On the Shapes of Tangled Curves

by

Patrick Kessler

B.S. (University of California, Berkeley) 2001
M.S. (University of California, Berkeley) 2003
M.A. (University of California, Berkeley) 2006

A dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy
in

Engineering- Mechanical Engineering

in the

GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA, BERKELEY

Committee in charge:

Oliver M. O'Reilly, Chair
Professor Tarek I. Zohdi
Professor Carlo H. Séquin

Fall 2007

The dissertation of Patrick Kessler is approved.

Chair Date

Date

Date

University of California, Berkeley

Fall 2007

On the Shapes of Tangled Curves

by

Patrick Kessler

Abstract

On the Shapes of Tangled Curves

by

Patrick Kessler

Doctor of Philosophy in Engineering- Mechanical Engineering

University of California, Berkeley

Oliver M. O'Reilly, Chair

A new method is introduced for investigating the shapes of space curves. The method involves analyzing intersections between a curve and a moving manifold. The positions and orientations of the manifold are determined by the curve itself, and so the method is invariant under curve translations and rotations.

The method is investigated in detail in two instances, one in which the manifold is a planar triangle and the other in which the manifold is a disk. Each of these cases involves unique subtleties that are explored in depth. In all instances of the method, the intersections between manifold and curve are illustrated as a two dimensional point set. This is the power of the method, that it allows intricately convoluted space curves to be represented as a single digestible planar image. Whether one segment of curve wraps around or encircles another is conveyed by the presence of a line in the plane, no matter how deeply buried in a tangle these segments may be. Other aspects of the way in which segments fit together also follow from the method, such as whether the encirclement of one segment about another is tight or loose.

Although the method provides topological information (such as the linking number of two curves), we pursue it more for what it tells about curve shape and geometry. The planar image generated by the method rapidly communicates information about curve structure, and so the image can be used as a *shape descriptor*, that is, as an abbreviation of critical shape parameters. As shape descriptors, images from the method can be used to rapidly compare and categorize libraries of curve structures. As a proof of concept demonstration, the method is used in this way to navigate a representative sample of protein backbone curves. The method successfully distinguishes between protein molecules from different structural families.

Oliver M. O'Reilly
Dissertation Committee Chair

Contents

Acknowledgements

My thanks and appreciation go foremost and above all to Oliver O'Reilly who has given this work and my studies at large his unwavering support. My thanks to Carlo Séquin, who applied the full force of his amazing energy and enthusiasm to this project- his innumerable ideas helped make this a better dissertation. Finally, I would also like to thank Tarek Zohdi.

In addition to these three committee members, literally hundreds of people have had a significant positive influence on my work. These include professors and researchers at Cal and other schools, my students, my labmates, my family, and my friends. I am grateful to each and every person who helped me out.

Chapter 1

Introduction

Consider a plate of spaghetti, or the interwoven root tendrils of a plant. These coiled and tangled curves exhibit a wild disorder that engineers usually avoid when they design machines. These curves however are ubiquitous in the mechanisms of nature, where they form beautiful and highly functional structures (see Figure 1.1). This dissertation is about understanding the geometry and shape of these structures.

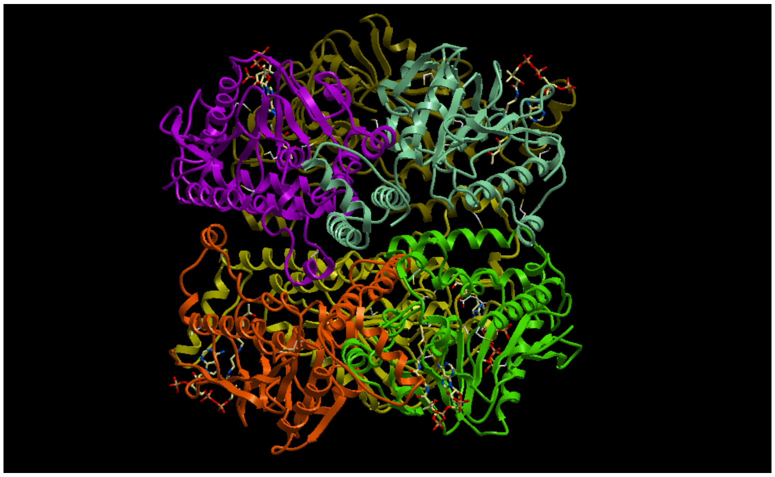


Figure 1.1. Protein molecules consist of tangled chains of atoms; the shape of the tangle determines the function of the protein. Here we show the protein molecule with PDB identifier 1DUB. This image is from http://bioinfo.nist.gov/hmpd/gallery/room1.html .

In Chapter 2 we review existing tools for characterizing the shapes of space curves. In

Chapters 3 and 4 we introduce a novel method for characterizing curve shape. Some of the most exciting applications of this method are to biological fibers, such as entangled vines, supercoiled DNA, and protein molecules. These fibers have fascinating shapes that oftentimes need to be classified. In Chapter 6 we use our novel method to construct a metric on protein molecules. This metric gives rise to new protein classification schemes.

Masses of tangled strands arise all the time in nature, and we have already mentioned several biological fibers that tangle. For an example of tangled strands made by people, consider the intertwined microscopic fibers that comprise felt [1]. The properties of individual felt fibers and the geometry of entanglement can be related to the bulk properties of the larger continuum. How strongly can we pull on a felt garment before it rips? How does the strength of the garment depend on the way in which the microscopic fibers are twisted together? We anticipate that our methods of looking at entangled strands will help answer these questions. A suggestive image of microscopic fibers is given in Figure 1.2, and images of large woven fibers are given in Figure 1.3.

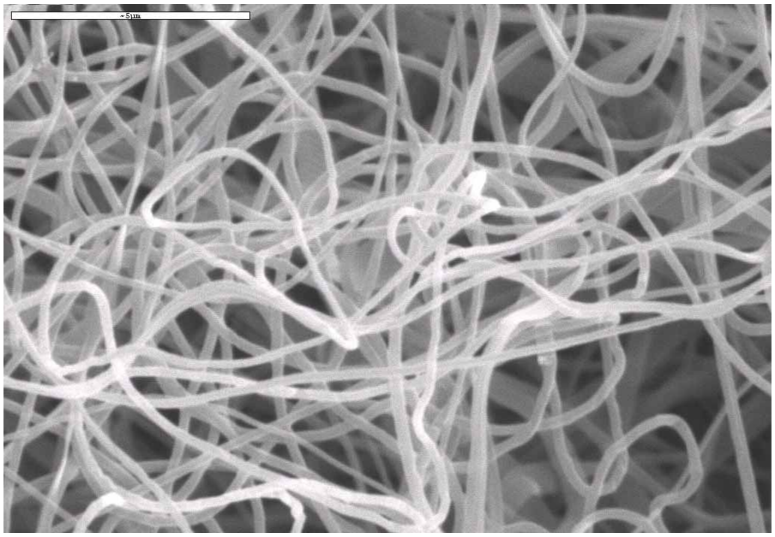


Figure 1.2. Microscopic tangled fibers. The bar in the upper left hand corner is 5 μm long. This image comes from the website of the nonwovens research group in the Department of Textile Industries at the University of Leeds, www.nonwovens.leeds.ac.uk/ .

