinges remain fixed with dihedral angle π. This configuration corresponds to just a single internal degree of freedom. Thus, the transition between these two modes, when the linkage is completely planar, there is a singularity and the number of degrees of freedom is not defined.

As stated previously, we have not observed this type of behavior from any Building Game intermediates. This leads us to speculate that the degenerate nature of this example is down to an uncommon alignment of the hinged edges. Clearly, this linkage could never be an intermediate of a convex polyhedron. Perhaps the Building Game geometric configuration space has no singularities and is hence a manifold for

all convex polyhedra.

4.2.3 Results

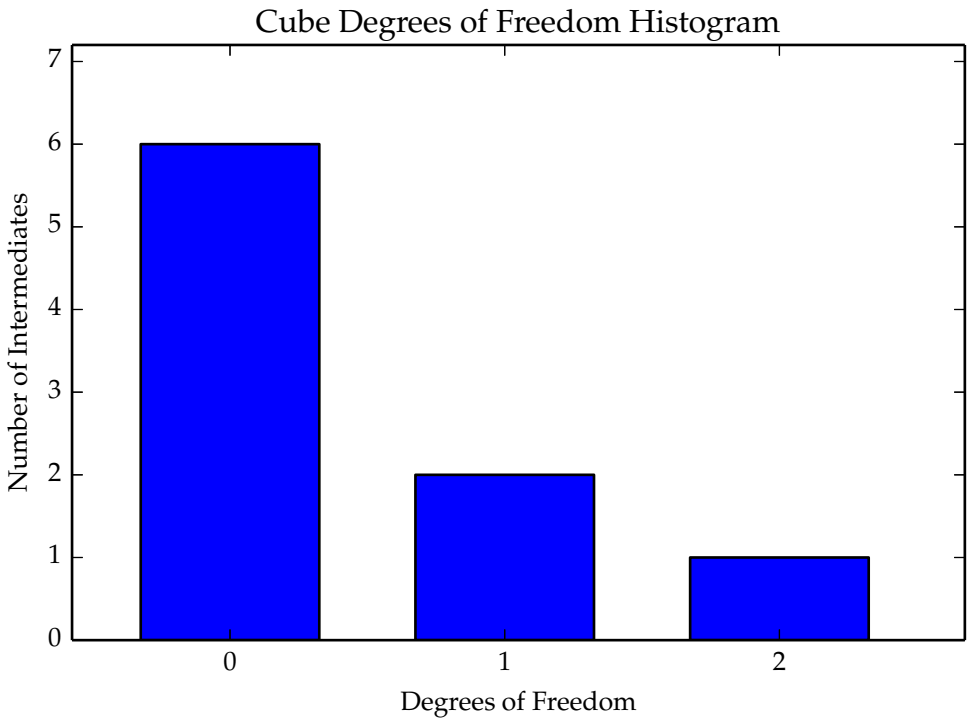

Figure 4.3: Distribution of internal degrees of freedom amongst Cube intermediates.

Using the Jacobian method for computing internal degrees of freedom, we have computed the number of degrees of freedom for all intermediates of the Platonic solids. Figures 4.3, 4.4, 4.5, and 4.6 are histograms of the number of internal degrees of freedom for the intermediates of each of the Platonic solids. We do not include the Tetrahedron, since it only has four intermediates and only the one composed of two triangles is not rigid. Interestingly, the cube and dodecahedron have a large number of rigid intermediates due to the fact that their vertices are the rigid meeting of only three faces. The octahedron, with four faces meeting at each vertex has a more varied distribution of internal degrees of freedom. The icosahedron, however,

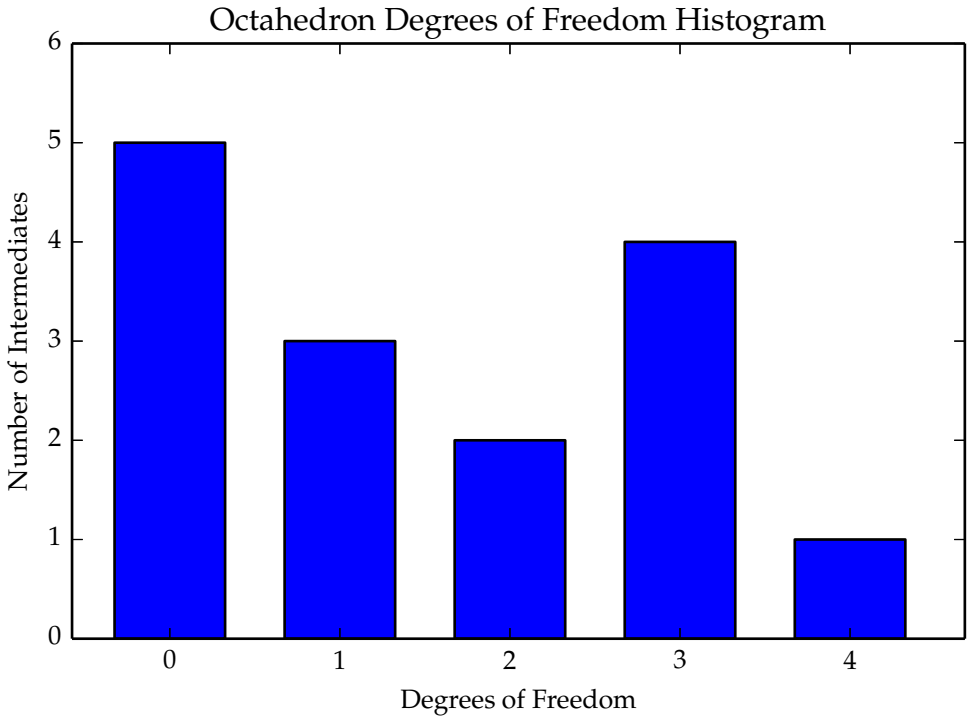

Figure 4.4: Distribution of internal degrees of freedom amongst Octahedron intermediates.

has a very small number of rigid intermediates with some intermediates having as many as 13 internal degrees of freedom.

To provide a more detailed look at the distribution of internal degrees of freedom, figures 4.7, 4.8, 4.9, and 4.10 break down the previous histograms by the number of faces in the intermediate. Each column of the plots is a histogram for the number of internal degrees of freedom for the intermediates with that specific number of faces. The general theme seems to be that intermediates with almost all or almost none of their faces are most rigid and the intermediates with around half of the total possible faces have the most internal degrees of freedom. The most interesting case remains the icosahedron with a wide variation in internal degrees of freedom for most face counts.

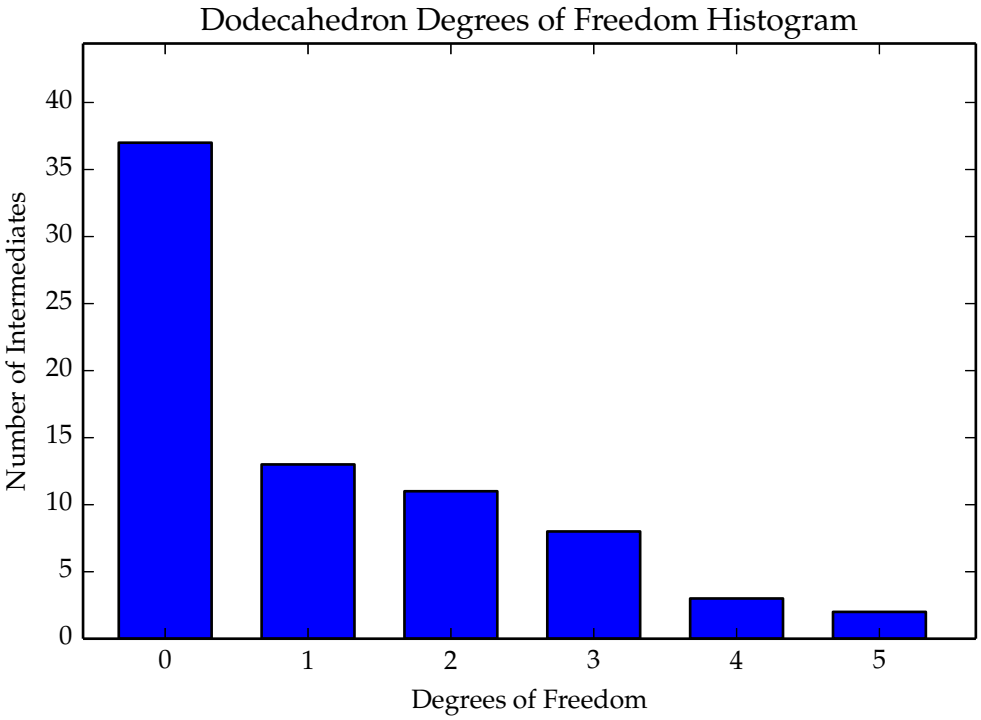

Figure 4.5: Distribution of internal degrees of freedom amongst Dodecahedron intermediates.

4.2.4 Explicit Removal of Trivial Degrees of Freedom

Since we are often solely interested the internal degrees of freedom, there are a few was to augment the constraint equations to mod out the translation or rotations causing the trivial degrees of freedom. One possible choice is to pick a face and constrain its vertex locations to match a fixed template. The corresponding constraint equations for fixing the jth face would be

$$c_{fix}^{j,k,d}(z) = v_d^{j,k} - \hat{v}_d^{j,k} \tag{4.20}$$

$$\frac{\partial c_{fix}^{j,k,d}}{\partial z_i} = \begin{cases} 1 & \text{if } z_i = v_d^{j,k} \\ 0 & \text{else} \end{cases} \tag{4.21}$$

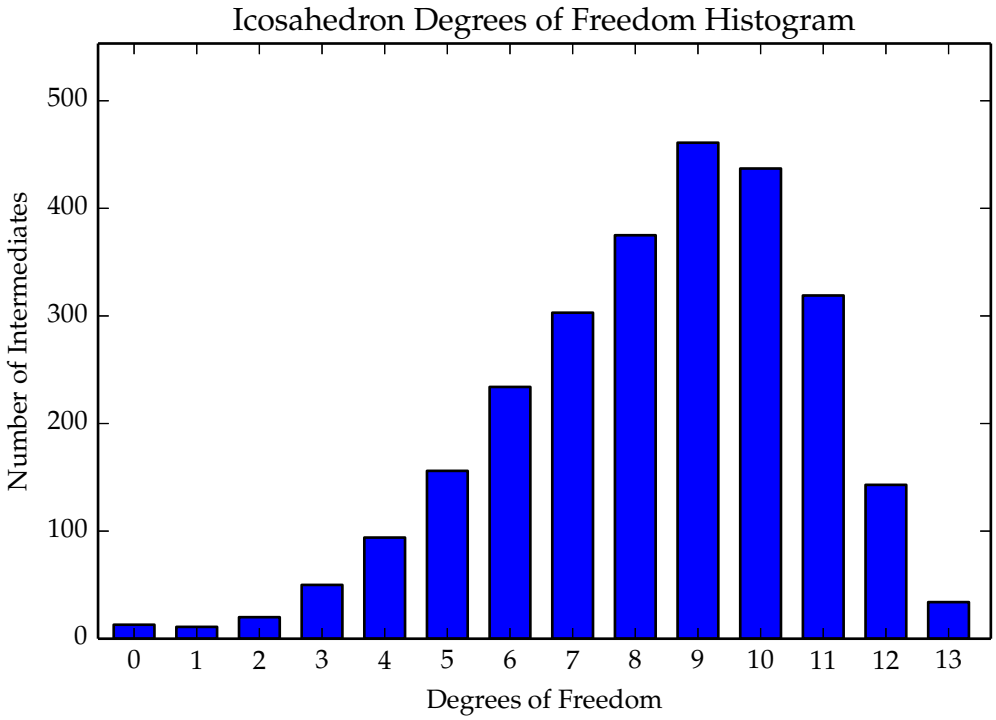

Figure 4.6: Distribution of internal degrees of freedom amongst Icosahedron intermediates.

for each dimension d of each vertex k of face j. For reasons we will explain in the next chapter, this may not always be an ideal choice if there is information other than the degrees of freedom we which to derive from the constraint equations.

An alternative that prevents translation, but does allow rotation is to fix the configuration's center of mass.

$$c_{com}^{d}(z) = \sum_{j=1}^{|F_x|} \sum_{k=1}^{s_{f_j}} v_d^{j,k} \tag{4.22}$$

$$\frac{\partial c_{com}^{d}}{\partial z_i} = \begin{cases} 1 & \text{if } z_i = v_d^{j,k} \\ 0 & \text{else} \end{cases} \tag{4.23}$$

This amounts to the sum over the dth entry of each vertex in the configuration.

