Statistical Shape Analysis of Helices

Mai F. Alfahad

A thesis submitted in accordance with the requirements for the degree of Doctor of Philosophy

University of Leeds
School of Mathematics

April 2018

The candidate confirms that the work submitted is her own and that appropriate credit has been given where reference has been made to the work of others.

Acknowledgements

My greatest appreciation and gratitude go to both my supervisors, Prof. John Kent, and Prof. Kanti Mardia, for their invaluable guidance and advice throughout my PhD research. It was a pleasure working with them.

In addition, I wish to thank all my friends and colleges in school of mathematics at university of Leeds for their wonderful collaboration, friendship and numerous interesting discussions.

Finally, thanks goes to my mother, husband, sister, grandmother, aunty, cousins and my friends in Kuwait for their unconditional love, support and cheering me up throughout my entire life.

Thank you very much, everyone!

Mai Alfahad
Leeds university,
April 2018

Abstract

Consider a sequence of equally spaced points along a helix in three-dimensional space, which are observed subject to statistical noise. In this thesis, maximum likelihood (ML) method is developed to estimate the parameters of the helix. Statistical properties of the estimator are studied and comparisons are made to other estimators found in the literature.

Methods are established here for the fitting of unkinked and kinked helices. For an unkinked helix an intial estimate of a helix axis is estimated by a modified eigen-decomposition or a method from the literature. Mardia-Holmes model can be used to estimate the initial helix axis but it is often not very successful one since it requires initial parameters. A better method for initial axis estimation is the Rotfit method. If the the axis is known, we minimize the residual sum of squares (RSS) to estimate the helix parameters and then optimize the axis estimate. For a kinked helix, we specify a test statistic by simulating the null distribution of unkinked helices. If the kink position is known, then the test statistic approximately follows an F-distribution. If the null hypothesis is rejected i.e. the helix has a change point, and then cut the helix into two sub-helices between the change point where the helix has the maximum statistic. Statistics test are studied to test how differ these two sub-helices from each other. Parametric bootstrap procedure is used to study these statistics. The shapes of protein α-helices are used to illustrate the procedure.

Contents

List of Figures

List of Tables

Index of Notation

$\mathbb{R}$	Real numbers.
n	Number of points.
n_1	The helix initial point at n_1.
n_2	The helix last point at n_2.
t	Time.
δ	The spacing parameter.
$\boldsymbol{w}$	The helix axis.
$\boldsymbol{w}^{(\ell)}$	The ℓ^{th} sub-helix axis.
$\boldsymbol{w}_0 = [0, 0, 1]^T$	The helix axis point to north pole.
Γ_1	The rotation matrix which rotate the helix axis to $[0, 0, 1]^T$.
Γ_2	The rotation matrix which rotate the helix about the axis.
$\boldsymbol{w}_1^T = \boldsymbol{w}^T\Gamma_1$	The helix axis of the vertical helix after stage 1.
$\boldsymbol{u}, \boldsymbol{v}$	The orthogonal vectors perpendicular to the axis.
$\boldsymbol{u}^{(\ell)}, \boldsymbol{v}^{(\ell)}$	The ℓ^{th} orthogonal vectors perpendicular to the axis.
r	The radius of the helix.
$r^{(\ell)}$	The radius of the ℓ^{th} sub-helix.
$2\pi c$	The pitch of the helix.
$2\pi c^{(\ell)}$	The pitch of the ℓ^{th} sub-helix.
$\boldsymbol{b}$	The location vector of the helix.
$\boldsymbol{b}^{(\ell)}$	The location vector of the ℓ^{th} sub-helix.
τ	The angle between the initial point of the helix and the point $(r, 0, 0)$.
$\boldsymbol{y}_i$	The i^{th} point of the helix.
$\boldsymbol{y}_i^{(\ell)}$	The i^{th} point in the ℓ^{th} sub-helix.

$\boldsymbol{y}_i'$	The i^{th} point of centered helix.
H	The $n \times 3$ data helix matrix.
H^o	The $n \times 3$ data helix matrix in semi- canonical coordinates.
k	The position of the change point.
$H_k^{(\ell)}$	The ℓ^{th} sub-helix.
atan2	The function of two arguments arctan function which returns angle in radians between $-\pi$ and π.
κ	The concentration paramater.
Σ	The variance covariance matrix.
df	The degrees of freedom.
$df_T = 3n - 8$	The residual degrees of freedom after fitting a single helix.
$df_W = df_T - 8$	The residual degrees of freedom after fitting a bent helix.
$df_B = df_T - df_W$	The between degrees of freedom.
SST	The total residual sum of squares.
SSW_k	The within residual sum of squares.
SSB_k	The between residual sum of squares.
$\hat{\sigma}^2 = \frac{SSB_k + SSW_k}{df_B + df_W}$	The residual variance.
$\hat{\sigma}_p^2 = \frac{SSW_k}{df_W}$	The pooled residual variance.
$F_k = \frac{SSB_k/df_B}{SSW_k/df_W}$	The function for the F-statistic.
$F_{\max}$	The maximum value of F_k.
$F_{\max}^*$	The value of $F_{\max}$ for * simulated bootstrap sample.
$F_{\max}^{*(\alpha)}$	The threshold value at $(1-\alpha)\%$ quantile.

Index of Notation

OptLS	The fitting method of a regular helix.
Bending-Detector	The analysing process of a bent helix.
The change point phase of Bending-Detector	Estimate the change point.
The testing phase of Bending-Detector	Test for the presence of a change point.
The features analysis phase of Bending-Detector	Investigate the reasons for the change point .
Kink	A region of points which represents gradual change in the helix axis direction.
Bending point	A single point which represents sharp change in the helix axis direction, see Chapter 5.

Chapter 1

Introduction

1.1 Overview

Polypeptides, or proteins, are biomolecule compounds that consist of a chain of amino acid residues, which are of particular importance as they can be found in every living organism. An amino acid is formed of four parts: the amine (NH_2), the carboxyl (COOH), the R groups, and a hydrogen (H) atom all attached to the carbon C_α atom. In a protein, two adjacent amino acids are bonded together (a peptide bond) and a H_2O (water) is released; these amino acids are then called amino acid residues. The primary structure of the protein has two components: the main chain made up of carbon, nitrogen and oxygen atoms; and the side chain of R-groups which may differ from one residue to another. There is a remaining connected sequence $NC_\alpha CNC_\alpha C\ldots$ of carbon and nitrogen atoms, which form one of three possible 3-dimensional curves (secondary structure), see Section 1.4.1. Of these, we are interested only in the α-helix and the C_α atoms for the purposes of this thesis since this is the most common structure. We wish to study the shape of a helix and, more generally, a helix with kinks (i.e. points where the helix axis changes direction see Wilman et al. (2014a)), since kinks are functionally important in membrane proteins.

Many researchers have developed their own procedure for meeting the following objective: given a helix, determine if it is kinked; and, if so, find the position of the kink, see for example, the Kinkfinder by Wilman et al. (2013) and Kink-Detector by Mardia et al. (2018). In this thesis we produce a new procedure, called the *Bending-Detector*, to meet the same objective. The Kinkfinder uses all the atoms on the helix, whereas the Kink-Detector and our Bending-Detector use the C_α atoms. On the other hand, Kinkfinder and Kink-Detector give a region of points as a kink position, but with the Bending-Detector we identify a single point. In order to draw a meaningful comparison between Bending-Detector and Kink-Detector by Mardia et al. (2018), we used the same data helices as these used by Mardia et al. (2018).

For the remainder of this section we outline in more detail the content of each chapter. In this chapter, we begin by defining the mathematical and statistical models for the helix and specify some special cases. In addition, we review some background materials on the structure of a protein, and in particular the α-helix which appears most frequently as the secondary structure.

Chapter 2 is dedicated to matrix algebra. The purpose of this chapter is to review some of the theory that is used in this thesis and to provide more details where desirable. In Section 2.1 we present how to obtain the first perturbation of an eigenvalue from a 3×3 square symmetric matrix that has been perturbed. In Section 2.2 we discuss the full and reduced forms of the spectral decomposition (SD) and the singular value decomposition (SVD) of a given matrix. However, the standard version of the SD is not quite unique because: firstly, we may always make choices of signs for the eigenvectors; and secondly, the matrix may have an eigenvalue of multiplicity at least 2, which implies that the corresponding eigenvectors are not unique (see Section 2.2.1.1 for more details). More importantly, a unique version can be constructed using projection matrices. Furthermore, we show how one of these decompositions can be derived from the other in Section 2.2.3. In the case of a square matrix, we modify the SVD to obtain three

more related decompositions; and we demonstrate how one of these may be used in matrix optimization. The last section, 2.3, discusses the Cholesky decomposition (CD) that is used in the special case when the given matrix is symmetric positive definite.

In Chapter 3 we fit the straight or as we call it *regular* helix, which begins by estimating the helix axis. We start by presenting our modified principal component method for estimating the helix axis, we called it *Difeigenfit*. We also investigate three of the five methods studied by Christopher et al. (1996) for estimating the unkinked (straight) helix axis. These are the parametric least squares (Parlsq) method; the Eigenvector (Eigenfit) method; the Rotational least squares (Rotfit) method. In Section 3.5, we develop our own method, which we call "Optimized least squares" (OptLS), for estimating the helix parameters. The OptLS method consists of three stages: in stage 1 we estimate an initial helix axis; by knowing the helix axis, in stage 2 we estimate the other parameters by least squares and get the residual sum of squares (RSS); and in last stage we improve the helix axis estimation by optimizing this RSS function over the helix axis. In Section 3.6, we compare the OptLS method to the three methods mentioned above (Christopher et al., 1996). We conclude from simulation studies that the Parlsq and Eigenfit methods give a poor initial estimation of the helix axis, whereas Rotfit and Difeigenfit give a much better initial estimation. However, our OptLS iterative method outperforms any initial estimate of the axis, which is evident from a comparison of the variances of the estimated helix axes (see Table 3.3). In Section 3.8 we analyse Difeigenfit by finding the asymptotic distribution, using perturbation theory, of the variance of Difeigenfit helix axis estimate. In addition, we fit our OptLS to real α-helix data in Section 3.9, and to a simulated cone helix in Section 3.10, in order to study further how accurate the OptLS method is.

Another way to estimate the helix axis could be the Mardia-Holmes (M-H) model, (see Mardia and Holmes, 1980), in Chapter 4. We start by investigating

the fitting of an ellipse (and, as a special case, a circle) using the M-H model. We develop a series of programs to estimate the parameters (location, concentration, ellipse major and minor axis) of an ellipse under various restricted versions of the M-H model. The first of these programs assumes just concentration is unknown, whereas the last supposes all the parameters are unknown. These programs work by iteratively optimizing the likelihood given starting values of the parameters, and we are able to make these programs work faster in R by building a new unconstrained parametrization for some of the parameters. Overall, we find that the maximum likelihood estimate (MLE) of the concentration noticeably increases as we increase the numbers of unknown parameters in our model. In addition, we prove that the asymptotic distribution of the M-H model under high concentration is normal in Section 4.5. The M-H model can be used to estimate a helix axis, which we explain in Section 4.6, and by simulation we find that the OptLS method is still a better method.

In Chapter 5, we develop a strategy that depends on the likelihood ratio test (statistic) which is based on the likelihood ratio of the null model (regular helix) to the alternative model (bent helix). This allows us to decide if a given helix has a change point (bent helix) or not. If we assume the position of the potential change point is known, then this test statistic approximately follows an F-distribution (see Section 5.2). For a large value of this test statistic we reject the null hypothesis and conclude that the helix has a change point. The threshold is specified by simulating from the null distribution and the threshold value is the $(1-\alpha)\%$ quantile of the simulated test statistics. If the null hypothesis is rejected, we look at 6 features at the change point. We call this procedure *Bending-Detector*.

The regular helix is fitted using OptLS in Chapter 3. For a bent helix, we work on a helix that is known to contain at least one kink and determine the change point(s) of this helix. Our Model is essentially a classic Change Point model (see e.g. Chen and Gupta, 2011, pp. 7-35). In addition, we give a direct

comparison between our Bending-Detector procedure and the Kink-Detector by Mardia et al. (2018).

1.2 Helix geometry

Here we begin by defining the mathematical helix (continuous or discrete) at various levels of standardization (canonical coordinates, semi-canonical coordinates, and general coordinates); we then introduce a statistical helix incorporating errors. The discrete statistical helix is fitted to protein α-helix data.

1.2.1 The mathematical helix model

A mathematical helix is a smooth curve spiralling along an axis $\boldsymbol{w}$ in 3-dimensional space, where the helix axis direction is $\boldsymbol{w}$. Then a helix can be defined as a function of an independent variable t, as

$$\boldsymbol{f}(t) = r\cos(t)\boldsymbol{u} + r\sin(t)\boldsymbol{v} + ct\boldsymbol{w} + \boldsymbol{b}, \tag{1.1}$$

where

- $\Gamma = \begin{bmatrix}\boldsymbol{u} & \boldsymbol{v} & \boldsymbol{w}\end{bmatrix}$ is an orthogonal matrix whose three columns define the $frame$ of the helix. In particular the vector $\boldsymbol{w}$ defines the *helix axis*, and the vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ define the plane normal to the helix axis, where $\boldsymbol{u}$ is the direction of the initial point (i.e. at time t=0). These vectors are called *orientation* parameters.
- $r > 0$ is the *helix radius*,
- $2\pi c > 0$ is the *helix pitch*,

- $\boldsymbol{b} \in \mathbb{R}^3$ is an *intercept*, which is a shift vector (location parameter),

and the constant speed t denotes "time" , where t increases along the curve (see e.g. O'Neill, 1997, p. 16).

The parameters of a helix can be divided into two types. The *registration* parameters are the orthonormal 3×1 vectors $\boldsymbol{u}, \boldsymbol{v}$ and $\boldsymbol{w}$, and the 3×1 vector $\boldsymbol{b} = [b_1, b_2, b_3]^T$. The *shape* parameters are the radius of the helix r, and the pitch c. Helices can be regarded as right-handed or left-handed depending on whether $\det(\Gamma) = +1$ or -1, respectively.

There are two helpful ways to think about whether the helix is right or left-handed. First find the helix axis $\boldsymbol{w}$, together with the direction determined by increasing t. Position the axis so that it is perpendicular to the plane of your face with the smallest value of time nearest to you. Thus, the helix axis moves away from you as time increases. Next consider points on the helix; if increasing time induces a clockwise screwing motion, then it is called a right-handed helix; otherwise, it is a left-handed helix, see Figure 1.1, where the eye is at the bottom of the figure and the helix axis moves upward as time increases.

Alternatively, for a right-handed helix take your right hand and identify the three orthonormal directions $\boldsymbol{u}, \boldsymbol{v}$ and $\boldsymbol{w}$ with the first finger, the second finger, and the thumb, respectively, where the fingers are positioned perpendicular to each other. Place the hand in front of the eyes so that the thumb is pointing away from you. As t increases from 0 the corresponding point on the helix rotates clockwise from the $\boldsymbol{u}$ direction to the $\boldsymbol{v}$ direction as presented in Figure 1.2. In this thesis we are interested in a right-handed helix since the protein α-helix is a right-handed helix (see e.g. Campbell and Farrell, 2009).

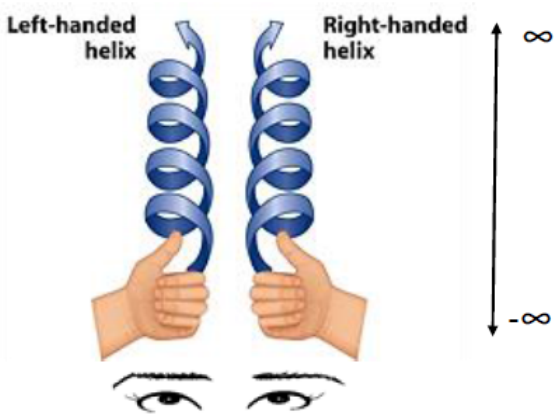

FIGURE 1.1: The right/left handed helices, where the time axis goes from $-\infty$ to ∞. Adopted from (Quizlet: https://quizlet.com/68685155/c41202-12-structure-of-dna-and-rna-flash-cards/).

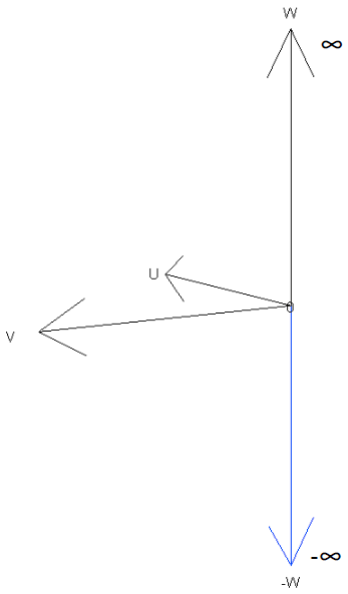

FIGURE 1.2: The right handed orientation.

1.2.2 The discrete mathematical helix model

There is also a discrete version of the helix obtained by restricting equation (1.1) to a finite set of equally spaced times, $t_i = (i-1)\delta$, where i is the sequence number $i = n_1, n_1 + 1 \ldots, n_2 - 1, n_2$, n_1 and n_2 are the first and last points on the helix and $n_1 < n_2$. Then the discrete mathematical helix model is given by

$$\boldsymbol{f}(t_i) = r\cos(t_i)\boldsymbol{u} + r\sin(t_i)\boldsymbol{v} + ct_i\boldsymbol{w} + \boldsymbol{b}, \tag{1.2}$$

where δ is the spacing parameter i.e. the *turn angle*. If $n_1 = 1$, then the initial time is $t_1 = 0$, and then the initial point is $\boldsymbol{f}(0) = r\boldsymbol{u} + \boldsymbol{b}$.

1.2.3 Special cases

1. *Canonical coordinates*: In this case $\Gamma = [\boldsymbol{u}\ \boldsymbol{v}\ \boldsymbol{w}] = \Gamma_0$, where $\Gamma_0 = [\boldsymbol{u}_0\ \boldsymbol{v}_0\ \boldsymbol{w}_0] = I_3$ is the identity matrix. In particular, $\boldsymbol{w}_0 = [0\ 0\ 1]^T$ is aligned with the z-axis in (1.2).

2. *Semi-canonical coordinates*: We have in (1.2) $\boldsymbol{w} = \boldsymbol{w}_0$, $\boldsymbol{u}$ and $\boldsymbol{v}$ are unrestricted other than being orthogonal to one another and $\boldsymbol{w}_0$. Then

$$\Gamma = \begin{bmatrix} \cos\tau & \sin\tau & 0 \\ -\sin\tau & \cos\tau & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

   where for $n_1 = 1$, τ is the angle between the helix initial point (x_1, y_1, z_1) and the point

$$\Gamma^T f(t_1) = (r, 0, 0)^T.$$

3. *General coordinates*: In this case there are no restrictions on the orthonormal matrix $\Gamma = [\boldsymbol{u}\ \boldsymbol{v}\ \boldsymbol{w}]$ in (1.2).

1.2.4 The statistical helix model

A discrete statistical helix is defined to be a discrete mathematical helix (1.2) with errors. We have points $\{\boldsymbol{y}_i := \boldsymbol{y}(t_i) := (y_{i1}, y_{i2}, y_{i3})^T, i = n_1, \ldots, n_2\}$, $n_1 < n_2$, $n_1, n_2 \in \mathbb{N}$, of $n = n_2 - n_1 + 1$ points or landmarks around the helix in three dimensions which satisfy the model

$$\boldsymbol{y}(t_i) = r\cos(t_i)\boldsymbol{u} + r\sin(t_i)\boldsymbol{v} + ct_i\boldsymbol{w} + \boldsymbol{b} + \boldsymbol{\varepsilon}_i, \tag{1.3}$$

where the $\boldsymbol{\varepsilon}_i \in \mathbb{R}^3$ are error terms, taken here to follow independent isotropic normal distributions $N_3(\boldsymbol{0}, \sigma^2 I)$. In this thesis, we assume points are equally spaced along a helix, and the time $t_i = (i-1)\delta$. We call this straight or unkinked statistical helix model the *regular statistical helix model*.

1.3 Proteins

Proteins are important to the structure and function (as enzymes, antibodies, hormones and transport molecules) of organic cells. Some protein examples are: membrane proteins, soluble globular proteins, fibrous proteins, and disordered proteins, where the most common type is the membrane proteins. The study of protein 3-dimensional structure is helpful to understand a protein's functions, so that we can know if anything is wrong with them. This is important since proteins participate in many processes in the cells. Overall, the study of helix structures, especially the kink of the helix, helps in the search of new drugs (Rigoutsos et al., 2003).

1.4 Protein structure

The functional properties depend on protein structure; therefore studying this structure has much benefit experimentally and computationally. Protein structure is arranged in a hierarchical organization as follows

1.4.1 Protein primary structure

Protein primary structure is a sequence of amino acids. There are 20 different types of amino acids in proteins, each represented by an alphabetical code. The main chain of the amino acid is the backbone. Each amino acid consists of a hydrogen atom H, a carboxyl group ($COOH$), and an amino group (NH_2). In addition, amino acids also contain a side chain R-group which differ from one amino acid to another (and that gives amino acids their chemical properties), it is attached to the carbon C_α atom as shown in Figure 1.3; for more details see Mardia (2013), Branden and Tooze (1999) and Creighton (1993).

In amino acids the carboxyl group ($COOH$), known as the C-terminal, and the amino group (NH_2), known as the N-terminal, connect amino acids to each other by the peptide bonds. Therefore, a protein sometimes called a polypeptide, since it is a linear sequence of amino acids connected by peptide bonds. Proteins can contain two or more polypeptide chains, called subunits (see Chou and Cai, 2003). The N-terminal is always written on the left of a protein chain.

1.4.2 Protein secondary structure

The secondary structure is the formation of the polypeptide into regular and repetitive patterns of amino acids. The polypeptide (the linear sequence of amino acids) forms of itself into a 3-dimensional structure held together by

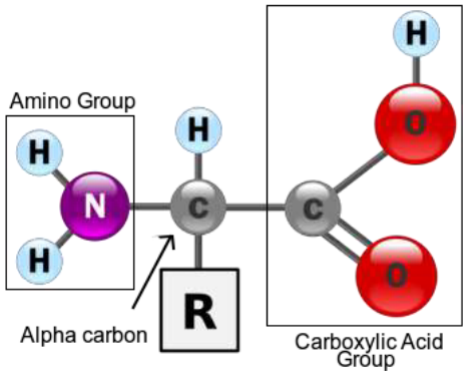

FIGURE 1.3: Amino acid diagram. (About education: http://biology.about.com/od/molecularbiology/ss/amino-acid.htm).

hydrogen bonds. The hydrogen bonds are between C=O and H-N groups. These 3-dimensional structures are α-helices and β-pleated sheets.

The 3-dimensional protein structure is a key property to identify the protein function, which can be determined by X-ray crystallography, X-ray diffraction or nuclear magnetic resonance (NMR). X-ray crystallography has an issue as it is not easy to let some proteins form crystals (see Miao et al., 1999). NMR also has an issue as it applies to only small protein molecules (see Heise et al., 2013, p. 98). The most common way to explain protein structure is X-ray diffraction. The Protein Data Bank (PDB) has released protein structures to the public domain (for more information see Berman et al., 2007).

1.4.3 Tertiary structure and Quaternary structure

Tertiary structure is the overall folding of the whole polypeptide, the 3-dimensional structure, where the atoms of the side chains of the amino acids are bonded together (for more information see Campbell and Farrell, 2009, pp. 98-99; Chou and Cai, 2003).

Quaternary protein structure is a combination of several protein chains into a single larger shape, where the subunits can work together to give the special

properties not as a single subunit. Some examples of quaternary structure protein are haemoglobin and ion channels, (see Campbell and Farrell, 2009, p. 106; Chou and Cai, 2003).

1.5 α-helix

The most common type of protein secondary structure is the α-helix. A protein chain can organize itself into a helix, called an α-helix. A protein α-helix, for our purpose, can be treated as a time-ordered sequence of points (C_α) in $\mathbb{R}^3$. The α-helix known parameters are: the pitch of a helix is the vertical distance covered by one complete helix turn, $2\pi c = 5.4\mathring{A}$; and the radius $r = 2.3\mathring{A}$. There are 3.6 C_α atoms per turn of 2π radians. Therefore, C_α atoms are plotted every $360/3.6 = 100^o$ around a circle, as shown in Figure 1.4 (see Mardia, 2014). We can project the 3-dimensional helix onto xy-plane that is perpendicular to the axis, see e.g. Figure 1.5, which clearly presents three turns of 3.6 amino-acid residues per turn of the helix. The "times" $t_i = (i-1)\delta$ presents moving along the helix, and the "spacing" parameter is $\delta = \frac{2\pi}{3.6}$ radians, i=1, . . . , n, where n is the number of C_α atoms (landmarks or points).

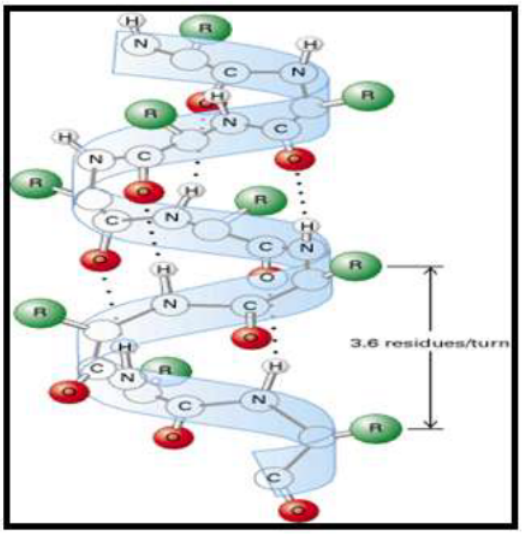

FIGURE 1.4: Protein α-helix. (Indian Biological Sciences and Research Institute, Biology: http://ibripharmacy.blogspot.co.uk/2014/05/structural-biology.html).

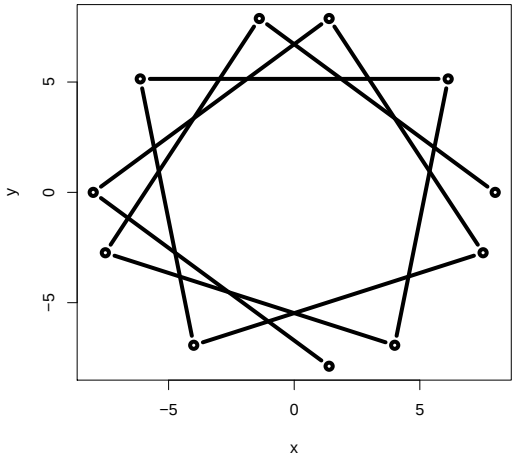

FIGURE 1.5: Projected helix mimics protein α-helix.

1.5.1 Helix kink

As we said before, a kink is a region of points where the helix axis changes direction, where the axis is no longer straight; for more details about kink see

Wilman et al. (2014a); Deville et al. (2008); Kumar and Bansal (2012). It is actually break in the hydrogen bonds in the amino acid backbone and thus the helix will lose its flexibility of bending (see Hall et al., 2009). An α-helix could be straight or it could consist of one kink as shown in Figure 1.6, (Mardia et al., 2018). Since a kink is a common distortion feature of α-helix and a functionally important structural feature in soluble and membrane proteins. Various studies are available on protein helix kinks, such as Kinkfinder by Wilman et al. (2013) and Kink-Detector by Mardia et al. (2018). For more details see Section 1.1.

It is more common in membrane proteins than in soluble proteins because the membrane protein α-helices are usually longer (i.e. number of landmarks $n \geq 20$). Most α-helices in soluble protein are shorter (see Wilman et al., 2014b).

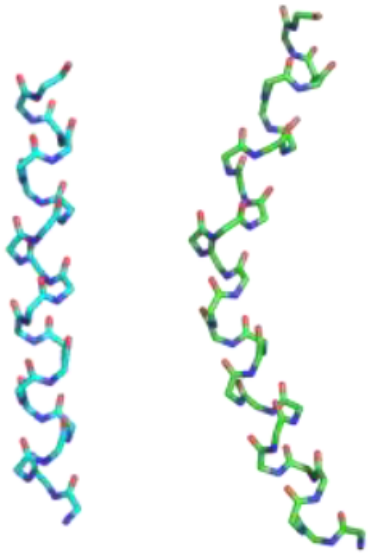

FIGURE 1.6: Straight and kinked α-helix. (Oxford university, Oxford Protein Informatics Group: http://www.blopig.com/blog/2013/09/what-is-a-kink/).

A biochemical helix is allowed to bend (see Mardia et al., 2018) and such bending is not considered a kink. Second, and related to this bending, in reality a kink in the α-helix is not a sharp change in direction of the helix axis (see Mardia et al., 2018), but it is more of a gradual change. In this thesis we study a simple

version of a kink, as it is a global change on the helix axis rather than a local change. Our straight helix is a regular helix with no bend, i.e. we do not allow for a curve as sometimes occurs in a real protein helix. Our bent helices include kinked and curved helices.

Chapter 2

Matrix algebra

The purpose of this chapter is to review some matrix properties that we use later in our calculations. There is a section on perturbation theory (for example see Mardia et al., 1979, Appendix A). We discuss various matrix decompositions including the spectral decomposition (SD), the singular value decomposition (SVD), the Cholesky decomposition (CD) and the optimally signed singular value decomposition (OS-SVD).

More generally, we identify the full and reduced versions of relationship between the spectral decomposition (SD) and the singular value decomposition (SVD). The standard version of the SD is not quite unique when some singular values are equal. However, a unique version can be constructed using projection matrices.

Moreover, we discuss the relationship between spectral decomposition (SD) and the singular value decomposition (SVD) by showing how to determine the SD from SVD.

Also we present the Cholesky decomposition and the optimally signed singular value decomposition (OS-SVD) techniques. Then we describe the use of the

optimally signed singular value decomposition (OSR-SVD) in matrix optimization. Finally we present an application to Procrustes Analysis.

2.1 Perturbation theory

Given a symmetric matrix $A_{3\times 3}$ with distinct eigenvalues, we can perturb A for small ε, by Taylor expanding in ε, as follows

$$A(\varepsilon) = A^{(0)} + \varepsilon A^{(1)} + \varepsilon^2 A^{(2)} + O(\varepsilon^3),$$

where $A(0) = A^{(0)} = A$ is the unperturbed matrix and $\varepsilon A^{(1)}$ is the first perturbation term, and $\varepsilon^2 A^{(2)}$ the second perturbation term (see Kato, 2013, pp. 63-64; Kent et al., 1983, and appendix A). The matrices $A^{(0)}, A^{(1)}$ and $A^{(2)}$ of size 3×3 are assumed symmetric.

An eigenvalue $\lambda = \lambda(\varepsilon)$ of $A(\varepsilon)$ satisfies $\det(A(\varepsilon) - \lambda I) = 0$. The corresponding eigenvector $\boldsymbol{\nu} = \boldsymbol{\nu}(\varepsilon)$ satisfies $(A(\varepsilon) - \lambda I)\boldsymbol{\nu} = 0$ and can be expanded in powers of ε as:

$$\lambda = \lambda^{(0)} + \varepsilon\lambda^{(1)} + \varepsilon^2\lambda^{(2)} + O(\varepsilon^3),$$
$$\boldsymbol{\nu} = \boldsymbol{\nu}^{(0)} + \varepsilon\boldsymbol{\nu}^{(1)} + \varepsilon^2\boldsymbol{\nu}^{(2)} + O(\varepsilon^3).$$

The eigenvalue equation $A(\varepsilon)\boldsymbol{\nu} = \lambda\boldsymbol{\nu}$, up to the first perturbation term, is

$$(A^{(0)} + \varepsilon A^{(1)})(\boldsymbol{\nu}^{(0)} + \varepsilon\boldsymbol{\nu}^{(1)}) = (\lambda^{(0)} + \varepsilon\lambda^{(1)})(\boldsymbol{\nu}^{(0)} + \varepsilon\boldsymbol{\nu}^{(1)}),$$

so that

$$A^{(0)}\boldsymbol{\nu}^{(0)} = \lambda^{(0)}\boldsymbol{\nu}^{(0)}, \tag{2.1}$$
$$A^{(0)}\boldsymbol{\nu}^{(1)} + A^{(1)}\boldsymbol{\nu}^{(0)} = \lambda^{(0)}\boldsymbol{\nu}^{(1)} + \lambda^{(1)}\boldsymbol{\nu}^{(0)}. \tag{2.2}$$

If we know $A^{(1)}, \lambda^{(0)}$ and $\boldsymbol{\nu}^{(0)}$ then we can derive $\lambda(\varepsilon)$ up to first perturbation. Multiply the equation (2.2) on the left by $\boldsymbol{\nu}^{(0)T}$ to give

$$\boldsymbol{\nu}^{(0)T} A^{(0)} \boldsymbol{\nu}^{(1)} + \boldsymbol{\nu}^{(0)T} A^{(1)} \boldsymbol{\nu}^{(0)} = \lambda^{(0)} \boldsymbol{\nu}^{(0)T} \boldsymbol{\nu}^{(1)} + \lambda^{(1)} \boldsymbol{\nu}^{(0)T} \boldsymbol{\nu}^{(0)}. \tag{2.3}$$

Secondly after using equation (2.1), the first term of the left-side of equation (2.3) becomes

$$\boldsymbol{\nu}^{(0)T} A^{(0)} \boldsymbol{\nu}^{(1)} = (\boldsymbol{\nu}^{(1)T} A^{(0)} \boldsymbol{\nu}^{(0)})^T = (\boldsymbol{\nu}^{(1)T} \lambda^{(0)} \boldsymbol{\nu}^{(0)})^T = \lambda^{(0)} \boldsymbol{\nu}^{(0)T} \boldsymbol{\nu}^{(1)}. \tag{2.4}$$

Then substitute (2.4) into equation (2.3) to get

$$\lambda^{(1)} = \boldsymbol{\nu}^{(0)T} A^{(1)} \boldsymbol{\nu}^{(0)},$$

where $\boldsymbol{\nu}^{(0)T} \boldsymbol{\nu}^{(0)} = 1$, then the perturbed eigenvalue is given by

$$\lambda(\varepsilon) = \lambda^{(0)} + \varepsilon \boldsymbol{\nu}^{(0)T} A^{(1)} \boldsymbol{\nu}^{(0)}.$$

Next, the formula of $\boldsymbol{\nu}^{(1)}$ by Kent et al. (1983) is

$$\boldsymbol{\nu}^{(1)} = \left(\sum_{\lambda_i \neq \lambda^{(0)}} (\lambda^{(0)} - \lambda_i)^{-1} \boldsymbol{\nu}_i \boldsymbol{\nu}_i^T \right) A^{(1)} \boldsymbol{\nu}^{(0)}, \tag{2.5}$$

where $\boldsymbol{\nu}_i$ is the i^{th} eigenvector corresponding to eigenvalue λ_i of A. Let for example

$$A = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix},$$

where $\lambda_1 = \lambda^{(0)}$ and $\boldsymbol{\nu}^{(0)} = \boldsymbol{\nu}_1$, where $\boldsymbol{\nu}_1 = [1, 0, 0]^T$. Now we can use equation (2.5) to derive the first perturbed eigenvector,

$$\begin{aligned}\boldsymbol{\nu}^{(1)} &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{\lambda_2 - \lambda_1} & 0 \\ 0 & 0 & \frac{1}{\lambda_3 - \lambda_1} \end{bmatrix} \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{1,2} & a_{2,2} & a_{2,3} \\ a_{1,3} & a_{2,3} & a_{3,3} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ \frac{a_{1,2}^{(1)}}{\lambda_2 - \lambda_1} \\ \frac{a_{1,3}^{(1)}}{\lambda_3 - \lambda_1} \end{bmatrix}\end{aligned}$$

2.2 Spectral decomposition (SD) and singular value decomposition (SVD)

In this section we present the spectral decomposition (SD) and singular value decomposition (SVD) of matrices in both full and reduced forms. The SD is not quite unique as usually presented, whereas it can be made unique using symmetric projection matrices. In addition, we view the relationship between SVD and SD, and show how to start with the SD of matrices $C = A^T A$ and $B = AA^T$, and from this evaluate the SVD of A. The standard properties are adopted from Mardia et al. (1979), pp. 469-474, Golub and Reinsch (1971), and Lay (2006), pp. 266-270, 398 and 414-420.

2.2.1 Spectral decomposition (SD)

A spectral decomposition of a square symmetric positive semi definite matrix $A_{n\times n}$ has eigenvalues $\lambda_i \geq 0, i = 1, \ldots, n$ which can be reduced to p non-zero eigenvalues. The SD of $A_{n\times n}$ with p non-zero eigenvalues takes the reduced form as follows:

$$A = \Gamma\Lambda\Gamma^T = \sum_{i=1}^{p} \lambda_i \boldsymbol{\gamma}_{(i)} \boldsymbol{\gamma}_{(i)}^T \tag{2.6}$$

where Γ is an orthogonal matrix whose columns are the eigenvectors of A, and Λ is a diagonal matrix whose entries are the non-zero eigenvalues of A. The λ_i are eigenvalues of the columns of Γ, which are eigenvectors, i.e. $A\boldsymbol{\gamma}_{(i)} = \lambda_i \boldsymbol{\gamma}_{(i)}$.

Note that we pad out Γ and Λ in the reduced form of the SD in (2.6), to get $\Gamma^\star$ and $\Lambda^\star$. The full SD of $A_{n\times n}$ can takes the form of

$$\Gamma^\star_{n\times n}\ \Lambda^\star_{n\times n}\ \Gamma^{T\star}_{n\times n} = \left[\ \Gamma_{n\times p} \quad \Gamma^\perp_{n\times(n-p)}\ \right]_{n\times n} \left[\begin{array}{cc} \Lambda_{p\times p} & 0_{p\times(n-p)} \\ 0_{(n-p)\times p} & 0_{(n-p)\times(n-p)} \end{array}\right]_{n\times n} \left[\begin{array}{c} \Gamma^T_{p\times n} \\ \Gamma^{T\perp}_{(n-p)\times n} \end{array}\right]_{n\times n},$$

where 0 is a zero matrix. The matrix $\Gamma^\star_{n\times n}$ is an orthogonal and $\Gamma^\perp$ is an orthonormal column matrix with $(n-p)$ columns, and each column vector in Γ is perpendicular to each column vector in $\Gamma^\perp$, $\Gamma \perp \Gamma^\perp$, that is the dot product of one column from Γ and one column from $\Gamma^\perp$ is zero.

2.2.1.1 A spectral decomposition is not unique

A spectral decomposition is not quite unique, in the following sense. Firstly, changing the sign of any column of Γ does not affecting the validity of (2.6). For example, since $A = \Gamma\Lambda\Gamma^T$, if we change the sign of one column of Γ, the result will be the same because of the multiplicity of the sign (the change in sign will cancel itself out in the product $\boldsymbol{\gamma}_{(i)}\boldsymbol{\gamma}^T_{(i)} = (-\boldsymbol{\gamma}_{(i)})(-\boldsymbol{\gamma}^T_{(i)})$).

Secondly, if an eigenvalue has multiplicity of two or more, then there is a set of distinct linearly independent eigenvectors. For example, if all the eigenvalues are equal to three ($\lambda_1 = \lambda_2 = 3$) in the case n=2, then

$$A = \left[\begin{array}{cc} 3 & 0 \\ 0 & 3 \end{array}\right] = 3I,$$

is a scalar multiple of the identity matrix, i.e. the SD is $3I = \Gamma 3I \Gamma^T$, where Γ can be any orthogonal square matrix. Thus the eigenvector matrix is not unique.

