

UNLINKING DNA: XERC/XERD SITE-SPECIFIC RECOMBINATION AND  
TETRAHEDRAL TANGLES

A thesis presented to the faculty of  
San Francisco State University  
In partial fulfilment of  
The Requirements for  
The Degree

Master of Arts  
In  
Mathematics

by

Crystal Ann Moreno  
San Francisco, California

May 2015

Copyright by  
Crystal Ann Moreno  
2015

## CERTIFICATION OF APPROVAL

I certify that I have read *Unlinking DNA: XerC/XerD Site-Specific Recombination and Tetrahedral Tangles* by Crystal Ann Moreno and that in my opinion this work meets the criteria for approving a thesis submitted in partial fulfillment of the requirements for the degree: Master of Arts in Mathematics at San Francisco State University.

---

Federico Ardila  
Professor of Mathematics

---

Mariel Vazquez  
Professor of Mathematics

UNLINKING DNA: XERC/XERD SITE-SPECIFIC RECOMBINATION AND  
TETRAHEDRAL TANGLES

Crystal Ann Moreno  
San Francisco State University  
2015

Knotted and linked DNA cause complications during DNA replication and transcription, and therefore simplifying these topological forms is essential. In *Escherichia coli*, the XerCD-FtsK complex has been found to effectively unlink DNA links produced by replication of the circular chromosome. *TangleSolve*, a computer implementation of the tangle method of Ernst and Sumners, can be used to compute possible enzymatic mechanisms. In 2005, Vazquez and colleagues showed that the three mechanisms proposed for the action of XerCD on a circular unknotted DNA molecule could be interpreted as different projections of a 3D object. In 2012, Wono extended this idea by entrapping tangles inside a regular tetrahedron. Here we expand *TangleSolve*, and develop a computer visualization tool called *Tangle3D*, that automates and extends Wono's work. Using *Tangle3D* we define equivalence classes for the projections of tetrahedral tangles and explore an application to DNA unlinking by Xer recombination.

I certify that the Abstract is a correct representation of the content of this thesis.

---

Chair, Thesis Committee

Date

## ACKNOWLEDGMENTS

I wish to express to extend my gratitude to my thesis committee: Mariel for inspiring me and teaching me knot theory and tangle calculus, Javier for teaching me how to conduct research in mathematical biology, and Federico for introducing me to combinatorics and its applications. I thank my family and friends for their love and support. I also thank (CM)2 directors: Matt for his encouragements and Brandy for her advices. I thank all of the faculty members in the mathematics department at SFSU. Lastly, the completion of this thesis would not be possible without my advisers' support during the last year while I was recovering my health.

This research was supported by the SFSU (CM)2 program, NSF 0841164, NSF Grant DMS 1057284, and NIH MBRS-RISE: R25-GM059298.

## TABLE OF CONTENTS

|       |   |    |
|-------|---|----|
| 1     | Introduction . . . . .  | 1  |
| 2     | Biological Background . . . . .   | 4  |
| 2.1   | Replication of the bacterial chromosome . . . . .   | 4  |
| 2.1.1 | Enzymes that change the topology of DNA: topoisomerases<br>and site-specific recombinases . . . . . | 7  |
| 2.1.2 | XerC and XerD Site-Specific Recombinases . . . . .  | 11 |
| 3     | Mathematical Background . . . . .   | 13 |
| 3.1   | Knots . . . . .   | 13 |
| 3.1.1 | Set Theory . . . . .  | 13 |
| 3.1.2 | Basic Definitions in Topology and Knot Theory . . . . .   | 16 |
| 3.1.3 | Projection of a Knot . . . . .  | 23 |
| 3.1.4 | Knot Diagrams . . . . .   | 25 |
| 3.1.5 | Reidemeister Moves . . . . .  | 26 |
| 3.2   | Links . . . . .   | 27 |
| 3.3   | Tangles . . . . .   | 29 |
| 3.3.1 | Tangle Definition . . . . .   | 29 |
| 3.3.2 | Projection of a Tangle . . . . .  | 31 |
| 3.3.3 | Types of tangles . . . . .  | 32 |
| 3.3.4 | Rational Tangles . . . . .  | 33 |

|       |   |     |
|-------|---|-----|
| 3.3.5 | Tangle Calculus . . . . .   | 37  |
| 4     | The Tangle Method for Site-Specific Recombination . . . . .                   | 42  |
| 5     | Tetrahedral Tangles: Embedding a Tangle Inside the Tetrahedron . . . . .      | 45  |
| 5.1   | Groups . . . . .  | 45  |
| 5.2   | Tetrahedral Tangles and Computer Implementation . . . . .                     | 57  |
| 5.3   | Tetrahedral Tangle Projections . . . . .                                      | 68  |
| 5.4   | Equivalence Classes for Tetrahedral Tangle Projections . . . . .              | 72  |
| 6     | Results: Application of Tetrahedral Tangles to XerCD-FtsK Unlinking . . . . . | 76  |
| 6.1   | 3D Analysis of XerCD-FtsK Unlinking Solutions . . . . .                       | 76  |
| 6.1.1 | Unknot $\rightarrow$ Unlink . . . . .   | 78  |
| 6.1.2 | Hopf Link $\rightarrow$ Unknot . . . . .                                      | 80  |
| 6.1.3 | RH Trefoil Knot $\rightarrow$ Hopf Link . . . . .                             | 84  |
| 7     | Conclusions and Future Directions . . . . .                                   | 87  |
|       | Appendix A: Elementary Number Theory . . . . .                                | 89  |
|       | Appendix B: <i>TangleSolve</i> and <i>Tangle3D</i> . . . . .                  | 94  |
|       | Appendix C: Tetrahedral Tangle Projections . . . . .                          | 99  |
|       | Appendix D: Algorithms . . . . .  | 103 |

Bibliography . . . . . 108

## LIST OF TABLES

| Table |  | Page |
|-------|--|------|
| 6.1   | The possible solutions of the XerCD site-specific reactions on the unknot substrate at <i>dif</i> generated by <i>TangleSolve</i> . . . . .              | 79   |
| 6.2   | Equivalence classes for projections of tetrahedral tangle $\Delta_{(0)}^{\varphi_1}$ . . . . .   | 80   |
| 6.3   | The possible solutions of the XerCD site-specific reactions on the Hopf link substrate at <i>dif</i> generated by <i>TangleSolve</i> . . . . .           | 81   |
| 6.4   | Equivalence classes for projections of tetrahedral tangle $\Delta_{(2)}^{\varphi_i}$ . . . . .   | 82   |
| 6.5   | Equivalence classes for projections of tetrahedral tangle $\Delta_{(-2)}^{\varphi_i}$ . . . . .  | 83   |
| 6.6   | The possible solutions of the XerCD site-specific reactions on the right-handed trefoil knot substrate at <i>dif</i> generated by <i>TangleSolve</i> . . | 85   |
| 6.7   | Equivalence classes for projections of tetrahedral tangle $\Delta_{(3)}^{\varphi_1}$ . . . . .   | 86   |

## LIST OF FIGURES

| Figure   | Page |
|--|------|
| 2.1 A typical rod shaped bacterium: the chromosome and plasmids make up the organism's genetic material or genome. . . . .   | 4    |
| 2.2 Example of normal bacterial cell division: (a) a double-stranded circular parental chromosome is indicated as a circle inside the cytoplasm of a bacterium (b) replication results in two copies of the parental chromosome that are interlinked (replication links) (c) unlinking of replication links through strand passage (d) two daughter cells that each inherit one copy of the parental DNA. . . . .  | 5    |
| 2.3 Example of improper bacterial cell division (a) a double-stranded circular parental chromosome is indicated as a circle in the cytoplasm of a bacterial cell (b) replication results in two copies of the parental chromosome that are interlinked (replication links) (c) enzymes fail to properly segregate (unlink) the two DNA molecules (d) one daughter cell inherits two copies of the DNA and the second daughter cell does not inherit any DNA. . . . . | 6    |
| 2.4 The synaptosome is the part of the synaptic complex that consists of the recombinase (enzyme) bound to two segments of double stranded DNA (dsDNA), the recombination sites. Notice that the two DNA segments could be intertwined in non-trivial ways. . . . .  | 8    |

|     |  |    |
|-----|--|----|
| 2.5 | Site-specific recombination on circular DNA. First, the enzymes recognize the recombination sites where they are brought together to form the synaptic complex. Next, the site-specific recombinases cleave the DNA, perform strand exchange, and reseal the breaks. . . . .   | 9  |
| 2.6 | Site-specific recombination for recombination sites arranged in anti-parallel on two DNA molecules resulting in integration. . . . .   | 10 |
| 2.7 | Site-specific recombination for recombination sites arranged in parallel resulting in excision. . . . .  | 10 |
| 2.8 | Tyrosine site-specific recombinases XerC and XerD (XerCD) attached to recombination site <i>dif</i> on the <i>E. coli</i> chromosome, which is a 28 base pair (bp) long dsDNA segment. . . . .   | 11 |
| 3.1 | An example of sets $A$ and $B$ where $A \subset B$ . We say that $A$ is a subset of $B$ , or $A$ is contained in $B$ . . . . .   | 14 |
| 3.2 | The shaded region represents the union of sets $A$ and $B$ , denoted by $A \cup B$ . . . . .   | 14 |
| 3.3 | The shaded region represents the intersection of sets $A$ and $B$ , denoted by $A \cap B$ . . . . .  | 15 |
| 3.4 | Examples of topologies on the set $X = \{u, v, w\}$ : (a) $\mathcal{T}_1 = \{X, \emptyset\}$ and (b) $\mathcal{T}_2 = \{\emptyset, \{u\}, \{v\}, \{u, v\}, X\}$ . $\mathcal{T}_1$ and $\mathcal{T}_2$ satisfy all three axioms <b>O1</b> , <b>O2</b> , and <b>O3</b> . . . . . | 17 |

|      |   |    |
|------|---|----|
| 3.5  | Examples of collections that are not topologies on the set $X = \{u, v, w\}$ . (a) $\mathcal{T}_3$ is not a topology on the set $X$ because it fails to satisfy axiom <b>O2</b> . The union of sets $\{v\}$ and $\{w\}$ is not an element of $\mathcal{T}_3$ . (b) $\mathcal{T}_4$ is not a topology on the set $X$ because it violates axiom <b>O3</b> . The intersection of the sets $\{u, v\}$ and $\{v, w\}$ is not an element of $\mathcal{T}_4$ . . . . . | 18 |
| 3.6  | A topological space $X$ is Hausdorff if each pair of distinct points $x, y \in X$ belong to disjoint open sets $U$ and $V$ such that $x \in U$ and $y \in V$ . . . . .  | 19 |
| 3.7  | Example of two embeddings $f_0$ and $f_1$ that map $X$ , which consists of a single point $\{p\}$ , to the line segment $Y = [a, b]$ . In this case $f_0(p) = c_1$ and $f_1(p) = c_2$ , and each $f(p, t)$ is a point in $Y$ between $a$ and $b$ . . . . .  | 20 |
| 3.8  | Example of a level preserving isotopy $F : X \times I \hookrightarrow Y \times I$ that connects embeddings $f_0, f_1 : X \hookrightarrow Y$ , where $X = \{p\}$ , $Y = [a, b]$ , and $I = [0, 1]$ . Here $F$ is defined as $F(f(x, t), t)$ where $f(p, 0) = f_0$ and $f(p, 1) = f_1$ and $f(x, t)$ is continuous for all $t$ . . . . .  | 20 |
| 3.9  | $K$ is a knot if there is an embedding $H$ that takes the unit circle $C = \{(x, y) \in \mathbb{R}^2 : x^2 + y^2 = 1\}$ to a closed curve in $\mathbb{R}^3$ , where $H(C) = K$ . . . . .  | 22 |
| 3.10 | Examples of knots: (a) unknot or trivial knot; and (b) trefoil. . . . .   | 22 |

|      |   |    |
|------|---|----|
| 3.11 | Planar projection of a five crossing torus knot $5_1$ onto its mirror image $5_1^*$ . We refer the reader to [5] for a biologically relevant knot table. . .  | 24 |
| 3.12 | The planar projection of these two arcs results in a double point $p_i$ , where points $p_{i_1}$ and $p_{i_2}$ are mapped to $p_i$ in the projection $\mathcal{P}(K)$ . . .   | 24 |
| 3.13 | Intersections of a knot projection: (a) double point intersection; (b) multiple point intersection; (c) vertex double point intersection; (d) a second form of a vertex double point intersection. Only the double point intersection (a) is allowed in a regular knot projection. (Redrawn from [22].) . . . . . | 25 |
| 3.14 | Crossing: the continuous line segment is said to cross over the broken line segment. . . . .  | 26 |
| 3.15 | Examples of knot diagrams for the: (a) $3_1^*$ , the mirror image of the trefoil $3_1$ ; and (b) five crossing torus knot. . . . .  | 26 |
| 3.16 | The three Reidemeister moves: (a) Reidemeister move I, uncrossing a loop; (b) Reidemeister move II, overlapping strands; (c) Reidemeister move III, moving the uppermost strand. . . . .  | 27 |
| 3.17 | Torus knots and links can be thought of as curves drawn on a 2-dimensional torus $T^2 \cong \mathbb{S}^1 \times \mathbb{S}^1$ . Here $\mathbb{S}^1 = \{\mathbf{x} \in \mathbb{R}^2 : \ \mathbf{x}\  = 1\}$ is the unit 1-sphere. . . . .  | 28 |
| 3.18 | Examples of links: (a) the Hopf link $2_1^2$ ; (b) the four crossing right handed torus link $4_1^2$ or 4-cat; and (c) the unlink or trivial link $0_1^2$ . . .   | 28 |

|      |   |    |
|------|---|----|
| 3.19 | Example of a 2-string tangle $(D^3, t)$ where the endpoints of the two tangle strings $t$ are anchored to the boundary of the unit ball $D^3$ at the equatorial points NW, NE, SW, and SE. . . . .  | 30 |
| 3.20 | Examples of regular tangle diagrams: (a) Tangle diagram for the tangle in Figure 3.19; (b) Tangle diagram for a tangle with 2 crossings.  | 31 |
| 3.21 | Examples of tangles: (a) a rational tangle is a tangle that can be obtained from a tangle, such as the tangle in Figure 3.20a, by moving the strings' endpoints on the boundary of the ball (and not allowing the strings to cross each other or self-intersect), (b) a locally knotted tangle is a tangle such that there exists a local knot in one of the strings, (c) a prime tangle is defined as a tangle that is neither locally knotted nor rational. . . . . | 33 |
| 3.22 | Four special types of rational tangles known as the <b>exceptional tangles</b> , and denoted as $(0)$ , $(\infty)$ , $(-1)$ , and $(+1)$ . . . . .  | 34 |
| 3.23 | Construction of positive and negative horizontal twists with four crossings: (a) starting tangle $(0)$ , (b) positive twist with four crossings, (c) negative twist with four crossings. . . . .  | 35 |
| 3.24 | Construction of positive and negative vertical twists with four crossings: (a) starting tangle $(\infty)$ , (b) positive twist with four crossings, (c) negative twist with four crossings. . . . .   | 36 |

|      |  |    |
|------|--|----|
| 3.25 | Tangle with classifying rational number $\frac{16}{9}$ . The entries of the tangle vector $(2, 3, 1, 1)$ describe the horizontal and vertical twists of the tangle. . . . .                                      | 36 |
| 3.26 | Horizontal addition, denoted by $T_1 + T_2$ , connects the points NE and SE of $T_1$ to the points NW and SW of $T_2$ by unknotted arcs. . . . .   | 38 |
| 3.27 | Vertical addition, denoted by $T_1 +' T_2$ , connects the points SW and SE of $T_1$ to the points NW and NE of $T_2$ by unknotted arcs. . . . .  | 39 |
| 3.28 | The horizontal addition of tangles $T_1 + T_2$ , where $T_1 = (\infty)$ and $T_2 = (\infty)$ , produces an object that is not a 2-string tangle. . . . .   | 39 |
| 3.29 | The numerator closure of a tangle $T$ , denoted by $N(T)$ , connects the points NW and NE, and SW and SE by unknotted arcs outside of $B^3$ . . . . .  | 40 |
| 3.30 | Numerator closure of $T_1 + T_2$ , denoted by $N(T_1 + T_2)$ . . . . .   | 40 |
| 3.31 | Example of a link $K$ (Hopf link $2_1^2$ ) produced from $N(T_1 + T_2 + T_3)$ , where $T_1 = (1)$ , $T_2 = (0)$ , and $T_3 = (1)$ . . . . .  | 41 |
| 4.1  | The pre-recombination synaptic complex is modeled by Equation 4.1. Here $P$ , the parental tangle, contains the recombination sites. . . . .   | 43 |
| 4.2  | The post recombination synaptic complex is modeled by Equation 4.2, where $R$ replaces $P$ . $R$ is called the recombinant tangle. . . . .   | 44 |
| 4.3  | The Tangle Method can be modified so that the pre-recombination synaptic complex is modeled by $N(O+P)$ , and the post-recombination synaptic complex is modeled by $N(O + R)$ , where $O = O_f + O_b$ . . . . . | 44 |

|     |   |    |
|-----|---|----|
| 5.1 | Visual representation of a permutation $\sigma$ on a set $\Omega = \{1, 2, 3, 4\}$ . . .  | 48 |
| 5.2 | Illustration of the permutation $\sigma = (143)$ . . . . .  | 49 |
| 5.3 | Examples of regular polygons with clockwise numbered vertices: (a)<br>3-gon, and (b) 4-gon. . . . .   | 51 |
| 5.4 | The regular 3-gon (or equilateral triangle) has two types of axes of<br>symmetry. (a) The equilateral triangle has three axes of symmetry.<br>Each of these passes through the midpoint of an edge and the opposite<br>vertex. For example, the permutation $(23)$ corresponds to a reflection<br>about the M-axis in the figure. (b) The axis of rotational symmetry<br>that passes through the center of the 3-gon. In this case from the<br>reader's perspective this axis (denoted by L) would pass through the<br>paper. . . . . | 52 |
| 5.5 | A multiplication table is used to prove that the symmetries or rigid<br>motions of a regular 3-gon satisfy the four axioms ( <b>G1</b> , <b>G2</b> , <b>G3</b> , <b>G4</b> )<br>that define a group. . . . .  | 53 |

|      |   |    |
|------|---|----|
| 5.6  | The regular tetrahedron has two types of axes of symmetry. (a) The line that passes through a vertex and the midpoint of the opposite face. For example, the permutation (142) corresponds to a 120° rotation about the M-axis in the figure. (b) The axis of rotational symmetry that passes through the midpoint of an edge and the midpoint of the opposite edge. For example, the permutation (24)(13) corresponds to a 180° rotation about the L-axis in the figure. . . . . | 54 |
| 5.7  | The twelve elements of the tetrahedral group $G = \{ (234), (243), (134), (143), (124), (142), (123), (132), (12)(34), (14)(23), (13)(24), (e) \}$ generated by the rotations about the M and L axes. Each face is painted a different color {pink, black, violet, white}. . . . .  | 56 |
| 5.8  | Tetrahedral tangles allow us to consider a non-planar geometry for the synaptosome in a site-specific recombination reaction. . . . .   | 58 |
| 5.9  | Tangle diagram for tangle vector (3, 5, 4, 2). . . . .  | 59 |
| 5.10 | Dimensions for each horizontal and vertical twist in the tangle diagram corresponding to the vector $(a_0, a_1, a_2, a_3) = (3, 5, 4, 2)$ , shown in Figure 5.9. . . . .  | 60 |
| 5.11 | The sine curves for the horizontal twist $a_1$ are given by the equations 5.1 and 5.2 with parameter values $n = 5$ , $c = 10$ , $r = 100$ , and the horizontal twist padding is 10. . . . .  | 61 |
| 5.12 | Identification of the twist corners. . . . .  | 63 |

|      |   |    |
|------|---|----|
| 5.13 | Visual representation of the tangle diagram for $(3, 5, 4, 2)$ embedded inside of a unit square. The $y$ -axis is reversed since the computer-graphics program we use orients the origin at the top left hand corner of the screen. . . . . | 63 |
| 5.14 | Visual representation of the mapping $\varphi_1 : t_1 \cup t_2 \rightarrow \mathbb{R}^3$ that takes the points for each tangle string bounded by the unit square to points inside of the tetrahedron $\Delta$ . . . . .                     | 64 |
| 5.15 | Tangle strings mapped inside of the tetrahedron where the crossings are constructed with helices. . . . .   | 66 |
| 5.16 | Visual representation of the mappings which takes the tangle (2) drawn in the unit square to the tetrahedron $\Delta$ via $\varphi_1$ and $\varphi_2$ . . . . .   | 67 |
| 5.17 | Computer implementation of tetrahedral tangle $\Delta_{(2)}^{\varphi_1}$ . . . . .  | 68 |
| 5.18 | Projections of $\Delta_{(2)}^{\varphi_1}$ , where the starting view is changed by a transformation equivalent to the (123) transformation. . . . .  | 69 |
| 5.19 | Projections of tetrahedral tangle $\Delta_{(2)}^{\varphi_1}$ generated by the twelve elements of the tetrahedral group $G = \{ (234), (243), (134), (143), (124), (142), (123), (132), (12)(34), (14)(23), (13)(24), (e) \}$ . . . . .      | 70 |
| 5.20 | Projections of tetrahedral tangle $\Delta_{(2)}^{\varphi_2}$ generated by the twelve elements of the tetrahedral group $G = \{ (234), (243), (134), (143), (124), (142), (123), (132), (12)(34), (14)(23), (13)(24), (e) \}$ . . . . .      | 71 |
| 5.21 | Tangle diagrams for tangles: (a) $(-2)$ , (b) $(2)$ , and (c) $(1)$ . . . . .   | 72 |

|      |  |    |
|------|--|----|
| 5.22 | The projections of tetrahedral tangle $\Delta_{(2)}^{\varphi_1}$ generated by the transformations (e), (123), and (124) can each be interpreted as one of the tangles in Figure 5.21. (a) can be interpreted as the tangle diagram for tangle $(-2)$ , (b) can be interpreted as the tangle diagram for tangle $(2)$ , and (c) can be interpreted as the tangle diagram for tangle $(1)$ . . . . . | 73 |
| 5.23 | Tangle diagram interpretations (a) $(-2)$ and (b) $(2, 0)$ for the tetrahedral tangle $\Delta_{(2)}^{\varphi_1}$ projections generated by $\mathcal{R}_1$ . . . . .  | 73 |
| 5.24 | Tangle diagrams interpretations (a) $(2)$ and (b) $(-2, 0)$ for the tetrahedral tangle $\Delta_{(2)}^{\varphi_1}$ projections generated by $\mathcal{R}_2$ . . . . .   | 74 |
| 5.25 | Tangle diagram interpretations (a) $(1)$ and (b) $(-1)$ for the tetrahedral tangle $\Delta_{(2)}^{\varphi_1}$ projections generated by $\mathcal{R}_3$ . . . . .   | 74 |
| 5.26 | Cyclic orderings $(1\ 4\ 2\ 3)$ and $(1\ 3\ 2\ 4)$ of the vertices for the tetrahedral tangle $\Delta_{(2)}^{\varphi_1}$ projections generated by $\mathcal{R}_1$ . . . . .  | 74 |
| 5.27 | Cyclic orderings $(1\ 2\ 3\ 4)$ and $(1\ 4\ 2\ 3)$ of the vertices for the tetrahedral tangle $\Delta_{(2)}^{\varphi_1}$ projections generated by $\mathcal{R}_2$ . . . . .  | 75 |
| 5.28 | Cyclic orderings $(1\ 3\ 4\ 2)$ and $(1\ 2\ 4\ 3)$ of the vertices for the tetrahedral tangle $\Delta_{(2)}^{\varphi_1}$ projections generated by $\mathcal{R}_3$ . . . . .  | 75 |
| 6.1  | Unlinking pathway of XerCD-FtsK complex for the right-handed 6-crossing torus link. The topology is simplified until the unlink topology is reached and the two DNA molecules are properly segregated. . . . .   | 77 |

|     |   |    |
|-----|---|----|
| 6.2 | Unknot converted to unlink after XerCD recombination at site <i>dif</i> . . .   | 78 |
| 6.3 | Visual representation of the three <i>TangleSolve</i> solutions for the unknot and unlink system. . . . .   | 79 |
| 6.4 | Hopf link converted to the unknot via one round of XerCD recombination at site <i>dif</i> . . . . .   | 80 |
| 6.5 | Visual representation of the three biologically relevant solutions for the Hopf Link and unknot system. . . . .   | 82 |
| 6.6 | Connected tetrahedral tangles generated by <i>Tangle3D</i> . . . . .  | 84 |
| 6.7 | Right-handed trefoil knot converted to the Hopf link via one round of XerCD recombination at site <i>dif</i> . . . . .  | 84 |
| 6.8 | Visual representation of the three biologically relevant <i>TangleSolve</i> solutions for the right-handed trefoil knot and Hopf link system. . . .   | 85 |
| 7.1 | Screenshot of Java Applet <i>TangleSolve</i> . When the program is started two windows will open: <i>Tangle Solver</i> and <i>Input Pane</i> . <i>Input Pane</i> is where information on the topology of the substrate and product (from experimental data) is entered. . . . .                               | 95 |
| 7.2 | <i>TangleSolve</i> Step 2: Enter the desired values for tangles $O_f$ , $O_b$ and $R$ into the <i>Input Pane</i> window. Here $O_f = (2)$ , $O_b = (0)$ , and $R = (-1)$ . In this example the substrate and product topologies are given by $N((2) + (0) + (0))$ and $N((2) + (0) + (-1))$ respectively. . . | 95 |

|     |  |     |
|-----|--|-----|
| 7.3 | <i>TangleSolve</i> Step 3: Select the <i>Display</i> button to view the tangle diagrams and 4-plats corresponding to the system $(\star)$ of tangle equations in the <i>Display Pane</i> . <i>TangleSolve</i> identifies the Hopf link as the substrate and the unknot as the product. . . . . | 96  |
| 7.4 | <i>TangleSolve</i> Step 4: Move the cursor over the tangle diagram for either the substrate or the product and right click on the image. A drop down menu will appear with the options <i>Save Image As...</i> , <i>Show In 3D...</i> and <i>Export....</i> . . . . .                          | 97  |
| 7.5 | <i>TangleSolve</i> Step 5: Move the cursor down the menu and hover over the <i>Show In 3D...</i> option. Another drop down menu will appear with several different combinations of the tangles to view in 3D. In this example the “Show ‘ $O_f$ ’ tangle in 3D” option was selected. . . .     | 97  |
| 7.6 | <i>TangleSolve</i> Step 6: After selecting the “Show ‘ $O_f$ ’ tangle in 3D” option, another window will open that displays tangle $O_f = (2)$ embedded inside of the tetrahedron and allows the user to freely rotate it in space. . . . .  | 98  |
| 7.7 | Projections of tetrahedral tangle $\Delta_{(0)}^{\varphi_1}$ generated by the twelve elements of the tetrahedral group. . . . .  | 100 |
| 7.8 | Projections of tetrahedral tangle $\Delta_{(1)}^{\varphi_1}$ generated by the twelve elements of the tetrahedral group. . . . .  | 101 |

7.9 Projections of tetrahedral tangle  $\Delta_{(3)}^{\varphi_1}$  generated by the twelve elements of the tetrahedral group. . . . . 102

# Chapter 1

## Introduction

*Escherichia coli* (*E. coli*) is a Gram-negative rod shaped bacterium that is a natural part of the human gut flora. However, some strains of *E. coli*, such as *E. coli* O157:H7, are known to cause serious and sometimes fatal infections in humans [4]. Bacterial infections, and the growing problem of antibiotic resistance, is a major health concern for society [2].

