



STUDY OF POLYMORPHIC SICKLE  
HEMOGLOBIN POLYMERS

James Amarando

Advisor: Dr. Frank A. Ferrone

Submitted in partial fulfillment of the requirements  
for the degree of Bachelor of Science in Physics

Drexel University

Philadelphia, PA

May 2015

# 1 Abstract

Hemoglobin is a protein in red blood cells. Sometimes hemoglobin will abnormally form long polymer strands, which can cause sickle cell disease. These strands consist of stacks of hemoglobin molecules, connected via contact points on each molecule. These stacks have a slight twist to them within the polymer.

This project will test that if given an alternative set of axial contacts, a simulated polymer can be created that matches the electron microscopy of the sickle cell. This will be done via computer simulation involving both a quantitative way of checking contact points as well as a qualitative method of inspection.

A better understanding of where these molecules come together will lead to a better understanding of the disease. Knowing more about the contact points leads to a better framework for someone designing a drug to break up these polymers.

Advisor: Dr. Frank Ferrone

# Contents

<b>1</b>	<b>Abstract</b>	<b>2</b>
<b>2</b>	<b>Introduction</b>	<b>4</b>
<b>3</b>	<b>Approach</b>	<b>8</b>
<b>4</b>	<b>Methods</b>	<b>8</b>
4.1	Starting Point . . . . .	8
4.2	Algorithm Development - Coding Side . . . . .	9
4.3	Algorithm Development - Math Side . . . . .	11
4.4	Math Behind the Building Algorithm - Failed Attempts . . . . .	13
4.5	The Quaternions . . . . .	14
4.6	The Quaternions Relation to the Algorithm . . . . .	15
4.7	Determining the Value of the Flex Angle . . . . .	16
<b>5</b>	<b>Results - Planar Polymers</b>	<b>16</b>
<b>6</b>	<b>Discussion - Planar Polymers</b>	<b>17</b>
<b>7</b>	<b>Results - Twisted Polymers</b>	<b>20</b>
<b>8</b>	<b>Discussion - Twisted Polymers</b>	<b>21</b>
<b>9</b>	<b>Conclusion</b>	<b>22</b>

## 2 Introduction

Sickle cell disease is a very serious disease in which the body produces abnormally shaped red blood cells. Instead of being the normal doughnut shape, the mutation causes the red blood cells become long and thin, shaped like a sickle or crescent in the absence of oxygen. These blood cells are stiff and “sticky” and reduce blood flow to different parts of the body [1]. They can be thought of more as a brick rather than the bean bag of a healthy cell.

Researchers have tracked the cause of the mutation to a protein called hemoglobin. Hemoglobin (Hb) is an iron rich protein found in the red blood cell. Hemoglobin carries oxygen from the lungs to the rest of the body. It is made up of two alpha chains and two beta chains (figure 1).

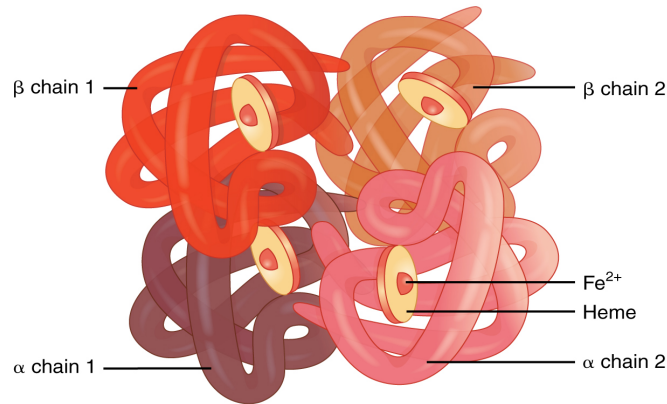


Figure 1: Hemoglobin structure of sub-protein chains [2]

Hemoglobin directly contributes to the shape of the red blood cell. In a healthy cell, the hemoglobins tend to just float around in the cell and do not polymerize. However someone who has sickle disease has a mutated form of hemoglobin called sickle hemoglobin (HbS).

HbS occurs when a glutamic acid in one of the beta chains is replaced by a valine [3]. This substitution is a direct cause of the ability of HbS to polymerize while Hb does not.

HbS has two states, oxygenated and deoxygenated. When HbS becomes deoxygenated the molecule tends to twist within itself. This twist causes a shift in the sub-protein chains which properly aligns the contact regions in order for polymerization to occur. For this reason all talks about HbS polymers are that of deoxy HbS.

Valine is a hydrophobic amino acid. Hydrophobic chains will move around due to an entropic

force in order to find a way that will allow them to not be in contact with water. A valine on one HbS molecule will then find another hydrophobic pocket along a neighboring HbS molecule. This hydrophobic interaction causes the two hemoglobin to stick together. This effect will take place over many hemoglobins, causing a polymer to form [4]. Figure 2 shows a cartoon drawing of a polymer.

Copyright © The McGraw-Hill Companies, Inc. Permission required for reproduction or display.

### Sickle Cell Anemia

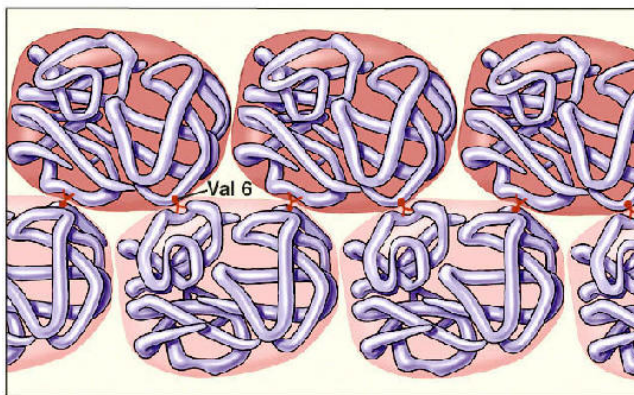


Figure 2: Hemoglobin polymerization. The valine of a sub-beta chain of the HbS connects to a hydrophobic pocket on the sub-beta chain of another HbS [5]

As stated previously, these strands of hemoglobin are very rigid. This is why this disease can be painful for those affected. There currently is no straightforward cure. The belief is that if we can get a better understanding of exactly what these polymers look like, a drug can be developed to break up these strands. This drug would introduce specific molecules into the blood stream, which would bond to the hemoglobin molecules and break them apart from each other. For this reason, the focus of the study is on the geometry of how the polymers are formed.