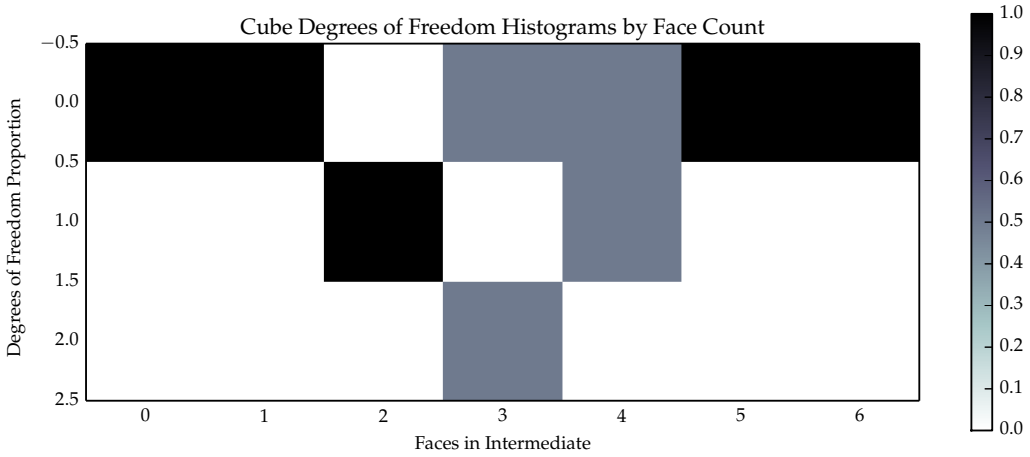

Figure 4.7: Each vertical column gives a histogram of the number of internal degrees of freedom k-faced intermediates of the cube have.

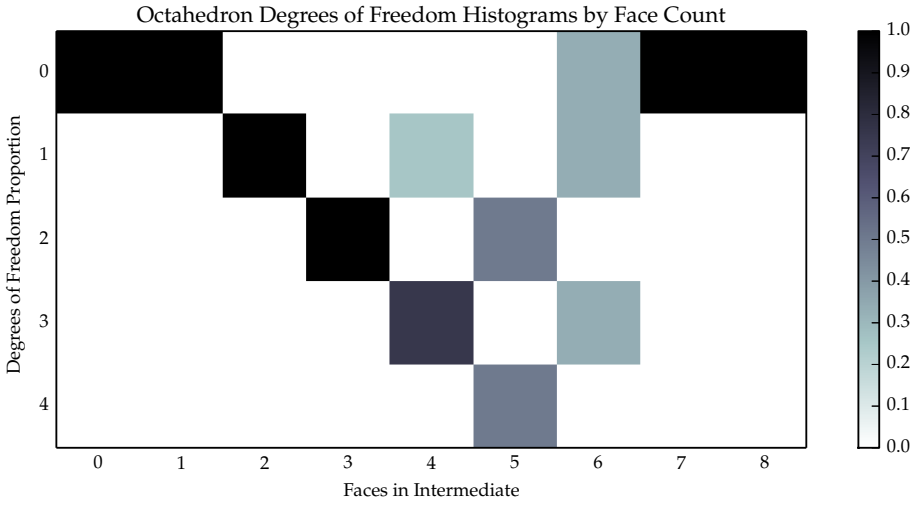

Figure 4.8: Each vertical column gives a histogram of the number of internal degrees of freedom k-faced intermediates of Octahedron have.

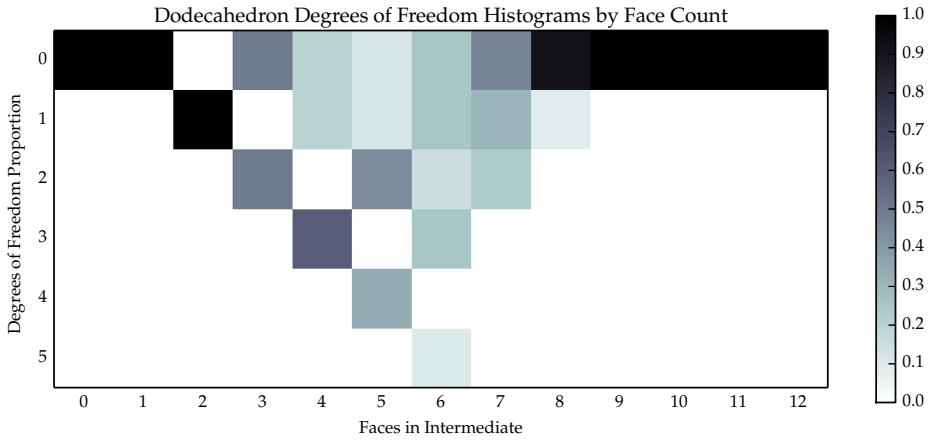

Figure 4.9: Each vertical column gives a histogram of the number of internal degrees of freedom k-faced intermediates of the Dodecahedron have.

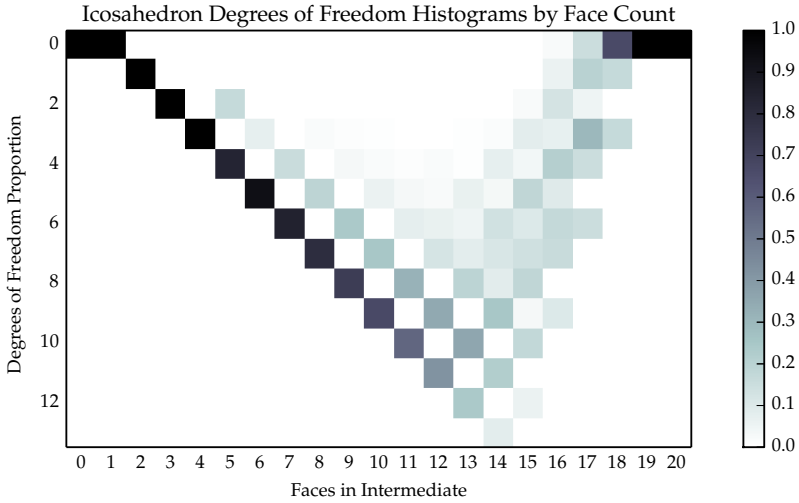

Figure 4.10: Each vertical column gives a histogram of the number of internal degrees of freedom k-faced intermediates of the icosahedron have.

Chapter Five

Processes in Constraint Spaces

5.1 Constrained Dynamics

Sometimes we are interested in more than simply the number of degrees of freedom a particular configuration has. To explore the configuration space, find similarities between different configurations, analyze the geometric configuration space's topology, or compute other relevant statistics, it can be useful to compute dynamic processes on the geometric configuration space. There are clear computational challenges to computing such dynamics due to computational handling of constraints, the lack of an explicit parameterizations, and the fact that geometric configuration space is often of much smaller dimension than the ambient space it sits in.

5.2 Brownian Motion on a Manifold

To explore various properties of the Building Game geometric configuration spaces, we simulate a Brownian motion on the corresponding constraint space. Assuming that the constraint space has no singularities, it is a manifold. Brownian motion on a Riemannian manifold is defined as a diffusion process generated by $\Delta/2$ where Δ is the Laplace-Beltrami operator

$$\Delta = g_{ij}\frac{\partial^2}{\partial x_i \partial x_j} - g_{ij}\Gamma^k_{ij}\frac{\partial}{\partial x_k} \text{ [9].} \tag{5.1}$$

Since the geometric configuration space is defined implicitly by constrain functions, an analytic solution of the Brownian density function is not possible. Using a two step scheme, we use a random walk to approximate the Brownian motion. To compute each new random walk step, we first take a step in the tangent space and then project this point back to the manifold.

5.2.1 Computational Scheme

Suppose we have a m dimensional manifold $\mathcal{M} \subset \mathbb{R}^n$ defined implicitly by $\mathcal{M} \doteq \{z \in \mathbb{R}^n : c(z) = 0\}$ for some function $c : \mathbb{R}^n \to \mathbb{R}^{n-m}$ with $c_1, \ldots, c_{n-m}$ independent constraint functions. We seek to approximate a Brownian motion on $\mathcal{M}$ by constructing a random walk. As mentioned before, this is done in two stages; starting at a point $z \in \mathcal{M}$ we take a step of size $\sqrt{\Delta t}$ in a tangent direction w and then we project the point $z + \sqrt{\Delta t} w$ back onto $\mathcal{M}$.

Sampling in Tangent Space

At each point $z \in \mathcal{M}$ the tangent space is $\mathcal{T}_z\mathcal{M} \doteq \{w \in \mathbb{R}^n : C(z)\, w = 0\}$ where $C : \mathbb{R}^n \to \mathbb{R}^{(n-m)\times n}$ is the Jacobian of c. Thus, sampling from the tangent space reduces to sampling from the null space of $C(z)$.

First we must construct a basis for $\mathcal{T}_z\mathcal{M}$. Let $A = [C^T(z)B]$ be the concatenation of $C^T(z)$ and a randomly drawn matrix $B \in \mathbb{R}^{n\times m}$. We assume each entry of B is independent with $B_{jk} \sim \text{unif}(0, 1)$.

Lemma 5. *The matrix A is of full rank with probability 1.*

Proof. Since the constraints of c are independent, C is of full rank. Additionally, since B is drawn randomly, each column of B will be independent of the columns of C^T and also the other columns of B almost surely. Thus, the columns of A are independent and A is of full rank. □

Now, we compute the QR decomposition

$$A = \begin{bmatrix} C^T B \end{bmatrix} = QR = \begin{bmatrix} Q^{(1)} Q^{(2)} \end{bmatrix} R \tag{5.2}$$

where $Q^{(1)} \in \mathbb{R}^{(n-m)\times n}$ and $Q^{(2)} \in \mathbb{R}^{m\times n}$.

Theorem 8. *The columns of $Q^{(2)}$ are an orthonormal basis for $\mathcal{T}_z\mathcal{M} = N(C)$ and the columns of $Q^{(1)}$ are an orthonormal basis for $(\mathcal{T}_z\mathcal{M})^\perp = (N(C))^\perp$.*

Proof. The second claim is true by simple linear algebra arguments and the properties of the QR decomposition.

$$\begin{aligned} \text{col}(Q^{(1)}) &= \text{col}(C^T) && (5.3)\\ &= ((\text{col}(C^T))^\perp)^\perp && (5.4)\\ &= (N((C^T)^T)^\perp && (5.5)\\ &= (N(C))^\perp && (5.6) \end{aligned}$$

Here we use the vector space identities $X = (X^\perp)^\perp$ and $(\text{col}X)^\perp = N(X^T)$. This shows that columns of $Q^{(1)}$ are an orthonormal basis for $(\mathcal{T}_z\mathcal{M})^\perp$. Now, since

$$\begin{aligned} \text{col}(Q^{(1)}) \oplus \text{col}(Q^{(2)}) &= \text{col}(A) && (5.7)\\ &= \mathbb{R}^n && (5.8)\\ &= (N(C)) \oplus (N(C))^\perp && (5.9)\\ &= (N(C)) \oplus \text{col}(Q^{(1)}) && (5.10) \end{aligned}$$

we must have that $\text{col}(Q^{(2)}) = N(C)$ and that the columns of $Q^{(2)}$ provide an orthonormal basis for $\mathcal{T}_z\mathcal{M}$. □

Now, with our orthonormal basis for $\mathcal{T}_z\mathcal{M}$, we can define any $w \in \mathcal{T}_z\mathcal{M}$ by a vector $\alpha \in \mathbb{R}^m$ as follows.

$$w = \sum_{i=1}^{m} \alpha_i Q_i^{(2)} = Q^{(2)}\alpha$$

So, for any two $w, \tilde{w} \in \mathcal{T}_z\mathcal{M}$ we can define the manifold metric g as

$$g(w, \tilde{w}) = g\left(\sum_{i=1}^{m} \alpha_i Q_i^{(2)}, \sum_{j=1}^{m} \tilde{\alpha}_j Q_j^{(2)}\right) \tag{5.11}$$

$$= \sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \tilde{\alpha}_j g\left(Q_i^{(2)}, Q_j^{(2)}\right) \tag{5.12}$$

$$= \sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \tilde{\alpha}_j G_{ij} \tag{5.13}$$

$$= \alpha^T G \tilde{\alpha} \tag{5.14}$$

Then, our goal is to sample uniformly from $\Omega_G = \left\{w = Q^{(2)}\alpha : \alpha^T G \alpha = 1\right\}$. Since this set corresponds to a level set of the Multivariate normal distribution with mean zero and covariance matrix G^{-1}, our sampling problem reduces to sampling $u \sim \mathcal{N}(0, G^{-1})$. Given u, $w \doteq \frac{u}{|u|}$ is sampled from Ω_G as desired. We generally choose $G = I$, but this is not required.