The eigenvectors, for example, can be taken as

$$\boldsymbol{\gamma}_{(1)} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \boldsymbol{\gamma}_{(2)} = \begin{bmatrix} -1 \\ 0 \end{bmatrix},$$

or

$$\boldsymbol{\gamma}_{(1)} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} / \sqrt{2}, \boldsymbol{\gamma}_{(2)} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} / \sqrt{2}.$$

2.2.1.2 A unique version of spectral decomposition

A unique version of the SD can be constructed using symmetric projection matrices (see e.g. Ohya and Watanabe (2008), p. 218). Recall a projection matrix P is idempotent and symmetric, i.e. $P = P^2$ and $P = P^T$. A projection matrix has eigenvalues either 1 or 0 and this result can be proved as follows: Let λ be an eigenvalue of P corresponding to eigenvector $\boldsymbol{v} \neq 0$. Then

$$\begin{aligned} P\boldsymbol{v} &= \lambda \boldsymbol{v} \\ \Rightarrow \lambda^2 \boldsymbol{v} &= P^2 \boldsymbol{v} = P\boldsymbol{v} = \lambda \boldsymbol{v} \\ \Rightarrow \lambda^2 &= \lambda \\ \Rightarrow \lambda &= 0 \text{ or } 1. \end{aligned}$$

There are two ways to label the eigenvalues of $A_{n\times n}$. First let $(\lambda^{(1)}, \ldots, \lambda^{(q)})$ be the q distinct eigenvalues and then let $(\lambda_1, \ldots, \lambda_n)$ be all the eigenvalues (some with multiplicity at least two), where $q \leq n$. Then

$$A_{n\times n} = \Gamma \Lambda \Gamma^T = \sum_{i=1}^{n} \lambda_i \boldsymbol{\gamma}_{(i)} \boldsymbol{\gamma}_{(i)}^T = \sum_{j=1}^{q} \sum_{i:\lambda_i = \lambda^{(j)}} \lambda^{(j)} P_j,$$

and $P_j = \sum_{j=1}^{q} \boldsymbol{\gamma}_{(j)} \boldsymbol{\gamma}_{(j)}^T$. We consider an explicit example. Let

$$
\begin{aligned}
A &= \begin{bmatrix} 3 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix} \\
&= \Gamma \Lambda \Gamma^T = \sum_{i=1}^{n} \lambda_i \boldsymbol{\gamma}_{(i)} \boldsymbol{\gamma}_{(i)}^T = \sum_{j=1}^{q} \sum_{i:\lambda_i = \lambda^{(j)}} \lambda^{(j)} P_j \\
&= 3(\boldsymbol{\gamma}_{(1)} \boldsymbol{\gamma}_{(1)}^T + \boldsymbol{\gamma}_{(2)} \boldsymbol{\gamma}_{(2)}^T + \boldsymbol{\gamma}_{(3)} \boldsymbol{\gamma}_{(3)}^T) + 2(\boldsymbol{\gamma}_{(4)} \boldsymbol{\gamma}_{(4)}^T + \boldsymbol{\gamma}_{(5)} \boldsymbol{\gamma}_{(5)}^T),
\end{aligned}
$$

where $\lambda_1 = \lambda_2 = \lambda_3 = \lambda^{(1)} = 3$, and $\lambda_4 = \lambda_5 = \lambda^{(2)} = 2$. The projection matrices are

$$
\begin{aligned}
P_1 &= \gamma_{(1)} \gamma_{(1)}^T + \gamma_{(2)} \gamma_{(2)}^T + \gamma_{(3)} \gamma_{(3)}^T, \\
&= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},
\end{aligned}
$$

and

$$
\begin{aligned}
P_2 &= \gamma_{(4)} \gamma_{(4)}^T + \gamma_{(5)} \gamma_{(5)}^T, \\
&= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.
\end{aligned}
$$

2.2.2 Singular value decomposition (SVD)

The SVD is a decomposition of a rectangular matrix $A_{n\times m}$, where $n > m$, into two orthonormal column matrices and a diagonal matrix. Let $\text{rank}(A)$ be the rank of matrix A and $p = \text{rank}(A) \leq \min(n, m)$. The SVD of A takes the form

$$A = ULV^T = \sum_{i=1}^{p} \ell_i \boldsymbol{u}_{(i)} \boldsymbol{v}_{(i)}^T, \tag{2.7}$$

where $U_{n\times p}$ is a column orthonormal matrix of left singular vectors, $V_{m\times p}$ is column orthonormal matrix of right singular vectors, and $L = \text{diag}(\ell_1, \ell_2, \ldots, \ell_p)$ is a diagonal matrix of singular values, $\ell_1 > \ell_2 > \ldots > \ell_p > 0$. Note that

$$\begin{aligned} A\boldsymbol{v}_{(i)} &= ULV^T \boldsymbol{v}_{(i)}, \\ &= UL \begin{bmatrix} \boldsymbol{v}_{(1)}^T \\ \vdots \\ \boldsymbol{v}_{(p)}^T \end{bmatrix} \boldsymbol{v}_{(i)} = UL\boldsymbol{e}_{(i)} = U\ell_i \boldsymbol{e}_{(i)}, \\ &= \ell_i \begin{bmatrix} \boldsymbol{u}_{(1)} & \ldots & \boldsymbol{u}_{(p)} \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \ell_i \boldsymbol{u}(i). \end{aligned}$$

Conversely, $A^T \boldsymbol{u}_{(i)} = \ell_i \boldsymbol{v}_{(i)}$.

Comment:

1. Equation (2.7) is sometimes called the "reduced" form of SVD. If $n \geq m$ then the full SVD takes the form

$$U^{\star}_{n\times m} L^{\star}_{m\times m} V^{T\star}_{m\times m} = \begin{bmatrix} U_{n\times p} & U^{\perp}_{n\times(m-p)} \end{bmatrix}_{n\times m} \begin{bmatrix} L_{p\times p} & 0_{p\times(m-p)} \\ 0_{(m-p)\times p} & 0_{(m-p)\times(m-p)} \end{bmatrix}_{m\times m} \begin{bmatrix} V^{T}_{p\times m} \\ V^{T\perp}_{(m-p)\times m} \end{bmatrix}_{m\times m}$$

where 0 is a zero matrix. The matrix $V^{\star}_{m\times m}$ is orthogonal; $U^{\star}_{n\times m}$ is a column orthonormal matrix; $V^{\perp}$ is a column orthonormal matrix with $m-p$ columns, and $V \perp V^{\perp}$; $U^{\perp}$ is a column orthonormal matrix with $m-p$ columns, and $U \perp U^{\perp}$.

2.2.3 Relationship between singular value decomposition and spectral decomposition

Start with $A = ULV^T$, let $B = AA^T$ and $C = A^TA$. Then the SDs of B and C can be deduced from the SVD of A as follows:

$$\begin{aligned} C_{m\times m} &= A^T_{m\times n} A_{n\times m}, \\ &= (U_{n\times p} L_{p\times p} V^T_{p\times m})^T (U_{n\times p} L_{p\times p} V^T_{p\times m}), \\ &= V_{m\times p} L^T_{p\times p} U^T_{p\times n} U_{n\times p} L_{p\times p} V^T_{p\times m}, \\ &= V_{m\times p} L^T_{p\times p} L_{p\times p} V^T_{p\times m}, \\ &= V_{m\times p} \Lambda_{p\times p} V^T_{p\times m}, \\ &= \sum_{i=1}^{p} \ell_i^2 \boldsymbol{v}_{(i)} \boldsymbol{v}^T_{(i)}. \end{aligned}$$

Similarly, we deduce the SD of B as follows:

$$\begin{aligned} B_{n\times n} &= A_{n\times m}A^T_{m\times n}, \\ &= (U_{n\times p}L_{p\times p}V^T_{p\times m})(U_{n\times p}L_{p\times p}V^T_{p\times m})^T, \\ &= U_{n\times p}L_{p\times p}V^T_{p\times m}V_{m\times p}L^T_{p\times p}U^T_{p\times n}, \\ &= U_{n\times p}L_{p\times p}L^T_{p\times p}U^T_{p\times n}, \\ &= U_{n\times p}\Lambda_{p\times p}U^T_{p\times n}, \\ &= \sum_{i=1}^{p}\ell_i^2\boldsymbol{u}_{(i)}\boldsymbol{u}^T_{(i)}. \end{aligned}$$

We explain the deduction: since U and V are column orthonormal matrices, $U^T_{p\times n}U_{n\times p} = I_p$ and $V^T_{p\times m}V_{m\times p} = I_p$. Let $L^T_{p\times p}L_{p\times p} =: \Lambda_{p\times p}$, i.e. $\ell_i^2 = \lambda_i > 0$ where λ_i is the i^{th} diagonal entry of Λ and ℓ_i is the i^{th} diagonal entry of L. Recall that $L_{p\times p}$ is a diagonal matrix; thus B and C have the same p non zero eigenvalues. The SD for C is $V_{m\times p}\Lambda_{p\times p}V^T_{p\times m}$ where $V_{m\times p}$ contains the p eigenvectors corresponding to the p non-zero eigenvalues $\Lambda = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_p)$ for C. The SD of B is $U_{n\times p}\Lambda_{p\times p}U^T_{p\times n}$, where $U_{n\times p}$ contains the p eigenvectors corresponding to the p non-zero eigenvalues $\Lambda = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_p)$ for B.

The columns of the matrix U are orthonormal eigenvectors of AA^T. The columns of the matrix V are orthonormal eigenvectors of A^TA. The eigenvectors of C are known as the right singular vectors for A and the eigenvectors of B are known as the left singular vectors of A.

Proof of how the singular value decomposition may be derived from the spectral decomposition

Start with $A_{n\times m}$ where $n > m$ and let $p = \min(n, m)$. Let $C = A^TA$ and consider the SD of C, $C = V\Lambda V^T$, where the columns of V are orthogonal eigenvectors of C for the non zero eigenvalues p. Set $L = \Lambda^{\frac{1}{2}}$ i.e. $\ell_i = \sqrt{\lambda_i}$ for all i. From V if

we define $U = AVL^{-1}$, then $A_{n\times m} = ULV^T$ and $U^TU = I$, since

$$\begin{aligned} C = A^TA &= VL^2V^T \\ &= A^TAVL^{-1} = VL \\ &= A^TU = VL, \end{aligned}$$

then $A^T = VLU^T$, and

$$U^TU = LV^TA^TAVL^{-1} = I,$$

hence we can say that the unique SD of $B = AA^T$ from C is constructed as follows: We choose an orthogonal matrix V and a diagonal matrix $\Lambda = L^2$ of distinct eigenvalues. Also we constructed $\boldsymbol{u}_i$ from $\boldsymbol{v}_i$, in other words $\boldsymbol{u}_{(i)}$ depends on our choice of $\boldsymbol{v}_{(i)}$, and individually the sign is not determined but the pair $(\boldsymbol{u}_{(i)}, \boldsymbol{v}_{(i)})$ is determined up to sign.

2.2.4 Optimally signed singular value decomposition (OS-SVD)

In this section, we consider three main modified versions of the standard singular value decomposition (SVD) for a square matrix. For clarity we call the standard singular value decomposition the *positive singular value decomposition* (P-SVD), also the three variants are the *signed singular value decomposition* (S-SVD), the *signed rotation singular value decomposition* (SR-SVD), and the *optimally signed singular value decomposition* (OSR-SVD).

The standard singular value decomposition, or P-SVD, of a square matrix A of size $n \times n$ is given by

$$A = U^pL^pV^{pT}, \tag{2.8}$$

where U^p and V^p are orthogonal matrices of size $n \times n$, and L^p is a diagonal matrix of non-negative values ℓ_i^p, $i = 1, 2, \ldots, n$. In the variants of the SVD, the elements of L are allowed to be negative and in some of the variants the matrices U and V are required to be rotation matrices.

- *Signed singular value decomposition* (S-SVD), $A = U^s L^s V^{sT}$. In this version the elements of L^s are allowed to be positive or negative; U^s and V^s are still just required to be orthogonal matrices.

- *Signed rotation singular value decomposition* (SR-SVD), $A = U^R L^R V^{RT}$. The elements of L^R are allowed to be positive or negative; U^R and V^R are required to be rotation matrices.

- *Optimally signed singular value decomposition* (OSR-SVD), $A = ULV^T$. This is a special version of SR-SVD where at most the smallest singular value is negative (if $|A| < 0$).

  All previous definitions are adopted from Kent and Mardia (2001).

It is possible to obtain the variants from the P-SVD decomposition through the use of diagonal matrices D_1 and D_2 of size $n \times n$ with entries ± 1 by

$$A = (U^p D_1)(D_1 L^p D_2)(D_2 V^{pT}). \tag{2.9}$$

In particular

1. If $\det(A) > 0$ then either $\det(U^p) = \det(V^p) = 1$ or $\det(U^p) = \det(V^p) = -1$. If the case of $\det(U^p) = \det(V^p) = -1$, then for example, by choosing $D_1 = D_2 = \text{diag}(-1, 1, \ldots, 1)$, we have $\det(U^p D_1) = \det(V^p D_2) = 1$. Then $U^p D_1$ and $V^p D_2$ are rotation matrices and we have a rotation version of P-SVD decomposition.

2. If $\det(A) < 0$, then we have $\det(U^p)$ (or $\det(V^p)$) is -1, but not both. We can change the sign of the last diagonal entry of D_1 (or D_2), so that $U^p D_1$ and $V^p D_2$ are rotation matrices. The smallest singular value in L^p has become negative in $(D_1 L^p D_2)$, and thus we have the OSR-SVD decomposition.

3. If $\det(A) = 0$, we have $\det(L^p) = 0$ i.e $\ell_i = 0$ for all i, and we can change the sign of the last column in U^p and V^p in $U^p D_1$ and $V^p D_2$ to ensure $\det(U^p D_1)$ and $\det(V^p D_2)$ are positive, so we have SR-SVD.

2.2.5 The use of the OSR-SVD in matrix optimization

Theorem 2.1

Let Z be a $n \times n$ square matrix and let the OSR-SVD of Z be given by $Z = ULV^T$. Let $f(R) = \text{tr}(Z^T R)$, and consider the optimization problem max $f(R)$ over $R \in SO(n)$ where $SO(n)$ is the rotation group. Then $\max_R \text{tr}(Z^T R) = \text{tr}L$ and $R_{opt} = UV^T$.

Proof

Note that if $\det(Z) \geq 0$, then the proof is easy as all entries of L are non-negative. If not, then the proof need to some work as the smallest entry of L is negative, see for example Mardia et al. (1979) pp. 416-417, and Dryden and Mardia (2016), pp. 70-71. We use this result later in Section 3.4.

The proof proceeds in three steps. The first step is to show that if R_{opt} is the optimal choice of R, then $H = Z^T R_{opt}$ is symmetric. Secondly for any symmetric matrix, we note that the eigenvalues are equal to the singular values up to sign. The third step is to find the optimal choice of signs.

Step 1 Suppose R_{opt} is the optimal choice of R. We claim $H = Z^T R_{opt}$ is symmetric. We prove this claim by contradiction. Hence, suppose H is not symmetric. Recall the following construction of a rotation matrix. If S is a skew symmetric matrix $S = (s_{ij})_{1 \leq i,j \leq n}$ (i.e. $s_{ij} = -s_{ji}$) then

$$Q(S) = \exp(S) = I + S + \frac{(S)^2}{2!} + \cdots + \frac{(S)^k}{k!} + \cdots , \tag{2.10}$$

is a rotation matrix, (see Marsden and Ratiu, 1995, p. 285).

Next replace S by εS and let the magnitude $\varepsilon > 0$ be small. Multiplying the equation (2.10) by H and taking the trace gives

$$\begin{aligned} \mathrm{tr} HQ(\varepsilon S) &= \mathrm{tr} H(I + \varepsilon S + O(\varepsilon^2)) \\ &= \mathrm{tr} H + \varepsilon \mathrm{tr}\ HS + O(\varepsilon^2). \end{aligned} \tag{2.11}$$

The second term of equation (2.11), can be simplified as follows

$$\begin{aligned} \varepsilon\ \mathrm{tr}\ HS &= \varepsilon \sum_{ij} h_{ij} s_{ij} \\ &= \varepsilon \sum_{i<j} (h_{ij} - h_{ji}) s_{ji}, \end{aligned}$$

as $s_{ij} = -s_{ji}$ for all $i, j = 1, \ldots, n$. Since H is not symmetric, then there exist i_o, j_o such that $h_{i_o j_o} - h_{j_o i_o} \neq 0$. Construct a matrix S_o whose elements are all zeros except for positions (i_o, j_o) and (j_o, i_o), with $s_{i_o,j_o} = -s_{j_o,i_o}$=1. Set

$$\begin{aligned} g(\varepsilon) &=: \mathrm{tr} HQ(\varepsilon S_o) \\ &= \mathrm{tr} H + \varepsilon(h_{i_o j_o} - h_{j_o i_o}) + O(\varepsilon^2), \end{aligned}$$

then

$$g(0) = \mathrm{tr}H,$$

and by optimality of R_{opt}

$$\frac{\partial}{\partial \varepsilon} g(0) = 0. \tag{2.12}$$

Then

$$g(\varepsilon) - g(0) = \varepsilon(h_{i_o j_o} - h_{j_o i_o}) + O(\varepsilon^2).$$

this implies the first derivative of $g(0)$ with respect to ε is

$$\frac{\partial}{\partial \varepsilon} g(0) = (h_{i_o j_o} - h_{j_o i_o}) \neq 0,$$

that is a contradiction with the result in equation (2.12), and hence H is symmetric.

Step 2 Any real symmetric matrix can be decomposed by the spectral decomposition theorem. The spectral decomposition of the symmetric matrix H is almost the same as the standard singular value decomposition up to the sign of the singular values. In particular, the SD here is an example of S-SVD, so that

$$\begin{aligned} H &= Z^T R \\ &= G \Lambda G^T \\ &= U^s L^s V^{sT}, \end{aligned}$$

where $U^s = G, V^s = G$ and $L^s = \Lambda$. Thus $\ell_i^s = \lambda_i$, and $\lambda_i = (-1)^{\alpha_i} \ell_i$, where $\ell_i \geq 0$, $i = 1, \ldots, n$ and $\alpha_i = 0$ or 1.

Step 3 We want to find the optimal choice for the signs i.e. the optimal choice of the α_i. If H is symmetric matrix then the eigenvalues are equal to the singular

values up to sign i.e. $\ell_i = |\lambda_i| > 0$, and to maximize $\mathrm{tr}Z^T R$ we need to minimize the sum of the negative eigenvalues, $\sum \lambda_i$, so that we choose Z to be OSR-SVD, then

1. If $\det(Z) > 0$, then the number of $\alpha_i = 1$ is even. Thus $\mathrm{tr}H$ is largest if $\alpha_i = 0$, for all i.

2. If $\det(Z) < 0$, then the number of $\alpha_i = 1$ is odd. Thus $\mathrm{tr}H$ is largest if only one $\alpha_i = 1$, namely $\alpha_n = 1$ and other $\alpha_i = 0$.

Application to Procrustes Analysis

Let $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$ be vectors of n points in p-dimensions with mean $\bar{\boldsymbol{x}}$ and suppose $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n$, with mean $\bar{\boldsymbol{y}}$, $i = 1, 2, \ldots, n$, are related by the model

$$\boldsymbol{x}_i = A^T \boldsymbol{y}_i + \boldsymbol{b} + \boldsymbol{\varepsilon_i},$$

where $A_{3\times 3}$ is a rotation matrix and $\boldsymbol{b}_{3\times 1}$ is a shift vector. Assume $\boldsymbol{\varepsilon_i} = [\varepsilon_{1i}, \varepsilon_{2i}, \varepsilon_{3i}]^T$ is independent and identically distributed $N_3(0, \sigma^2 I_3)$.

Our aim is to estimate the rotation matrix A and the shift vector $\boldsymbol{b}$ by least squares. To do this we first centre the response variable $\boldsymbol{y}' = \boldsymbol{y}_i - \bar{\boldsymbol{y}}$, and the explanatory variable $\boldsymbol{x}' = \boldsymbol{x}_i - \bar{\boldsymbol{x}}$, for $i = 1, 2, \ldots, n$, so the new variables have a sample mean equals zero. The shift vector changes to $\boldsymbol{b}' = \bar{\boldsymbol{x}}' - A^T \bar{\boldsymbol{y}}' = 0$. The initial centring remove the translation parameter. We want to minimize the residual sum of squares of the data, with respect to A and $\boldsymbol{b}'$. The residual sum of squares is, as follows:

$$M^2 = \sum_{i=1}^{n} (\boldsymbol{x}'_i - A^T \boldsymbol{y}'_i - \boldsymbol{b}')^T (\boldsymbol{x}'_i - A^T \boldsymbol{y}'_i - \boldsymbol{b}'). \tag{2.13}$$

Substituting $\boldsymbol{b}' = \mathbf{0}$ in equation (2.13), and it can be written in matrix form as

$$\begin{aligned} M^2 &= \mathrm{tr}(X' - Y'A)^T(X' - Y'A), \\ &= \mathrm{tr}(X'^T X' - X'^T Y'A - A^T Y'^T X' + A^T Y'^T Y'A), \\ &= \mathrm{tr}X'^T X' - \mathrm{tr}X'^T Y'A - \mathrm{tr}A^T Y'^T X' + \mathrm{tr}A^T Y'^T Y'A, \\ &= \mathrm{tr}X'^T X' - \mathrm{tr}X'^T Y'A - \mathrm{tr}(X'^T Y'A)^T + \mathrm{tr}A^T AY'^T Y', \text{ (trace properties)}, \\ &= \mathrm{tr}X'X'^T + \mathrm{tr}Y'Y'^T - 2\mathrm{tr}X'^T Y'A, \end{aligned} \tag{2.14}$$

since $\mathrm{tr}(X'^T Y'A) = \mathrm{tr}(A^T Y'^T X')$. In equation (2.14) only $\mathrm{tr}X'^T Y'A$ depends on A, so that

$$\text{Min } M^2 = \text{Max } \mathrm{tr}X'^T Y'A.$$

Then using the result in the previous section, $Y'^T X' = VLU^T$ is OSR-SVD and the optimal rotation matrix $A = UV^T$, we conclude that

$$\text{Max } \mathrm{tr}X'^T Y'A = \mathrm{tr}L,$$

where L is a matrix of optimal signed singular values. Taking eigen-decomposition of the Procrustes rotation matrix $A = UV^T$ gives one real eigenvalue 1 and the other eigenvalues are complex. This result will be used in Section 3.4.

2.3 Cholesky decomposition

The Cholesky decomposition is a factorization of a symmetric positive definite matrix A into a unique product of a lower unit triangular matrix L, a diagonal matrix G and a transpose of the lower unit triangular matrix L^T, (see Boyd and

Vandenberghe, 2004, pp. 669-671), which is presented as follows, where n=2:

$$
\begin{aligned}
A &= LGL^T, \\
&= \begin{pmatrix} 1 & 0 \\ l_{12} & 1 \end{pmatrix} \begin{pmatrix} g_1 & 0 \\ 0 & g_2 \end{pmatrix} \begin{pmatrix} 1 & l_{12} \\ 0 & 1 \end{pmatrix}, \\
&= \begin{pmatrix} g_1 & g_1 l_{12} \\ g_1 l_{12} & g_1 l_{12}^2 + g_2 \end{pmatrix} \\
&= \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix},
\end{aligned}
$$

where, $g_1 = a_{11}$, $g_2 = a_{22} - \frac{a_{12}^2}{a_{11}}$, and $l_{12} = \frac{a_{12}}{a_{11}}$. We use Cholesky decomposition in Section 4.3.1.

Chapter 3

Estimation process for fitting a regular helix

The α-helix is a smooth curve in 3-dimensional space, as we highlighted previously in Section 1.2. In this chapter our aim is to estimate the registration parameters (orthonormal vectors and shift vector) and the shape parameters (radius and pitch) of the α-helix. Note that, the radius and the pitch are known parameters in the α-helix. The helix spacing parameter is assumed to be known as the ideal value of α-helix parameter.

The estimation of a regular helix's parameters is divided into two stages. In the first stage, an initial estimate of the helix axis is produced. After this, the data are rotated to "semi-canonical" form, for which the estimated helix axis is vertical. After transforming to semi-canonical coordinates we use a least squares method to estimate the remaining parameters. Then, we update our initial estimate of the helix axis by an optimal estimate. We call this method the *Optimized Least Squares* (OptLS), which is defined explicitly later in this chapter in Section 3.5.

Several methods of finding the axis of a regular α-helix were discussed by Christopher et al. (1996). Among these methods we study parametric least

squares (Parlsq), eigenvector method (Eigenfit), and rotational least squares (Rotfit). Finally, we draw a comparison between previous methods and our OptLS. We conclude that our OptLS is the most accurate method to find the axis, followed by Rotfit.

3.1 The difference eigenvector method (Difeigenfit)

This section shows how to estimate the initial helix axis for a regular discrete statistical helix in general coordinates. Given data helix $H_{n\times 3}$ with points $\boldsymbol{y}_i = [y_{i1}, y_{i2}, y_{i3}]^T$, $\mathrm{i} = n_1, \ldots, n_2$, where $n_1 = 1$, and $n_2 = n$, define the increments as

$$\boldsymbol{d}_i = \boldsymbol{y}_i - \frac{\boldsymbol{y}_{i+1} + \boldsymbol{y}_{i-1}}{2}, \quad i = n_1 + 1, \ldots, n_2 - 1. \tag{3.1}$$

Put the vectors $\boldsymbol{d}_i$ into an $(n-2)\times 3$ matrix D, and set $E = D^T D$. Taking the eigen-decomposition of this matrix E gives one very small eigenvalue and two large eigenvalues which are approximately equal to each other. The helix axis $\boldsymbol{w}$ is the eigenvector corresponding to the smallest eigenvalue. In canonical coordinates, vectors $\boldsymbol{d}_i$ have most of the variability in the xy-plane and very small variability along the z-axis. For a mathematical helix,

$$\boldsymbol{d}_i = r(-\frac{1}{2}\cos t_{i-1} - \frac{1}{2}\cos t_{i+1} + \cos t_i)\boldsymbol{u} + r(-\frac{1}{2}\sin t_{i-1} - \frac{1}{2}\sin t_{i+1} + \sin t_i)\boldsymbol{v} + 0\boldsymbol{w},$$

the vectors $\boldsymbol{d}_i$ is perpendicular to $\boldsymbol{w}$. Then $D\boldsymbol{w} = \boldsymbol{0}$, and $E\boldsymbol{w} = \boldsymbol{0}$. Thus, the helix axis is the eigenvector $\boldsymbol{w}$ of E corresponding to eigenvalue zero. The collection of vectors $\boldsymbol{d}_i$ lies in a plane perpendicular to the helix axis. Figure 3.1 illustrates the behaviour of $\boldsymbol{d}_i$ for the mathematical helix in canonical coordinates.

The eigen-decomposition of E determines an initial estimate of a helix axis $\boldsymbol{w}$, but it does not specify the sign of $\boldsymbol{w}$. That is, if $\boldsymbol{w}$ is an eigenvector then so

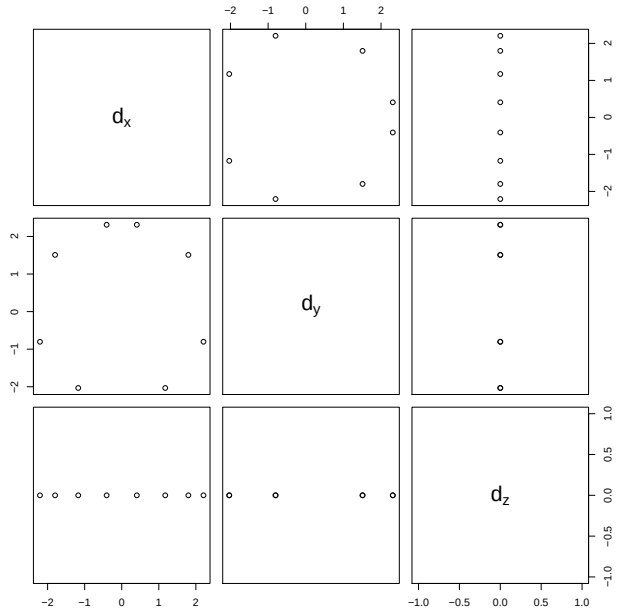

FIGURE 3.1: Pairs plot of a mathematical helix, where variables $\boldsymbol{d_i}$ represent the difference between the original point and the midpoint. All pairs of different variables are plotted as scatter plots above and below the diagonal. The scatter plot between $\boldsymbol{d_x}$ and $\boldsymbol{d_y}$ presents a circle, and the scatter plot between $\boldsymbol{d_z}$ and any other coordinate presents a line since $\boldsymbol{d_z} = \mathbf{0}$.

is $-\boldsymbol{w}$. We need to choose the sign of $\boldsymbol{w}$ and to determine if the helix is right- or left-handed. Consider a mathematical helix $H_{n\times 3}$ in general coordinates, (see Section 1.2.2),

$$\boldsymbol{y}_i = r\cos t_i\boldsymbol{u} + r\sin t_i\boldsymbol{v} + ct_i\boldsymbol{w} + \boldsymbol{b}.$$

We determine $\boldsymbol{w}$ by Difeigenfit up to sign. Then we calculate the vertical distance

between the first and last helix points to make sure that the helix is winding upward i.e. $c > 0$.

$$\begin{aligned}
\boldsymbol{y}_d &= \boldsymbol{w}^T(\boldsymbol{y}_n - \boldsymbol{y}_1) \\
&= \boldsymbol{w}^T(r\cos t_n\boldsymbol{u} + r\sin t_n\boldsymbol{v} + ct_n\boldsymbol{w} + \boldsymbol{b} - r\cos t_1\boldsymbol{u} - r\sin t_1\boldsymbol{v} - ct_1\boldsymbol{w} - \boldsymbol{b}) \\
&= \boldsymbol{w}^T((r\cos t_n - r)\boldsymbol{u} + r\sin t_n\boldsymbol{v} + c(t_n - t_1)\boldsymbol{w}) \\
&= c(t_n - t_1).
\end{aligned}$$

We have $t_i = (i-1)\delta$, and $\delta = \frac{2\pi}{3.6}$, then $t_n - t_1 = (n-1)\frac{2\pi}{3.6} > 0$ i.e. $c > 0$. If this distance $\boldsymbol{w}^T\boldsymbol{y}_n - \boldsymbol{w}^T\boldsymbol{y}_1 < 0$, i.e. $c < 0$, then we have to change the sign of $\boldsymbol{w}$, as in the definition of the helix the pitch c is positive, see Section 1.2.1. Next, to know whether the helix is right- or left-handed, first we project the helix onto the horizontal plane. Let the projection matrix onto vertical axis $\boldsymbol{w}$ be $P = \boldsymbol{w}\boldsymbol{w}^T$ and then the projection matrix onto xy-plane be $I - P$. We can project the helix onto xy-plane as follows

$$\begin{aligned}
\boldsymbol{y}_{p,i} &= (\boldsymbol{y}_i - \boldsymbol{b}) - P(\boldsymbol{y}_i - \boldsymbol{b}) \\
&= (r\cos t_i\boldsymbol{u} + r\sin t_i\boldsymbol{v} + ct_i\boldsymbol{w} + \boldsymbol{b} - \boldsymbol{b}) - \boldsymbol{w}\boldsymbol{w}^T(r\cos t_i\boldsymbol{u} + r\sin t_i\boldsymbol{v} + ct_i\boldsymbol{w} + \boldsymbol{b} - \boldsymbol{b}) \\
&= r\cos t_i\boldsymbol{u} + r\sin t_i\boldsymbol{v}.
\end{aligned}$$

Let $\boldsymbol{y}_{c,i}$ be the cross product of the two successive (adjacent) projected points $\boldsymbol{y}_{p,i}$ and $\boldsymbol{y}_{p,i+1}, i = 1 \ldots n-1$, as follows

$$\begin{aligned}
\boldsymbol{y}_{c,i} &= \boldsymbol{y}_{p,i} \times \boldsymbol{y}_{p,i+1} \\
&= (r\cos t_i\boldsymbol{u} + r\sin t_i\boldsymbol{v}) \times (r\cos t_{i+1}\boldsymbol{u} + r\sin t_{i+1}\boldsymbol{v}) \\
&= (r^2\cos t_i\sin t_{i+1}\boldsymbol{u} + r^2\cos t_{i+1}\sin t_i\boldsymbol{v})\boldsymbol{w} \\
&= r^2\sin(t_{i+1} - t_i)\boldsymbol{w}.
\end{aligned}$$

Then

$$\begin{aligned}\boldsymbol{y}_c &= \sum_{i=1}^{n-1} \boldsymbol{w}^T \boldsymbol{y}_{c,i} \\ &= \sum_{i=1}^{n-1} r^2 \sin(t_{i+1} - t_i)\end{aligned}$$

If $\boldsymbol{y}_c > 0$, then the helix is right-handed, otherwise, left-handed. For example, imagine we have a protein α-helix where $r = 2.3$, $t_i = (i-1)\delta$, and $\delta = \frac{2\pi}{3.6}$, then $t_{i+1} - t_i = \frac{2\pi}{3.6}$ and $\sin(t_{i+1} - t_i) = 0.98 > 0$. Thus, we need $t_{i+1} - t_i < 180$ degrees so that $\sin(t_{i+1} - t_i) > 0$.

3.2 Parametric least squares (Parlsq)

Christopher et al. (1996) concluded in their study that the Parlsq is the fastest method. In this section, we will present the Parlsq method.

For n points on a helix, time $t_i = (i-1)\delta$, where $\delta = \frac{2\pi}{3.6}$, $i = n_1, \ldots, n_2$, $n_1 = 1$, and $n_2 = n$, Christopher et al. (1996) identified three linear equations

$$y_{i1} = b_1 + t_i w_1^* + \varepsilon_{i1}, \tag{3.2}$$

$$y_{i2} = b_2 + t_i w_2^* + \varepsilon_{i2}, \tag{3.3}$$

$$y_{i3} = b_3 + t_i w_3^* + \varepsilon_{i3}, \tag{3.4}$$

where b_j, $j = 1, 2, 3$, are the shift parameters, and $w_j = \frac{w_j^*}{\sqrt{w_1^{*2} + w_2^{*2} + w_3^{*2}}}$ are the helix axis parameters, for all j. Equations (3.2), (3.3) and (3.4) are a special case of equation (1.3) if $r = 0$.

Christopher et al. (1996) fitted each equation separately by using ordinary least squares to find unknown parameters. Finally, they standardized the vector $\hat{\boldsymbol{w}}^*$ to get the direction of the helix as a unit vector $\hat{\boldsymbol{w}} = [w_1, w_2, w_3]^T$. This method is illustrated by examples later in Sections 3.6.1 and 3.7.

3.3 Eigenvector method (Eigenfit)

Christopher et al. (1996) did not require the helix points to be equally spaced as in the previous methods. Their method is easy to follow: First they started by centring the data $\boldsymbol{y}_i = [y_{i1}, y_{i2}, y_{i3}]^T$. Second they found the eigen-decomposition of the least squares information matrix $H^T H$ (i.e. the principal component analysis (PCA). The eigenvector with the highest eigenvalue is an approximation of the helix axis $\boldsymbol{w}$, which has the most variation, other eigenvalues are expected to be approximately equal. However, this is not always the case. As Christopher et al. (1996) mentioned, this method is more accurate for long helices than for short ones. In biology most of helices are long, see Section 1.5.1. The eigenvalues for a short helix are almost equal to each other, so that we can not determine which of the axes is the helix axis. We illustrate Eigenfit by simulation later in Sections 3.6.2 and 3.7.

3.4 Rotational least squares (Rotfit)

The Rotfit method was recommended by Christopher et al. (1996) as most accurate method for identifying helix axis among their studied methods (more details are in Section 3.7). To motivate the method, we start with a mathematical helix $H_{n\times 3}$ with landmarks $\boldsymbol{y}_i, i = n_1, \ldots, n_2$, where $n_1 = 1$, and $n_2 = n$. Consider two modified versions of $H_{n\times 3}$, the first one is H_1 of size $(n-1)\times 3$, obtained by deleting the last point of the original dataset (the n^{th} point). The second one is H_2 of size $(n-1)\times 3$, obtained by deleting the first point (the 1^{th} point). These two data sets are related. If you twist one of the helices about its axis and shift it along its axis, this screwing action maps H_1 onto H_2, i.e. mapping from $\boldsymbol{y}_i$ to $\boldsymbol{y}_{i+1}$.

In order to fit the helix, we need to estimate how much the helix moves forward and rotates as time moves forward one step. We rotates the helix one

atom ahead i.e. rotate H_1 one point ahead to reach H_2. This is obtained by solving the equation

$$\boldsymbol{y}_{i+1} = \Gamma \boldsymbol{y}_i, \tag{3.5}$$

where $i = n_1, \ldots, n_2 - 1$, $n_1 = 1, n_2 = n$, the rotation matrix is Γ. The movement forward is representing by the reindexing from i to $i+1$. We centre two datasets H_1 and H_2 and denote them by H_1' and H_2'' respectively, where

$$\boldsymbol{y}_i' = \boldsymbol{y}_i - \bar{\boldsymbol{y}}^{'},$$
$$\boldsymbol{y}_{i+1}'' = \boldsymbol{y}_{i+1} - \bar{\boldsymbol{y}}^{''},$$

where $\bar{\boldsymbol{y}}^{'} = [\bar{y}_1^{'}, \bar{y}_2^{'}, \bar{y}_3^{'}]^T$ is the mean of the first dataset, and $\bar{\boldsymbol{y}}^{''} = [\bar{y}_1^{''}, \bar{y}_2^{''}, \bar{y}_3^{''}]^T$ is the mean of the second data set, as follows

$$\bar{y}_1^{'} = \frac{1}{n-1}\sum_{i=1}^{n-1} \boldsymbol{y}_{i1}, \quad \bar{y}_2^{'} = \frac{1}{n-1}\sum_{i=1}^{n-1} \boldsymbol{y}_{i2}, \quad \bar{y}_3^{'} = \frac{1}{n-1}\sum_{i=1}^{n-1} \boldsymbol{y}_{i3},$$

and

$$\bar{y}_1^{''} = \frac{1}{n-1}\sum_{i=1}^{n-1} \boldsymbol{y}_{i+1,1}, \quad \bar{y}_2^{''} = \frac{1}{n-1}\sum_{i=1}^{n-1} \boldsymbol{y}_{i+1,2}, \quad \bar{y}_3^{''} = \frac{1}{n-1}\sum_{i=1}^{n-1} \boldsymbol{y}_{i+1,3};$$

and (3.5) can be written as

$$\boldsymbol{y}_{i+1}' = \Gamma \boldsymbol{y}_i''. \tag{3.6}$$

To find the helix axis and the rotation matrix, which rotates about the axis, we can use the *Procrustes* procedure (see Kent and Mardia, 2001). The *Procrustes* procedure is easy to implement in R. For the mathematical helix, the Procrustes procedure gives the exact rotation matrix that rotates one version onto the other.

For a statistical helix, we follow the same equation (3.6) and add noise as follows

$$\boldsymbol{y}'_{i+1} = \Gamma \boldsymbol{y}''_i + \boldsymbol{\varepsilon_{i+1}},$$

where $\boldsymbol{\varepsilon_{i+1}}$ is the error vector. Let $Z = H_2'^T H_1'$, then we decompose Z using the standard singular value decomposition (in Section 2.2.4), $Z = ULV^T$. The optimal choice of the rotation matrix can be found as $\Gamma = UV^T$ (see Section 2.2.5 and Andrade et al. (2004) for more details). Taking eigen-decomposition of the Procrustes rotation matrix Γ gives the helix axis, which is the eigenvector corresponding to the real eigenvalue 1. Other eigenvalues are complex. An illustration of this method on simulated data is given in Sections 3.6.1 and 3.7.

3.5 Estimation process for fitting regular helix

This section describes the estimation process of a regular statistical data helix parameters $\boldsymbol{y}_i, i = n_1, \ldots, n_2$, where the dataset is assumed to follow the model given in equation (1.3). As we described earlier, the estimation process consists of two stages. The first stage is to estimate an initial helix axis $\boldsymbol{w}$. From the helix axis we can first find a 3×3 rotation matrix, which rotates the helix into "semi-canonical coordinates", so that the new axis direction $\boldsymbol{w}$ is the north pole. After that, we obtain the least squares estimates of the other six unknown helix parameters: the radius r; the pitch c; the shift parameters b_1, b_2, b_3; the angle τ; and the residuals sum of squares (RSS). Finally, we optimize these estimates by minimizing the residual sum of squares over the choice of axis $\boldsymbol{w}$.

In addition, a special case is when the helix axis is known, which could be in general coordinates, or in semi-canonical coordinates. If the known axis is in general coordinates, then we need to rotate the helix to be in semi-canonical coordinates to be able to estimate the other parameters, which is described in detail in Section 3.5.4.

3.5.1 Stage 1: Estimate of an initial axis

In this section, we estimate the initial helix axis for a regular discrete statistical helix in general coordinates $H_{n\times 3}$. We can estimate the initial axis by our estimation method Difeigenfit described in Section 3.1 or by a method from the literature, such as Rotfit method described in Section 3.4. The exact choice does not matter since we update this initial estimate later in last step (see Section 3.5.3). Recall, the regular discrete statistical helix model (1.3) for $\boldsymbol{y}(t_i) = [y_{i1}, y_{i2}, y_{i3}]^T, i = n_1, \ldots, n_2, n_1 = 1$ and $n_2 = n$, points around the helix is

$$\boldsymbol{y}(t_i) = r\cos(t_i)\boldsymbol{u} + r\sin(t_i)\boldsymbol{v} + ct_i\boldsymbol{w} + \boldsymbol{b} + \boldsymbol{\varepsilon}_i.$$

After we estimate the initial axis we can define a 3×3 rotation matrix, Γ_1, which takes the helix axis to point to the north pole i.e. $\boldsymbol{w}^T\Gamma_1 = [0, 0, 1]^T$, where $\boldsymbol{w}$ is the third column, and other two columns are any orthonormal columns that satisfy the properties of a rotation matrix (e.g. Arfken and Weber, 2001, pp. 195-197). This rotation matrix, Γ_1, is not uniquely determined the third column, but not the first and second columns. For Difeigenfit, we can choose these columns to be the eigenvectors of E (see Section 3.1). The right-handed helix winds clockwise upwards to north pole (i.e. the pitch c is positive) after multiplying it by the rotation matrix Γ_1, where $\det(\Gamma_1) = 1$, (e.g. see Murray et al., 1994). If, however, the determinant of Γ_1 is negative, then we need to change the sign of $\boldsymbol{u}$ or $\boldsymbol{v}$. Overall, we need to look at the sign of determinant of $\Gamma_1 = [\boldsymbol{u}\ \boldsymbol{v}\ \boldsymbol{w}]$ to understand if this is a rotation matrix or a reflection matrix.