In the past, the polymers have been studied via electron microscopy and x-ray crystallography [6] [7] [8] [9]. These studies have given an understanding of what these polymers look like. They have shown these polymers are composed of pairs of stacks of hemoglobin molecules.

Each pair of strands is referred to as a double strand. When deoxy HbS is crystallized it is seen to be a linear set of double strands. When studied via electron microscopy it is seen that the double strands have a rotation about the axis. The twist is that of a helical shape, similar to a rope.

A commonly used way to depict these polymers is to represent each individual molecule as a sphere. This is called the "bead model" of hemoglobin. An example of a full polymer of HbS represented as a bead model is given in figure 3.

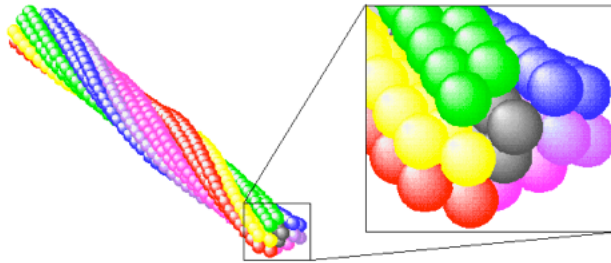


Figure 3: Sickle Hemoglobin Polymer Model [10]

However while these studies accurately depict the shape of the polymer, the methods cannot distinguish the exact orientation of each molecule. The focus of this thesis is looking at possible contact points.

The valine connections between each HbS are called the “lateral contacts”. They are lateral in the sense of they connect molecules within a cross section. These contact points are very well defined and are not the focus.

The focus is on the so called “axial contacts”. These are the contacts that connect one molecule to the one on top of it along the length of the polymer. The axial contacts are also hydrophobic interactions, but are much weaker than those of the laterals [4]. Figure 4 shows a cartoon of a polymer with axials and laterals labeled.

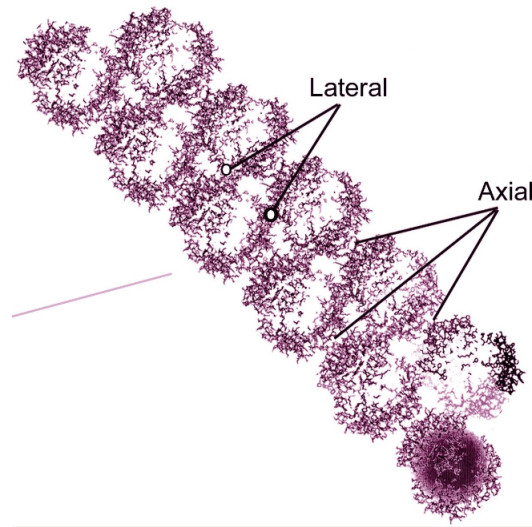


Figure 4: Cartoon of a polymer [11]

The currently accepted pairing of the axial contacts is a connection between a region in the alpha chain of one HbS and a region of a beta chain in the connecting HbS. This is based on the x-ray structure of double strands. This is shown in figure 5a. I will refer to these contacts as the “standard” axial contacts.

However, because there are two beta chains and not just one, it is possible to find a second pair of axial contacts. In this proposed pairing the partners are reversed. They are reversed in the sense that instead of an alpha of HbS one connected to the beta of HbS two, it is just a feasible to have the beta of HbS one connected to the alpha of HbS two. This proposed pairing would have the same molecules in contact. This new pairing was originally proposed by Bogdan Barz, Brigita Urbanc and Frank A. Ferrone (unpublished). It is shown in figure 5b.

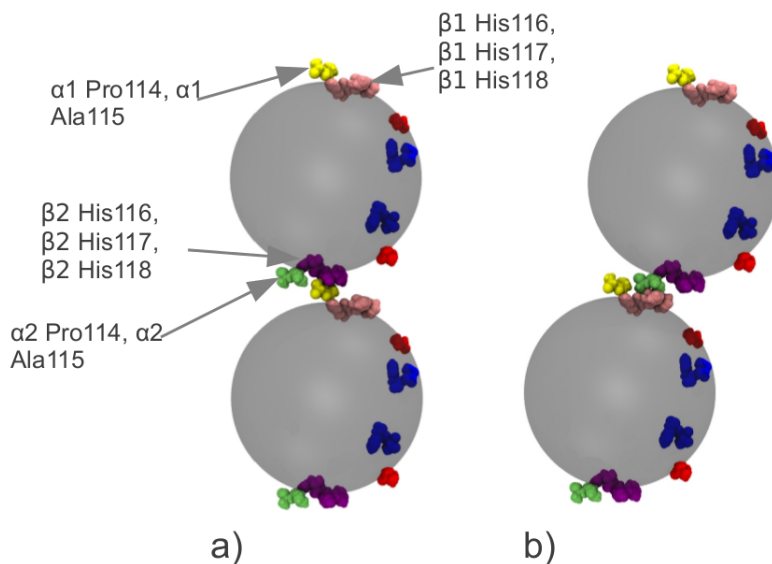


Figure 5: The “standard” axial contacts pairing (a) and the “alternative” proposed axial contacts pairing (b)

The focus of this project is to see if given the alternative set of axial contact pairings, a feasible model of a HbS polymer be formed. A feasible model of a polymer is a polymer with all contact points touching that has no physical violations.

### 3 Approach

This question is addressed via computer simulation. The goal is to have a computer program which generates a three dimensional bead model of a HbS polymer. These polymers are built by generating double strands using the lateral contact points as well as both sets of axial contacts.

As stated most previous studies have been done in a lab by analyzing strands of HbS. The resulting structure in the form of a bead model will be the metric to which the computer model will be compared. For the model to be feasible it should at the very least qualitatively look like that in figure 3.