Targeting enzymatic actions involved in essential DNA transactions is one way to prevent infection. Every life form carries specific instructions for protein synthesis in its genetic material. Collectively, this genetic material is referred to as the *genome*. The preservation of the genome is vital to the health of the cell; without it the cell will perish. Bacteria are prokaryotic organisms; in general their genome consists of one single circular chromosome. On the other hand, eukaryotes are multi-cellular organisms including yeast and fungi. Circular DNA can become entwined and form knots or links. Knotted or linked DNA pose problems for the cell, because they

interfere with important biological processes including replication and transcription [31, 9, 17].

In the case of *E. coli*, special enzymes known as *topoisomerases* disentangle knotted and linked DNA to prevent cell death. It has recently been shown that in the absence of topo IV, a type II topoisomerase, XerC and XerD, which are *site-specific recombinases* can also disentangle these knots or links [15]. Site-specific recombination can be studied with knots and tangles in a model known as the tangle method [11]. The tangle method models the synaptosome, or enzyme(s) bound to the DNA, as a 2-string tangle with co-planar endpoints, which may not reflect the geometry of the recombination sites. This paper addresses this limitation with an application of the DNA unlinking mechanism of site-specific recombinases XerC and XerD (XerCD) in *E. coli*.

We give an introduction to the microbiology behind these enzymatic actions as well as the mathematics (knot theory, tangle theory) and the computational tools used to model and visualize these processes. In Chapter 2, we provide the biological background for XerCD site-specific recombination. In Chapter 3, we introduce the mathematical theory used to model site-specific recombination. In Chapter 4, we discuss the details of the tangle method. In Chapter 5, we define a tetrahedral tangle using results from group theory and vector calculus, and define equivalence classes for the projections of tetrahedral tangles. In Chapter 6, we apply tetrahedral tangles to the study of XerCD-FtsK unlinking. To study tetrahedral tangles we developed a

computer program called *Tangle3D* that works with *TangleSolve*, a Java application that implements the tangle method. A tutorial on how to use *TangleSolve's* new features, including *Tangle3D*, is provided in Appendix B.

## Chapter 2

# Biological Background

### 2.1 Replication of the bacterial chromosome

Bacteria are prokaryotic organisms that possess one double-stranded circular DNA molecule, that makes up most of the cell's genome, densely packed in a region called the *nucleoid* [6]. Bacteria may also contain DNA in the form of *plasmids*: small, circular, self-replicating DNA molecules (see Figure 2.1).

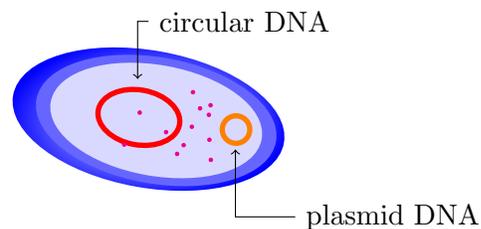


Figure 2.1: A typical rod shaped bacterium: the chromosome and plasmids make up the organism's genetic material or genome.

Sometimes the bacterial chromosome becomes entangled, this entanglement may

include knotting, linking, and even supercoiling. Knotted and linked DNA molecules pose a serious problem to the cell for they interfere with important cellular processes [10]. To remedy this problem, *enzymes*, called *topoisomerases*, untangle DNA knots and links through *strand passage*. Unknotting and unlinking DNA is vital to the health of the cell. For example, upon DNA replication, circular chromosomes become interlinked as shown in Figure 2.2 (b).

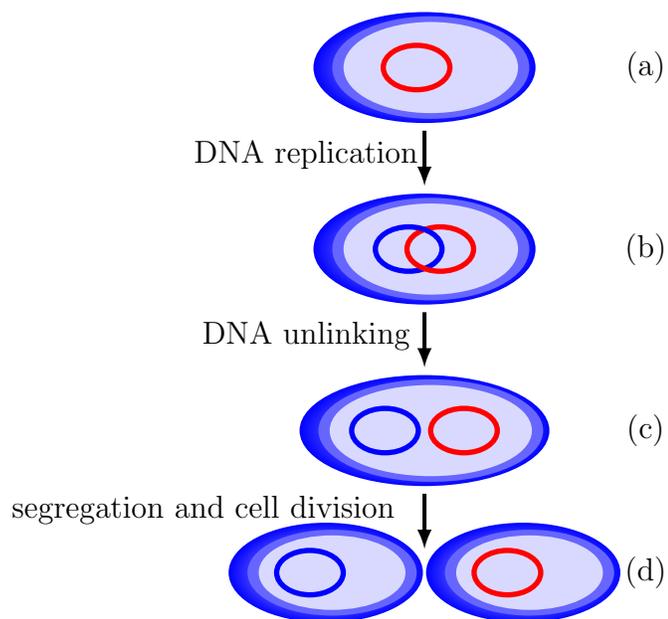


Figure 2.2: Example of normal bacterial cell division: (a) a double-stranded circular parental chromosome is indicated as a circle inside the cytoplasm of a bacterium (b) replication results in two copies of the parental chromosome that are interlinked (replication links) (c) unlinking of replication links through strand passage (d) two daughter cells that each inherit one copy of the parental DNA.

If the two copies of DNA are not properly segregated, then several outcomes can

occur: (1) one daughter cell will inherit two copies of the genome, and the other daughter cell will receive no DNA resulting in its untimely death, (2) the interlinked DNA will experience breakage and one daughter cell will receive some form of bad inheritance. Figure 2.2 shows the steps taken to resolve interlinked DNA molecules. Figure 2.3 illustrates one possible outcome if the two copies of DNA are not properly segregated.

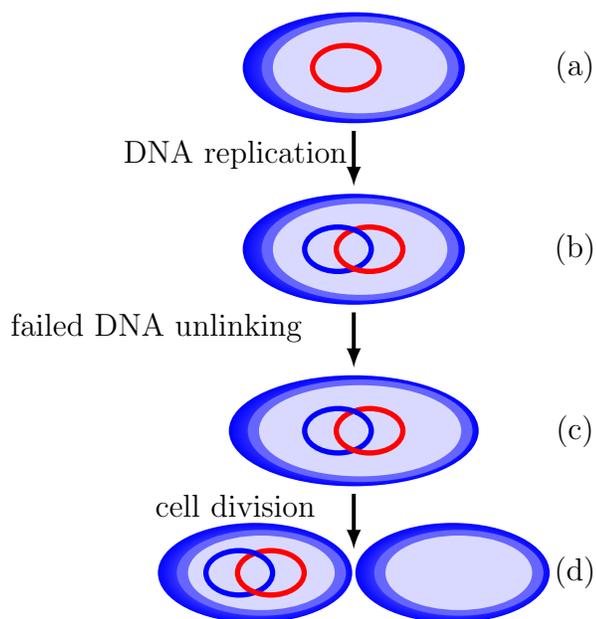


Figure 2.3: Example of improper bacterial cell division (a) a double-stranded circular parental chromosome is indicated as a circle in the cytoplasm of a bacterial cell (b) replication results in two copies of the parental chromosome that are interlinked (replication links) (c) enzymes fail to properly segregate (unlink) the two DNA molecules (d) one daughter cell inherits two copies of the DNA and the second daughter cell does not inherit any DNA.

### 2.1.1 Enzymes that change the topology of DNA: topoisomerases and site-specific recombinases

In 1971, James Wang discovered *topoisomerases*, enzymes able to change the topology of circular DNA, in *E. coli* [30]. Topoisomerases have been shown to play an integral role in solving topological problems associated with DNA replication and transcription by introducing temporary single- or double-strand breaks in the DNA, and mediating single- or double-strand passage [1]. The action of *type I* and *type II topoisomerase* is characterized by changes in linking number of  $\pm 1$  and  $\pm 2$ , respectively. The *linking number* is the number of times the closed circular DNA molecules are linked.

*Recombination* is the exchange of genetic material in DNA. There are two major types: *homologous recombination* and *site-specific recombination*. During meiosis, a two-stage type of cell division in sexually reproducing organisms that results in cells with half the chromosome number of the original cell [6], homologous recombination breaks and recombines a pair of homologous DNA sequences creating a mixture of the inherited genes [3]. Homologous recombination is also used as a faithful DNA repair mechanism [27]. Site-specific recombination is a more specialized process which requires the recognition of specific sites, and is discussed in more detail in Section 2.1.1. *Site-specific recombination* occurs when one or more enzymes, known as *site-specific recombinase(s)*, identify and attach themselves to the *substrate* DNA molecule(s) at the pair of *recombination sites*. A recombination site is a short

segment of duplex DNA whose sequence is recognized by the recombinase [18]. The combined complex consisting of the site-specific recombinases bound to the DNA substrate is called a *synaptic complex* [22]. Figure 2.4 represents the enzymes as a 3-dimensional ball, and the two segments of duplex DNA as two arcs inside the ball.

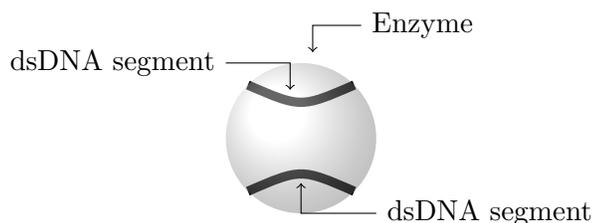


Figure 2.4: The synaptosome is the part of the synaptic complex that consists of the recombinase (enzyme) bound to two segments of double stranded DNA (dsDNA), the recombination sites. Notice that the two DNA segments could be intertwined in non-trivial ways.

Notice that the DNA is modeled as the curve drawn by the axis of the double helix. The DNA resulting from site-specific recombination is called the *product* [22].

**Site-specific recombination** is an enzymatic reaction that involves the following components:

- (a) two segments of double stranded DNA (dsDNA) called **recombination sites**;
- (b) one or more specialized enzymes, called **site-specific recombinases**, that are responsible for recognizing the sites and for breaking and rejoining the DNA within these sites;

and (c) a mechanism that involves DNA breakage and reunion with conservation of the phosphodiester bond energy (see [7, 26]).

Figure 2.5 illustrates an example of the site-specific recombination process.

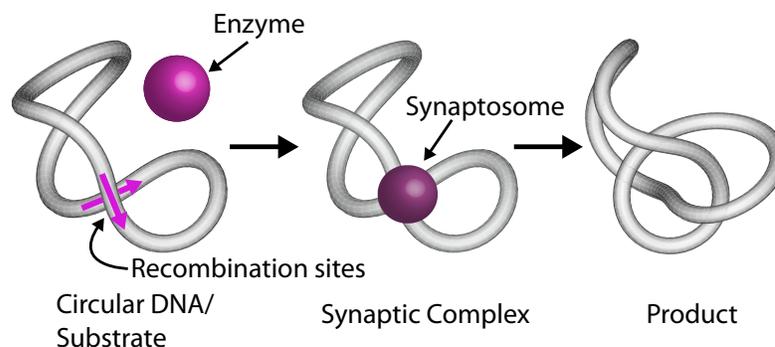


Figure 2.5: Site-specific recombination on circular DNA. First, the enzymes recognize the recombination sites where they are brought together to form the synaptic complex. Next, the site-specific recombinases cleave the DNA, perform strand exchange, and reseal the breaks.

The substrate is an unknotted circular DNA molecule not bound to enzymes before site-specific recombination. The synaptic complex consists of circular DNA substrates where two recombination sites are bound to one or more enzymes. The product is illustrated by a knotted circular DNA molecule that has undergone one or more rounds of site-specific recombination, and is not bound to the site-specific recombinases and accessory proteins.

There are two families of recombinases: *tyrosine* and *serine*. In Section 2.1.2 we discuss two tyrosine recombinases, XerC and XerD. Site-specific recombination can produce different types of structural changes such as *integration*, *excision*, or *inversion*. Each of these changes depends on whether the recombination sites are located on the same molecule or on separate molecules, and on the relative orientation of the sites prior to recombination. When the two sites are located on separate molecules,

recombination will result in integration. An example of site-specific recombination that results in integration is shown in Figure 2.6.

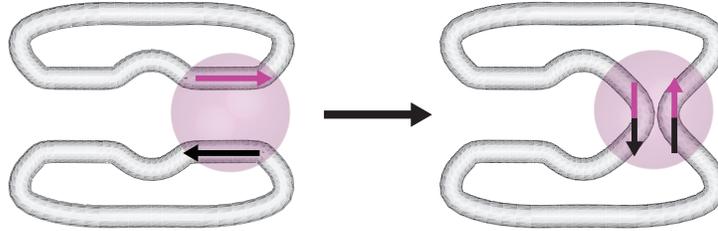


Figure 2.6: Site-specific recombination for recombination sites arranged in anti-parallel on two DNA molecules resulting in integration.

When the two sites are located on the same DNA molecule, depending on the initial arrangement of the sites, recombination will result in either excision, inversion or transposition. An example of excision is shown in Figure 2.7.

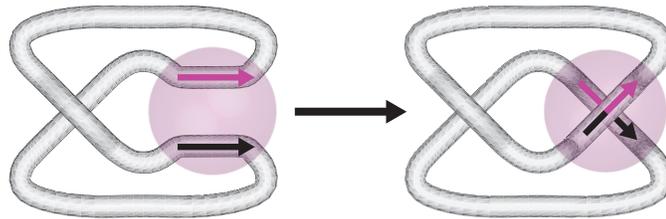


Figure 2.7: Site-specific recombination for recombination sites arranged in parallel resulting in excision.

There are two possible arrangements for the recombination sites. Recall that a recombination site is a DNA segment of specific length and nucleotide sequence. For example, site-specific recombinases XerC and XerD recognize the *dif* site on the *E. coli* chromosome. The *dif* site is a 28 base pair (bp) long dsDNA segment described

by the sequence GGTGCGCATAATGTATATTATGTTAAAT. When the sequence of the recombination site is not palindromic, as in the case of *dif*, one can assign an orientation onto the site, which induces an orientation onto the molecule. In Figure 2.7 the sites are arranged in anti-parallel and in Figure 2.6 the they are parallel.

### 2.1.2 XerC and XerD Site-Specific Recombinases

XerC and XerD (XerCD) are tyrosine recombinases that catalyze recombination through a *Holliday junction intermediate*. Figure 2.8 gives a schematic for XerCD attached to the recombination site *dif*, a 28 bp long DNA segment along the *E. coli* chromosome.

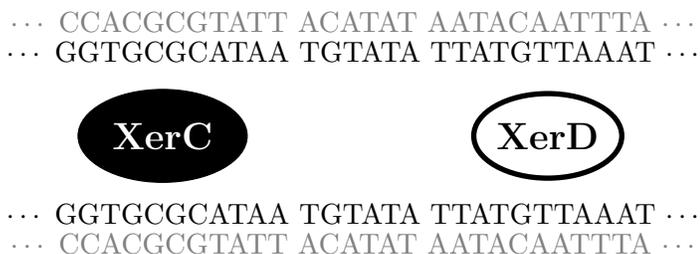


Figure 2.8: Tyrosine site-specific recombinases XerC and XerD (XerCD) attached to recombination site *dif* on the *E. coli* chromosome, which is a 28 base pair (bp) long dsDNA segment.

In 2003, it was shown that recombinases XerCD and DNA translocase FtsK<sub>50C</sub> can unlink catenated DNA circles *in vitro* [16]. These experiments were performed on catenanes, interlinked circular DNA molecules, with parallel *dif* sites that were produced by Cre recombination on a plasmid with directly repeated *loxP* sites. Cre

is a tyrosine recombinase. The results of these experiments suggested that XerCD and FtsK can convert catenanes to free circles in a stepwise fashion, simplifying the topology with knot intermediates. In 2007, David Sherrat's group showed that XerCD-*dif*-FtsK can unlink the various types of replication catenanes produced *in vivo* when topoIV, a type II topoisomerase, is inhibited [15].

Furthermore, Sherrat *et. al.* confirmed that XerCD-*dif*-FtsK interconverts catenanes to knots when simplifying to free DNA circles, and proposed an unlinking scheme. By observing changes in DNA topology (knotting and linking) it is possible to characterize the local enzymatic action through the use of knot theory and tangle calculus. In 2005, Vazquez *et. al.* used tangle analysis to show that the three solutions for XerCD recombination at *psi* sites (on plasmid DNA) can be unified into one three-dimensional object [21]. More recently, in 2013 Shimokawa *et. al.* used the crystal structure of Cre-DNA as a model for XerCD, to show how that the last three unlinking steps of XerCD recombination can be unified into one three dimensional object. In the following chapter, we introduce the mathematical tools used to model Xer recombination.

## Chapter 3

# Mathematical Background

### 3.1 Knots

Mathematical knots and links are used to model the *topology* of circular DNA. Intuitively, a knot can be thought of as the knot tied with a rope or shoelace. Here we think of a knot as an entwined circle without self-intersection in 3-dimensional space. How this circle is entwined in space determines the topology. Before giving the formal definition of a knot, it is necessary to give some background from *general topology* or *point set topology*.

#### 3.1.1 Set Theory

The following definitions rely on the notion of a set. A *set* is a collection of objects and a precise description used to determine if an object belongs to it. The objects that comprise a set, normally denoted by lower case letters, are called its *elements*.

The *empty set* is the set that does not contain any elements, and is also referred to as the *trivial set*. See [13] for an introduction to set theory, the following definitions are borrowed from this text.

**Definition 3.1.** A set  $A$  is a **subset** of a set  $B$ , written as  $A \subset B$ , if every element of  $A$  is an element of  $B$  (see Figure 3.1).

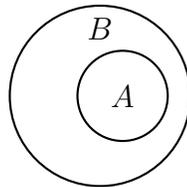


Figure 3.1: An example of sets  $A$  and  $B$  where  $A \subset B$ . We say that  $A$  is a subset of  $B$ , or  $A$  is contained in  $B$ .

**Definition 3.2.** The **union** of sets  $A$  and  $B$ , denoted by  $A \cup B$ , is the set of all elements which belong to  $A$  or  $B$  or both (see Figure 3.2); it follows that

$$A \cup B = \{x : x \in A \text{ or } x \in B\}.$$

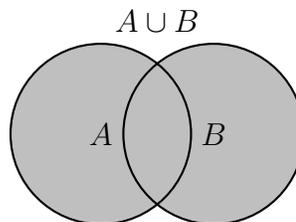


Figure 3.2: The shaded region represents the union of sets  $A$  and  $B$ , denoted by  $A \cup B$ .

**Definition 3.3.** The **intersection** of sets  $A$  and  $B$ , denoted by  $A \cap B$ , is the set of all elements which belong to both  $A$  and  $B$  (see Figure 3.3), i.e.,

$$A \cap B = \{x : x \in A \text{ and } x \in B\}.$$

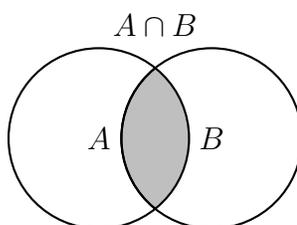


Figure 3.3: The shaded region represents the intersection of sets  $A$  and  $B$ , denoted by  $A \cap B$ .

**Definition 3.4.** Let  $A$  and  $B$  be two sets. The **Cartesian product set** of  $A$  and  $B$ , written as  $A \times B$ , consists of all ordered pairs  $(a, b)$  where  $a \in A$  and  $b \in B$ , i.e.,

$$A \times B = \{(a, b) : a \in A, b \in B\}.$$

**Definition 3.5.** A **binary relation** between two sets  $A$  and  $B$ , denoted by  $\sim$ , is a subset  $R$  of  $A \times B$ . We say that “ $a$  is related to  $b$ ”, and denote it by  $a \sim b$ , if the ordered pair  $(a, b)$  is in  $R$ . A binary relation from a set  $A$  to the same set  $A$  is called a **relation** on  $A$ .

**Definition 3.6.** A relation  $\sim$  on a set  $A$ , i.e. a subset of  $A \times A$ , is an **equivalence relation** if and only if it satisfies the following axioms:

**E1:** For every  $a \in A$ ,  $a \sim a$ .

**E2:** If  $a \sim b$ , then  $b \sim a$ .

**E3:** If  $a \sim b$  and  $b \sim c$ , then  $a \sim c$ .

In general, a relation is said to be **reflexive** if and only if (iff) it satisfies **E1**, **symmetric** iff it satisfies **E2**, and **transitive** iff it satisfies **E3**.

**Definition 3.7.** If  $\sim$  is an equivalence relation on  $A$ , then the equivalence class of any element  $a \in A$ , denoted by  $[a]$ , is the set of elements to which  $a$  is related i.e.,

$$[a] = \{x \in A : x \sim a\}.$$

**Theorem 3.1.** *Let  $\sim$  be an equivalence relation in  $A$  and let  $[a]$  be the equivalence class of  $a \in A$ . Then:*

(i) *For every  $a \in A$ ,  $a \in [a]$ .*

(ii)  *$[a] = [b]$  if and only if  $a \sim b$ .*

(iii) *If  $[a] \neq [b]$ , then  $[a]$  and  $[b]$  are disjoint.*

### 3.1.2 Basic Definitions in Topology and Knot Theory

**Definition 3.8.** Let  $X$  be a nonempty set. A collection  $\mathcal{T}$  of subsets of  $X$  is a **topology** on  $X$  if and only if  $\mathcal{T}$  satisfies the following axioms:

**O1:**  $\emptyset$  (empty set) and  $X$  are in  $\mathcal{T}$ .

**O2:** The union of the elements of any subcollection of  $\mathcal{T}$  is in  $\mathcal{T}$ .

**O3:** The intersection of the elements of any finite subcollection of  $\mathcal{T}$  is in  $\mathcal{T}$ .

The members of  $\mathcal{T}$  are then called  $\mathcal{T}$ -**open sets**, or simply **open sets**. The pair  $(X, \mathcal{T})$  is called a **topological space**.

Given the set  $X = \{u, v, w\}$ , it can be shown that the collections  $\mathcal{T}_1 = \{\emptyset, X\}$ , and  $\mathcal{T}_2 = \{\emptyset, \{u\}, \{v\}, \{u, v\}, X\}$  satisfy the three axioms in Definition 3.8 (see Figure 3.4 ).

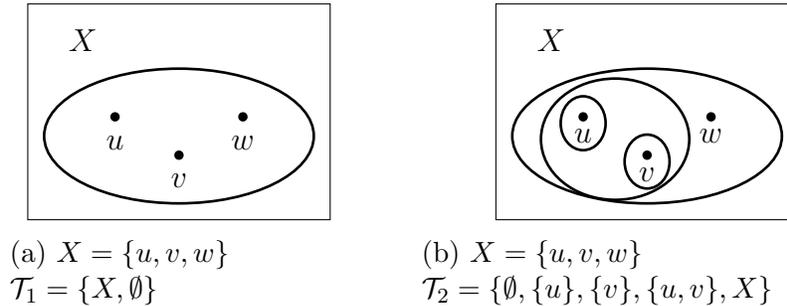


Figure 3.4: Examples of topologies on the set  $X = \{u, v, w\}$ : (a)  $\mathcal{T}_1 = \{X, \emptyset\}$  and (b)  $\mathcal{T}_2 = \{\emptyset, \{u\}, \{v\}, \{u, v\}, X\}$ .  $\mathcal{T}_1$  and  $\mathcal{T}_2$  satisfy all three axioms **O1**, **O2**, and **O3**.

Collection  $\mathcal{T}_3 = \{\emptyset, \{v\}, \{w\}, X\}$  is not a topology because it does not satisfy axiom **O2**, the union of  $\{v\}$  and  $\{w\}$  is not in  $\mathcal{T}_3$  (see Figure 3.5a). Collection  $\mathcal{T}_4 = \{\emptyset, \{u, v\}, \{v, w\}, X\}$  is also not a topology because it violates axiom **O3**. The intersection of  $\{u, v\}$  and  $\{v, w\}$  is not in  $\mathcal{T}_4$  (see Figure 3.5b).

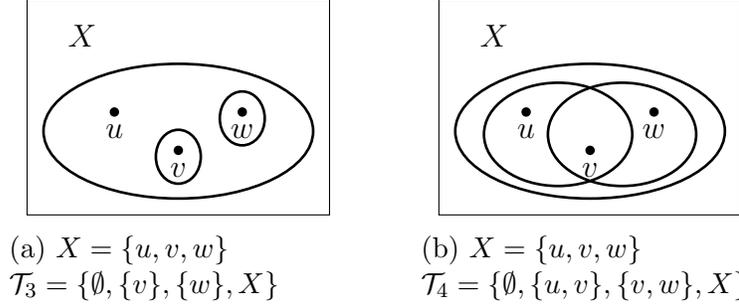


Figure 3.5: Examples of collections that are not topologies on the set  $X = \{u, v, w\}$ . (a)  $\mathcal{T}_3$  is not a topology on the set  $X$  because it fails to satisfy axiom **O2**. The union of sets  $\{v\}$  and  $\{w\}$  is not an element of  $\mathcal{T}_3$ . (b)  $\mathcal{T}_4$  is not a topology on the set  $X$  because it violates axiom **O3**. The intersection of the sets  $\{u, v\}$  and  $\{v, w\}$  is not an element of  $\mathcal{T}_4$ .