The physical properties of real world curves such as a strand of DNA or a collection of cloth fibers are not sufficiently captured by topological invariants. For these real world

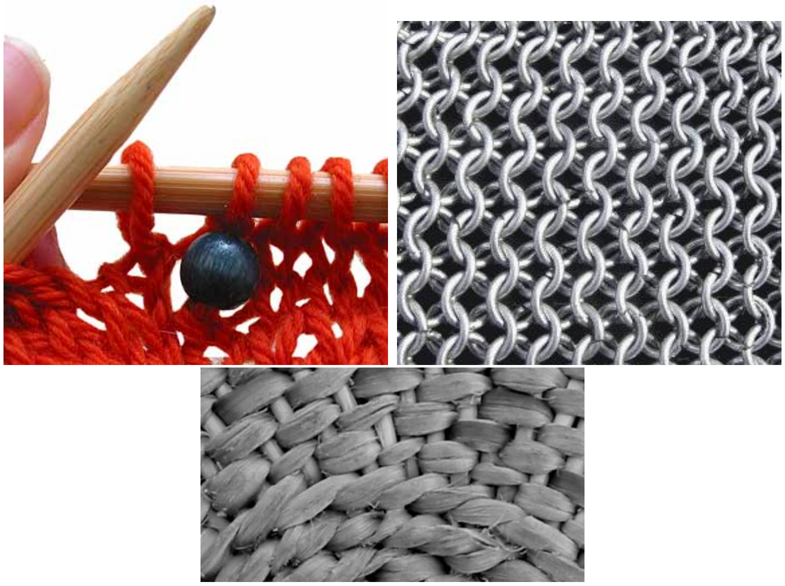


Figure 1.3. Larger scale arrangements of fiber. The image of a knitted garment comes from www.stitchdiva.com , the chain mail comes from www.wikipedia.com , and the basket comes from www.primitiveways.com .

curves, we need to know for instance how *tightly* and how *loosely* two curves wrap around one another. In this dissertation we introduce an elegant characterization of encirclement that provides this information. This characterization distinguishes encirclement at different scales, just as the Fourier transform identifies components of a signal at different frequencies. In addition to a new way of understanding curve shape which is important for real world curves (often called *physical knots*), our characterization provides information on curve topology, for instance yielding the linking number of two closed curves.

Generally, we consider the pattern of intersections between a curve $\mathbf{x}$ and a d-scale manifold (e.g., a disk, sphere, or triangle with maximum dimension d) associated with a point $\mathbf{x}(s)$ on the curve, (see Figure 1.4). As we vary s, the manifold translates and rotates along the curve, and intersects the curve at different points. We keep track of where these intersections occur on the (arc-length parameterized) space curve with a finger-print like *intersection set* (I-set) that is easy to analyze and interpret [21]. Manifolds of different shapes and sizes result in intersection diagrams that emphasize particular structural features of the underlying curve. For instance, helices leave distinctive signature markings when the manifold is a triangle, and parallel strands leave distinctive markings when the manifold is a disk. Intersection diagrams reflect more than local information. The problem with the usual global invariants is that they are often too lossy in cases when details of the curve shape are important.

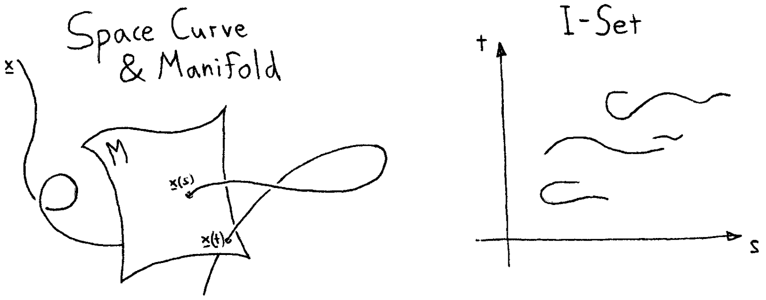


Figure 1.4. A manifold M is associated with each point $\mathbf{x}(s)$ on a curve $\mathbf{x}$. This manifold may intersect $\mathbf{x}$ at other points such as $\mathbf{x}(t)$. The pattern of these intersections is recorded in an I-set (shown at right).

In our approach we represent a curve in 3D as a collection of curves in 2D (i.e., an I-set). The metric we introduce on space curves is based on comparing their associated I-subsets. The computational cost associated with this comparison can be dramatically reduced by turning the 2D I-set into a 1D object. In ongoing work with researchers in structural bioinformatics, we have used this approach to construct a fast algorithm for comparing protein molecules (see chapter 5).

Our comparison of space curves is related to work done in the field of computer graphics on comparing general three dimensional shapes [25, 36]. Web-based search and retrieval algorithms for shape have been developed that are accurate, fast, and robust. These can distinguish for instance between something shaped like an airplane and something shaped like a coffee mug. Many ingenious approaches are used to do this, such as *shape distributions* [30], in which a spatial object is characterized by a histogram of the distances between pairs of points sprinkled randomly over its surface. The histogram is a *shape descriptor*; something easier to work with and compare than the original object which nonetheless captures the object's essential shape attributes. The I-sets which we consider in this dissertation are also shape descriptors. In Chapter 7 we study the shapes of twelve protein molecules using I-sets as well as shape distributions.

The dissertation ends with a set of concise appendices which offer detail on some of the algorithms and theoretical constructions that we use. We consider multi-dimensional scaling in Appendix A, the Kuhn-Munkres algorithm for the assignment problem in Appendix B, sequential list alignment in Appendix C, and finally, the linking number in Appendix D.

Chapter 2

Survey of Curve Characterizations

Here we review common characterizations of three-dimensional space curves. Many of these are described on an intuitive level in [7], with more rigor in [22], and in contemporary terms in [13]. The setting for our work is an oriented[1] three-dimensional inner product space over $\mathbb{R}$, often called *Euclidean* 3-space and denoted by $\mathbb{E}^3$. This space rather than $\mathbb{R}^3$ is the appropriate arena for work with *physical curves*, in which the distance between curve points is important. Workers who consider curves only in $\mathbb{R}^3$ are usually interested in topological questions. Work on curve topology in $\mathbb{R}^3$ applies to curves in the (more structured) $\mathbb{E}^3$ due to a natural homeomorphism between these two spaces. With $(\mathbf{E}_1, \mathbf{E}_2, \mathbf{E}_3)$ an orthonormal basis for $\mathbb{E}^3$, this map takes $(x_1, x_2, x_3) \in \mathbb{R}^3$ to $\sum x_i \mathbf{E}_i \in \mathbb{E}^3$; the inverse map takes $\mathbf{x} \in \mathbb{E}^3$ to $(\mathbf{x} \cdot \mathbf{E}_1, \mathbf{x} \cdot \mathbf{E}_2, \mathbf{x} \cdot \mathbf{E}_3) \in \mathbb{R}^3$.

A *curve* is the image of an imbedding $\mathbf{x} : K \longrightarrow \mathbb{E}^3$, where K is a connected subset of $\mathbb{R}$. We ambiguously use $\mathbf{x}$ to denote both the imbedding and its image. This is harmless in our work as we almost never switch between imbeddings of the same curve. We assume that $\mathbf{x}$ is as differentiable as necessary, and that the distance along $\mathbf{x}$ from $\mathbf{x}(s_1)$ to $\mathbf{x}(s_2)$ is equal to $|s_2 - s_1|$, i.e., that s is an *arc-length parameter* for $\mathbf{x}$. A *closed curve* is a curve in which K has the form $[\alpha, \beta)$, and in which $\mathbf{x}^{[n]}(s) \longrightarrow \mathbf{x}^{[n]}(\alpha)$ as $s \longrightarrow \beta$, for $n \in \{0, 1, 2, \ldots, N\}$ with $N \geq 0$. ($\mathbf{x}^{[n]}(s)$ denotes the n^{th} derivative of $\mathbf{x}$ with respect to s.)

[1] The space needs to be oriented for the cross product to be well defined.

2.1 Tools from Differential Geometry and Topology

2.1.1 The Frenet Triad

At any point $\mathbf{x}(s)$ on $\mathbf{x}$, we define the *tangent vector* $\mathbf{t} = \frac{d\mathbf{x}}{ds}$, the *curvature* $\kappa = \|\frac{d\mathbf{t}}{ds}\|$, the *normal vector* $\mathbf{n} = \frac{1}{\kappa}\frac{d\mathbf{t}}{ds}$, the *binormal vector* $\mathbf{b} = \mathbf{t} \times \mathbf{n}$, and finally the *torsion* τ, by $\frac{d\mathbf{b}}{ds} = -\tau\mathbf{n}$. We call $r = \frac{1}{\kappa}$ the *radius of curvature* of $\mathbf{x}$ at s. Although plenty of vectors besides $\mathbf{t}$ and $\mathbf{n}$ are geometrically tangent and normal to the curve, these definitions allow us to say "the tangent vector" and "the normal vector" without ambiguity, (provided $\kappa \neq 0$). The orthonormal vectors $\mathbf{t}$, $\mathbf{n}$, and $\mathbf{b}$ are referred to as the *Frenet Triad*, see Figure 2.1.