The coordinates for the sets of axial and lateral contacts on one molecule were given. Also given was a set of essentially “repulsion” points. One metric of a feasible model is that these two repulsers should not be touching. This is one method that will help make this simple model relatable to a more realistic model. The repulsers will help avoid unphysical arrangements that may arise.

Currently all work done is purely on just one double strand. A “polymer” in use in the following sections is just referring to a pair of beads and not the full cross section of fourteen.

### 4 Methods

#### 4.1 Starting Point

The first step is to write code which will be able to generate one hemoglobin molecule with correct contact points. For the 3D image I decided to use the visual package in python. The output of one bead from the original data shown in figure 6.

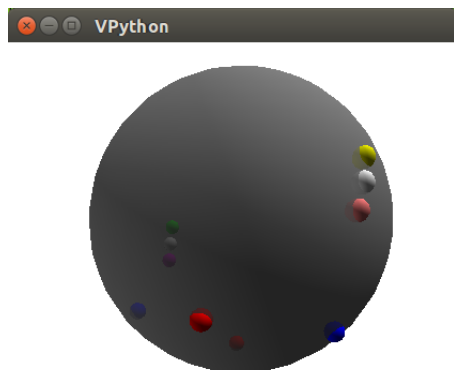


Figure 6: Single molecule in the bead model - initial data



The smaller spheres represent the contact points. With a similar color scheme to that of figure 5, the red spots represent the valine lateral contact. The blue represents the pockets the valine goes into. The purple and yellow spots represent the original set of axial pairs. The green and pink represent the alternative set of axial pairs.

Rotational cosine matrices were then used to rotate the bead to be facing “up and right”. This was done purely for aesthetic purposes. Figure 7 shows the result of the rotation. This orientation is the starting point from which all polymers will be formed. It also shows the code used to generate that bead

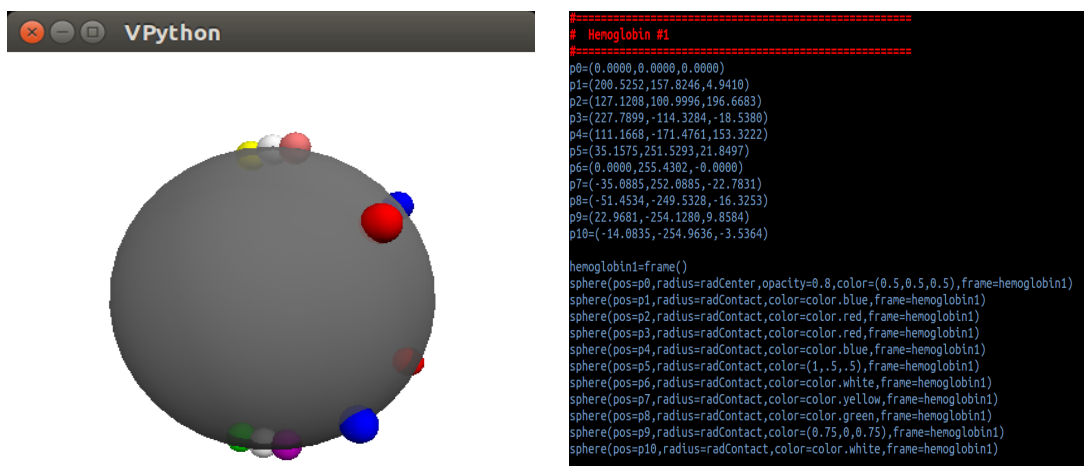


Figure 7: Left - starting bead for a polymer. Right - code used to generate the image.

## 4.2 Algorithm Development - Coding Side

Now that one bead is generated, an algorithm for generating more must be designed. It is immediately clear that it would be tedious to have to type in all the coordinates for each bead. Also it would be very limited if the code only allowed for a fixed number of beads to be generated. To allow for flexible use, I based my algorithm on a code generating code system. I will now describe some of the major components of the code.

At the end of the algorithm, a single python script (called polymer.py) must be executed in order to generate the images. This script is generated entirely by other code which appends strings into the end of that file. This way when building a polymer, it can be done with any number of beads and the user does not need to edit any code.

Above the polymer.py, there must be some code which generates that code. This script not only must do that, but also must bear the bulk of the weight of doing the math to position each bead within the polymer. This script I originally wrote in python. However after a few weeks of trying to figure out some of the methods of the math (to be explained shortly) I realized that too much time was wasted purely on the act of correctly coding through the matrices. After some thought I switched to Matlab.

The Matlab script (called makeBead.m) is a function which is given the indexed number of the bead to be generated and does some math to determine the (x,y,z) coordinates of the bead. The function then outputs the formatted strings to the end of the polymer.py.

Another Matlab script (called makePolymer.m) is a wrapper function around makeBead.m which directs it how many beads to generate and which set of axials to use.

Finally all the scripts and functions are surrounded by a bash wrapper (called run.sh). This wrapper is the only piece of code the user interacts with. It has various command line input options to control the polymer to be built. The high level overview of the algorithm is shown in figure 8.

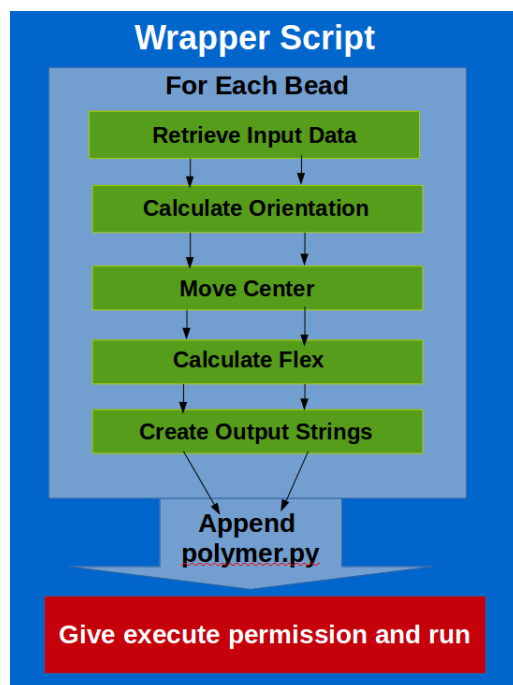


Figure 8: High level diagram of algorithm

### 4.3 Algorithm Development - Math Side

As hinted at in the previous figure, there is a very structured method for generating a polymer. The basic idea is to "connect" the next bead onto the previous using the lateral contacts. Once connected there is a "flex" put in about the lateral contact to make the next molecule line up its axials with the specified ones on the previous molecule. This is best to visualize when thinking of a system of three beads.