**Definition 3.9.** Two topological spaces  $(X, \mathcal{T})$  and  $(Y, \mathcal{T}^*)$  are called **homeomorphic** or **topologically equivalent** if there exists a bijective function  $h : X \rightarrow Y$  such that  $h$  and  $h^{-1}$  are continuous. The function  $h$  is called a **homeomorphism** or **topological map**.

For this research we consider Euclidean spaces  $\mathbb{R}^2$  and  $\mathbb{R}^3$ .

**Definition 3.10.** A topological space  $(X, \mathcal{T})$  is a **Hausdorff space** (see Figure 3.6) if and only if it satisfies the following axiom:

**H1:** For each pair of distinct points  $x, y \in X$  there exist open sets  $U$  and  $V$  such that  $x \in U$  and  $y \in V$  and  $U \cap V = \emptyset$ .

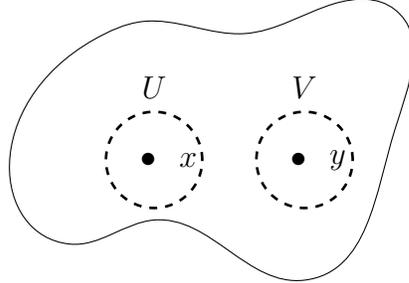


Figure 3.6: A topological space  $X$  is Hausdorff if each pair of distinct points  $x, y \in X$  belong to disjoint open sets  $U$  and  $V$  such that  $x \in U$  and  $y \in V$ .

**Definition 3.11.** Let  $X$  and  $Y$  be Hausdorff spaces. A mapping  $f : X \hookrightarrow Y$  is called an **embedding** if  $f : X \rightarrow f(X)$  is a homeomorphism.

**Definition 3.12.** Two embeddings,  $f_0, f_1 : X \hookrightarrow Y$  are **isotopic** if there is an embedding

$$F : X \times I \hookrightarrow Y \times I, \quad (x, t) \mapsto F(x, t) =: (f(x), t),$$

where  $f(x, 0) = f_0(x)$ ,  $f(x, 1) = f_1(x)$ , and  $f(x, t)$  is continuous, for all  $t \in I = [0, 1]$ .

$F$  is called a **level-preserving isotopy** connecting  $f_0$  and  $f_1$ .

Generally speaking an isotopy means that given two embeddings  $f_0$  and  $f_1$ , there exists another embedding connecting them. It is helpful to use a simple example to explain this definition. Let  $X$  consist of a single point  $\{p\}$ ,  $Y = [a, b]$  be a line segment, and  $f_0$  and  $f_1$  be two embeddings from  $X$  to  $Y$  (see Figure 3.7).

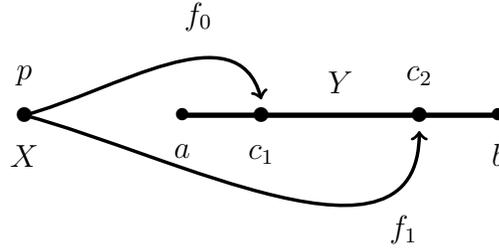


Figure 3.7: Example of two embeddings  $f_0$  and  $f_1$  that map  $X$ , which consists of a single point  $\{p\}$ , to the line segment  $Y = [a, b]$ . In this case  $f_0(p) = c_1$  and  $f_1(p) = c_2$ , and each  $f(p, t)$  is a point in  $Y$  between  $a$  and  $b$ .

In this example  $f_0(p) = c_1$  and  $f_1(p) = c_2$ , where  $c_1, c_2 \in [a, b]$ . We say that  $f_0$  and  $f_1$  are isotopic if there exists an embedding  $F$  that takes the line segment  $X \times I$  to the rectangle  $Y \times I$ , where  $I = [0, 1]$ , and connects  $f_0$  and  $f_1$ .  $F$  is called a level preserving isotopy (see Figure 3.8).

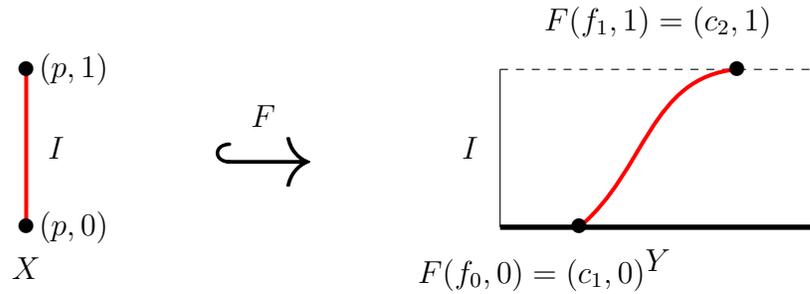


Figure 3.8: Example of a level preserving isotopy  $F : X \times I \hookrightarrow Y \times I$  that connects embeddings  $f_0, f_1 : X \hookrightarrow Y$ , where  $X = \{p\}$ ,  $Y = [a, b]$ , and  $I = [0, 1]$ . Here  $F$  is defined as  $F(f(x, t), t)$  where  $f(p, 0) = f_0$  and  $f(p, 1) = f_1$  and  $f(x, t)$  is continuous for all  $t$ .

**Definition 3.13.** Two embeddings  $f_0, f_1 : X \hookrightarrow Y$  are **ambient isotopic** if there

is a level preserving isotopy

$$H : Y \times I \hookrightarrow Y \times I, \quad H(y, t) = H(h_t(y), t),$$

with  $f_1 = h_1 f_0$  and  $h_0 = \text{Id}_Y$ , the identity map in  $Y$ . The mapping  $H$  is called an **ambient isotopy**.

**Definition 3.14.** Let  $X, Y$  be polyhedra and  $f_0, f_1 : X \hookrightarrow Y$  be piecewise linear embeddings,  $f_0$  and  $f_1$  are **piecewise-isotopic** if there is a level-preserving piecewise linear embedding

$$F : X \times I \hookrightarrow Y \times I, \quad F(x, t) = (f_t(x), t), \quad 0 \leq t \leq 1.$$

The piecewise linear embeddings  $f_0$  and  $f_1$  are **piecewise-ambient isotopic** if there is a level-preserving piecewise linear-isotopy

$$H : Y \times I \hookrightarrow Y \times I, \quad H(y, t) = (h_t(y), t),$$

with  $f_1 = h_1 f_0$  and  $h_0 = \text{Id}_Y$ .

**Definition 3.15.**  $K$  is a **knot** if there exists an embedding  $H : C \hookrightarrow \mathbb{R}^3$  such that  $K = H(C)$ , where  $C = \{(x, y) \in \mathbb{R}^2 : x^2 + y^2 = 1\}$  is the unit circle in  $\mathbb{R}^2$  (see Figure 3.9). Figure 3.10 displays the simplest knots, namely the *unknot*  $0_1$  and the *trefoil*  $3_1$ .

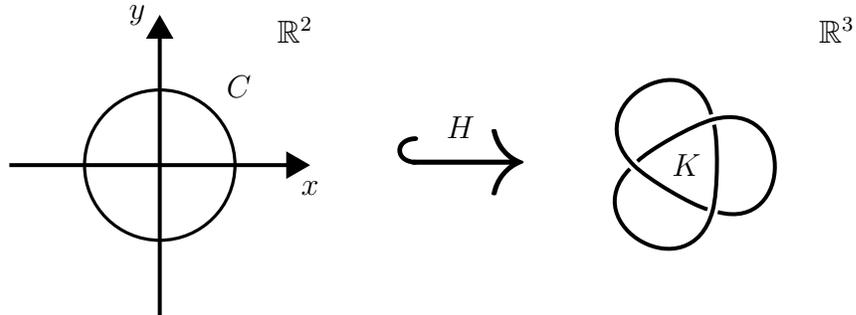
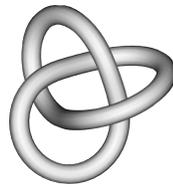


Figure 3.9:  $K$  is a knot if there is an embedding  $H$  that takes the unit circle  $C = \{(x, y) \in \mathbb{R}^2 : x^2 + y^2 = 1\}$  to a closed curve in  $\mathbb{R}^3$ , where  $H(C) = K$ .



(a) Unknot  $0_1$ .



(b) Trefoil,  $3_1$ .

Figure 3.10: Examples of knots: (a) unknot or trivial knot; and (b) trefoil.

Definition 3.15 defines a knotted configuration in  $\mathbb{R}^3$ . We are interested in determining when two knotted configurations are equivalent. Intuitively, two knotted configurations are topologically equivalent if one can be deformed into the other in space. A formal definition of knot equivalence requires some additional mathematical concepts.

**Definition 3.16.** A **knot type** is the equivalence class of piecewise-linear embeddings of a circle into  $\mathbb{R}^3$ .

**Definition 3.17.** Two piecewise linear knots are **equivalent** if they are piecewise linear ambient isotopic.

From this point on we will omit the prefix piecewise linear.

### 3.1.3 Projection of a Knot

In the previous section a knot (or knotted configuration) was defined as an embedding of a circle in  $\mathbb{R}^3$ . Without loss of generality, we here refer to the knot as the image under the embedding. In this case, a knot  $K$  is a 3-dimensional object, but it is helpful to draw and work with knots in the plane. This is achieved by taking their *planar projections*.

**Definition 3.18.** If  $K$  is a knot in  $\mathbb{R}^3$ , the **projection** of  $K$  is the image of  $K$ , denoted by  $\mathcal{P}(K)$ , under a map that takes a point in  $\mathbb{R}^3$  onto a point in a plane.

**Example 3.1.** The function from  $\mathbb{R}^3$  to  $\mathbb{R}^2$  which takes the triple  $(x, y, z)$  to the pair  $(x, y)$  is called the **orthogonal projection** onto the  $xy$ -plane.

To visualize Example 3.1, imagine shining a light onto a three dimensional object in a direction perpendicular to the  $xy$ -plane, and taking its shadow as the projection. In general, a projection can be thought of as shining a light onto a knot in any one direction, and hence there are infinitely many projections. (see Figure 3.11).

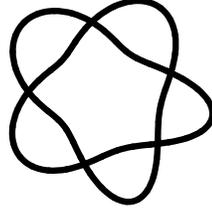


Figure 3.11: Planar projection of a five crossing torus knot  $5_1$  onto its mirror image  $5_1^*$ . We refer the reader to [5] for a biologically relevant knot table.

**Definition 3.19.** A point  $p$  in the image  $\mathcal{P}(K)$  is called a **multiple point** if  $|\mathcal{P}^{-1}(p) \cap K| > 1$ . The **order** of  $p \in \mathcal{P}(K)$  is the cardinality of  $(\mathcal{P}^{-1}(p) \cap K)$ . A **double point** is a multiple point of order 2 (see Figure 3.12).

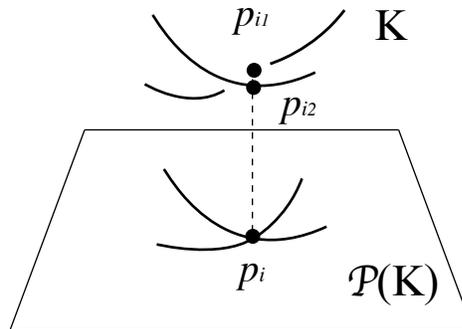


Figure 3.12: The planar projection of these two arcs results in a double point  $p_i$ , where points  $p_{i_1}$  and  $p_{i_2}$  are mapped to  $p_i$  in the projection  $\mathcal{P}(K)$ .

**Definition 3.20.** A projection  $\mathcal{P}(K)$  of a knot  $K$  is called **regular** if there are only finitely many multiple points  $\{p_i | 1 \leq i \leq n, n \in \mathbb{N}\}$ , and all multiple points are double points. That is, for every  $p_i \in \mathcal{P}(K)$ ,  $|\mathcal{P}^{-1}(p_i) \cap K| \leq 2$ .

To study knots we restrict the type of projections to *regular projections*. For the knot projections drawn here, only the double point intersection in Figure 3.13 (a) is allowed, otherwise the mapping would be ambiguous.

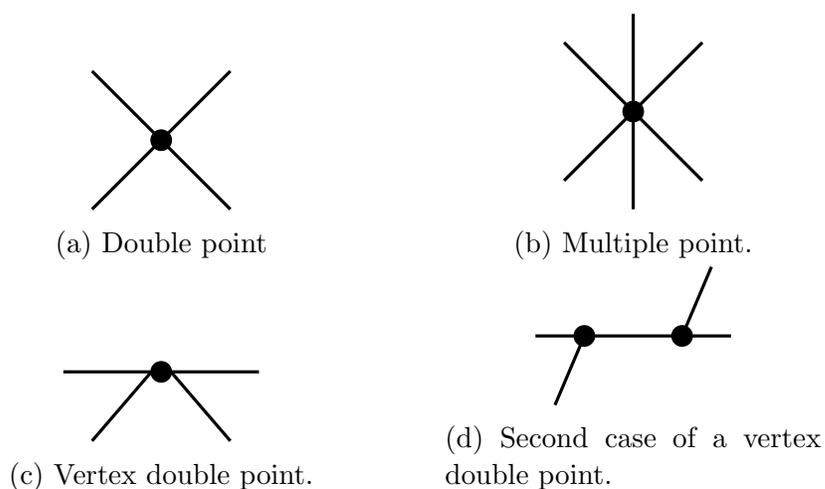


Figure 3.13: Intersections of a knot projection: (a) double point intersection; (b) multiple point intersection; (c) vertex double point intersection; (d) a second form of a vertex double point intersection. Only the double point intersection (a) is allowed in a regular knot projection. (Redrawn from [22].)

### 3.1.4 Knot Diagrams

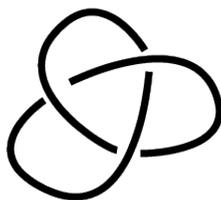
One disadvantage with using regular projections is that the crossing information is lost. That is, it is impossible to discern which parts of the knot cross over or under other parts of the knot. To resolve this problem, gaps are left in the drawings of regular projections. The drawing of a knot with these modifications is called a *knot diagram*. Figure 3.14 shows how the gaps of a knot diagram are interpreted. The

continuous line segment is said to cross over the broken line segment.

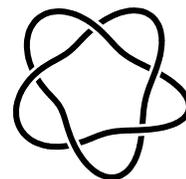


Figure 3.14: Crossing: the continuous line segment is said to cross over the broken line segment.

Figures 3.15 illustrates the knot diagrams for the trefoil  $3_1^*$ , the mirror image of the trefoil  $3_1$ , and for  $5_1$ , the five crossing torus knot.



(a) Knot diagram for  $3_1^*$ .



(b) Knot diagram for  $5_1$ .

Figure 3.15: Examples of knot diagrams for the: (a)  $3_1^*$ , the mirror image of the trefoil  $3_1$ ; and (b) five crossing torus knot.

### 3.1.5 Reidemeister Moves

It is important to note that different spatial realizations of a knot type in  $\mathbb{R}^3$  can give rise to different knot diagrams. In 1926, Kurt Reidemeister showed that two knot diagrams represent the same knot type if one knot diagram can be converted into

the other by a finite sequence of *Reidemeister moves* [23]. The three Reidemeister moves are illustrated in Figure 3.16.

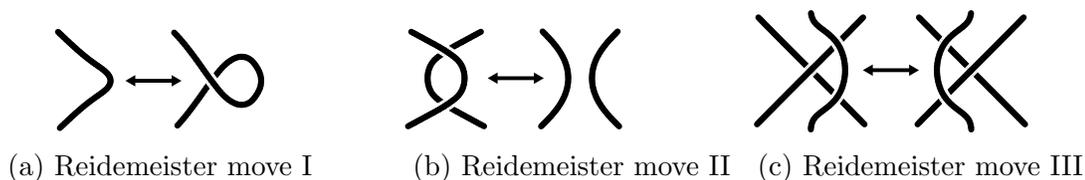


Figure 3.16: The three Reidemeister moves: (a) Reidemeister move I, uncrossing a loop; (b) Reidemeister move II, overlapping strands; (c) Reidemeister move III, moving the uppermost strand.

**Theorem 3.2.** (*Reidemeister 1927*) *Two knots  $K_1$  and  $K_2$  are said to be **equivalent** the knot diagram for  $K_1$  can be deformed into the knot diagram for  $K_2$  by a finite sequence of Reidemeister moves.*

## 3.2 Links

**Definition 3.21.** A **link** is a finite disjoint union of knots  $L = K_1 \sqcup K_2 \sqcup \cdots \sqcup K_n$ . Each knot  $K_i$  is called a component of the link  $L$ , where  $1 \leq i \leq n$ . In particular, a knot is a link with one component.

The most commonly observed substrates and products of Xer recombination are torus knots and torus links, so called because they can be defined as curves drawn on the surface of a torus (see Figure 3.17).

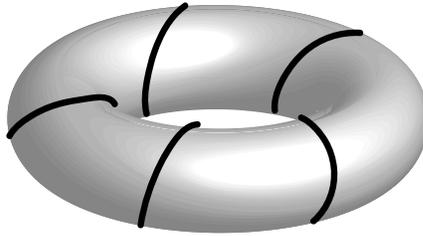
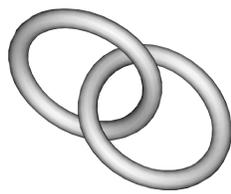
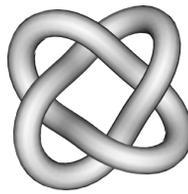


Figure 3.17: Torus knots and links can be thought of as curves drawn on a 2-dimensional torus  $T^2 \cong \mathbb{S}^1 \times \mathbb{S}^1$ . Here  $\mathbb{S}^1 = \{\mathbf{x} \in \mathbb{R}^2 : \|\mathbf{x}\| = 1\}$  is the unit 1-sphere.

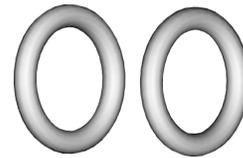
Figure 3.18 illustrates some examples of links. The notation for a link is given by  $N_Y^C$ , where  $N$  is the number of crossings,  $C$  is the number of components, and  $Y$  represents the knot type.



(a) Hopf Link ( $2_1^2$ )



(b) 4-cat ( $4_1^2$ )



(c) Unlink ( $0_1^2$ )

Figure 3.18: Examples of links: (a) the Hopf link  $2_1^2$ ; (b) the four crossing right handed torus link  $4_1^2$  or 4-cat; and (c) the unlink or trivial link  $0_1^2$ .

## 3.3 Tangles

### 3.3.1 Tangle Definition

In 1990, Ernst and Sumners developed *tangle calculus* to create a model for DNA recombination, known as the *Tangle Method*, that uses knot theory and *low-dimensional topology* [11]. In Chapter 4 we discuss the details of the *Tangle Method*. Here we provide some preliminary background of tangle theory.

**Definition 3.22.** A **2-string tangle (tangle)** is a pair  $(B^3, t)$ , where  $B^3$  is a 3-ball and  $t$  is a pair of disjoint (unoriented) arcs properly embedded in  $B^3$  with  $t \cap \partial B^3 = \partial t$  [20]. Note that the arcs may intertwine in non-trivial ways.

To construct a tangle, consider the unit ball centered at the origin in  $\mathbb{R}^3$ ,  $D^3 = \{\mathbf{x} \in \mathbb{R}^3 : \|\mathbf{x}\| \leq 1\}$ , and embed two curves inside  $D^3$  such that each of the four endpoints are anchored to one of the following four points  $\{\text{NW}, \text{NE}, \text{SW}, \text{SE}\} \subset \partial D^3$  as defined below:

$$\begin{aligned} \text{NW} &= \left(0, \frac{-1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right), & \text{NE} &= \left(0, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right), \\ \text{SW} &= \left(0, \frac{-1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}\right), & \text{SE} &= \left(0, \frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}\right). \end{aligned}$$

In general the tangle ball  $B^3$  is not the unit ball in  $\mathbb{R}^3$ . We can modify the definition of a 2-string tangle to consist of a triple  $((B^3, t), \Psi)$ , where  $(B^3, t)$  are as in Definition

3.22 and  $\Psi : (B^3, t) \rightarrow (D^3, t_0)$  is a homeomorphism of pairs taking  $B^3$  to the unit ball  $D^3$ , and anchoring the endpoints of  $t$  to the points NW, NE, SW, SE as in Figure 3.19.

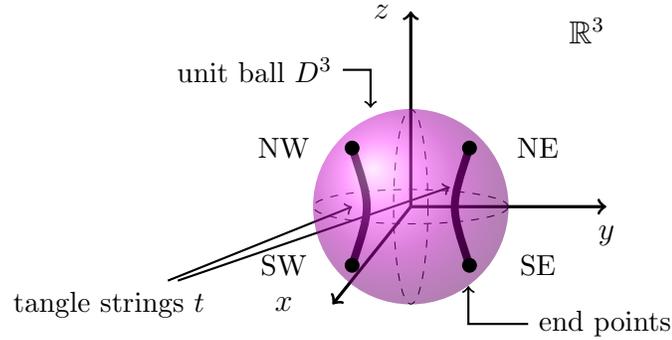


Figure 3.19: Example of a 2-string tangle  $(D^3, t)$  where the endpoints of the two tangle strings  $t$  are anchored to the boundary of the unit ball  $D^3$  at the equatorial points NW, NE, SW, and SE.

This construction provides a framing for the tangle, where the tangle string end points are coplanar.

**Definition 3.23.** A 2-string tangle is said to have **parity 0** if the string which enters at the NW position exits at the NE position, **parity  $\infty$**  if the string which enters at the NW position exits at the SW position, and **parity 1** if the string which enters at the NW position exits at the SE position.

### 3.3.2 Projection of a Tangle

As with knots, tangles are studied through their planar projections. We consider those projections that are regular, and modify them so that gaps in the drawing preserve crossing information. These drawings are referred to as *regular tangle diagrams*, or simply *tangle diagrams*. Figure 3.20 illustrates some examples of tangle diagrams.

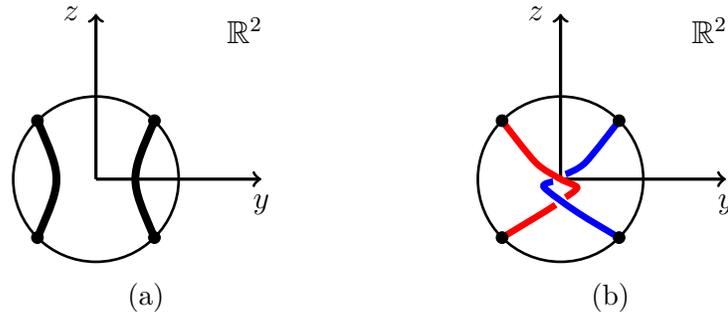


Figure 3.20: Examples of regular tangle diagrams: (a) Tangle diagram for the tangle in Figure 3.19; (b) Tangle diagram for a tangle with 2 crossings.

For simplicity, the coordinate axes are omitted from the tangle diagram. A given tangle can be represented by different tangle diagrams. Just as with knots, Reidemeister moves for tangles are used to determine if two tangle diagrams represent the same tangle.

**Definition 3.24.** Let  $T_1$  and  $T_2$  be two different tangle diagrams. If  $T_1$  can be deformed into  $T_2$  by performing a finite sequence of Reidemeister moves, keeping the tangle strings' endpoints fixed, then  $T_1$  and  $T_2$  are said to be **equal** or **equivalent**

[22].

**Definition 3.25.** Two tangles  $A$  and  $B$  are said to be **ambient isotopic** (isotopic), denoted by  $A \sim B$ , if it is possible to deform  $A$  into  $B$  by fixing the tangle strings' endpoints without moving the strings outside the sphere  $\mathbb{S}^2$ .

### 3.3.3 Types of tangles

There are three types of tangles: *rational*, *locally knotted*, and *prime*. A *rational tangle* is a tangle that can be obtained from a tangle without any crossings (as in Figure 3.20a) by moving the strings' endpoints on the boundary of the ball. A formal treatment of rational tangles is included in Section 3.3.4. A *locally knotted tangle* is a tangle such that there exists a local knot in one of the strings. A *prime tangle* is one that is neither rational nor locally knotted. See Figure 3.21 for examples of these three different tangle types.

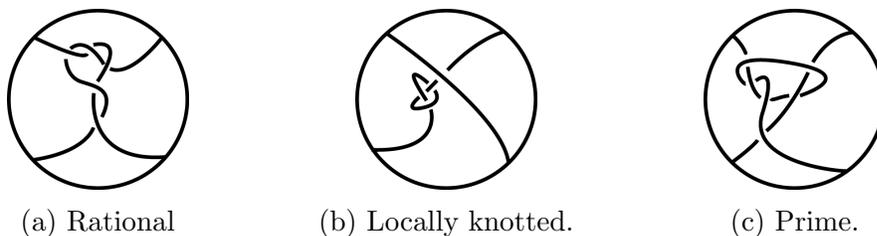


Figure 3.21: Examples of tangles: (a) a rational tangle is a tangle that can be obtained from a tangle, such as the tangle in Figure 3.20a, by moving the strings' endpoints on the boundary of the ball (and not allowing the strings to cross each other or self-intersect), (b) a locally knotted tangle is a tangle such that there exists a local knot in one of the strings, (c) a prime tangle is defined as a tangle that is neither locally knotted nor rational.

### 3.3.4 Rational Tangles

Rational tangles are often used as the building blocks for the knots and links that model the substrates and products of site-specific recombination. Interesting mathematical questions involve determining if the involved tangles are rational or prime (e.g. [29]).

**Definition 3.26.** A tangle is **rational** if there exists a homeomorphism of pairs from  $(B^3, t)$  to the trivial tangle  $(D^2 \times I, \{x, y\} \times I)$ . Here  $D^2$  is the unit 2-ball in  $\mathbb{R}^2$ ,  $\{x, y\}$  are points in the interior of  $D^2$ , and  $I$  is the unit interval.