3.5.2 Stage 2: Estimation of the shape and registration parameters

After the first stage, the helix data matrix of size $n \times 3$ is in the semi-canonical coordinates (see Section 1.2.3)

$$H^o = H\Gamma_1 = \begin{bmatrix} z_{n_1,1} & z_{n_1,2} & z_{n_1,3} \\ \vdots & \vdots & \vdots \\ z_{n_2,1} & z_{n_2,2} & z_{n_2,3} \end{bmatrix},$$

where the new axis direction $\boldsymbol{w}_1$ is to the north pole i.e. $\boldsymbol{w}_1^T = \boldsymbol{w}^T\Gamma_1 = \boldsymbol{w}_0^T, \boldsymbol{w}_0 = [0,0,1]^T$. This section focuses on rotating of the helix about $\boldsymbol{w}_1$ and shifting it to the origin, so that the initial point is proportional to $[r,0,0]^T$. The helix model in equation (1.3) can now be expressed as

$$\begin{aligned} \boldsymbol{z}(t_i) &= r\cos(t_i-\tau)\begin{bmatrix}1\\0\\0\end{bmatrix} + r\sin(t_i-\tau)\begin{bmatrix}0\\1\\0\end{bmatrix} + ct_i\begin{bmatrix}0\\0\\1\end{bmatrix} + \boldsymbol{b}^o \\ &= (\alpha_1\cos t_i + \alpha_2\sin t_i)\begin{bmatrix}1\\0\\0\end{bmatrix} + (\alpha_1\sin t_i - \alpha_2\cos t_i)\begin{bmatrix}0\\1\\0\end{bmatrix} + ct_i\begin{bmatrix}0\\0\\1\end{bmatrix} + \boldsymbol{b}^o, \end{aligned} \tag{3.7}$$

where $i = n_1, \ldots, n_2$, $\alpha_1 = r\cos\tau$, $\alpha_2 = r\sin\tau$ and τ is the angle measured between the standard starting point $(r,0,0)$ and the initial point after the first rotation $\boldsymbol{y}_1^T\Gamma_1$. The equation (3.7) can be written in matrix form as a multivariate linear regression

$$\boldsymbol{z} = X\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

where the response vector of size $3n \times 1$ is

$$\begin{aligned}\boldsymbol{z} &= \begin{bmatrix} z_{n_1,1} & \cdots & z_{n_2,1} & z_{n_1,2} & \cdots & z_{n_2,2} & z_{n_1,3} & \cdots & z_{n_2,3} \end{bmatrix}^T \\ &= \begin{bmatrix} \boldsymbol{z}_1^T & \boldsymbol{z}_2^T & \boldsymbol{z}_3^T \end{bmatrix}^T,\end{aligned}$$

and the design matrix of size $3n \times 6$ is

$$X = \begin{bmatrix} 1 & 0 & 0 & \cos t_{n_1} & \sin t_{n_1} & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & \cos t_{n_2} & \sin t_{n_2} & 0 \\ 0 & 1 & 0 & \sin t_{n_1} & -\cos t_{n_1} & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & 0 & \sin t_{n_2} & -\cos t_{n_2} & 0 \\ 0 & 0 & 1 & 0 & 0 & t_{n_1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 1 & 0 & 0 & t_{n_2} \end{bmatrix},$$

which can be written as a matrix of vectors

$$= \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \boldsymbol{c} & \boldsymbol{s} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \boldsymbol{c} & -\boldsymbol{c} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \boldsymbol{t} \end{bmatrix},$$

where

$$\boldsymbol{c} = \begin{bmatrix} \cos t_{n_1} \\ \vdots \\ \cos t_{n_2} \end{bmatrix}, \quad \boldsymbol{s} = \begin{bmatrix} \sin t_{n_1} \\ \vdots \\ \sin t_{n_2} \end{bmatrix}, \quad \boldsymbol{t} = \begin{bmatrix} t_{n_1} \\ \vdots \\ t_{n_2} \end{bmatrix}, \quad \mathbf{1} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}, \quad \mathbf{0} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix},$$

and the regression parameters $\boldsymbol{b}^o, \alpha_1, \alpha_2$ and c can be viewed as a 6×1 vector

$$\begin{aligned}\boldsymbol{\beta} &= \begin{bmatrix} b_1 & b_2 & b_3 & \alpha_1 & \alpha_2 & c \end{bmatrix}^T \\ &= \begin{bmatrix} \boldsymbol{\beta}_0^T & \boldsymbol{\beta}_1^T \end{bmatrix}^T,\end{aligned}$$

where $\boldsymbol{\beta}_0 = [b_1, b_2, b_3]^T$, and $\boldsymbol{\beta}_1 = [\alpha_1, \alpha_2, c]^T$. To estimate the parameters, the least squares method seeks to minimize the sum of squared errors (residuals) of the helix model (Mardia et al., 2018). The least squares solution provides estimates of the unknown parameters. These estimates are also the maximum likelihood estimates in a general regression model with normal errors (see Garthwaite et al., 2002, p. 61). Our model is a special case of the general regression model, so least squares gives the same result as maximum likelihood estimation.

In order to fit the regression model, we centre each vector $\boldsymbol{z}_j = [z_{n_1,j} \cdots z_{n_2,j}]^T, j = 1, 2, 3$, of size $n \times 1$, to simplify the algebra. We also centre the design data matrix X, so that the mean of each column becomes zero, as

$$X' = \begin{bmatrix} \boldsymbol{c}' & \boldsymbol{s}' & 0 \\ \boldsymbol{s}' & -\boldsymbol{c}' & 0 \\ 0 & 0 & \boldsymbol{t}' \end{bmatrix}.$$

Note that the columns of X' are orthogonal. The centred variables are

$$\boldsymbol{c}' = \begin{bmatrix} c'_{n_1} \\ \vdots \\ c'_{n_2} \end{bmatrix}, \quad \boldsymbol{s}' = \begin{bmatrix} s'_{n_1} \\ \vdots \\ s'_{n_2} \end{bmatrix}, \quad \boldsymbol{t}' = \begin{bmatrix} t'_{n_1} \\ \vdots \\ t'_{n_2} \end{bmatrix},$$

where

$$c'_i = \cos t_i - \overline{C}, \quad s'_i = \sin t_i - \overline{S}, \quad t'_i = t_i - \overline{T},$$

$$\overline{C} = \frac{1}{n}\sum_{i=n_1}^{n_2} \cos t_i, \quad \overline{S} = \frac{1}{n}\sum_{i=n_1}^{n_2} \sin t_i, \quad \overline{T} = \frac{1}{n}\sum_{i=n_1}^{n_2} t_i, \quad \overline{R} = \sqrt{\overline{C}^2 + \overline{S}^2},$$

and $\bar{\boldsymbol{z}} = [\bar{\boldsymbol{z}}_1, \bar{\boldsymbol{z}}_2, \bar{\boldsymbol{z}}_3]^T$, $\boldsymbol{z}' = \boldsymbol{z} - \bar{\boldsymbol{z}} = [z'_{n_1,1}, \ldots, z'_{n_2,1}, z'_{n_1,2}, \ldots, z'_{n_2,2}, z'_{n_1,3}, \ldots, z'_{n_2,3}]^T$, where $\bar{\boldsymbol{z}}_j = \frac{1}{n}\sum_{i=n_1}^{n_2} z_{i,j}$ for all $j = 1, 2, 3$ and $n = n_2 - n_1 + 1$. We can also derive the shift vector $\hat{\boldsymbol{b}} = [\hat{b}_1, \hat{b}_2^o, \hat{b}_3]^T$ as follows

$$\begin{aligned}
\hat{b}_1 &= \overline{z}_1 - \hat{\alpha}_1\overline{C} - \hat{\alpha}_2\overline{S}, \\
\hat{b}_2 &= \overline{z}_2 - \hat{\alpha}_1\overline{S} + \hat{\alpha}_2\overline{C}, \\
\hat{b}_3 &= \overline{z}_3 - c\overline{T}.
\end{aligned}$$

Then the corresponding least squares estimator is given by

$$||\boldsymbol{z}' - X'\boldsymbol{\beta}_1||^2. \tag{3.8}$$

The least squares estimator of the parameter vector $\boldsymbol{\beta}_1$ can be derive by taking the first derivative of (3.8) with respect to each parameter, and setting it equal to zero (MLE of the parameters). Then the least squares estimator takes the form

$$\begin{aligned}
\hat{\alpha}_1 &= \sum_{i=n_1}^{n_2}(c'_i z'_{i1} + s'_i z'_{i2})/\{n(1-\overline{R}^2)\}, \\
\hat{\alpha}_2 &= \sum_{i=n_1}^{n_2}(s'_i z'_{i1} + c'_i z'_{i2})/\{n(1-\overline{R}^2)\}, \\
\hat{c} &= \sum_{i=n_1}^{n_2} t'_i z'_{i3}/\{\sum_{i=n_1}^{n_2}(t_i - \overline{T})^2\}.
\end{aligned}$$

Since $\alpha_1 = r\cos\tau$ and $\alpha_2 = r\sin\tau$, then $\hat{r}$ can be derived as

$$\begin{aligned}
r^2(\cos^2\tau + \sin^2\tau) &= \alpha_1^2 + \alpha_2^2 \\
r^2 &= \alpha_1^2 + \alpha_2^2.
\end{aligned}$$

Then

$$\hat{r} = \sqrt{\hat{\alpha}_1^2 + \hat{\alpha}_2^2},$$

and the angle $\hat{\tau}$ can be derived as

$$\frac{r\sin\tau}{r\cos\tau} = \frac{\alpha_2}{\alpha_1}$$
$$\tan\tau = \frac{\alpha_2}{\alpha_1},$$

so that

$$\hat{\tau} = \text{atan2}(\hat{\alpha}_2, \hat{\alpha}_1).$$

The least squares fitted values for $\boldsymbol{z}(t_i)$ is

$$\hat{\boldsymbol{z}}(t_i) = \hat{r}\cos(t_i - \hat{\tau}) \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \hat{r}\sin(t_i - \hat{\tau}) \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + \hat{c}t_i \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} \hat{b}_1 \\ \hat{b}_2 \\ \hat{b}_3 \end{bmatrix}.$$

In addition, we can derive the residual sum of squares, which is a function of the helix axis $\boldsymbol{w}$, as

$$\text{RSS}(\boldsymbol{w}) = ||Z - \hat{Z}||^2,$$

where Z and $\hat{Z}$ are the data and the fitted matrices respectively. After estimating the parameters for the helix, where the axis direction is $\boldsymbol{w}^T\Gamma_1 = \boldsymbol{w}_0$, the n$\boldsymbol{u}$ and $\boldsymbol{v}$ take the form

$$\begin{bmatrix} \boldsymbol{u} & \boldsymbol{v} \end{bmatrix} = \begin{bmatrix} \cos(\hat{\tau}) & \sin(\hat{\tau}) \\ -\sin(\hat{\tau}) & \cos(\hat{\tau}) \end{bmatrix}.$$

To put the helix in canonical form, we can rotate the helix about its axis $\boldsymbol{w}_1$ by the rotation matrix $\Gamma_2(\hat{\tau})$, where

$$\Gamma_2(\hat{\tau}) = \begin{bmatrix} \cos(\hat{\tau}) & \sin(\hat{\tau}) & 0 \\ -\sin(\hat{\tau}) & \cos(\hat{\tau}) & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Overall, after the first rotation of the axis (applying Γ_1) i.e. $\boldsymbol{w} = \boldsymbol{w}_0$, we rotate the data helix about the axis (applying Γ_2), and finally we shift the data helix by $\hat{\boldsymbol{b}} = [b_1, b_2, b_3]^T$, so that the data helix is in canonical coordinates, as follows

$$Z^*_{(n\times 3)} = (Z_{(n\times 3)} - 1_n \hat{\boldsymbol{b}}_{(3\times 1)})\Gamma_2.$$

3.5.3 Optimized least squares (OptLS) method

In stage 2, in Section 3.5.2, we have assumed that $\boldsymbol{w}$ is known and constructed a goodness of fit statistic given by the residual sum of squares RSS($\boldsymbol{w}$). In this section, we improve the fit by minimizing RSS($\boldsymbol{w}$) over $\boldsymbol{w}$. The function RSS($\boldsymbol{w}$) can be evaluated for any $\boldsymbol{w}$, and can be minimized numerically using for example the function `nlm` in R (see R Core Team, 2014).

We have an optimization problem of curved manifold, namely the sphere, where the unit vector $\boldsymbol{w}$ is under the constraint $||\boldsymbol{w}||^2 = 1$, but optimization methods work more easily with unconstrained parameters. To parametrize $\boldsymbol{w}$, first it is helpful to rotate an initial estimate $\boldsymbol{w}_{\text{init}}$ to point towards the north pole $[0, 0, 1]^T$. Then a general 3-dimensional unit vector $\boldsymbol{w}$ can be represented in 2-dimensional stereographic coordinates about $\boldsymbol{w}_{\text{init}}$ by a 2-dimensional vector $[p_1, p_2]^T$, where

$$\begin{aligned}
\hat{w}_1 &= \frac{2\hat{p}_1}{1 + \hat{p}_1^2 + \hat{p}_2^2}, \\
\hat{w}_2 &= \frac{2\hat{p}_2}{1 + \hat{p}_1^2 + \hat{p}_2^2}, \\
\hat{w}_3 &= \frac{-1 + \hat{p}_1^2 + \hat{p}_2^2}{1 + \hat{p}_1^2 + \hat{p}_2^2}.
\end{aligned}$$

Then the `nlm` procedure works on the two free parameters p_1 and p_2 with an initial value $[p_1, p_2]^T = [0, 0]^T$. Minimizing the residual sum of squares is equivalent to

maximizing the log likelihood since the log likelihood for the axis, after optimizing over the other parameters, is given by

$$-\frac{n}{2}\log(2\pi) - \frac{n}{2}\log(\frac{\text{RSS}(\boldsymbol{w})}{n}) - \frac{n}{2}.$$

3.5.4 A known helix axis

We previously described in detail how to fit a regular helix in general coordinates where the axis is unknown. In this subsection, we discuss two special cases in which the helix axis $\boldsymbol{w}$ is known. First we discuss the case when the helix axis is in general coordinates, and second the case when the helix axis is in semi-canonical coordinates.

Known helix axis $\boldsymbol{w}$ in general coordinates

For a known helix axis $\boldsymbol{w}$ in general coordinates, we start from stage 2 of our estimation procedure which is described in Section 3.5.2, and Γ_1 is a 3×3 rotation matrix where the third column is the axis $\boldsymbol{w}$, and the other two vectors can be any two columns that satisfy the rotation matrix properties. We can determine the other two columns of Γ_1 by $\boldsymbol{w}$. The matrix Γ_1 can be taken as the eigenvectors of $R = I_3 - \boldsymbol{w}\boldsymbol{w}^T$, where I_3 is the 3×3 identity matrix. We make sure that the Γ_1 will rotate the helix so that $c > 0$, if $\boldsymbol{w}^T\boldsymbol{y}_{n_2} - \boldsymbol{w}^T\boldsymbol{y}_{n_1} > 0$, otherwise, we need to change the sign of the third column of the matrix Γ_1. In addition, we need to make sure that Γ_1 is a rotation matrix not a reflection matrix, so the determinant of R should be equal to 1, otherwise we need to change the sign of one of the columns 1 or 2. After that we can use the OptLS from stage 2 in Section 3.5.2.

Known helix axis $\boldsymbol{w}$ in semi-canonical coordinates

For a known vertical helix axis $\boldsymbol{w} = \boldsymbol{w}_0$ we start the OptLS from stage 2 in Section 3.5.2 of our estimation procedure and we can set the rotation matrix $\Gamma_1 = I_3$.

3.6 Drawbacks of Parlsq, Eigenfit, and Rotfit methods

In this section, we look at several other methods of estimating helix axis and investigate their drawbacks. We study Parlsq in Section 3.2, Eigenfit in Section 3.3, and Rotfit in Section 3.4 methods in order to study how these methods estimate helix axis in different shapes of helices. Helices can come short fat like tuna can or tall and thin like beans can.

We start by illustrating each method on a single example of a mathematical helix. In addition, we carry out $q = 1000$ simulation studies, simulating each as follows: first we create n points that lie on an regular helix in canonical coordinates (see Section 1.2.1); then we simulate q independently and identically distributed random errors from $N_3(\mathbf{0}, \sigma^2 I_3)$, $\sigma^2 = 0.05$ see Mardia et al. (2018),(for simulation see Lele and Richtsmeier, 2001, Section 2.8). The helix parameters are chosen to closely mimic a protein α-helix, which has radius $r = 2.3$, pitch $c = \frac{5.4}{2\pi}$, and the angle between successive points on the helix is $\delta = \frac{2\pi}{3.6}$ radians. For convenience we set the shift parameter $\boldsymbol{b} = 0$ and the times $t = (i-1)\delta$. There are 3.6 points per loop, fewer points per loop than were used in Christopher et al. (1996) (approximately 12 points per loop). These simulation studies are carried out on various n, r, c and δ to study different shapes of helices. Helices can be short and fat, like a tuna can, or tall and thin like a baked bean can.

Let the angle between the estimated axis $\hat{\boldsymbol{w}}$ and the true axis $\boldsymbol{w}$ be θ. If $\theta = 0$, the estimated axis is a perfect fit, so $\hat{\boldsymbol{w}}\boldsymbol{w}^T = \cos\theta = 1$ (Deville et al., 2008), and $1 - \hat{\boldsymbol{w}}^T\boldsymbol{w} = 0$. If the estimated axis is not a perfect fit and the true

axis $\boldsymbol{w} = [0, 0, 1]^T$, then

$$\begin{aligned} 1 - \hat{\boldsymbol{w}}^T \boldsymbol{w} &= 1 - \hat{w}_3 \\ &= 1 - \sqrt{1 - \hat{w}_1^2 - \hat{w}_2^2} \\ &\approx \frac{1}{2}(\hat{w}_1^2 + \hat{w}_2^2), \text{(from Taylor series)}. \end{aligned}$$

For a sample $j = 1, \ldots, q$, we have q estimated axes $\hat{\boldsymbol{w}}_j$ and the average of these estimated axes is

$$\hat{\bar{\boldsymbol{w}}} = \frac{1}{q} \sum_{j=1}^{q} \hat{\boldsymbol{w}}_j.$$

Then the sample mean

$$\hat{\bar{\boldsymbol{w}}}^T \boldsymbol{w} = \frac{1}{q} \sum_{j=1}^{q} \hat{w}_3^{(j)},$$

and

$$\begin{aligned} 1 - \hat{\bar{\boldsymbol{w}}}^T \boldsymbol{w} &= \frac{1}{q}(q - \sum_{j=1}^{q} \hat{w}_3^{(j)}) \\ &\approx \frac{1}{q}\frac{1}{2} \sum_{j=1}^{q} (\hat{w}_1^{(j)2} + \hat{w}_2^{(j)2}). \end{aligned}$$

If the helix is balanced, then $1 - \hat{\bar{\boldsymbol{w}}}^T \boldsymbol{w}$ is the mean squared error (MSE). Then for a helix which balance almost balanced,

$$1 - \hat{\bar{\boldsymbol{w}}}^T \boldsymbol{w} \approx \frac{1}{2}(\text{var}(\hat{w}_1) + \text{var}(\hat{w}_2)),$$

since $E(\hat{w}_1) = 0 = E(\hat{w}_2)$.

From these simulation studies, we conclude that Rotfit is the most accurate method among these three methods that studied by Christopher et al. (1996), whereas, Parlsq and Eigenfit are not effective methods for estimating the helix

axis. For a short fat helix Parlsq gives a poor estimation of a helix axis and Eigenfit can not even estimate the helix axis.

3.6.1 Parlsq method

In this subsection, we simulate three sets of data and we apply the methods discussed by Christopher et al. (1996), in order to study how Parlsq estimates the helix axis in different shapes of helices. The first set of helices contains far fewer points ($n = 7$) than Christopher et al. (1996) used in their example ($n = 36$) to show the effect of changing the number of points. The second set of helices are also short with $n = 7$ points but with wider radius $r = 7$ to show the effect of changing the number of points relative to the radius. The third set of helices are also short $n = 7$ with wider radius $r = 7$ but with shorter distance per one turn $c = 0.1$ to show the effect of radius r relative to the length of the helix nc.

First we start with one set of $n = 7$ points on a mathematical helix in canonical form (see Section 1.2.1). The helix axis in this case is $\boldsymbol{w} = [0, 0, 1]^T$. In order to apply Parlsq to the data, we carry out an ordinary least squares estimation for each of the three coordinate components as in equations (3.2), (3.3) and (3.4).

Figure 3.2 shows one plot of our simulated helix and a plot for each set of coordinates versus the index i which is related to the "time" by $t_i = (i - 1)\delta$. Panels (b), (c) and (d) display the theoretical behaviour of the equation of a mathematical helix in canonical coordinates. Panel (d) shows that the z-axis is the helix axis because it is straight. Further, the vertical axis of panel (d) is the highest scale axis between the other three panels which is lies between $z = 0$ and $z = 8.899$. The chart also displays panels (b) and (c) of each set of x and y coordinates, respectively, which both have vertical scale axis with fewer scaling options that lie between $-r$ and r. The panel of the y-coordinate versus t_i presents sine waves in dots and the panel of the x-coordinate versus t_i presents

cosine waves in dots. If we ignore the waves and deal with each equation as a line, we can find the slope w_j and the intercept b_j. After applying the least squares method to each line, we standardized the estimated vector to get the helix axis

$$\hat{\boldsymbol{w}} = [-0.2026, -0.1170, 0.9722]^T. \tag{3.9}$$

Figure 3.2 also shows the fitted line (dashed line) for the three coordinates. We expect the two fitted lines for x and y plots to be flat (horizontal), but in fact they are sloped, so this is not a good fit.

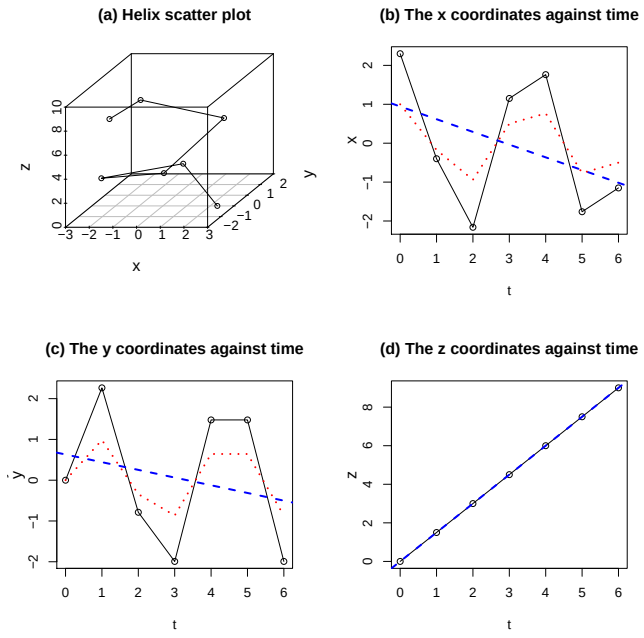

FIGURE 3.2: Plot of simulated discrete mathematical helix of seven points in canonical coordinates. Panel (a): the 3d simulated helix plot; panels (b)-(d): each coordinates versus index i. The solid black line connects the points of the helix. The dashed blue lines in (b) - (d) show the fitted by Parlsq.

It can be seen from (3.9) that the estimate of the helix axis using the Parlsq is different from the canonical helix axis $\boldsymbol{w} = [0, 0, 1]^T$. In addition, the residual sum of squares for each linear model shows that all the variability is in the x and y coordinates, whereas there is zero variability on the z coordinate (i.e. a perfect fit) for our simulated data without errors. This is because z is essentially the index number multiplied by c and δ.

Next, we simulate $q = 1000$ helices with $n = 7$ points as explained in Section 3.6. The mean squared error of the helix axis estimate $1-(\hat{\boldsymbol{w}}^T\boldsymbol{w})$ from Parlsq is 0.031. Secondly, we simulate a set of 1000 helices with $n = 7$ points around a canonical helix with a wide radius of $r = 7$ and keep all other parameters, as in the previous simulation, identical to the protein α-helix. To check the effect of the ratio of radius to cn on the estimation of a helix axis, we then carry out an ordinary least squares as in the previous example. The MSE $1-(\hat{\boldsymbol{w}}^T\boldsymbol{w})$, from Parlsq is 0.207. This shows that Parlsq gives a poorer estimation for a short fat helix, where the ratio of r to cn is greater than 1, compared to the estimation in the previous simulation, where this ratio was less than 1.

In addition, we simulate a set of 1000 short fat helices with $n = 7$ points, $r = 7$ but shorter than the previous set as the vertical distance of one helix turn is shorter $c = 0.1$ and fewer points per loop, as $\delta = \frac{4\pi}{3.6}$. The estimate of the helix axis becomes much poorer by Parlsq when the ratio of r to cn is 10, as $1-(\hat{\boldsymbol{w}}^T\boldsymbol{w}) = 0.612$ see Table 3.1.

TABLE 3.1: MSE comparison of different simulated set of 1000 helices of $n = 7$ and $\sigma^2 = 0.05$ with various r, c and δ.

Method	set 1	set 2	set 3
r	2.3	7	7
c	$\frac{5.4}{2\pi}$	$\frac{5.4}{2\pi}$	0.1
δ	$\frac{2\pi}{3.6}$	$\frac{2\pi}{3.6}$	$\frac{4\pi}{3.6}$
Parlsq	0.031	0.207	0.612
OptLS	6×10^{-4}	3×10^{-4}	3×10^{-4}

Overall, after applying the Parlsq method to various datasets, we conclude that OptLS is effective for all data helices, whereas Parlsq is not. Table 3.1 presents the MSE for all simulations provides by OptLS which show that the helix axis estimates are much better by OptLS than Parlsq. There are four characteristics of the helix that can be problematic with Parlsq: the total number of points; the number of points per loop; the pitch; and the ratio of r relative to cn. As n increases we obtain a better estimation of the canonical α-helix axis and

a better least squares fit. However, it is not enough only to increase the number of points; one must also check whether the data helix is more than one loop and that there are at least 3 points per loop. Furthermore, Parlsq does not work with a helix if the ratio of r relative to length of the helix cn is greater than or equal to one. As soon as one has a reasonable value of n, you could obtain a good estimate of the axis. We obtained good results for $n = 15$ points on a canonical α-helix.

3.6.2 Eigenfit method

In order to study Eigenfit we simulate two sets of 1000 helices as explained in Section 3.6. A set of long thin helices of $n = 30$ and a set of short fat helices of $n = 7$, and $r = 7$.

We start with a mathematical helix (see Section 1.2.1) with $n = 30$ points in canonical form, where the parameters mimic a protein α-helix. For this data, Eigenfit provides three eigenvalues: 19248.9, 81.5 and 76.9. The eigenvector corresponding to the largest eigenvalue of the covariance matrix always points in the direction of the largest variance of the data, i.e. the helix axis.

In addition, we create another mathematical helix of $n = 9$ points in canonical form, where the radius of the helix is $r = 9$, and shorter distance per helix turn $c = 0.1$. Here the Eigenfit provides eigenvalues 364.690, 364.500, and 6.024. Recall the axis according to the concept of Eigenfit is the eigenvector corresponding to the largest eigenvalue. Thus this result did not provide us with a clue for a helix axis since the first two eigenvalues are approximately equal. Whereas, OptLS provides us with eigenvalues of 409.9, 371.1 and 0, and according to the OptLS the helix axis is the eigenvector that corresponds to the smallest eigenvalue λ=0 in canonical form, which is $\boldsymbol{w} = [0, 0, 1]^T$. Therefore, OptLS provides a better result than the Eigenfit method does.

Next, we simulate 1000 long helices of $n = 30$ mimic protein α-helix with $\sigma^2 = 0.05$ as in Section 3.6 and we obtained a good estimation of the helix axis by Eigenfit and OptLS as the MSE $1-(\hat{\hat{\boldsymbol{w}}}^T\boldsymbol{w})$ are 2×10^{-5} and 8×10^{-6}, respectively. Moreover, we simulate 1000 short fat helices of $n = 7$ points and wider radius of $r = 7$ and obtained MSE $1 - (\hat{\hat{\boldsymbol{w}}}^T\boldsymbol{w})$ from Eigenfit is 0.665, which suggests a poor estimation.

In conclusion, for a short helix, the Eigenfit method gives two eigenvalues that are approximately equal, so we do not know which of these eigenvectors is the helix axis. However, for a long α-helix it works well, as mentioned in Christopher et al. (1996). Thus, the simulation of short helices shows a poorer estimation of the helix axis by Eigenfit for a short helix, whereas OptLS works well in all the cases.

3.6.3 Rotfit method

In this section, we study Rotfit. We begin by applying the Rotfit method (explained in section 3.4) to mathematical helices, and it works well as the input helix matches the estimated helix. For a simulated helix that mimic protein α-helix with $n = 15$ and the errors are assumed normally distributed with mean 0 and variance 0.05, the axis estimate by Rotfit and by OptLS methods are $\hat{\boldsymbol{w}}_R^T\boldsymbol{w} = 0.999995$, and $\hat{\boldsymbol{w}}^T\boldsymbol{w} = 0.999998$, respectively. Then the MSE of the helix axis estimate with the true axis $1 - \hat{\hat{\boldsymbol{w}}}^T\boldsymbol{w}$ by Rotfit 5×10^{-6} and by OptLS 2×10^{-6} indicate good estimates of the helix axis by the two methods. The MSE for OptLS shows that the OptLS estimate of the helix axis is more accurate than the Rotfit estimate. Figure 3.3 shows the estimate of the helix axis using Rotfit in red as it rotates the helix one atom ahead.

We simulate 4 sets of 1000 helices as explained in Section 3.6: the parameters of the first and second sets are mimic protein α-helix with $n = 15$ and $n = 7$, respectively; for the third set we set $n = 7$ and $r = 7$ whilst keeping the other

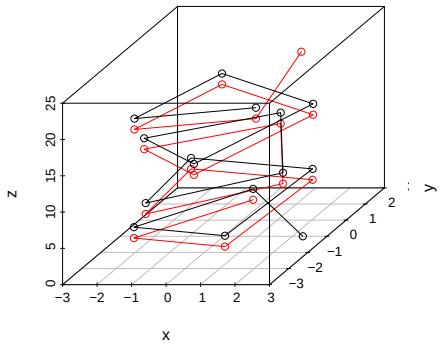

FIGURE 3.3: The figure presents two helices: the data helix in black and the fitted helix using Rotfit in red.

parameters as before; and finally a set helices with $n = 7, r = 7, c = 0.1$ and $\delta = \frac{4\pi}{3.6}$. We summarize the MSE $1 - (\hat{\boldsymbol{w}}^T \boldsymbol{w})$, of the estimates of the helix axis using Rotfit and OptLS in Table 3.2. Table 3.2 shows that for all simulated sets, Rotfit gives good estimate of the helix axis but OptLS gives a better one.

TABLE 3.2: Variance comparison of different sets of 1000 helices.

Method	set 1	set 2	set 3	set 4
Rotfit	9×10^{-5}	8×10^{-4}	3×10^{-4}	4×10^{-4}
OptLS	7×10^{-5}	6×10^{-4}	2×10^{-4}	2×10^{-4}

3.7 Comparison of different methods of estimating helix axis

Our particular interest is the performance of the various methods with regard to estimation of the helix axis. In this subsection, the proposed methodologies Difeigenfit and OptLS in subsections in Sections 3.5.1 and 3.5.3, respectively, are compared via simulations of regular statistical helices, with Parlsq (in Section 3.2), Eigenfit (in Section 3.3) and Rotfit (in Section 3.4).

To compare different methods we simulate 1000 helices of $n = 30$ landmarks as in Section 3.6 where $\sigma^2 = 0.05$. We study the estimate of r and c for a helix in

canonical form with parameters that mimic a protein α-helix. We expect that for Difeigenfit and OptLS, the estimates are close to the true α-helix parameters. In addition, we will compare the estimates of the helix axis $\hat{\boldsymbol{w}}$ by all methods with the axis in the canonical form $\boldsymbol{w} = [0, 0, 1]^T$.

Using the OptLS for 1000 simulated helices gives samples of $\hat{r}, \hat{c}$ and $1-\boldsymbol{w}^T\hat{\boldsymbol{w}}$. We draw histograms of these samples in Figure 3.4 and Figure 3.5 which show bell-shaped curves around the true values of r and c, respectively. These figures present that the distribution of r and c are around the α-helix parameters value where Difeigenfit has the variability ($\sigma_r^2 = 0.003$ and $\sigma_c^2 = 1.2 \times 10^{-5}$) greater than the OptLS ($\sigma_r^2 = 0.002$ and $\sigma_c^2 = 1.1 \times 10^{-5}$).

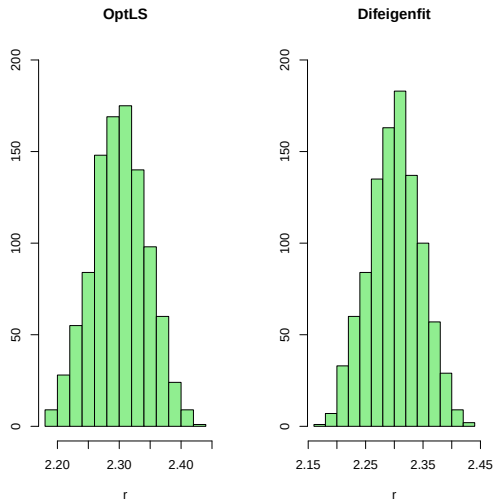

FIGURE 3.4: Histograms of simulated sample of estimates $\hat{r}$.

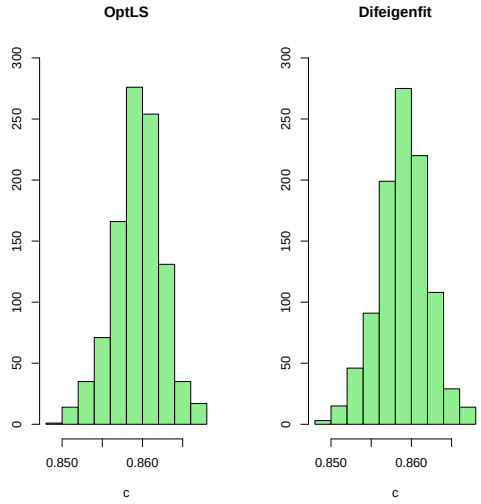

FIGURE 3.5: Histograms of simulated sample of estimates $\hat{c}$.

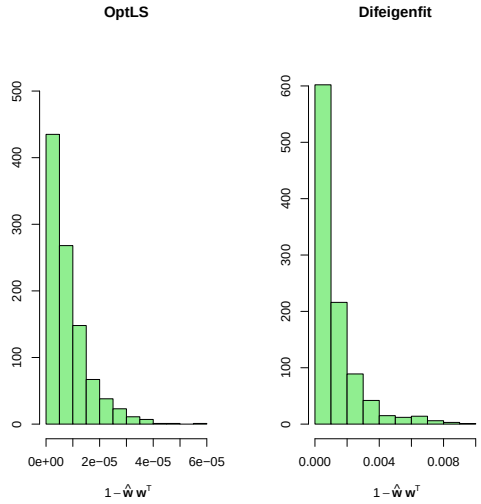

FIGURE 3.6: Histograms of simulated sample of estimates $\boldsymbol{w}^T\hat{\boldsymbol{w}}$.

The histograms of Difeigenfit and OptLS in Figures 3.6 present the distribution of the MSE $1-\hat{\hat{\boldsymbol{w}}}^T\boldsymbol{w}$ of the $q = 1000$ helices for each method. These figures show a skewed longer tail going off to the right, where Difeigenfit has greater variability than OptLS. In other words, the range of $1-\hat{\boldsymbol{w}}_j^T\boldsymbol{w}, j = 1,\ldots,q$ for Difeigenfit and OptLS are $[9\times10^{-8}, 9\times10^{-3}]$ and $[8\times10^{-9}, 5\times10^{-5}]$, respectively, where $\boldsymbol{w} = [0,0,1]^T$. That is, in both approach, all the simulated helices have variance very close to 0, hence our approach works well. Overall we conclude that OptLS is more accurate than Difeigenfit.

Next, we use the simulated data to find the helix axis by Parlsq, Eigenfit and Rotfit and compare the results with Difeigenfit and OptLS methods. The main characteristic that differentiates one method from another for finding the axis is the MSE of the axis, $(1-\hat{\hat{\boldsymbol{w}}}^T\boldsymbol{w})$ which is presented in Table 3.3. For 1000 simulated helices mimic protein α-helix with $n = 30$ landmarks and $\sigma^2 = 0.05$, as in Section 3.6, OptLS determines the best fit as seen in Figure 3.7, where the variation of estimate $\boldsymbol{w}$ by $\hat{\boldsymbol{w}}$ is baseline variance which is very small. The Rotfit axis variance is close to OptLS axis variance, whereas Parlsq and Eigenfit have are 10 times worse than OptLS. The Difeigenfit has the worst variance. Therefore, Rotfit is the second best method. In addition, the histograms of the distribution of $1-\hat{\boldsymbol{w}}_j^T\boldsymbol{w}$ for OptLS and Rotfit are skewed to the right of zero, which are placed in the top of Figure 3.7. The Parlsq and Eigenfit are approximately symmetric and slightly far from 0, so they determine bad estimates of $\boldsymbol{w}$. In addition to the first set of simulated data, we simulate 3 more sets of 1000 different shapes of helices: the first set of short fat helices where $n = 12, r = 7, c = 0.1$ and $\delta = \frac{2\pi}{3.6}$; the second set of long thin helices of $n = 30, r = 1, c = \frac{5.4}{2\pi}$ and $\delta = \frac{2\pi}{3.6}$; and the third set of helices mimic protein α-helix with $n = 20$. Table 3.3 also shows that Eigenfit cannot estimate the axis for short fat helix in set 2 where the variance is very large, as we discussed before in Section 3.6.1. Parlsq has also large variance of estimate $\boldsymbol{w}$ for set 2. Overall, OptLS and Rotfit have the smallest variance among other methods, where OptLS has the smallest variance.

TABLE 3.3: Variance comparison of different methods.

set	OptLS	Rotfit	Eigenfit	Parlsq	Difeigenfit
1	8.21×10^{-6}	9.13×10^{-6}	2.58×10^{-5}	2.88×10^{-5}	1.17×10^{-3}
2	2.05×10^{-4}	2.13×10^{-4}	0.987	0.285	5.86×10^{-4}
3	8.34×10^{-6}	9.31×10^{-6}	1.03×10^{-5}	1.23×10^{-5}	6.90×10^{-3}
4	3.37×10^{-5}	3.72×10^{-5}	6.86×10^{-5}	5.56×10^{-5}	1.75×10^{-3}

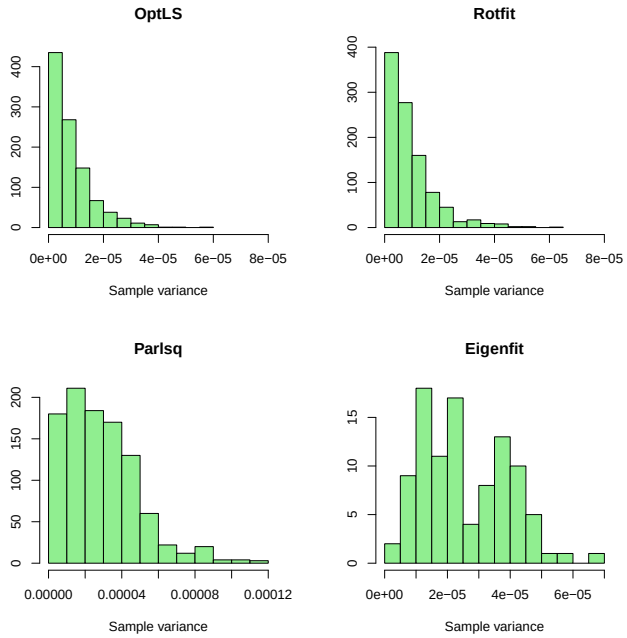

FIGURE 3.7: The histogram of the frequency of the MSE $1 - \boldsymbol{w}^T\hat{\boldsymbol{w}}$ of different methods.