Firstly let's say you have one bead (figure 9 left) and want to add another. The code needs to first determine the bead's orientation. If it is an even indexed bead it is naturally going to be facing roughly opposite of an odd indexed bead. The code takes care of this by performing a rotation about the y axis by  $\pi$  radians if it is an even indexed bead. It will then have its acceptor lateral contact connect with the donor lateral contact. Then working backwards from the original data set, the center of the bead is calculated and all resulting contact locations are calculated. This is shown in the right of figure 9.

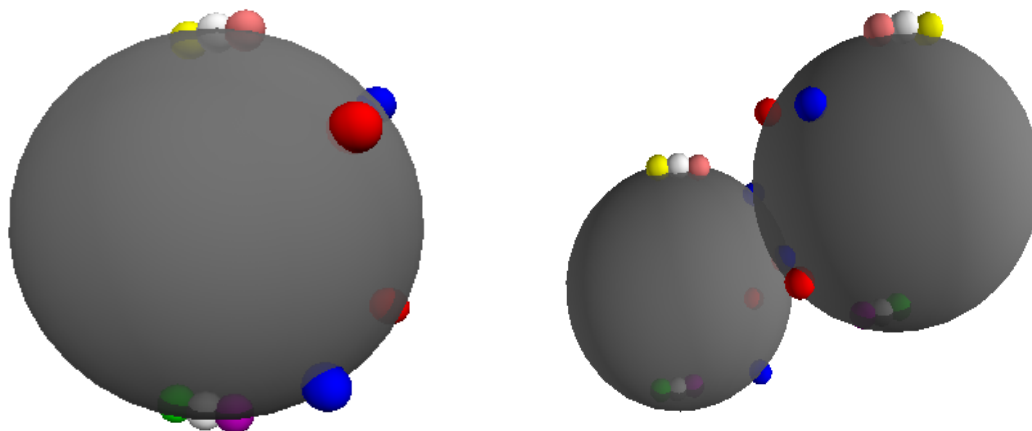


Figure 9: Left - starting bead for a polymer. Right - second added on.

The notion of the flex angle is the core reason any of this is possible. To illustrate this point, figure 10 follows the same method of adding a third bead onto the the second using the laterals. Notice that the two white repulsers are touching but both sets of axials are touching. This is an example of a model which is not feasible.

So clearly some thought must go into the flex at this. It is important to note that the whole reason this is possible is due to the fact that the lateral contact is a hydrophobic bond. Unlike an ionic bond

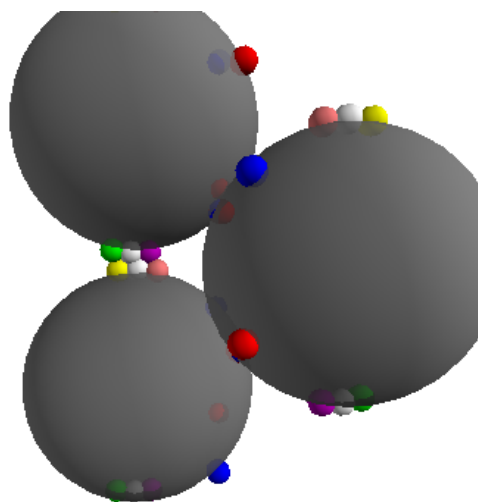


Figure 10: No flex in the lateral contact. This is NOT a feasible model

or a covalent bond, this type of bond is not as rigid. The bond is happy as long as the valine is not in contact with water. The result of this is that the bond acts similar to a ball and socket joint. The molecule is free to move about that joint to a certain degree.

Since this work is done with a bead model (and not a full model of the geometry of the molecule) the flex angle is deemed feasible so long as the surfaces of the center spheres do not touch each other. This flex angle can be viewed as an independent variable which is altered in order to achieve a feasible model.

A flex angle about the lateral contact of the even indexed beads of 18 degrees achieves a feasible model of connecting using the new axial contacts. Likewise a flex angle of -10 degrees will achieve a feasible model using the old contacts. These are shown in figure 11.

The following section will demonstrate the mathematical principles behind the flexing of the angle. At this stage this algorithm will result in a planar/linear strand instead of a twisted one.

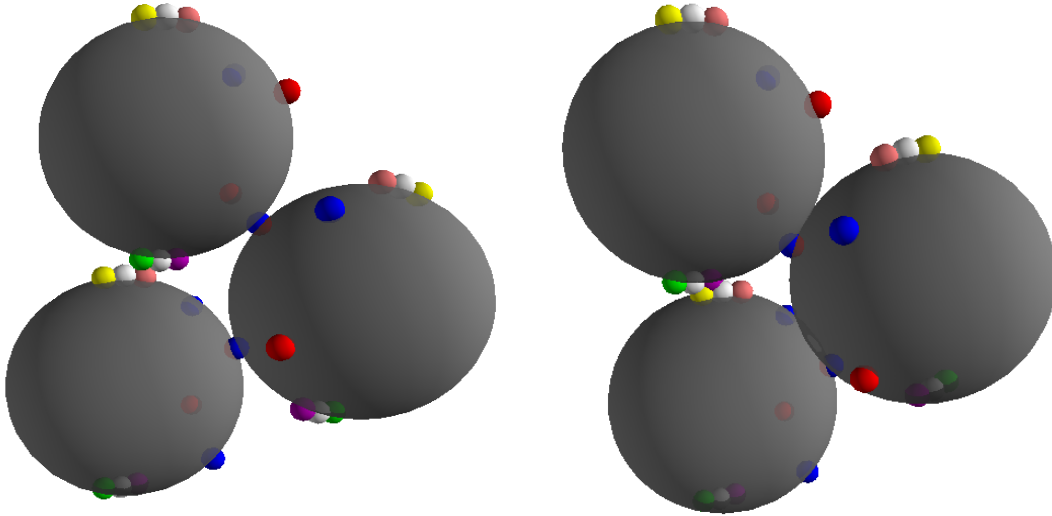


Figure 11: Left - new axials connected. Right - old axials connected.