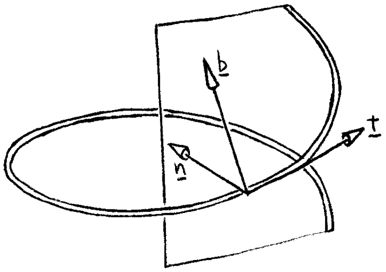


Figure 2.1. The Frenet triad vectors $\mathbf{t}$, $\mathbf{n}$ and $\mathbf{b}$ associated with a point on a space curve.

2.1.2 The Normal Injectivity Radius

The *normal injectivity radius* r_i of $\mathbf{x}$ is the radius of the largest singularity-free tube with centerline $\mathbf{x}$. Singularities arise when a point on the boundary of the normal disk centered at $\mathbf{x}(s_1)$ intersects a point on the boundary of the normal disk centered at $\mathbf{x}(s_2 \neq s_1)$, and so a singularity is identified by a pair (s_1, s_2) of arc length values. The set S of all such values is a collection of closed intervals on the real line. We establish an equivalence class on tube singularities (s_i, s_j) by grouping together those for which the segment of S containing s_i is connected to the segment of S containing s_j. Both classes of singularity are shown in Figure 2.2.

Professor Séquin has suggested considering tubes with variable thickness. The radius of these tubes could depend for instance on the curvature of the tube centerline. Thick tubes

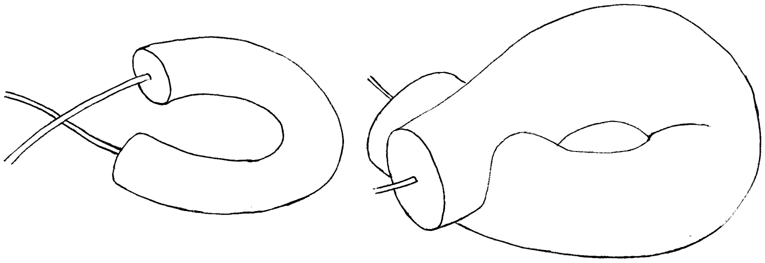


Figure 2.2. The same curve is the centerline for tubes with different radius values. The larger tube exhibits singularities from two different classes.

in all their forms apply to many interesting areas of current research, including the growth of plant tendrils.

2.1.3 Global Radius of Curvature

The *global radius of curvature* of the curve $\mathbf{x} : C \longrightarrow \mathbb{E}^3$ at the point $s \in C$ is given by

$$r_g(s) = \min_{t,u \in C} R(s,t,u) \tag{2.1}$$

where $R(s,t,u)$ is the radius of the circle circumscribing the triangle T_{stu} with vertices $\mathbf{x}(s)$, $\mathbf{x}(t)$, and $\mathbf{x}(u)$. $R(s,t,u)$ can be computed using the relation

$$4R(s,t,u)A(s,t,u) = \|\mathbf{x}(s) - \mathbf{x}(t)\| \cdot \|\mathbf{x}(s) - \mathbf{x}(u)\| \cdot \|\mathbf{x}(t) - \mathbf{x}(u)\|, \tag{2.2}$$

where $A(s,t,u)$ is the area of T_{stu}.

The global radius of curvature was first introduced in [16]. This radius is a continuous function on C, and satisfies $0 \le r_g(s) \le r(s)$, where $r(s)$ is the standard curvature of $\mathbf{x}$ at s. The minimum value of the global radius of curvature over C equals the normal injectivity radius r_i of $\mathbf{x}$. In [15] it is noted that a billiard ball with radius less than r_i cannot find a stable resting place on the curve, (i.e., it cannot intersect the curve at three or more points, counting tangency points twice). Gonzalez and his coworkers call an *ideal* curve with a particular knot type one which has the maximum possible r_i (among all curves with the same knot type and the same length). They also prove that a curve $\mathbf{x}$ is ideal only if there is a constant $a > 0$ such that

$$\begin{aligned} r_g(s) &\geq a \qquad \text{for} \qquad s \in J^*, \\ r_g(s) &= a \qquad \text{for} \qquad s \in C \backslash J^*, \end{aligned} \tag{2.3}$$

where $J^* \subset C$ is the set of points at which the standard radius of curvature vanishes. Physically, a is the thickness of the ideally shaped curve $\mathbf{x}$.

2.1.4 Winding and Turning

The (signed) number of times a closed plane curve $\mathbf{x} : S^1 \longrightarrow \mathbb{E}^2$ wraps around a point $\mathbf{p} \in \mathbb{E}^2$ is called the *winding number* of $\mathbf{x}$ with respect to $\mathbf{p}$, and is denoted $W(\mathbf{x}, \mathbf{p})$ [29]. Here we relax the injectivity requirement on $\mathbf{x}$, and allow the curve to intersect itself at a finite number of points, (see Figure 2.3).

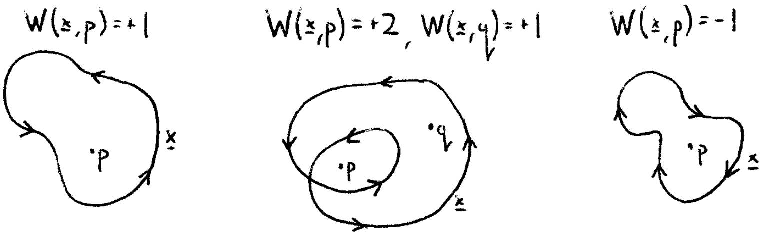


Figure 2.3. Examples of plane curves and their winding numbers.

Formally, $W(\mathbf{x}, \mathbf{p})$ can be calculated by considering the intersections of $\mathbf{x}$ with a ray R emanating from $\mathbf{p}$, where $R = \{\mathbf{p} + \alpha\mathbf{u} \mid \alpha \geq 0\}$ for some vector $\mathbf{u}$. The only requirement on $\mathbf{u}$ is that it be chosen so that R intersects $\mathbf{x}$ at finitely many points, and that w_k =sign$\big(\mathbf{u} \times \frac{\partial \mathbf{x}}{\partial s}\big)$ is nonzero at each $s_k \in S^1$ corresponding to an intersection. Then $W(\mathbf{x}, \mathbf{p})$ is the sum of the w_k's. Alternatively, $W(\mathbf{x}, \mathbf{p})$ can be defined as the *degree of the map* which takes $s \in S^1$ to the unit vector from $\mathbf{p}$ to $\mathbf{x}(s)$. We discuss a *degree of map* definition in detail in Appendix D, in the context of the linking number of two space curves.

The *turning number* of $\mathbf{x}$ is the winding number of $\dot{\mathbf{x}}$ with respect to the origin. Unlike the winding number, the turning number does not depend on some point in addition to the given space curve.

2.1.5 Linking Number

Let $\mathbf{x}$ and $\mathbf{y}$ be closed directed curves in $\mathbb{R}^3$. The linking number of $\mathbf{x}$ with respect to $\mathbf{y}$ is an integer that captures the number of times one of these curves wraps around the other. Let T be the product of the domains of $\mathbf{x}$ and $\mathbf{y}$. The linking number can be defined as the *degree of the map* $\mathbf{u} : T \longrightarrow S^2$, which assigns to (s, t) the unit vector from $\mathbf{x}(s)$ to $\mathbf{y}(t)$ (see Appendix D). This is analogous to the winding number; both linking and winding can be defined as degrees of maps, corresponding to the number of times one thing wraps around another.