**Theorem 3.3. *Classification of Rational Tangles*** (Conway, 1970) *There exists a 1-1 correspondence between equivalence classes of rational tangles and the*

extended rational numbers  $\beta/\alpha \in \mathbb{Q} \cup \{1/0 = \infty\}$ , where  $\alpha \in \mathbb{N} \cup \{0\}$ ,  $\beta \in \mathbb{Z}$  and  $\gcd(\alpha, \beta) = 1$  [8].

Figure 3.22 displays the four *exceptional tangles* and their classifying rational numbers. The way in which the strings cross or crossing type convention shown in Figure 3.22 is the convention used in biology and is the convention used throughout this thesis, the opposite convention is used in mathematics.

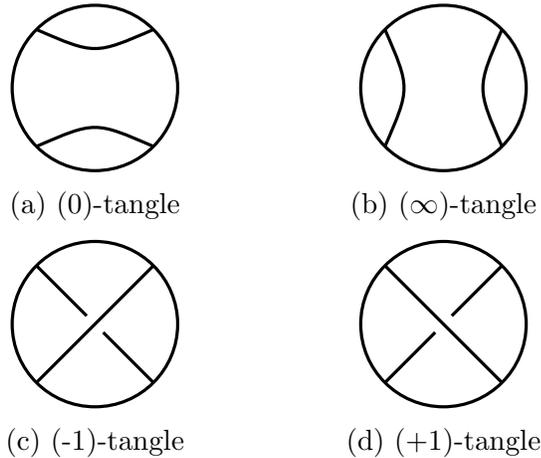


Figure 3.22: Four special types of rational tangles known as the **exceptional tangles**, and denoted as  $(0)$ ,  $(\infty)$ ,  $(-1)$ , and  $(+1)$ .

**Theorem 3.4.** *Two rational tangles are ambient isotopic if and only if they have the same classifying rational number.*

Goldman and Kauffman provided an elegant proof of *Conway's Classification Theorem of Rational Tangles* that uses *twists* to construct a *continued fraction* that

uniquely represents the tangle [12]. See Appendix A for a discussion on continued fractions. From this point on we will refer to rational tangles simply as tangles. Given a starting tangle  $(0)$ , one can obtain a positive horizontal twist with four crossings by twisting the strings horizontally four times in a left-hand way, and a negative twist with four crossings in a right-handed direction as illustrated in Figure 3.23.

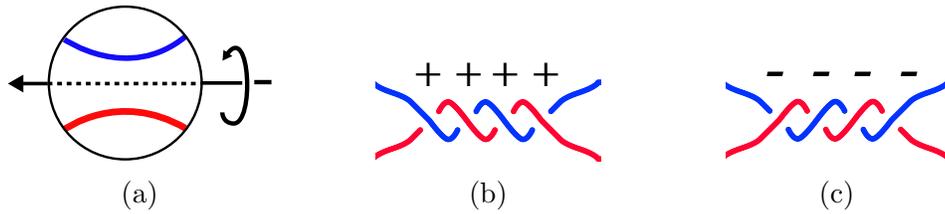


Figure 3.23: Construction of positive and negative horizontal twists with four crossings: (a) starting tangle  $(0)$ , (b) positive twist with four crossings, (c) negative twist with four crossings.

Similarly, given a starting tangle  $(\infty)$ , one can obtain a positive vertical twist with four crossings by twisting the strings vertically four times in a right-hand way, and a negative twist with four crossings in a left-handed direction as shown Figure 3.24.

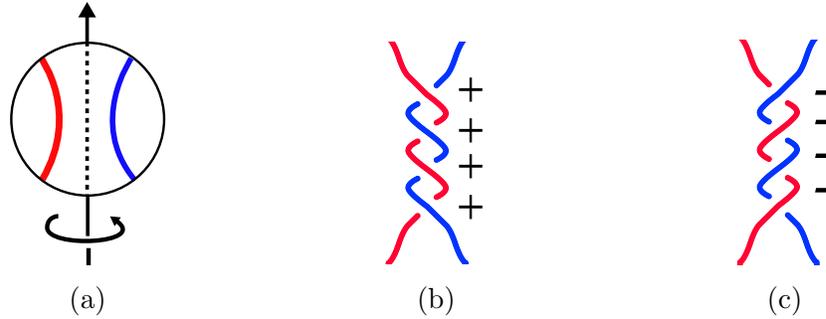


Figure 3.24: Construction of positive and negative vertical twists with four crossings: (a) starting tangle ( $\infty$ ), (b) positive twist with four crossings, (c) negative twist with four crossings.

Given a tangle vector  $T = (a_0, a_1, \dots, a_n)$  each entry  $a_i, 0 \leq i \leq n$ , represents a horizontal or vertical twist, where  $a_n$  is horizontal. The twist type alternates with each entry so that if the tangle vector has even length then  $a_0$  is a vertical twist, and if the tangle vector has odd length then  $a_0$  is a horizontal twist.

Figure 3.25 displays the tangle with classifying rational number  $\frac{16}{9}$ .

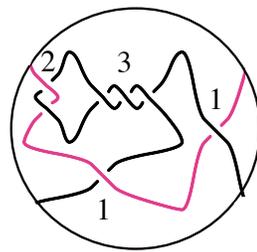


Figure 3.25: Tangle with classifying rational number  $\frac{16}{9}$ . The entries of the tangle vector  $(2, 3, 1, 1)$  describe the horizontal and vertical twists of the tangle.

By Theorem 7.4 in Appendix A, the rational number  $\frac{16}{9}$  can be expressed as a

finite continued fraction as shown below

$$\begin{aligned} \frac{16}{9} &= 1 + \frac{1}{1 + \frac{1}{3 + \frac{1}{2}}} \\ &= [1, 1, 3, 2]. \end{aligned}$$

The partial quotients of the continued fraction is read from right to left to construct the *tangle vector*  $(2, 3, 1, 1)$  (see Figure 3.25). For a more complete introduction to tangle theory and tangle vectors we refer the reader to *Knot Theory and its applications* by Kunio Murasugi [22]. Unlike prime and locally knotted tangles, rational tangles can be simplified to the  $(0)$  or  $(\infty)$  tangle by undoing the horizontal and vertical twists. We are particularly interested in **integral tangles**, that are made up of  $n$  horizontal twists,  $n \in \mathbb{Z}$ .

### 3.3.5 Tangle Calculus

#### Tangle Operations

Tangle addition combines tangles to form a new tangle. There are two types of tangle addition: *horizontal addition* and *vertical addition*, denoted by  $+$  and  $+'$  respectively.

**Definition 3.27.** Let  $T_1$  and  $T_2$  be tangles, **horizontal addition**, denoted by  $T_1 + T_2$ , connects the points NE and SE of  $T_1$  to the points NW and SW of  $T_2$  by unknotted arcs as shown in Figure 3.26.

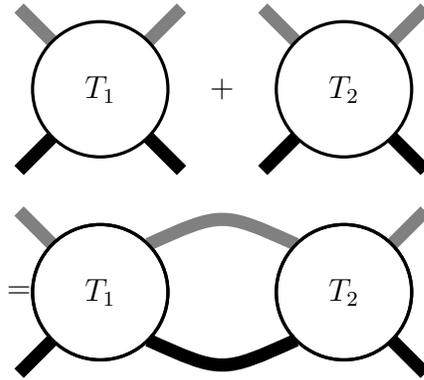


Figure 3.26: Horizontal addition, denoted by  $T_1 + T_2$ , connects the points NE and SE of  $T_1$  to the points NW and SW of  $T_2$  by unknotted arcs.

**Definition 3.28.** Let  $T_1$  and  $T_2$  be tangles, **vertical addition**, denoted by  $T_1 + 'T_2$ , connects the points SW and SE of  $T_1$  to the points NW and NE of  $T_2$  by unknotted arcs as shown in Figure 3.27.

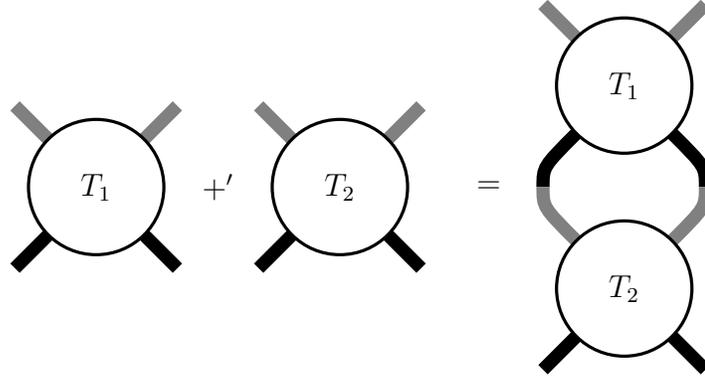


Figure 3.27: Vertical addition, denoted by  $T_1 +' T_2$ , connects the points SW and SE of  $T_1$  to the points NW and NE of  $T_2$  by unknotted arcs.

*Note.* Vertical and horizontal tangle addition are not well defined binary operations in the set of 2-string tangles.

The horizontal addition of tangles  $T_1 = (\infty) = T_2$  does not produce a 2-string tangle as shown in Figure 3.28.

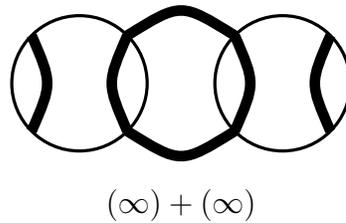


Figure 3.28: The horizontal addition of tangles  $T_1 + T_2$ , where  $T_1 = (\infty)$  and  $T_2 = (\infty)$ , produces an object that is not a 2-string tangle.

Algorithm 3 describes how horizontal and vertical addition piece together the twists of a tangle vector, as discussed in Section 3.3.4, to construct a rational tangle (see Appendix D).

Tangle closures are tangle operations that convert tangles into knots or links. In particular, we are interested in the *numerator* closure, denoted by  $N(T)$ .

**Definition 3.29.** The knot (or 2-component link) obtained from a tangle  $T$  by connecting the points NW and NE, SW and SE by unknotted curves outside of  $B^3$ , is called the **numerator** of  $T$ , and is denoted by  $N(T)$  (see Figure 3.29).

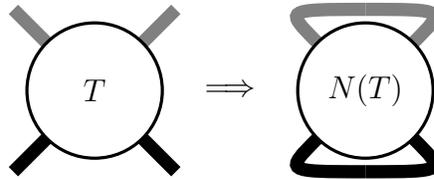


Figure 3.29: The numerator closure of a tangle  $T$ , denoted by  $N(T)$ , connects the points NW and NE, and SW and SE by unknotted arcs outside of  $B^3$ .

*Note.* Tangle closures can be applied to the horizontal sum of an arbitrary number of tangles (see Figures 3.30 and 3.31). This composition of operations is written as  $N(T_1 + T_2 + \cdots + T_n) = K$ , where  $K$  is a knot or link.

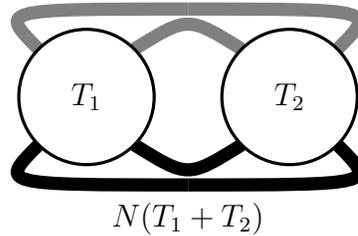
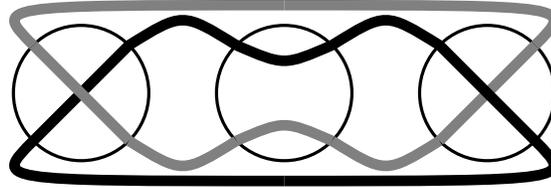


Figure 3.30: Numerator closure of  $T_1 + T_2$ , denoted by  $N(T_1 + T_2)$ .

Figure 3.31 gives an example where  $N((1) + (0) + (1))$  produces the Hopf link  $2_1^2$ .



$$N((1) + (0) + (1)) = 2_1^2$$

Figure 3.31: Example of a link  $K$  (Hopf link  $2_1^2$ ) produced from  $N(T_1 + T_2 + T_3)$ , where  $T_1 = (1)$ ,  $T_2 = (0)$ , and  $T_3 = (1)$ .

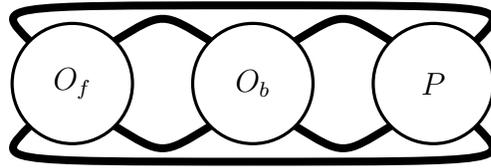
## Chapter 4

# The Tangle Method for Site-Specific Recombination

In 1990, Ernst and Sumners introduced a way to model site-specific recombination with tangle calculus called the *Tangle Method*. For a complete introduction to the Tangle method we refer the reader to *A calculus for rational tangles: applications to DNA recombination* [11].

Recall that during recombination two specific sites are identified and brought together in space. The first step in the process is here referred to as *pre-recombination* [11]. The pre-recombination unbound DNA molecule(s) is called the *substrate*. The post-recombination unbound DNA molecule(s) is called the *product*. The synaptoosome as well as other parts of the *pre-recombination synaptic complex* that are bound to accessory proteins are modeled using tangles. The ball  $B^3$  represents the

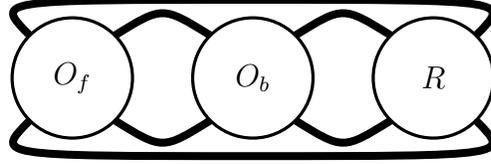
enzyme(s), and the pair of tangle strings  $t$  represent the two DNA strands attached to the enzyme(s). The Tangle Method models a single recombination event as a system of two tangle equations. Equation 4.1 models the pre-recombination synaptic complex, where tangle  $P$  contains the recombination sites and is referred to as the *parental tangle*. The remaining complexity of the pre-recombination synaptic complex is modeled with tangles,  $O_f$  and  $O_b$ , called the *outside free tangle* and the *outside bound tangle*, respectively (see Figure 4.1).



$$N(O_f + O_b + P) = K_1 \quad (4.1)$$

Figure 4.1: The pre-recombination synaptic complex is modeled by Equation 4.1. Here  $P$ , the parental tangle, contains the recombination sites.

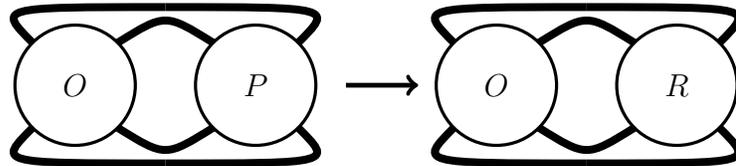
Equation 4.2 models the post-recombination synaptic complex. For recombinases XerCD, it is assumed that the local action of recombination, which takes place entirely in  $P$ , is constant. That is, during recombination, the spatial configuration of the pre-recombination synaptic complex remains fixed outside of  $P$ . After recombination, the tangle  $P$  is replaced with tangle  $R$ , called the *recombinant tangle* (see Figure 4.2).



$$N(O_f + O_b + R) = K_2 \quad (4.2)$$

Figure 4.2: The post recombination synaptic complex is modeled by Equation 4.2, where  $R$  replaces  $P$ .  $R$  is called the recombinant tangle.

Sometimes we consider the tangle  $O = O_f + O_b$  to write a system of tangle equations (see Figure 4.3).



$$N(O + P) = K_1 \quad N(O + R) = K_2 \quad (4.3)$$

Figure 4.3: The Tangle Method can be modified so that the pre-recombination synaptic complex is modeled by  $N(O + P)$ , and the post-recombination synaptic complex is modeled by  $N(O + R)$ , where  $O = O_f + O_b$ .

Here it is assumed that  $P = (0)$ , that leaves two unknown variables, namely  $O$  and  $R$ . Since Xer recombination results in a single crossing, we are only interested in solutions  $R = (\pm 1)$  or  $(\infty)$ , depending on the local alignment of the recombination sites. The solutions for  $O$  reveal information about the spatial configuration of the substrate prior to recombination, which is the primary focus of this research and is discussed in Section 6.

## Chapter 5

# Tetrahedral Tangles: Embedding a Tangle Inside the Tetrahedron

### 5.1 Groups

It has been shown for site-specific recombinase Cre that the recombination sites loxP are not coplanar during recombination [28]. Hence the framing for a tangle, as defined in Section 3.3.1, does not reflect the geometry of the recombination site. In 2012, Wono relaxed this assumption and investigated a modified version of the *Tangle Method* by entrapping a tangle inside of a regular tetrahedron, and used the *tetrahedral group* to show that different planar projections can correspond to different tangle solutions of a given recombination system [32]. Here we provide the necessary background from group theory to define the tetrahedral group.

**Definition 5.1.** A **binary operation**, denoted by  $\star$ , on a non-empty set  $G$  is a function  $\star : G \times G \rightarrow G$ . For any  $a, b \in G$  we shall write  $a \star b$  for  $\star(a, b)$ .

**Definition 5.2.** A **group** is a set  $G$  together with a binary operation  $\star$ , written as  $(G, \star)$ , that satisfies the following four axioms:

**G1:** Closure.  $a \star b \in G$  for all  $a, b \in G$ .

**G2:** Associativity.  $(a \star b) \star c = a \star (b \star c)$  for all  $a, b, c \in G$ .

**G3:** Identity. There exists an element  $e \in G$  such that  $e \star a = a \star e = a$  for every  $a \in G$ .

**G4:** Inverse. For every  $a \in G$  there exists  $a^{-1} \in G$  such that  $a^{-1} \star a = a \star a^{-1} = e$ .

The group operation  $\star$  is often referred to as the *product*.

**Example 5.1.**  $G = (\mathbb{Z}, +)$  is a group. Here the identity element is  $e = 0$ .

**Definition 5.3.** A nonempty subset  $H$  of a group  $G$  is called a **subgroup** of  $G$ , if  $H$  is itself a group under the product of  $G$ .

**Example 5.2.** Let  $H$  be the set of even integers, then  $H$  is a subgroup of  $G = (\mathbb{Z}, +)$ .

Notation: To make the notation more concise the binary operation symbol may be omitted, and for any elements  $a, b \in G$  the operation  $a \star b$  can be written simply as  $ab$ . Since  $\star$  is associative,  $a \star (b \star c)$  can be written as  $abc$ . The product  $\underbrace{aaa \cdots a}_{n \text{ terms}}$  is denoted by  $a^n$ .

**Lemma 5.1.** *Let  $G$  be a group. Then:*

- (i) The identity element  $e$  is unique.*
- (ii) Every  $a \in G$  has a unique inverse  $a^{-1} \in G$ .*
- (iii) If  $a \in G$ ,  $(a^{-1})^{-1} = a$ .*
- (iv) For  $a, b \in G$ ,  $(ab)^{-1} = b^{-1}a^{-1}$ .*

**Definition 5.4.** If a group  $G$  has  $n$  elements for some  $n \in \mathbb{N}$ ,  $G$  is called a **finite group of order  $n$** .

For the purpose of this study we are interested in finite groups and in particular the group of permutations.

**Corollary 5.2.** *If  $G$  is a finite group and  $H$  is a nonempty subset of  $G$  closed under the product of  $G$ , then  $H$  is a subgroup of  $G$ .*

**Definition 5.5.** A bijective map from a set  $\Omega$  of  $n$  distinct elements to itself is a **permutation** of degree  $n$ , and is denoted by  $\sigma$ .

A visual representation of a permutation  $\sigma$  on a set  $\Omega = \{1, 2, 3, 4\}$  is shown in Figure 5.1.

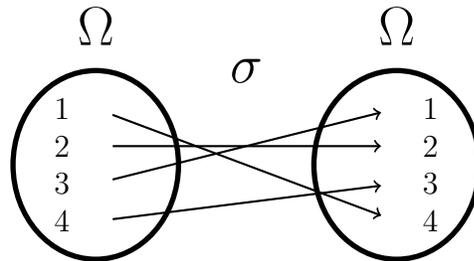


Figure 5.1: Visual representation of a permutation  $\sigma$  on a set  $\Omega = \{1, 2, 3, 4\}$

There are several ways to represent the permutation  $\sigma$  shown in Figure 5.1. Below is the *two-line* notation for  $\sigma$  where the top row of elements are mapped to the bottom row of elements.

$$\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 2 & 1 & 3 \end{pmatrix}$$

Note that the order in which the pair of top and bottom elements are listed does not change the permutation.

$$\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 2 & 1 & 3 \end{pmatrix} = \begin{pmatrix} 4 & 1 & 2 & 3 \\ 3 & 4 & 2 & 1 \end{pmatrix} = \begin{pmatrix} 3 & 4 & 1 & 2 \\ 1 & 3 & 4 & 2 \end{pmatrix} = \dots$$

Let  $\beta$  be the following permutation

$$\beta = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 4 & 3 \end{pmatrix},$$

then the composition of permutations  $\sigma$  and  $\beta$  is written as follows:

$$\sigma \circ \beta = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 2 & 1 & 3 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 4 & 3 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 2 & 3 & 1 \end{pmatrix}.$$

For permutations on a set of natural numbers  $\{1, 2, \dots, n\}$ , the standard convention is to write the top line of elements in ascending order. There is a more abbreviated version of this permutation notation called *cycle notation*. The cycle  $(k_1, k_2, \dots, k_m)$  sends  $k_i$  to  $k_{i+1}$  for  $1 \leq i \leq m - 1$  and  $k_m$  to  $k_1$ . Any elements that remain fixed under  $\sigma$  are omitted from the cycle. The cycle notation for the permutation  $\sigma$  in Figure 5.1 is  $\sigma = (143)$ . Figure 5.2 gives a visual of the cycle notation for the permutation  $\sigma = (143)$ .

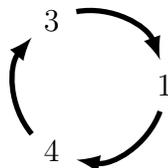


Figure 5.2: Illustration of the permutation  $\sigma = (143)$

Similarly, with the previous notation, it does not matter the order of the elements

as long as the cycle is preserved. For instance,

$$\sigma = (143) = (314) = (431).$$

More complex permutations are grouped into disjoint cycles as shown below.

$$(k_1 k_2 \dots k_{m_1})(k_{m_1+1} k_{m_1+2} \dots k_{m_2}) \dots (k_{m_l+1} \dots k_{m_j})$$

The set of permutations on a set of  $n$  elements forms a group under the composition function  $\circ$ . When the set consists of the elements  $\{1, 2, 3, \dots, n\}$  this group is called the **symmetric group**, and is denoted by  $S_n$ . The symmetric group  $S_n$  has  $n!$  elements. Wono used a subgroup of  $S_4$  to study the planar projections of a tangle embedded inside of a regular tetrahedron [32]. Advancements of this work is discussed in the following section.

Consider the class of groups whose elements are the symmetries of geometric objects. A symmetry is a rigid motion that takes a copy of the object and moves the structure in space such that it perfectly fits the original object. The following definitions for geometric objects are borrowed from *Geometry: Euclid and Beyond* by Robin Hartshorne [14].

**Definition 5.6.** A **polygon** is a closed planar figure bounded by straight line segments as sides. An  **$n$ -gon** is a polygon with  $n$  sides (see Figure 5.3). A **regular polygon** is an equilateral and equiangular polygon.

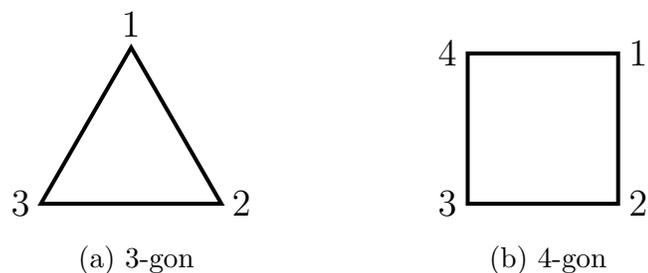


Figure 5.3: Examples of regular polygons with clockwise numbered vertices: (a) 3-gon, and (b) 4-gon.

In the case of a regular  $n$ -gon embedded in  $\mathbb{R}^3$  the rigid motions are maps in  $\mathbb{R}^3$  taking the  $n$ -gon to itself. We start by labeling each vertex of the  $n$ -gon from 1 to  $n$  in a clockwise fashion as shown in Figure 5.3. A symmetry of an  $n$ -gon is uniquely determined by a permutation  $\sigma$  on  $\{1, \dots, n\}$ . In the case of the regular 3-gon, the symmetries can be described as rotations or reflections about two types of axes (see Figure 5.4).

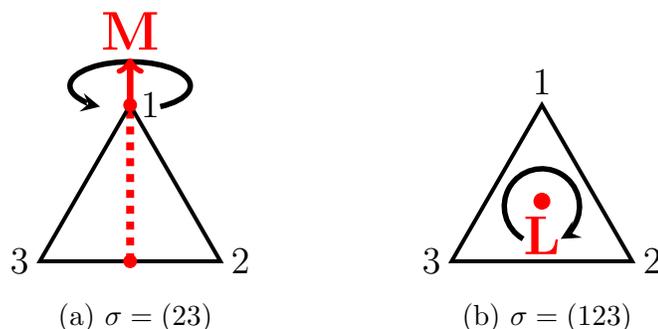


Figure 5.4: The regular 3-gon (or equilateral triangle) has two types of axes of symmetry. (a) The equilateral triangle has three axes of symmetry. Each of these passes through the midpoint of an edge and the opposite vertex. For example, the permutation  $(23)$  corresponds to a reflection about the M-axis in the figure. (b) The axis of rotational symmetry that passes through the center of the 3-gon. In this case from the reader's perspective this axis (denoted by L) would pass through the paper.

The reflections correspond to permutations  $(12)$ ,  $(23)$ , and  $(13)$ . Rotating clockwise about the L-axis by  $0^\circ(360^\circ)$ ,  $120^\circ$ , and  $240^\circ$  corresponds to the permutations  $(e)$ ,  $(123)$ , and  $(132)$ , respectively. The *multiplication table* is used to show that the six symmetries  $\{(e), (123), (132), (12), (23), (13)\}$  satisfy the four axioms (**G1**, **G2**, **G3**, **G4**) that define a group, as shown in Figure 5.5.