#### 4.4 Math Behind the Building Algorithm - Failed Attempts

The math behind the algorithm was the most problematic of the project. These spheres are just general objects in three dimensional space. Each flex at a contact point has six degrees of freedom. Just the idea of moving the beads around is something very easy to show in the physical world, but does not translate easily into code.

The math behind the movement is something that is highly abstract and general. However the math behind the algorithm is something that needs to be funneled down to be highly specific. This is best highlighted by the flex about the lateral contact. This is a rotation at an arbitrary point about an arbitrary axis.

The first method tried was a bit of a work-around. In order to flex by an amount  $\theta$ , the bead was first rotated about the y-axis by an amount  $\phi$  so that the bead is in the  $x = 0$  plane. The bead was then flexed by  $\theta$  about the z-axis. It was then rotated back by  $-\phi$  to put it back in the original location. The whole thing summarized to:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = [R_y(-\phi)][R_z(\theta)][R_y(\phi)] \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

While this is a mathematically valid method of rotation, this did not end up working well for the system. It wound up essentially pulling the bead away from where it needed to go.

Attempts were then made at correcting this by having a second flex. This flex was about a different axis and its goal was to bring the axials back together. After a lot of tweaking around the algorithm wound up not working.

After further research into rotations, a method of using the quaternions was developed. First a brief background on the quaternions is in order.

## 4.5 The Quaternions

Normally in basic complex analysis the complex plane is discussed as two axis, one being the Real line and one being the Imaginary line. This is basically a one dimensional representation of complex numbers.

The quaternions are however a three dimensional representation of the complex numbers [13]. So notationally, in stead of just having one imaginary number  $i$ , you instead have  $i, j, k$ . They were first conjectured by Hamilton in 1843 [14], for the purpose of analyzing three dimensional mechanics. The most notable property of the Quaternions is that multiplication is not commutative. For example:

$$\begin{array}{lll} ij = k & jk = i & ki = j \\ ji = -k & kj = -i & ik = -j \end{array}$$

Also they have the property that  $i^2 = j^2 = k^2 = -1$ .

Using all these properties it is possible to derive a rotation matrix using the quaternions. The difference between this type of rotation matrix (denoted  $R_n(\theta)$ ) instead of the standard Cartesian rotation matrices is that this matrix facilitates a rotation about an arbitrary axis  $\hat{u}$  instead of one of the (x,y,z) axes.

For the scope of this project, the derivation is not explicitly written out. The final rotation matrix  $R_n(\theta)$  is given in figure 12. This is a rotation about the  $\hat{u}$  axis where  $\hat{u} = (u_x, u_y, u_z)$  where  $(u_x, u_y, u_z)$  are the projection of  $\hat{u}$  into the Cartesian axes.

$$R = \begin{bmatrix} \cos \theta + u_x^2 (1 - \cos \theta) & u_x u_y (1 - \cos \theta) - u_z \sin \theta & u_x u_z (1 - \cos \theta) + u_y \sin \theta \\ u_y u_x (1 - \cos \theta) + u_z \sin \theta & \cos \theta + u_y^2 (1 - \cos \theta) & u_y u_z (1 - \cos \theta) - u_x \sin \theta \\ u_z u_x (1 - \cos \theta) - u_y \sin \theta & u_z u_y (1 - \cos \theta) + u_x \sin \theta & \cos \theta + u_z^2 (1 - \cos \theta) \end{bmatrix}$$

Figure 12: Quaternion derived rotation matrix

## 4.6 The Quaternions Relation to the Algorithm

The method by which the flex is performed is by using this quaternion derived rotation matrix in the above figure. This equation is essentially all ready to plug into the code in order to flex. The only thing left to do is to calculate the rotation angle.

The axis of this rotation from a physical point of view is extremely arbitrary. Once again, the molecule is free to rotate in any direction at that lateral contact. That means the choice of this vector is both very easy and very hard to get correct. It is easy in the sense of whatever axis is chosen will probably represent a feasible flex. However, that axis in no way is the only option and may turn out to be irrelevant in the scheme of things.

The choice of this axis for me was thought out from a mathematical perspective. Figure 13 is just a zoomed in view of figure 10. I want to highlight the axial contacts. These are going to be the indicator I use to choose the flex angle. As a reminder, the setup shown in figure 10/13 is not a feasible model. This was achieved by just attaching beads using the lateral contacts and by not putting in any flex at all

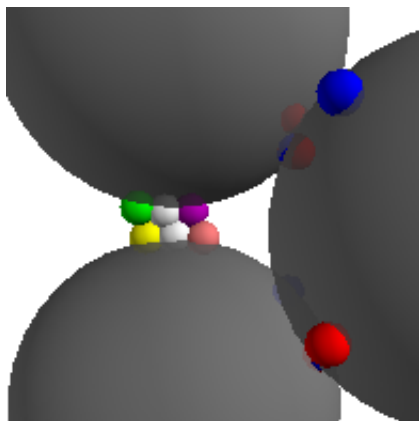


Figure 13: No flex in the lateral contact. This is NOT a feasible model

The end goal is to either line up the green and the pink (for the case of the “new” contacts) or to line up the purple and yellow (for the “old” contacts). For the ease of discussion I am only going to

work with assembling the new contacts. The other set can be generated in a very analogous way.

The first step is to create a vector in the axis of movement. This is just a vector created by subtracting the location of the pink contact minus the location of the yellow contact. This vector will be orthogonal to the direction of rotation. This axis can be referred to as  $v$ .