3.8 Distribution of $1-\hat{\boldsymbol{w}}^T\boldsymbol{w}$ using the Difeigenfit method

Our aim in this section is to find the asymptotic distribution of $1 - \hat{\boldsymbol{w}}^T\boldsymbol{w}$ for small σ^2 (see Mardia et al., 1979, p. 230), where $\hat{\boldsymbol{w}}$ has been obtained by using Difeigenfit method. This presents how close the true axis vector is to the estimated one. In order to compute the distribution of $1 - \hat{\boldsymbol{w}}^T\boldsymbol{w}$, we use asymptotic distribution (see Harris, 2001) by using perturbation (see Kent et al., 1983; Kato, 2013), respectively. We first find the distribution of $\hat{\boldsymbol{w}}^T\boldsymbol{w}$ by the perturbation of the estimated helix axis $\hat{\boldsymbol{w}}$, then we deduce the distribution of $1 - \hat{\boldsymbol{w}}^T\boldsymbol{w}$ from this. If we have small errors in the original model, these errors percolate through almost linearly into small perturbations in $\hat{\boldsymbol{w}}$.

For a data helix H of size $n \times 3$, where $\boldsymbol{y}_i = [y_{i1}, y_{i2}, y_{i3}]^T$, recall the helix model (1.3)

$$\boldsymbol{y}_i = \boldsymbol{y}(t_i) = r\cos(t_i)\boldsymbol{u} + r\sin(t_i)\boldsymbol{v} + ct_i\boldsymbol{w} + \boldsymbol{b} + \boldsymbol{\varepsilon}_i,$$

where $t_i = (i-1)\delta, \delta = \frac{2\pi}{3.6}$, and $\boldsymbol{\varepsilon}_i = [\varepsilon_{i1}, \varepsilon_{i2}, \varepsilon_{i3}]^T$, where all ε_{ij} are assumed to be independent and $N(0, \sigma^2)$ for $i = 1, \ldots, n$ and $j = 1, 2, 3$. We expand the increments in (3.1) as follows

$$\begin{aligned}
\boldsymbol{d}_i &= \boldsymbol{y}_i - \frac{\boldsymbol{y}_{i+1} + \boldsymbol{y}_{i-1}}{2} \\
&= r(\cos t_i - \frac{1}{2}\cos(t_i + \delta) - \frac{1}{2}\cos(t_i - \delta))\boldsymbol{u} \\
&+ (\sin t_i - \frac{1}{2}\sin(t_i + \delta) - \frac{1}{2}\sin(t_i - \delta))\boldsymbol{v} \\
&+ 0\boldsymbol{w} + (\boldsymbol{\varepsilon}_i - \frac{1}{2}\boldsymbol{\varepsilon}_{i+1} - \frac{1}{2}\boldsymbol{\varepsilon}_{i-1}) \\
&= r(\cos t_i - \frac{1}{2}(\cos t_i \cos\delta - \sin t_i \sin\delta + \cos t_i \cos\delta + \sin t_i \sin\delta))\boldsymbol{u} \\
&+ r(\sin t_i - \frac{1}{2}(\sin t_i \cos\delta - \cos t_i \sin\delta + \sin t_i \cos\delta + \cos t_i \sin\delta))\boldsymbol{v} + \boldsymbol{e}_i \\
&= r((\cos t_i - \cos t_i \cos\delta)\boldsymbol{u} + r(\sin t_i - \sin t_i \cos\delta)\boldsymbol{v}) + \boldsymbol{e}_i \\
&= r(\cos t_i(1 - \cos\delta)\boldsymbol{u} + \sin t_i(1 - \cos\delta)\boldsymbol{v}) + \boldsymbol{e}_i.
\end{aligned}$$

Recall $\delta = \frac{2\pi}{3.6}$, then $1 - \cos\delta = 1.17$, so

$$\begin{aligned}
\boldsymbol{d}_i &= 1.17r(\cos t_i\boldsymbol{u} + \sin t_i\boldsymbol{v}) + \boldsymbol{e}_i \\
&= 1.17r \begin{bmatrix} \cos t_i \\ \sin t_i \\ 0 \end{bmatrix} + \begin{bmatrix} e_{i1} \\ e_{i2} \\ e_{i3} \end{bmatrix}.
\end{aligned}$$

Let $\boldsymbol{\mu_i} = 1.17r[\cos t_i, \sin t_i, 0]^T$ and $\boldsymbol{e}_i = \boldsymbol{\varepsilon}_i - \frac{\boldsymbol{\varepsilon}_{i-1} + \boldsymbol{\varepsilon}_{i+1}}{2}$, $i = 1, \ldots, n$. In consequence, we have that $\boldsymbol{e}_i \sim N_3(\boldsymbol{0}, \frac{3}{2}\sigma^2 I_3)$. Since $\boldsymbol{d}_i = \boldsymbol{y}_i - \frac{\boldsymbol{y}_{i+1} + \boldsymbol{y}_{i-1}}{2}$, the vectors $\boldsymbol{d}_i$ are dependent. We treat the $\boldsymbol{e}_i$ as independent to simplify the analysis below. Then the $\boldsymbol{d}_i$ are identically distributed, following the normal distribution with

mean vector $[\alpha_i, \beta_i, 0]^T$ and with variance-covariance matrix $\frac{3}{2}\sigma^2 I_3$, and we treat them as independent.

Recall that the eigen-decomposition of $E = D^T D$ gives that the first and second principal components are expected to be approximately equal and have larger variation ($\lambda_1 \approx \lambda_2$) than the third component. The third component is orthogonal to the first and second components and has the smallest variation (λ_3) since $\boldsymbol{d}_i$ are computed in such a way that the z-coordinate of each point are zero for a mathematical helix and a small value for a statistical helix (see Section 3.5.1). Then $E = \sum_{i=2}^{n-1} \boldsymbol{d}_i \boldsymbol{d}_i^T$ can be written as

$$\begin{aligned} E &= \sum_{i=2}^{n-1} \boldsymbol{d}_i \boldsymbol{d}_i^T \\ &= \sum_{i=2}^{n-1} \boldsymbol{\mu}_i^T \boldsymbol{\mu}_i + (\boldsymbol{\mu}_i^T \boldsymbol{e}_i + \boldsymbol{e}_i^T \boldsymbol{\mu}_i) + \boldsymbol{e}_i^T \boldsymbol{e}_i \\ &= A + B + C, \end{aligned}$$

where $A = \sum_{i=2}^{n-1} \boldsymbol{\mu}_i^T \boldsymbol{\mu}_i$, $B = \sum_{i=2}^{n-1} \boldsymbol{\mu}_i^T \boldsymbol{e}_i + \boldsymbol{e}_i^T \boldsymbol{\mu}_i$ of order $||\boldsymbol{e}_i||$, and $C = \sum_{i=2}^{n-1} \boldsymbol{e}_i^T \boldsymbol{e}_i$ of order $||\boldsymbol{e}_i^2||$, where A, B and C are symmetric. We can rewrite E as

$$\begin{aligned} E &= \sum_{i=2}^{n-1} \boldsymbol{d}_i \boldsymbol{d}_i^T \\ &= \sum_{i=2}^{n-1} (1.17)^2 \begin{bmatrix} r\cos t_i + e_{i1} \\ r\sin t_i + e_{i2} \\ e_{i3} \end{bmatrix} \begin{bmatrix} r\cos t_i + e_{i1} & r\sin t_i + e_{i2} & e_{i3} \end{bmatrix} \\ &= \sum_{i=2}^{n-1} (1.17)^2 \begin{bmatrix} (r\cos t_i + e_{i1})^2 & (r\cos t_i + e_{i1})(r\sin t_i + e_{i2}) & (r\cos t_i + e_{i1})e_{i3} \\ (r\cos t_i + e_{i1})(r\sin t_i + e_{i2}) & (r\sin t_i + e_{i2})^2 & (r\sin t_i + e_{i2})e_{i3} \\ (r\cos t_i + e_{i1})e_{i3} & r(\sin t_i + e_{i2})e_{i3} & e_{i3}^2 \end{bmatrix}, \end{aligned}$$

where

$$A = \sum_{i=2}^{n-1} (1.17r)^2 \begin{bmatrix} \cos^2 t_i & \cos t_i \sin t_i & 0 \\ \cos t_i \sin t_i & \sin^2 t_i & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$B = \sum_{i=2}^{n-1} 1.17r \begin{bmatrix} 2e_{i1} \cos t_i & e_{i2} \cos t_i + e_{i1} \sin t_i & e_{i3} \cos t_i \\ e_{i2} \cos t_i + e_{i1} \sin t_i & 2e_{i2} \sin t_i & e_{i3} \sin t_i \\ e_{i3} \cos t_i & e_{i3} \sin t_i & 0 \end{bmatrix},$$

$$C = \sum_{i=2}^{n-1} \begin{bmatrix} e_{i1}^2 & e_{i1}e_{i2} & e_{i1}e_{i3} \\ e_{i1}e_{i2} & e_{i2}^2 & e_{i2}e_{i3} \\ e_{i1}e_{i3} & e_{i2}e_{i3} & e_{i3}^2 \end{bmatrix}.$$

The points $(\cos t_i, \sin t_i)$ are "almost balanced" points in the sense that their centre of gravity is near the origin. There would be exact balance if it were the case that $\delta = \frac{2\pi}{k}$ for some $k \geq 2$ and n were a multiple of k; see Appendix A.2. Then A can be written approximately as

$$A \approx \frac{n(1.17r)^2}{2} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Therefore, eigenvalues of $A = \text{diag}(\lambda_j), j = 1, 2, 3$ are approximately $\lambda_1 = \lambda_2 = (1.17r)^2 \frac{n}{2}$ and $\lambda_3 = 0$. The *Moore-Penrose* generalized inverse of A , (see Magnus and Neudecker, 2003, pp. 172-175), is

$$T = -\frac{2}{n(1.17r)^2} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Then we can derive the perturbation of the unit vector $\hat{\boldsymbol{w}}$ by $\boldsymbol{w}^{(1)} = TB\boldsymbol{w}$ as

$$\boldsymbol{w}^{(1)} = -\sum_{i=2}^{n-1} \frac{2(1.17r)}{n(1.17r)^2} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 2e_{i1}\cos t_i & e_{i2}\cos t_i + e_{i1}\sin t_i & e_{i3}\cos t_i \\ e_{i2}\cos t_i + e_{i1}\sin t_i & 2e_{i2}\sin t_i & e_{i3}\sin t_i \\ e_{i3}\cos t_i & e_{i3}\sin t_i & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$
$$= -\frac{2}{1.17rn} \begin{bmatrix} \sum_{i=2}^{n-1} e_{i3}\cos t_i \\ \sum_{i=2}^{n-1} e_{i3}\sin t_i \\ 0 \end{bmatrix}.$$

Since the helix axis $\hat{\boldsymbol{w}}$ is a unit vector, the third component of $\hat{\boldsymbol{w}}$ can be deduced from $\boldsymbol{w}^{(1)}$ as

$$1 - ((-\frac{2}{1.17rn}\sum_{i=2}^{n-1} e_{i3}\cos t_i)^2 + (-\frac{2}{1.17rn}\sum_{i=2}^{n-1} e_{i3}\sin t_i)^2)$$
$$= 1 - \frac{4}{(1.17rn)^2}((\sum_{i=2}^{n-1} e_{i3}\cos t_i)^2 + (\sum_{i=2}^{n-1} e_{i3}\sin t_i)^2).$$

Then $1 - \hat{\boldsymbol{w}}^T\boldsymbol{w}$ is half the squared distance between $\boldsymbol{w}$ and $\hat{\boldsymbol{w}}$ which can be written as

$$\frac{1}{2}||\boldsymbol{w} - \hat{\boldsymbol{w}}||^2 = 1 - \hat{\boldsymbol{w}}^T\boldsymbol{w},$$
$$= \frac{4}{(1.17rn)^2}((\sum_{i=2}^{n-1} e_{i3}\cos t_i)^2 + (\sum_{i=2}^{n-1} e_{i3}\sin t_i)^2).$$

Recall $\frac{2}{1.17rn}\sum_{i=2}^{n-1} e_{i3}\cos t_i$ and $\frac{2}{1.17rn}\sum_{i=2}^{n-1} e_{i3}\sin t_i$ are approximately normal with mean 0 and approximate variance $\frac{6\sigma^2}{(1.17rn)^2}\sum_{i=2}^{n-1}\cos^2 t_i$ and they are approximate uncorrelated, where e_{i3} is assumed to follow $N(0, \frac{3}{2}\sigma^2)$. Then 1-$\boldsymbol{w}^T\hat{\boldsymbol{w}}$ follows $\frac{(1.17rn)^2}{6\sigma^2\sum_{i=2}^{n-1}\cos^2 t_i}\ \chi^2_2$. But this is not quite true for two reasons: first we assumed that $e_{i3}, i = 1, \ldots, n$ are independent; second we also assumed $\cos t_i$ and $\sin t_i$ are perpendicular to each other in equally spaced points around the helix. The second perturbation $\boldsymbol{w}^{(2)}$ is derived by the equation

$$\boldsymbol{w}^{(2)} = TB\boldsymbol{w}^{(1)} - \frac{1}{2}\{\boldsymbol{w}^{(1)T}\boldsymbol{w}^{(1)T}\}\boldsymbol{w} - \lambda^{(1)}T^2B\boldsymbol{w} + TC\boldsymbol{w},$$

and

$$
\begin{aligned}
&TB\boldsymbol{w}^{(1)} \\
&= \sum_{i=2}^{n-1} \frac{4(1.17r)}{n^2(1.17r)^3} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 2e_{i1}\cos t_i & e_{i2}\cos t_i + e_{i1}\sin t_i & e_{i3}\cos t_i \\ e_{i2}\cos t_i + e_{i1}\sin t_i & 2e_{i2}\sin t_i & e_{i3}\sin t_i \\ e_{i3}\cos t_i & e_{i3}\sin t_i & 0 \end{bmatrix} \begin{bmatrix} e_{i3}\cos t_i \\ e_{i3}\sin t_i \\ 0 \end{bmatrix} \\
&= \sum_{i=2}^{n-1} \frac{4}{(1.17rn)^2} \begin{bmatrix} 2e_{i1}\cos t_i & e_{i2}\cos t_i + e_{i1}\sin t_i & e_{i3}\cos t_i \\ e_{i2}\cos t_i + e_{i1}\sin t_i & 2e_{i2}\sin t_i & e_{i3}\sin t_i \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} e_{i3}\cos t_i \\ e_{i3}\sin t_i \\ 0 \end{bmatrix} \\
&= \frac{4}{(1.17rn)^2} \begin{bmatrix} 2\sum e_{i1}e_{i3}\cos^2 t_i + \sum e_{i2}e_{i3}\cos t_i \sin t_i + \sum e_{i1}e_{i3}\sin^2 t_i \\ \sum e_{i2}e_{i3}\cos^2 t_i + \sum e_{i1}e_{i3}\cos t_i \sin t_i + 2\sum e_{i2}e_{i3}\sin^2 t_i \\ 0 \end{bmatrix},
\end{aligned}
$$

and

$$
\begin{aligned}
\{\boldsymbol{w}^{(1)T}\boldsymbol{w}^{(1)}\}\boldsymbol{w} &= \frac{4}{(1.17rn)^2} \begin{bmatrix} \sum_{i=2}^{n-1} e_{i3}\cos t_i & \sum_{i=2}^{n-1} e_{i3}\sin t_i & 0 \end{bmatrix} \begin{bmatrix} \sum_{i=2}^{n-1} e_{i3}\cos t_i \\ \sum_{i=2}^{n-1} e_{i3}\sin t_i \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\
&= \frac{4}{(1.17rn)^2} \begin{bmatrix} 0 \\ 0 \\ (\sum_{i=2}^{n-1} e_{i3}\cos t_i)^2 + (\sum_{i=1}^{n} e_{i3}\sin t_i)^2 \end{bmatrix},
\end{aligned}
$$

where

$$
T^2 = \frac{4}{n^2(1.17r)^4} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix},
$$

and

$$T^2B\boldsymbol{w} = \sum_{i=2}^{n-1} \frac{4(1.17r)}{n^2(1.17r)^4} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 2e_{i1}\cos t_i & e_{i2}\cos t_i + e_{i1}\sin t_i & e_{i3}\cos t_i \\ e_{i2}\cos t_i + e_{i1}\sin t_i & 2e_{i2}\sin t_i & e_{i3}\sin t_i \\ e_{i3}\cos t_i & e_{i3\sin t_i} & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$= \sum_{i=2}^{n-1} \frac{4}{n^2(1.17r)^3} \begin{bmatrix} 2e_{i1}\cos t_i & e_{i2}\cos t_i + e_{i1}\sin t_i & e_{i3}\cos t_i \\ e_{i2}\cos t_i + e_{i1}\sin t_i & 2e_{i2}\sin t_i & e_{i3}\sin t_i \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$= \sum_{i=2}^{n-1} \frac{4}{n^2(1.17r)^3} \begin{bmatrix} e_{i3}\cos t_i \\ e_{i3}\sin t_i \\ 0 \end{bmatrix},$$

and

$$TC\boldsymbol{w} = -\sum_{i=2}^{n-1} \frac{2}{(1.17rn)^2} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} e_{i1}^2 & e_{i1}e_{i2} & e_{i1}e_{i3} \\ e_{i1}e_{i2} & e_{i2}^2 & e_{i2}e_{i3} \\ e_{i1}e_{i3} & e_{i2}e_{i3} & e_{i3}^2 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$= -\sum_{i=1}^{n} \frac{2}{(1.17rn)^2} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} e_{i1}e_{i3} \\ e_{i2}e_{i3} \\ e_{i3}^2 \end{bmatrix}$$

$$= -\sum_{i=1}^{n} \frac{2}{(1.17rn)^2} \begin{bmatrix} e_{i1}e_{i3} \\ e_{i2}e_{i3} \\ e_{i3}^2 \end{bmatrix}.$$

Then

$$\boldsymbol{w}^{(2)} = \frac{4}{(1.17rn)^2}\begin{bmatrix} 2\sum e_{i1}e_{i3}\cos^2 t_i + \sum e_{i2}e_{i3}\cos t_i \sin t_i + \sum e_{i1}e_{i3}\sin^2 t_i \\ \sum e_{i2}e_{i3}\cos^2 t_i + \sum e_{i1}e_{i3}\cos t_i \sin t_i + 2\sum e_{i2}e_{i3}\sin^2 t_i \\ 0 \end{bmatrix}$$
$$- \frac{2}{(1.17rn)^2}\begin{bmatrix} 0 \\ 0 \\ (\sum_{i=1}^{n} e_{i3}\cos t_i)^2 + (\sum_{i=1}^{n} e_{i3}\sin t_i)^2 \end{bmatrix}$$
$$- \sum_{i=2}^{n-1}\frac{2}{1.17rn}\begin{bmatrix} e_{i3}\cos t_i \\ e_{i3}\sin t_i \\ 0 \end{bmatrix} - \sum_{i=1}^{n}\frac{2}{(1.17r)^2}\begin{bmatrix} e_{i1}e_{i3} \\ e_{i2}e_{i3} \\ e_{i3}^2 \end{bmatrix}.$$

We know that the helix axis $\boldsymbol{w}$ perturbation as follows

$$\boldsymbol{w} = \boldsymbol{w}^{(0)} + \boldsymbol{w}^{(1)} + \boldsymbol{w}^{(2)},$$

then

$$\begin{aligned} 1 &= \boldsymbol{w}^T\boldsymbol{w} \\ &= [\boldsymbol{w}^{(0)} + \boldsymbol{w}^{(1)} + \boldsymbol{w}^{(2)}]^T[\boldsymbol{w}^{(0)} + \boldsymbol{w}^{(1)} + \boldsymbol{w}^{(2)}] \\ &= \boldsymbol{w}^{(0)T}\boldsymbol{w}^{(0)} + 2\boldsymbol{w}_3^{(1)} + 2\boldsymbol{w}_3^{(2)} + \boldsymbol{w}^{(1)T}\boldsymbol{w}^{(1)} \\ &= 1 + 0 + (2\boldsymbol{w}_3^{(2)} + \boldsymbol{w}_1^{(1)2} + \boldsymbol{w}_2^{(1)2}), \end{aligned}$$

So that the third component of $\boldsymbol{w}^{(2)}$ up to the first order perturbation is

$$w_3^{(2)} = -\frac{1}{2}(w_1^{(1)2} + w_1^{(2)2}),$$

which makes us sure that $\boldsymbol{w}^{(1)}$ is correct.

3.9 Fitting OptLS method to the data α-helix

For a real dataset of $n = 15$ points on α-helix from Mardia et al. (2018) (see Appendix B, helix 8 in Table B.8) which is presented in Figure 3.8, we apply OptLS to estimate the parameters.

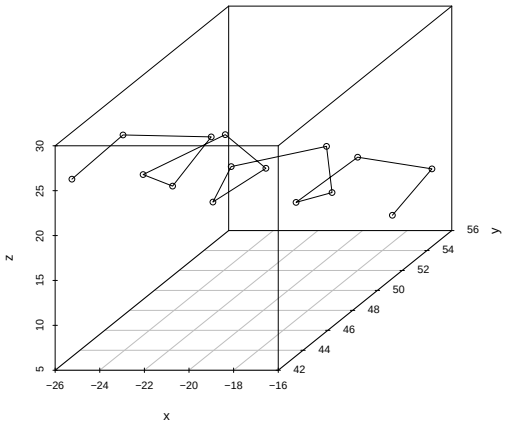

FIGURE 3.8: The data helix of 15 points.

The first rotation matrix $\hat{\Gamma}_1$ is

$$\hat{\Gamma}_1 = \begin{bmatrix} 0.212 & 0.924 & 0.318 \\ 0.762 & -0.360 & 0.537 \\ 0.612 & 0.129 & -0.780 \end{bmatrix},$$

where the third column of $\hat{\Gamma}_1$ is the helix axis $\hat{\boldsymbol{w}}$ corresponding to the smallest eigenvalue of E (see Section 3.1). Figure 3.9 presents the helix after the first rotation. Recall, to estimate the other helix parameters we put the helix into semi-canonical coordinates by rotating the $n \times 3$ data matrix H by this rotation matrix $\hat{\Gamma}_1$, $H^o = H\hat{\Gamma}_1$, which is explained in Section 3.5.2.

The least squares estimates of the shape parameters (radius r and pitch c) and the registration parameters (shift vector $\boldsymbol{b}$ and angle τ) are shown in the

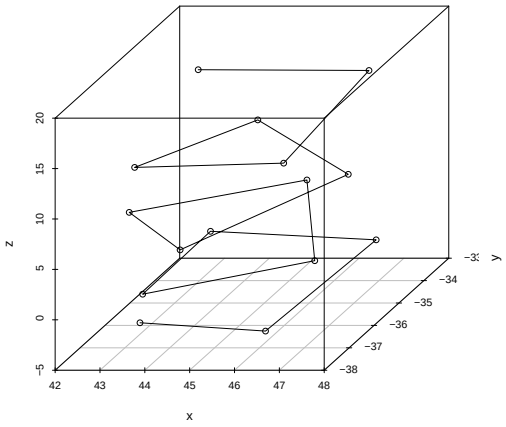

FIGURE 3.9: The data helix of 15 points after applying the first rotation.

following least squares fit equation

$$\hat{\boldsymbol{z}}(t_i) = 2.279\cos(t_i + 1.655)\begin{bmatrix}1\\0\\0\end{bmatrix} + 2.279\sin(t_i + 1.655)\begin{bmatrix}0\\1\\0\end{bmatrix} + 0.851t_i\begin{bmatrix}0\\0\\1\end{bmatrix} + \begin{bmatrix}44.848\\-35.384\\-6.416\end{bmatrix}.$$

We also need to note that the estimate $\hat{\tau}$ gives the rotation matrix $\hat{\Gamma}_2$, which rotates the helix about the z-axis,

$$\hat{\Gamma}_2 = \begin{bmatrix}-0.084 & -0.996 & 0\\ 0.996 & -0.084 & 0\\ 0 & 0 & 1\end{bmatrix}.$$

Then Figure 3.10 presents the data helix in canonical coordinates after shifting and rotating.

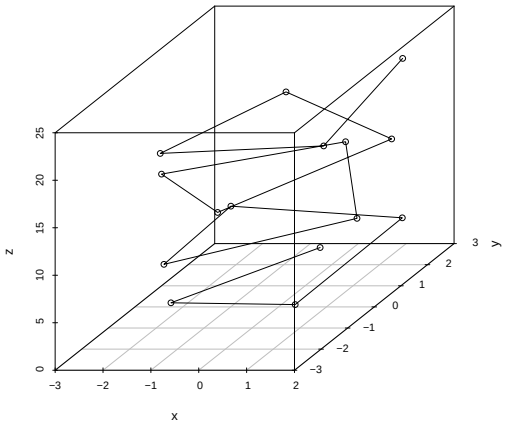

FIGURE 3.10: The data helix of 15 points in canonical coordinates.

The least squares procedure also provides the standard deviation of the residuals $\sigma = 0.246$, which measures how close the fitted helix is to the data helix. The residual sum of squares has a length measurement unit, angstroms $\mathring{A}^2$, and can be calculated as

$$\begin{aligned} SS_E &= (3n - p)\sigma^2 \\ &= 39 \cdot (0.061) = 2.379, \end{aligned}$$

where $p = 6$ is the number of parameters. The squared multiple correlation coefficient $R^2 = 0.996$ and the adjusted R^2 demonstrate that the model has good fit to real data. In other words, the variables explain 99.6% of the variability in the data. The multiple correlation coefficient $R = 0.998$ suggests that the correlation between the predicted and observed values is strong.

In general, we fit the helix very closely. Comparing the standard deviation $\sigma = 0.246$ $\mathring{A}$ to the radius $r = 2.3$ $\mathring{A}$ suggests that the fitted helix is a good fit, but not as highly as the computed R^2 claimed.

We are interested in estimating the relation of z'_{i1}, z'_{i2} and z'_{i3}, for all i, with the helix model (i.e. design matrix X'), so that we minimize the deviations

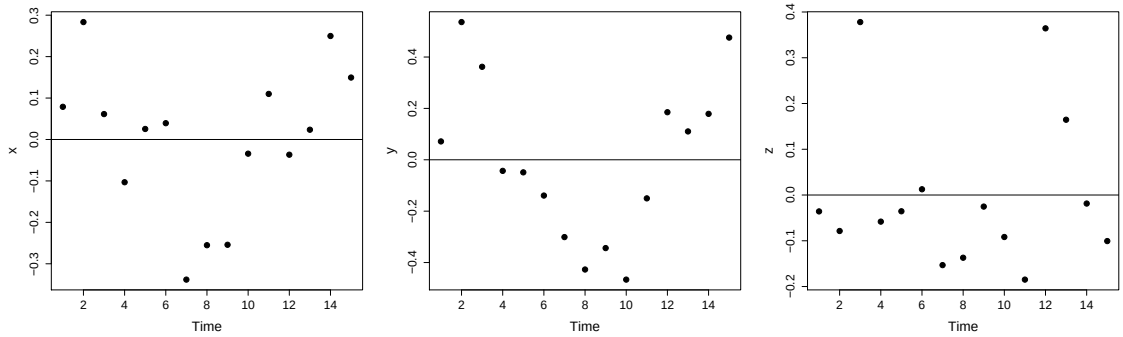

FIGURE 3.11: The data helix three coordinates residuals.

corresponding to the three coordinates. Figure 3.11 presents three residuals corresponding to the fitted values of the three coordinates: the left panel presents the residuals which are perpendicular to the helix axis ($e_{i1} = \sum_{i=n_1}^{n_2}(z'_{i1} - \hat{z}'_{i1})^2$), the middle panel presents the residuals which are along the helix axis ($e_{i3} = \sum_{i=n_1}^{n_2}(z'_{i3} - \hat{z}'_{i3})^2$), and the right panel presents the residuals which are perpendicular to helix axis ($e_{i2} = \sum_{i=n_1}^{n_2}(z'_{i2} - \hat{z}'_{i2})^2$), $n_1 = 1, n_2 = 15$. Figure 3.11 shows a quadratic behaviour (V-shape) between the residuals of x-coordinate and time, and between the residuals of y-coordinate and time, which is not captured by the OptLS method. The V-shape suggests that the axis could be bent. In order to investigate x and y coordinates residual plots to check for any indication of a bend, we test the null hypothesis H_0 : 'all of the quadratic coefficients are zero' i.e. the helix has no bend.

We fit two quadratic functions to each of the x and y residuals plots against time using least squares. Let ρ_i be the x residual quadratic function and γ_i the y residual quadratic function of the form

$$\rho_i = a_0 + a_1 t_i + a_2 t_i^2 + \varepsilon_i, \tag{3.10}$$

$$\gamma_i = b_0 + b_1 t_i + b_2 t_i^2 + \nu_i, \tag{3.11}$$

where ε_i and ν_i are the errors and $a_j, b_j \in \mathbb{R}$. Our null hypothesis is thus H_0 :

$b_2 = 0 = a_2$. The estimate of the quadratic coefficient from the quadratic fit of x residuals is $\hat{a}_2 = -0.059$ with small standard error of 9×10^{-4} and p-value of $3 \times 10^{-4} < 0.01$. The null hypothesis is rejected and we need to add this parameter to the model. The estimate of the quadratic coefficient from the quadratic fit of y residuals is $\hat{b}_2 = 0.005$ with small standard error of 6×10^{-4} and p-value of $4 \times 10^{-4} < 0.01$. The null hypothesis is rejected and we need to add this parameter to the model.

We choose to rotate the helices clockwise about the z-axis so that the axis of the sub-helix $H^{(2)}$, after the change point position, will lie on the x-axis. We will, therefore, expect changes in the x and y residuals plots such that one shows random noise, since the quadratic coefficient will become 0, and the other vividly displays a quadratic behavior. To this end, we find the angle ϕ between the $H^{(2)}$ axis and the positive x-coordinate. Hence, the 3D clockwise rotation matrix R is

$$R = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Since we rotate about the z-axis, no z-coordinate changes and it is sufficient to work with just the $x-$ and y-coordinates. Hence, R rotates the column vector $[\hat{a}_2, \hat{b}_2]^T$ clockwise to $[a_2^*, b_2^*]^T$ as follows:

$$\begin{bmatrix} a_2^* \\ b_2^* \end{bmatrix} = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \hat{a}_2 \\ \hat{b}_2 \end{bmatrix} \propto \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

So that $[a_2^*, b_2^*]^T$ is given by

$$a_2^* = \hat{a}_2 \cos\phi - \hat{b}_2 \sin\phi \propto 1,$$
$$b_2^* = \hat{a}_2 \sin\phi + \hat{b}_2 \cos\phi \propto 0,$$

and then the angle ϕ between the two vectors $[\hat{a}_2, \hat{b}_2]^T$ and $[1, 0]^T$ can be derived as

$$\hat{\phi} = \text{atan2}(\hat{b}_2, \hat{a}_2).$$

TABLE 3.4: The estimates of quadratic coefficients before and after rotation from the quadratic fit of the x and y residuals plots.

coordinates	$\hat{a}_2$	a_2^*	$\hat{b}_2$	b_2^*
x	-0.059	0.135	0.002	0.005
y	-0.121	-1×10^{-3}	0.005	0

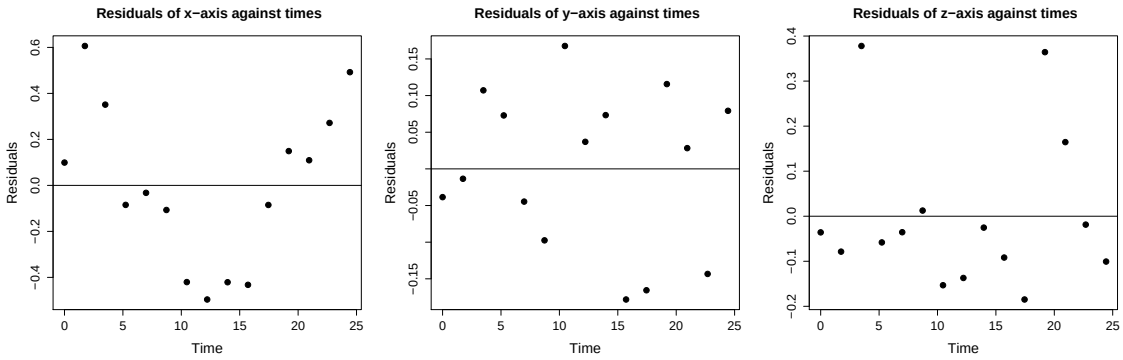

FIGURE 3.12: The data helix three coordinates residuals after rotation.

Figure 3.12 shows a quadratic behaviour (V-shape) in the x residual plot and the y residuals are randomly distributed. This result agrees with the data presented in Table 3.4. Panel (c) is not affected by the rotation since we rotate about the z-axis. This V-shape in the x residual plot indicates a bend where there is a need for further investigation (which we test by bootstrapping later in Chapter 5, see Section 5.5).

3.10 Cone helix

We fit a cone-helix of 36 landmarks using our estimation method of regular helix OptLS. The purpose of fitting a cone helix is to compare the residuals plot of a regular helix with the residuals plot of the cone helix. We can see below that the cone-helix residual plots present strong pattern.

Figure 3.13 displays four figures illustrating how the fitted points behave. The residual plots in panels (a), (b) and (c) presents each coordinate's residuals against time. Panels (a) and (b) show non-random patterns as the differences between the simulated and predicted points are very high and then decrease to close zero at the middle of the helix and then get higher again. These Panels confirm what we can see in the first panel of Figure 3.13 that the fitted points in red at the middle of the cone-helix are very close to the simulated points in black. Panel (c) shows a non-random pattern as the residuals spreads in a cone-shaped pattern.

3.11 Conclusion

We developed a method, OptLS, for fitting a regular helix. If we know $\boldsymbol{w}$, then we can use least squares method to estimate the parameters ($r, c, \boldsymbol{b}$, and τ) and calculate the residual sum of squares. If we do not know $\boldsymbol{w}$, then we let $\boldsymbol{w}$ vary and minimize the residual sum of squares, which we do numerically and need an initial axis to start. Several methods have been developed for estimating the helix axis which can be used to estimate the initial axis. Some of these methods are parametric least squares (Parlsq), eigenvector method (Eigenfit), and rotational least squares (Rotfit). We studied these methods and found that Parlsq and Eigenfit gave poor estimates of the axis for a fat short helix (tuna can), while Rotfit gave an accurate estimate. We also developed a new method based on modified principal components to estimate the initial helix axis for a

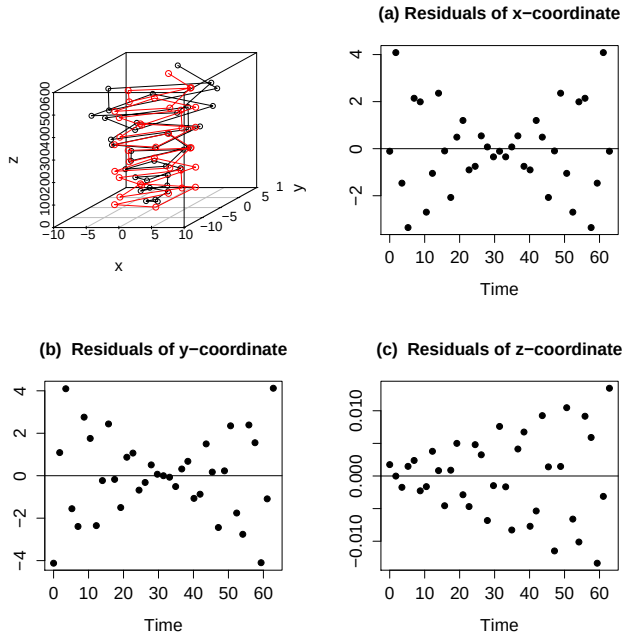

FIGURE 3.13: Four figures illustrate how the fitted of a cone helix of 36 landmarks: (a) Scatter plot presents the simulated helix in black and the fitted helix in red. Both (b), and (c), present the x- and y-coordinates residuals show bow tie shaped pattern. (d) The z- coordinates residuals clearly shows a cone-shaped pattern.

regular helix, called the difference eigenvector method (Difeigenfit). We did some simulations to study OptLS and Difeigenfit and found that the distributions of our estimates of r and c show bell-shaped curves around the real protein α-helix values. After that, we compared OptLS and Difeigenfit with Parlsq, Eigenfit, and Rotfit by simulation and found OptLS is the most accurate method among these methods and Rotfit comes after.

Finally, we fitted a real data helix by OptLS and found the estimated parameters r and c are close to the ideal values and the square of the coefficient of multiple correlation R^2 is close to 1, which indicates that the fitted helix is a good fit. In addition, $\hat{\sigma}^2 = 0.061$ is close to the theoretical $\sigma^2 = 0.065$ as empirically based estimate of a number of straight helices (unkinked as called by Mardia et al. (2018)). On the other hand, the x-coordinates residuals plot showed a V-shape pattern which indicates a bend that certainly required further investigation which we shall return to later in Chapter 5.

Further, we investigated fitting a simulated cone helix. The residual plots showed a non-random pattern in the three coordinates which suggests that OptLS did not fit the data well.

Chapter 4

Helix modelling through the Mardia-Holmes model framework

4.1 Introduction

A circle is a special case of a helix as we will describe later in this chapter. Recall the mathematical helix model (1.1), with $\boldsymbol{b} = \mathbf{0}$, is

$$\begin{aligned} y_1 &= r\cos(t), \\ y_2 &= r\sin(t), \\ y_3 &= ct. \end{aligned} \tag{4.1}$$

The first two coordinates y_1 and y_2 of the mathematical helix model (4.1) draw points around a circle, but the third coordinate y_3 varies as t varies, $0 \leq t \leq 2\pi$.

The projected helix (4.1) onto xy-plane is a circle

$$\begin{aligned} y_1 &= r\cos(t), \\ y_2 &= r\sin(t), \\ y_3 &= 0, \end{aligned} \tag{4.2}$$

where y_1 and y_2 are the coordinates form a circle with radius r. The statistical circle equation is exactly equation (4.2) but with added noise, as follows

$$\begin{aligned} y_{1i} &= r\cos(t_i) + \varepsilon_{1i}, \\ y_{2i} &= r\sin(t_i) + \varepsilon_{2i}. \end{aligned} \tag{4.3}$$

The errors ε_{1i} and ε_{2i} are assumed to be from independent normal distributions with mean 0 and variance σ^2. The starting point is $(r,\, 0)$ at $t = 0$, and the points move counter-clockwise around the circle if we look from above. A special case of this, is when $r = 1$, then the coordinates give a unit circle as t varies, with starting point $(1,\, 0)$.

In this chapter, we adapt the model by Mardia and Holmes (1980) for fitting circle and ellipse to data helix, to estimate the helix axis. It was originally designed for stone uniformly spread on circle or ellipse. This model has the mode set of circle or ellipse. The unknown parameters in the M-H model from Mardia and Holmes (1980) are the positive real κ (concentration), the vector $\boldsymbol{\alpha} = [a\ b]^T$ (location) and the 2×2 matrix Σ. If we have data points $\boldsymbol{z}_i = [y_{1i}\ y_{2i}]^T \in \mathbb{R}^2$, $i = 1, \ldots, n$, then the M-H model is

$$f(\boldsymbol{z}) = C(\kappa)|\Sigma|^{-1/2} \exp\{-\frac{1}{2}\kappa[(\boldsymbol{z}-\boldsymbol{\alpha})^T\Sigma^{-1}(\boldsymbol{z}-\boldsymbol{\alpha}) - 1]^2\}, \tag{4.4}$$

where the normalization constant is $C(\kappa) = \frac{(\kappa/2\pi)^{1/2}}{\pi\Phi(\kappa^{1/2})}$, and the ellipse, $(\boldsymbol{z} - \boldsymbol{\alpha})^T\Sigma^{-1}(\boldsymbol{z}-\boldsymbol{\alpha}) = 1$, is the mode.

We would like to fit a data set of n points $\boldsymbol{z}_i = [y_{1i}\ y_{2i}]^T, i = 0, \ldots, n$, in the

two-dimensional plane using Mardia-Holmes (M-H) model. If we could write the 2×2 matrix Σ as $\Sigma = \rho^2 I_2$, where $\rho^2 \in \mathbb{R}^+$ and I_2 is the identity matrix, then the model gives a circle with radius ρ, otherwise this model gives an ellipse. We will estimate the unknown parameters in five cases.

CASE 1 Unit-circle with one unknown parameter κ (concentration), where $\Sigma = I$ and $\boldsymbol{\alpha} = \mathbf{0}$.

CASE 2 Unit-circle with unknown parameters κ and $\boldsymbol{\alpha}$, where $\Sigma = I$.

CASE 3 Circle with unknown parameters κ, $\boldsymbol{\alpha}$ and radius ρ, where $\Sigma = \rho^2 I$.

CASE 4 Ellipse with Σ unknown but κ and $\boldsymbol{\alpha}$ are known.

CASE 5 Ellipse with all the parameters, κ, $\boldsymbol{\alpha}$ and Σ are unknown.

To illustrate we create a toy dataset of 10 points around the circle in Section 4.4.

In addition, we use the algorithm `nlm` (the unconstrained optimization algorithm routine in R, see R Core Team (2014)), which minimizes the negative log-likelihood function (equivalent to maximizing the log-likelihood function) to estimate the unknown parameters. The `nlm` algorithm is a straightforward algorithm since we can give the function without adding the derivative. In order to work faster in R we add the gradient to the algorithm.