Like the winding number, $Lk(\mathbf{x}, \mathbf{y})$ is a topological invariant; it does not change if the space in which $\mathbf{x}$ and $\mathbf{y}$ are imbedded undergoes a continuous transformation. That is, if we were to construct $\mathbf{x}$ and $\mathbf{y}$ out of string, $Lk(\mathbf{x}, \mathbf{y})$ would have the same value for all deformations of the string that do not involve cutting either $\mathbf{x}$ or $\mathbf{y}$. In Appendix D, we show that $Lk(\mathbf{x}, \mathbf{y})$ is given by the following *Gauss Integral*

$$Lk(\mathbf{x}, \mathbf{y}) = \frac{1}{4\pi} \int_T \left(\frac{d\mathbf{x}}{ds} \times \frac{d\mathbf{y}}{dt} \right) \cdot \frac{\mathbf{x} - \mathbf{y}}{\|\mathbf{x} - \mathbf{y}\|^3} ds\, dt. \tag{2.4}$$

Gauss integrals have been used as the basis for an automatic protein classification scheme [32]. Although the ability to separate $\mathbf{x}$ and $\mathbf{y}$ without cutting implies that $Lk(\mathbf{x}, \mathbf{y}) = 0$, the converse implication is false, as shown in Figure 2.4.

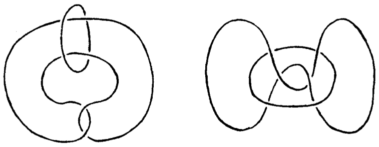


Figure 2.4. The Whitehead link shows that a zero linking number does not imply that two closed curves can be physically separated.

2.1.6 Definition of a Strip

Let $\mathbf{t}(s)$ be a vector tangent to the space curve $\mathbf{x}$ at the point $\mathbf{x}(s)$. Let $\mathbf{u}$ be a smooth vector field on the domain D of $\mathbf{x}$, for which $\mathbf{u}(s) \cdot \mathbf{t}(s) = 0$ for every $s \in D$. Following [14], we refer to the pair $(\mathbf{x}, \mathbf{u})$ as a *strip*. We call the limit of $Lk(\mathbf{x}, \mathbf{x} + \epsilon\mathbf{u})$ as $\epsilon \longrightarrow 0$ the linking number *of the strip* $(\mathbf{x}, \mathbf{u})$. An example of a field $\mathbf{u}$ associated with a curve $\mathbf{x}$ is the field $\mathbf{n}$ of Frenet normal vectors associated with $\mathbf{x}$.

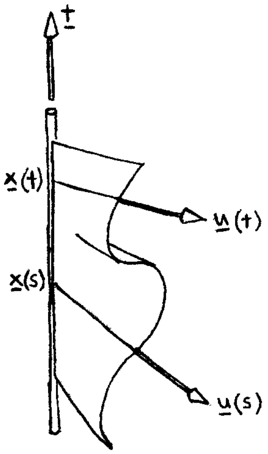


Figure 2.5. Example of a strip where $\mathbf{x}$ is a vertical straight line.

2.1.7 Twist and Total Twist Number

Let $\mathbf{x} : C \longrightarrow \mathbb{E}^3$ be a space curve, and let $\mathbf{e}_1(s)$ and $\mathbf{e}_2(s)$ be unit tangent and normal vectors respectively to $\mathbf{x}(s)$ at $s \in C$. These two vectors together with $\mathbf{e}_3(s) = \mathbf{e}_1(s) \times \mathbf{e}_2(s)$ comprise an orthonormal triad that rotates like a rigid body with change in s, (i.e., $\mathbf{e}_i = \mathbf{Q}\mathbf{E}_i$ for some proper-orthogonal operator $\mathbf{Q} = \mathbf{Q}(s)$, where $(\mathbf{E}_1, \mathbf{E}_2, \mathbf{E}_3)$ is a right-handed orthonormal basis for $\mathbb{E}^3$). Let $\boldsymbol{\omega}$ be the angular velocity vector associated with this motion, (i.e., $\boldsymbol{\omega} \times \mathbf{a} = \frac{d\mathbf{Q}}{ds}\mathbf{Q}^T\mathbf{a}$ for every $\mathbf{a} \in \mathbb{E}^3$).

The *twist* ω_t of the strip $(\mathbf{x}, \mathbf{e}_2)$ is the $\mathbf{e}_1$ component of $\boldsymbol{\omega}$. If $\mathbf{e}_2$ is the Frenet normal vector (which is well defined only when the curvature κ of $\mathbf{x}$ is nonzero), the associated twist is the Frenet torsion τ of the curve.

The *total twist number* of the strip $(\mathbf{x}, \mathbf{e}_2)$ is given by $\frac{1}{2\pi} \oint \omega_t \, ds$. We denote the total twist number by $Tw(\mathbf{x}, \mathbf{e}_2)$, and we note that this number need not be an integer. The integral $\oint \omega_t \, ds$ is often called the *total twist* of the strip.

2.1.8 Writhing Number

The *writhing number* of a closed curve $\mathbf{x}$ is defined as

$$Wr(\mathbf{x}) = \frac{1}{4\pi}\int_{K\times K}\left(\frac{d\mathbf{x}}{ds}\times\frac{d\mathbf{y}}{dt}\right)\cdot\frac{\mathbf{x}-\mathbf{y}}{\|\mathbf{x}-\mathbf{y}\|^3}ds\,dt. \tag{2.5}$$

In light of (2.4), the writhing number of $\mathbf{x}$ is often called the *self-linking number* of $\mathbf{x}$. The writhing number is discussed in further detail in Appendix D. As noted in [14], $Wr(\mathbf{x})$ is independent of rigid motions, dilations, and the direction of travel along $\mathbf{x}$. The writhing, linking, and total twist numbers associated with a closed curve $\mathbf{x}$ and field of normal vectors $\mathbf{u}$ satisfy

$$Wr(\mathbf{x}) = Lk(\mathbf{x},\mathbf{x}+\epsilon\mathbf{u}) - Tw(\mathbf{x},\mathbf{u}). \tag{2.6}$$

Reflections change the signs of Lk, Tw, and Wr. Equation 2.6 and the name *writhing number* were first given by Fuller [14] in 1971, however the writhing number concept was first introduced by Călugăreanu in a series of papers [8, 9, 10] published from 1959 to 1961. The writhing number is often used as a measure of the extent to which a curve wraps and coils around itself. Note that (2.5) corresponds to the *inductance* of a wire in the shape of a $\mathbf{x}$. This formula can be motivated by "current through a wire" arguments, and the Biot-Savart law [3].

2.1.9 Directional Writhing Number

If $\mathbf{d}$ is a fixed vector not parallel to any tangent vector of $\mathbf{x}$, then following [14], we say that $\mathbf{x}$ and $\mathbf{d}$ are in *general position*, and we define the *directional writhing number* of $\mathbf{x}$ and $\mathbf{d}$ to be the limit of $Lk(\mathbf{x},\mathbf{x}+\epsilon\mathbf{d})$ as $\epsilon \longrightarrow 0$. We denote this number by $Wr(\mathbf{x},\mathbf{d})$. As described in [14], $Wr(\mathbf{x},\mathbf{d})$ is simply the sum of the (signed) over and under crossings of $\mathbf{x}$ projected onto the plane normal to $\mathbf{d}$, and averaging $Wr(\mathbf{x},\mathbf{d})$ over all unit vectors $\mathbf{d}$ (with respect to area on the unit sphere S^2) gives the writhing number $Wr(\mathbf{x})$

$$Wr(\mathbf{x}) = \frac{1}{4\pi}\int_{S^2} Wr(\mathbf{x},\mathbf{d})dA. \tag{2.7}$$

2.2 Tools from Molecular Modeling

Several methods of characterizing overall curve shape have been developed for the particular purpose of characterizing protein molecules.

2.2.1 Ramachandran Plots

A protein molecule consists of a chain of atoms. A Ramachandran plot is a graph of angles along the length of the main chain that describe how much it is bending. These plots were introduced by G.N. Ramachandran and his coworkers in 1963. Tamar Schlick [34] recommends Ramachandran's biography [33] for historical details.

Consider a chain of N nodes $(\mathbf{x}_k)$ in $\mathbb{E}^3$, with $\mathbf{x}_k$ connected to $\mathbf{x}_{k+1}$ for $k = 1, \ldots, N-1$. Let $\mathbf{e}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$, and for $k = 2, \ldots, N-3$, define the *dihedral angle* associated with $\mathbf{x}_k$ to be the angle between the plane containing $\mathbf{e}_{k-1}$ and $\mathbf{e}_k$, and the plane containing $\mathbf{e}_k$ and $\mathbf{e}_{k+1}$.