**Definition 5.7.** Let  $G = \{g_1, g_2, \dots, g_n\}$  be a finite group with  $g_1 = (e)$ . The **multiplication table** of  $G$  is the  $n \times n$  matrix whose  $i, j$  entry is the group element  $g_i g_j$ .

|              |            |              |              |             |             |             |
|--------------|------------|--------------|--------------|-------------|-------------|-------------|
|              | <b>(e)</b> | <b>(132)</b> | <b>(123)</b> | <b>(23)</b> | <b>(12)</b> | <b>(13)</b> |
| <b>(e)</b>   | (e)        | (132)        | (123)        | (23)        | (12)        | (13)        |
| <b>(123)</b> | (123)      | (e)          | (132)        | (13)        | (23)        | (12)        |
| <b>(132)</b> | (132)      | (123)        | (e)          | (12)        | (13)        | (23)        |
| <b>(23)</b>  | (23)       | (13)         | (12)         | (e)         | (123)       | (132)       |
| <b>(12)</b>  | (12)       | (23)         | (13)         | (132)       | (e)         | (123)       |
| <b>(13)</b>  | (13)       | (12)         | (23)         | (123)       | (132)       | (e)         |

Figure 5.5: A multiplication table is used to prove that the symmetries or rigid motions of a regular 3-gon satisfy the four axioms (**G1**, **G2**, **G3**, **G4**) that define a group.

Groups can also be generated from the symmetries of solid geometric objects. Here we are interested in the *tetrahedral group*.

**Definition 5.8.** A **solid** is an enclosed portion of space bounded by plane (flat) and curved surfaces. A **polyhedron** is the surface of a solid figure bounded by plane polygons. A polyhedron is **convex** if for any two points on the surface of the polyhedron, the line segment between those two points is entirely contained in the solid figure bounded by the polyhedron. A **regular** polyhedron is a convex polyhedron, with all of its faces congruent regular polygons, and with the same number of faces meeting at each vertex.

An example of a regular polyhedron is the *regular tetrahedron*.

**Definition 5.9.** A **regular tetrahedron** (tetrahedron) is a regular polyhedron with four faces that are all equilateral triangles (see Figure 5.6).

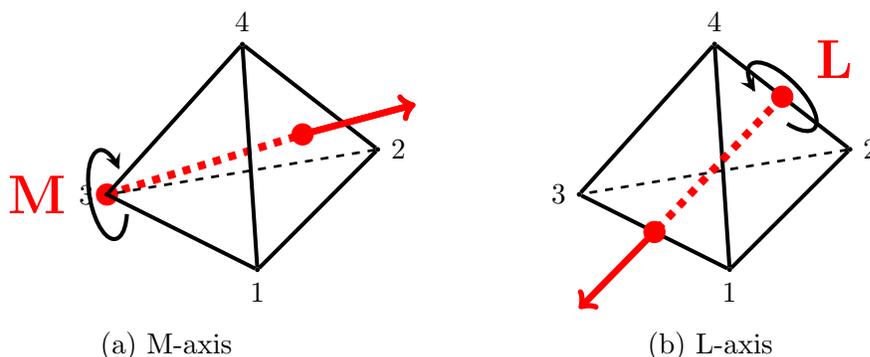


Figure 5.6: The regular tetrahedron has two types of axes of symmetry. (a) The line that passes through a vertex and the midpoint of the opposite face. For example, the permutation  $(142)$  corresponds to a  $120^\circ$  rotation about the M-axis in the figure. (b) The axis of rotational symmetry that passes through the midpoint of an edge and the midpoint of the opposite edge. For example, the permutation  $(24)(13)$  corresponds to a  $180^\circ$  rotation about the L-axis in the figure.

From this point on we omit the prefix regular. The symmetries of a tetrahedron are obtained by rigid rotations on two types of axes of symmetry. The axes of rotational symmetry that pass through a vertex and the midpoint of the opposite face, correspond to the permutations  $(123)$ ,  $(132)$ ,  $(124)$ ,  $(142)$ ,  $(143)$ ,  $(134)$ ,  $(243)$ , and  $(234)$ . The axes of rotational symmetry that pass through the midpoints of opposite edges correspond to permutations  $(13)(24)$ ,  $(12)(34)$ , and  $(14)(23)$ . See Figure 5.6 for examples of axes of rotational symmetries.

The symmetries of the tetrahedron define a group, called the **tetrahedral group**. The dimensions of the multiplication table for the tetrahedral group is  $(12 \times 12)$ . Figure 5.7 displays the twelve elements of the tetrahedral group  $\{ (234), (243), (134), (143), (124), (142), (123), (132), (12)(34), (14)(23), (13)(24), (e) \}$ .

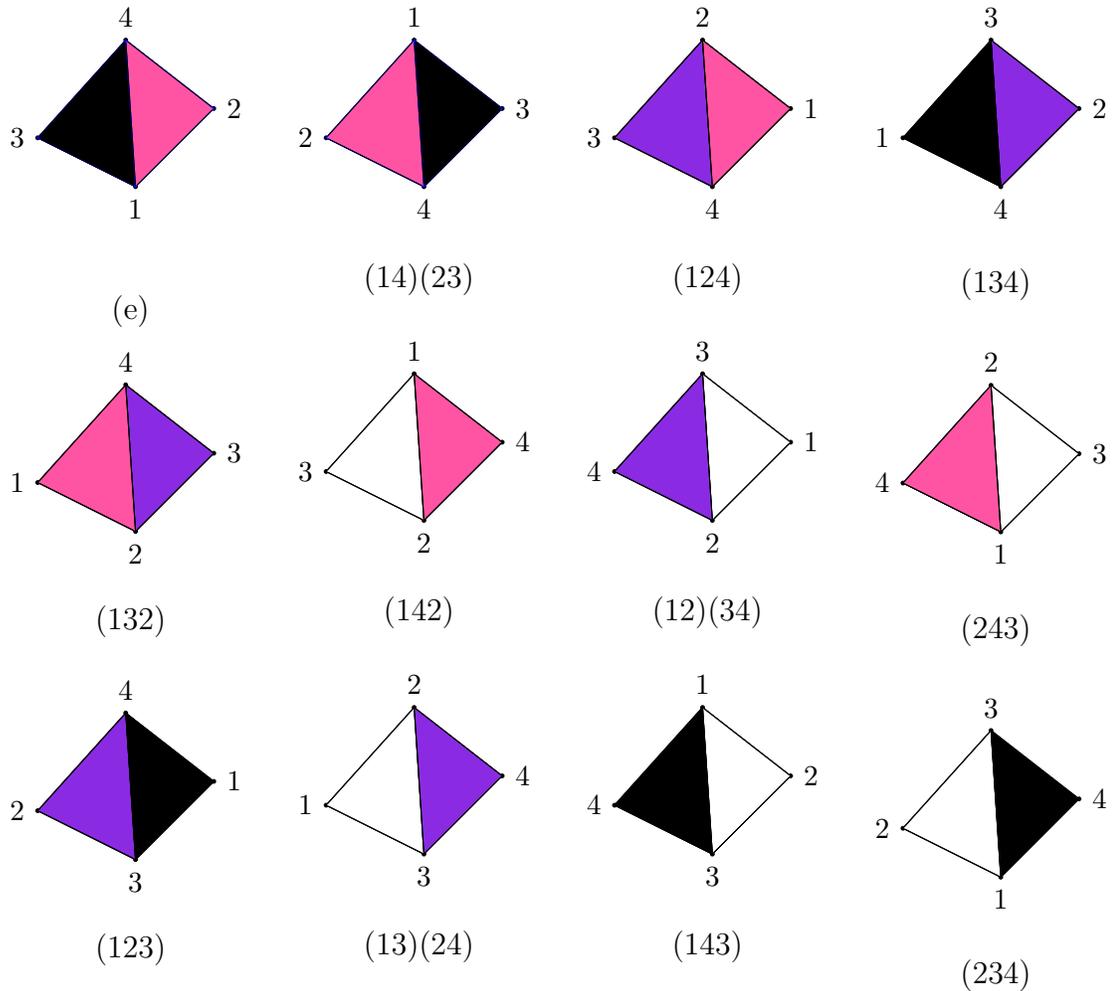


Figure 5.7: The twelve elements of the tetrahedral group  $G = \{ (234), (243), (134), (143), (124), (142), (123), (132), (12)(34), (14)(23), (13)(24), (e) \}$  generated by the rotations about the M and L axes. Each face is painted a different color {pink, black, violet, white}.

In the next section we explain how we use the tetrahedral group to study the different

planar projections of a tangle embedded inside of a tetrahedron.

## 5.2 Tetrahedral Tangles and Computer Implementation

In 2005, Vazquez *et. al.* showed that the three possible solutions for the XerCD action on an unknotted circular DNA molecule could be interpreted as different projections of a single 3-dimensional object [21]. In 2013, Shimokawa *et. al.* used the crystal structure data of Cre-DNA as a model for XerCD, and showed that the off-planar geometry of the recombination sites can give rise to different projections where the sites appear parallel and anti-parallel [19]. In an effort to provide a more realistic model of site-specific recombination and to give tangles a more interesting geometry, in 2012, Wono redefined the 2-string tangle  $(B^3, t)$  so that each endpoint of  $t$  lies on one of the vertices of a tetrahedron [32]. As part of the research in this thesis, we define an embedding for the tangle strings inside of a tetrahedron, and develop a computer visualization tool called *Tangle3D* that automates and extends Wono's work (see Appendix B).

As discussed in Section 3.3.4, rational tangles can be represented by tangle vectors. In 2002, Saka and Vazquez developed a program called *TangleSolve* that implements the tangle method and provides a tool to solve and visualize systems of tangle equations [25]. In particular, given a tangle vector, *TangleSolve* constructs a tangle diagram. We here implement a new algorithm in Java for *Tangle-*

*Solve.* In this new implementation a tangle vector yields a tangle diagram drawn inside of the unit square, and also an embedding of the corresponding 3D tangle inside of a tetrahedron. We call this 3-dimensional object a *tetrahedral tangle* (see Figure 5.8). The source code for *Tangle3D* and the newest version of *TangleSolve* (see Appendix B) are available online through the GitHub code host at <https://github.com/morenocrista/TangleSolve>. First, we describe the approach used to construct the tangle diagram inside of the unit square.

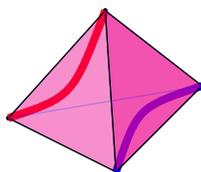


Figure 5.8: Tetrahedral tangles allow us to consider a non-planar geometry for the synaptosome in a site-specific recombination reaction.

Recall from Section 3.3.4 that a 3D tangle is constructed from the tangle vector  $T = (a_0, \dots, a_n)$  by connecting alternating horizontal and vertical twists. In general, the tangle is constructed by the summation  $(((((\dots) + ' a_{n-3}) + a_{n-2}) + ' a_{n-1}) + a_n)$ , where  $+$  and  $+'$  denote the horizontal and vertical sums of tangles (Section 3.3.5). The tangle diagram corresponding to the tangle vector  $(a_0, a_1, a_2, a_3) = (3, 5, 4, 2)$  is constructed by performing the tangle sum  $((3 + 5) + ' 4) + 2$ , as shown in Figure 5.9.

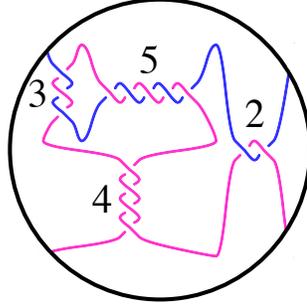


Figure 5.9: Tangle diagram for tangle vector  $(3, 5, 4, 2)$ .

Given a tangle vector, *Tangle3D* sketches the corresponding tangle diagram by first constructing each twist  $a_i, 0 \leq i \leq n$ , inside of a rectangle, where the dimensions are determined by the number of crossings and the dimensions of the rectangles for the previous twists. In Appendix D we include pseudocode for Algorithm 4, which describes how the dimensions of the rectangle for each twist are computed. This algorithm is implemented in the *Tangle3D* class.

We will refer to the height and width of the twist to mean the height and width of the rectangle for the twist. In general, the height of a vertical twist is given by the number of crossings, and the width is the sum of the widths of the previous two twists. The width of a horizontal twist is given by the number of crossings and the height is the sum of the heights of the previous two twists. Here, we give an example to illustrate this construction. Figure 5.10 displays the dimensions of the twists for the tangle  $(3, 5, 4, 2)$  shown in Figure 5.9.

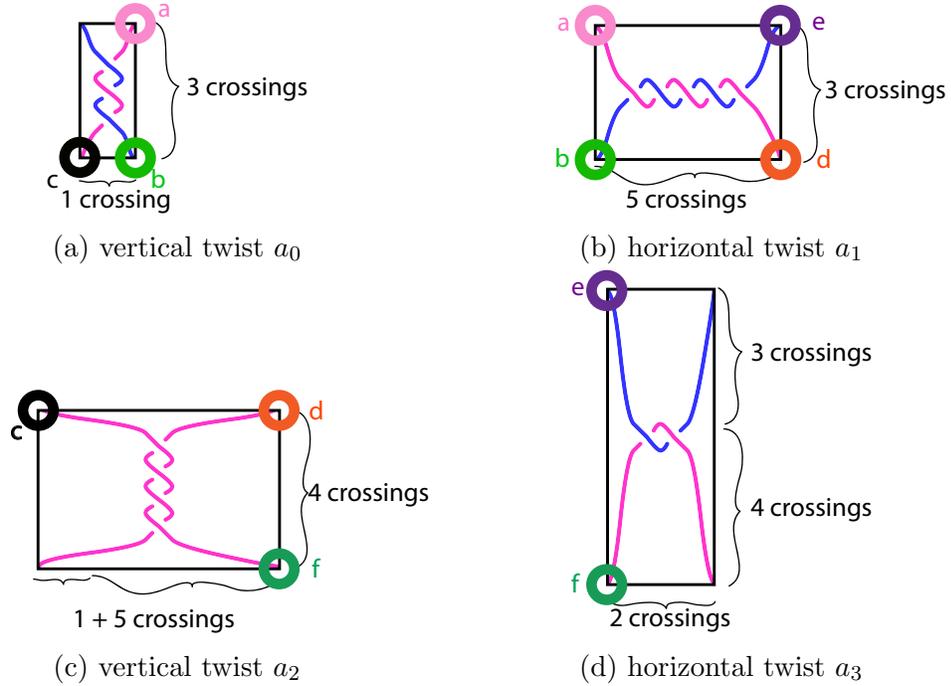


Figure 5.10: Dimensions for each horizontal and vertical twist in the tangle diagram corresponding to the vector  $(a_0, a_1, a_2, a_3) = (3, 5, 4, 2)$ , shown in Figure 5.9.

For each twist  $Tangle3D$  computes the  $(x, y)$  coordinates for the string(s) that enter and exit the twist, we refer the reader to the `getVerticesLocal` algorithm written in the  $Tangle3D$  class. Here we use an example to explain this algorithm. Let  $n$  be the number of crossings for a horizontal twist, and  $c$  be the desired crossing size. We construct a sine curve with the parametric equations

$$x = \frac{ct}{2\pi}, \quad (5.1)$$

$$y = c \sin\left(t - \frac{\pi}{2}\right) \quad (5.2)$$

where  $\Delta t = \frac{\pi}{r}$ ,  $t = i\Delta t$ , and  $i$  takes on all integer values ranging from 0 to  $nr$ . Here  $r$  is the resolution or number of points for the time step  $\Delta t$ . In the case of a vertical twist, we apply a  $90^\circ$  rotation.

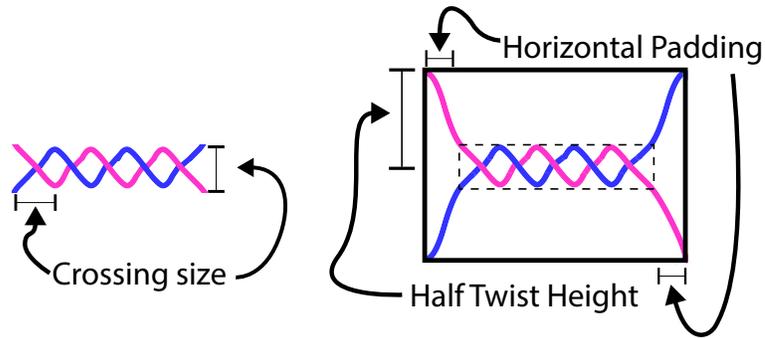


Figure 5.11: The sine curves for the horizontal twist  $a_1$  are given by the equations 5.1 and 5.2 with parameter values  $n = 5$ ,  $c = 10$ ,  $r = 100$ , and the horizontal twist padding is 10.

*Tangle3D* then connects the twists by tracing each tangle string through the corners of the twists. Given a twist corner for which a string enters or exits, the other twist corner that it exits or enters is determined by the number of crossings and the twist type. This procedure is described in Algorithm 1.

---

**Algorithm 1** getEndCorner
 

---

```

1: procedure ENDCORNER(startCorner, twistOrientation, numberCrossings)
2:   endCorner  $\leftarrow$  NULL
3:   if numberCrossings mod 2 = 0 then ▷ even number of crossings
4:     case horizontal
5:       endCorner  $\leftarrow$  mirrorHorizontal
6:     case vertical
7:       endCorner  $\leftarrow$  mirrorVertical
8:   else
9:     endCorner  $\leftarrow$  mirrorDiagonal
10:  end if
11:  return endCorner
12: end procedure

```

---

To connect the twists in Figure 5.10 we refer the reader to the method called *getNextTwistAndTwistCorner* in the *Tangle3D* class. In this example, *Tangle3D* starts at the top left corner of twist  $a_0$ , and by Algorithm 1, traverses the string to the corner labeled b in the figure. This corner is then connected to the corner labeled b of twist  $a_1$ . Then, the string is traversed from corner b to corner e in twist  $a_1$ . This corner is then connected to corner e of twist  $a_3$ . In a similar fashion, the corners are connected for the second string. This procedure is repeated until all corners are identified as illustrated in Figure 5.12.

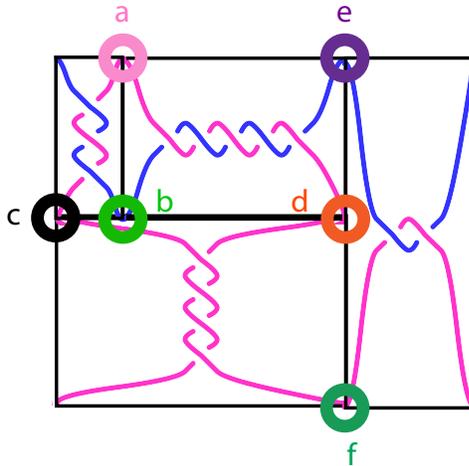


Figure 5.12: Identification of the twist corners.

*Tangle3D* then scales the tangle diagram so that it is bounded by a unit square (see Figure 5.13).

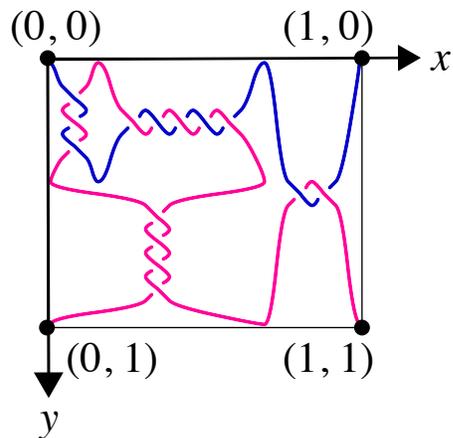


Figure 5.13: Visual representation of the tangle diagram for  $(3, 5, 4, 2)$  embedded inside of a unit square. The  $y$ -axis is reversed since the computer-graphics program we use orients the origin at the top left hand corner of the screen.

For each tangle string, *Tangle3D* computes the  $(x, y)$  coordinates in the unit square. These sets of points are *List* objects of type *Point3f* in the *Tangle3D* class.

They are called *firstStrandVertices* and *secondStrandVertices*, we will refer to these sets as  $t_1$  and  $t_2$ . Now let  $A, B, C,$  and  $D$  be the vertices of a tetrahedron  $\Delta$ . Our goal is to map the sets  $t_1$  and  $t_2$  to points inside of  $\Delta$ , where the points  $(0, 0), (0, 1), (1, 0),$  and  $(1, 1)$  are mapped to the four vertices of  $\Delta$ . Since there are 24 possible ways of mapping these four points to the vertices of the tetrahedron, we define a set of 24 mappings  $\Phi = \{\varphi_1, \varphi_2, \dots, \varphi_{24}\}$ . Let  $\varphi_1 : t_1 \cup t_2 \rightarrow \mathbb{R}^3$  be an element of  $\Phi$  which maps the points  $(0, 0), (0, 1), (1, 1),$  and  $(1, 0)$  to the vertices  $A, B, C,$  and  $D$ , respectively, and the two strings in the diagram to two strings enclosed by the tetrahedron (see Figure 5.14).

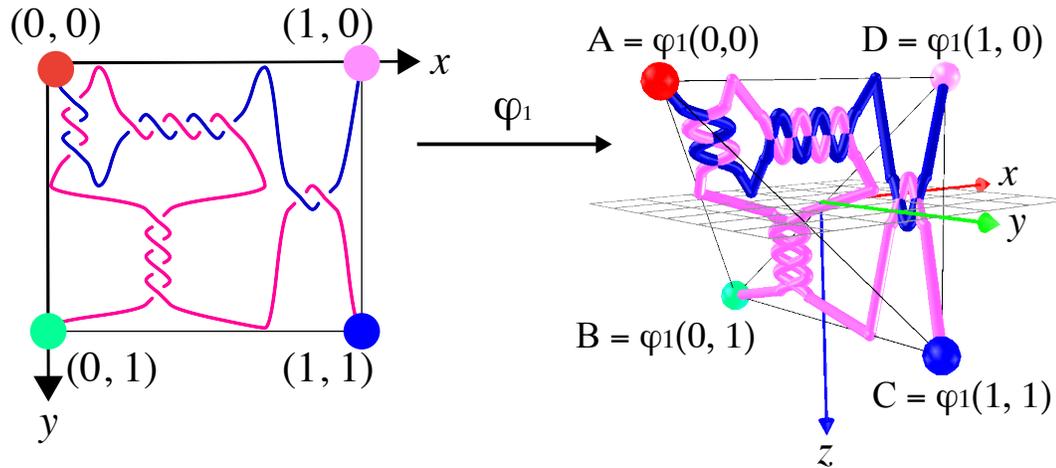


Figure 5.14: Visual representation of the mapping  $\varphi_1 : t_1 \cup t_2 \rightarrow \mathbb{R}^3$  that takes the points for each tangle string bounded by the unit square to points inside of the tetrahedron  $\Delta$ .

Let  $(x, y)$  be a point in  $t_1 \cup t_2$ . Recall from vector calculus that if  $\vec{u} = (u_1, u_2, u_3)$  and  $\vec{v} = (v_1, v_2, v_3)$  are vectors in  $\mathbb{R}^3$ , then the **sum**  $\vec{u} + \vec{v} = (u_1 + v_1, u_2 + v_2, u_3 + v_3)$  and **scalar multiplication**  $a * \vec{v} = (av_1, av_2, av_3)$ , where  $a$  is a scalar. We let  $\varphi_1$

be given by Equation 5.3

$$\varphi_1(x, y) = w_a(x, y)\vec{A} + w_b(x, y)\vec{B} + w_c(x, y)\vec{C} + w_d(x, y)\vec{D}, \quad (5.3)$$

so that the following holds:

$$\left\{ \begin{array}{l} w_a = 1 \text{ and } w_b = w_c = w_d = 0, \quad \text{if } (x, y) = (0, 0), \\ w_b = 1 \text{ and } w_a = w_c = w_d = 0, \quad \text{if } (x, y) = (0, 1), \\ w_c = 1 \text{ and } w_a = w_b = w_d = 0, \quad \text{if } (x, y) = (1, 1), \\ w_d = 1 \text{ and } w_a = w_b = w_c = 0, \quad \text{if } (x, y) = (1, 0). \end{array} \right.$$

The conditions for functions  $w_a$ ,  $w_b$ ,  $w_c$ , and  $w_d$  are met by the coefficients given in Equation 5.4.

$$\varphi_1(x, y) = (1 - x)(1 - y)\vec{A} + y(1 - x)\vec{B} + xy\vec{C} + x(1 - y)\vec{D} \quad (5.4)$$

For the computer implementation of this mapping, we let  $A = (-1, 0 - 1/\sqrt{2})$ ,  $B = (0, -1, 1/\sqrt{2})$ ,  $C = (0, 1, 1/\sqrt{2})$ , and  $D = (1, 0, -1/\sqrt{2})$ . To recreate the crossings, for each twist *Tangle3D* defines a *helix* with  $x$  and  $y$  as defined in Equations 5.1 and 5.2, and  $z = \pm \cos(t - \pi/2)$  for negative and positive crossings, respectively (see Figure 5.15).

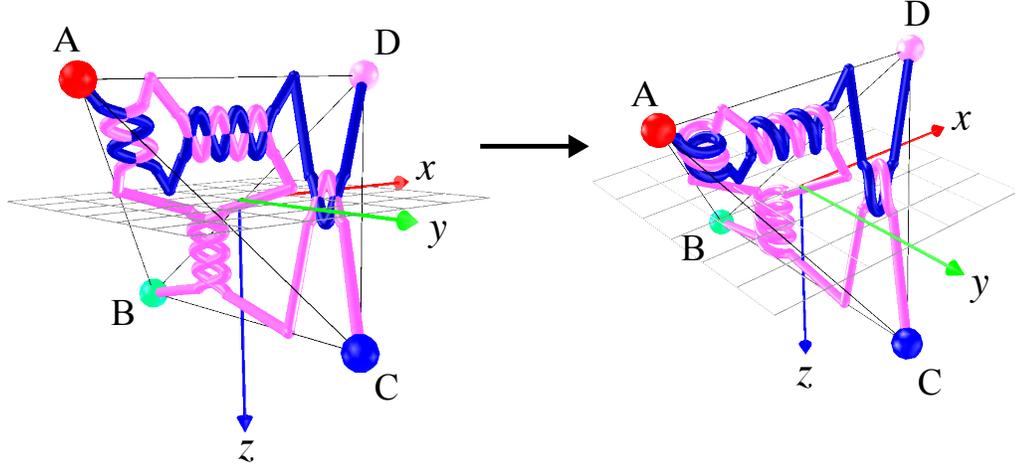


Figure 5.15: Tangle strings mapped inside of the tetrahedron where the crossings are constructed with helices.