Projection onto $\mathcal{M}$

After making a step away from z of size $\sqrt{\Delta t}$ in the direction $w \in N(C)$, our new point $z + \sqrt{\Delta t} w$ is not likely to be in $\mathcal{M}$. To return our point to $\mathcal{M}$, we make a projection step $w^{\perp} \in (N(C))^{\perp}$. Since $Q^{(1)}$ provides a basis for $(N(C))^{\perp}$, we can write any such step as

$$w^{\perp} = \sum_{i=1}^{n-m} \gamma_i Q_i^{(1)} = Q^{(1)}\gamma.$$

with $\gamma \in \mathbb{R}^{n-m}$. Thus, we want to identify a γ such that $c(z + \sqrt{\Delta t}w + Q^{(1)}\gamma) = 0$. Since c is nonlinear, we use Newton-Raphson iteration to solve for γ. By defining an objective function F and its Jacobian $\mathcal{J}$ as

$$F(\gamma) = c(z + \sqrt{\Delta t}w + Q^{(1)}\gamma) \tag{5.15}$$

$$\mathcal{J}_{jk}(\gamma) = \frac{\partial F_j}{\partial \gamma_k}(\gamma) \tag{5.16}$$

$$= \sum_{i=1}^{n-m} \frac{\partial c_j}{\partial z_i}(z + \sqrt{\Delta t}w + Q^{(1)}\gamma)Q^{(1)}_{ik} \tag{5.17}$$

$$\mathcal{J}(\gamma) = C(z + \sqrt{\Delta t}w + Q^{(1)}\gamma)Q^{(1)} \tag{5.18}$$

we arrive at the iteration routine below, which typically commences with the initial guess $\gamma_0 = 0$.

$$C(z + \sqrt{\Delta t}w + Q^{(1)}\gamma)Q^{(1)}\left(\gamma_{k+1} - \gamma_k\right) = \mathcal{J}(\gamma_k)\left(\gamma_{k+1} - \gamma_k\right) \tag{5.19}$$

$$= -F(\gamma_k) \tag{5.20}$$

$$= -c(z + \sqrt{\Delta t}w + Q^{(1)}\gamma) \tag{5.21}$$

Sampling Algorithm

5.2.2 Validation and Test Cases

An interesting test of our sampling scheme is unitary matrices. Since unitary matrices are complex matrices that satisfy $U^*U = UU^* = I$, we can use a system constraints to represent the space of unitary $N \times N$ matrices as an algebraic variety sitting in a $n = 2N^2$ dimensional ambient space. If we use the coordinates $U = A + iB$ for real, $N \times N$ matrices A and B. We use the constraints given by the

for $k = 1, \ldots, N$ **do**
  $B \leftarrow \text{unif}(0,1)^{n \times m}$
  $A \leftarrow \left[C^T B\right]$
  $[Q^{(1)}, Q^{(2)}], R \leftarrow$QRdecomposition$(A)$
  $\alpha \leftarrow \mathcal{N}(0, G^{-1})$
  $w \leftarrow \frac{Q^{(2)}\alpha}{|Q^{(2)}\alpha|}$
  $\gamma \leftarrow 0^{n-m}$
  while $|c(z + \sqrt{\Delta t}w + Q^{(1)}\gamma)| > \epsilon$ **do**
    $\Delta\gamma \leftarrow (C(z + \sqrt{\Delta t}w + Q^{(1)}\gamma)Q^{(1)})^{-1}(-c(z + \sqrt{\Delta t}w + Q^{(1)}\gamma))$
    $\gamma \leftarrow \gamma + \Delta\gamma$
  end while
  $z \leftarrow z + \sqrt{\Delta t}w + Q^{(1)}\gamma$
end for

Figure 5.1: Random Walk algorithm on $\mathcal{M}$.

following.

$$I = U^*U \tag{5.22}$$

$$= (A^T - iB^T)(A + iB) \tag{5.23}$$

$$= (A^TA + B^TB) + i(A^TB + B^TA) \tag{5.24}$$

$$\mathbb{1}_{j=k} = A_j \cdot A_k + B_j \cdot B_k \tag{5.25}$$

$$0 = A_j \cdot B_k + A_k \cdot B_j \tag{5.26}$$

Considering each $1 \leq j \leq N$ gives us $2 * (N + \frac{1}{2}N(N-1)) = N + N^2$ independent polynomial constraint equations.

Using our random walk scheme to simulate a Brownian motion of the constraint space. Considering first the case of $N = 3$, the ambient space is $\mathbb{R}^{18}$. The resulting constraint manifold has dimension 6 with the co-dimension 12 corresponding to the $12 = 3 + 3^2$ independent constraints imposed.

A uniform distribution according to the Haar measure corresponds to a uniform sampling of of the arguments θ of the matrices eigenvalues [19]. Using our scheme,

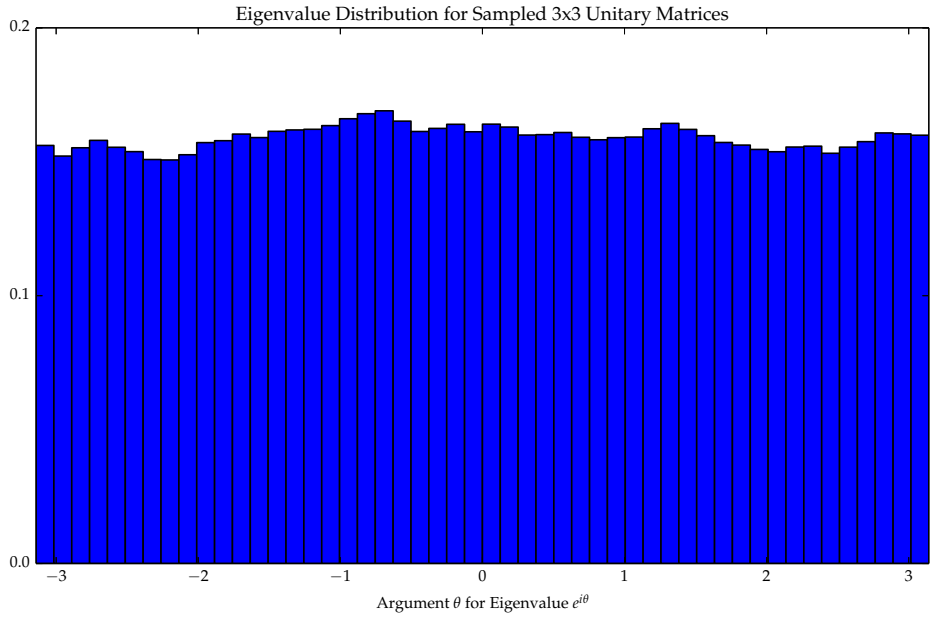

Figure 5.2: Empirical distribution of 3×3 Unitary matrices is uniform.

$3,000,000$ samples were drawn for both $N = 3$ and $N = 10$. Figures 5.2 and 5.3 show the eigenvalue distribution of our sampled matrices to be uniform.

5.2.3 Brownian Motion on Geometric Configuration Spaces

As an initial test case of our Brownian Motion scheme on Building Game geometric configuration spaces, we first consider two simple triangle linkages. The first is a dimer composed of two triangles linked together along a hinged edge. As seen in figure 5.4, the linkage can be specified by the 3-dimensional locations of four vertices, giving an ambient space of $\mathbb{R}^3$. In addition to the six trivial degrees of freedom, this configuration also carries an internal degree of freedom corresponding to the dihedral angle formed at the hinged edge. Figure 5.5 shows a histogram of this dihedral angle as sampled using out random walk scheme. Interestingly, the distribution is not uniform and appears to be of the form $\alpha_0 + \alpha_1 \cos(\theta) + \alpha_2 \cos(2\theta)$.

Figure 5.3: Empirical distribution of 10×10 Unitary matrices is uniform.

A second test case we consider is a linkage with three triangles. As depicted in figure 5.6, the configuration is composed of an interior triangle connected along two of its edges to an additional two triangles and is parameterized by five vertices. This configuration has two internal degrees of freedom, which are represented my the two dihedral angles. Figure 5.7 is a 2-dimensional histogram of the dihedral angles sampled by our scheme. As a test of convergence for our scheme, figure 5.8 show the Kolmogorov-Smirnov distance of the empirical distribution on the first dihedral angle of the three triangle linkage as a function of the number of samples. The line of best fit has a slope of -0.461. This is nearly -0.5 indicating that the expected square root rate of convergence is close to being achieved.

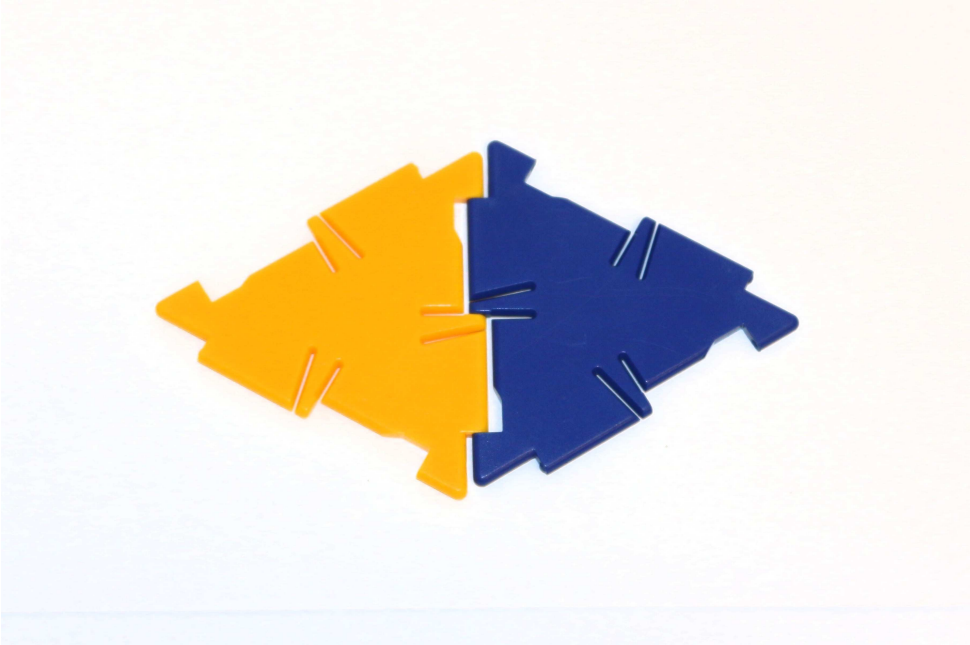

Figure 5.4: Two triangle linkage.

Fixing Trivial Degrees of Freedom

Since we are only interested in internal degrees of freedom and statistics invariant to rigid body movement, the inclusion of the six trivial degrees of freedom in our simulations may not be necessary. In fact, if the geometric configuration space is reduced by six dimensions, sampling efficiency will improve.

As mentioned in section 4.2.4, one way of fixing the translational degrees of freedom is to fix the center of mass of the configuration. While this method does not address the rotational degrees of freedom, it does appear to maintain the same distribution on the internal modes. In fact, the Kolmogorov-Smirnov distance between the sampled points for the first dihedral angle in figures 5.7 and 5.10 is just 0.00296 indicating an excellent agreement. Thus, it remains to account for the rotational degrees of freedom separately.

Now, to address rotations, we seek to find the three dimensional subspace of

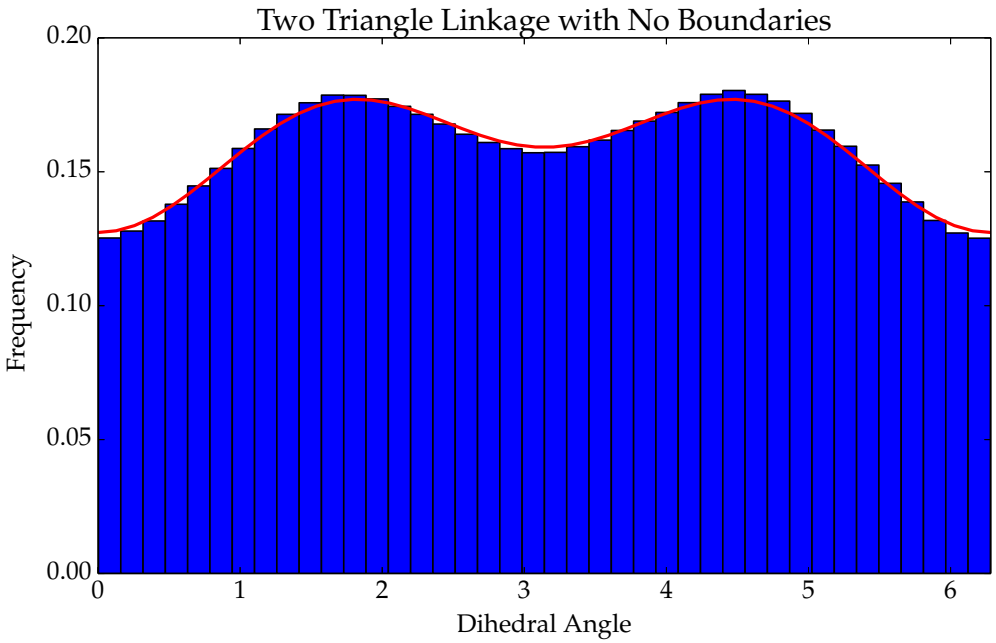

Figure 5.5: Histogram of the dihedral angle in sampled configurations of a two triangle linkage. The red curve has the form $\alpha_0 + \alpha_1 \cos(\theta) + \alpha_2 \cos(2\theta)$.

the tangent space and restrict our random walk to the space orthogonal to these rotational directions. Since such a step in the tangent space is then projected to the manifold, some rotation may occur in the projection step. To account for this, we must add a correction step that rotates the configuration back to its original frame of reference.