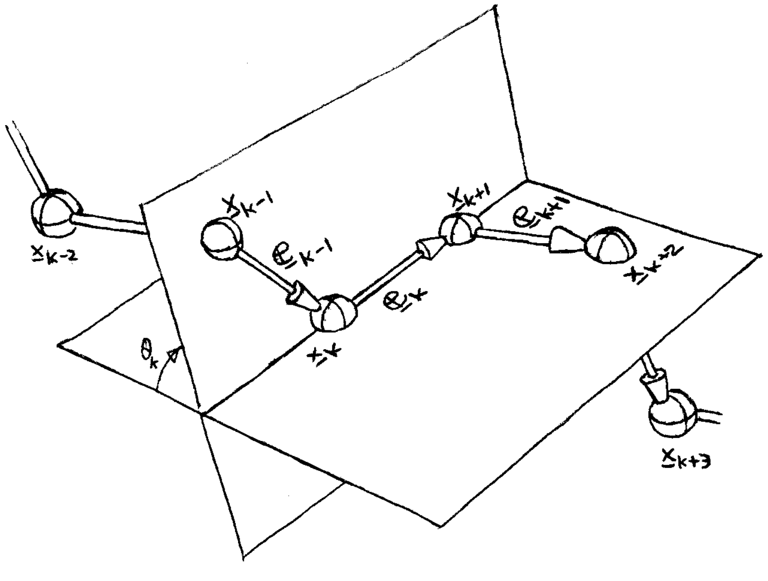


Figure 2.6. Notation for angles and vectors associated with a chain segment.

Formally, let $\mathbf{a}$ be the unit vector in the direction of $\mathbf{e}_{k-1} - \mathbf{e}_k(\mathbf{e}_k \cdot \mathbf{e}_{k-1})$, let $\mathbf{b}$ be the unit vector in the direction of $\mathbf{a} \times \mathbf{e}_k$, and let $\mathbf{c}$ be the unit vector in the direction of

$\mathbf{e}_{k+1} - \mathbf{e}_k(\mathbf{e}_k \cdot \mathbf{e}_{k+1})$. The dihedral angle associated with $\mathbf{x}_k$ is then given by

$$\theta_k = \text{arctan2}(y, x), \quad \text{where } \; x = -\mathbf{a} \cdot \mathbf{c}, \quad \text{and where } \; y = \mathbf{b} \cdot \mathbf{c}. \tag{2.8}$$

The arctan2 function accepts Cartesian coordinates x and y for a point in the plane, and returns the associated angle on $(-\pi, \pi]$. Dihedral angles alone are not enough to reconstruct the original (piecewise linear) curve in $\mathbb{E}^3$, for instance the dihedral angles of any planar curve are all zero. However, for certain families of curves (such as the backbone curves of protein molecules), dihedral angles can indicate important information about curve structure.

The nodes in a protein molecule consist of repeating groups of three atoms: an N, a C^α, and a C. Numbering these groups (called *residues*) from 1 to n, a protein chain can be represented as follows

$$(N - C^\alpha - C)_1 - (N - C^\alpha - C)_2 - \quad \cdots \quad - (N - C^\alpha - C)_n.$$

We define ϕ_k to be the dihedral angle associated with N in the k^{th} residue, and we define ψ_k to be the dihedral angle associated with C^α in the k^{th} residue. A Ramachandran plot consists of the pairs (ϕ_k, ψ_k) plotted in the plane $[-\pi, \pi) \times [-\pi, \pi)$.

These dihedral angle pairs capture rotation about each of the links in a protein chain (i.e., local curvature). The density and placement of point clusters in a Ramachandran plot reveal information about low level protein structural components such as α-helices and β-sheets, but extracting precise information about higher order structures is difficult.

2.3 Tools from Shape Analysis

Many ingenious methods have been developed in the context of computer graphics for comparing three dimensional shapes [25, 36]. Here we describe one of these as it applies to space curves.

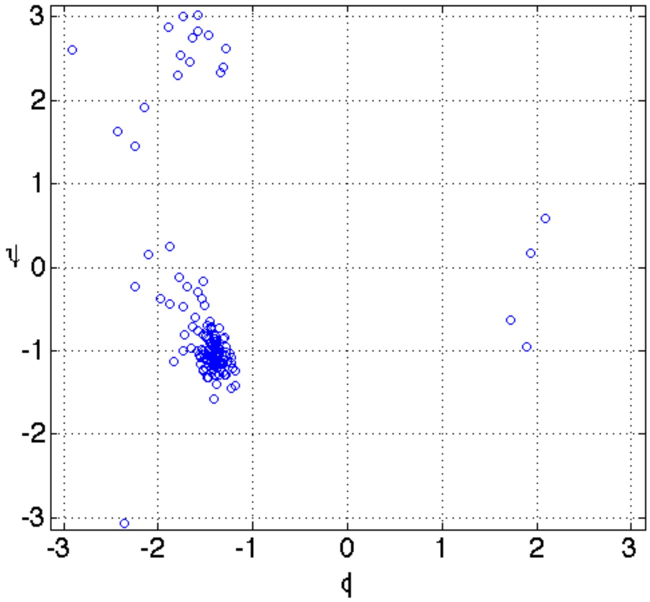
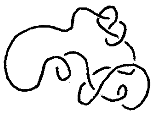


Figure 2.7. Ramachandran plot for the protein molecule with PDB identifier 5MBA.

2.3.1 Shape Distributions

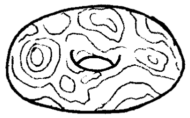
A shape distribution [30], is a histogram of the distances between pairs of points sprinkled randomly over an object. When the object is a space curve $\mathbf{x} : K \longrightarrow \mathbb{E}^3$, these distances are the image values of $d : K \times K \longrightarrow \mathbb{R}$. In the diagram below, the squiggles on the torus are level sets of the function d. Moving along one of these squiggles corresponds to moving *two* points along the space curve in such a way that the distance between the points is constant. The pairs of space curve points which are less than a distance s apart correspond to a (possibly disconnected) region on the torus. We call the percentage of the total torus area taken up by this region $A(s)$, (i.e., $A(s)$ is the fraction of the area of $K \times K$ over which d has value less than s). When s is big enough, all pairs of curve points are less than this distance apart, and so $A(s) = 1$. If s is small, a relatively low percentage of the total number of curve point pairs is less than this distance apart, and so $A(s)$ is also small. $A(s)$ is a probability distribution function. The associated density function $\frac{dA}{ds}$ is used as a shape descriptor for the space curve $\mathbf{x}$.

Curve



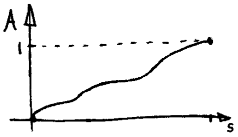
$\mathbf{x} : K \longrightarrow \mathbb{E}^3$

Distances



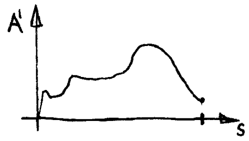
$d : K \times K \longrightarrow \mathbb{R}$
$(s_1, s_2) \mapsto \|\mathbf{x}(s_1) - \mathbf{x}(s_2)\|$

Probability Distribution



$A(s)$ is the percentage of the torus area over which $d < s$.

Probability Density



A' is short for $\frac{dA}{ds}$

Chapter 3

Intersection with Triangles

Here we consider the intersections that a space curve makes with a set of planar triangles. Each triangle has vertices *on* the space curve, and so comprises a planar region with an orientation and extent that depends on the space curve geometry; these triangles can be thought of as frames that match the local orientation of the space curve. Intersections between these triangles and other points on the curve correspond to an encirclement or wrapping of one segment of curve about another. The collective set of all these intersections is rich with information about curve shape and rich with patterns generally. In our eventual application (Chapter 7), we use these patterns as *shape descriptors* for curves; two curves with similar shapes have intersection sets that are similar. One of the most significant advantages of using intersection sets to compare curves is that they are invariant under curve translation and rotation (although curiously not under curve reflection).