Since every map in  $\Phi$  can be defined as a permutation of the vertices  $A$ ,  $B$ ,  $C$ , and  $D$ , we can modify Equation 5.4 by permuting the vectors  $\vec{A}$ ,  $\vec{B}$ ,  $\vec{C}$ , and  $\vec{D}$  to obtain all 24 mappings. For example, we can define the mapping  $\varphi_2$  in Equation 5.5, by permuting the vectors  $\vec{B}$  and  $\vec{C}$ . In comparison to  $\varphi_1$ ,  $\varphi_2$  exchanges the endpoints for the two tangle strings connected to vertices  $B$  and  $C$ . Figure 5.16 displays each of these embeddings generated by *Tangle3D*.

$$\varphi_2(x, y) = (1 - x)(1 - y)\vec{A} + y(1 - x)\vec{C} + xy\vec{B} + x(1 - y)\vec{D} \quad (5.5)$$

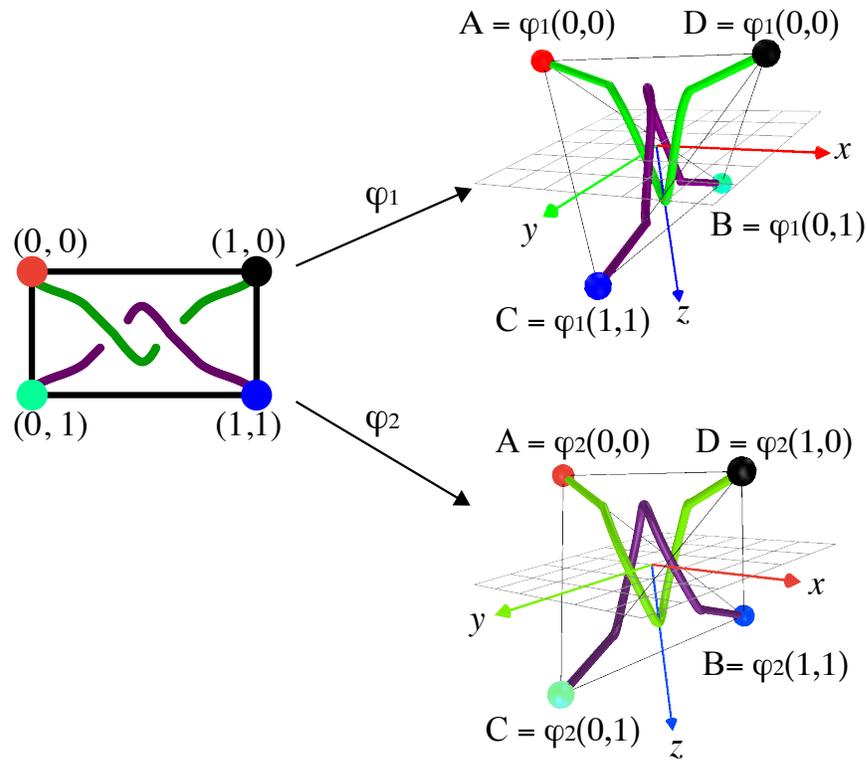


Figure 5.16: Visual representation of the mappings which takes the tangle (2) drawn in the unit square to the tetrahedron  $\Delta$  via  $\varphi_1$  and  $\varphi_2$ .

For this research we study embeddings of integral tangles, as discussed in Section 3.3.4. Now we can formally define a tetrahedral tangle.

**Definition 5.10.** A **tetrahedral tangle** is an embedding of a tangle  $(B, t)$  inside of a regular tetrahedron  $\Delta$ , where the  $x$  and  $y$  coordinates are determined by the mapping  $\varphi$  that maps the tangle diagram drawn in a unit square to  $\Delta$ . We denote the tetrahedral tangle by  $\Delta_{(a_0, \dots, a_n)}^\varphi$ .

### 5.3 Tetrahedral Tangle Projections

Wono studied physical models of a tangle inside of a tetrahedron through their *planar projections* obtained by using the rotational symmetries of the tetrahedron. *Tangle3D* is a digital implementation that expedites this process and eliminates human error. It also provides a visualization of tangle solutions computed by *TangleSolve* when interpreted as tetrahedral tangles, and builds on the work of Wono and Vazquez [32]. *Tangle3D* uses the tetrahedral group defined in Section 5.1 to generate the projections of a tetrahedral tangle, referred to as *tetrahedral tangle projections* or simply as *projections*.

It should be noted that from a computational perspective we study tetrahedral tangles that are 3D graphics, or objects that are created on a 3-dimensional stage, where the current view is derived from the camera angle and light sources, similar to the real world. Figure 5.17 displays the computer implementation of the tetrahedral tangle  $\Delta_{(2)}^{\varphi_1}$ .

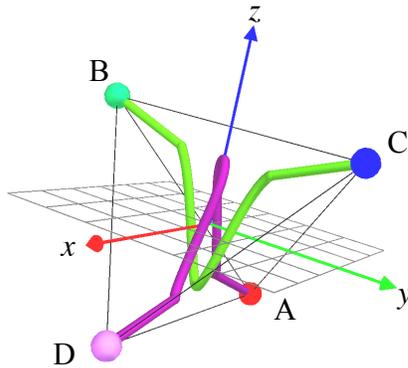


Figure 5.17: Computer implementation of tetrahedral tangle  $\Delta_{(2)}^{\varphi_1}$ .

To perform rotations on these objects, instead of recreating the object, *Tangle3D* changes the view, and moves the camera in space. Consider the tetrahedral tangle  $\Delta_{(2)}^{\varphi_1}$ , where vertices A, B, C, and D are labeled 1, 2, 3, and 4, respectively. Recall the 12 elements of the tetrahedral group from Section 5.1. We will refer to the starting view of the tetrahedral tangle as the projection generated by the transformation (e). From this starting view the camera is moved in space to that the view gives a projection that is equivalent to a (123) transformation, see Figure 5.18.

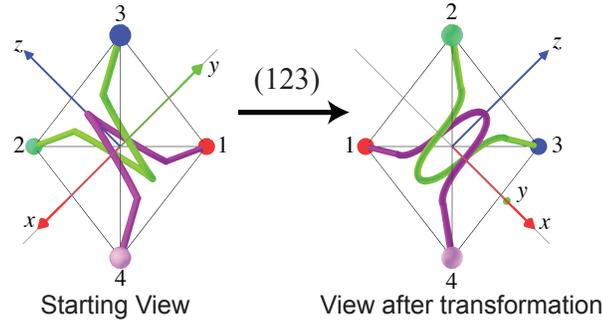


Figure 5.18: Projections of  $\Delta_{(2)}^{\varphi_1}$ , where the starting view is changed by a transformation equivalent to the (123) transformation.

Figures 5.19 and 5.20 display the projections of tetrahedral tangles  $\Delta_{(2)}^{\varphi_1}$  and  $\Delta_{(2)}^{\varphi_2}$ , respectively, generated by the twelve rotational symmetries of the tetrahedron. In Section 5.4, we reproduce Wono's results for the tangle (2) embedded inside of the tetrahedron, and define equivalence classes for the projections of the tetrahedral tangle  $\Delta_{(2)}^{\varphi_1}$ . In Appendix C we include the projections of tetrahedral tangles  $\Delta_{(0)}^{\varphi_1}$ ,  $\Delta_{(1)}^{\varphi_1}$ , and  $\Delta_{(3)}^{\varphi_1}$ .

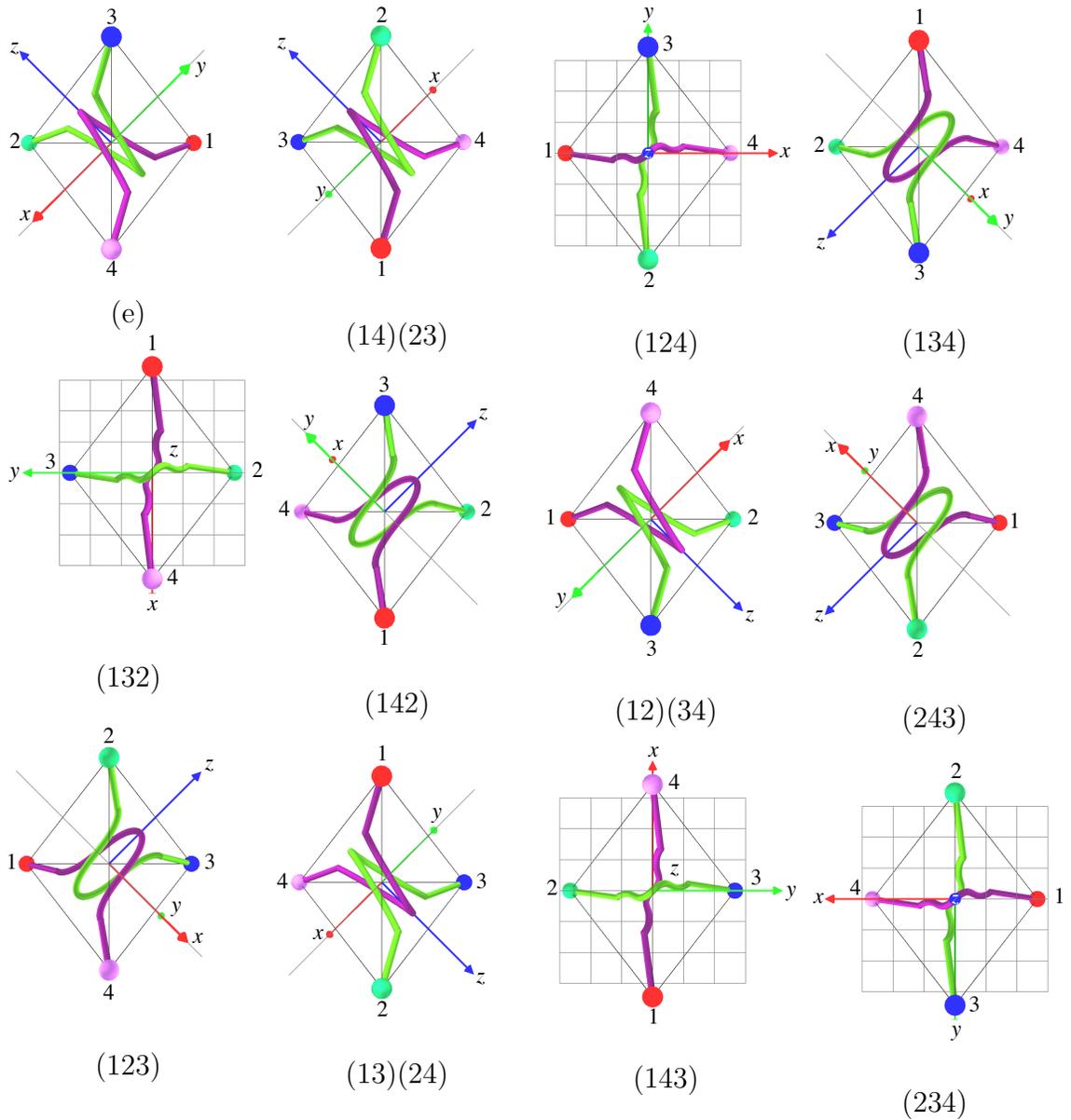


Figure 5.19: Projections of tetrahedral tangle  $\Delta_{(2)}^{\varphi_1}$  generated by the twelve elements of the tetrahedral group  $G = \{ (234), (243), (134), (143), (124), (142), (123), (132), (12)(34), (14)(23), (13)(24), (e) \}$ .

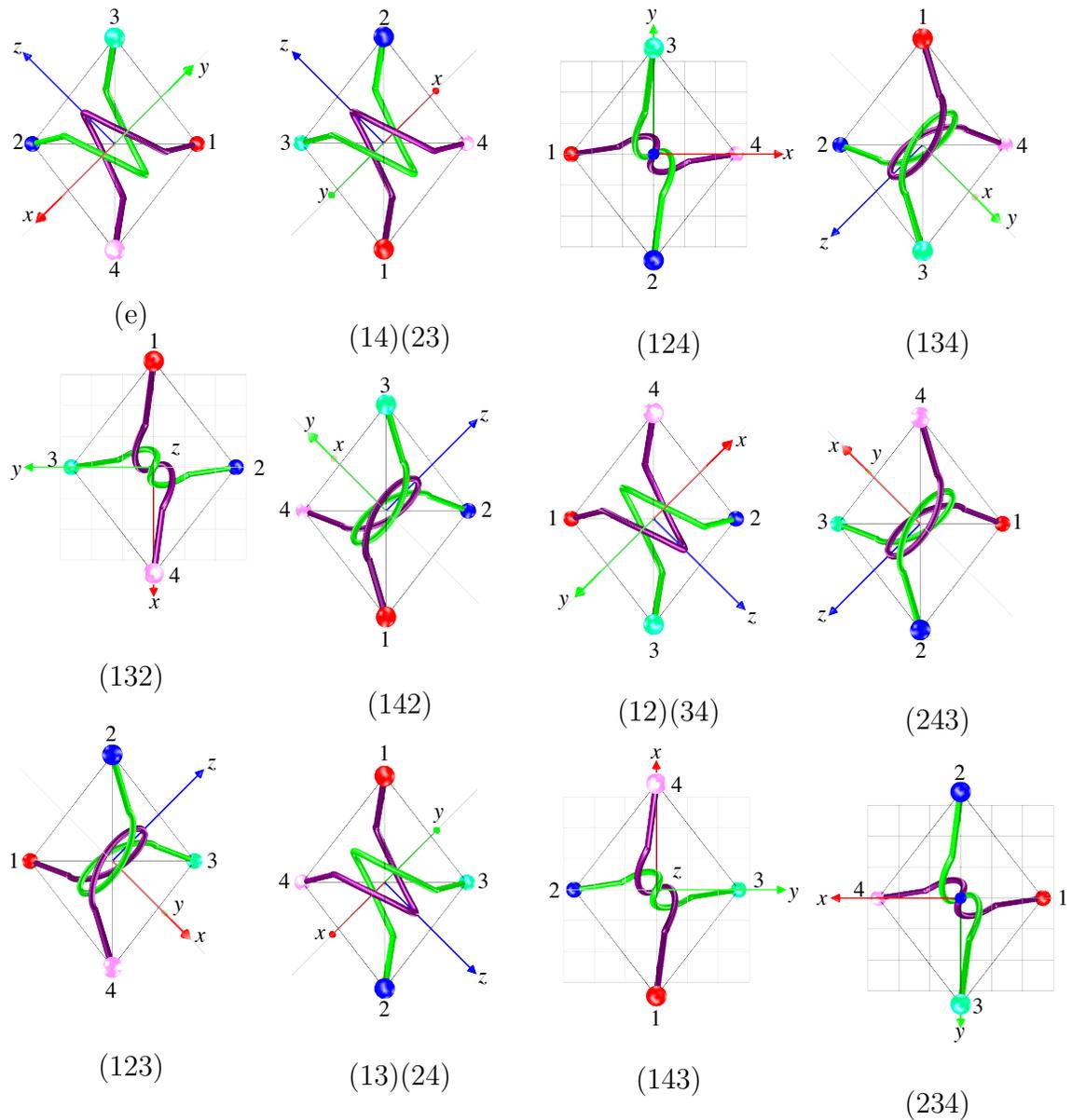


Figure 5.20: Projections of tetrahedral tangle  $\Delta_{(2)}^{\varphi_2}$  generated by the twelve elements of the tetrahedral group  $G = \{ (234), (243), (134), (143), (124), (142), (123), (132), (12)(34), (14)(23), (13)(24), (e) \}$ .

## 5.4 Equivalence Classes for Tetrahedral Tangle Projections

Tetrahedral tangles serve as a three-dimensional representation of tangles inside tetrahedra. Here we show that for any embedding of an integral tangle, the set of tetrahedral tangle projections forms a set of at most three equivalence classes. In tangle theory, the tangle diagrams shown in Figure 5.21 represent different tangles.

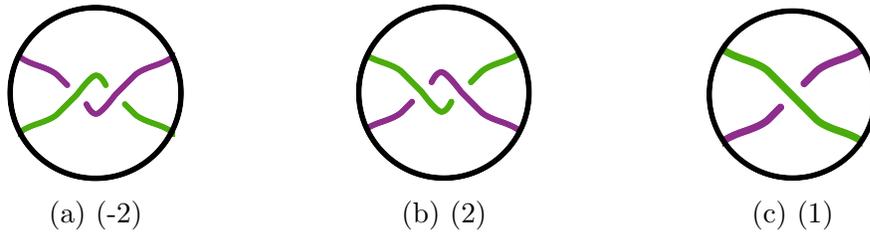


Figure 5.21: Tangle diagrams for tangles: (a) (-2), (b) (2), and (c) (1).

The projections for tetrahedral tangle  $\Delta_{(2)}^{\varphi_1}$ , shown in Figure 5.22, can each be interpreted as one of the tangles in Figure 5.21. For convenience the axes, labels, and tetrahedron edges are removed from the projections.

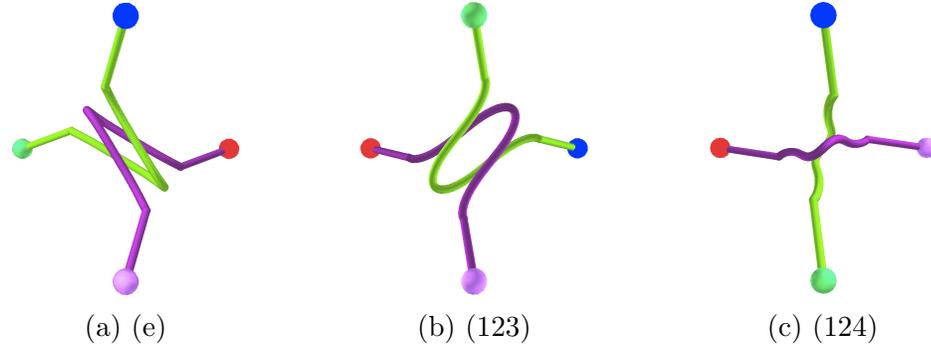


Figure 5.22: The projections of tetrahedral tangle  $\Delta_{(2)}^{\varphi_1}$  generated by the transformations (e), (123), and (124) can each be interpreted as one of the tangles in Figure 5.21. (a) can be interpreted as the tangle diagram for tangle  $(-2)$ , (b) can be interpreted as the tangle diagram for tangle  $(2)$ , and (c) can be interpreted as the tangle diagram for tangle  $(1)$ .

It should be emphasized that the same three-dimensional object has projections that correspond to tangle diagrams for different tangles. Consider the projections given by the set of rotations  $\mathcal{R}_1 = \{ (e), (14)(23), (13)(24), (12)(34) \}$  in Figure 5.19. Note that all four projections can be interpreted as a tangle diagram for tangles  $(-2)$  and  $(2, 0)$ , as shown in Figure 5.23.

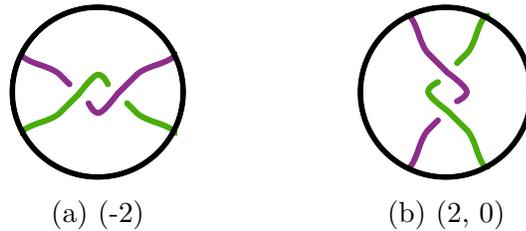


Figure 5.23: Tangle diagram interpretations (a)  $(-2)$  and (b)  $(2, 0)$  for the tetrahedral tangle  $\Delta_{(2)}^{\varphi_1}$  projections generated by  $\mathcal{R}_1$ .

Similarly, the projections given by  $\mathcal{R}_2 = \{ (123), (134), (142), (243) \}$  can be

interpreted as tangle diagrams for tangles (2) and  $(-2, 0)$ , as shown in Figure 5.24.

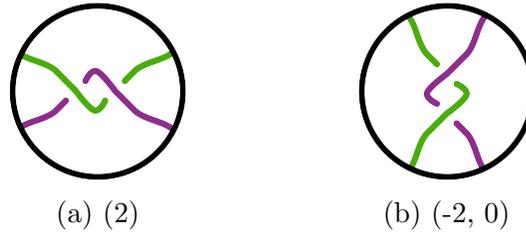


Figure 5.24: Tangle diagram interpretations (a) (2) and (b)  $(-2, 0)$  for the tetrahedral tangle  $\Delta_{(2)}^{\varphi_1}$  projections generated by  $\mathcal{R}_2$ .

The projections given by  $\mathcal{R}_3 = \{ (124), (132), (143), (234) \}$  can be interpreted as tangle diagrams for tangles (1) and  $(-1)$ , as illustrated in Figure 5.25.



Figure 5.25: Tangle diagram interpretations (a) (1) and (b)  $(-1)$  for the tetrahedral tangle  $\Delta_{(2)}^{\varphi_1}$  projections generated by  $\mathcal{R}_3$ .

Notice that what distinguishes  $\mathcal{R}_1$ ,  $\mathcal{R}_2$ , and  $\mathcal{R}_3$  is the cyclic ordering of the vertices on the plane of the page. The projections given by  $\mathcal{R}_1$ ,  $\mathcal{R}_2$ , and  $\mathcal{R}_3$  have the cyclic orderings  $(1\ 4\ 2\ 3)$ ,  $(1\ 2\ 3\ 4)$ , and  $(1\ 2\ 4\ 3)$ , respectively, in either the clockwise or counterclockwise direction (see Figures 5.26, 5.27, and 5.28).



Figure 5.26: Cyclic orderings  $(1\ 4\ 2\ 3)$  and  $(1\ 3\ 2\ 4)$  of the vertices for the tetrahedral tangle  $\Delta_{(2)}^{\varphi_1}$  projections generated by  $\mathcal{R}_1$ .



Figure 5.27: Cyclic orderings (1 2 3 4) and (1 4 2 3) of the vertices for the tetrahedral tangle  $\Delta_{(2)}^{\varphi_1}$  projections generated by  $\mathcal{R}_2$ .



Figure 5.28: Cyclic orderings (1 3 4 2) and (1 2 4 3) of the vertices for the tetrahedral tangle  $\Delta_{(2)}^{\varphi_1}$  projections generated by  $\mathcal{R}_3$ .

This suggests that the projections of  $\Delta_{(n)}^{\varphi}$  will partition into three sets of cyclic orderings, for any  $n$  and  $\varphi$ . Also notice that the only way to eliminate or introduce a crossing in the tangle diagram interpretation of a projection is by changing the cyclic ordering of the vertices. This observation allows us to define an equivalence relation on the set of projections. Let  $p_1$  and  $p_2$  be two different projections of a tetrahedral tangle  $\Delta_{(n)}^{\varphi}$ . We define the relation  $\sim$  such that if the corresponding sets of tangle diagram interpretations for  $p_1$  and  $p_2$  are the same, then  $p_1 \sim p_2$ . Since equality of sets satisfy the three properties of an equivalence relation as defined in Section 3.1.1, then the relation  $\sim$  defines an equivalence class on the projections of a tetrahedral tangle. This observation leads to the following remark that should be investigated further in future work.

*Remark 5.1.* The relation  $\sim$  defines a set of three equivalence classes on the projections of a tetrahedral tangle  $\Delta_{(n)}^{\varphi}$  where  $n \neq 0, \pm 1, \infty$ .

## Chapter 6

# Results: Application of Tetrahedral Tangles to XerCD-FtsK Unlinking

### 6.1 3D Analysis of XerCD-FtsK Unlinking Solutions

During the process of replication, the cell normally produces links that need to be properly segregated before it divides. Site-specific recombinases XerCD work together with DNA translocase FtsK to segregate the chromosome in a stepwise fashion through multiple rounds of site-specific recombination [16]. The XerCD-FtsK complex simplifies the topology of the circular DNA molecules until the unlink is reached, as shown in Figure 6.1.

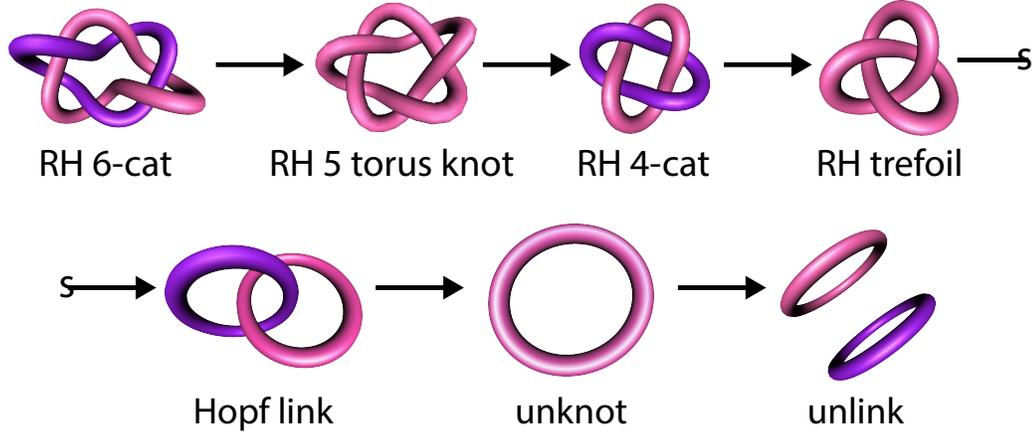


Figure 6.1: Unlinking pathway of XerCD-FtsK complex for the right-handed 6-crossing torus link. The topology is simplified until the unlink topology is reached and the two DNA molecules are properly segregated.

The tangle method models the substrate and product of a single recombination event as a system of two tangle equations: Equation (6.1) and Equation (6.2).

$$N(O + P) = K_1 \quad (\text{substrate}) \quad (6.1)$$

$$N(O + R) = K_2 \quad (\text{product}) \quad (6.2)$$

*TangleSolve* then computes the tangle solutions  $O$  and  $R$  for this system. We are interested in the sequence of topologies for the recombination events that unlink the right-handed 6-crossing torus link. For each step in this unlinking process *TangleSolve* produces three biologically relevant solutions. It was proposed by Shimokawa *et. al.* that the three solutions for the last three unlinking steps can each be uni-

fied into a single three-dimensional object [19]. Here we explore an application of tetrahedral tangles to the XerCD-FtsK unlinking steps.