We treat our constraint space as the product of the special orthogonal group (rotations) and an algebraic variety corresponding to the remaining degrees of freedom $\mathcal{M} = SO(3) \times \tilde{\mathcal{M}}$. For every non-singular $z \in \mathcal{M}$ we notice that there is a local diffeomorphism from $\mathcal{M}$ to $SO(3) \times \tilde{\mathcal{M}}$. Using this knowledge, we aim to find local coordinates $Q \in SO(3)$ and $\tilde{z} \in \tilde{\mathcal{M}}$ such that the tangent space $\mathcal{T}_z\mathcal{M} = \mathcal{T}_Q SO(3) \times \mathcal{T}_{\tilde{z}}\tilde{\mathcal{M}}$. Since we already have a method for deriving $\mathcal{T}_z\mathcal{M}$, the goal now is to decompose this tangent space into the two orthogonal subspaces corresponding to the rotational and internal degrees of freedom.

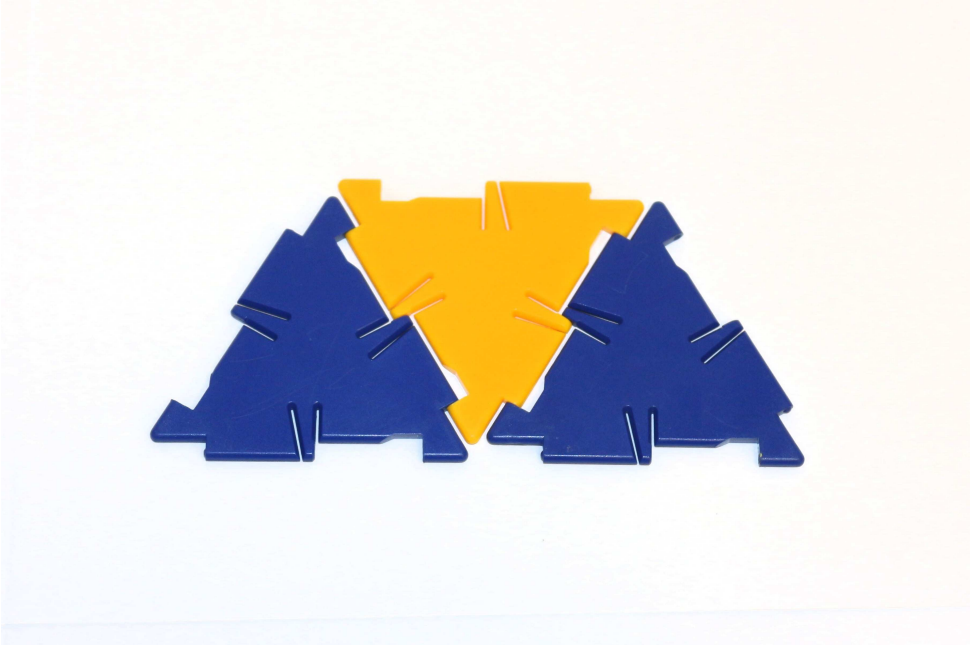

Figure 5.6: Three triangle linkage.

Since $z = (v^1, v^2, \ldots, v^N)$ for $v^i \in \mathbb{R}^3$, the rotation by a $Q \in SO(3)$ is really the vertex-wise rotation $\bar{Q}z = (Qv^1, Qv^2, \ldots, Qv^N)$. In the case of $Q = I$ the tangent directions are generated by the skew-symmetric matrices. A basis for the skew-symmetric matrices is given by

$$\mathfrak{e}_1 = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{5.27}$$

$$\mathfrak{e}_2 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix} \tag{5.28}$$

$$\mathfrak{e}_3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix}. \tag{5.29}$$

Thus, in the case of $Q = I$ we can represent the rotational directions of the tangent

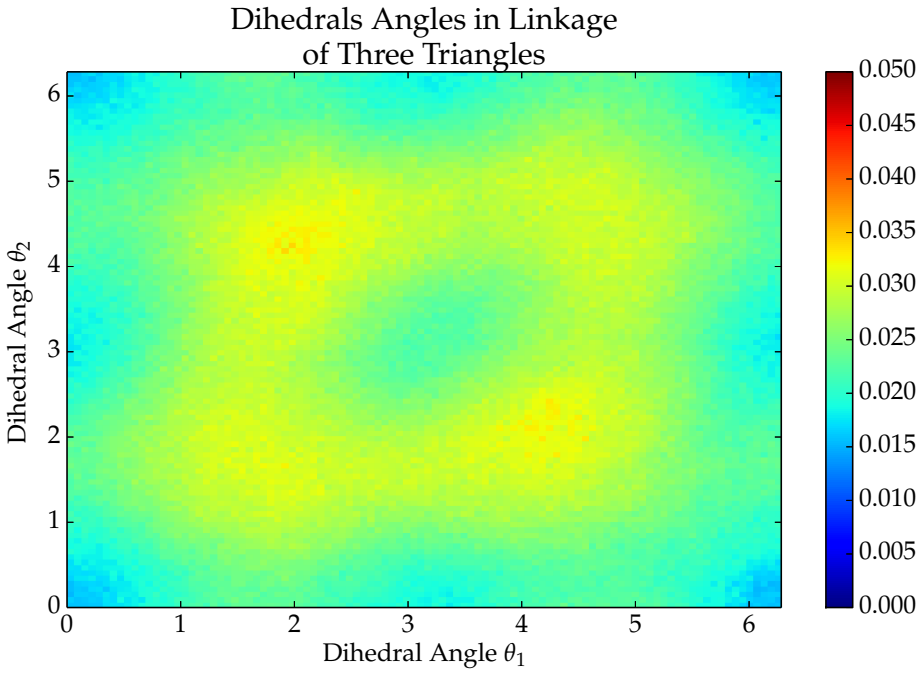

Figure 5.7: Histogram of the dihedral angle in sampled configurations of a three triangle linkage.

space as $\mathcal{T}_I SO(3) = \text{span}(w_1, w_2, w_3)$ where the bases w_1, w_2, w_3 are defined by

$$w_1 = (\mathfrak{e}_1 v^1, \mathfrak{e}_1 v^2, \ldots, \mathfrak{e}_1 v^N) \tag{5.30}$$

$$w_2 = (\mathfrak{e}_2 v^1, \mathfrak{e}_2 v^2, \ldots, \mathfrak{e}_2 v^N) \tag{5.31}$$

$$w_3 = (\mathfrak{e}_3 v^1, \mathfrak{e}_3 v^2, \ldots, \mathfrak{e}_3 v^N). \tag{5.32}$$

With this formulation, we can alter our method for finding the bases of the tangent space from theorem 8 to get the decomposition we desire. Specifically, setting

$$\tilde{A} = \left[C^T(z)(w_1, w_2, w_3)\tilde{B} \right]$$

where $B \in \mathbb{R}^{n \times (m-3)}$ is again random and taking the QR decomposition will give us

$$\tilde{A} = \left[Q^{(1)} Q^{(2)}_{rot} Q^{(2)}_{int} \right] R.$$

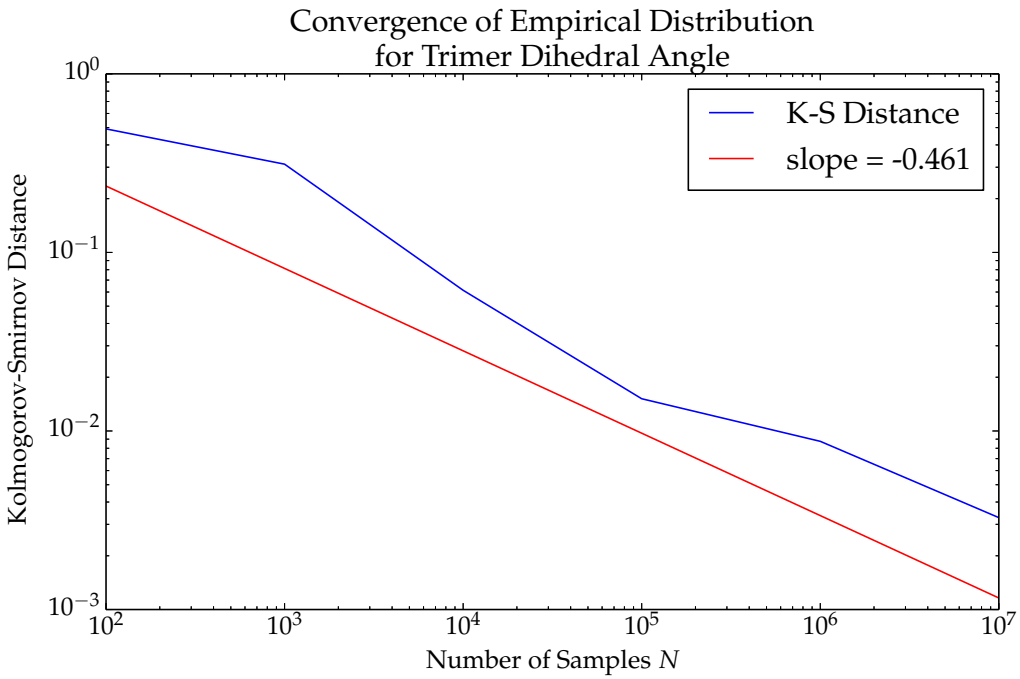

Figure 5.8: Convergence of the sampling scheme for the three-triangle linkage.

As before, $Q^{(2)} = \left[Q^{(2)}_{rot} Q^{(2)}_{int}\right]$ is a basis for $\mathcal{T}_z\mathcal{M}$, but we will also have $Q^{(2)}_{rot}$ as a basis for the rotational directions $\mathcal{T}_I SO(3)$ and $Q^{(2)}_{int}$ as a basis for the remaining internal tangent directions $\mathcal{T}_{\tilde{z}}\tilde{\mathcal{M}}$. Thus by modifying our random walk routine to sample from the directions of $Q^{(2)}_{int}$, we will fix the rotation of the proposed step in the tangent space.