3.1 Definition of an E-set

Let $\mathbf{x}$ and $\mathbf{y}$ be arc-length parameterized curve in $\mathbb{E}^3$, as illustrated in Figure 3.1. Pick a point $\mathbf{x}(s)$ on $\mathbf{x}$, and a scale value $d > 0$ such that $\mathbf{x}(s-d)$ and $\mathbf{x}(s+d)$ are defined. The points $\mathbf{x}(s-d)$, $\mathbf{x}(s)$, and $\mathbf{x}(s+d)$ are the vertices of an open triangle in $\mathbb{E}^3$ that we call the (d-scale) *encirclement triangle* based at s. If the point $\mathbf{y}(t)$ intersects this triangle, then the pair (s, t) is said to be an element of the (d-scale) *encirclement set* E_d of $\mathbf{x}$ about $\mathbf{y}$; we say that $\mathbf{x}$ at s *encircles* $\mathbf{y}$ at t, and that $\mathbf{y}$ at t is *encircled* by $\mathbf{x}$ at s.

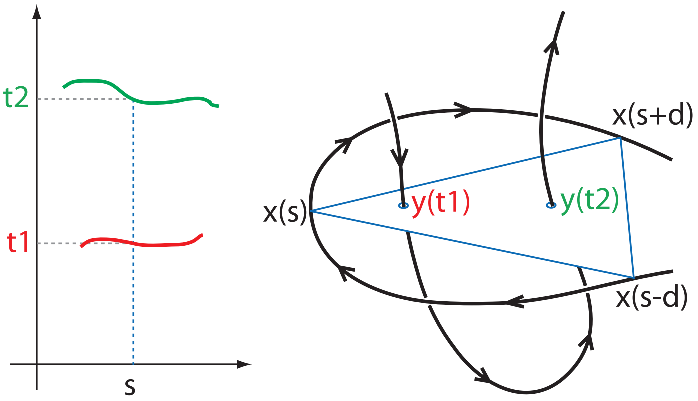


Figure 3.1. A d-scale E-set (left) gives information about the structure of the space curves (right). The point s on the E-set x-axis corresponds to the point $\mathbf{x}(s)$ on the space curve $\mathbf{x}$. The triangle with vertices $\mathbf{x}(s)$, $\mathbf{x}(s-d)$, and $\mathbf{x}(s+d)$ intersects the space curve $\mathbf{y}$ at $\mathbf{y}(t_1)$ and at $\mathbf{y}(t_2)$, and so (s, t_1) and (s, t_2) are E-set elements. The encirclement at (s, t_2) is positive, and the encirclement at (s, t_1) is negative.

An E-set is a particular type of I-set. In general, elements of an I-set (short for encirclement set) almost always comprise a finite collection of open curves (called *strands*) in the st-plane. We identify the *sign* of an E-set element $(s, t) \in E_d$ with the sign of the scalar triple product $[\mathbf{x}(s-d) - \mathbf{x}(s), \mathbf{x}(s+d) - \mathbf{x}(s), \mathbf{u}(t)]$, where $\mathbf{u}(t)$ is the unit tangent vector at $\mathbf{y}(t)$.

When $\mathbf{x}$ is an open curve, (s, t) is in E_d only if $\mathbf{x}(s)$ is at least a distance d along $\mathbf{x}$ from either endpoint of $\mathbf{x}$. This causes E_d to consist of strands in the rectangle $[d, l_x - d] \times [0, l_y]$, where the arc-length parameters for $\mathbf{x}$ and $\mathbf{y}$ take values on the intervals $[0, l_x]$ and $[0, l_y]$ respectively (see Figure 3.2).

3.1.1 Self-Encirclement

The encirclement construction is meaningful for a single curve, as we show in Figure 3.3. We identify this case by referring to it as *self-encirclement*.

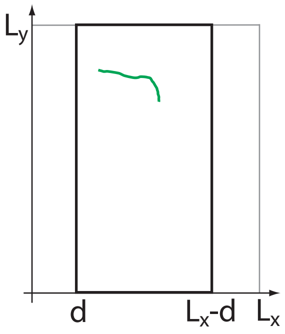


Figure 3.2. When $\mathbf{x}$ is open, E_d consists of strands in $[d, l_x - d] \times [0, l_y]$.

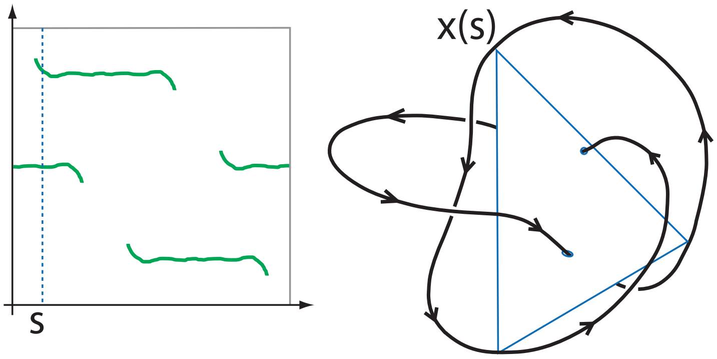


Figure 3.3. Self-encirclement of a trefoil knot. The arc-length position of the triangle vertex $\mathbf{x}(s)$ corresponds to the dotted vertical line crossing the E-set x-axis at s. The E-set strands intersecting this dotted line correspond to the two points at which $\mathbf{x}$ intersects the triangle.

3.1.2 Combining the E-sets of Multiple Curves

There are four distinct E-sets associated with the space curves $\mathbf{x}$ and $\mathbf{y}$. Using $E_d(\mathbf{x}, \mathbf{y})$ to denote the E-set of $\mathbf{x}$ with respect to $\mathbf{y}$, these four E-sets are given by $E_d(\mathbf{x}, \mathbf{y})$, $E_d(\mathbf{y}, \mathbf{x})$, $E_d(\mathbf{x}, \mathbf{x})$, and $E_d(\mathbf{y}, \mathbf{y})$. As shown in Figure 3.4, these E-sets correspond to two squares and two rectangles that can be positioned in the st-plane so as to give a single square *meta* E-set.

In general, the n^2 E-sets associated with the n space curves $\mathbf{x}_1$, $\mathbf{x}_2$, $\ldots$, $\mathbf{x}_n$ can be assembled in the st-plane to give a single meta E-set which shows the intersection properties

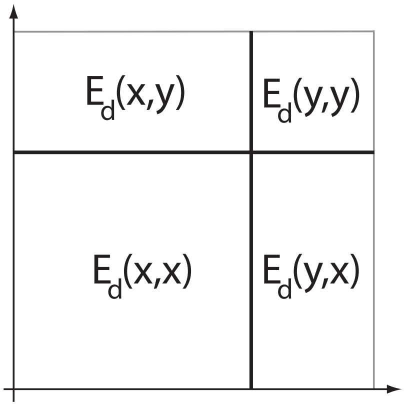


Figure 3.4. Meta E-set for curves $\mathbf{x}$ and $\mathbf{y}$.

of the collection of curves as a whole. Formally, with l_i the length of curve $\mathbf{x}_i$, and with $l_0 = 0$, we use $s_i \in [l_{i-1}, l_i]$ as an arc-length parameter for $\mathbf{x}_i$, and we construct the meta E-set by taking the union of the E-sets $E_d(\mathbf{x}_i, \mathbf{x}_j)$.

The converse to collecting different curves together is to partition a single curve into segments, and to consider the E-set for the single curve as a meta E-set which shows the intersection properties of the many curve segments. This point of view reveals the extent to which one segment of the single curve encircles or is encircled by other segments of the same curve. It is easy to imagine ways in which this construction could be useful; for instance, if the single curve $\mathbf{x}$ consists of two separate globules in $\mathbb{E}^3$ that are connected by a single strand, then the E-set for $\mathbf{x}$ will have vacant off-diagonal blocks, as illustrated in Figure 3.5.

In several industrial processes (e.g., the felting of nonwoven fabric [1]), the mobility of a segment of strand through a bulk of tangled strands is important. The degree to which a segment is encircled affects this mobility.