### 6.1.1 Unknot $\longrightarrow$ Unlink

The recombination event that takes the unknot to the unlink (Figure 6.2) is modeled by the system of equations given by  $(\star)$ . The tangle solutions are computed by *TangleSolve* and are listed in Table 6.1 and are illustrated in Figure 6.3.

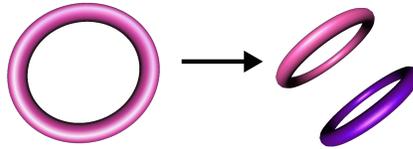


Figure 6.2: Unknot converted to unlink after XerCD recombination at site *dif*.

$$(\star) = \begin{cases} N(O + P) = 0_1, & (\text{unknot}) \\ N(O + R) = 0_1^2, & (\text{unlink}) \end{cases}$$

| Substrate        | Product            | Possible Solutions  |
|------------------|--------------------|---|
| unknot ( $0_1$ ) | unlink ( $0_1^2$ ) | $O = (0) + (-1), R = (1)$<br>$O = (0) + (+1), R = (-1)$<br>$O = (0) + (0), P = (\infty), R = (0)$ |

Table 6.1: The possible solutions of the XerCD site-specific reactions on the unknot substrate at *dif* generated by *TangleSolve*.

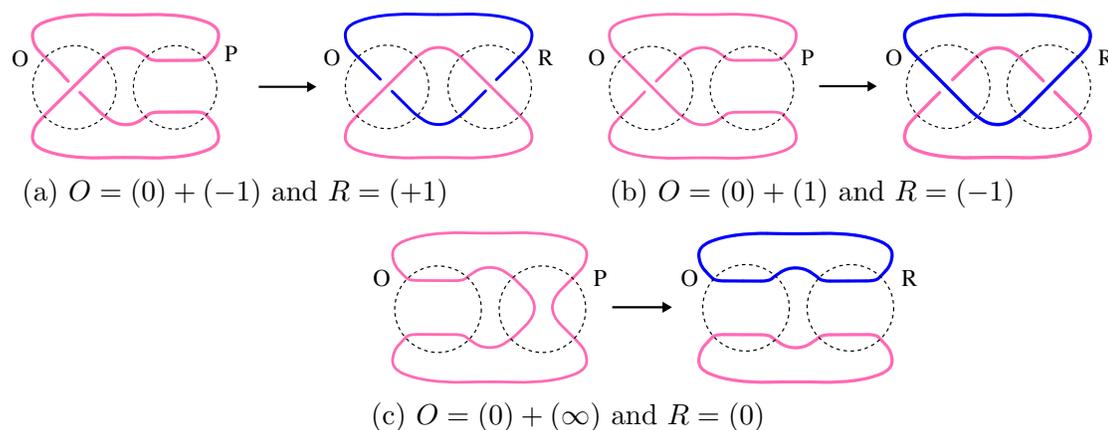


Figure 6.3: Visual representation of the three *TangleSolve* solutions for the unknot and unlink system.

The equivalence classes for projections of tetrahedral tangle  $\Delta_{(0)}^{q_1}$  include projections that correspond to all three tangle solutions  $(-1)$ ,  $(1)$ , and  $(0)$  as summarized in Table 6.2 (see Appendix C).

| Tetrahedral Tangle $\Delta_{(0)}^{\varphi_1}$ Projections |                   |                                |
|---|-------------------|--------------------------------|
| Tetrahedral Tangle  | Equivalence Class | Tangle Diagram Interpretations |
| $\Delta_{(0)}^{\varphi_1}$                                | $\mathcal{R}_1$   | $\{(0), (\infty)\}$            |
|   | $\mathcal{R}_2$   | $\{(\infty), (0)\}$            |
|   | $\mathcal{R}_3$   | $\{(-1), (1)\}$                |

Table 6.2: Equivalence classes for projections of tetrahedral tangle  $\Delta_{(0)}^{\varphi_1}$ .

This suggests that the tetrahedral tangle  $\Delta_{(0)}^{\varphi_1}$  could be a 3D object whose different planar projections correspond the other two tangle solutions.

### 6.1.2 Hopf Link $\longrightarrow$ Unknot

The recombination event that converts the Hopf link to the unknot (Figure 6.4) is modeled by the system of equations given by  $(\star\star)$ . The solutions for this system are listed in Table 6.3.

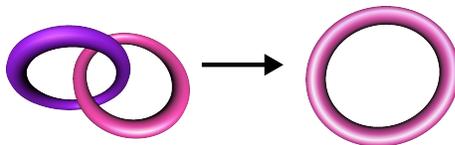


Figure 6.4: Hopf link converted to the unknot via one round of XerCD recombination at site *dif*.

$$(\star\star) = \begin{cases} N(O + P) = 2_1^2 & (\text{Hopf link}) \\ N(O + R) = 1_1 & (\text{unknot}) \end{cases}$$

| Substrate             | Product          | Possible Solutions   |
|-----------------------|------------------|--|
| Hopf link ( $2_1^2$ ) | unknot ( $0_1$ ) | $O = (0) + (2), R = (-1)$<br>$O = (0) + (2), R = (-3)$<br>$O = (0) + (2, 1, 0), R = (-1)$<br>$O = (0) + (-2), R = (3)$<br>$O = (0) + (-2, -1, 0), R = (1)$<br>$O = (0) + (-2), R = (1)$<br>$O = (0) + (2, 0), P = (\infty), R = (0)$<br>$O = (0) + (-2, 0), P = (\infty), R = (0)$ |

Table 6.3: The possible solutions of the XerCD site-specific reactions on the Hopf link substrate at *dif* generated by *TangleSolve*.

From the list of solutions in Table 6.3, we consider those that are biologically relevant, or the solutions for which the recombinant tangle  $R$  is equal to  $\pm 1$ , these solutions are illustrated in Figure 6.5. Solutions that do not satisfy this condition are removed because it is well understood that a single recombination event by XerCD adds at most a single crossing to the  $P$  tangle.

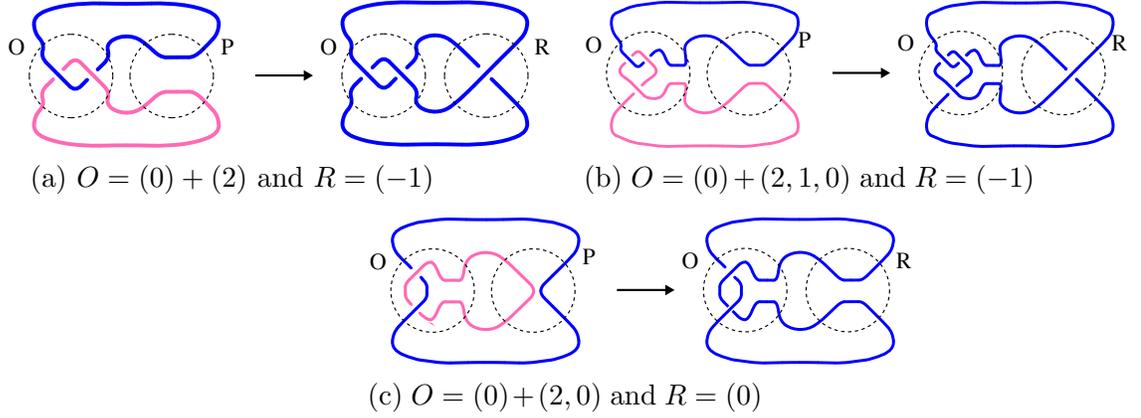


Figure 6.5: Visual representation of the three biologically relevant solutions for the Hopf Link and unknot system.

| Tetrahedral Tangle $\Delta_{(2)}^{\varphi_i}$ Projections |                   |                                |
|---|-------------------|--------------------------------|
| Tetrahedral Tangle  | Equivalence Class | Tangle Diagram Interpretations |
| $\Delta_{(2)}^{\varphi_1}$                                | $\mathcal{R}_1$   | $\{(-2, 0), (2)\}$             |
|   | $\mathcal{R}_2$   | $\{(-2), (2, 0)\}$             |
|   | $\mathcal{R}_3$   | $\{(1), (-1)\}$                |
| $\Delta_{(2)}^{\varphi_2}$                                | $\mathcal{R}_1$   | $\{(-2, 0), (2, 1, 0)\}$       |
|   | $\mathcal{R}_2$   | $\{(2), (-2, 0)\}$             |
|   | $\mathcal{R}_3$   | $\{(-3, 0), (3)\}$             |

Table 6.4: Equivalence classes for projections of tetrahedral tangle  $\Delta_{(2)}^{\varphi_i}$ .

| Tetrahedral Tangle $\Delta_{(-2)}^{\varphi_i}$ Projections |                   |                                |
|--|-------------------|--------------------------------|
| Tetrahedral Tangle   | Equivalence Class | Tangle Diagram Interpretations |
| $\Delta_{(-2)}^{\varphi_1}$                                | $\mathcal{R}_1$   | $\{(2, 0), (-2)\}$             |
|  | $\mathcal{R}_2$   | $\{(-3), (3, 0)\}$             |
|  | $\mathcal{R}_3$   | $(-2, -1, 0)$                  |
| $\Delta_{(-2)}^{\varphi_2}$                                | $\mathcal{R}_1$   | $\{(-2, 0), (2)\}$             |
|  | $\mathcal{R}_2$   | $\{(-2), (2, 0)\}$             |
|  | $\mathcal{R}_3$   | $\{(1), (-1)\}$                |

Table 6.5: Equivalence classes for projections of tetrahedral tangle  $\Delta_{(-2)}^{\varphi_i}$ .

The equivalence classes for projections of tetrahedral tangle  $\Delta_{(-2)}^{\varphi_1}$  include projections that correspond to tangle solutions (2) and (2, 0), whereas the equivalence classes for projections of  $\Delta_{(-2)}^{\varphi_2}$  include projections that correspond to tangle solutions (2) and (2, 1, 0). Since neither tetrahedral tangle gives projections that correspond to all three tangle solutions, we suggest that the planar projections of multiple tetrahedral tangles joined together be investigated. We incorporated this feature in *Tangle3D*, and an example of two tetrahedral tangles joined together by unknotted arcs is illustrated in Figure 6.6.

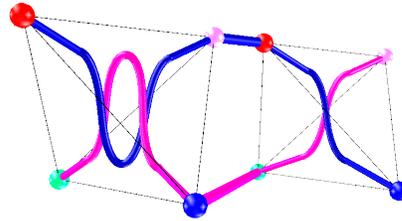


Figure 6.6: Connected tetrahedral tangles generated by *Tangle3D*.

### 6.1.3 RH Trefoil Knot $\longrightarrow$ Hopf Link

The recombination event that converts the right-handed trefoil knot to the Hopf link (Figure 6.7) is modeled by the system of equations given by  $(\star\star\star)$ . The solutions for this system are listed in Table 6.6 and are illustrated in Figure 6.8.

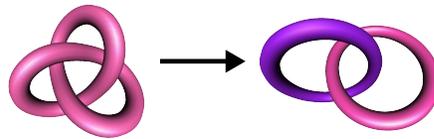


Figure 6.7: Right-handed trefoil knot converted to the Hopf link via one round of XerCD recombination at site *dif*.

$$(\star\star\star) = \begin{cases} N(O + P) = 3_1 & \text{(RH trefoil knot)} \\ N(O + R) = 1_1 & \text{(unknot)} \end{cases}$$

| Substrate                 | Product          | Possible Solutions  |
|---------------------------|------------------|---|
| RH trefoil knot ( $3_1$ ) | unknot ( $1_1$ ) | $O = (0) + (3), R = (-1)$<br>$O = (0) + (3), R = (-5)$<br>$O = (0) + (-2, -1, -1, 0), R = (+1)$<br>$O = (0) + (2, 1, 0), P = (\infty), R = (0)$ |

Table 6.6: The possible solutions of the XerCD site-specific reactions on the right-handed trefoil knot substrate at *dif* generated by TangleSolve.

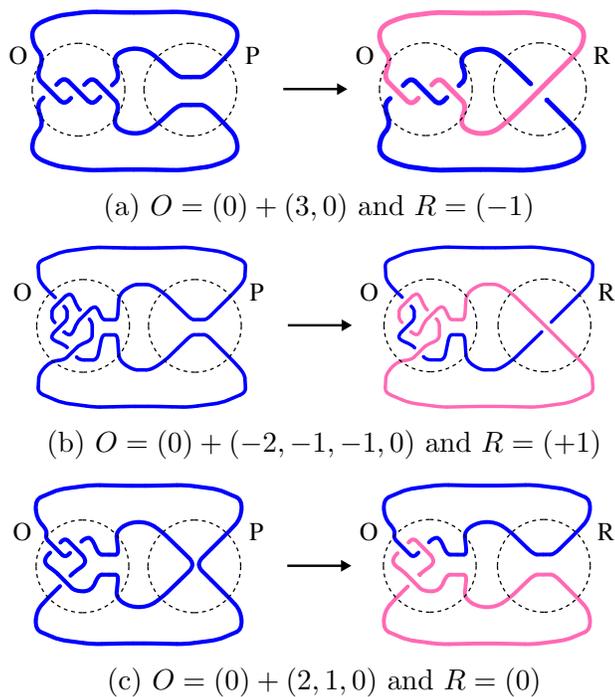


Figure 6.8: Visual representation of the three biologically relevant *TangleSolve* solutions for the right-handed trefoil knot and Hopf link system.

| Tetrahedral Tangle $\Delta_{(3)}^{\varphi_i}$ Projections |                   |                                |
|---|-------------------|--------------------------------|
| Tetrahedral Tangle  | Equivalence Class | Tangle Diagram Interpretations |
| $\Delta_{(3)}^{\varphi_1}$                                | $\mathcal{R}_1$   | $\{(2, 1, 0)\}$                |
|   | $\mathcal{R}_2$   | $\{(-2, 0), (3)\}$             |
|   | $\mathcal{R}_3$   | $\{(2), (-2, 0)\}$             |

Table 6.7: Equivalence classes for projections of tetrahedral tangle  $\Delta_{(3)}^{\varphi_1}$ .

Table 6.7 gives the equivalence classes for the projections of tetrahedral tangle  $\Delta_{(3)}^{\varphi_1}$ . Note that, as with the Hopf link and unknot system, the tetrahedral tangles only give projections that correspond to two of the three tangle solutions. This trend occurs for all XerCD-FtsK unlinking steps of the right-handed 6-crossing torus link, which further motivates the study of planar projections for connected tetrahedral tangles.

## Chapter 7

# Conclusions and Future Directions

In this thesis, we have seen the mathematical tools used to model site-specific recombination. We extend the work of Wono and Vazquez and give a formal definition of a tetrahedral tangle: an embedding of the tangle strings inside of a regular tetrahedron. We develop a computer visualization tool called *Tangle3D* that works with *TangleSolve* to construct tetrahedral tangles. We also define a set of equivalence classes for the projections of tetrahedral tangles, generated by the rotations of the tetrahedral group. We then apply this 3-dimensional analysis to the unlinking recombination events of XerCD-FtsK at the site *dif*.

One limitation of this research we would like to address is that this thesis only considers embedding the integral tangle solutions inside of a tetrahedron. Although it is unlikely that the tetrahedral tangles for non-integral tangle solutions would produce projections that correspond to integral tangle solutions.

For future work we propose incorporating an energy minimizing function for the strings of a tetrahedral tangle. This would produce cleaner projections of more complex tetrahedral tangles, and would be an interesting project that would take a 3D representation of the tangle method to a more realistic and dynamic model. Also, we propose that future work investigate embedding tangles inside of higher order polyhedra. The advantage with the tetrahedron is it provides the greatest difference in the angle of the tangle strings.

## Appendix A: Elementary Number Theory

To give a proof of *Conway's Classification Theorem of Rational Tangles* which states that there is a one-to-one correspondence between rational tangles and the rational numbers [8], it is necessary to provide some background from elementary number theory. In particular, tangles can be nicely understood algebraically with their corresponding *continued fraction* representations. The continued fraction of a given rational number is computed using the *Division algorithm* and the *Euclidean algorithm*.

**Theorem 7.1. *The Division Algorithm*** *If  $a$  and  $b$  are integers such that  $b > 0$ , then there are unique integers  $q$  and  $r$  such that  $a = bq + r$  with  $0 \leq r < b$ . [24]*

In the equation of the division algorithm,  $q$  is referred to as the *quotient* and  $r$  as the *remainder*. The integers  $a$  and  $b$  are called the *dividend* and *divisor* respectively.

**Example 7.1.** If  $a = 86$  and  $b = 7$ , then  $q = 12$  and  $r = 2$  because

$$\begin{array}{ccccccc} \underline{86} & = & \underline{7} & * & \underline{12} & + & \underline{2} \\ \text{dividend} & & \text{divisor} & & \text{quotient} & & \text{remainder} \end{array}$$

**Definition 7.1.** The **greatest common divisor** of two integers  $a$  and  $b$ , which are not both 0, is the largest integer that divides both  $a$  and  $b$ , denoted by  $(a, b)$ .

[24]

**Theorem 7.2. The Euclidean Algorithm** Let  $r_0 = a$  and  $r_1 = b$  be integers such that  $a \geq b > 0$ . If the division algorithm is successfully applied to obtain  $r_j = r_{j+1}q_{j+1} + r_{j+2}$ , with  $0 < r_{j+2} < r_{j+1}$  for  $j = 0, 1, 2, \dots, n - 2$  and  $r_{n+1} = 0$ , then  $(a, b) = r_n$ , the last nonzero remainder. [24]

The *Euclidean algorithm* applied to Example 7.1 is shown in Example 7.2.

---

**Algorithm 2** Euclid's algorithm

---

1: **procedure** EUCLID( $a, b$ )

2:      $r \leftarrow a \bmod b$

3:     **while**  $r \neq 0$  **do**

4:          $a \leftarrow b$

5:          $b \leftarrow r$

---

---



---

```

6:       $r \leftarrow a \bmod b$ 
7:  end while
8:  return  $b$                                 ▷ The gcd is  $b$ 
9: end procedure

```

---

**Example 7.2.** Let  $a = 86$  and  $b = 7$

$$86 = 7 * 12 + 2 \quad (a = 86, b = 7, r = 2)$$

$$7 = 2 * 3 + 1 \quad (a = 7, b = 2, r = 1)$$

$$2 = 1 * 2 + 0 \quad (a = 2, b = 1, r = 0)$$

then  $(a, b) = 1$

**Definition 7.2.** A **finite continued fraction** is an expression of the form

$$[a_0, a_1, a_2, \dots, a_n] = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \dots + \frac{1}{a_{n-1} + \frac{1}{a_n}}}}$$

where  $a_0, a_1, a_2, \dots, a_n$  are real numbers with  $a_1, a_2, a_3, \dots, a_n$  positive. The real numbers  $a_1, a_2, a_3, \dots, a_n$  are called the **partial quotients** of the continued fraction.

[24]

**Definition 7.3.** A finite continued fraction is called **simple** if the partial quotients are all integers.

**Theorem 7.3.** *Every finite simple continued fraction represents a rational number.*

Theorem 7.3 can be proven by induction as shown below.

*Proof.* Proof by induction:

**Base Case:**  $n = 1$  we have

$$[a_0, a_1] = a_0 + \frac{1}{a_1} = \frac{a_0 a_1 + 1}{a_1}$$

the base case is true.

**Inductive step:** Assume that for  $k$

$$[a_0, a_1, a_2, \dots, a_k] = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \dots + \frac{1}{a_{k-1} + \frac{1}{a_k}}}}$$

is a rational number. Now let  $a_0, a_1, \dots, a_{k+1}$  be integers with  $a_1, \dots, a_{k+1}$  be posi-

tive. Note that

$$\begin{aligned} [a_0, a_1, a_2, \dots, a_{k+1}] &= a_0 + \frac{1}{[a_1, a_2, a_3, \dots, a_{k+1}]} \\ &= a_0 + \frac{1}{r/s} \\ &= \frac{a_0 r + s}{r} \end{aligned}$$

where  $[a_1, a_2, a_3, \dots, a_{k+1}]$  is a rational number  $r/s$ . □

**Theorem 7.4.** *Every rational number can be expressed by a finite simple continued fraction. [24]*

Every rational number has exactly two finite simple continued fraction expansions: one with an even number of partial quotients and the other with an odd number of partial quotients. [24]

**Example 7.3.** The rational number  $\frac{7}{11} = [0, 1, 1, 1, 3] = [0, 1, 1, 1, 2, 1]$ .

Example 7.3 gives the two finite simple continued fraction expansions of the rational number  $\frac{7}{11}$ .

## Appendix B: *TangleSolve* and *Tangle3D*

In 2002, Saka and Vazquez developed *TangleSolve*, a Java applet/application which implements the Tangle Method and offers a user interface for analyzing and visualizing recombination mechanisms [25]. For details on how to get access to this program and a tutorial on how to use it please visit <http://bio.math.berkeley.edu/TangleSolve/>.

As part of this thesis, we have rewritten *TangleSolve* and added several new features with special thanks to professional Java programmer Nicolas Gramlich. The key contribution to this thesis is the development of *Tangle3D*, a computer implementation of the embedding of tangles inside of tetrahedra. *Tangle3D* allows the user to create and manipulate tetrahedral tangles. Below are some visual instructions for using *Tangle3D* within *TangleSolve*.

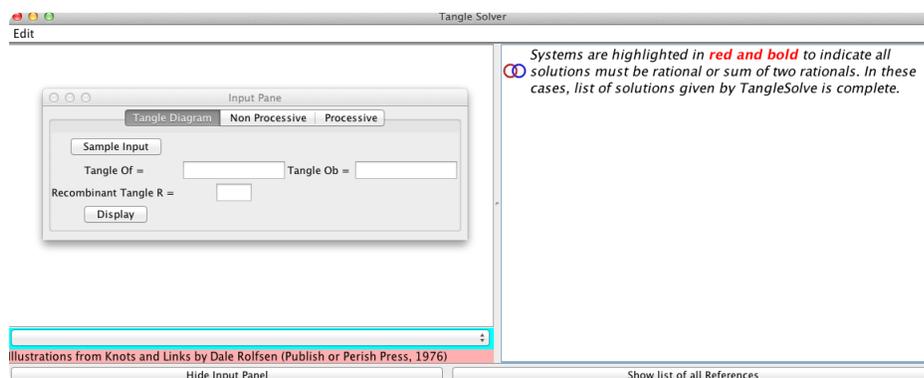


Figure 7.1: Screenshot of Java Applet *TangleSolve*. When the program is started two windows will open: *Tangle Solver* and *Input Pane*. *Input Pane* is where information on the topology of the substrate and product (from experimental data) is entered.

(1) **Step 1:** Start the *TangleSolve* application. Two windows will open: *Tangle Solver* and *Input Pane* as shown in Figure 7.1. *Input Pane* is where the experimental data is entered. The *Input Pane* has three tabs: *Tangle Diagram*, *Non Processive* and *Processive*. We here use the *Tangle Diagram* tab to demonstrate the *3DTangle* feature. Detailed instructions on using *TangleSolve* are available on the *TangleSolve* website.

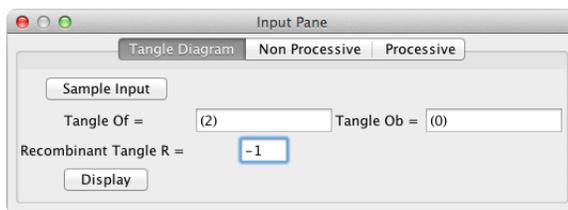


Figure 7.2: *TangleSolve* Step 2: Enter the desired values for tangles  $O_f$ ,  $O_b$  and  $R$  into the *Input Pane* window. Here  $O_f = (2)$ ,  $O_b = (0)$ , and  $R = (-1)$ . In this example the substrate and product topologies are given by  $N((2) + (0) + (0))$  and  $N((2) + (0) + (-1))$  respectively.

(2) **Step 2 :** Enter the values for parameters  $O_f$ ,  $O_b$  and  $R$  into the *Input Pane* window (Figure 7.2). The example in Figure 7.2 illustrates the following input:

$O_f = (2)$ ,  $O_b = (0)$ , and  $R = (-1)$ . The system  $(\star)$  displays the tangle equations for this input

$$(\star) = \begin{cases} N((2) + (0) + (0)) = K_1, & \text{(substrate)} \\ N((2) + (0) + (-1)) = K_2, & \text{(product)} \end{cases}$$

where  $K_1$  and  $K_2$  represents the topologies of the substrate and product respectively.

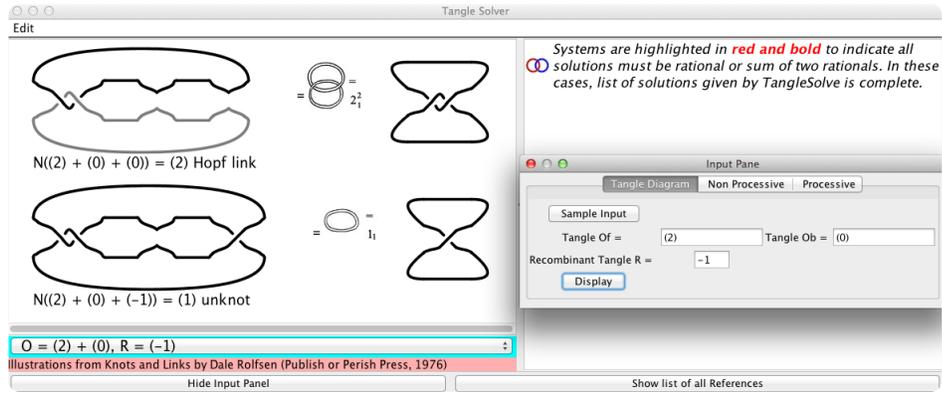


Figure 7.3: *TangleSolve* Step 3: Select the *Display* button to view the tangle diagrams and 4-plats corresponding to the system  $(\star)$  of tangle equations in the *Display Pane*. *TangleSolve* identifies the Hopf link as the substrate and the unknot as the product.

- (3) **Step 3:** Select the *Display* button to view the tangle diagrams and 4-plats corresponding to the system  $(\star)$  of tangle equations in the *Display Pane* window (Figure 7.3). In this case, *TangleSolve* identifies the Hopf link as the substrate  $K_1$  and the unknot as the product  $K_2$ .