Two issues must still be addressed. First, we have assumed that that the local rotation coordinate Q is the identity. Additionally, even though our step in tangent space does not contain any of the rotational directions, when it is projected back to the constraint space, some rotation may occur. We address these two issues simultaneously in the form of a correction step after the projection has been made. To do this, the configuration is simply rotated back to its original frame of reference using the Kabsch algorithm which finds the optimal rotation by which to match two sets of points in three dimensions [12]. The implicit assumption here is that if we have bad a sufficiently small random walk step, the resulting configuration will be

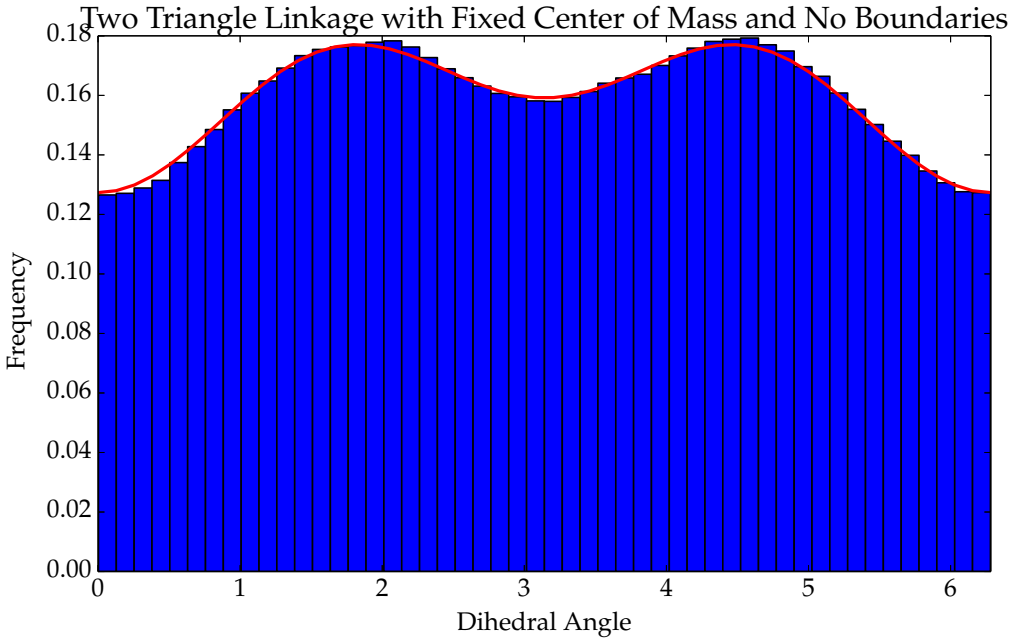

Figure 5.9: Sampling of two triangle linkage with fixed center of mass. The red curve has the form $\alpha_0 + \alpha_1 \cos(\theta) + \alpha_2 \cos(2\theta)$.

close to the one at the previous step. Moreover, if $Q = I$ at the previous step, this rotation will bring the configuration back to the same frame of reference.

Figure 5.11 shows the results for a sampling of configurations for the two triangle linkage. The distribution of dihedral angle takes the form of a cosine function which differs from the method which does not fix rotations. Similarly, the three triangle linkage also produces a different distribution on dihedral angles. As seen in figure 5.12, the distribution is not the simple product of one-dimensional distributions. There are local maxima at $(0, 0), (\frac{\pi}{3}, \frac{5\pi}{3})$, and $(\frac{5\pi}{3}, \frac{\pi}{3})$ With a local minima at (π, π).

Since the fixing of rotation alters our sampling distribution, this method is not optimal. An approach that may be considered in the future is to use the quotient metric for g as was similarly used in [8].

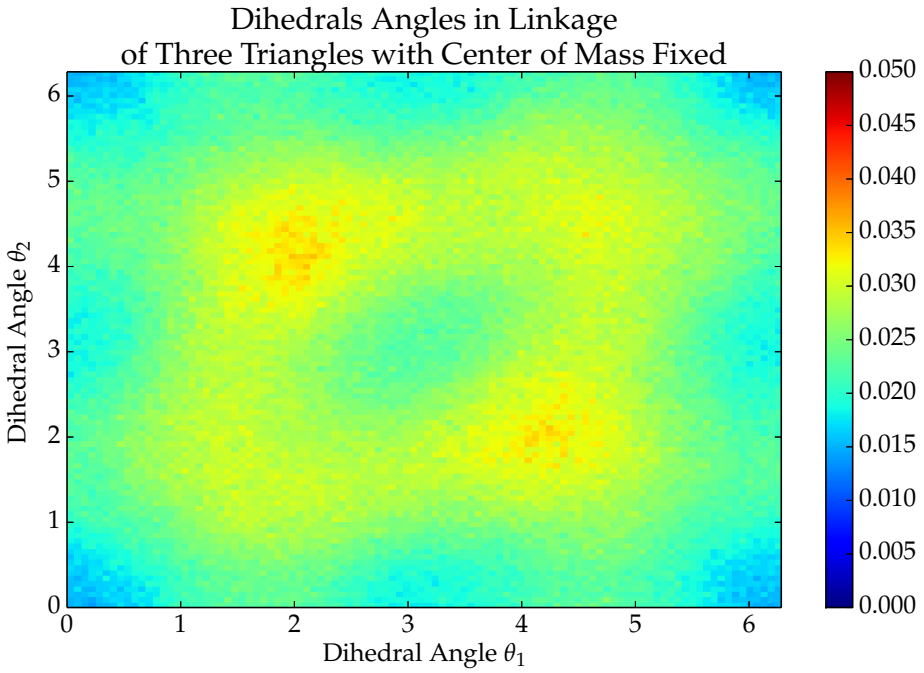

Figure 5.10: Sampling of three triangle linkage with fixed center of mass.

5.3 Reflected Brownian Motion on a Manifold

In our sampling of Building Game geometric configurations, we see that many of the sampled configurations of Brownian motion paths are self-intersecting. These configuration are nonphysical and can be avoided by using reflected Brownian motion.

5.3.1 Computational Scheme

To sample from a reflected Brownian motion on the manifold, we use a rejection scheme that first generates a proposal configuration according to the previous random walk scheme. Using our implicit boundary function that simply returns true if a proposed configuration is valid (not self-intersecting), and false otherwise, we accept or reject the proposed configuration. This process repeats until a proposal is accepted.

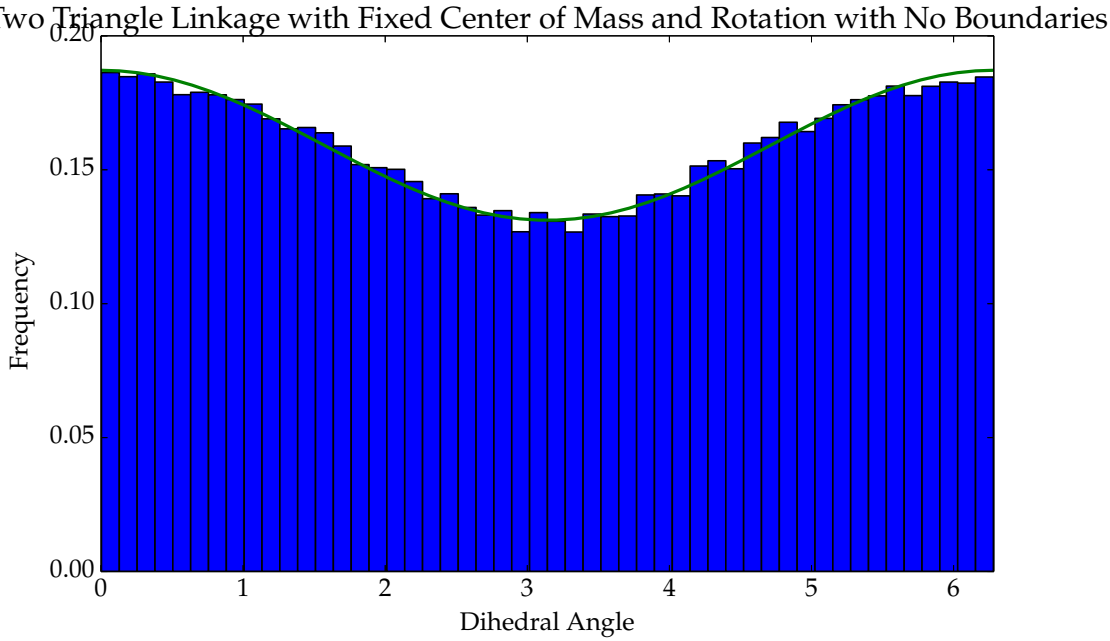

Figure 5.11: Sampling of two triangle linkage with fixed center of mass and rotations. The green curve has the form $\alpha_0 + \alpha_1 \cos(\theta)$.

Testing for Self-Intersection

In order to compute our implicit self-intersection function at each step, we must ensure that no two triangles intersect. Thus, for an intermediate $[x]$, each function call requires that $\frac{1}{2}|x|(|x|-1)$ pairs of triangle be tested for intersection. For the scheme to remain computationally feasible, it is imperative that these comparisons are carried out as efficiently as possible. We use an algorithm presented by Möller, which has been shown to be effective [20].

5.3.2 Validation and Test Cases

One way to test whether our rejection scheme has the properties that coincide with a reflected Brownian motion is to look at its distribution after a finite amount of time

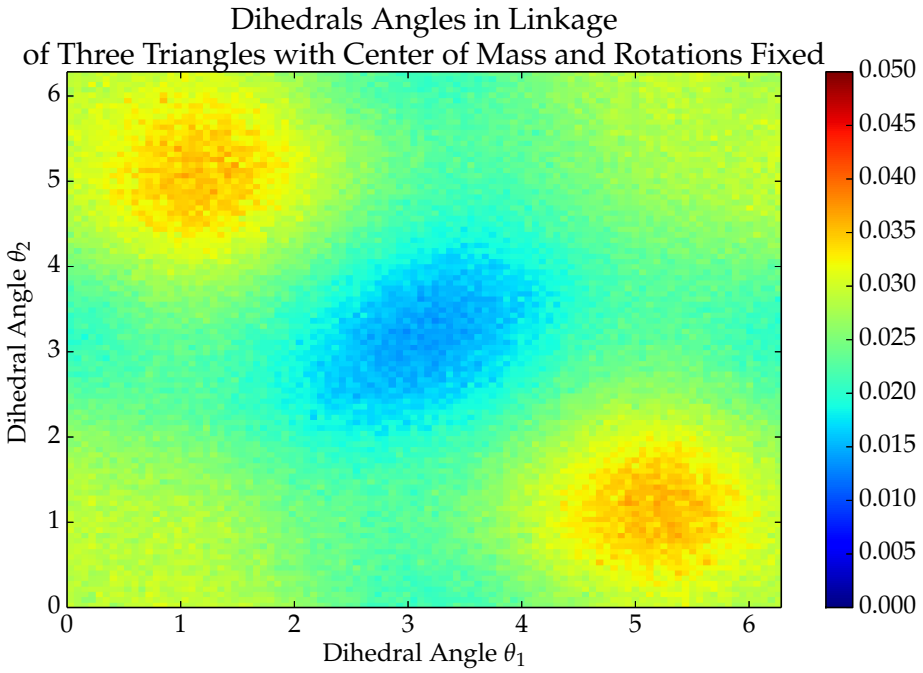

Figure 5.12: Sampling of three triangle linkage with fixed center of mass and rotations.

rather than only looking at long time statistics. In figure 5.13 we consider the case of a d-dimensional hyper-cube for $d = 2, 3, 4, 5$ and compare the distribution after a finite simulation time with the exact solution. The results match almost perfectly with the biggest errors occurring at the boundaries. This is to be expected since the rejection method biases the sampling slightly to points away from the boundary.

5.3.3 Reflected Brownian Motion on Geometric Configuration Spaces

Now, we again consider the two and three triangle linkage examples from section 5.2.3. Figure 5.14 depicts the same conditions (no fixed trivial degrees of freedom) as figure 5.5 previously, but with the addition of boundaries that prevent the two triangles from rotating through each other. The agreement in the two histograms is strong with the greatest differences being at the boundary as expected.

Similarly, we consider the three triangle linkage with no fixed degrees of freedom and impose boundaries. This test is less trivial than the two triangle linkage since the exterior triangles of the linkage may interact more easily. Comparing the original in figure 5.7, there is nice agreement except in the regions in the lower-right and upper-left corners of the plot which correspond to the aforementioned triangle intersection. Also, a noticeable layer of lower probability is seen around the periphery of the plot which reflects the boundary preventing the triangles from rotating through each other.

Again, we test the convergence for our scheme, using show the Kolmogorov-Smirnov distance of the empirical distribution on the first dihedral angle of the three triangle linkage with boundaries as a function of the number of samples. As seen in figure 5.16 the line of best fit has a slope of -0.449 which is again close to square root convergence.

Now we compare the analogous cases in which we fixed all of the trivial degrees of freedom with and without a boundary. Figure 5.17 is the boundaried version of figure 5.11 and figure 5.18 is the boundaried version of figure 5.12. Both cases exhibit a close comparison to their counterparts with the regions near the boundary possessing the largest deviations.

5.4 Computational Implementation

As with our enumeration work, simulations were carried out on a desktop computer running 64-bit Ubuntu 14.04 LTS with 15.6 GiB memory and a quad-core 3.20GHz Intel processor. The module for simulating the Brownian motion on a manifold was

implemented in Python and was able to sample at a rate on the order of $3,000,000$ samples per hour. This rate varies with use of a boundary and the dimension of the manifold and ambient spaces as well.