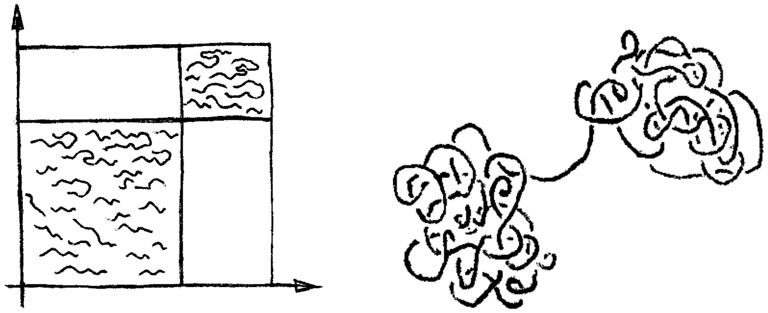


Figure 3.5. Vacant off-diagonal blocks indicate separate globules connected by a single strand.

3.2 Local Geometry

Suppose that (s,t) is an element of the encirclement set E_d of $\mathbf{x}$ with respect to $\mathbf{y}$. Let $\mathbf{a}$ be the unit vector from $\mathbf{y}(s)$ to $\mathbf{y}(s-d)$, let $\mathbf{b}$ be the unit vector from $\mathbf{y}(s)$ to $\mathbf{y}(s+d)$, and let $\mathbf{t}$ be the unit tangent vector at $\mathbf{y}(t)$.

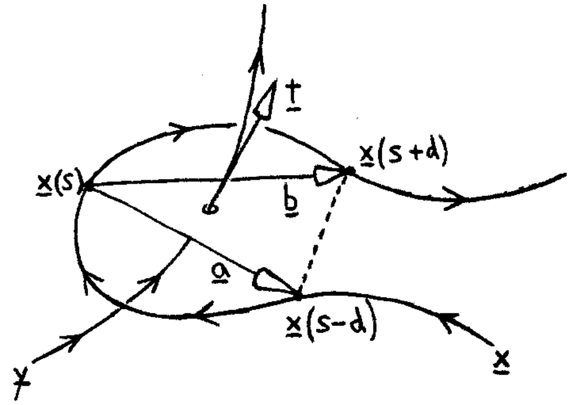


Figure 3.6. Diagram of vectors associated with $(s,t) \in E_d$.

The scalar triple product $[\mathbf{a}, \mathbf{b}, \mathbf{t}]$, and the angle made by $\mathbf{t}(t)$ and the plane of the encirclement triangle both give information about the local geometry of the encirclement at (s,t). It could be useful to display a tangled curve with segments colored according to the values of the angle and scalar triple product associated with encirclement. In Section 3.6 we propose integrating these quantities along the length of the curve to obtain a single scalar quantification of encirclement.

3.3 Global Geometry

Non-local structural properties of two curves can be deduced from their E-sets. Consider a curve **x** that coils many times around the straight curve **y**. If the radius of curvature along **x** is roughly constant (say equal to r), and if the center of curvature of **x** always is roughly at the same point on **y**, then the $2r$-scale E-set of **x** about **y** will contain a strand that is flat. In contrast, if the center of curvature of **x** moves along **y** (as it does when **x** is the red curve wrapping around the central axis of a candy cane), then the corresponding E-set strand will be tilted. If **x** coils tightly around **y** (that is, if many coils occur over a short length of **y**), then the minimum scale d at which this coiling shows up in the E-set of **x** about **y** will be less than it would be if the coils in **x** were loose.

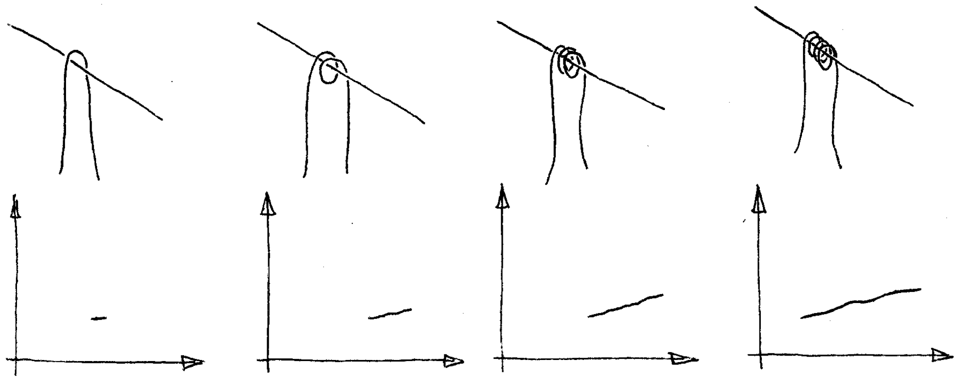


Figure 3.7. An E-set reflects the number of times one curve wraps around another.

Figure 3.7 shows how an E-set reflects the number of times one curve wraps around another, and Figure 3.8 shows how an E-set reflects the quality of the wrapping. The progression of curves in Figure 3.8 (especially from C to E) involves no cutting or major distortions. The E-set curves are discontinuous with respect to this progression, and the E-set corresponding to the middle configuration (D) in Figure 3.8 is empty. The curve in this case is on the surface of a convex cylinder, (in general, if **x** is a curve on the surface of a convex region in 3D, then its E-set is empty).

The discontinuity in Figure 3.8 motivates us to consider what would happen if we extended each E-set triangle to an infinite plane. We do this in Figure 3.9, with a family

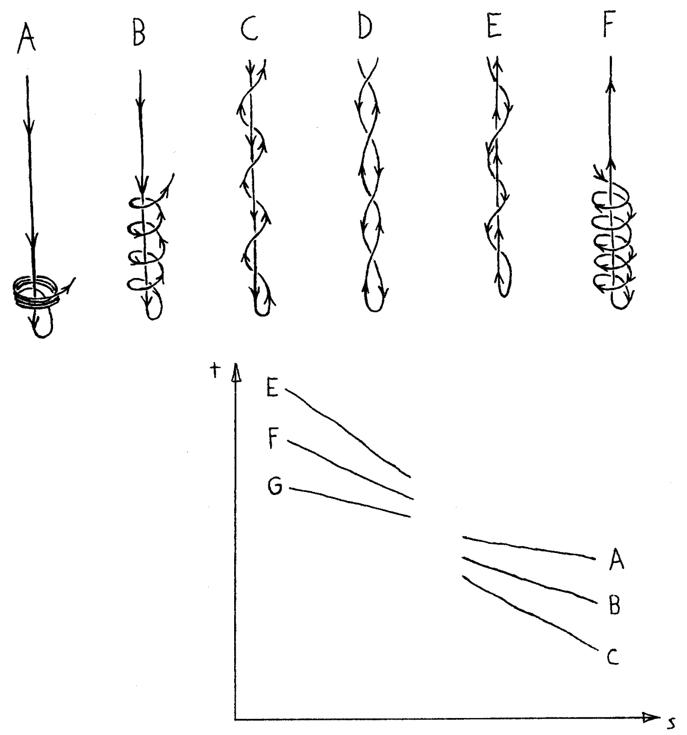


Figure 3.8. Sequence of configurations showing how the (self) encirclement diagram shows the difference between *wrapping around* and *being wrapped around.* The E-set for case D is empty.

of closed curves that undergoes a transition similar to the one in Figure 3.8. Each of these curves has the same length; the transition in Figure 3.9 involves variation in the radii of the curve helices. The resulting point sets (called P-sets for *plane*) on the *st* torus consist entirely of closed curves, and show interesting patterns and properties. For instance, it appears that only the P-set curves corresponding to the generating (E-set) triangle vertices fail to partition the torus into two distinct regions.

E-sets by construction are *subsets* of P-sets. Although E-sets suffer from discontinuities and are sometimes empty, they give direct information about whether or not one segment of curve is wrapping around another. Those P-sets we have constructed are beautiful and intriguing, however they are bewilderingly complex, and difficult to interpret.