A second axis which will be orthogonal to the first is one that will go radially straight down through the sphere. As I discussed earlier I rotated the data so that the bead was oriented in this way. This rotation was chosen so that the white sphere (the repulser) in between the two axials is set on the pole. Knowing this, I create my second vector  $w$  to be a vector going from this pole straight down into the center of the bead.

The third vector  $u$  is then simply the cross product of  $v$  and  $w$ . This axis  $u$  is the axis which will be flexed about. It is normalized to become  $\hat{u}$  and then can be plugged into  $R_{\hat{u}}(\theta)$ .

## 4.7 Determining the Value of the Flex Angle

This is a very interesting part of the algorithm. There are two ways that this can be performed. The first way is to analytically predict what it should be using geometry. While this problem is possible to solve, given all the degrees of freedom of the rotation, coupled with the fact that the code needs specific Cartesian coordinates, I deemed that this method would be too time consuming.

Instead a more experimental method was used. Since I had code to perform the rotations for me, I felt it would be more efficient to just try different values of  $\theta$  and see what the result looked like.

After trying many of angles in this method, a flex angle of 18 degrees was determined to be sufficient. As for matching up the old contacts, a flex angle of -10 degrees was sufficient. The results of this are pictured in figure 11.

## 5 Results - Planar Polymers

Currently two polymers of a double strand have been formed. One was using the new contacts, one was using the old contacts. The Matlab scripts builds them up following the pattern described in the methods section. The even number molecules are the only ones that are actively flexed at the lateral contacts.

This is not a violation because a flex of 18 degrees of the second bead and 0 degrees of the first and third beads can also be seen as a flex of 9 degrees amongst all the beads. The flip side is, from a



coding perspective, that with this method every odd indexed bead is essentially just the initial bead with a linear shift of its center.

The result of all this is that the double strands for both the new contact sets and the old contact sets are planar. The following figure is the result of generating polymers of length 16 beads.

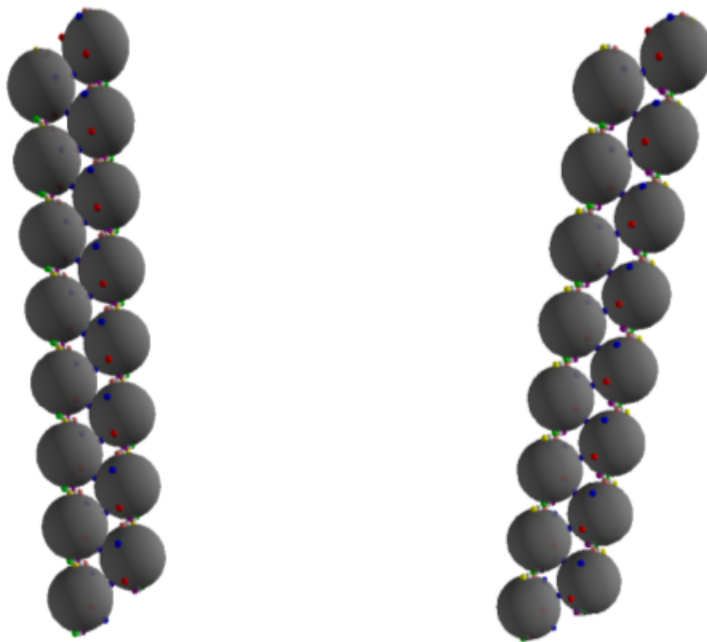


Figure 14: Left - standard axial used. Right - alternative axial used.

## 6 Discussion - Planar Polymers

The current results only generate a planar strand. While this is accurate for the case of crystallized HbS, it is known through electron microscopy that there is a twist in the polymer. The next step is to try to figure out how this twist can be formed.

It is known from previous work [14] that it is definitely possible to have this twist using these contacts. A styrofoam model was built using the coordinates. The beads were stuck together using toothpicks. Figure 15 is the result of this.

The problem now is how does this twist seem to intrinsically form when using the styrofoam balls, but when the exact same contact points are used in the computer simulation something planar is formed? First I decided to look at my bead model a little closer to take a look at the angles at which



Figure 15: Styrofoam model - R. Klaiss and F. A. Ferrone (unpublished)

these molecules meet each other.

To get a better understanding of these angles, I decided it would be easier to look at these as vectors rather than spheres. I added an option into my code to make an “arrow” model of the polymer rather than the bead model. The arrow model contains the same information as the bead model, but differs in visualization.

One visual change is that the center sphere is shrunk down into the same size as the contact spheres. Also I have added four radial vectors to each molecule. These are represented as arrows that go from the center bead to the axial and lateral contacts being used in each molecule.

Figure 16 shows an arrow model using the standard axials, and figure 17 shows an arrow model using the alternative axials. Each figure has two different angles of the same polymer. Also each figure is a polymer of three molecules. The structure of the polymer is not changing as the number of molecules increases, so I chose three molecules for ease of analysis.