Since we want to use an unconstrained optimizer such as `nlm`, it is necessary to parameterize the constrained parameters. The parameters κ and ρ are constrained by $\kappa > 0$, and $\rho > 0$, so we transform using the natural logarithm to the unconstrained parameters $\eta = \log(\kappa)$ and $\tau = \log(\rho)$. Then taking the exponential of these new parameters guarantees that the values of κ and ρ are positive.

In addition, we must be careful when choosing the starting values as a bad choice for the starting point of the location parameter can lead to a singularity of the likelihood. More specifically, the likelihood blows up to $+\infty$ as the initial

value for the center of the circle moves far away from the origin and the other parameters are suitably chosen. In particular, if $\boldsymbol{\alpha} = s\boldsymbol{\alpha}_0$ where $\boldsymbol{\alpha}_0$ is any unit two-dimentional vector, the negative log likelihood function (Section 4.2.3) satisfies

$$\begin{aligned}
-\log L(\kappa, \boldsymbol{\alpha}, \rho) &\simeq \frac{\kappa}{2} \sum_{i=1}^{n} \left[\frac{|\boldsymbol{z}_i - s\boldsymbol{\alpha}_0|^2}{\rho^2} - 1 \right]^2 - \frac{n}{2} \log \kappa \\
&= \frac{\kappa}{2} \sum_{i=1}^{n} \left[\frac{\boldsymbol{z}_i^T \boldsymbol{z}_i}{s^2} - \frac{2\boldsymbol{z}_i^T \boldsymbol{\alpha}_0}{s} \right]^2 - \frac{n}{2} \log \kappa \\
&= \frac{1}{2} \left[\sum_{i=1}^{n} \frac{(\boldsymbol{z}_i^T \boldsymbol{z}_i)^2}{s^2} - \frac{4 \sum_{i=1}^{n} (\boldsymbol{z}_i^T \boldsymbol{z}_i)(\boldsymbol{z}_i^T \boldsymbol{\alpha}_0)}{s} + 4 \sum_{i=1}^{n} (\boldsymbol{z}_i^T \boldsymbol{\alpha}_0)^2 \right] - \frac{n}{2} \log s^2 \\
&\to -\infty,
\end{aligned}$$

as $\kappa = \rho^2 = s^2 \to \infty$.

The M-H model can be used to estimate the helix axis; we shall give examples in Section 4.6. The mode is an ellipse or a circle. The M-H model can be used to describe the behaviour of the helix data after projecting on the plane perpendicular to the helix axis. The parameter ρ is matched to the helix radius r and the concentration parameter κ ia asymptotically the helix variance σ^2, see Section 4.5. Start by estimating the initial helix axis $\boldsymbol{w}$ using any helix axis estimation method as we discussed in Chapter 3. After that, we can project the helix data on the xy-plane so that data point are fall around circle. Then we can fit the M-H model to the data on the xy-plane and obtain residual sum of squares RSS($\boldsymbol{w}$) which is a function of the helix axis. The estimate of the helix axis is the value which minimises RSS($\boldsymbol{w}$), see Section 4.6. There are two problems, one is we need to estimate an initial helix axis in order to project the data and then find the axis. The other problem is that M-H method is not a stable method since the `nlm` optimizer in R needs initial values, and it could fail to find the accurate axis if the initial values are bad.

In the following two sections we estimate the unknown parameters in the

M-H model for the circle and the ellipse cases. We split the circle case into 3 cases and the ellipse into two cases.

4.2 Fitting a circle using the Mardia-Holmes model

The angular distribution $\theta_i = \text{atan2}(y_{2i}, y_{1i})$ of pair data points here are distributed uniformly around the circle, i.e. the distribution of the angle θ_i between the points is uniform with density of $\frac{1}{2\pi}$ within the interval $[0, 2\pi)$. Looking at the M-H model (4.4), for the circle case we have $\Sigma = \rho^2 I_2$ where $\rho^2 > 0$ is the radius of the circle. For estimation of the unknown parameters, we begin by assuming κ is unknown, then both κ and $\boldsymbol{\alpha}$ are unknown, and finally all κ, $\boldsymbol{\alpha}$ and ρ^2 are unknown. In the first two cases for estimating the parameters we assume $\rho = 1$, and in the very first case we translate the plane to assume $\boldsymbol{\alpha} = \mathbf{0}$.

4.2.1 One unknown parameter κ

In this subsection, we have a unit circle centred at the origin ($\rho = 1, \boldsymbol{\alpha} = \mathbf{0}$) with κ unknown. The aim here is to estimate κ using maximum likelihood estimation (MLE), see Mardia and Holmes (1980). For large κ the squared Mahalanobis distance $(\boldsymbol{z} - \boldsymbol{\alpha})^T \Sigma^{-1} (\boldsymbol{z} - \boldsymbol{\alpha})$ is asymptotically normally distributed (for more details see Section 4.5 when $\kappa = 100$), so that the MLE estimates are the same as the least squares estimates (LSE), (see e.g. Garthwaite et al. (2002), p. 61). Then under these assumptions, the M-H model (4.4) simply becomes

$$f(\boldsymbol{z}_i) = C(\kappa) \exp\{-\frac{\kappa}{2}(r_i^2 - 1)^2\}, \tag{4.5}$$

where $\boldsymbol{z}_i = [y_{1i}\ y_{2i}]^T$ and $r_i^2 = y_{1i}^2 + y_{2i}^2, r_i \geq 0,\ i = 1, \cdots, n$. The likelihood function is

$$L(\kappa \mid \boldsymbol{z}_i) = \prod_{i=1}^{n} f(\boldsymbol{z}_i \mid \kappa). \tag{4.6}$$

So the log-likelihood function is

$$\log L(\kappa \mid \boldsymbol{z}_i) = n \log C(\kappa) - \frac{\kappa}{2} \sum_{i=1}^{n} (r_i^2 - 1)^2.$$

The first derivative of the log-likelihood with respect to κ is

$$\frac{\partial \log L(\kappa \mid \boldsymbol{z}_i)}{\partial \kappa} = \frac{n}{2} \left(\frac{1}{\kappa} - \frac{1}{\sqrt{\kappa}} \frac{\phi(\sqrt{\kappa})}{\Phi(\sqrt{\kappa})} \right) - \frac{1}{2} \sum_{i=1}^{n} (r_i^2 - 1)^2, \tag{4.7}$$

where $\frac{1}{n} \sum_{i=1}^{n} (r_i^2 - 1)^2$ is the variance of r^2 about 1, does not depend on κ. The first derivative of the log of the normalization constant with respect to κ is positive, and this matches Figure 4.1 which shows that it is monotonically decreasing. Then $C(\kappa)$ is increasing as in Figure 4.2.

The second derivative of the log-likelihood with respect to κ in equation (4.8) is negative which shows that $C(\kappa)$ is concave. A concave function plus the linear function of the data is still a concave function (see Simon and Blume (1994), chapter 21), thus the log likelihood function is concave.

$$\frac{\partial^2 \log L(\kappa \mid \boldsymbol{z}_i)}{\partial \kappa^2} = \frac{-n}{2\kappa^2} + \frac{1}{2k^{3/2}} \frac{\phi(\sqrt{\kappa})}{\Phi(\sqrt{\kappa})} + \frac{n}{2\kappa} \frac{\frac{\sqrt{\kappa}}{\sqrt{2\pi}} e^{-\frac{k}{2}} \Phi(\sqrt{\kappa}) + \phi(\sqrt{\kappa})}{\Phi(\sqrt{\kappa})}. \tag{4.8}$$

It is difficult to compute an estimate of κ by hand, so we use numerical optimization. The likelihood takes the form of an exponential family therefore we can maximize the function by `nlm`, (e.g. Moller and Waagepetersen (2004) or Garthwaite et al. (2002)) . Since the function $\frac{\partial \log C(\kappa)}{\partial \kappa}$ is monotonically decreasing,

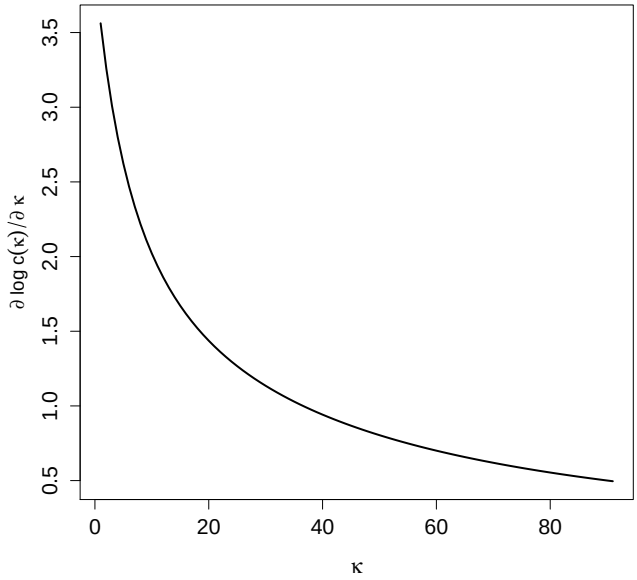

FIGURE 4.1: The plot of $\frac{\partial \log C(\kappa)}{\partial \kappa}$.

then we can search to the left of the function to get the positive values and to the right to get the negative value called bounds of κ, see Section 4.4, then the initial value in `nlm` of κ is chosen here the lower bound value of 10.

4.2.2 Unknown parameters κ and $\boldsymbol{\alpha}$

For the unit circle $\rho = 1$, $\Sigma = I$, κ and $\boldsymbol{\alpha} = [a \quad b]^T$ we substitute our assumptions in the M-H model (4.4), and this gives

$$\begin{aligned} f(\boldsymbol{z}_i) &= C(\kappa)|I|^{-1/2} \exp\{-\frac{\kappa}{2}[(\boldsymbol{z}_i - \boldsymbol{\alpha})^T I^{-1}(\boldsymbol{z}_i - \boldsymbol{\alpha}) - 1]^2\} \\ &= C(\kappa) \exp\{-\frac{\kappa}{2}[(\boldsymbol{z}_i - \boldsymbol{\alpha})^T (\boldsymbol{z}_i - \boldsymbol{\alpha}) - 1]^2\}, \end{aligned} \tag{4.9}$$

where $I^{-1} = I$, $|I|^{-1/2} = 1$, and $C(\kappa) = \frac{(\kappa/2\pi)^{1/2}}{\pi \Phi(\kappa^{1/2})}$. The likelihood function is

$$L(\kappa, \boldsymbol{\alpha} \mid \boldsymbol{z}_i) = \prod_{i=1}^{n} f(\boldsymbol{z}_i \mid \kappa, \boldsymbol{\alpha}),$$

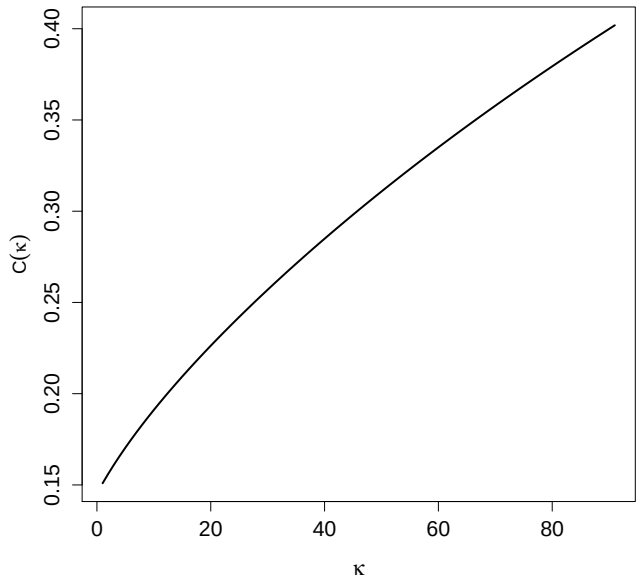

FIGURE 4.2: The plot shows the function $C(\kappa)$ is increasing.

while the log-likelihood function is

$$\log L(\kappa, \boldsymbol{\alpha} \mid \boldsymbol{z}_i) = n \log C(\kappa) - \frac{\kappa}{2} \sum_{i=1}^{n} [(\boldsymbol{z}_i - \boldsymbol{\alpha})^T (\boldsymbol{z}_i - \boldsymbol{\alpha}) - 1]^2.$$

We calculate the first derivatives of the log-likelihood with respect to $\boldsymbol{\alpha} = [a, b]^T$. We differentiate with respect to a by letting $v_i = (\boldsymbol{z}_i - \boldsymbol{\alpha})^T (\boldsymbol{z}_i - \boldsymbol{\alpha}) - 1 =$

$(y_{1i}-a)^2+(y_{2i}-b)^2-1$ and $m=-\frac{\kappa}{2}\sum_{i=1}^{n}v_i^2$, then

$$\begin{aligned}\frac{\partial \log L(\kappa,\boldsymbol{\alpha}\mid \boldsymbol{z}_i)}{\partial a} &= \frac{\partial}{\partial a}(-\frac{\kappa}{2}\sum_{i=1}^{n}[(\boldsymbol{z}_i-\boldsymbol{\alpha})^T(\boldsymbol{z}_i-\boldsymbol{\alpha})-1]^2)\\ &= \frac{\partial}{\partial a}(-\frac{\kappa}{2}\sum_{i=1}^{n}[(y_{1i}-a)^2+(y_{2i}-b)^2-1]^2)\\ &= \frac{\partial}{\partial a}(-\frac{\kappa}{2}\sum_{i=1}^{n}v_i^2)\\ &= \sum_{i=1}^{n}\frac{\partial m}{\partial v_i}\frac{\partial v_i}{\partial a}\\ &= (-\kappa\sum_{i=1}^{n}v_i)(-2(y_{1i}-a))\\ &= 2\kappa\sum_{i=1}^{n}v_iy_{1i}-2\kappa\sum_{i=1}^{n}v_ia. \end{aligned} \tag{4.10}$$

To estimate a we let the first derivative of the log-likelihood with respect to a in equation (4.10) be equal to zero, and then

$$\hat{a}=\frac{\sum_{i=1}^{n}v_iy_{1i}}{\sum_{i=1}^{n}v_i}.$$

The derivative with respect to b is

$$\begin{aligned}\frac{\partial \log L(\kappa,\boldsymbol{\alpha}\mid \boldsymbol{z}_i)}{\partial b} &= \frac{\partial}{\partial a}(-\frac{\kappa}{2}\sum_{i=1}^{n}v_i^2)\\ &= (-\kappa\sum_{i=1}^{n}v_i)(-2(y_{2i}-b))\\ &= 2\kappa\sum_{i=1}^{n}v_iy_{2i}-2\kappa\sum_{i=1}^{n}v_ib_i. \end{aligned} \tag{4.11}$$

To estimate b, let the first derivative of the log-likelihood with respect to b in equation (4.11) be equal to zero. We get

$$\hat{b}=\frac{\sum_{i=1}^{n}v_iy_{2i}}{\sum_{i=1}^{n}v_i},$$

where each v_i is dependent on a and b. We have to guess initial values of a and b to substitute into v_i. Another way to try to estimate a and b is to use the `polyroot` function in R. Since the gradient with respect to a in (4.10) is a polynomial in a for each b and the gradient with respect to b in (4.11) is a polynomial in b for each a. We start with an initial guess of a and update b, and iteratively cycle back and forth until the estimates converge. This is an example of alternating optimization procedure as each variable optimize give the other. In a limited simulation study it has been found to yield the same soluation as `nlm`, as expected. However, the method still requires an initial estimate of a or b, and so it does not avoid the problem of needing sensible initial estimates. Further, in general, alternating optimization procedures are typically slower than methods that treat a and b jointly, such as `nlm`.

The first derivative of the log-likelihood with respect to κ is

$$\frac{\partial \log L(\kappa, \boldsymbol{\alpha} \mid \boldsymbol{z}_i)}{\partial \kappa} = n\left(\frac{1}{2\kappa} - \frac{1}{2\sqrt{\kappa}}\frac{\phi(\sqrt{\kappa})}{\Phi(\sqrt{\kappa})}\right) - \frac{1}{2}\sum_{i=1}^{n} v_i^2. \tag{4.12}$$

We used the iterative procedure `nlm` in R to estimate all the parameters $\boldsymbol{\alpha} = [a, b]^T$ and κ with the same dataset in the previous case and we attribute `nlm` with the gradient vector to make the `nlm` work faster. The initial parameters values are chosen to be $\boldsymbol{\alpha} = [\bar{y}_1, \bar{y}_2]^T$, where $\bar{y}_1 = \frac{1}{n}\sum_{i=1}^{n} y_{1i}, \bar{y}_2 = \frac{1}{n}\sum_{i=1}^{n} y_{2i}$, and $\kappa = 10$.

4.2.3 Unknown parameters κ, $\boldsymbol{\alpha}$ and ρ

In this case we estimate the unknown parameters κ, $\boldsymbol{\alpha}$, and ρ, where we have a circle of radius ρ centred at $\boldsymbol{\alpha} = [a\ b]^T$ with concentration κ. Thus, we have $\Sigma = \rho^2 I$ and the M-H model is

$$f(\boldsymbol{z}_i) = C(\kappa)|\rho^2 I|^{-1/2} \exp\{-\frac{\kappa}{2}[(\boldsymbol{z}_i - \boldsymbol{\alpha})^T \rho^{-2} I^{-1} (\boldsymbol{z}_i - \boldsymbol{\alpha}) - 1]^2\},$$

where $I^{-1} = I$, $|I|^{-1/2} = 1$, and $C(\kappa) = \frac{(\kappa/2\pi)^{1/2}}{\pi\Phi(\kappa^{1/2})}$. The likelihood function is

$$L(\kappa, \boldsymbol{\alpha}, \rho \mid \boldsymbol{z}_i) = \prod_{i=1}^{n} f(\boldsymbol{z}_i \mid \kappa, \boldsymbol{\alpha}, \rho),$$

and the log-likelihood function is

$$\begin{aligned}
&\log L(\kappa, \boldsymbol{\alpha}, \rho \mid \boldsymbol{z}_i) \\
&= n \log C(\kappa) - \frac{n}{2} \log(|\rho^2 I|) - \frac{\kappa}{2} \sum_{i=1}^{n} [(\boldsymbol{z}_i - \boldsymbol{\alpha})^T \rho^{-2} I (\boldsymbol{z}_i - \boldsymbol{\alpha}) - 1]^2 \\
&= n \log C(\kappa) - \frac{n}{2} \log(\rho^4) - \frac{\kappa}{2} \sum_{i=1}^{n} [\frac{1}{\rho^2}(y_{1i} - a)^2 + \frac{1}{\rho^2}(y_{2i} - b)^2 - 1]^2,
\end{aligned}$$

where $n \log C(\kappa) = \frac{n}{2}(\log(\kappa) - \log(2\pi)) - n(\log(\pi) + \log \Phi(\sqrt{\kappa}))$. The first derivatives of the log-likelihood with respect to κ, a and b are the same as in the unit-circle except v is different, since $\Sigma = \rho^2 I$. We denote the new v by w, where $w_i = \frac{1}{\rho^2}(y_{1i} - a)^2 + \frac{1}{\rho^2}(y_{2i} - b)^2 - 1$, so we have the following first derivatives

$$\frac{\partial \log L(\kappa, \boldsymbol{\alpha}, \rho \mid \boldsymbol{z}_i)}{\partial \kappa} = n(\frac{1}{2\kappa} - \frac{1}{2\sqrt{\kappa}} \frac{\phi(\sqrt{\kappa})}{\Phi(\sqrt{\kappa})}) - \frac{1}{2} \sum_{i=1}^{n} w_i^2,$$

$$\frac{\partial \log L(\kappa, \boldsymbol{\alpha}, \rho \mid \boldsymbol{z}_i)}{\partial a} = \frac{2\kappa}{\rho^2} \sum_{i=1}^{n} w_i (y_{1i} - a),$$

and

$$\frac{\partial \log L(\kappa, \boldsymbol{\alpha}, \rho \mid \boldsymbol{z}_i)}{\partial b} = \frac{2\kappa}{\rho^2} \sum_{i=1}^{n} w_i (y_{2i} - b).$$

The first derivative of the log-likelihood with respect to ρ is

$$\begin{aligned}\frac{\partial \log L(\kappa, \boldsymbol{\alpha}, \rho \mid \boldsymbol{z}_i)}{\partial \rho} &= \frac{\partial}{\partial \rho}(-\frac{n}{2}\log(\rho^4) - \frac{\kappa}{2}\sum_{i=1}^{n} w_i^2) \\ &= -\frac{4n\rho^3}{2\rho^4} + \frac{2\kappa}{\rho^3}\sum_{i=1}^{n} w_i((y_{1i}-a)^2 + (y_{2i}-b)^2) \\ &= -\frac{2n}{\rho} + \frac{2\kappa}{\rho^3}\sum_{i=1}^{n} w_i((y_{1i}-a)^2 + (y_{2i}-b)^2).\end{aligned}$$

then

$$\hat{\rho} = \sqrt{\frac{2\kappa \sum_{i=1}^{n} w_i((y_{1i}-a)^2 + (y_{2i}-b)^2)}{2n}}$$

As in previous case, we used iterative procedure `nlm` in R adding the gradient vector to estimate all the parameters $\boldsymbol{\alpha}, \kappa$ and ρ. The initial values are $\boldsymbol{\alpha} = [\bar{y}_1, \bar{y}_2]^T$, $\rho = 1$, and $\kappa = 10$.

4.3 Fitting an ellipse using the Mardia-Holmes model

4.3.1 The matrix Σ is unknown

The aim of this section is to fit an ellipse with known location, assume $(0, 0)$, using the M-H model, we assume the concentration parameter $\kappa = 10$, since in practice the lower boundary of κ is 1 and the upper bound of 64. (see Section 4.4). Then the function (4.4) can be written as

$$f(\boldsymbol{z}_i) = \frac{(2.303/2\pi)^{\frac{1}{2}}}{(\pi\Phi(\sqrt{2.303}))}|\Sigma|^{-1/2}\exp\{-\frac{2.303}{2}[\boldsymbol{z}_i^T\Sigma^{-1}\boldsymbol{z}_i - 1]^2\}.$$

The unknown symmetric positive-definite matrix Σ can be decomposed by the Cholesky decomposition (see Section 2.3) into a unique product of a lower unit

triangular matrix, a diagonal matrix and a transpose of the lower unit triangular matrix, which is presented as follows:

$$\begin{aligned}\Sigma &= LGL^T, \\ &= \begin{pmatrix} 1 & 0 \\ l_{12} & 1 \end{pmatrix} \begin{pmatrix} \exp(g_1) & 0 \\ 0 & \exp(g_2) \end{pmatrix} \begin{pmatrix} 1 & l_{12} \\ 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} \exp(g_1) & \exp(g_1)l_{12} \\ \exp(g_1)l_{12} & \exp(g_1)l_{12}^2 + \exp(g_2) \end{pmatrix}. \end{aligned} \tag{4.13}$$

The matrix Σ contains three unknown parameters, namely $l_{12}, \exp(g_1)$ and $\exp(g_2)$. The parameters l_{12}, g_1 and g_2 are unconstrained. The resulting matrix Σ is then positive-definite.

Before writing down the log-likelihood function, we need to calculate $\det(\Sigma)$, Σ^{-1} and $\boldsymbol{z}^T\Sigma^{-1}\boldsymbol{z}$ as follows. First,

$$\begin{aligned}\det(\Sigma) &= \exp(g_1) \times (\exp(g_1)l_{12}^2 + \exp(g_2)) - \exp(g_1)^2 l_{12}^2 \\ &= \exp(g_1)^2 l_{12}^2 + \exp(g_1)\exp(g_2) - \exp(g_1)^2 l_{12}^2 \\ &= \exp(g_1)\exp(g_2).\end{aligned}$$

From that and equation (4.13) we have

$$\Sigma^{-1} = \frac{1}{\exp(g_1)\exp(g_2)} \begin{pmatrix} \exp(g_1)l_{12}^2 + \exp(g_2) & -\exp(g_1)l_{12} \\ -\exp(g_1)l_{12} & \exp(g_1) \end{pmatrix}$$

since $\exp(g_1)\exp(g_2) > 0$. Finally,

$$\begin{aligned}
z^T\Sigma^{-1}z &= \begin{pmatrix} x & y \end{pmatrix} \begin{pmatrix} \frac{l_{12}^2}{\exp(g_2)} + \frac{1}{\exp(g_1)} & -\frac{l_{12}}{\exp(g_2)} \\ -\frac{l_{12}}{\exp(g_2)} & \frac{1}{\exp(g_2)} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \\
&= \begin{pmatrix} x(\frac{l_{12}^2}{\exp(g_2)} + \frac{1}{\exp(g_1)}) - \frac{yl_{12}}{\exp(g_2)} & -\frac{xl_{12}}{\exp(g_2)} + \frac{y}{\exp(g_2)} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \\
&= \frac{x^2l_{12}^2}{\exp(g_2)} + \frac{x^2}{\exp(g_1)} - \frac{xyl_{12}}{\exp(g_2)} - \frac{xyl_{12}}{\exp(g_2)} + \frac{y^2}{\exp(g_2)} \\
&= \frac{x^2l_{12}^2}{\exp(g_2)} + \frac{x^2}{\exp(g_1)} - \frac{2xyl_{12}}{\exp(g_2)} + \frac{y^2}{\exp(g_2)}.
\end{aligned}$$

Letting $v = z^T\Sigma^{-1}z - 1$, the log-likelihood function is

$$\log L(\Sigma|\boldsymbol{z}_i) = \frac{n}{2}\log(\frac{2.303}{2\pi}) - n\log(\pi\Phi(\sqrt{2.303})) - \frac{n}{2}(g_1+g_2) - \frac{2.303}{2}\sum_{i=1}^{n} v_i^2.$$

Now we find the analytical solution to our log-likelihood function. To do this we start by taking the first derivative of the log-likelihood function with respect to g_1

$$\begin{aligned}
\frac{\partial \log L(\Sigma|\boldsymbol{z}_i)}{\partial g_1} &= \frac{\partial \log}{\partial g_1}(-\frac{n}{2}(g_1+g_2) - \frac{2.303}{2}\sum_{i=1}^{n} v_i^2) \\
&= -\frac{n}{2} + 2.303\sum_{i=1}^{n} v_i \frac{y_{1i}^2}{\exp(g_1)}.
\end{aligned}$$

The first derivative of the log-likelihood with respect to g_2

$$\frac{\partial \log L(\Sigma|\boldsymbol{z}_i)}{\partial g_2} = -\frac{n}{2} + 2.303\sum_{i=1}^{n} v_i \left(\frac{y_{1i}^2 l_{12}^2}{\exp(g_2)} - \frac{2y_{1i}y_{2i}l_{12}}{\exp(g_2)} + \frac{y_{2i}^2}{\exp(g_2)}\right).$$

Finally, the first derivative of the log-likelihood with respect to l_{12}

$$\frac{\partial \log L(\Sigma|\boldsymbol{z}_i)}{\partial l_{12}} = -2.303\sum_{i=1}^{n} v_i \left(\frac{2y_{1i}^2 l_{12}}{\exp(g_2)} - \frac{2y_{1i}y_{2i}}{\exp(g_2)}\right).$$

We used the iterative procedure `nlm` in R to estimate all the parameters l_{12}, g_1 and g_2 as in the previous case and we attribute `nlm` with the gradient vector to make the `nlm` work faster. The initial values of these parameters are $l_{12} = 0$, and $g_1 = \log(1) = g_2$, so that $\exp(\log(1)) = 1$.

4.3.2 The parameters $\boldsymbol{\alpha}$, κ and the matrix Σ are unknown

We want to estimate the unknown parameters κ, $\boldsymbol{\alpha}$ and Σ to fit an ellipse. As in the previous case we decompose the matrix Σ using the Cholesky decomposition. Therefore, we need to estimate six parameters, which are κ, a, b, g_1, g_2, and ℓ_{12}. Recall the M-H model in (4.4)

$$f(\boldsymbol{z}_i) = \frac{(\kappa/2\pi)^{\frac{1}{2}}}{\pi\Phi((\kappa)^{\frac{1}{2}})}|\Sigma|^{-1/2}\exp\{-\frac{\kappa}{2}[(\boldsymbol{z}_i - \boldsymbol{\alpha})^T\Sigma^{-1}(\boldsymbol{z}_i - \boldsymbol{\alpha}) - 1]^2\},$$

and,

$$\begin{aligned}
(\boldsymbol{z} - \boldsymbol{\alpha})^T\Sigma^{-1}(\boldsymbol{z} - \boldsymbol{\alpha}) &= \begin{pmatrix} y_1 - a & y_2 - b \end{pmatrix} \begin{bmatrix} \frac{l_{12}^2}{\exp(g_2)} + \frac{1}{\exp(g_1)} & -\frac{l_{12}}{\exp(g_2)} \\ -\frac{l_{12}}{\exp(g_2)} & \frac{1}{\exp(g_2)} \end{bmatrix} \begin{bmatrix} y_1 - a \\ y_2 - b \end{bmatrix} \\
&= \begin{bmatrix} (y_1 - a)(\frac{l_{12}^2}{\exp(g_2)} + \frac{1}{\exp(g_1)}) - \frac{(y_2-b)l_{12}}{\exp(g_2)} & -\frac{(y_1-a)l_{12}}{\exp(g_2)} + \frac{y_2-b}{\exp(g_2)} \end{bmatrix} \begin{bmatrix} y_1 - a \\ y_2 - b \end{bmatrix} \\
&= \frac{(y_1 - a)^2 l_{12}^2}{\exp(g_2)} + \frac{(y_1 - a)^2}{\exp(g_1)} - \frac{(y_1 - a)(y_2 - b)l_{12}}{\exp(g_2)} - \frac{(y_1 - a)(y_2 - b)l_{12}}{\exp(g_2)} + \frac{(y - b)^2}{\exp(g_2)} \\
&= \frac{(y_1 - a)^2 l_{12}^2}{\exp(g_2)} + \frac{(y_1 - a)^2}{\exp(g_1)} - \frac{2(y_1 - a)(y_2 - b)l_{12}}{\exp(g_2)} + \frac{(y_2 - b)^2}{\exp(g_2)}.
\end{aligned}$$

Let $v_i = (z_i - \alpha)^T\Sigma^{-1}(z_i - \alpha) - 1$, then the log-likelihood of the M-H model is

$$\begin{aligned}
\log L(\kappa, \boldsymbol{\alpha}, \Sigma|\boldsymbol{z}_i) =& \frac{n}{2}\log(\kappa) - \frac{n}{2}\log(2\pi) - n\log(\pi) - n\log(\Phi(\sqrt{\kappa})) \\
&- \frac{n}{2}\log(\exp(g_1)\exp(g_2)) - \frac{\kappa}{2}\sum_{i=1}^{n} v_i^2.
\end{aligned}$$

As in the previous case we used `nlm` in R to estimate all the parameters $a, b, \kappa, l_{12}, g_1$ and g_2. The initial values of these parameters are $a = \bar{y}_1, b = \bar{y}_2, l_{12} = 0, g_1 = \log(1) = g_2$ and $\kappa = 10$.

4.4 Applications

We want to estimate the unknown parameters of the M-H model, which are the concentration κ, the centre vector $\boldsymbol{\alpha}$, and the ellipse major and minor axes that are obtained from Σ. We started from the simplest case, where we had a unit circle centred at the origin (since circle case is a special case of the ellipse), with a single unknown parameter, then worked our way up to the general case of the ellipse with all parameters κ, $\boldsymbol{\alpha}$ and Σ are unknown. We estimated the unknown parameters in each of these cases using the `nlm` package in R. We give to `nlm` the negative log likelihood using the data and an initial guess of the unknown parameters. We can give the `nlm` function for the gradients by hand, to make the program run faster.

We create a toy example, to illustrate these five cases, of $n = 10$ points around a unit circle using equation (4.3) by letting $r = 1, t_i = \frac{2\pi}{360}37i$, $i = 0, \cdots, 9$ and e_i be simulated from a normal distribution $N(0, \sigma^2 = 0.1)$. Figure 4.3 presents a toy dataset of 10 points around a circle which is also presented in tabular form in Table 4.1.

For a unit circle centred at the origin as discussed in Section 4.2.1, we use the `nlm` function to estimate κ which require an initial value, so that we want to find the lower and the upper bounds for κ. To find these bounds for κ, we develop an R program using a "while loop" using this first derivative (4.7) and try to reach conditions. Since $\frac{\partial \log C(\kappa)}{\partial \kappa} > 0$, we note that $C(\kappa)$ is increasing and so we can find

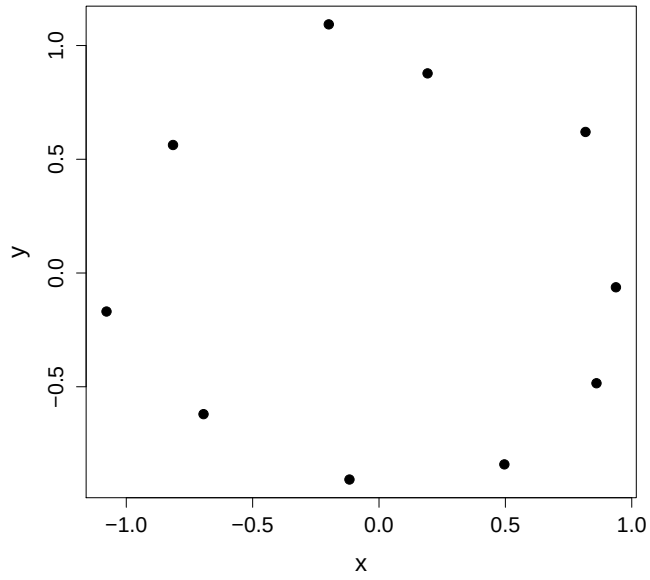

FIGURE 4.3: The 10 point artificial data set.

TABLE 4.1: The 10 point data set.

x	y
0.937	-0.063
0.817	0.620
0.192	0.878
-0.199	1.093
-0.815	0.563
-1.078	-0.169
-0.694	-0.620
-0.117	-0.908
0.496	-0.841
0.860	-0.485

unique crossing bounds of the horizontal axis. Starting with an arbitrary $\kappa = 1$ and keep moving to the left by a factor of 2 until reach the condition $\frac{\partial \log C(\kappa)}{\partial \kappa} < 0$ and then keep moving to the right until reach the condition $\frac{\partial log C(\kappa)}{\partial \kappa} > 0$. The output gave us a lower bound of 1, and an upper bound of 64. Then the `nlm` function gave us an estimate $\hat{\kappa} = 3.66$

For unit circle with unknown parameters κ and $\alpha = [a, b]^T$ in Section 4.2.2,

we used `nlm` to estimate a, b and κ, where the starting points of the centre α is chosen to be the means of y_{1i} and y_{2i} and $\kappa = 10$, within the boundary of κ. We have tried many starting points and all of them gave us the same result. Next, for a circle with unknown parameters κ, $\boldsymbol{\alpha} = [a, b]^T$ and ρ in Section 4.2.3, as in previous case we used `nlm` to estimate the unknown parameters. The initial point of the centre as in previous case is chosen to be the means of x_i and y_i, $\kappa = 10$, and using protein knowledge that radius is 2.3 we choose $\rho = 1$, which is a bit different to check the behaviour of the algorithm. Both $(\kappa), \log(\rho)$ and κ, ρ can be estimated. After the estimation of these parameters we take the exponential function of the estimates.

For the ellipse cases, case 1: Σ is unknown and the other parameters are known, which has been studied in Section 4.3.1. We assume $\boldsymbol{\alpha} = [0\ 0]^T$ and $\kappa = 10$ to estimate the parameters $l_{12}, \exp(g_1)$ and $\exp(g_2)$ of Σ in equation (4.13), as before we use the `nlm` to estimate these parameters. The initial values of these parameters are chosen using the protein knowledge (circular helix) to be $l_{12} = 0$ and $g_1 = g_2 = \log(1)$. In the second case the parameters $\Sigma, \boldsymbol{\alpha}$ and κ are unknown, see Section 4.3.2. We use the `nlm` to estimate these parameters. The starting values of these parameters are as the previous ones, the centre is the means of y_{1i} and y_{2i}, $\kappa = \exp(\log(10))$, $l_{12} = 0$ and $g_1 = g_2 = \log(1)$. All the results are summarized in Table 4.2.

In particular, $\hat{\kappa} = 3.66$ in case 1 where the model does not fit well and $\hat{\kappa} = 72.17$, a much higher value, in case 5, which fits data much more closely. This improvement was expected, since we estimate more parameters by fitting an ellipse, i.e. model fit the data better, and hence gives us a better result. The estimates of the centre in cases 3 and 5 are equal.

TABLE 4.2: Parameters estimates of the M-H model under the circle and the ellipse cases.

Case	**Unknown parameters**	**Known parameters**	**Estimates of the parameters**
1	κ	$\Sigma = I$ $\boldsymbol{\alpha} = [0\ 0]^T$	$\hat{\kappa} = 3.66$
2	κ $\boldsymbol{\alpha}$	$\Sigma = I$	$\hat{\kappa} = 4.56$ $\hat{\boldsymbol{\alpha}} = [-2.02\ \ -1.09]^T$
3	κ $\boldsymbol{\alpha}$ $\Sigma = \rho I$	-	$\hat{\kappa} = 65.63$ $\hat{\boldsymbol{\alpha}} = [-0.03\ 0.04]^T$ $\hat{\rho} = 0.99$
4	Σ (g_1, g_2, ℓ_{12})	$\kappa = 15.413$ $\boldsymbol{\alpha} = [0\ 0]^T$	$\hat{\Sigma} = \begin{bmatrix} 0.66 & -0.01 \\ -0.01 & 0.87 \end{bmatrix}$
5	κ $\boldsymbol{\alpha}$ Σ (g_1, g_2, ℓ_{12})	-	$\hat{\kappa} = 72.17$ $\hat{\boldsymbol{\alpha}} = [-0.03\ 0.04]^T$ $\hat{\Sigma} = \begin{bmatrix} 1.02 & -0.04 \\ -0.04 & 0.95 \end{bmatrix}$

Our initial ellipse is a unit circle centred at $(\bar{x}, \bar{y})$ and $\kappa = 10$, shown in blue in Figure 4.4. The fitted ellipse is presented in green in Figure 4.4, where concentration $\kappa = 72.17$, centre (-0.03,-0.04), and

$$\Sigma = \begin{pmatrix} 1.02 & -0.04 \\ -0.04 & 0.95 \end{pmatrix},$$

The `nlm` algorithm needs initial values for the parameters. The choice is important because for a bad starting point the `nlm` will not converge. For example,

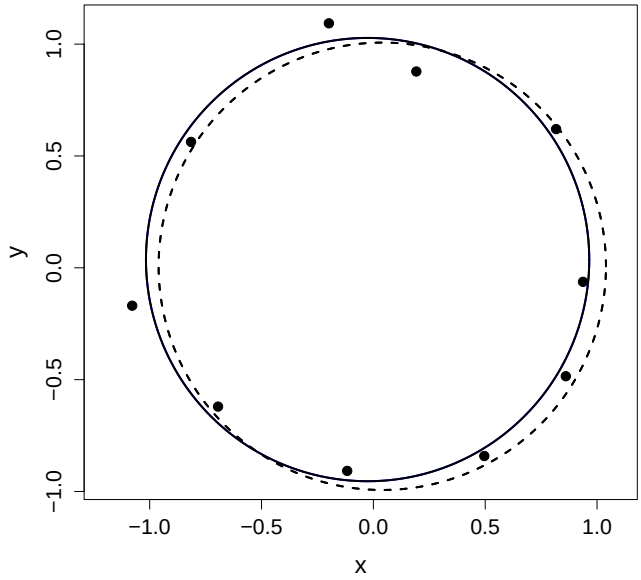

FIGURE 4.4: Plot of the fitted ellipse in solid line and initial started ellipse in dashed line.

for the dataset (10 points around a circle) in Table 4.1 a suitable choice of initial values of the centre is the mean of y_{1i} and the mean of y_{2i} since the points lie on a circle.

4.5 Asymptotic behaviour

Our goal in this section is to prove that the distribution specified by M-H model is asymptotically normal in distribution for large κ for the circular case in Section 4.2.1. Let us discuss the M-H model for the unit circle (or at least $r > 1$) centred at the origin. Recall the model is

$$f(\boldsymbol{z}) \propto \exp\{-\frac{\kappa}{2}(\boldsymbol{z}^T\Sigma^{-1}\boldsymbol{z} - 1)^2\}, \tag{4.14}$$

with mode on the ellipse $\boldsymbol{z}^T\Sigma^{-1}\boldsymbol{z} = 1$ and $\boldsymbol{z} = r[\cos\theta, \sin\theta]^T$ where $\partial z = r\partial r\partial\theta$. Let $\Sigma = I$, then $\boldsymbol{z}^T\boldsymbol{z} = r^2$ where $r^2 > 0$ and let $\boldsymbol{z}^T\boldsymbol{z} = s + 1$, so $s = r^2 - 1$ and $s > -1$.