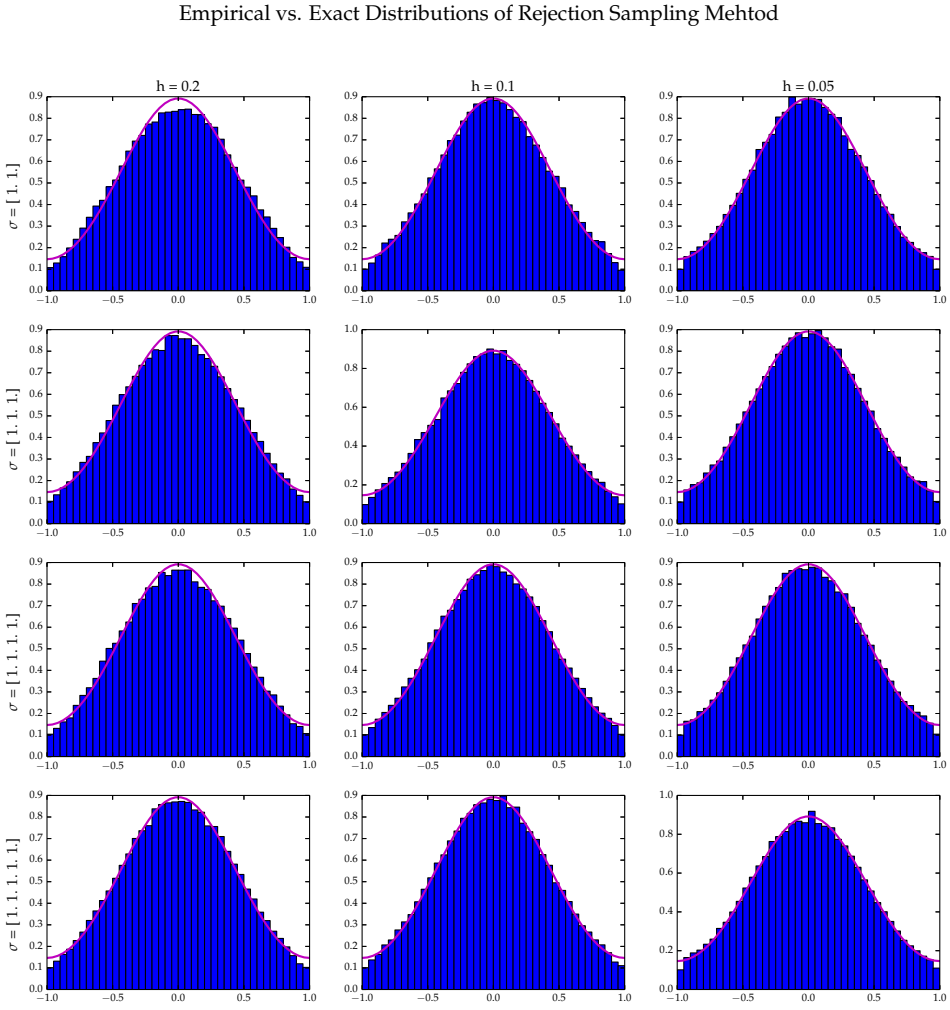

Figure 5.13: Finite time distribution of first dimension of rejection sampled points in hyper-cubes of increasing dimension across different choices of timestep.

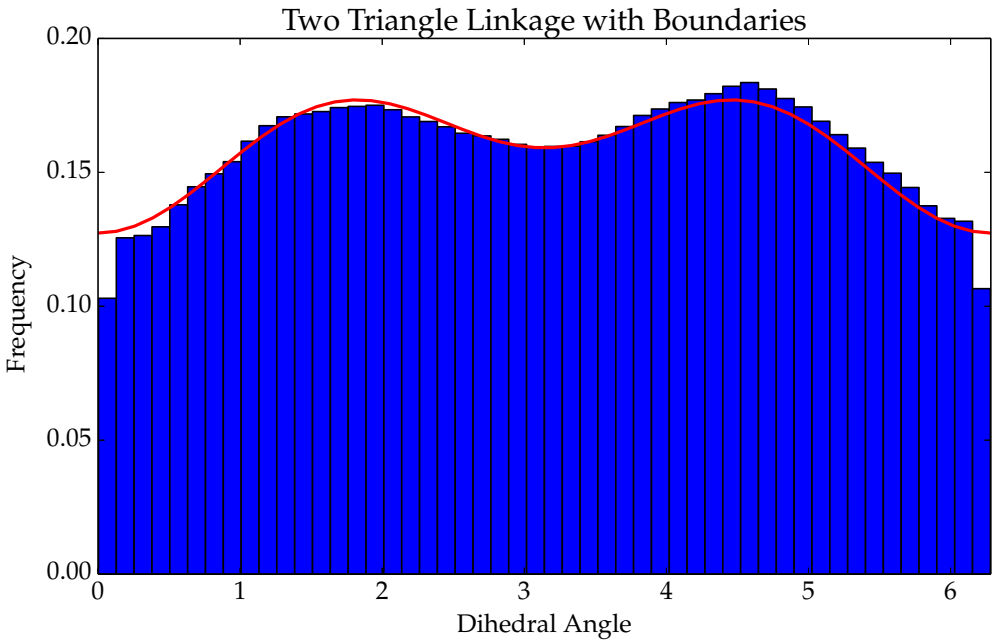

Figure 5.14: Sampling of two triangle linkage with boundaries. The red curve has the form $\alpha_0 + \alpha_1 \cos(\theta) + \alpha_2 \cos(2\theta)$.

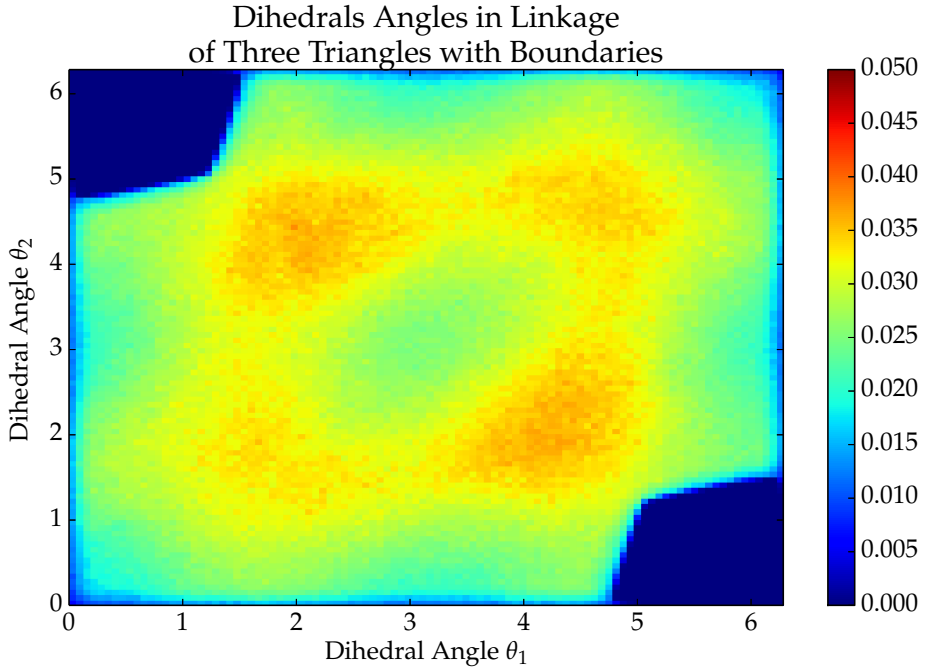

Figure 5.15: Sampling of three triangle linkage with boundaries.

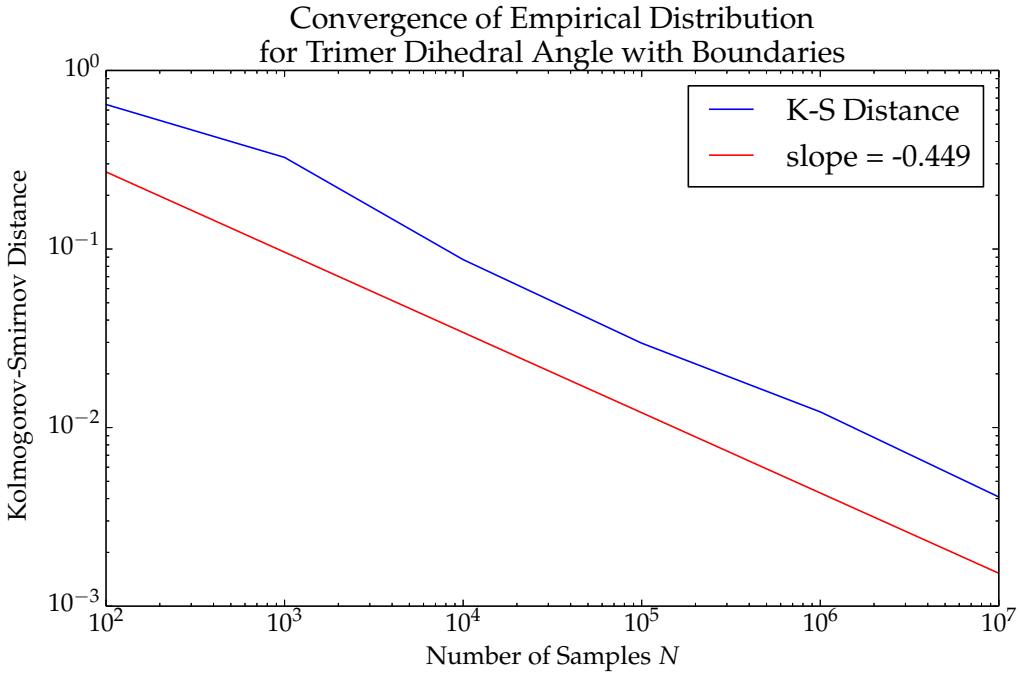

Figure 5.16: Convergence of the sampling scheme for the three-triangle linkage with self-intersection boundaries.

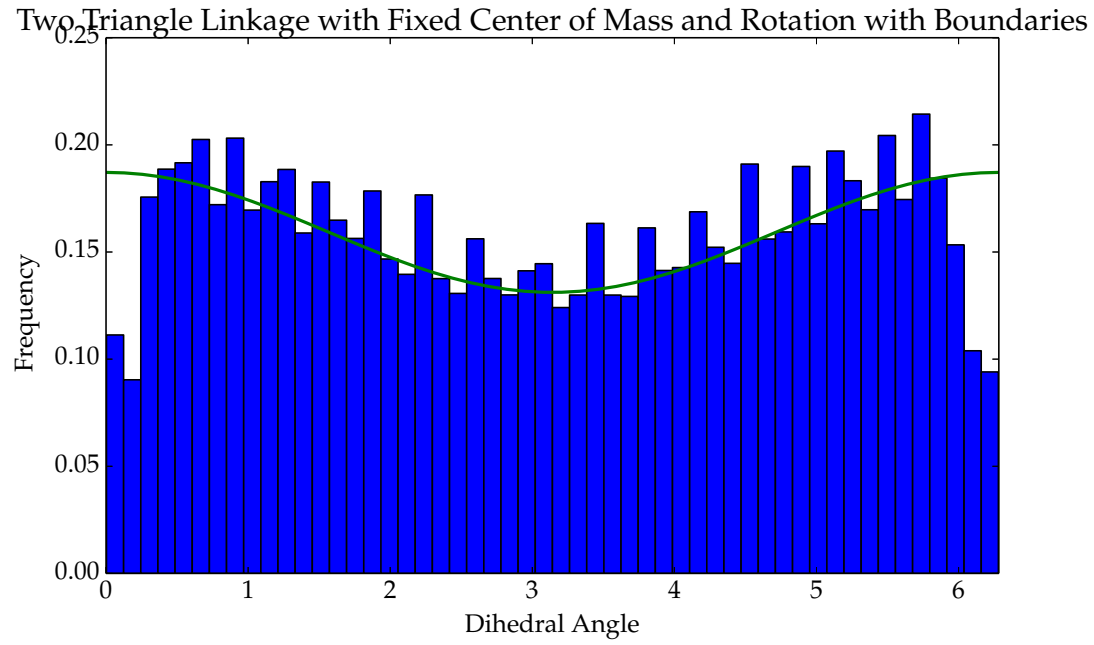

Figure 5.17: Sampling of two triangle linkage with boundaries and fixed center of mass and rotations. The green curve has the form $\alpha_0 + \alpha_1 \cos(\theta)$.