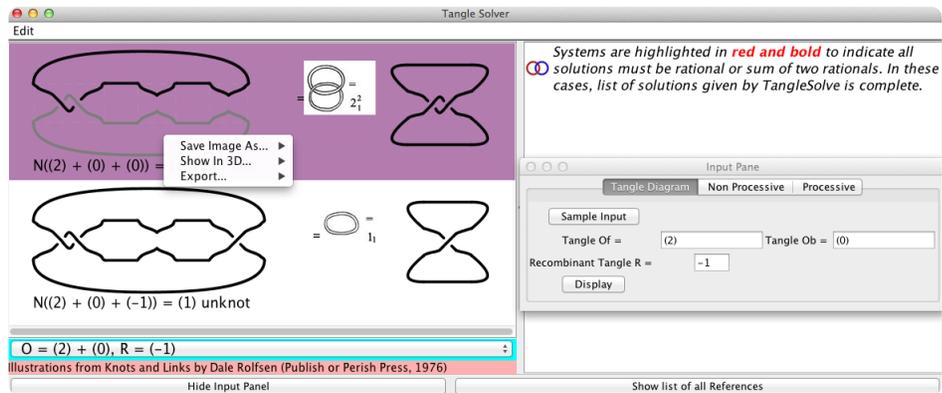


Figure 7.4: *TangleSolve* Step 4: Move the cursor over the tangle diagram for either the substrate or the product and right click on the image. A drop down menu will appear with the options *Save Image As...*, *Show In 3D...* and *Export...*

(4) **Step 4:** Move the cursor over the tangle diagram for either the substrate or the product, and right click on the image as shown in Figure 7.4. A drop down menu will appear with the options *Save Image As...*, *Show In 3D...* and *Export...*

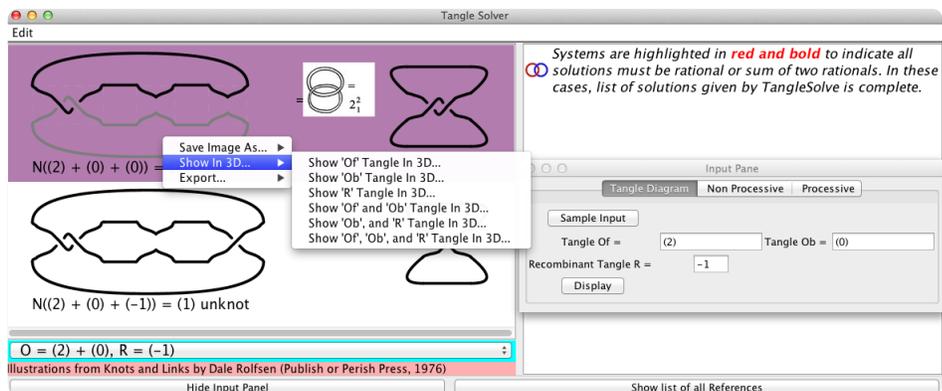


Figure 7.5: *TangleSolve* Step 5: Move the cursor down the menu and hover over the *Show In 3D...* option. Another drop down menu will appear with several different combinations of the tangles to view in 3D. In this example the “Show ‘ $O_f$ ’ tangle in 3D” option was selected.

(5) **Step 5:** Move the cursor down the menu and hover over the *Show In 3D...* option, another drop down menu will appear with several different combinations

of the tangles to view in 3D (Figure 7.5). In this example the “Show ‘ $O_f$ ’ tangle in 3D” option was selected..

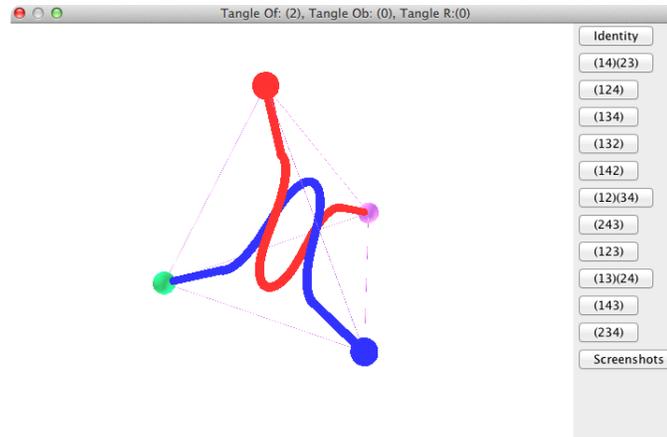


Figure 7.6: *TangleSolve* Step 6: After selecting the “Show ‘ $O_f$ ’ tangle in 3D” option, another window will open that displays tangle  $O_f = (2)$  embedded inside of the tetrahedron and allows the user to freely rotate it in space.

(6) **Step 6:** The tangle selected in **Step 5** will appear in the window *Tangle: (T)*.

This window displays the tangle, in this case  $O_f = (2)$ , embedded inside of the tetrahedron and allows the user to freely rotate it in space as shown in Figure 7.6.

# Appendix C: Tetrahedral Tangle Projections

Here we provide the projections of tetrahedral tangles  $\Delta_{(0)}^{\varphi_1}$ ,  $\Delta_{(1)}^{\varphi_1}$ , and  $\Delta_{(3)}^{\varphi_1}$ .

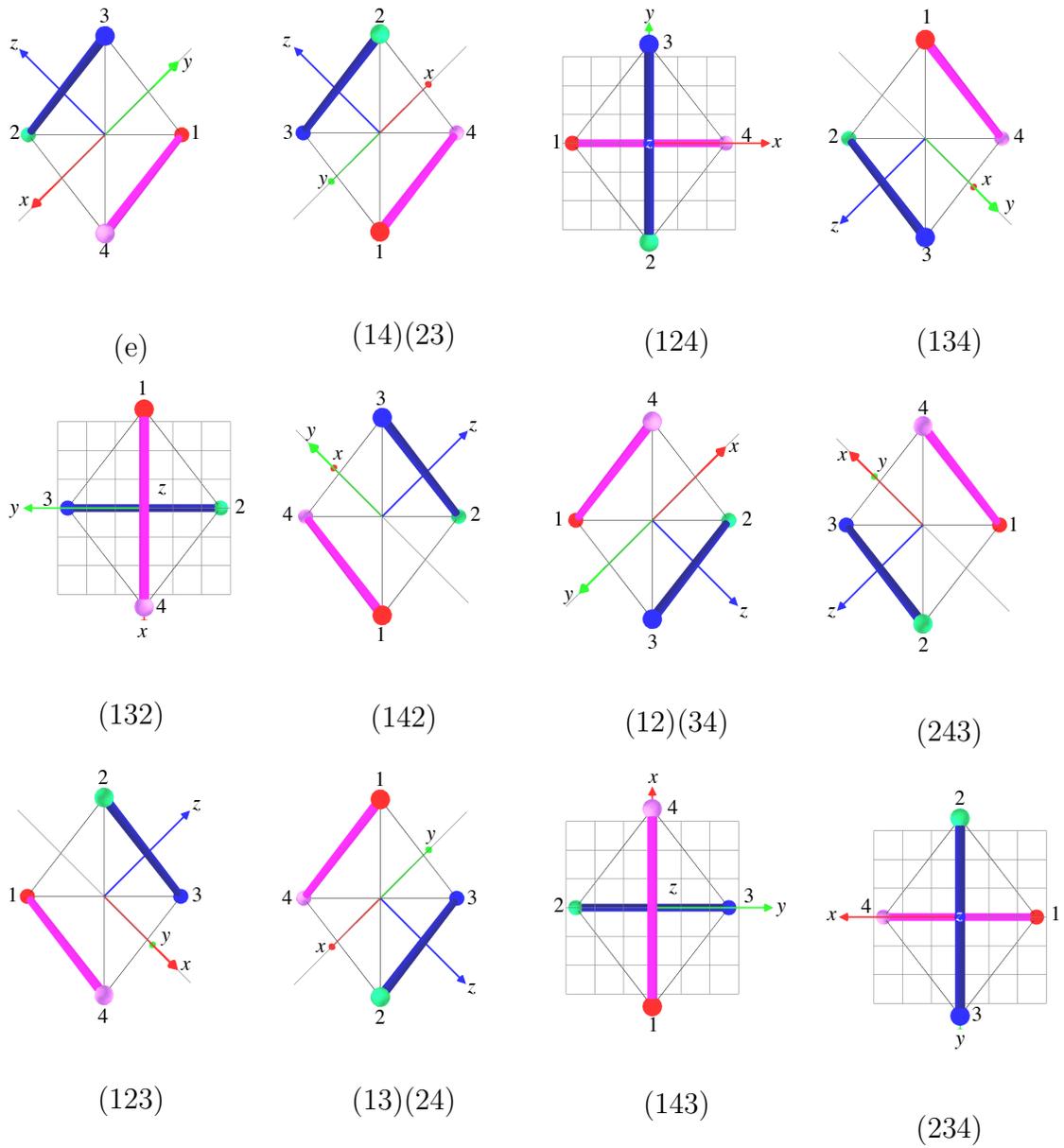


Figure 7.7: Projections of tetrahedral tangle  $\Delta_{(0)}^{\varphi_1}$  generated by the twelve elements of the tetrahedral group.

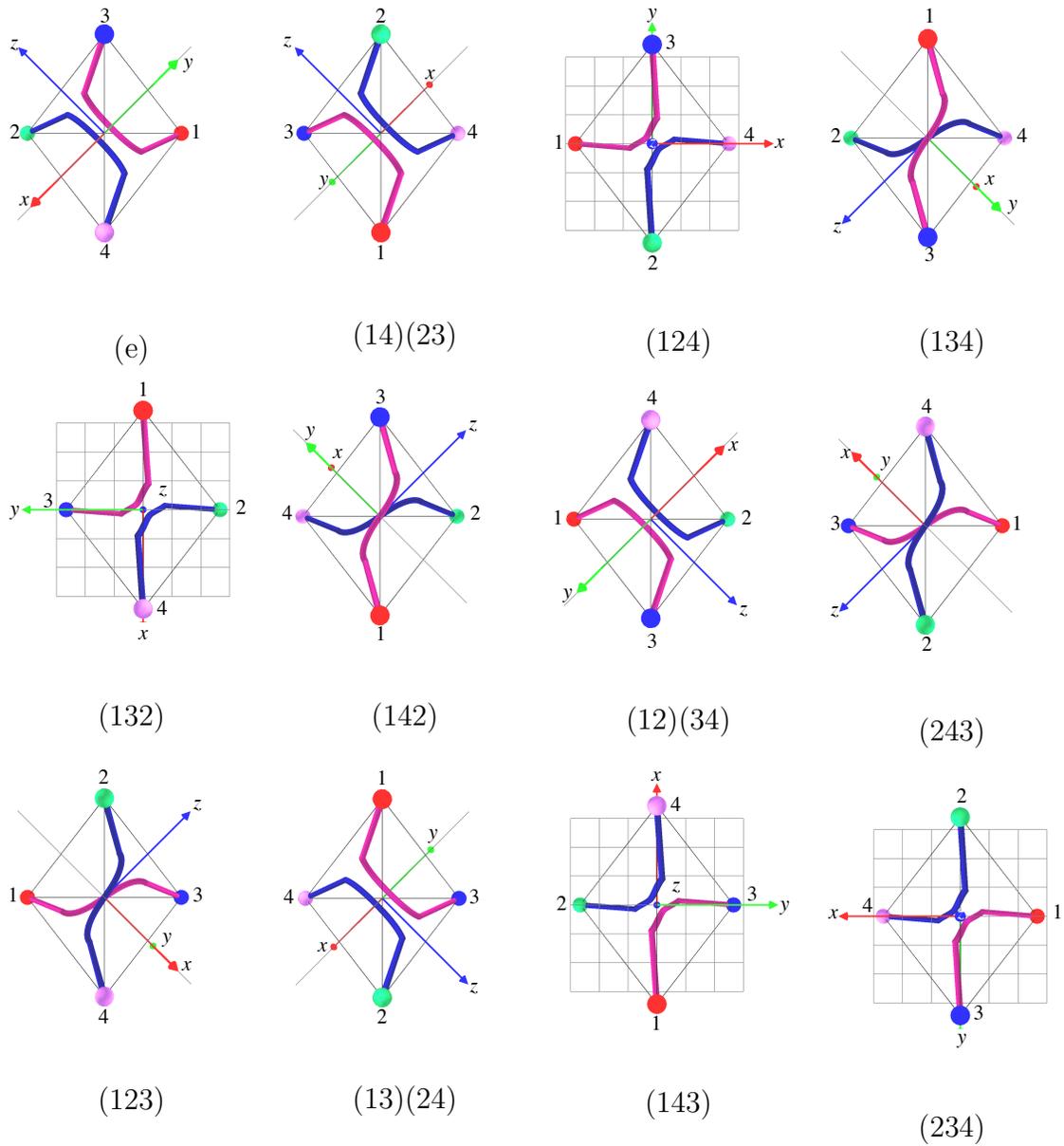


Figure 7.8: Projections of tetrahedral tangle  $\Delta_{(1)}^{\varphi_1}$  generated by the twelve elements of the tetrahedral group.

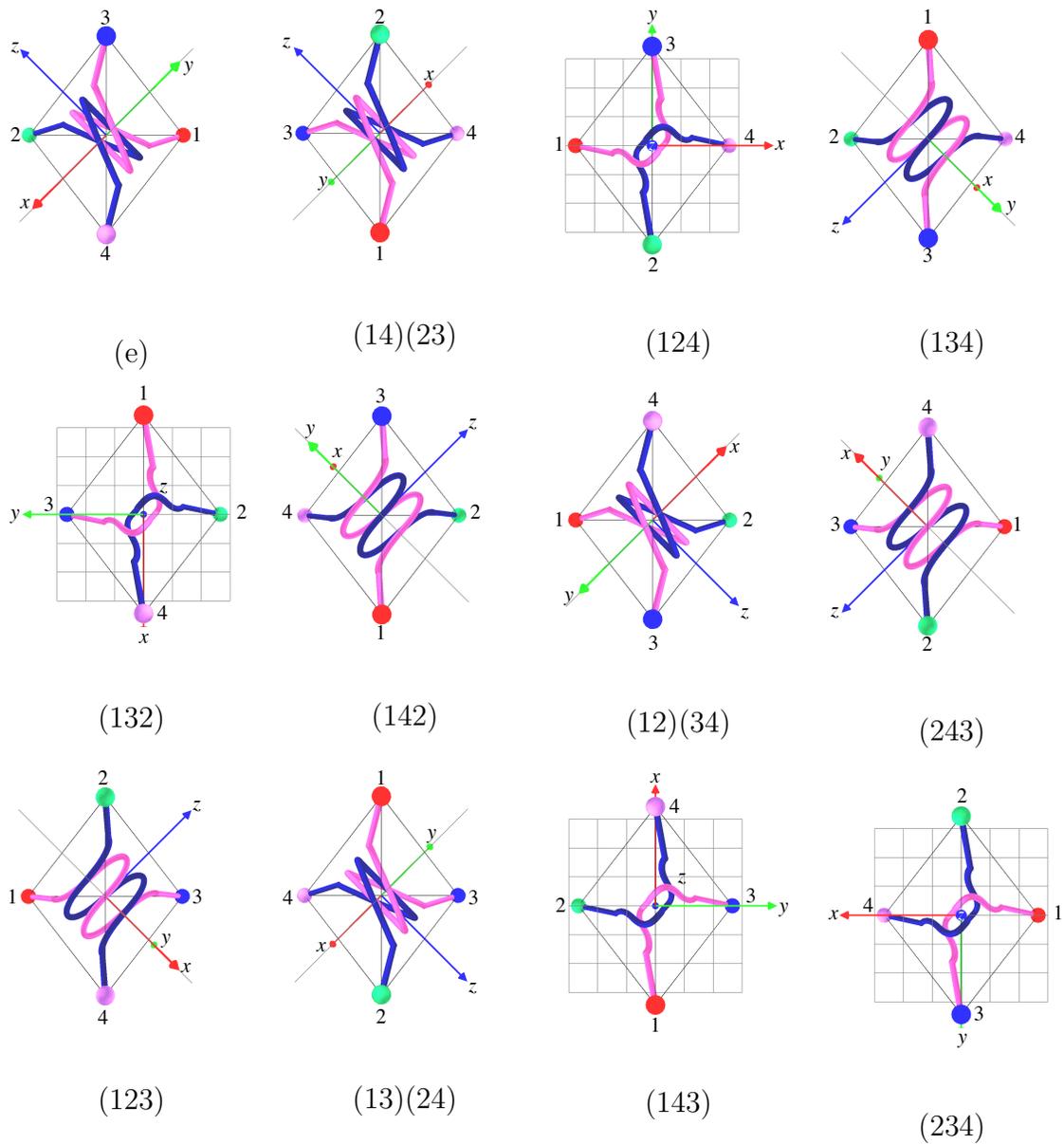


Figure 7.9: Projections of tetrahedral tangle  $\Delta_{(3)}^{\varphi_1}$  generated by the twelve elements of the tetrahedral group.

## Appendix D: Algorithms

---

### Algorithm 3 Construction of a Rational Tangle

---

```

1: Let  $\{a_0, a_1, \dots, a_n : n \in \mathbb{N}\}$  and  $T$  the constructed tangle
2:
3: if  $n \bmod 2 = 0$  then                                     ▷ vector length is odd
4:    $a_0$  is horizontal
5: else                                                       ▷ vector length is even
6:    $a_0$  is vertical
7: end if
8:
9:  $T = a_0$ 
10: for  $i = 1$  to  $n$  do
11:   if  $a_0$  is horizontal then                               ▷ first twist is horizontal
12:     if  $i$  is even then
13:        $T = T + a_i$ 

```

---

---



---

```

14:     else
15:          $T = T +' a_i$ 
16:     end if
17: else ▷ first twist is vertical
18:     if  $i$  is even then
19:          $T = T +' a_i$ 
20:     else
21:          $T = T + a_i$ 
22:     end if
23: end if
24: end for

```

---

Algorithm 4 provides the pseudo code used to construct the *Twist* object for each entry in a tangle vector. The code is implemented in the *Tangle3D* class.

---

**Algorithm 4** Computes the *Twist* object for each entry in the tangle vector.

---

```

1: procedure TWIST( $T = (a_0, a_1, \dots, a_n)$ )                                ▷ input tangle vector
2:    $x \leftarrow 0$                                                             ▷  $x$  coordinate
3:    $y \leftarrow 0$                                                             ▷  $y$  coordinate
4:   twistOrientation                                                            ▷ starting twist type
5:   if  $n \bmod 2 = 0$  then
6:     twistOrientation  $\leftarrow$  vertical
7:   else
8:     twistOrientation  $\leftarrow$  horizontal
9:   end if

```

---

---

---

```
10:   for all  $i$  in  $T$  do
11:     numberCrossings  $\leftarrow |a_i|$ 
12:     if  $a_i > 0$  then
13:       crossingType  $\leftarrow$  positive
14:     else
15:       crossingType  $\leftarrow$  negative
16:     end if
17:     twistX  $\leftarrow$  0                                 $\triangleright$  start  $x$  position of twist
18:     twistY  $\leftarrow$  0                                 $\triangleright$  start  $y$  position of twist
19:     twistWidth  $\leftarrow$  0                             $\triangleright$  horizontal length of twist
20:     twistHeight  $\leftarrow$  0                           $\triangleright$  vertical length of twist
21:     switch twistOrientation do
22:       case horizontal
23:         twistX  $\leftarrow x$ 
24:         twistY  $\leftarrow$  0
25:         twistWidth  $\leftarrow$  numberCrossings * crossingSize
26:         if  $i = 0$  then
27:           twistHeight  $\leftarrow$  crossingSize
28:         else
29:           twistY  $\leftarrow y$ 
30:         end if
31:       case vertical
32:         twistX  $\leftarrow$  0
```

---

---

---

```
33:         twistY  $\leftarrow$  y
34:         if  $i = 0$  then
35:             twistWidth  $\leftarrow$  crossingSize
36:         else
37:             twistWidth  $\leftarrow$  x
38:         end if
39:         twistHeight  $\leftarrow$  numberCrossings * crossingSize
40:         switch twistOrientation do                                 $\triangleright$  Update the  $x$  and  $y$  points
41:             case horizontal
42:                  $x \leftarrow x +$  twistWidth
43:                  $y \leftarrow$  twistY + twistHeight
44:             case vertical
45:                  $x \leftarrow$  twistX + twistWidth
46:                  $y \leftarrow y +$  twistHeight
47:         switch twistOrientation do
48:             case horizontal
49:                 twistOrientation  $\leftarrow$  vertical
50:             case vertical
51:                 twistOrientation  $\leftarrow$  horizontal
52:         end for
53:         return twist
54: end procedure
```

---

## Bibliography

- [1]
- [2] *Antibiotic Resistance Threats in the United States, 2013*, <http://www.cdc.gov/drugresistance/threat-report-2013/>, Accessed: 2015-05-22.
- [3] B Alberts, A Johnson, J Lewis, and et al., *Molecular Biology of the Cell*, Garland Science, New York, 2002.
- [4] RE. Besser, SM. Lett, JT Weber, and et al., *An outbreak of diarrhea and hemolytic uremic syndrome from Escherichia coli O157:H7 in fresh-pressed apple cider.*, The Journal of the American Medical Association **269** (1993), 2217–2220.
- [5] Reuben Brasher, G. Scharein Rob, and Mariel Vazquez, *New biologically motivated knot table.*, Biochemical Society Transactions **41** (2013), 606–611.
- [6] N.A. Campbell and J.B. Reece, *Biology*, 2002.
- [7] SD Colloms, C. Alén, and DJ. Sherratt, *The ArcA/ArcB two-component regulatory system of Escherichia coli is essential for Xer site-specific recombination at psi.*, Mol Microbiol **3** (1998), 521–530.
- [8] J.H. Conway, *An enumeration of knots and links and some of their algebraic properties*, Proceedings of the conference on Computational problems in Abstract Algebra held at Oxford in 1967 (1970), 329358.
- [9] R. Deibler, J. Mann, Z. Lynn, De Witt Sumners, and et al., *Hin-mediated DNA knotting and recombining promote replicon dysfunction and mutation.*, BMC Molecular Biology **8** (2007), 44.

- [10] R.W. Deibler, S. Rahmati, and E.L. Zechiedrich, *Topoisomerase IV, alone, unknots DNA in E. coli.*, Genes Dev **15** (2001), 748–761.
- [11] C. Ernst and D.W. Sumners, *Solving tangle equations arising in a DNA recombination model*, Math. Proc. Camb. Phil. Soc. **126** (1999), 23–36.
- [12] Jay R. Goldman and Louis H. Kauffman, *Rational Tangles*, Advances in Applied Mathematics **18** (1997), no. AM960511, 300–332.
- [13] Paul R. Halmos, *Naive Set Theory*, Litton Educational Publishing, Incorporated, 1960.
- [14] Robin Hartshorne, *Geometry: Euclid and Beyond*, Springer, 1997.
- [15] Grainge I., Bregu M., Vazquez M., Sivanathan V., Ip S., and Sherratt D., *Unlinking chromosome catenanes in vivo by site-specific recombination*, The EMBO Journal **26** (2007), 4228–4238.
- [16] SC. Ip, M. Bregu, FX. Barre, and DJ. Sherratt, *Decatenation of DNA circles by FtsK-dependent Xer site-specific recombination*, EMBO J **22** (2003), 6399–6407.
- [17] Portugal J. and Rodríguez-Campos A., *T7 RNA polymerase cannot transcribe through a highly knotted DNA template.*, Nucleic Acids Research **24** (1996), 4890–4894.
- [18] Champoux JJ., *Lifting the curtain: Using topology to probe the hidden action of enzymes.*, Not. American Mathematical Society **42** (1995), 528–537.
- [19] Shimokawa Koya, Ishihara Kai, Grainge Ian, J. Sherratt David, and Vazquez Mariel, *FtsK-dependent XerCD-dif recombination unlinks replication catenanes in a stepwise manner*, Proceedings of the National Academy of Sciences USA **110** (2013), 20906–20911.
- [20] W. B. Raymond Lickorish, *Prime Knots and Tangles*, Transactions of the American Mathematical Society **267** (1981), 321–332.
- [21] D.W. Sumners M. Vazquez, S.D. Colloms, *Tangle Analysis of Xer Recombination Reveals only Three Solutions, all Consistent with a Single 3-dimensional Topological Pathway*, J. Mol. Biol **346** (2005), 493–504.

- [22] Kunio Murasugi, *Knot Theory and Its Applications*, Birkhäuser Boston, 1996.
- [23] Kurt Reidemeister, *Elementare Begründung der Knotentheorie*, Abh. Math. Sem. Univ. Hamburg **5** (1926), 24–32.
- [24] Kenneth H. Rosen, *Elementary Number Theory and its applications*, Pearson Education, 2005.
- [25] Y. Saka and M. Vazquez, *Tanglesolve: topological analysis of site-specific recombination*, Bioinformatics **18** (2002), 1011–1012.
- [26] . Colloms Sean D, Bath Jonathan, and Sherratt David J., *Topological selectivity in Xer site-specific recombination.*, Cell **88** (1997), 855–864.
- [27] Larry H. Thompson and David Schild, *The contribution of homologous recombination in preserving genome integrity in mammalian cells*, Biochimie **81** (1999), 87–105.
- [28] G.D. Van Duyne, *A Structural View of Cre-loxP Site-Specific Recombination*, Annu. Rev. Biophys. Biomol. Struct. **30** (2001), 87–104.
- [29] Mariel Vazquez and De Witt Summers, *Tangle analysis of Gin site-specic recombination*, Math. Proc. Camb. Phil. Soc. **136** (2004), 565–582.
- [30] J.C. Wang, *Interaction between DNA and an Escherichia coli protein  $\omega$ .*, J.Mol.Biol **55** (1971), 523–533.
- [31] ———, *Cellular roles of DNA topoisomerases: a molecular perspective.*, Nature Reviews Molecular Cell Biology **3** (2002), 430–440.
- [32] Katrina Wono, *Three Dimensional Analysis of TnpI Site-Specific Recombination*, Maters thesis, San Francisco State University (2012).