The equation (4.14) can be witten as a joint function of s and θ as

$$g(s,\theta) \propto \frac{1}{2} e^{-\frac{\kappa}{2}s^2}.$$

Then the marginal distribution of s is

$$\begin{aligned} g(s) &= \int_0^{2\pi} g(s,\theta)\partial\theta \\ &\propto e^{-\frac{1}{2}\frac{s^2}{\frac{1}{\kappa}}}\partial s, \end{aligned}$$

which is proportional to the truncated normal density with mean 0 and variance $\frac{1}{\kappa}$. Therefore, the variance decreases as κ increases which produces a narrower curve in Figure 4.5 for $\kappa = 100$.

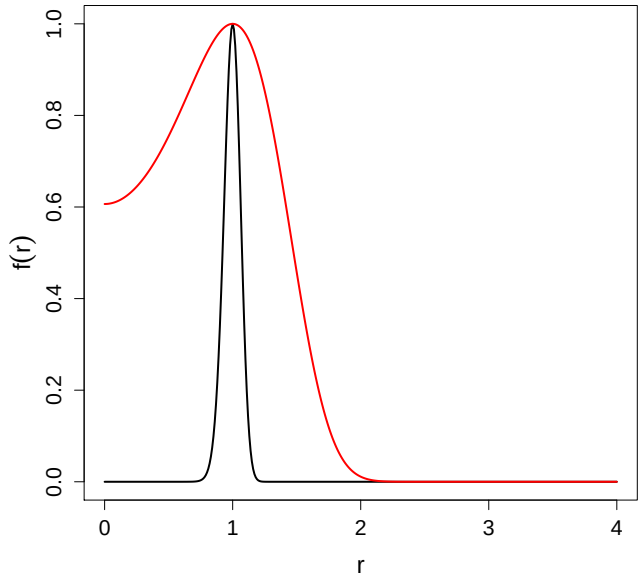

FIGURE 4.5: The M-H model density for $\kappa = 1$ in red and for $\kappa = 100$ in black.

Under high concentration κ, the mass of the M-H model is close to the mode, for example the M-H model of a unit circle centred at the origin (4.5) is close to the mode $\boldsymbol{z}_i^T \Sigma^{-1} \boldsymbol{z}_i = 1$ when $\kappa = 100$ as seen in Figure 4.5. Expanding the horizontal axis of Figure 4.5 for the $\kappa = 100$ curve, gives Figure 4.6 that the

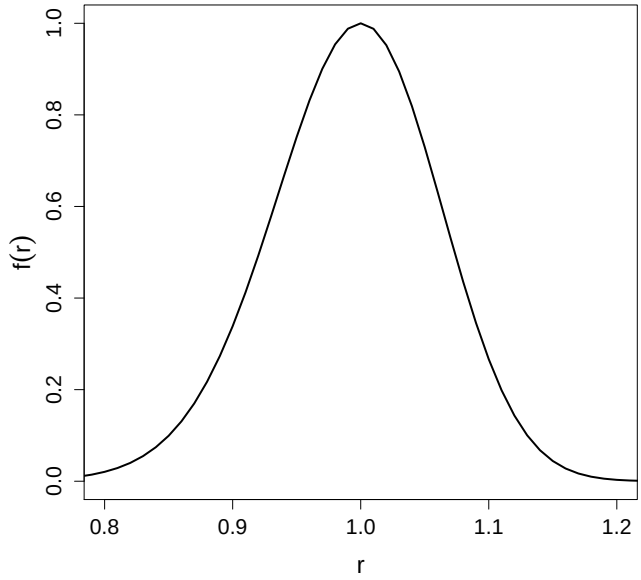

FIGURE 4.6: The M-H model density for $\kappa = 100$.

density looks normal (bell shape). Figure 4.5 shows that for $\kappa = 100$ case, when we go a little bit away from the mode, the density is zero. The truncation point is far away from the body of the distribution; hence the truncated normal is essentially the same as the normal distribution.

4.6 Estimating the helix axis using the M-H model

The Mardia-Holmes model is designed to fit a dataset consisting of points that are reasonably equally spread around a circle, and we can determine this by looking at the helix from above. Since the protein α-helix meets this criterion, we may project the helix data onto 2-dimensions (after estimating the helix axis) and then use the M-H model to find the optimal helix axis by maximizing the log-likelihood of the M-H model.

Now we describe what must happen before using the M-H model. Firstly, we estimate the initial helix axis $\hat{\boldsymbol{w}}$ using the OptLS method (see Section 3.5.3), and then we rotate the 3-dimensional data helix to semi-canonical coordinates as in the previous chapter (see section 3.5 specifically section 3.5.1).

If we want to project the 3-dimensional data helix points onto the 2-dimensional plane, that is perpendicular to the helix axis, we use the standard projection $f : \mathbb{R}^3 \to \mathbb{R}^2$, gives by $(y_{i1}, y_{i2}, y_{i3}) \to (y_{i1}, y_{i2})$. The projection matrix can be derived from the helix axis $\hat{\boldsymbol{w}}$ as follows: Let A be a rank one positive semi-definite matrix

$$A = \hat{\boldsymbol{w}}\hat{\boldsymbol{w}}^T.$$

Now we take the spectral decomposition of the matrix $A = ULU^T$, where $U = \left[\boldsymbol{u}_{(1)}, \boldsymbol{u}_{(2)}, \boldsymbol{u}_{(3)}\right]$ is a matrix of orthonormal eigenvectors of A and $L = \text{diag}(\ell_1, \ell_2, \ell_3)$ is a diagonal matrix of corresponding eigenvalues. Since A is a projection matrix, then the eigenvalues are either 0 or 1, where $\ell_1 = 0$. The eigenvector $\boldsymbol{u}_{(1)}$ correspond to the smallest eigenvalue is the helix axis $\hat{\boldsymbol{w}}$, that is $\hat{\boldsymbol{w}} = \boldsymbol{u}_{(1)}$. Thus, if we want to project the helix onto the plane that is perpendicular to the helix axis, then we need to use the second and the third, eigenvectors of A as the first and second columns of the projection matrix P. Hence, the projection matrix (see Lay (2006) , 452-453) is

$$P = \begin{bmatrix} \boldsymbol{u}_{(2)} & \boldsymbol{u}_{(3)} \end{bmatrix}_{3\times 2}, \tag{4.15}$$

Recall that our aim here is finding the optimal axis which maximizes the log-likelihood of the M-H model, and in order to do this we need two functions, say f_1 and f_2. Note that f_1 is nested in f_2. Function f_1 maximizes the log-likelihood of the M-H model in equation (4.6) numerically in R using `nlm` as in Section 4.2.2, assuming the helix axis is known. This function outputs the maximized the log likelihood. The idea behind the second function f_2 is to find the optimal axis that maximizes the output of f_1 over $\boldsymbol{w}$ using `nlm`. The inputs are the 3-dimensional

data helix points (y_{i1}, y_{i2}, y_{i3}), $i = 1, \ldots, n$. Recall $\boldsymbol{w}$ is parameterized using p_1 and p_2 in stereographic coordinates, see Section 3.5.3.

Given $\boldsymbol{w}$, f_2 projects the data onto the plane perpendicular to $\boldsymbol{w}$ and fits the M-H model in equation (4.6) to the 2-dimensional projected data. The analysis maximizes the log-likelihood of the M-H model using `nlm`. The initial parameters of the circle (the parameters are concentration $\kappa = 10$, radius $\rho = 2.3$, two location parameters $a = \text{mean}(y_{i2})$ and $b = \text{mean}(y_{i3})$), as discussed in Section 4.2.3.

Then f_2 maximizes the output of f_1 over $\boldsymbol{w}$ using `nlm`. The initial choose for p_1 and p_2 is (0,0) in f_2 since we point the helix to north pole. Multiple starting point near the optimal answer were trial; we always obtained the same estimator.

Note that the use of `nlm` both inside and outside f_2 slows our program in R. The M-H program takes 3.6 times longer than OptLS to estimate the helix axis.

4.6.1 Simulation studies

We now present an example of a helix that mimics a protein α-helix in semi-canonical form, with $n = 30$, and with errors that are simulated from the normal distribution with mean 0 and variance 0.05. First we estimate the helix axis using the OptLS method, and project the helix to the xy-plane. Then we use the functions described in Section 4.6 to fit the circle and to find the optimal axis. In order to run the functions, `nlm` requires initial values for the circle parameters and for the helix axis parameters. We use the means of the projection data as the initial parameters of the location (i.e. $a = \bar{y}_1, b = \bar{y}_2$); $r = 2$ as initial parameter for the radius (since the protein α-helix has radius 2.3); and $\kappa = 10$ as the initial parameter for the concentration of the data around a circle. The estimate of the helix axis acquired by the OptLS method to obtain the initial parameters of p_1 and p_1 of the second function, but since our helix is in semi-canonical form then initial parameters are chosen to be (0,0). Upon running `nlm` we find that

the maximum log-likelihood of the M-H model is 43.4. The estimated axis by the OptLS mean square error from the true axis $\boldsymbol{w} = (0, 0, 1)^T$ is $1 - \hat{\boldsymbol{w}}^T\boldsymbol{w} = 5 \times 10^{-6}$ (Section 3.6). Then M-H program for estimating the axis gives the maximum log-likelihood of the M-H model is 42.8 and the estimated axis by M-H model mean square error is $1 - \hat{\boldsymbol{w}}_{M-H}^T\boldsymbol{w} = 2 \times 10^{-5}$.

Next we simulate 100 helices that mimic a protein α-helix for different choices of sample size n and parameter values r, c, and σ^2. After implementing the OptLS and M-H models on the simulated helices, we get 100 estimates of the helix axis $\hat{\boldsymbol{w}}_{M-H,i}$ using the M-H model and 100 estimates of $\hat{\boldsymbol{w}}_i$ using the OptLS method. The mean square error (see Section 3.6) allows us to calculate how accurate these estimates are. The mean square error of the M-H model $1 - \hat{\bar{\boldsymbol{w}}}_{M-H}^T\boldsymbol{w}$ is greater than the mean square error of the OptLS $1 - \hat{\bar{\boldsymbol{w}}}^T\boldsymbol{w}$ as shown in Table 4.3 below (see Section 3.6) from large number of simulated helices for different choices of sample size or parameter values r, c and σ^2. From the results shown on Table 4.3 we can conclude that OptLS is always better than M-H model, as sometimes by a factor of two interms of variance, othertimes is much worse.

TABLE 4.3: Comparison between M-H model and OptLS by the mean square error computed from different simulated helices.

	set1	set 2	set 3	set 4	set 5	set 6
n	30	30	12	12	12	12
r	2.3	2.3	2.3	2.3	7	7
c	$\frac{5.4}{(2\pi)}$	$\frac{5.4}{(2\pi)}$	$\frac{5.4}{(2\pi)}$	$\frac{5.4}{(2\pi)}$	0.1	$\frac{5.4}{(2\pi)}$
σ^2	0.001	0.05	0.05	0.1	0.05	0.05
M-H	2.8×10^{-7}	1.5×10^{-5}	2.4×10^{-4}	4.5×10^{-4}	1.2×10^{-2}	2.3×10^{-4}
OptLS	1.2×10^{-7}	5.9×10^{-6}	1.4×10^{-4}	2.8×10^{-4}	1.6×10^{-4}	8.1×10^{-5}

Overall, the M-H model gives us a way to estimate a helix axis, but this is not better than our OptLS method. As seen in Table 4.3, the mean square error of the M-H model is much larger than that of the OptLS method, and so we may conclude that the M-H model is less accurate than OptLS method.

The M-H model uses nested optimization functions (`nlm` within another `nlm`), which drastically slows down the program. Moreover, the use of `nlm` requires initial parameters, but in the M-H model some of these must be guessed; for a bad choice it will either not converge at all, or it will converge to a local optimal solution rather than a global optimal. Even though the OptLS method uses `nlm` to optimize the helix axis estimate, no initial parameters for the helix axis need to be guessed because they are obtained from the stage one of the algorithm.

Chapter 5

Estimation process for fitting a bent helix

Recall that a *kink* in a helix is where the helix axis changes its direction (see Section 1.5.1). A kink can also be thought of as a *change point*. In this chapter we treat the location of a change point as occurring half way between two data points, and label the change point position by the last point of the first block of data. A helix with a change point is accordingly called a *bent* helix. Bend sounds more gradual than the helix. This terminology goes against the normal English usage which would define 'kink' and 'bend' the other way round.

In chapter 3 we described a method, the OptLS method in Section 3.5, to fit a regular helix. In this chapter, however, we develop a methodology to deal with bent helices. We will see in Section 5.1 that this strategy, the change point phase of Bending-Detector, encompasses the OptLS method.

The change point literature is large e.g Chen and Gupta (2011), and Kim and Siegmund (1989). Much of the literature focuses on changes in means and variances, but there is some discussion of change in slope, e.g. Miao (1989). A change in slope for an original process, e.g. $\{X_t\}$, means a change point in the mean in the differenced process, $Y_t = X_t - X_{t-1}$. A helix change point is a change

in the 3-dimensional direction of an axis; hence it can be regarded as a change in slope for a 3-dimensional process.

The general idea of the change point phase of Bending-Detector (see Section 5.1) is thus: firstly, find all the possible change points; make appropriate cuts of the helix at these points to obtain regular helices; and then use the OptLS to fit these resultant helices.

In order to test for the presence of a change point we develop a simulation based test in Section 5.2, called the testing phase of Bending-Detector. We simulate from the null hypothesis H_0 : 'the helix has no change point' of regular helices with various σ^2 and various number of landmarks to establish a *threshold* (95% quantile). The null distribution of the test statistic depends on: (a) whether the change point location is known or not; (b) number of points n; (c) variance σ^2. In this chapter we take the protein α-helix value of $\delta = \frac{2\pi}{3.6}$.

If we find that the helix has a change point, we investigate further to understand more about the change point. In particular, we define six statistics which describe how the two blocks of the whole dataset differ. We study these statistics using a parametric bootstrap procedure in Section 5.3. We call this step the features analysis phase of Bending-Detector.

Overall, we have a procedure to look for a change point in a helix. First, we test the hypothesis H_0 : 'the helix has no change point' against the hypothesis H_1 : 'the helix has a change point' (the testing phase of Bending-Detector). Second, if the null hypothesis is rejected, then we find the change point or bend position, once a change point has been identified, we fit each sub-helix separately (the change point phase of Bending-Detector). Third, we investigate the reasons for the change point (the features analysis phase of Bending-Detector). In conclusion, a statistical test using the parametric bootstrap is used to categorise if the helix is bent or not (in Section 5.2), find a change point (in Section 5.1), and to study the reasons of this bend (in Section 5.3). We call this procedure *Bending-Detector*,

see Alfahad et al. (2018). We compare our Bending-Detector with Kink-Detector by Mardia et al. (2018). Kink-Detector is a technique which categorises a helix with a kink or not, and if so locates the kink position. It starts by looking at a moving window of 12 points on a helix, and each time it calculates the dot product of the two axes ($\cos\theta$), the first six points axis and the second six points axis, where θ is the angle between these two axes. If a helix has a region of four consecutive points with $\cos\theta < 0.9818$, then this helix is categorised as a kinked helix. This procedure permits a helix to have some curvature in its axes but not be categorised as a kink. Since we study the protein α-helix which is allowed to have some bending, and this bending is not classified as a kink (change point). Bending-Detector treats a change point from a global point of view, whereas, Kink-Detector treats a change point from a local point of view. Differences between the two methods on a sample data helices are discussed in Section 5.5.

5.1 Bending-Detector Change Point

In this section, our main goal is to describe the change point phase of Bending-Detector. We define the change point phase of Bending-Detector for a helix which is known to have just one kink (change point), see Section 5.1.1.

5.1.1 A helix with a single kink

Our strategy for finding a change point in a bent helix is to start by choosing one or more points that we think it may be the change point. We take each of these points in turn, and each time cut the helix at the point and fit each resultant part of the helix using OptLS. In this way we will obtain a collection of residual sum of squares and the minimum of these values will indicate the estimated change

point. We call this *the change point phase of Bending-Detector*. Note that OptLS is nested with the change point phase of Bending-Detector.

More explicitly, we start by assuming each point k, $n_1 \leq k \leq n_2$ is a potential change point. However, to avoid estimation problem near the endpoints we follow Mardia et al. (2018) and limit attention to display $n_1 + 5 \leq k \leq n_2 - 6$,

Step 1 First we need to put the whole helix, H, into semi-canonical coordinates, so H^o is in semi-canonical where $\boldsymbol{w} = [0, 0, 1]^T$, using our OptLS from Section 3.5.3. We recall (3.7) for convenience that H^o can then be described by the equation

$$\boldsymbol{z}(t_i) = r\cos(t_i - \tau)\boldsymbol{u} + r\sin(t_i - \tau)\boldsymbol{v} + ct_i\boldsymbol{w} + \boldsymbol{b} + \boldsymbol{\varepsilon}_i,$$

where $t_i = (i-1)\delta$, where $\delta = \frac{2\pi}{3.6}$ is assumed to be known as in the protein α-helix, and $\Gamma = [\boldsymbol{u}\ \boldsymbol{v}\ \boldsymbol{w}] = \Gamma_0$, where $\Gamma_0 = I_3$. After that, the helix axis is the z-coordinate (the vertical axis) and the x and y coordinates are in the horizontal plane. We cut this helix H^o between $\boldsymbol{z}_k$ and $\boldsymbol{z}_{k+1}$. This yields two helices $H_k^{(1)}, H_k^{(2)}$, which we call *sub-helices* of H^o, where $H_k^{(1)}$ consists of the points $\boldsymbol{z}_i, i = n_1, \ldots, k$, and $H_k^{(2)}$ consists of the points $\boldsymbol{z}_i, i = k+1, \ldots, n_2$. However, the two sub-helices $H_k^{(1)}$ and $H_k^{(2)}$ cannot be in semi-canonical coordinates simultaneously, so that the estimated matrices $\Gamma^{(\ell)} = [\boldsymbol{u}^{(\ell)}, \boldsymbol{v}^{(\ell)}, \boldsymbol{w}^{(\ell)}]$ for $\ell = 1, 2$ in (3.7) cannot both be the identity matrix. Therefore, $H^{(\ell)}$, can be modelled by

$$\boldsymbol{y}^{(\ell)}(t_i) = r^{(\ell)}\cos(t_i)\boldsymbol{u}^{(\ell)} + r^{(\ell)}\sin(t_i)\boldsymbol{v}^{(\ell)} + c^{(\ell)}t_i\boldsymbol{w}^{(\ell)} + \boldsymbol{b}^{(\ell)} + \boldsymbol{\varepsilon}_i^{(\ell)}, \tag{5.1}$$

where $n_1 \leq i \leq k$ for $\ell = 1$ and $k+1 \leq i \leq n_2$ for $\ell = 2$.

Step 2 Apply OptLS (Section 3.5.3) to $H_k^{(1)}$ and $H_k^{(2)}$, and obtain the residual sum of squares $\mathrm{RSS}_k^{(1)}$, and $\mathrm{RSS}_k^{(2)}$, respectively and set $SSW_k = RSS_k^{(1)} + RSS_k^{(2)}$. Then SSW_k is called the *within residual sum of squares*.

Step 3 Let $SSW_{\min} := \min\{SSW_k | n_1 + m - 1 \leq k \leq n_2 - m\}$, $m = 6$. The

corresponding estimated change point location will be denoted $\hat{k}$ where $SSW_{\min} = SSW_{\hat{k}}$. Note that $H_{\hat{k}}^{(1)}$ and $H_{\hat{k}}^{(2)}$ will be two regular helices (since we are assuming H has a single change point), which we have already fitted in step 2.

5.2 The testing phase of Bending-Detector

To decide if the helix is bent or not we specify a test statistic to test the null hypothesis H_0: 'the helix has no change point' against the alternative hypothesis H_1: 'the helix has a change point'. Our proposed test statistic determines if the change point phase of Bending-Detector gives a significantly better fit to the data. Recall that we assumed from the beginning that the errors are independent and isotropic normally distributed. For this reasons we choose an F-test (as in analysis of variance) as it meets our need for the test statistic, (see e.g.; Mood et al., 1974, p. 437; Knight, 2000, Section 8.1).

Fitting the whole (single) helix using OptLS gives the *total* residual sum of squares SST. The change point phase of Bending-Detector for a given value of k_0 gives the within residual sum of squares SSW_k. Note that $SST \geq SSW_k$ since the single helix model is a special case of the change point model. Finally, define $SSB_k = SST - SSW_k$. In the classical ANOVA setting SSB_k also has an explicit representation as a between residual sum of squares; however, in the helix setting, it can only be defined as a difference.

There are 8 parameters needed to specify a helix: two for the helix axis $\boldsymbol{w}$; and six in stage 2 which are r, c, b_1, b_2 and b_3; see Section 3.5. The total degrees of freedom is $df = 3n$, and the residual degrees of freedom after fitting a single helix is $df_T = 3n - 8$. The residual degrees of freedom after fitting a bent helix is $df_W = df_T - 8 = 3n - 16$, because the two sub-helices are estimated separately and each has 8 unknown parameters. We can also obtain the between degrees of

freedom $df_B = df_T - df_W = 8$. Then the function for the F-statistic is

$$F_k = \frac{SSB_k/df_B}{SSW_k/df_W} \sim F_{0.05}(8, 3n-16). \tag{5.2}$$

If this statistic is greater than the critical value we reject the null hypothesis and then we conclude the helix is a bent helix i.e. there is a change point. We expect the F_k-statistic defined by equation (5.2) to have an F-distribution $F_\alpha(df_B, df_W)$, $\alpha = 0.05$, because we assumed that the errors are independent isotropic normal distributions $N_3(\mathbf{0}, \sigma^2 I_3)$ with small σ^2. Thus we expect SSB and SSW to be approximately independent and have $\sigma^2\times$chi-squared distribution with df_B and df_W degrees of freedom, respectively, (see e.g.; Mood et al., 1974, p. 437; Rice, 2007, p. 482; Knight, 2000, Proposition8.2). Note σ^2 can be estimated either by the residual variance $\hat{\sigma}^2$ after fitting the single helix (under the null hypothesis), or by the pooled residual variance $\hat{\sigma}_p^2$ (see e.g. Reddy, 2011, p. 109) after fitting the bent helix (under the alternative hypothesis), as follows

$$\hat{\sigma}^2 = \frac{SSB_k + SSW_k}{df_B + df_W}, \quad \hat{\sigma}_p^2 = \frac{SSW_k}{df_W}. \tag{5.3}$$

Note that as SSW_k decreases, F_k will increase, so if $SSW_{\min}$ corresponds to the estimated change point then

$$F_{\max} := \max\{F_k : n_1 + m - 1 \leq k \leq n_2 - m\}, \tag{5.4}$$

will also correspond to the estimated change point.

A *threshold* value indicates how extreme any observed results need to be in order to reject the null hypothesis. More explicitly, if the statistic is greater than or equal to the threshold, then we reject the null hypothesis and conclude the helix has a change point.

In this section, we derive the threshold by simulation (as in Section 3.6) from the null hypothesis. We create a mathematical regular helix (1.1), with n

landmarks where the parameters r, c, and δ mimic a protein α-helix (see Section 1.5). For various values of σ^2 we simulate $3n \times n_{\text{boot}}$ sets of random errors from $N(\mathbf{0}, \sigma^2 I)$ and add each set to the mathematical regular helix, so we have $n_{\text{boot}} = 10,000$ regular statistical helices (see Section 1.2.4). For each simulated bootstrap sample j we calculate the maximum F_k, denoted $F^*_{\text{max},j}$, which yields the simulated distribution of F^*_{max}. Then the threshold $F^{*(\alpha)}_{\text{max}}$ is the $(1-\alpha)\%$ quantile of the values $\{F^*_{\text{max},j}\}_{j=1}^{n_{boot}}$

In order to derive the threshold in practice for a given dataset, we carry out a parametric bootstrap simulation of the null hypothesis. The simulation by parametric bootstrap is carried out in the following way:

- We fit the single helix by OptLS to estimate the 8 parameters (i.e. we have the fitted data).
- We fit the helix by the change point phase of Bending-Detector to estimate pooled residual variance estimate, $\hat{\sigma}^2_p$.
- We simulate $3n \times n_{\text{boot}}$ sets of random errors from $N(\mathbf{0}, \hat{\sigma}^2_p I)$ and add each set of errors to the single fitted data, so we have $n_{\text{boot}} = 10^4$ new helices.

If the F^*_{max} statistic follows an F-distribution, then the simulated threshold $F^{*(\alpha)}_{\text{max}}$ will be close to the tabulated value from F table at α-level (adapted from Sullivan (2008), p. 125). However, we can still use the statistic F^*_{max} without any reference to the F-distribution, and in this case we choose the threshold to be the $1-\alpha$ quantile for the simulated F^*_{max} distribution.

A Q-Q plot can be used to assess whether the F_{max}-statistic follows an F-distribution or not. We plot the statistics of 10 000 helices, sort in ascending order, versus 10 000 equally-spaced quantiles from an F-distribution with degrees of freedom 8 and 74. If the points in the Q-Q plot approximately lie on a straight diagonal line, then we can say that the statistic F_{max} is F-distributed.

In the following, we split our study into two cases: (a) when we assume the change point position (the fixed location) is known; and (b) when the change point position is unknown. In the first case, the simulated distribution of the test statistic is generally close to an F-distribution. In the case (b) the simulated distribution of the test statistic is always far away from an F-distribution. Hence we cannot use the F-distribution tabulated critical value to find the threshold when k is unknown so we use bootstrap.

5.2.1 Known change point position

For this subsection we assume that the change point position of our helix is known. Recall from Section 5.1.1 that the change point will lie between points $n_1 + 5$ and $n_2 - 6$. We assume here that the change point location $k = 8$, which is the 8^{th} landmark $\boldsymbol{y}_8$.

We simulate 10^4 regular helices that mimic protein α-helix with 30 landmarks, where the errors follow $\text{N}_3(\mathbf{0}, 0.05I_3)$, as in Section 5.2. We compute the statistic $F^*_{\max}$ in equation (5.4) for each simulated helix. This gives a sample form the distribution of $F^*_{\max}$. The 95^{th} quantile is $F^{*(0.05)}_{\max} = 2.058$, which is very close to the F-distribution threshold (tabulated critical value) $F_{0.05}(8, 74) = 2.066$ at $\alpha = 0.05$, i.e. the statistic approximately follows an F-distribution. There are 487 $F^*_{\max}$ statistics greater than the critical value i.e. p-value of 0.0487. Since the significance level appears to be $\alpha = 0.0487$ (the probability of rejecting the null hypothesis when it is true), then we can say that approximately 5% of the $F^*_{\max}$ statistics are greater than the critical value of $F_{0.05}(8, 74)$. This explains the Q-Q plot in Figure 5.1; most of points lie on the diagonal but there is some deviance on the upper end, i.e. the simulated distribution disagrees in the upper tail, only around 500 points are above the 0.95 quantile, so this is not enough to say it is not an F-distribution.

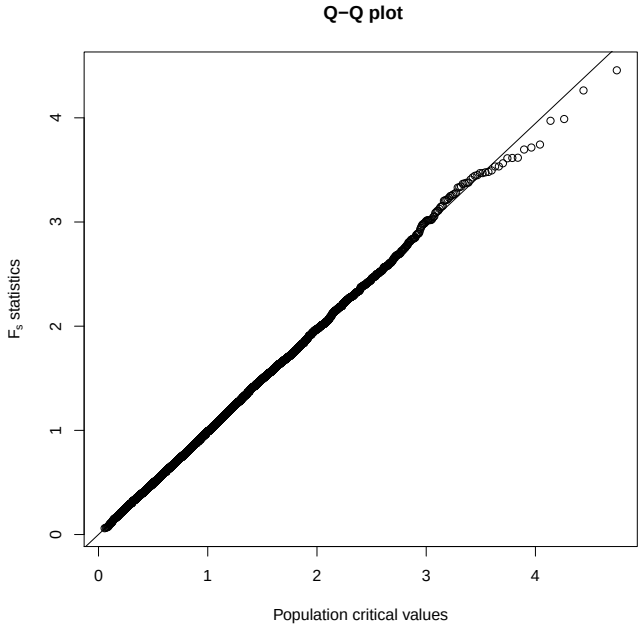

FIGURE 5.1: The Q-Q plot of F-distribution: 10 000 simulated ordered $F_{\max}$-statistics versus quantiles from an F-table, note the upper tail apart from the F distribution.

To test if $\alpha = 0.0487$ is compatible with the usual choice of $\alpha = 0.05$, we carry out a binomial test to test the null hypothesis $H_0 : \alpha = 0.05$ and the alternative hypothesis $H_1 : \alpha \neq 0.05$. We do not have evidence to reject the null hypothesis with an obtained p-value of 0.5663 and the 95% confidence interval for α is (0.045, 0.054). Therefore, we can say that 0.0487 is compatible with the used 0.05.

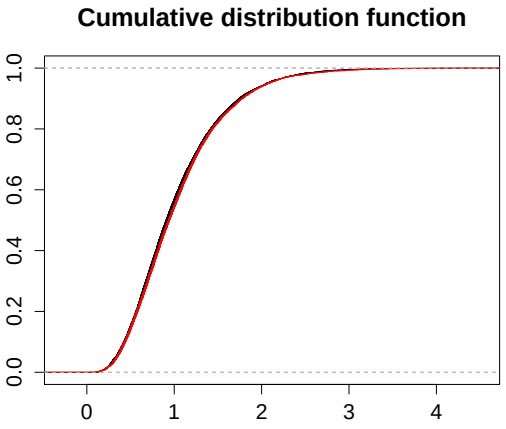

FIGURE 5.2: The plot of 10 000 CDF simulated $F^*_{\max}$-statistics in black and a F-distribution CDF in red, where $n = 30$ and $\sigma^2 = 0.05$. Note the two curves are matched perfectly.

Furthermore, Figure 5.2 shows that the F cumulative distribution function (CDF) with $df_B = 8$ and $df_W = 74$ degrees of freedom closely matches the CDF for $F^*_{\max}$. Therefore, we reject the null hypothesis if the F_k statistic value from the data helix is greater than the critical value $F_\alpha(df_B, df_W)$. In other words, we can categorize a data helix as bent helix if the F_k statistic obtained from the data exceeds the threshold of $F_\alpha(df_B, df_W) = 2.066$ with $\alpha = 0.05$.

5.2.2 Unknown change point position

Having considered the known change point case in Section 5.2.1, we now carry out a 10 000 simulation of regular protein α-helices, with n landmarks and where the errors are drawn from $N_3(\mathbf{0}, \sigma^2 I_3)$, but for which we do not know where the change point is. We vary σ^2 and n independently, whilst fixing all the other parameters, in order to see how this affects the threshold value and how close the $F^*_{\max}$ statistics follow an F-distribution. We carried out 5 simulations each of $n_{boots} = 1000$ as follows: (a) $n = 15, \sigma^2 = 0.05$; (b) $n = 25, \sigma^2 = 0.05$; (c) $n = 30, \sigma^2 = 0.05$; (d) $n = 30, \sigma^2 = 0.01$; and (e) $n = 30, \sigma^2 = 0.1$. Our results are presented in Table 5.1 below. For each case, the Q-Q plot in Figure 5.4, shows that the $F^*_{\max}$ statistic distribution is far away from the F-distribution. Furthermore, for fixed $\sigma^2 = 0.05$ but varying n, the threshold $F^{*(0.05)}_{\max}$ varies. We conclude that the simulated distribution is affected by the number of landmarks. Therefore, for each different dataset we obtain the threshold value $F^{*(\alpha)}_{\max}$ separately.

TABLE 5.1: Threshold $F^{*(0.05)}_{\max}$ for unknown k with $n_{\text{boot}} = 1000$ simulations in each case as the number of landmarks and error variance σ^2 vary.

n	σ^2	$F^{*(0.05)}_{\max}$
15	0.05	2.877
25	0.05	3.073
30	0.05	3.008
30	0.01	3.008
30	0.1	3.015

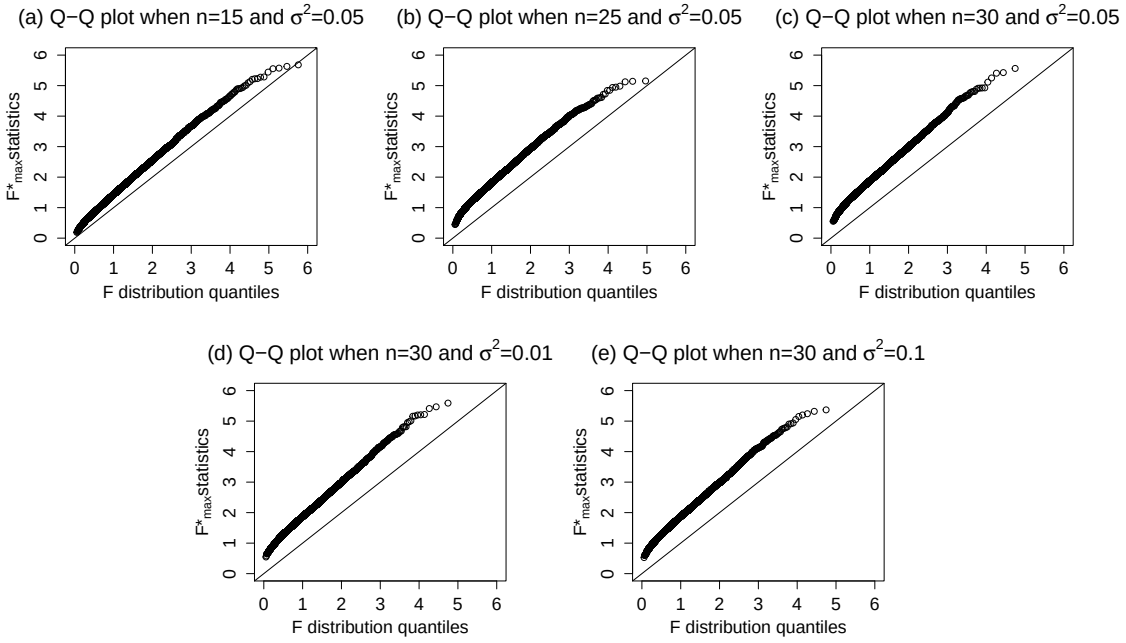

FIGURE 5.3: The Q-Q plots of various n and σ^2 where 10 000 $F_{\max}^{*(0.05)}$ statistics versus quantiles from the F-distribution.

5.3 Analysis of F_k

Given a data helix, we can determine the threshold $F_{\max}^{*(\alpha)}$ by simulation using parametric bootstrap. Recall, if the data statistic $F_{\max}$ is greater than the threshold $F_{\max}^{*(\alpha)}$ we reject the null hypothesis, i.e. the given helix has a change point, say at landmark k. In this situation, we have two regular sub-helices $H^{(\ell)}, \ell = 1, 2$. The real α-helix may have one change point or none (Mardia et al., 2018). Since our numerator degrees of freedom (df) of the F_k statistic is 8, there are 8 features in which the two sub-helices may agree. Note that for a regular helix, these two sub-helices meet perfectly and so they agree on all 8 features. We now explain the 8 characteristics in which the two sub-helices may differ, and we separate these into six groups of features.

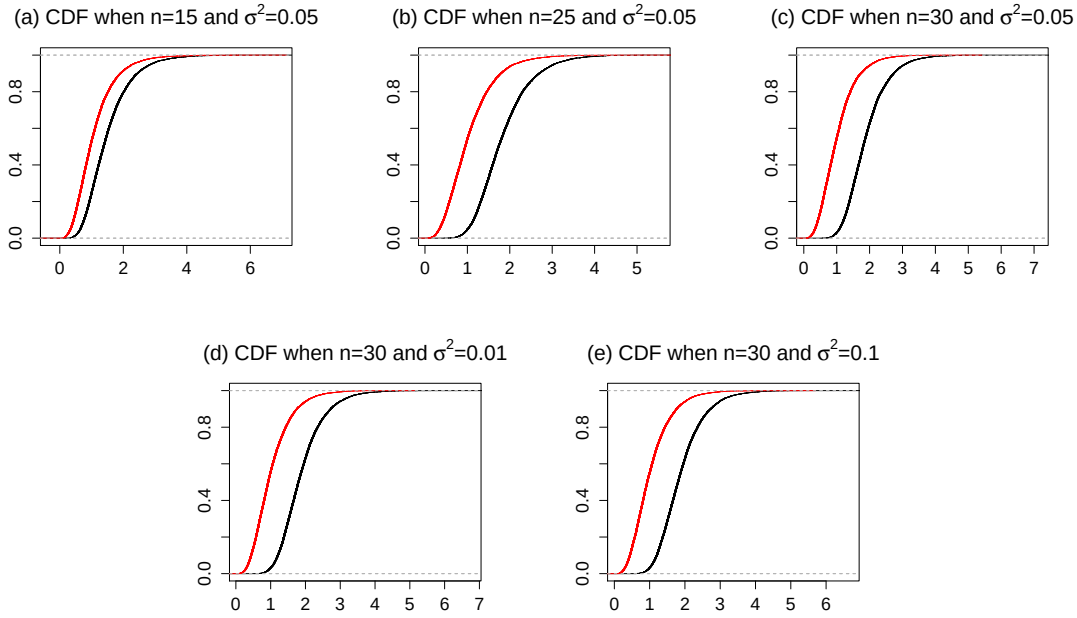

FIGURE 5.4: The CDF simulated $F_{\max}^{*(0.05)}$ statistics in black and a F-distribution CDF in red plots for various n and σ^2.

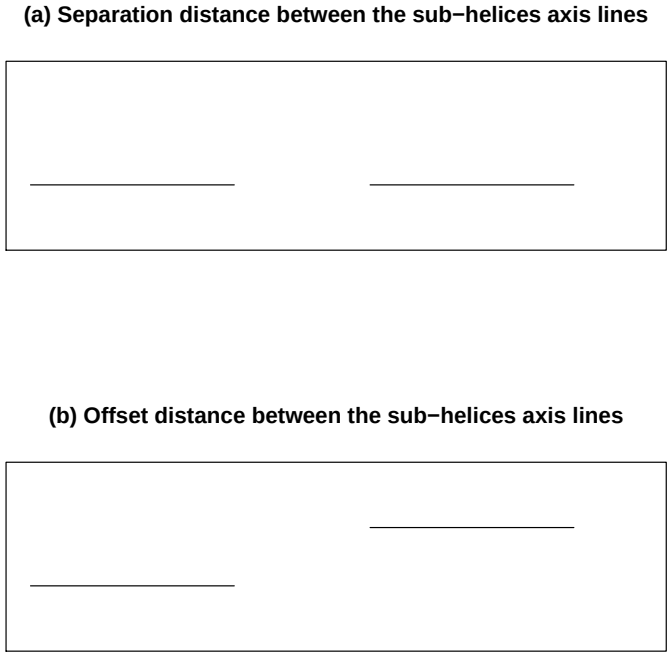

FIGURE 5.5: Example showing the separation between the two axes lines. Panel (a) presents the the separation explain in group 2 and panel (b) presents the separation explain in group 3.

GROUP 1 If the two sub-helices $H^{(1)}, H^{(2)}$ do not have the same helix axis direction, then $\boldsymbol{w}^{(1)} \neq \boldsymbol{w}^{(2)}$. Note that $\boldsymbol{w}^{(\ell)}, \ell = 1, 2$ has 3 coordinates, $\boldsymbol{w}^{(\ell)}$ is a unit vector and so any two coordinates determine the third (2 df).

GROUP 2 The two sub-helices $H^{(1)}$ and $H^{(2)}$ can differ by a shift along the helix axis (1 df) as presented in Figure 5.5 (a).

GROUP 3 The two sub-helices $H^{(1)}$ and $H^{(2)}$ can differ by an offset perpendicular to the helix axis (2 df) as presented in Figure 5.5 (b).

GROUP 4 One of the sub-helices $H^{(1)}$ or $H^{(2)}$ can spin with respect to the other (1 df).

GROUP 5 If the two sub-helices, $H^{(1)}$ and $H^{(2)}$, do not have the same helix radius then $r^{(1)} \neq r^{(2)}$ (1 df).

GROUP 6 If the two sub-helices, $H^{(1)}$ and $H^{(2)}$, do not have the same helix pitch then $c^{(1)} \neq c^{(2)}$ (1 df).