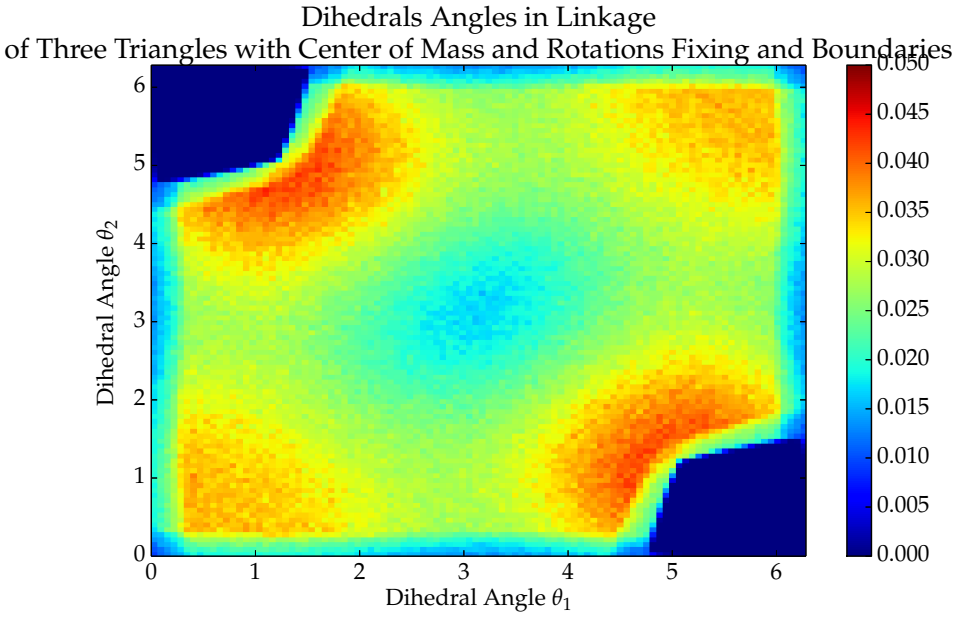

Figure 5.18: Sampling of three triangle linkage with boundaries and fixed center of mass and rotations.

Chapter Six

Results

Chapter 2 has provided a framework for modeling the self-assembly of a polyhedron using a Markov process. When using a transition rate matrix of the form

$$Q_{jk} = \begin{cases} S_{jk} e^{-\beta\left(E_{jk} - E_j\right)} & \text{if } [x^j] \leftrightarrow [x^k] \\ -z_j & \text{if } j = k \\ 0 & \text{else} \end{cases} \tag{6.1}$$

the intermediate energies E_j and transition barrier energies E_{jk} must be specified. Using minus the number of closed edges in an intermediate has a nice, physically motivated interpretation. The choice of barrier heights, however, has not yet been addressed. Using information of an intermediate's geometric configuration space that can be gathered using our manifold reflected Brownian motion scheme, we derive energy barrier heights and thus the rate matrix Q. This combination of combinatorial and geometric structure provides a unique insight into the various themes of self-assembly.

6.1 Deriving Rates

Consider the process of adding a face to an intermediate. In the combinatorial configurations space this is a transition from one node of the graph to another. However, for this transition to occur in a more physical setting, the geometry of the intermediate plays a large role. One can imagine that there are geometric configurations are contorted in a way as to prevent a face from being added at a particular location. For instance consider the linkage of two triangles as an intermediate of the tetrahedron. Since the combinatorial configuration space is rather trivial, adding a face will results in the rigid linkage of three triangles all sharing a vertex. Since

the equilateral triangle has angles of $\frac{\pi}{3}$, the angle between the two edges of the two triangle linkage that the third face attaches to must be close to $\frac{\pi}{3}$ for it to attach easily. We extend and formalize this idea for face attachment on a Building Game intermediate to obtain the rates we desire.

If there are three vertices v_a, v_b, v_c in the intermediate that each combine with one of the three vertices on the added triangle, and (v_a, v_b) and (v_b, v_c) are edges in one of the intermediate's existing triangles, the rate of attachment for the face is given by the proportion of time the configuration–undergoing manifold reflected Brownian motion–has $\frac{\pi}{3} - \epsilon < \angle(v_a, v_b, v_c) < \frac{\pi}{3} + \epsilon$. Here ϵ is a parameter reflecting how close to the ideal angle the configuration must be in order to allow attachment. If there are no such triplets v_a, v_b, v_c where a face is being attached, the attachment occurs at a unit rate. If there are more than one triplet v_a, v_b, v_c, the attachment rate is averaged across each such triplet. Taking this empirical rate and summing over the other faces corresponding to the degeneracies of the transition of interest, the empirical forward transitions rates $\hat{Q}_{jk}$ are derived. By comparing these computed rates with the analytic form for our transition rate matrix, we can derive the barrier heights E_{jk} for each transitions. Since the barriers are symmetric, this information also gives the reverse transition rates Q_{kj}.

$$\hat{Q}_{jk} = Q^0_{jk} \tag{6.2}$$

$$= S_{jk} e^{\beta_0 (E_{jk} - E_j)} \tag{6.3}$$

$$E_{jk} = E_j - \frac{1}{\beta_0} \log \left(\hat{Q}_{jk} / S_{jk} \right) \tag{6.4}$$

Here β_0 is the inverse temperature used to derive the barriers from the empirical computations. Since

$$Q_{jk} = S_{jk} e^{\beta (E_{jk} - E_j)} \tag{6.5}$$

$$= S_{jk} e^{-\beta\left(E_j - \frac{1}{\beta_0}\log\left(\hat{Q}_{jk}/S_{jk}\right) - E_j\right)} \tag{6.6}$$

$$= S_{jk}\left(\hat{Q}_{jk}/S_{jk}\right)^{\frac{\beta}{\beta_0}}, \tag{6.7}$$

the choice of β_0 effectively specifies the units of β. Typically, we choose $\beta_0 = 1$.

To derive the empirical rates, each intermediate was simulated for $10,000,000$ steps with a stepsize of $h = 0.05$. By storing each of the relevant angles either step by step, or in a histogram, the rates can be computed for each choice of ϵ at a later time as needed.

6.2 Self-Assembly Statistics

After we have computed the rate matrix Q for the octahedron as a function of ϵ and β, we are now able to compute the different statistics derived in section 2.3. For instance, the stationary distribution of each octahedron intermediate is plotted as a function of β in figure 6.1. To better view how these probabilities decay as the temperature decreases, figure 6.2 presents the natural logarithm of these stationary probabilities.

With the rate matrix, we can also look at how the occupation probabilities change over time. If the process begins with a single face, the probabilities diffuse through the graph according to $e^{Qt}e_1$ and eventually converge of the stationary distribution π as $t \to \infty$. Figure 6.3 shows how these occupation probabilities change over time.

Another way to visualize the dynamic behaviors of our Markov process is to look at individual transitions and the rates with which they occur. Figure 6.4 shows the

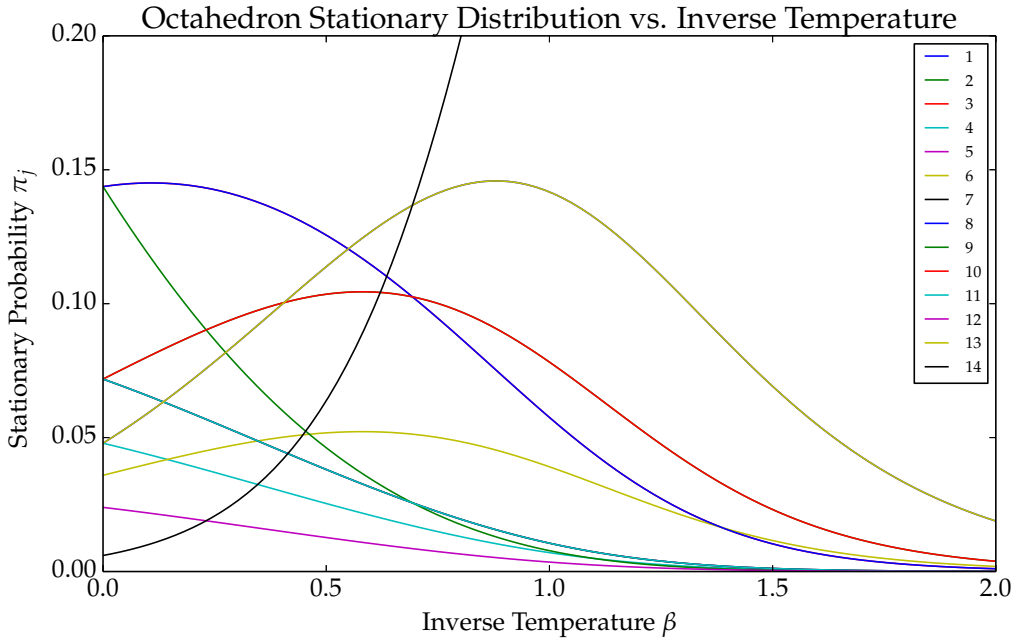

Figure 6.1: Octahedron stationary distributions as a function of β.

octahedron combinatorial configuration space with the edges labeled with the joint transition rate $\pi_j Q_{jk} = \pi_k Q_{kj}$ for the choices $\beta = 0.8$ and $\epsilon = 0.5$.

Since our model uses the two parameters ϵ and β, it is of particular interest to examine how the choice of these parameters affects the behaviors of the Markov process. Figure 6.5 shows how the stationary distributions (node size) and the transition rates (edge thickness) change for three choices of ϵ and three choices of β. These plots provide and effective way of visualizing which edges and pathways are more probable for different parameter choices.

One of the most interesting Markov process statistics is the expected formation time $E[\tau]$. Figure 6.6 shows the relation between β and the expected formation time for several choices of ϵ. At $\beta = 0$, the Markov process is essentially a random walk on the graph, weighted only by the degeneracy of each transition and unaffected by the choice of ϵ. On the other end, as β gets higher and higher, each transition

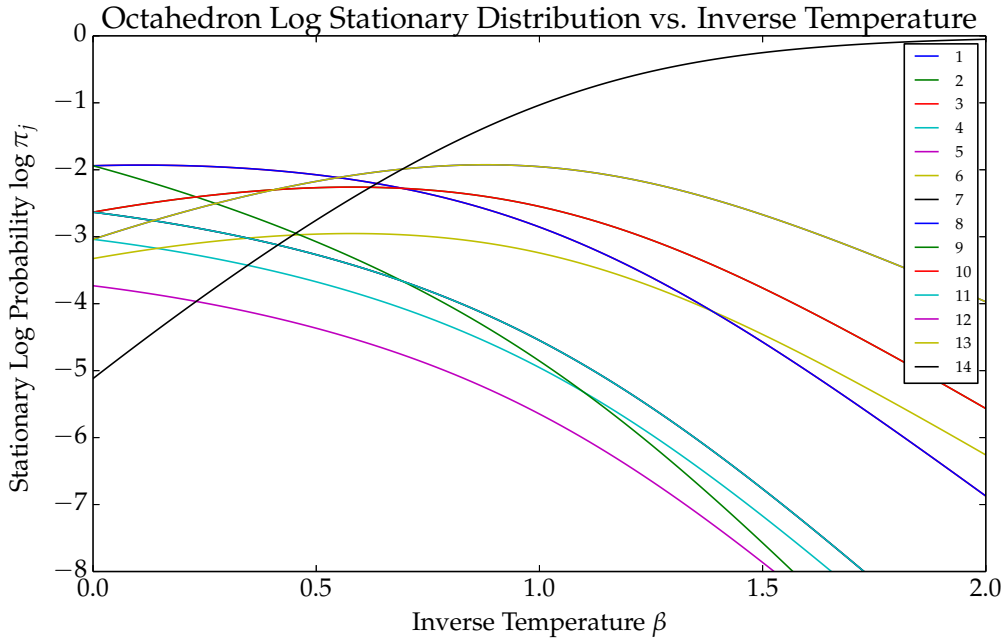

Figure 6.2: Natural log of the octahedron stationary distributions as a function of β.

occurs with an increasingly small rate. This leads to a sharp increase in formations times. Interestingly, there is an intermediary range of *beta*s where a minimum of the formation time function occurs.

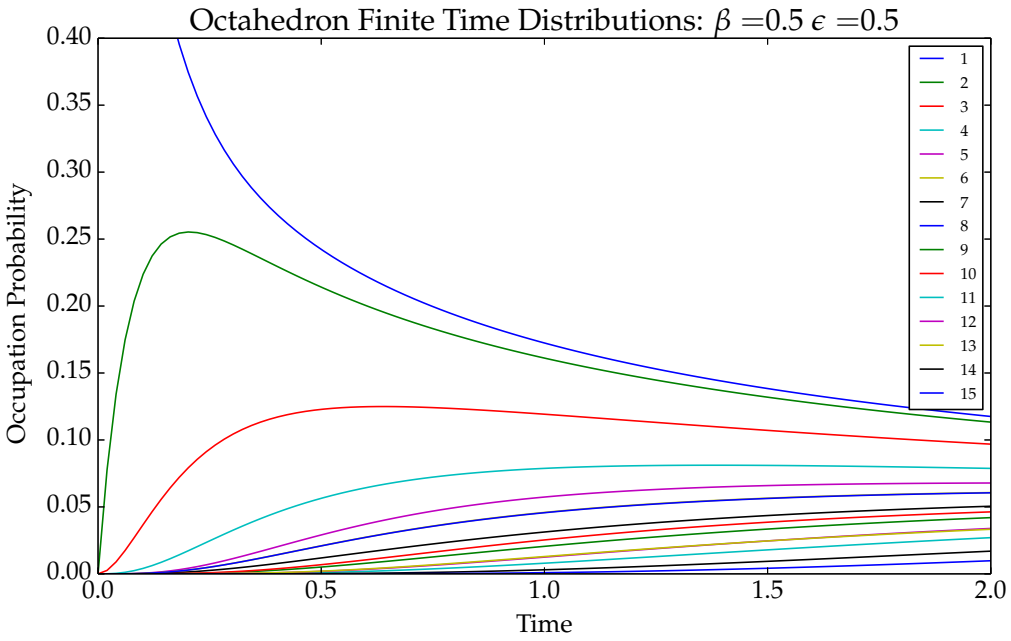

Figure 6.3: Occupation probabilities for Octahedron intermediates as a function of time.