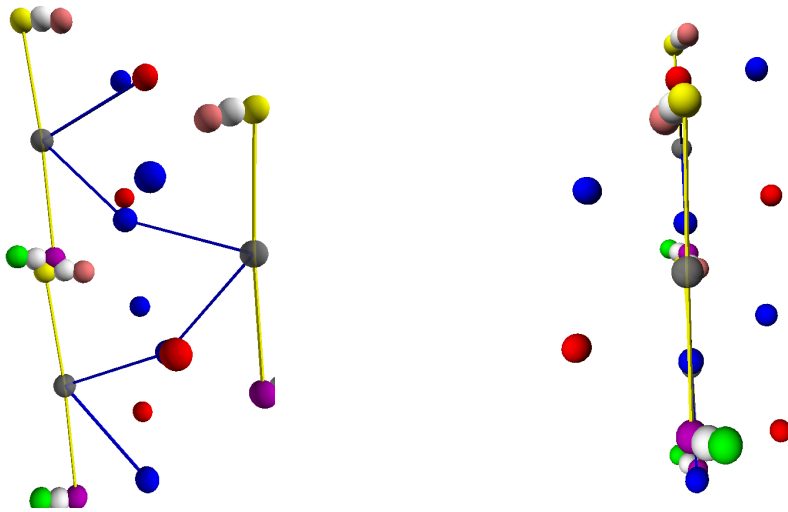


Figure 16: The arrow model using the standard contact points

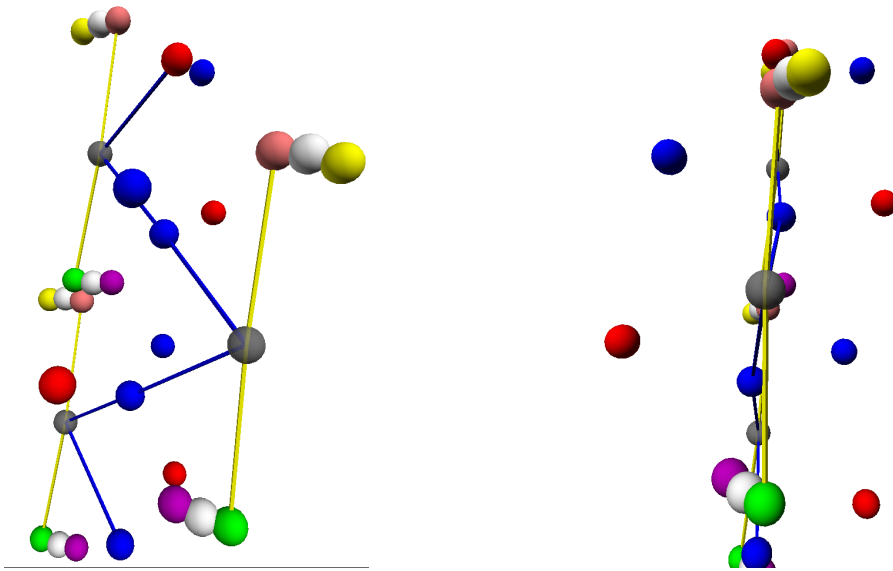


Figure 17: The arrow model using the alternative contact points

There are two things to note about these figures. Look at the left picture in each figure. It is interesting how the model of the alternative axials seems to line up so perfectly. It essentially has the two lateral arrows meeting with almost no angle in between them.

Looking at the right picture in each figure there is something very subtle to notice. The model using the standard axials shows that the arrows are exactly planar. However looking at the same picture for the alternative axials shows that it is not exactly planar. From this viewing angle of the model you can see a “zig-zag” of the laterals bouncing in and out of the plane. This zig-zag of the laterals on the model of the new axials shows that the polymers of the new axials must have some sort of non-planar arrangement.

## 7 Results - Twisted Polymers

Following from the discussion of the stick model, the goal is to try to reconcile the twisted polymer model with the planar model. The planar model can be thought of as two pieces of string hanging next to each other. A twisted polymer can be formed by grabbing the ends of the strings and beginning to turn the ends. This is essentially performing a rotation about an axis which is directly in between both pieces of string.

In the discrete case (like the polymer) this is equivalent to performing the described twist among each molecule by the same recursive amount (more on that in a moment). Specifically in the case of the polymer, the axis of rotation is modeled to be a vector going between the two lateral contacts being used. These laterals act as basically a hinge. Let  $\phi$  be the angle by which each bead is rotated. This angle is recursive, meaning each bead is rotated along that axis by an amount  $n \cdot \phi$  where  $n$  is the indexed number of the bead in the polymer.

Using a value of  $\phi = 4.1^\circ$ , a polymer of 44 beads was formed (see Discussions - Twisted Polymers for motivations). The resulting polymer is shown in figure 18.

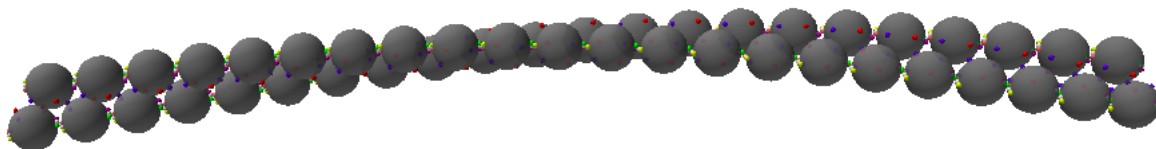


Figure 18: Twisted polymer of the bead model using alternative contacts

## 8 Discussion - Twisted Polymers

Electron microscopy provides the experimentally measured twist of an HbS polymer [20] (figure 19). The measurements showed that a polymer goes through a  $180^\circ$  rotation after approximately 1400 angstroms. Knowing that the diameter of a hemoglobin is approximately 64 angstroms, it can be determined that this is equivalent to a stack of 22 molecules high, or 44 beads total in a double twist.

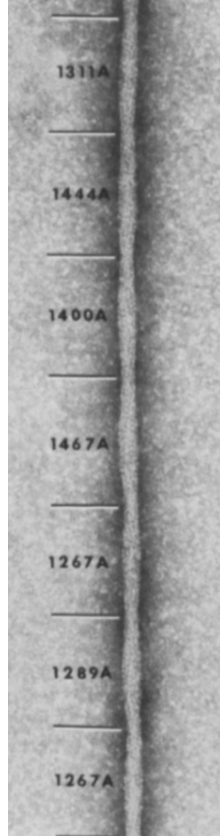


Figure 19: Electron microscopy analysis of a HbS polymer. The distance between marks is the distance of a  $180^\circ$  twist. [20]

The next parameter to decide is the angle  $\phi$ . The value of this angle was decided by arbitrarily trying a range of angles until one gave the correct twist (i.e. the twist of  $180^\circ$  over the polymer of 44 total beads. This method gave a value of  $\phi = 4.10^\circ$ . As a reminder, this angle is the value of rotation about the lateral axis which each bead is rotated from the previous bead. In closed form this is  $\{n \cdot \phi | n = 0, 1, \dots, 43\}$ . The resulting polymer of 44 beads with a twist of  $4.1^\circ$  per bead is pictured in figure 18.