From the discussion above, we have six statistics $A_1, \ldots, A_6$, where these statistics are not necessarily independent and correspond to the differences that are detailed in Group $1, \ldots, 6$, respectively. Before defining $A_1, \ldots, A_6$ explicitly, we need to introduce the theoretical point at which the sub-helices $H^{(1)}$ and $H^{(2)}$ "meet". Recall for a helix in semi-canonical coordinates and change point k, $\boldsymbol{z}(t_k)$ is the last point on $H^{(1)}$, while $\boldsymbol{z}(t_{k+1})$ is the initial point of $H^{(2)}$. Thus, we may take the point at which they meet to be at time $t_{k+\frac{1}{2}}$, which is between time t_k and t_{k+1}. Let $\hat{\Gamma} = [\hat{\boldsymbol{u}}\ \hat{\boldsymbol{v}}\ \hat{\boldsymbol{w}}]$ be the orientation matrix after the fitting the whole helix H. Let the transform data $\boldsymbol{q}^{(\ell)} = (q_1^{(\ell)}, q_2^{(\ell)}, q_3^{(\ell)})^T$ be the fitted helix point in semi-canonical coordinates at time $t_{k+\frac{1}{2}}$ for $H^{(\ell)}$, which can be defined as follows

$$\boldsymbol{q}^{(\ell)} = \hat{\Gamma}\boldsymbol{y}^{(\ell)}(t_{\hat{k}+\frac{1}{2}}),$$

and recall each of the sub-helices can be modelled as in equation (5.1), then the equation of landmark $\boldsymbol{y}^{(\ell)}(t_{\hat{k}+\frac{1}{2}})$, for $\ell = 1, 2$, is

$$\boldsymbol{y}^{(\ell)}(t_{\hat{k}+\frac{1}{2}}) = \hat{r}^{(\ell)} \cos(t_{k^*+\frac{1}{2}})\hat{\boldsymbol{u}}^{(\ell)} + \hat{r}^{(\ell)} \sin(t_{k^*+\frac{1}{2}})\hat{\boldsymbol{v}}^{(\ell)} + \hat{c}^{(\ell)} t_{k^*+\frac{1}{2}} \hat{\boldsymbol{w}}^{(\ell)} + \hat{\boldsymbol{b}}^{(\ell)}.$$

and the projection function is

$$\boldsymbol{g}^{(\ell)}(t_{\hat{k}+\frac{1}{2}}) = \hat{c}^{(\ell)} t_{k^*+\frac{1}{2}} \hat{\boldsymbol{w}}^{(\ell)} + \hat{\boldsymbol{b}}^{(\ell)}.$$

After projecting the helix onto its axis, we can find the fitted helix axis point $\boldsymbol{p}^{(\ell)} = (p_1^{(\ell)}, p_2^{(\ell)}, p_3^{(\ell)})^T$ of this notional landmark under the model for H_ℓ as

$$\boldsymbol{p}^{(\ell)} = \hat{\Gamma} \boldsymbol{g}^{(\ell)}(t_{\hat{k}+\frac{1}{2}}),$$

The difference between $\boldsymbol{q}^{(1)}$ and $\boldsymbol{q}^{(2)}$ will be used in testing the spin parameters. In addition, the difference between $\boldsymbol{p}^{(1)}$ and $\boldsymbol{p}^{(2)}$ is of our interest in testing the shift and offset parameters.

Group 1 We want to determine if the two sub-helix axis directions $\hat{\boldsymbol{w}}^{(1)}$ and $\hat{\boldsymbol{w}}^{(2)}$ are equal or not. For this consider the following statistic

$$A_1 := 1 - \cos\theta,$$

where $\hat{\boldsymbol{w}}^{(1)T}\hat{\boldsymbol{w}}^{(2)} = \cos\theta$ and θ is the angle between the two sub-helices axis. If the helix axis directions of $H^{(1)}$ and $H^{(2)}$ are equal, i.e. H is a regular helix, then we will have $\hat{\boldsymbol{w}}^{(1)T}\hat{\boldsymbol{w}}^{(2)} = 1$, and hence $A_1 = 0$. So we are interested in the magnitude of A_1.

Group 2 The two sub-helices $H^{(1)}$ and $H^{(2)}$ can be shifted from each other, as it is seen in Figure 5.5 (a). We want to test this difference of the fitted axis points'

third component $p_3^{(1)}$, $p_3^{(2)}$ so we define the shift parameter statistic as follows

$$A_2 := (p_3^{(1)} - p_3^{(2)})^2$$

Group 3 Similarly, the $H^{(1)}$ and $H^{(2)}$ might be offset from each other as it is seen in Figure 5.5 (b). This means they may differ in the 2 perpendicular directions to the axis direction, i.e. $\boldsymbol{p}^{(1)}$ may differ from $\boldsymbol{p}^{(2)}$, and any discrepancy can be detected by computing the Euclidean distance between these points in $\mathbb{R}^2$. Therefore, we define the offset parameters to be

$$A_3 := (p_1^{(1)} - p_1^{(2)})^2 + (p_2^{(1)} - p_2^{(2)})^2$$

Group 4 The spin parameter will test if the two sub-helices are aligned or if there is some twisting. We estimate the angles φ_1 and φ_2 between the fitted points $\hat{\boldsymbol{q}}^{(1)}$ and $\hat{\boldsymbol{q}}^{(2)}$ and the axis fitted points $\hat{\boldsymbol{p}}^{(1)}$ and $\hat{\boldsymbol{p}}^{(2)}$, and then the spin parameter is the difference between these angles. The estimates of φ_1 and φ_2 is

$$\hat{\varphi}_1 = \text{atan2}(q_2^{(1)} - p_2^{(1)}, q_1^{(1)} - p_1^{(1)}),$$
$$\hat{\varphi}_2 = \text{atan2}(q_2^{(2)} - p_2^{(2)}, q_1^{(2)} - p_1^{(2)}).$$

Then the spin parameter statistic is given by

$$A_4 :=\mid \hat{\varphi} \mid=\mid \hat{\varphi}_1 - \hat{\varphi}_2 \mid,$$

where $\hat{\varphi} \in [-\pi, \pi)$.

Group 5 and 6 The final two differences are the radius $r^{(\ell)}$ and pitch $c^{(\ell)}$ of the two sub-helices $\ell = 1, 2$. The radius parameter is

$$A_5 :=\mid \hat{r}^{(1)} - \hat{r}^{(2)} \mid,$$

and the pitch parameter is

$$A_6 :=\mid \hat{c}^{(1)} - \hat{c}^{(2)} \mid .$$

In order to test all these statistics $A_1, \ldots, A_6$, we first simulate two datasets one of a regular helix and the other of a bent helix. In addition, we use a 1000 parametric bootstrap for each of the nine real data helices from Mardia et al. (2018). For each helix this proceeds is follows. Using the fitted data, we simulate q helices of the same size where the errors are simulated from a normal distribution $N(\mathbf{0}, \hat{\sigma}_p^2 I_3)$, and $\hat{\sigma}_p^2$ is the estimated pooled variance (under the alternative hypothesis i.e. fit the helix by the change point phase of Bending-Detector). For each simulated helix we calculate $A_1^*, \ldots, A_6^*$ as described above and we have a distribution for each statistic. For each statistic we reject the null hypothesis if the computed statistic for our data helix falls in the upper right tail of this distribution, i.e. above 0.95 quantile.

5.4 Simulation studies

In this section we apply our method to two simulated datasets (for regular and bent helices) in order to test the accuracy of our Bending-Detector. We simulate a statistical regular helix with $n = 27$ landmarks that mimics a protein α-helix (i.e. $r = 2.3, 2\pi c = 5.4, \beta = \frac{2\pi}{3.6}$) with errors normally distributed with mean 0 and variance $\sigma^2 = 0.05$. For the bent helix, we simulate the same statistical helix as above but introduce a bend of $\theta = 0.3$ radians about the x-axis at $k = 12$. Table 5.3 presents the results when applying the Bending-Detector to these two simulated helices and, in parentheses, the p-values using a bootstrap sampling with $n_{\text{boot}} = 1000$.

First we discuss the results for the regular helix. The residual variance under the null hypothesis $\sigma^2 = 0.039$, and the pooled residual variance under the

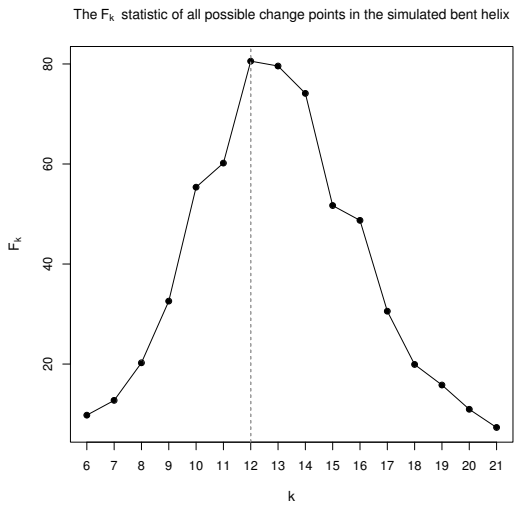

FIGURE 5.6: The F_k statistic against the possible choice of k for the simulated change point helix, where the maximum F_k at $k = 12$.

TABLE 5.2: The frequency table of k^* from 1000 bootstrap samples for the bent helix.

k	12	13	14
frequency	728	220	52

alternative $\sigma_p^2 = 0.035$, are approximately equal. Hence we can confirm that the regular helix has no bend, which agrees with our data. As shown in Table 5.3, the test statistic $F_{\max}$ is not significant, which also indicates the helix has no bend. Although the $F_{\max}$ is not significant, we analyse $F_{\max}$ by testing the features $A_1, \ldots, A_6$ and these are also not significant.

For the bent helix, the p-value for $F_{\max}$ is highly significant, suggesting that the helix is bent. The Bending-Detector estimated the bend position to be $\hat{k} = 12$ as shown in Figure 5.6, which is the true bend position. To see how accurate $\hat{k}$ we generate $n_{\text{boot}} = 1000$ bootstrap replicates of k^* from a single data set as in Section 5.2. Table 5.2 presents for 1000 simulated data, the $k^* = 12$ is most frequently value. An estimate of the angle between the two sub-helices $\hat{\theta} = 0.32$ radians is close to the true $\theta = 0.3$. As shown in Table 5.3, the feature A_1 (change in axis) is highly significant that the helix has a bend, which is due to a change in axis direction.

Figure 5.7 presents the fitted helices for the simulated straight and bent helices. When there is change in axis direction, the evidence above shows that Bending-Detector works well to detect whether a given helix is bent or not and, if it is bent, to find the position of the change point and the change in helix axis.

TABLE 5.3: The Bending-Detector estimates $\hat{\sigma}^2$, $\hat{\sigma}^2_{\mathrm{p}}$, $\hat{k}$, $\hat{\theta}$, the statistics $F_{\max}, A_1, \ldots, A_6$, and the bootstrap of $n_{boots} = 1000$ p-values (in bracket) of the simulated regular and bent helices.

Helix	regular	bent
$\hat{\sigma}^2$	0.039	0.223
$\hat{\sigma}^2_{\mathrm{p}}$	0.035	0.045
$F_{\max}$	0.924 (0.973)	26.3^{**}
$\hat{k}$	-	12
$\hat{\theta}$	-	0.32r
A_1	1×10^{-4} (0.765)	0.051^{**}
A_2	0.134 (0.455)	0.146 (0.447)
A_3	0.004 (0.981)	0.012 (0.954)
A_4	0.002 (0.965)	0.048 (0.392)
A_5	0.046 (0.666)	0.183 (0.125)
A_6	0.002 (0.919)	0.010 (0.724)

** indicates p-value < 0.001.

5.4.1 Residual plots for simulated datasets

In this subsection, we look at the three coordinates residual plots of the simulated regular and bent helices, which are studied in Section 5.4. We also look at the radial and tangential residual plots.

In Section 3.9 we studied how to rotate the residual plots clockwise about the z-axis, to investigate for any indication of a bend. In order to do that, we fit two quadratic functions to each of the x and y residuals plots against time using least

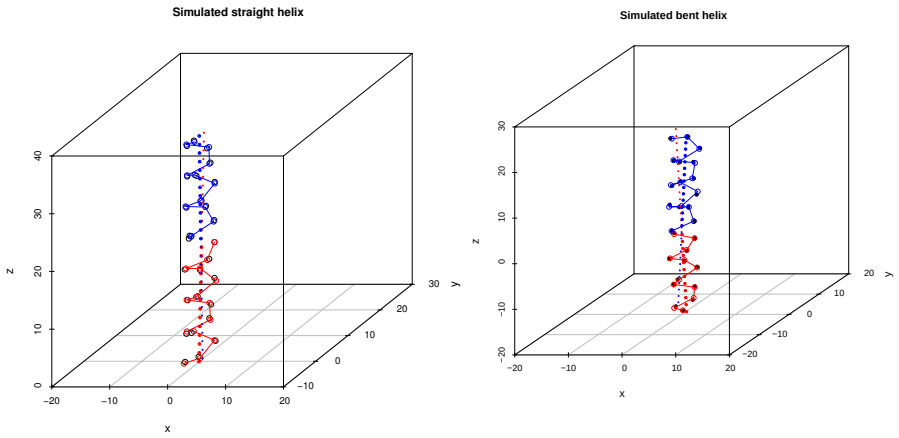

FIGURE 5.7: The fitted helix for straight helix and the two fitted sub-helices for bent helix.

squares method. The changes will be in the x and y residuals plots only, such that one shows random noise, since the quadratic coefficient will become 0, and the other vividly displays a quadratic behaviour. Figures 5.8 and 5.9 present the residuals plots for the simulated bent and regular helices, respectively. Figure 5.8 vividly shows a quadratic behaviour (V-shape) in the x residuals plot in panel (a), and random pattern in the y and z residuals in panels (b) and (c), respectively, which suggest the helix could be bent. On the other hand, the residuals plots in Figure 5.9 show fairly random pattern, which suggest the helix could be regular.

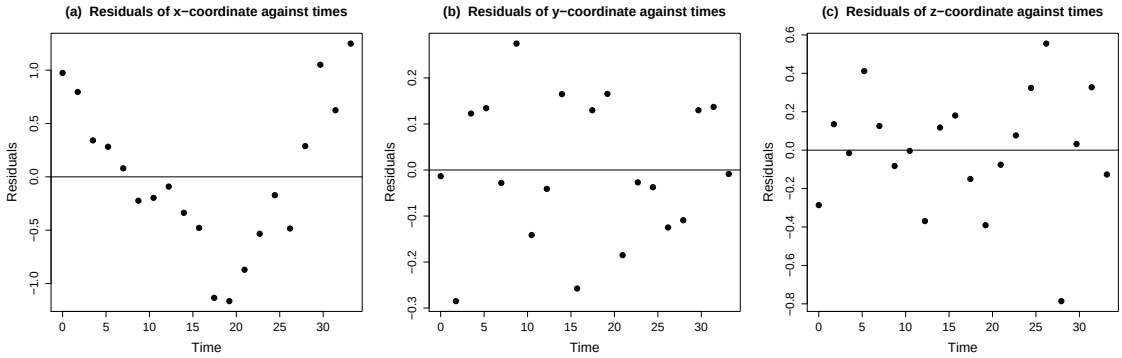

FIGURE 5.8: Simulated bent helix three coordinates residuals against time after rotation about the axis.

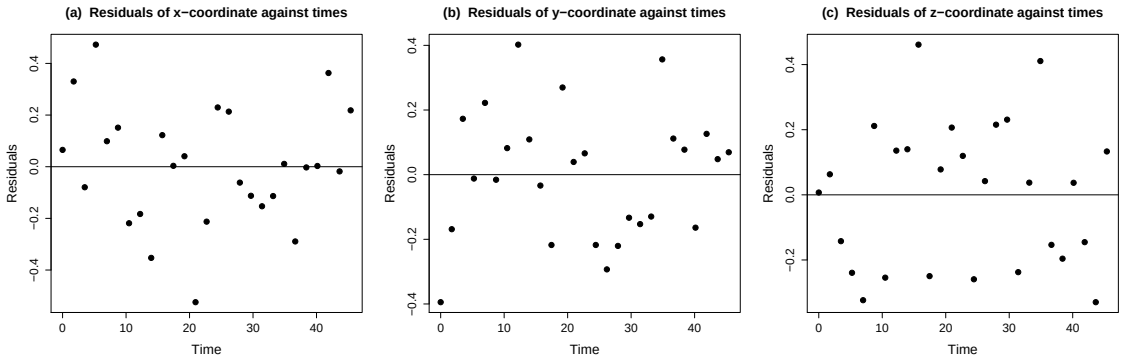

FIGURE 5.9: Simulated regular helix three coordinates residuals against time after rotation about the axis.

To investigate further the relationship between the observed data and the fitted helix, we plot the radial and the tangential residuals of the fitted points to the simulated points. First we need to explain how to determine the tangent residual plot. Recall that $\boldsymbol{y}_i$ is the data point, and $\hat{\boldsymbol{y}}_i$ is the fitted point for $i = n_1, \ldots, n_2$, $n_1 = 1, n_2 = 27$. Compute the angles δ_i of the simulated point, and the angle $\hat{\delta}_i$ for the fitted point from the origin. If $\hat{\delta}_i > \delta_i$, i.e. $\hat{\boldsymbol{y}}_i$ is over-estimated, then the tangent of the fitted point is said to be to the left of the data point, otherwise it is said to be to the right. Let angle δ_{id} be the difference between the angles of the tangent lines of the fitted and simulated points, then

$$\begin{aligned} \delta_i &= \text{atan2}(y_{i2}, y_{i1}), \\ \hat{\delta}_i &= \text{atan2}(\hat{y}_{i2}, \hat{y}_{i1}), \\ \delta_{id} &= \delta_i - \hat{\delta}_i. \end{aligned}$$

If $\delta_{id} > 0$, then $\hat{\boldsymbol{y}}_i$ is under-estimated, otherwise it is over-estimated. If $\delta_{id} = 0$, then $\hat{\boldsymbol{y}}_i$ is a perfect fit.

Figure 5.10 shows 4 panels. Panel (a) presents the 2D-data simulated bent helix in black and the fitted helix by OptLS in red, which does not fit the data well. Panel (b) shows the estimated radius values are somewhat under-estimated and close to the true radius value. Panels (c) and (d) present the angle between

the tangent lines of the fitted and simulated points. The angle is the difference between the angles of the tangent lines of the fitted and simulated points. The tangent residuals in panels (c) and (d) are randomly dispersed. Figure 5.11, panel (a) presents red helix fits the black helix data well. Panel (b) shows almost half of the estimated radius values are under-estimated and half are over-estimated and close to the true radius value. The tangential residuals in panels (c) and (d) are randomly scattered.

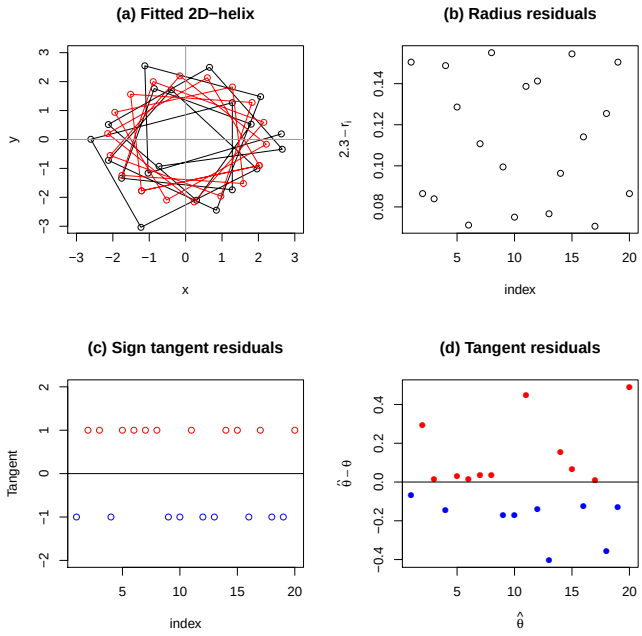

FIGURE 5.10: Illustration of the behaviour of the fitted points for the simulated bent helix (a) 2D-scatter plot presents the fitted helix in red does not fit well the bent simulated helix in black. (b) The radial plot presents the difference between the true radius and the radius of each fitted point. Both (c) and (d) present random positions of the angle between the tangent lines of the fitted and simulated points, which are either left (positive)/ right (negative) to that of the simulated points.

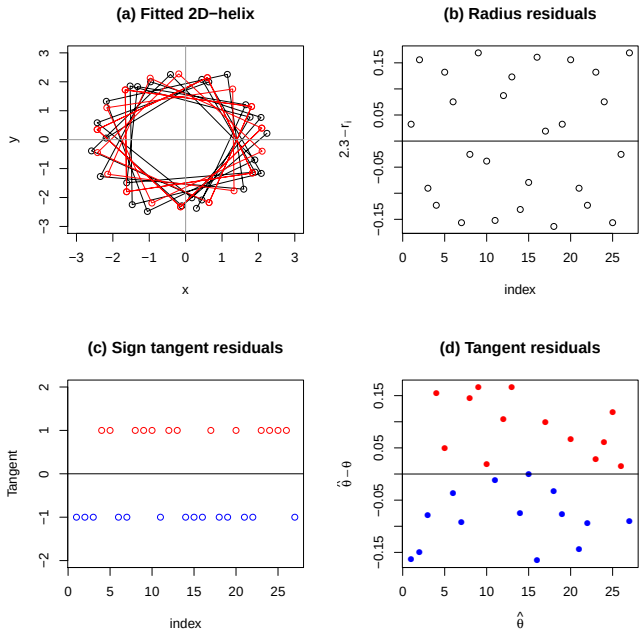

FIGURE 5.11: Illustration of the behaviour of the fitted points for the simulated regular helix (a) 2D-scatter plot presents the fitted helix in red fits well the data helix in black. (b) The radial plot presents the difference between the true radius and the radius of each fitted point. Both (c) and (d) present random positions of the angle between the tangent lines of the fitted and simulated points, which are either left (positive)/ right (negative) to that of the simulated points.

We can conclude that both the radial and the tangential residual plots for the simulated bent and regular helices give no difference i.e. no indication of a bent. On the other hand, the coordinates residual plots after rotation give good indication of a bent for the simulated bent helix as the x-coordinates residual plot shows a V-shape. Also the coordinates residual plots after rotation of the simulated regular helix show no pattern to the residuals.

To learn more about the residual plots of a bent helix, we create three mathematical bent helices. These mathematical helices mimic protein α-helix (i.e. $r = 2.3, 2\pi c = 5.4, \beta = \frac{2\pi}{3.6}$) with $n = 20$ landmarks. The bend is of $\theta = 0.3$ radians about the x-axis at $k = 8, 10$ and 12. We use the OptLS to fit these helices and then rotate the helices clockwise about the z-axis so that one of the x and y coordinates residual plots shows random noise and the other vividly displays

a quadratic behaviour as in Section 3.9. Figures 5.12, 5.13 and 5.14 present the three coordinates residual plots for each of the mathematical bent helices with $k = 8, 10$ and 12, respectively. In these Figures, the x-coordinate residual plot in panel (a) shows a V-shape. We can conclude that if the x-coordinate residual plot shows a V-shape, this suggests that the axis is bent.

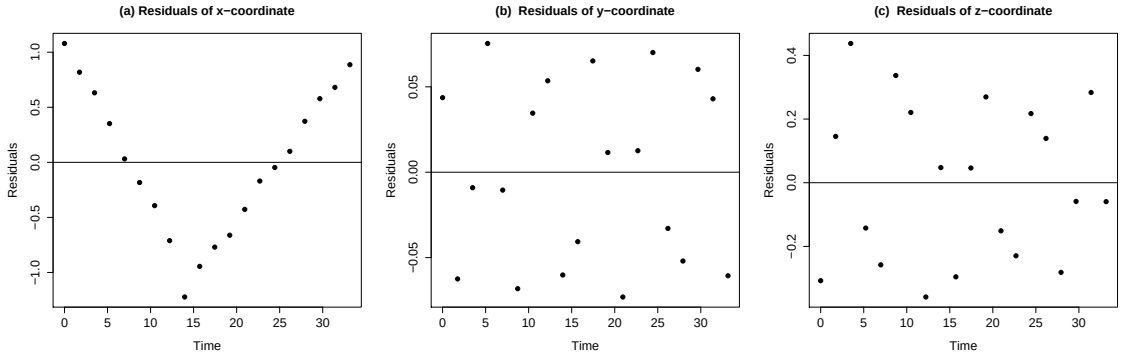

FIGURE 5.12: Simulated mathematical bent helix three coordinates residuals after rotation about the axis, where $k = 8$

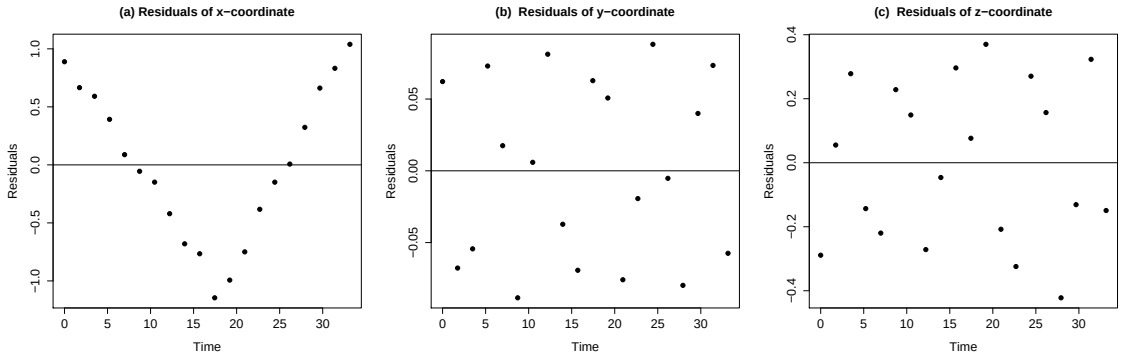

FIGURE 5.13: Simulated mathematical bent helix three coordinates residuals after rotation about the axis, where $k = 10$

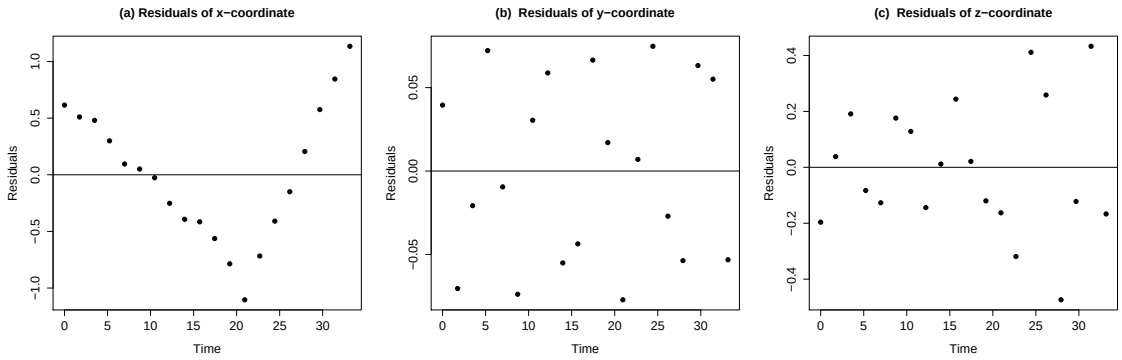

FIGURE 5.14: Simulated mathematical bent helix three coordinates residuals after rotation about the axis, where $k = 12$

5.5 Applications

In this section, we implement our Bending-Detector on nine α-helix datasets from Mardia et al. (2018) (see Appendix B). For the first seven of these nine helices, deciding if they were kinked proved to be difficult and Mardia et al. (2018) found different results to a crowdsourcing analysis carried out by experts in the protein field (Wilman et al., 2014a). However, for helices 8 and 9 the findings of Mardia et al. (2018) coincided with this crowdsourcing analysis (Wilman et al., 2014a). Recall that the regular real data helix eight, studied in Section 3.9, has V-shaped x-coordinate residuals, and we will investigate this behaviour more in this section. Note that the theoretical residuals variance σ^2 is 0.056 from Mardia et al. (2018).

Next, Table 5.4 contains: the values and the p-values for each statistic $F_{\max}, A_1, \ldots, A_6$; the residuals estimated variance $\hat{\sigma}^2$ under the null hypothesis and the pooled residual estimated variance $\hat{\sigma}_p^2$; the change point position $\hat{k}$; and the angle $\hat{\theta}$ between axes of the sub-helices. Table 5.4 shows that all the helices are bent, since all the p-values of $F^*_{\max}$ are highly significant (< 0.001); this is mainly due to change in axis statistic A_1 (p-value< 0.049). In addition to

significance in statistic A_1, some helices have significant in statistics A_3 (offset), A_4 (spin) and A_6 (pitch). However, statistics A_2 (shift) and A_5 (radius) are not significant in any helix suggests that these are not the reasons for the change point (bend). Moreover, the variance σ^2 obtained from fitting the single helix is greater than the pooled variance σ_p^2 obtained from fitting the two sub-helices, so we may conclude that the helices have a change point. The pooled variances σ_p^2 obtained from fitting the two sub-helices for helices 1, 3, 4, 5, 6, 7 and 9 are close to the theoretical variance $\sigma^2 = 0.056$ from Mardia et al. (2018), whereas the pooled variances is large for helix 2 ($\hat{\sigma}_p^2 = 0.144$) and small for helix 8 ($\hat{\sigma}_p^2 = 0.014$).

TABLE 5.4: Test statistics and estimates from Bending-Detector for helices $1, \ldots, 4$. The estimates of variance $\hat{\sigma}^2$, pooled variance estimate $\hat{\sigma}_{\mathrm{p}}^2$, the position $\hat{k}$, the angle $\hat{\theta}$ between the two sub-helices and the test statistics data (and p-values) of $F_{\max}, A_1, \ldots, A_6$ for each of the helices.

Helix	1	2	3	4
n	31	24	24	17
$\hat{\sigma}^2$	0.318	0.836	0.195	0.179
$\hat{\sigma}_{\mathrm{p}}^2$	0.083	0.144	0.060	0.034
$F_{\max}$	30.9 **	39.5 **	18.8 **	24.0 **
$\hat{k}$	14	7	9	10
$\hat{\theta}$	10.7°	25.6°	9.2°	8.8°
A_1	0.017**	0.098 **	0.013 **	0.012 **
A_2	0.005 (0.889)	0.454 (0.443)	0.076 (0.615)	0.012 (0.927)
A_3	1.983 **	1.518 (0.014)	0.343 (0.154)	0.880**
A_4	0.352 **	5.039 (0.014)	0.436 (0.008)	0.352 (0.002)
A_5	0.014 (0.930)	0.001 (0.998)	0.012 (0.917)	0.038 (0.737)
A_6	0.007 (0.803)	0.008 (0.847)	0.010 (0.730)	0.039 (0.141)

** indicates p-value < 0.001.

TABLE 5.4: Continued; Test statistics and estimates from Bending-Detector for helices $5, \ldots, 9$.

Helix	5	6	7	8	9
n	24	23	19	15	27
$\hat{\sigma}^2$	0.200	0.108	0.122	0.061	1.684
$\hat{\sigma}^2_{\mathrm{p}}$	0.065	0.062	0.045	0.014	0.102
$F_{\max}$	17.6 **	6.7 (0.002)	11.5 **	16.7 **	142.9**
$\hat{k}$	10	12	11	8	11
$\hat{\theta}^\circ$	6.6°	5°	12.6°	9.6°	31.9°
A_1	0.007 (0.008)	0.004 (0.049)	(0.024) **	0.014 **	0.151 **
A_2	0.191 (0.710)	1e-4 (0.988)	0.012 (0.975)	0.003 (0.978)	0.005 (0.946)
A_3	4.935 **	1.206 (0.002)	0.116 (0.572)	0.026 (0.597)	3.617 **
A_4	0.034 (0.599)	0.041 (0.519)	0.054 (0.342)	0.130 **	1.597 (0.013)
A_5	0.140 (0.322)	0.010 (0.964)	0.099 (0.410)	0.043 (0.566)	0.041 (0.813)
A_6	0.034 (0.267)	0.020 (0.514)	0.054 (0.052)	0.042 (0.025)	0.028 (0.409)

** indicates p-value < 0.001.

Helix 8 has $\sigma^2 = 0.061$ which is close to the theoretical variance 0.056 (see Mardia et al. (2018)), and it is classified as bent by Bending-Detector but as unkinked by Kink-Detector. Figure 5.15 presents the data and the fitted helices for helix 8, and the axis is clearly bent as the direction of the $H^{(1)}$ axis (in red) is different than the direction of the $H^{(2)}$ axis (in blue). In addition, we plot the statistics $F_{\max}, A_1, \ldots, A_6$ in Figures 5.16 and 5.17 respectively. Figure 5.16 shows a highly significant $F_{\max}$ ($F_{\max} \gg F_{\max}^{\alpha=0.05}$), then we can conclude that the helix is bent which is due to differences in axis A_1, spin angle A_4, and difference in pitch A_6 between the two sub-helices.

Further, Table 5.5 presents a comparison between our Bending-Detector and the Kink-Detector by Mardia et al. (2018) using the same nine data helices. From Table 5.5, we conclude that both methods categorised helices 1, 2 and 8

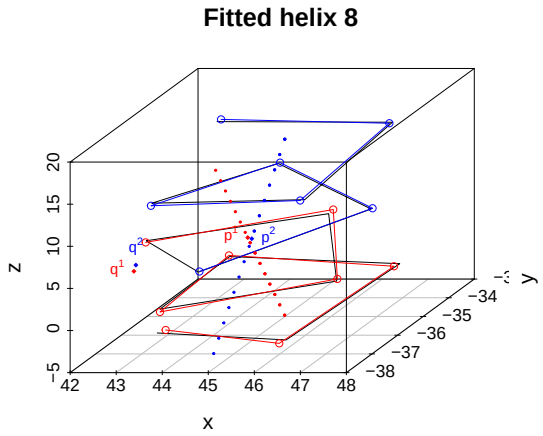

FIGURE 5.15: The data helix 8 in black and the fitted each sub-helices $H^{(\ell)}$ and the points $p^{(\ell)}, q^{(\ell)}$, $\ell = 1$ in red and $\ell = 2$ blue.

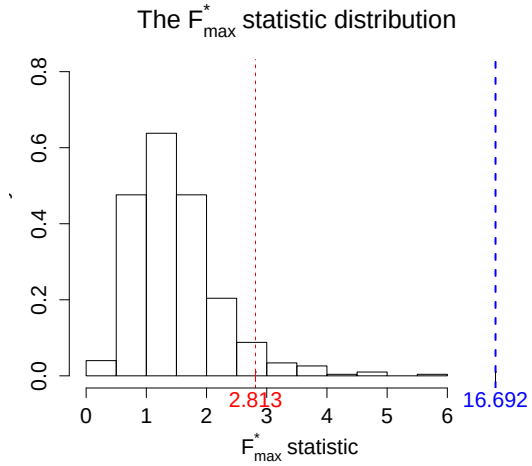

FIGURE 5.16: Helix 8 has a bend since the data statistic $F_{\max} = 16.692$ is greater than the threshold $F_{\max}^{*(\alpha=0.05)} = 2.813$.

differently, and the estimated change point locations are always quite different. This seems to be due to the difference in the way the change points are estimated. Kink-Detector is looking for a local change in the axes as it uses a moving window of 12 landmarks, whereas our method looks for a global change; the method use all the n points on the helix. Furthermore, Kink-Detector is a technique built for a protein α-helix as it looks for a region of four points, whereas we look for one critical change point so the bending is somewhat sharper. In addition, Kink-Detector estimates δ, whereas we assumed $\delta = \frac{2\pi}{3.6}$

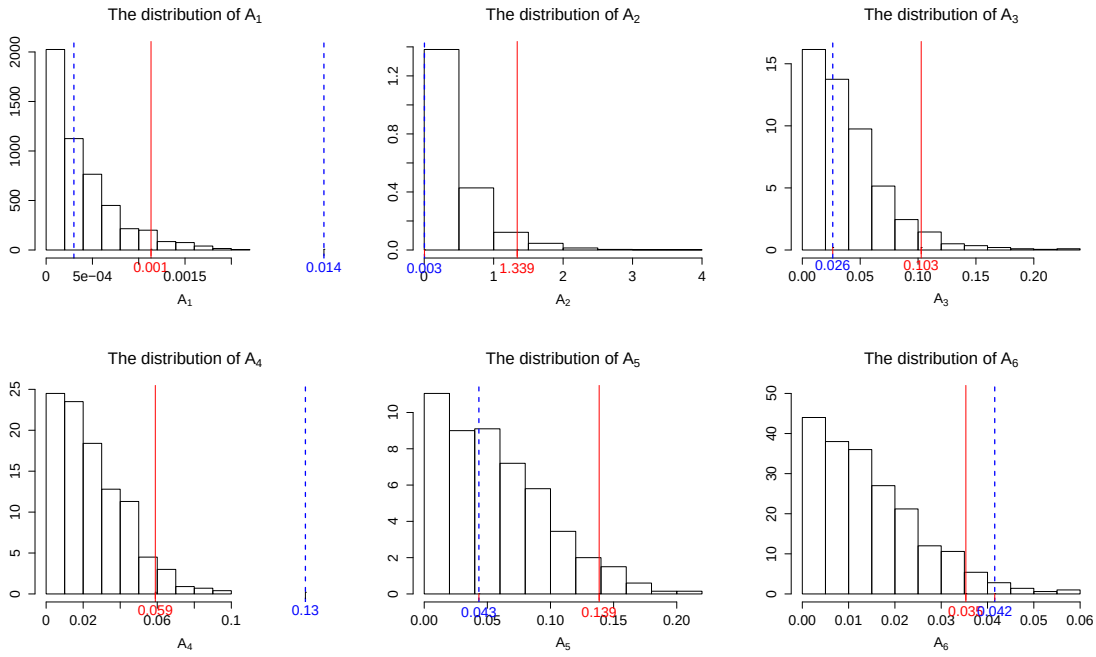

FIGURE 5.17: Helix 8 statistics $A_1, \ldots, A_6$ distributions, where the threshold is in blue and data statistics are in red.

TABLE 5.5: The kink position $\hat{k}$, the angle between the two sub-helices $\hat{\theta}$ in degrees, and the classification by Kink-Detector ("k"= kinked, "s"= straight), and the classification by Bending-Detector ("b"= bend, "r"= regular).

Helix	**Kink-Detector**			**Bending-Detector**			
	$\hat{k}$	$\hat{\theta}°$	classification	$\hat{k}$	$\hat{\theta}°$	classification	$F_{\max}$
1	–	–	s	14	10.7°	b	30.9
2	–	–	s	7	25.6°	b	39.5
3	13	18.7°	k	9	9.2°	b	18.8
4	7	15.9°	k	10	8.8°	b	24.0
5	7	22.8°	k	10	6.6°	b	17.6
6	10	20.4°	k	12	5.0°	b	6.7
7	13	20.0°	k	11	12.6°	b	11.5
8	–	–	s	8	9.6°	b	16.7
9	9	30.5°	k	11	31.9°	b	142.9

5.5.1 Residual plots for real datasets

We look at the residual plots against time for all the nine helices from fitting these helices using OptLS (fitting a single helix). We expect that a bent helix

will have a V-shaped residual plot for one of the coordinates residuals. For each of the nine helices, x-coordinates residual plot shows a V-shaped. In this section, we focus on studying helices 1, 2 and 8, since they are categorised as bent helices by Bending-Detector and unkinked by Kink-Detector.

In order to test the residual plots to check for any indication of a bend, we first test the null hypothesis H_0 : 'all of the quadratic coefficients are zero', i.e. the helix has no bend; for more details see Section 3.9. We fit two quadratic functions to each of the x and y residual plots against time using least squares with quadratic coefficients a_2, b_2, see equations (3.10) and (3.11).

Table 5.6 presents the estimates of the quadratic coefficients $\hat{a}_2, \hat{b}_2$ before the rotation and the quadratic coefficients a_2^*, b_2^* after rotation about the z-axis.

TABLE 5.6: The estimates of quadratic coefficients before and after rotation from the quadratic fit of the x and y residual plots for helices 1, 2 and 7.

Helix	coordinates	$\hat{a}_2$	a_2^*	$\hat{b}_2$	b_2^*
1	x	-0.112	-0.128	0.002	0.002
	y	0.061	-5×10^{-4}	-0.001	0
2	x	0.135	-0.149	-0.003	0.004
	y	0.064	-7×10^{-4}	-0.002	0
8	x	-0.059	-0.135	0.002	0.005
	y	-0.121	-1×10^{-3}	0.005	0

Figures 5.18, 5.19, and 5.20 present the three coordinates residual plots for helices 1, 2, and 8 respectively. These Figures are clearly show V-shape pattern in the x-coordinate residual plots, which indicate that these are bent helices. In addition, Figure 5.19 also shows a V-shape pattern in the z-coordinate residual plot, which suggests some other behaviour.

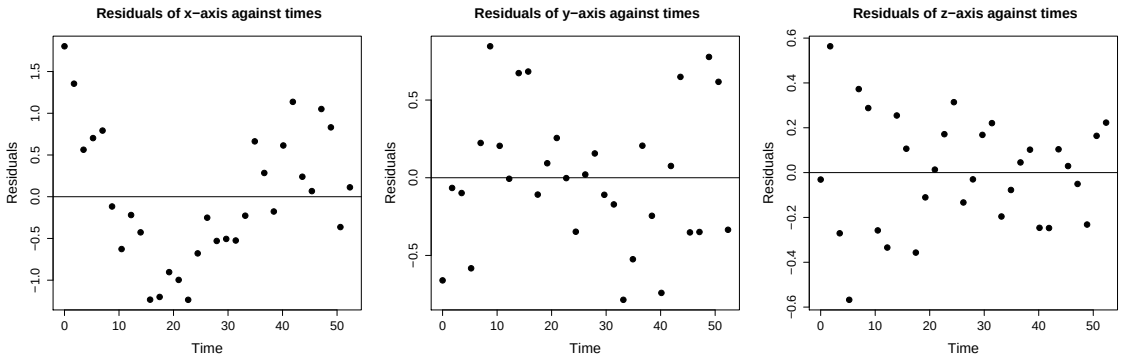

FIGURE 5.18: helix 1 three coordinates residuals against time after rotation.