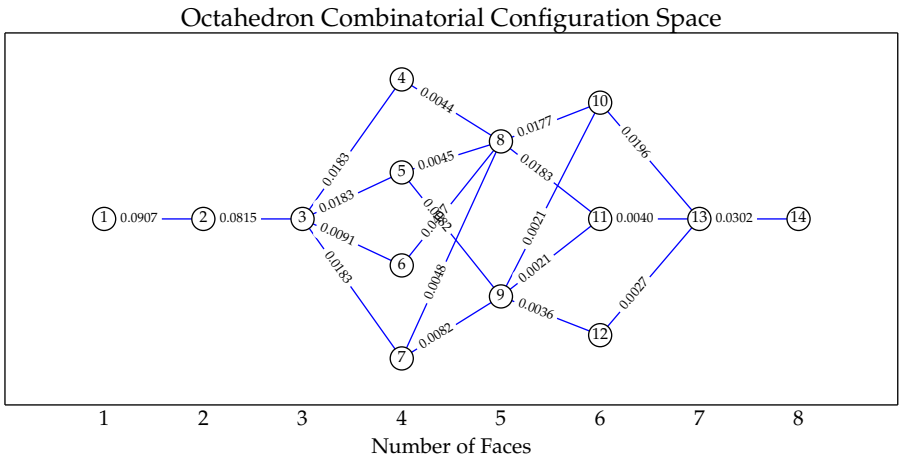

Figure 6.4: Combinatorial Configuration space for the Octahedron.

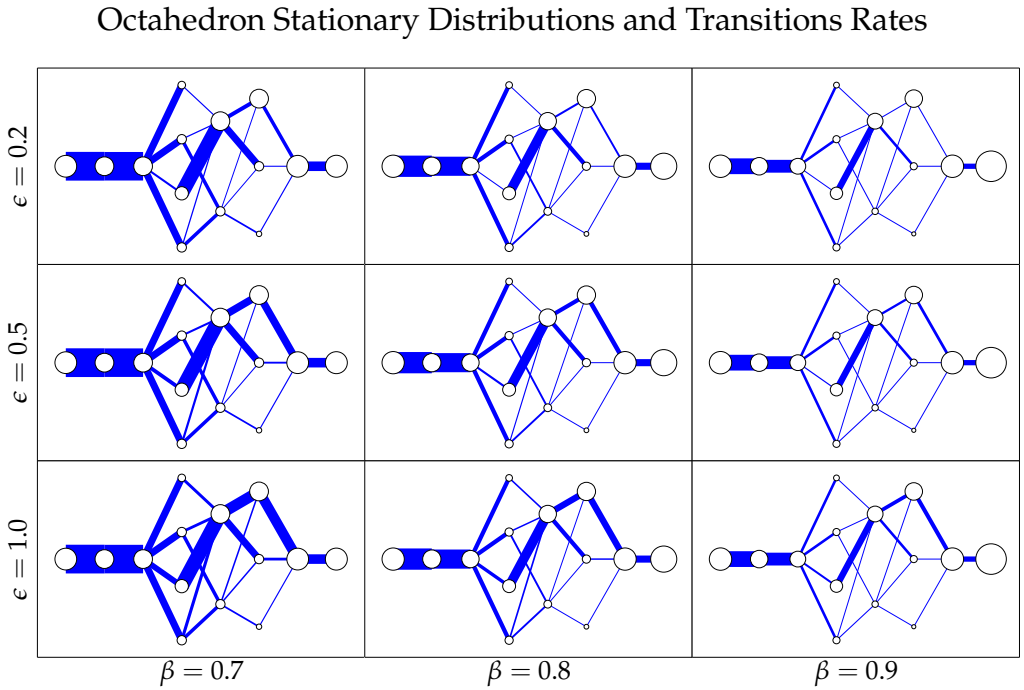

Figure 6.5: The Affect of β and ϵ on the transitions rates and stationary distribution of the Octahedron combinatorial configuration space.

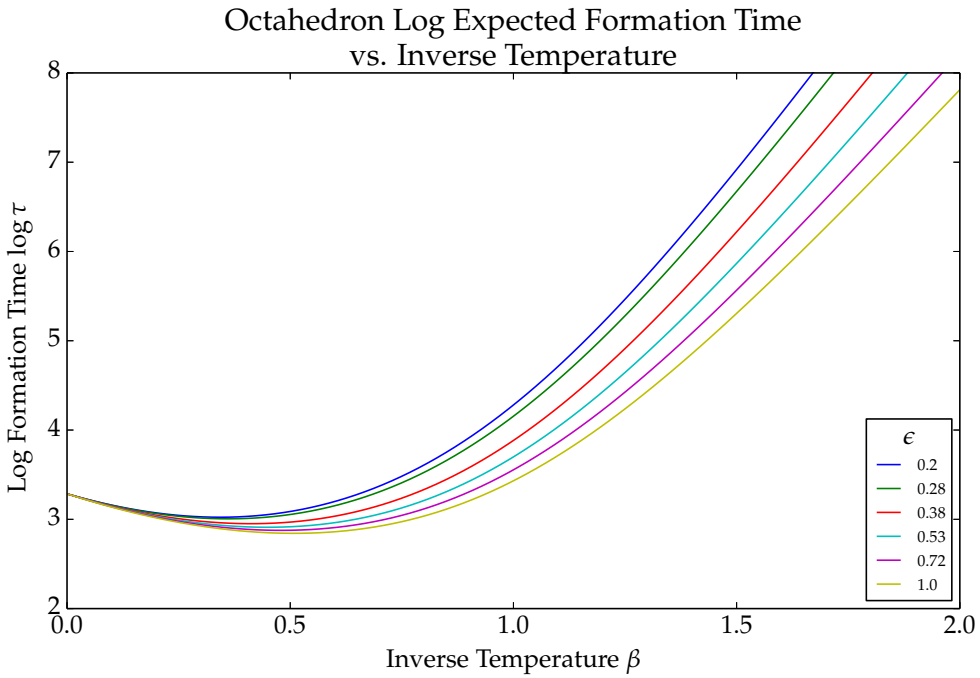

Figure 6.6: Expected formation times for the octahedron as a function of ϵ and β.

Bibliography

[1] B. Berger, P. Shor, L. Tucker-Kellogg, and J. King. Local rule-based theory of virus shell assembly. *Proceedings of the National Academy of Sciences*, 91:7732–7736, 1994.

[2] D. Caspar and A. Klug. Physical principles in the construction of regular viruses. *Cold Spring Harbor Symposium in Qantitative Biology*, 27:1–27, 1962.

[3] P. Cromwell. *Polyhedra*. Cambridge University Press, Cambridge, UK, 1997.

[4] J. Doye, M. Miller, and D. Wales. Evolution of the potential energy surface with size for lennard-jones clusters. *Journal of Chemical Physics*, 111:8417–8428, 1999.

[5] L. Vita E. Pennestri, M. Cavacece. On the computation of degrees-of-freedom: a didactic perspective. *Proceeddings of the IDETC*, 2005.

[6] D. Endres, M. Miyahara, P. Moisant, and A. Zlotnick. A reaction landscape identifies the intermediates critical for self-assembly of virus capsids and other polyhedral structures. *Protein Science*, 14:1518–1525, 2005.

[7] S. O'Brien R. Curl R. Smalley H. Kroto, J. Heath. c_{60}: Buckminsterfullerene. *Nature*, 318:162–163, 1985.

[8] M. Holmes-Cerfon, S. Gortler, and M. Brenner. A geometrical approach to computing free-energy landscapes from short-ranged potentials. *PNAS*, 110:E5 – E14, 2013.

[9] P. Hsu. Brownian motion and riemannian geometry. *Contemporary Mathematics*, 73:95–104, 1988.

[10] OEIS Foundation Inc. The On-Line Encyclopedia of Integer Sequences. http://oeis.org/A030138, 1999.

[11] F. Johansson et al. *mpmath: a Python library for arbitrary-precision floating-point arithmetic (version 0.18)*, December 2013. `http://mpmath.org/`.

[12] W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A*, 32(5):922–923, Sep 1976.

[13] T. Keef, C. Micheletti, and R. Twarock. Master equation approach to the assembly of viral capsids. *Journal of Theoretical Biology*, 242(3):713 – 721, 2006.

[14] D. Klarner. A procedure for improving the upper bound for the number of n-ominoes. *Canadian Journal of Mathematics*, 25:585–602, 1973.

[15] D. Li, W. Zhou, K. Landskron, S. Sato, C. Kiely, M. Fujita, and T. Liu. Viral-capsid-type vesicle-like structures assembled from m12l24 metalorganic hybrid nanocages. *Angewandte Chemie International Edition*, 50(22):5182–5187, 2011.

[16] K. Lindorff-Larsen, S. Piana, R. Dror, and D. Shaw. How fast-folding proteins fold. *Science*, 334(6055):517–520, 2011.

[17] Y. Liu, C. Hu, A. Comotti, and M. Ward. Supramolecular archimedean cages assembled with 72 hydrogen bonds. *Science*, 333:436–440, 2011.

[18] J. McCarthy. *Introduction to Theoretical Kinematics*. The MIT Press, Cambridge, Massachusetts, 1990.

[19] F. Mezzadri. How to generate random matrices from the classical compact groups. notices to the ams, 2007.

[20] T. Möller. A fast triangle-triangle intersection test. *Journal of Graphics Tools*, 2:25–30, 1997.

[21] J. Norris. *Markov Chains*. Cambridge University Press, New York, 1998.

[22] S. Pandey, M. Ewing, A. Kunas, S. Ngyen, D. Gracias, and G. Menon. Algorithmic design of self-folding polyhedra. *Proceedings of the National Academy of Sciences*, 108:19885–19890, 2011.

[23] S. Pandey, D. Johnson, R. Kaplan, J. Klobusicky, G. Menon, and D. Gracias. Self-assembly of mesoscale isomers: The role of pathways and degrees of freedom. *PLoS ONE*, 9(10), 10 2014.

[24] S. Pandey D. Gracias G. Menon R. Kaplan, J. Klobusicky. Building polyhedra by self-folding: theory and experiment. *Artificial Life*, 20:409–439, 2014.

[25] J. Rotman. *An Introduction to the Theory of Groups*. Springer-Verlag, New York, NY, fourth edition, 1995.

[26] H. Sachse. Ueber die geometrischen isomerien der hexamethylenderivate. *Berichte der deutschen chemischen Gesellschaft*, 23(1):1363–1370, 1890.

[27] R. Schwartz, P. Shor, P. Prevelige Jr., and B. Berger. Local rule simulation of the kinetics of virus capsd self-assembly. *Biophysical Journal*, 75:2626–2636, 1998.

[28] D. Sommerville. *Introduction to the Geometry of N Dimensions*. Mesthuen & Co. Ltd., London, 1929.

[29] Q. Sun, J. Iwasa, D. Ogawa, Y. Ishido, S. Sato, T. Ozeki, Y. Sei, K. Yamaguchi, and M. Fujita. Self-assembled M24L48 polyhedra and their sharp structural switch upon subtle ligand variation. *Science*, 328:1144–1147, 2010.

[30] D. Wales. Closed-shell structures and the building game. *Chemical Physics Letters*, 141:478–484, 1987.

[31] D. Wales, M. Miller, and Y. Walsh. Archetypical energy landscapes. *Nature*, 394:758–760, 1998.

[32] G. Ziegler. *Lectures on Polytopes*. Springer-Verlag, New York, NY, 1995.

[33] A. Zlotnick. An equilibrium model of the self assembly of polyhedral protein complexes. *Journal of Molecular Biology*, 241:59–67, 1994.