Bluemke et al [20] determined that the twist of each pair of molecules on top of the previous pair is equivalent to approximately  $7^\circ$ . In equivalent notation, the angle determined by this code was  $8.2^\circ$ .

## 9 Conclusion

Through all my troubleshooting with this I have come to realize that this highlights a very specific type of optimization. When someone is building a model out of styrofoam balls the whole system will optimize together. In other words if I were to gently pull on one ball in the polymer, all the other balls would simultaneously move in order to keep the polymer together.

When I am coding however, if I move just one bead around only that bead will move. This type of optimization is a stream type. If I change one thing I need to go through and individually fix every other one.

I have shown with the code that I have written that using the alternative contact points, a planar (or crystal) structure of a HbS polymer can be formed without any physical violations. I have also shown that this model can then be twisted to form a very close model of a twisted strand. This shows that the model I created does not necessitate a twist but does accommodate a twist.

While Bluemke et al [20] observed a slightly different twist angle than the one I generated, the difference can be attributed to the fact that the bead model is just a cartoon of a complicated physical system. However since the difference is small, it shows that even though the model is not perfect, it can be used under certain circumstances to represent the real model.

In the long run, improved models of the polymer can be used to suggest locations where a drug or other agent might disrupt the polymer and provide a therapy for the disease.

## References

- [1] *What Is Sickle Cell Anemia?* National Heart, Lung, and Blood Institute September 2012  
<http://www.nhlbi.nih.gov/health/health-topics/topics/sca>
- [2] (figure 1) <http://www.bio.miami.edu/cmallery/150/chemistry/hemoglobin.jpg>
- [3] Dickerson, Geis *Hemoglobin: Structure, Function, Evolution, and Pathology*  
Benjamin/Cummings Publishing Company Inc, 1982.
- [4] Frank A. Ferrone, Yihua Wang,  
*Dissecting the Energies that Stabilize Sickle Hemoglobin Polymers.* Biophys J. 2013 Nov 5;  
105(9): 21492156  
<http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3824546/>.
- [5] (figure 2) <http://faculty.southwest.tn.edu/rburkett/GB%20Genetics.htm>
- [6] Stanely J. Watowich, Leon J. Gross, and Robert Josephs,  
*Analysis of the intermolecular contacts within Sickle Hemoglobin Fibers: Effect of Site-Specific Substitutions, Fiber Pitch, and Double-Strand Disorder.* J Struct Biol. 1993  
Nov-Dec;111(3):161-79.  
<http://www.ncbi.nlm.nih.gov/pubmed/8003379/>.
- [7] Carragher B, Bluemke DA, Gabriel B, Potel MJ, Josephs R,  
*Structural Analysis of Polymers of Sickle Cell Hemoglobin.* Mol Biol. 1988 Jan 20;199(2):315-31.  
<http://www.ncbi.nlm.nih.gov/pubmed/3351926/>,
- [8] Daniel J. Harrington, Kazuhiko Adachi, William E. Royer Jr,  
*The High Resolution Crystal Structure of Deoxyhemoglobin S.*  
J Mol Biol. 1997 Sep 26;272(3):398-407.  
<http://www.ncbi.nlm.nih.gov/pubmed/9325099>
- [9] Wajcman H., Galacteros F.,  
*Sickle cell disease: laboratory and hemoglobin study.* Bull Soc Pathol Exot. 2001 May;94(2):80-4  
<http://www.ncbi.nlm.nih.gov/pubmed/11475032>.

- [10] Frank A. Ferrone, Drexel University  
[http://www.physics.drexel.edu/~ferrone/Sickle\\_Cell\\_files/polymer.png](http://www.physics.drexel.edu/~ferrone/Sickle_Cell_files/polymer.png)
- [11] Wang Z., Kishchenko G, Chen Y, Josephs R.,  
*Polymerization of deoxy-sickle cell hemoglobin in high-phosphate buffer.* J Struct Biol. 2000  
Sep;131(3):197-209  
<http://www.ncbi.nlm.nih.gov/pubmed/11052892>.
- [12] Shuchismita Dutta, David Goodsell May 2003 Molecule of the Month - PDB  
<http://www.rcsb.org/pdb/101/motm.do?momID=41>
- [13] Harrington DJ1, Adachi K, Royer WE Jr. *Crystal structure of deoxy-human hemoglobin beta6  
Glu -> Trp. Implications for the structure and formation of the sickle cell fiber*  
J Biol Chem. 1998 Dec 4;273(49):32690-6  
<http://www.ncbi.nlm.nih.gov/pubmed/9830011?dopt=Abstract>
- [14] Klaiss and Ferrone (unpublished)
- [15] Eaton, William A,  
*Linus Pauling and Sickle Cell Disease* , Biophysical Chemistry 100, 2003
- [16] R H Crepeau, S J Edelstein, M Szalay, R E Benesch, R Benesch, S Kwong, and R Edalji,  
*Sickle cell hemoglobin fiber structure altered by alpha-chain mutation*  
Proc Natl Acad Sci U S A. 1981 Mar; 78(3): 14061410  
<http://www.ncbi.nlm.nih.gov/pmc/articles/PMC319139/>.
- [17] Liliana Sokolov, Ishita Mukerji  
*Structure of Sickle Cell Hemoglobin Fibers Probed with UV Resonance Raman Spectroscopy*  
J. Phys. Chem. B, 2000, 104 (46), pp 1083510843  
<http://www.sicklecellinfo.net/Lab%20personnel/sokolov%20and%20mukerji%2000.pdf>.
- [18] John Huerta  
*Introducing the Quaternions*  
<http://math.ucr.edu/huerta/introquaternions.pdf>
- [19] Jack B. Kuipers *Quaternions and Rotation Sequences*  
<http://www.emis.ams.org/proceedings/Varna/vol1/PGEOM09.pdf>, 2000



- [20] David A. Bluemke, Bridget Carragher and Robert Josephs,  
*The Reconstruction of Helical Particles with Variable Pitch* Ultramicroscopy 26 (1988) 255-270.