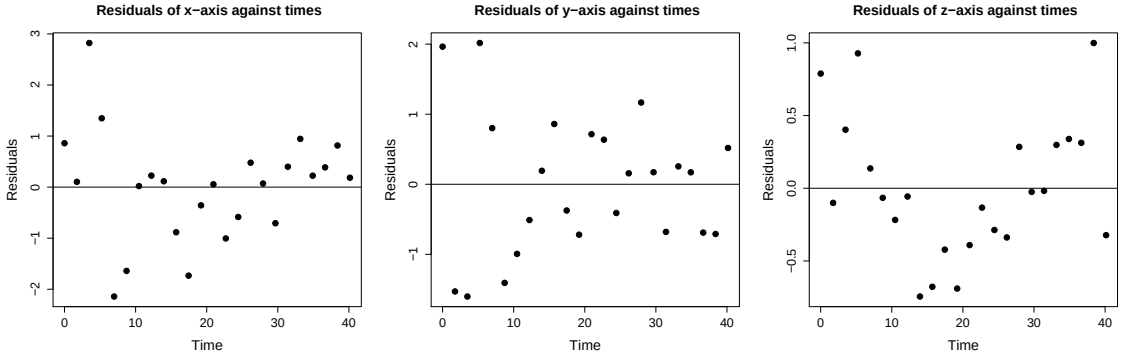

FIGURE 5.19: Helix 2 three coordinates residuals against time after rotation.

5.6 Alternative methods for analysing helices

As discussed earlier, a protein helix is allowed to have some bend, which is not classified as a kink. From Section 5.5, there are four reasons that Bending-Detector is distinct from Kink-Detector. Recall that Kink-Detector has four assumptions: (a) treat δ as unknown; (b) look for a change point in a moving window of 12; (c) look for a change point as a region of 4 points; and (d) choose 4 points where $\cos\hat{\theta} \geq 0.98$. In this section, we present a brief summary of altering Bending-Detector and, in particular, the way in which we determine a threshold under Kink-Detector assumptions. We will see that this procedure, which we call the $6-6$ method (see Section 5.6.1), will alter Bending-Detector to look for a change point in a moving window of 12. In Section 5.6.2 we permit a helix to

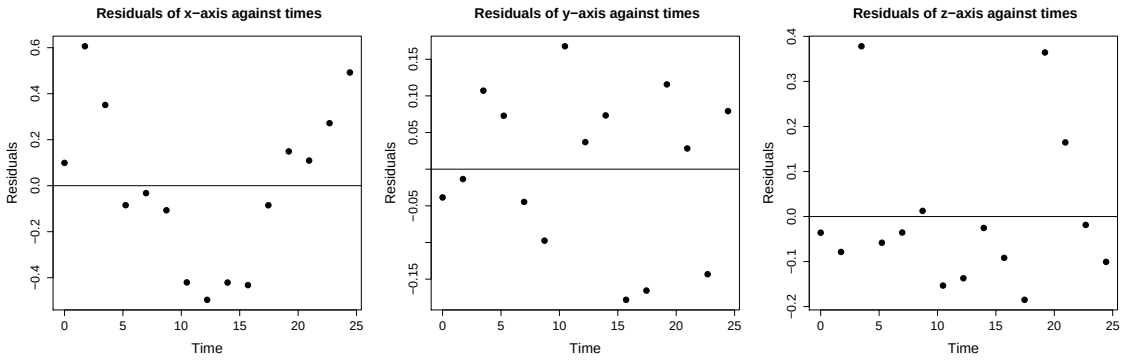

FIGURE 5.20: Helix 8 three coordinates residuals against time after rotation.

have some curvature in its axis but not be categorised as kinked i.e. to look for a change point as a region of 4 points. Finally, in Section 5.6.3, we alter Bending-Detector to look for a change point as a region of 4 points with $\cos\hat{\theta} \geq 0.98$ in a moving window of 12. We can conclude that under the assumption of Mardia et al. (2018), we get the same result as Mardia et al. (2018) Section 5.6.3.

5.6.1 The $6-6$ method

The idea behind the $6-6$ method is somewhat adopted from Mardia et al. (2018). Recall that Bending-Detector assume each of the points $n_1+5, \ldots, n_2-6$ is a potential change point, then make an appropriate cut to obtain two sub-helices. However, this time instead of including all n landmarks amongst the two sub-helices in each step, we will focus on just a small neighbourhood of a potential kink, extending six atoms in either direction i.e. 12 in total.

TABLE 5.7: The change point position, the $F_{\max}^{*(0.05)}$, the angle between two axes for the helices 1, 2 and 8.

Helix	Change point	$F_{\max}^{*(0.05)}$ statistic	$\hat{\theta}$
1	7	15.27	15.7°
2	11	8.03	13.5°
8	12	10.77	13.4°

Table 5.7 above shows that helices 1, 2 and 8 are still classified as bent since their p-value of $F_{\max}^{*(0.05)}$ are still < 0.001. Furthermore, the bent angle $\hat{\theta} \geq 13.4°$, between the two axes, is large enough to say that it is bent, whereas in Mardia et al. (2018) these helices are categorised as unkinked. In addition to other helices $3, \ldots, 7$ and 9 have $F_{\max}^{*(0.05)}$ p-value of < 0.001, which implies that all the helices have a change point.

5.6.2 The kink region method

A kink in protein α-helix is not a single point but a small region of points, we edit the Bending-Detector in order to reflect this feature. This in turn alters the threshold value.

We change the method in the following way. We now assume the point k is a change point, where $n_1 + 7 \leq k \leq n_2 - 8$, we remove points $k-1, \ldots, k+2$ and this naturally yields two sub-helices. So, to calculate the $F_{\max}^{*(0.05)}$ statistics, we must also remove these same 4 points when using the OptLS method. The p-value of $F_{\max}^{*(0.05)}$ statistics for all of the nine helices are still < 0.001, which imply that all the helices have a change point.

5.6.3 Kink-Detector

Moreover to allow helix bending we look at a kink position as a region of 4 atoms which has $\cos(\hat{\theta}) \geq 0.98$ as in Mardia et al. (2018). This changes the result in Table 5.7, so that the helices 1, 2, and 8 are classified here as unkinked as we can see in Figures 5.21, 5.22, and 5.23. This is the same result as in Mardia et al. (2018).

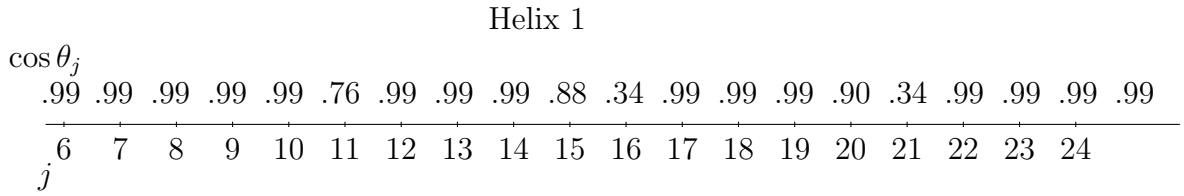

FIGURE 5.21: Change point possibility from landmark 6 to 24 on helix 1.

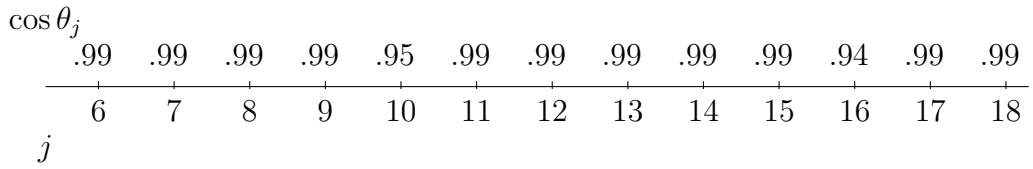

FIGURE 5.22: Change point possibility from landmark 6 to 18 on helix 2.

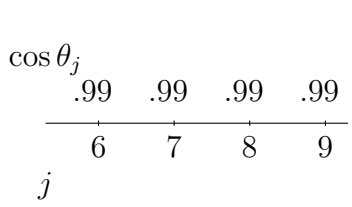

FIGURE 5.23: Change point possibility from landmark 6 to 9 on helix 8.

Chapter 6

Conclusions and further work

In this chapter we summarise the main work in this thesis and discuss some potential further lines of possible extensions. In particular, we discuss the extensions needed in order to extend our Bending-Detector. We can relax the assumption of the helix model, thus increasing the number of parameters to estimate. In addition, we can use our knowledge of the helix shape analysis to study more examples of helices in real life.

The two main chapters of this thesis are Chapter 3 and Chapter 5. In Chapter 3 we investigated methods to estimate the regular helix axis, and developed a new algorithm to fit a helix without a change point, OptLS, for the maximum likelihood estimation (MLE). In Chapter 5 we develop a new algorithm, Bending-Detector, to fit a helix with change points and used it to find the change points in helix structures and to investigate the properties of the change points. In addition, there is a chapter on M-H model, Chapter 4, as a possible way to estimate a regular helix without a change point, but not really a successful one. Note that Bending-Detector is applicable for any equally spaced 3-dimensional helix.

In this thesis we looked for at most one kink on a helix but we can look for more than one. Therefore, a natural generalisation can be done from the Bending-Detector strategy described in Section 5.1.1. We apply the method in

Section 5.1.1 to a helix H and, consequently, we have two sub-helices $H_k^{(1)}$ and $H_k^{(2)}$. If it is known that $H_k^{(\ell)}, \ell = 1, 2$, is bent then we can repeat the steps in Section 5.1.1 on $H_k^{(\ell)}$.

Recall that a regular helix is defined here as a set of a points that are equally spaced in a statistical right circular helix. It is regarded as a function of an independent variable t, where the observations are subject to independent isotropic normally distributed errors with mean 0 and variance $\sigma^2 I_3$. As discussed in Section 1.2.4, points are located at time $t_i = (i-1)\delta, i = n_1, \ldots, n_2$, $n_1, n_2 \in \mathbb{N}$, with a constant angle $\delta = \frac{2\pi}{3.6}$ radians between points around the helix curve. To fit this helix, we developed the optimization algorithm OptLS which estimates the axis $\boldsymbol{w}$, the radius r and the pitch c (the algorithm described in Chapter 3). We also investigated the accuracy of this method by comparing it with three other previously studied methods summarized in Christopher et al. (1996). From this comparison we found that OptLS has the smallest sample variance of the helix axis estimate. In addition, by simulation of protein α-helices, we found that the estimates of the parameters r and c are very close to the known values for a protein α-helix $r = 2.3$ and $c = \frac{5.4}{2\pi}$ (see Mardia et al., 2018). It would have been interesting if we allowed the points to be non-equally spaced as in Mardia et al. (2018). Then the optimization least squares approach, OptLS, which maximizes the likelihood estimation of unknown parameters would need to be extended to estimate the unknown parameters t_i or basically the angle between the adjacent points.

Another more realistic extension to the model is changing the assumption of isotropic errors. We assumed that the errors are independent within and between the three orientations of the right circular helix $\boldsymbol{y}(t_i) := (y_{i1}, y_{i2}, y_{i3})^T$ and the variance-covariance matrix of size $3n \times 3n$ is assumed to be $\Sigma = \sigma^2 I_3$. On the other hand, in practice, errors could be different for each orientation. Let S_1^2, S_2^2, S_3^2 be variance-covariance matrices of size $n \times n$ of the three coordinates y_{i1}, y_{i2}, y_{i3},

respectively, then the variance-covariance matrix Σ can be written as

$$\Sigma = \begin{bmatrix} S_1^2 & 0 & 0 \\ 0 & S_2^2 & 0 \\ 0 & 0 & S_3^2 \end{bmatrix},$$

One can also suggest that the errors for each orientation are correlated, then covariances between random variables y_{i1}, y_{i2}, and y_{i3} are not equal to zero. Let $\text{cov}(y_{ij}, y_{ik}) = S_{jk}, j = 1, 2, 3, k = 1, 2, 3, j \neq k$ be the variance-covariance matrix of size $n \times n$ between the j^{th} and k^{th} directions, then the variance-covariance matrix can be written as

$$\Sigma = \begin{bmatrix} S_1^2 & S_{12} & S_{13} \\ S_{12} & S_2^2 & S_{23} \\ S_{13} & S_{23} & S_3^2 \end{bmatrix}.$$

A more general case is when the normal and tangential errors at each point on the helix curve are different. Recall the helix equation in (1.3), and let $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$ be the normal and tangential components, respectively, for all i as follows

$$\boldsymbol{u}_i = \begin{bmatrix} \cos(i-1)\delta \\ \sin(i-1)\delta \\ 0 \end{bmatrix}, \qquad \boldsymbol{v}_i = \begin{bmatrix} -\sin(i-1)\delta \\ \cos(i-1)\delta \\ 0 \end{bmatrix}.$$

The error in the vertical direction is independent from the errors in the horizontal direction (xy-plane), so that the third column of Σ is $[0, 0, \gamma]^T$ with magnitude γ. In the plane the variance tangent to the helix is assumed to be different to the vector perpendicular to the helix. The variance-covariance matrix for variances

a, b of the normal and tangential errors, respectively, is

$$\Sigma = a\boldsymbol{u}_i\boldsymbol{u}_i^T + b\boldsymbol{v}_i\boldsymbol{v}_i^T + \gamma \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
$$= \frac{a+b}{2} I_3 + \frac{a-b}{2} \begin{bmatrix} \cos 2(i-1)\delta & \sin 2(i-1)\delta & 0 \\ \sin 2(i-1)\delta & -\cos 2(i-1)\delta & 0 \\ 0 & 0 & 0 \end{bmatrix} + \gamma \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

In this thesis, we have also developed the Bending-Detector procedure which: (1) tests if the helix has a change point; (2) estimates the position of this change point; (3) cuts the helix into two regular helices by this point and uses OptLS to fit each sub-helix separately; and (4) investigates the reason for this change point i.e. tests the six features. These six test statistics test how the two sub-helices can differ. The two sub-helices can differ by a shift along the helix axis; an offset perpendicular to the helix axis; a spin parameter; the helix radius; the helix pitch; and the helix axis direction. We can investigate further the properties of these test statistics. Such as simulate kinked helices where each time we made a differ one of the six feature and so we can study these features. In addition, for a bent helix we expect the x-coordinates residual plot has a V-shape, see Section 5.4.1. The x-coordinates residual plot of real dataset helix 2 has a V-shape (see Section 5.5.1, which suggest some other behaviour need to investigate.

The helix is a configuration of points in 3-dimensional space, that we can statistically analyse the shape properties invariant under location and rotation effects. Shape analysis is the statistical study of the geometrical information of an object which remains the same under location, scaling, rotation and reflection. However, in this thesis we are not interested in the scale and reflection effects because these kind of effects would change the sense and size of the helix. For example, the radius of an α-helix is a critical parameter as it is universal across proteins, so any change of the helix that would be affect this cannot be allowed.

Thus, we cannot scale the helix. Similarly, a reflection would change the helix axis direction which we also cannot permit. However, in other applications of shape analysis scale and reflection change points might allowed. We can use our knowledge to explore different settings of shape analysis.

One possible toy example might be a *slinky* which is a helix but not as rigid as a protein α-helix, as we can stretch it to a long helix or compress it to a short helix or even bend it. A real life example is DNA. DNA consists of two right handed helices joined together but it can bend, i.e. the DNA analogy has some similarity to a slinky.

Another toy example of a helix is the spiral staircase in old castles. A circular spiral staircase is a circular regular helix where the steps spiral around a central pole. The radius of the staircase is the length of a step from central pole to the edge of the staircase circle. The size and the depths of each step are almost equal, so that we can say spiral staircase points are almost equally spaced around a regular helix. We can investigate the irregularity in stairs. The main difference between our helix model in (1.3) and spiral staircase is the visibility of the central pole which supports the steps.

Appendix A

Basic proofs

This appendix records some elemantary properties of trigonometric functions used in our thesis.

Lemma A.1.

1. $\int_0^{2\pi} \sin t \, \mathrm{d}t = \int_0^{2\pi} \cos t \, \mathrm{d}t = 0$.
2. $\int_0^{2\pi} \sin t \cos t \, \mathrm{d}t = 0$.
3. $\int_0^{2\pi} \sin^2 t \, \mathrm{d}t = \int_0^{2\pi} \cos^2 t \, \mathrm{d}t = \frac{2\pi}{2}$.

Proof. 1. The integral of $\sin t$ is

$$\begin{aligned} \int_0^{2\pi} \sin t \, \mathrm{d}t &= -\cos t \Big|_0^{2\pi} \\ &= 0 \end{aligned} \tag{A.1}$$

and proof that $\int_0^{2\pi} \cos t \, \mathrm{d}t = 0$ similar.

2. The integral of the producot of $\sin t$ and $\cos t$ is

$$\begin{aligned}\int_0^{2\pi} \sin t \cos t \, \mathrm{d}t &= \frac{1}{2}\int_0^{2\pi} \sin 2t \, \mathrm{d}t \\ &= \frac{1}{4}\int_0^{2\pi} \sin u \, \mathrm{d}u \\ &= -\frac{1}{4}\cos 2t\Big|_0^{2\pi} \\ &= 0. \end{aligned} \tag{A.2}$$

3. The integral of $\sin^2 t$ is

$$\begin{aligned}\frac{1}{2\pi}\int_0^{2\pi} \sin^2 t \, \mathrm{d}t &= \frac{1}{2\pi}\int_0^{2\pi} \frac{1-\cos 2t}{2} \, \mathrm{d}t \\ &= \frac{1}{2\pi}\int_0^{2\pi} \frac{1}{2} \, \mathrm{d}t - \frac{1}{2\pi}\int_0^{2\pi} \frac{\cos 2t}{2} \, \mathrm{d}t \\ &= \frac{1t}{2\pi}\Big|_0^{2\pi} - \frac{1}{8\pi}\int_0^{2\pi} \cos u \, \mathrm{d}u \\ &= \frac{1}{2} - \frac{1}{8\pi}\sin 2t\Big|_0^{2\pi} \\ &= \frac{1}{2}. \end{aligned} \tag{A.3}$$

Similarly, we can derive $\int_0^{2\pi} \cos^2 t \, \mathrm{d}t = \frac{2\pi}{2}$.

□

For n points around the circle equally spaced (angles), we claim

Lemma A.2.

1. $\sum \sin \frac{2\pi j}{n} = \sum \cos \frac{2\pi j}{n} = 0.$
2. $\sum \sin \frac{2\pi j}{n} \cos \frac{2\pi j}{n} = 0.$
3. $\sum \cos^2 \frac{2\pi j}{n} = \frac{n}{2} = \sum \sin^2 \frac{2\pi j}{n}.$

Proof.

1. Let $x = \exp(\frac{2\pi i}{n}) = \cos\frac{2\pi}{n} + i\sin\frac{2\pi}{n}, i = \sqrt{-1}$, and note

$$x^j = \exp(\frac{2\pi ij}{n}) = \cos\frac{2\pi j}{n} + i\sin\frac{2\pi j}{n}.$$

   Recall the geometric series formula for the sum of x^j is

$$\sum_{j=0}^{n-1} x^j = \frac{1-x^n}{1-x},$$

   Since $x^n = e^{\frac{2\pi in}{n}} = 1$,

$$\begin{aligned}\sum_{j=0}^{n-1}(e^{\frac{2\pi i}{n}})^j &= \frac{1-x^n}{1-x}\\ &= \sum_{j=0}^{n-1}(\cos\frac{2\pi j}{n} + i\sin\frac{2\pi j}{n})\\ &= 0.\end{aligned}$$

   Hence $\sum_{j=0}^{n-1}\cos\frac{2\pi j}{n} = \sum_{j=0}^{n-1}\sin\frac{2\pi j}{n} = 0$.

2. Recall the geometric series formula for the sum of $(x^2)^j$ is

$$\sum_{j=0}^{n-1}(x^2)^j = \frac{1-(x^2)^n}{1-x^2},$$

Since $(x^2)^n = e^{\frac{4\pi in}{n}} = 1$,

$$\begin{aligned}\sum_{j=0}^{n-1}(e^{\frac{4\pi i}{n}})^j &= \frac{1-(x^2)^n}{1-x^2}\\ &= \sum_{j=0}^{n-1}(\cos\frac{4\pi j}{n} + i\sin\frac{4\pi j}{n})\\ &= 0.\end{aligned}$$

Hence $\sum_{j=0}^{n-1}\sin\frac{4\pi j}{n} = 0$, then

$$\begin{aligned}\sum_{j=0}^{n-1}\sin\frac{2\pi j}{n}\cos\frac{2\pi j}{n} &= \frac{1}{2}\sum_{j=0}^{n-1}\sin\frac{4\pi j}{n}\\ &= 0\end{aligned}$$

3. From 1,

$$\sum_{j=0}^{n-1}\cos^2\frac{2\pi j}{n} = \sum_{j=0}^{n-1}\frac{1-\cos\frac{2\pi j}{n}}{2} = \frac{n}{2}.$$
$$\sum_{j=0}^{n-1}\sin^2\frac{2\pi j}{n} = \sum_{j=0}^{n-1}\frac{1+\cos\frac{2\pi j}{n}}{2} = \frac{n}{2}.$$

□

In real protein α-helix, $t_j = \frac{2\pi}{3.6}(j-1)$, $j = 1,\ldots,n$, the points are balanced approximately around the circle, so that $\sum\cos^2 t_j \approx \frac{n}{2} \approx \sum\sin^2 t_j$ and $\sum\cos t_j \sin t_j \approx 0$. Table A.1 show that $\sum\cos t_j \approx 0 \approx \sum\sin t_j$, $\frac{1}{n}\sum\cos^2 t_j \approx \frac{1}{2} \approx \frac{1}{n}\sum\sin^2 t_j$ and $\sum\cos t_i \sin t_j \approx 0$ for various n arising in real protein α-helix. The typical α-helix is with $n = 11$ (see Mardia (2014)), and from our research the range of n is from 11 to 31.

TABLE A.1: For various n, $\frac{1}{n}\sum \cos t_j \approx 0 \approx \frac{1}{n}\sum \sin t_j$, $\frac{1}{n}\sum \cos^2 t_j \approx \frac{1}{2} \approx \frac{1}{n}\sum \sin^2 t_j$ and $\frac{1}{n}\sum \cos t_i \sin t_j \approx 0$.

n	$\frac{1}{n}\sum \sin t_j$	$\frac{1}{n}\sum \cos t_j$	$\frac{1}{n}\sum \sin^2 t_j$	$\frac{1}{n}\sum \cos^2 t_j$	$\frac{1}{n}\sum \cos t_j \sin t_j$
11	-0.013	0.016	0.497	0.503	-0.016
12	0.016	0.093	0.466	0.534	0.012
13	0.082	0.047	0.487	0.513	-0.022
14	0.030	-0.011	0.482	0.518	0.015
15	-0.015	0.041	0.478	0.522	-0.019
17	0.058	0.010	0.472	0.528	-0.010
18	0	0	0.500	0.500	0
20	0.049	0.041	0.498	0.501	-0.009
23	0.018	0.050	0.489	0.510	0.009
25	0.008	-0.005	0.497	0.503	0.006
27	0.031	0.037	0.500	0.500	0
28	0.030	0	0.482	0.518	0
30	0.007	0.037	0.486	0.514	0.005
31	0.034	0.020	0.495	0.505	-0.009

Note that unexpectedly several entries in Table A.1 are exactly 0 or $\frac{1}{2}$. The reason is as follows: Let $x = \frac{2\pi}{3.6}$, $t_j = jx$, and note $\sin\frac{x}{2} \neq 0$. An addition formulas for sines and cosines, (Gradshteyn and Ryzhik (2014), p. 30), states that

$$\sum_{j=0}^{n-1} \sin jx = \frac{\sin\frac{(n-1)x}{2}\sin\frac{nx}{2}}{\sin\frac{x}{2}}. \tag{A.4}$$

$$\sum_{j=0}^{n-1} \cos jx = \frac{\cos\frac{(n-1)x}{2}\sin\frac{nx}{2}}{\sin\frac{x}{2}}. \tag{A.5}$$

From these formulas we can deduce

$$\sum_{j=0}^{n-1} \cos jx \sin jx = \frac{1}{2} \sum_{j=0}^{n-1} \sin 2jx$$
$$= \frac{\sin (n-1)x \sin nx}{2 \sin x}. \tag{A.6}$$
$$\sum_{j=0}^{n-1} \sin^2 jx = \frac{1}{2} \sum_{j=0}^{n-1} (1 - \cos 2jx)$$
$$= \frac{n}{2} - \frac{\cos (n-1)x \sin nx}{2 \sin x}. \tag{A.7}$$
$$\sum_{j=0}^{n-1} \cos^2 jx = \frac{1}{2} \sum_{j=0}^{n-1} (1 + \cos 2jx)$$
$$= \frac{n}{2} + \frac{\cos (n-1)x \sin nx}{2 \sin x}. \tag{A.8}$$

In the first column we have 0 at $n = 18$ since $\sin nx = 0$ in equation (A.4). In the second column we have 0 at $n = 18, 28$ since $\sin nx = 0$ or $\cos \frac{(n-1)x}{2} = 0$ in equation (A.5). In the third and fourth column we have $\frac{1}{2}$ at $n = 18, 27$ since $\sin nx = 0$ in equations (A.7) and (A.8). In the fifth column we have 0 at $n = 18, 27, 28$ since $\sin nx = 0$ or $\sin (n-1)x = 0$ in equation (A.6). More generally, note that $\sin \frac{nx}{2}, \sin nx, \cos \frac{(n-1)x}{2}$, and $\sin (n-1)x$ equal 0 if either n or n-1 is a multiple of 9. Since $x = 100$ degrees then $\frac{9x}{2} = \frac{900}{2} = 450 = 360 + 90 = 90$ degrees.

Appendix B

Appendix B

In this appendix we present tables of 9 α-helix datasets that are employed in this thesis for various examples. All the 9 α-helix datasets are from Mardia et al. (2018).

TABLE B.1: Helix 1 dataset.

	y_1	y_2	y_3
1	62.45	289.05	168.93
2	59.00	289.89	170.33
3	58.24	286.13	170.32
4	61.44	285.67	172.32
5	60.29	288.26	174.90
6	56.89	286.57	175.27
7	58.54	283.29	176.24
8	60.95	285.11	178.57
9	58.03	286.93	180.23
10	56.09	283.68	180.70
11	59.20	282.29	182.38
12	59.61	285.28	184.69
13	55.90	285.64	185.47
14	55.79	282.04	186.68
15	58.51	282.90	189.19
16	56.61	286.06	190.21
17	53.42	284.00	190.78
18	55.27	281.28	192.65
19	57.12	283.62	195.04
20	54.19	285.96	195.72
21	51.81	283.05	196.40
22	54.47	281.32	198.53
23	54.76	284.52	200.64
24	50.96	284.84	200.90
25	50.78	281.19	201.92
26	53.49	281.49	204.57
27	51.82	284.60	206.03
28	48.60	282.60	206.41
29	50.55	279.65	207.88
30	52.22	281.83	210.56
31,	48.86	283.38	211.52

TABLE B.2: Helix 2 dataset.

	y_1	y_2	y_3
1	-2.34	-4.29	44.90
2	-2.49	-7.84	43.58
3	-4.99	-7.10	40.85
4	-2.45	-4.39	40.07
5	0.60	-6.68	39.90
6	-0.29	-9.67	37.72
7	-3.08	-7.97	35.78
8	-2.11	-4.32	35.07
9	1.64	-4.89	35.34
1	1.91	-8.04	33.21
11	-0.82	-7.15	30.70
12	0.71	-3.68	30.59
13	4.11	-5.02	29.55
14	2.33	-6.95	26.80
15	0.77	-3.74	25.54
16	4.20	-2.11	25.45
17	5.71	-4.79	23.21
18	2.60	-4.80	21.01
19	3.01	-1.08	20.30
20	6.62	-1.73	19.44
21	5.26	-4.26	16.94
22	3.09	-1.63	15.24
23	6.16	0.37	14.32
24	8.38	-2.72	14.30

TABLE B.3: Helix 3 dataset.

	y_1	y_2	y_3
1	12.30	29.47	26.39
2	15.22	27.77	24.67
3	12.79	25.68	22.67
4	11.28	28.95	21.35
5	14.65	30.22	20.17
6	15.55	26.88	18.67
7	12.24	26.63	16.85
8	12.45	30.15	15.47
9	16.11	29.88	14.72
10	15.91	26.46	12.97
11	12.91	27.46	10.93
12	14.75	30.41	9.52
13	17.39	28.57	7.48
14	14.88	25.83	6.83
15	12.60	28.27	5.04
16	15.57	29.80	3.43
17	16.83	26.41	2.29
18	13.51	25.84	0.50
19	13.56	29.23	-1.16
20	17.25	28.91	-2.08
21	16.42	25.56	-3.60
22	13.49	27.07	-5.38
23	15.76	29.67	-6.90
24	18.39	27.07	-7.70

TABLE B.4: Helix 4 dataset.

	y_1	y_2	y_3
1	87.66	1.81	-39.39
2	89.46	5.09	-38.75
3	86.19	6.94	-38.22
4	84.94	4.18	-35.87
5	88.11	4.49	-33.84
6	87.72	8.29	-33.67
7	84.14	8.22	-32.40
8	85.14	5.73	-29.68
9	88.14	7.83	-28.77
10	86.20	11.08	-28.28
11	83.98	9.27	-25.74
12	86.84	7.43	-24.02
13	89.07	10.51	-23.59
14	86.24	12.84	-22.58
15	84.70	10.17	-20.33
16	88.11	10.11	-18.54
17	88.37	13.92	-18.56

TABLE B.5: Helix 5 dataset.

	y_1	y_2	y_3
1	114.24	-8.43	3.17
2	111.65	-7.60	0.52
3	110.88	-4.22	2.10
4	114.58	-3.43	2.54
5	114.91	-3.75	-1.23
6	112.08	-1.28	-1.76
7	113.53	1.32	0.61
8	116.89	0.84	-1.11
9	115.59	1.61	-4.58
10	112.99	4.23	-3.59
11	114.72	6.32	-0.91
12	117.49	7.35	-3.33
13	114.82	9.24	-5.29
14	115.11	12.05	-2.74
15	118.80	12.37	-3.47
16	118.18	12.59	-7.22
17	115.64	15.40	-6.99
18	117.67	17.21	-4.32
19	120.48	17.24	-6.90
20	118.41	18.37	-9.88
21	117.32	21.38	-7.84
22	120.94	22.28	-7.09
23	121.69	22.09	-10.82
24	119.38	25.07	-11.30

TABLE B.6: Helix 6 dataset.

	y_1	y_2	y_3
1	-13.82	-1.72	-11.22
2	-17.70	-1.62	-11.33
3	-17.43	2.22	-11.58
4	-14.98	2.26	-14.62
5	-17.19	-0.37	-16.33
6	-20.45	1.70	-16.05
7	-18.74	5.08	-16.63
8	-17.21	3.64	-19.87
9	-20.57	2.23	-21.15
10	-21.98	5.69	-20.30
11	-19.95	7.13	-23.19
12	-20.89	4.12	-25.38
13	-24.69	4.65	-24.80
14	-24.10	8.21	-26.13
15	-22.22	6.94	-29.27
16	-25.35	4.76	-29.96
17	-27.56	7.86	-29.34
18	-25.30	10.10	-31.49
19	-25.44	7.48	-34.33
20	-29.25	7.11	-33.89
21	-29.63	10.94	-34.20
22	-26.92	11.13	-36.99
23	-29.24	9.41	-39.54

TABLE B.7: Helix 7 dataset.

	y_1	y_2	y_3
1	82.81	124.49	140.71
2	81.83	120.79	140.80
3	83.75	120.09	137.55
4	86.97	120.66	139.40
5	86.52	118.48	142.41
6	85.16	115.81	140.08
7	88.24	115.60	137.88
8	90.32	115.53	141.05
9	88.28	112.54	142.17
10	88.97	110.96	138.80
11	92.67	111.74	139.04
12	92.57	110.44	142.65
13	91.14	107.08	141.60
14	93.90	106.70	138.98
15	96.51	107.77	141.58
16	94.99	105.50	144.20
17	95.24	102.63	141.72
18	98.83	103.54	140.90
19	99.81	103.61	144.57

TABLE B.8: Helix 8 dataset.

	y_1	y_2	y_3
1	-25.71	42.83	25.36
2	-25.48	46.54	26.16
3	-21.76	46.96	25.48
4	-22.28	44.78	22.42
5	-24.93	47.17	21.05
6	-22.98	50.29	22.03
7	-20.17	48.51	20.26
8	-22.24	47.96	17.11
9	-23.42	51.56	17.06
10	-19.97	53.04	17.67
11	-18.50	50.84	14.97
12	-20.88	52.22	12.34
13	-20.08	55.75	13.45
14	-16.40	55.12	12.85
15	-17.24	53.44	9.55

TABLE B.9: Helix 9 dataset.

	y_1	y_2	y_3
1	-25.71	42.83	25.36
2	-25.48	46.54	26.16
3	-21.76	46.96	25.48
4	-22.28	44.78	22.42
5	-24.93	47.17	21.05
6	-22.98	50.29	22.03
7	-20.17	48.51	20.26
8	-22.24	47.96	17.11
9	-23.42	51.56	17.06
10	-19.97	53.04	17.67
11	-18.50	50.84	14.97
12	-20.88	52.22	12.34
13	-20.08	55.75	13.45
14	-16.40	55.12	12.85
15	-17.24	53.44	9.55

Bibliography

M. Alfahad, J. T. Kent, and K. V. Mardia. Statistical methods for analysis of helices. *Sankhya A*, 2018. In press.

J. Andrade, M. Gómez-Carracedo, W. Krzanowski, and M. Kubista. Procrustes rotation in analytical chemistry, a tutorial. *Chemometrics and Intelligent Laboratory Systems*, 72(2):123–132, 2004.

G. Arfken and H. Weber. *Mathematical Methods for Physicists*. Harcourt Academic, fifth edition, 2001.

H. Berman, K. Henrick, H. Nakamura, and J. L. Markley. The worldwide protein data bank (wwpdb): ensuring a single, uniform archive of pdb data. *Nucleic Acids Research*, 35(suppl 1):D301–D303, 2007.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

C. Branden and J. Tooze. *Introduction to Protein Structure*. Garland Pub, 1999.

M. K. Campbell and S. O. Farrell. *Biochemistry*. Brooks/Cole, 2009.

J. Chen and A. K. Gupta. *Parametric Statistical Change Point Analysis: with Applications to Genetics, Medicine, and Finance*. Springer Science & Business Media, 2011.

K. Chou and Y. Cai. Predicting protein quaternary structure by pseudo amino acid composition. *Proteins: Structure, Function, and Bioinformatics*, 53(2): 282–289, 2003.

J. A. Christopher, R. Swanson, and T. O. Baldwin. Algorithms for finding the axis of a helix: fast rotational and parametric least-squares methods. *Computers and Chemistry*, 20(3):339–345, 1996.

T. Creighton. *Proteins: Structures and Molecular Properties*. Freeman, second edition, 1993.

J. Deville, J. Rey, and M. Chabbert. Comprehensive analysis of the helix-x-helix motif in soluble proteins. *Proteins: Structure, Function, and Bioinformatics*, 72(1):115–135, 2008.

I. L. Dryden and K. V. Mardia. *Statistical Shape Analysis: With Applications in R*. John Wiley & Sons, second edition, 2016.

P. Garthwaite, I. Jolliffe, and B. Jones. *Statistical Inference*. Oxford University Press on Demand, 2002.

G. H. Golub and C. Reinsch. Singular value decomposition and least squares solutions. In *Linear Algebra*, pages 134–151. Springer, 1971.

I. S. Gradshteyn and I. M. Ryzhik. *Table of integrals, series, and products*. Academic press, 2014.

S. Hall, K. Roberts, and N. Vaidehi. Position of helical kinks in membrane protein crystal structures and the accuracy of computational prediction. *Journal of Molecular Graphics and Modelling*, 27(8):944–950, 2009.

R. Harris. *A Primer of Multivariate Statistics*. Psychology Press, 2001.

H. Heise, S. Matthews, and S. Appelt. *Modern NMR Methodology*. Springer, 2013.

T. Kato. *Perturbation Theory for Linear Operators*, volume 132. Springer Science & Business Media, 2013.

J. T. Kent and K. V. Mardia. Shape Procrustes tangent projections and bilateral symmetry. *Biometrika*, 88(2):469–485, 2001.

J. T. Kent, J. Briden, and K. V. Mardia. Linear and planar structure in ordered multivariate data, as applied to progressive demagnetization of palaeomagnetic remanence. *Geophysical Journal of the Royal Astronomical Society*, 75: 593–621, 1983.

Hyune-Ju Kim and David Siegmund. The likelihood ratio test for a change-point in simple linear regression. *Biometrika*, 76(3):409–423, 1989.

K. Knight. *Mathematical statistics*. Chapman and Hall/CRC Press, Boca Raton, 2000.

P. Kumar and M. Bansal. Helanal-plus: a web server for analysis of helix geometry in protein structures. *Journal of Biomolecular Structure and Dynamics*, 30 (6):773–783, 2012.

D. C. Lay. *Linear Algebra and Its Applications*. Addison-Wesley, 2006.

S. R. Lele and J. T. Richtsmeier. *An invariant approach to statistical analysis of shapes*. CRC Press, 2001.

J. Magnus and H. Neudecker. *Matrix Differential Calculus With Applications in Statistics and Econometrics*. Wiley, third edition, 2003.

K. V. Mardia. Statistical approaches to three key challenges in proteins structural bioinformatics. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 62(3):487–514, 2013.

K. V. Mardia. In-depth modelling of some angular shapes in proteins with applications: modelling conics and helices. Presentation at ADISTA, Brussels, 2014.

K. V. Mardia and D. Holmes. A statistical analysis of megalithic data under elliptic pattern. *Journal of the Royal Statistical Society*, 143(3):293–302, 1980.

K. V. Mardia, J. T. Kent, and J. Bibby. *Multivariate Analysis*. Academic Press, London, 1979.

K. V. Mardia, K. Sriram, and C. M. Deane. A statistical model for helices with applications. *Biometrics*, 74:845–854, 2018.

J. E. Marsden and T. S. Ratiu. Introduction to mechanics and symmetry. *Physics Today*, 48:1, 1995.

B. Q. Miao. Inference in a model with at most one slope-change point. In *Multivariate Statistics and Probability*, pages 375–391. Elsevier, 1989.

J. Miao, P. Charalambous, J. Kirz, and D. Sayre. Extending the methodology of x-ray crystallography to allow imaging of micrometre-sized non-crystalline specimens. *Nature*, 400(6742):342–344, 1999.

J. Moller and R. Waagepetersen. *Statistical Inference and Simulation for Spatial Point Processes*, volume 100. CRC, 2004.

A. M. Mood, F. A. Graybill, and D. Boes. Introduction to the theory of statistics, mcgraw-hill. *DaCosta, CJ and Baenziger, JE et al (2003). A rapid method for assessing lipid: protein and detergent: protein ratios in membrane–protein crystallization*, 59:77–83, 1974.

R. Murray, Z. Li, and S. Sastry. *Introduction to Robotic Manipulation*. CRC press, 1994.

M. Ohya and N. Watanabe. *Selected Papers of M. Ohya*. World Scientific, 2008.

B. O'Neill. *Elementary Differential geometry*. Academic Press, second edition, 1997.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2014.

T. A. Reddy. *Applied Data Analysis and Modeling for Energy Engineers and Scientists*. Springer Science & Business Media, 2011.

J. A. Rice. *Mathematical Statistics and Data Analysis*. Thomson, Belmont, CA, third edition, 2007.

I. Rigoutsos, P. Riek, R. Graham, and J. Novotny. Structural details (kinks and non-α conformations) in transmembrane helices are intrahelically determined and can be predicted by sequence pattern descriptors. *Nucleic Acids Research*, 31(15):4625–4631, 2003.

C. P. Simon and L. Blume. *Mathematics for Economists*, volume 7. Norton New York, 1994.

L. M. Sullivan. *Essentials of Biostatistics in Public Health*. Jones and Bartlett Publishers, Sudbury, Mass, third edition, 2008.

H. R. Wilman, J. Shi, C. M. Deane, J. Dunbar, A. Fuchs, K. V. Mardia, and R. Diagnostics. Describing protein structure geometry to aid in functional understanding. *LASR2013 Proceedings Statistical Models and Methods for non-Euclidean Data with Current Scientific Applications*, 2013.

H. R. Wilman, J. P. Ebejer, J. Y. Shi, C. M. Deane, and B. Knapp. Crowdsourcing yields a new standard for kinks in protein helices. *Journal of Chemical Information and Modeling*, 54(9):2585–2593, 2014a.

H. R. Wilman, J. Shi, and C. M. Deane. Helix kinks are equally prevalent in soluble and membrane proteins. *Proteins: Structure, Function, and Bioinformatics*, 82(9):1960–1970, 2014b.