Although an E-set contains information about curve structure, it does not contain enough to reconstruct a curve. Different curves exist with the same E-sets, for instance

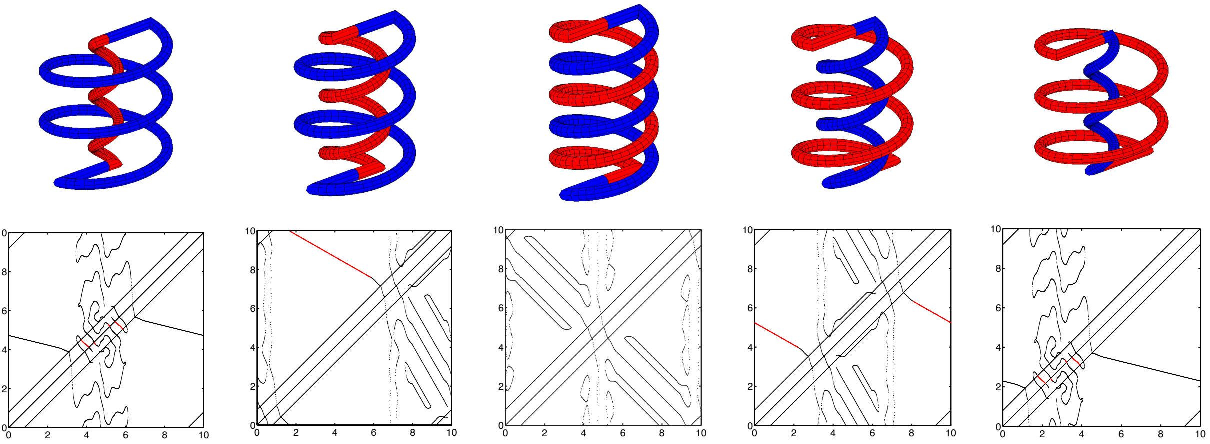


Figure 3.9. P-sets (thin black lines and thick red lines) and E-sets (thick red lines only) for a curve undergoing a transition similar to the one in Figure 3.8. As in Figure 3.8, the E-set for the middle configuration is empty.

the self E-sets for a straight line and a slightly curved line are both empty for all values of d.

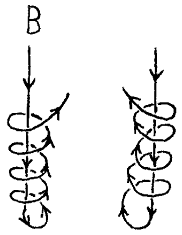


Figure 3.10. Curve B from Figure 3.8 alongside its reflection.

The *sign* of the encirclement is the same (negative) for each of the curve configurations in Figure 3.8. In Figure 3.10, we show configuration B from Figure 3.8 alongside a configuration for which the encirclement is positive. Generally, the handedness with which two curves twirl around each other follows from the sign of the associated E-set elements. The reflection of a curve changes only the sign of its E-set elements, and so two curves that differ only by a spatial reflection (for instance right and left handed trefoil knots) can be distinguished.

3.4 Encirclement Surfaces

E-sets are d-indexed cross sections of an open (often disconnected) surface embedded in a three dimensional space with coordinates s, t, and d (see Figure 3.11). We call this surface an *Esurf*. Information about encirclement and curve shape can be obtained from the intersection of the Esurf with a plane perpendicular to the d-axis (that is, an E-set).

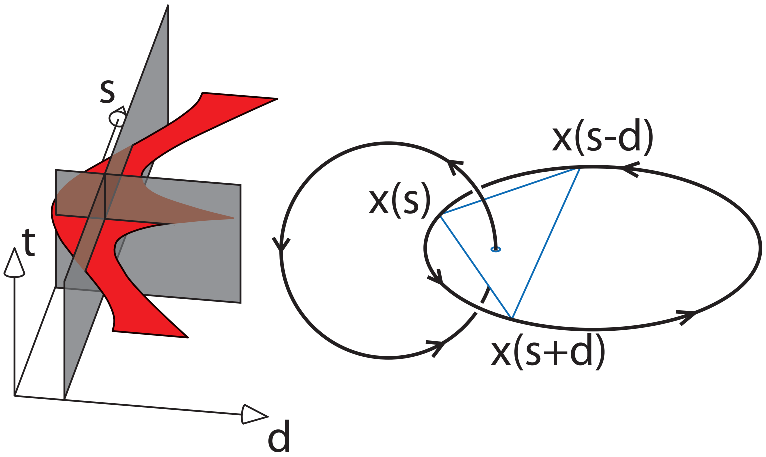


Figure 3.11. E-sets are sections of a surface embedded in $\mathbb{E}^3$. Here we show this surface for two linked circles. The intersection of the surface with the two planes at left corresponds to the intersection of the triangle and curve at right.

As with P-sets in the previous section, we can extend the Esurf construction by using the E-set triangle to define an infinite plane. We call the resulting surface in $\mathbb{E}^3$ a Psurf. Psurfs are very complicated and intricately folded; a Psurf d-section is a P-set such as those pictured in Figure 3.9. We focus in this work on Esurfs, which are subsets of Psurfs (just as E-sets are subsets of P-sets).

3.4.1 Esurf Examples

Here we consider Esurfs corresponding to a circle of radius r around a straight line a distance l from the circle center, and transverse to the plane containing the circle, (see Figure 3.12). We call the encirclement triangle edge from $\mathbf{x}(s-d)$ to $\mathbf{x}(s+d)$ the *cutting edge*. The Esurf boundary points corresponding to intersections of the cutting edge with

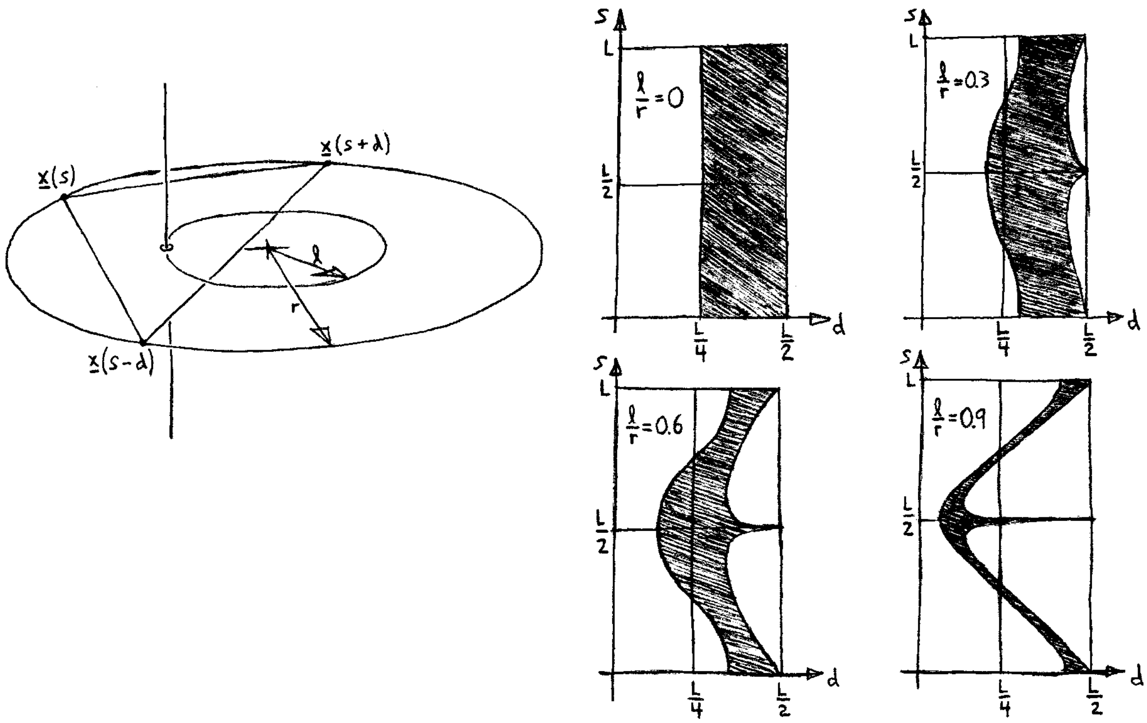


Figure 3.12. Esurfs corresponding to a straight line passing though a circle of radius r, at different distances l from its center.

the straight line are given by $r\cos\frac{d}{r} + l\cos\frac{s}{r} = 0$; the boundary points corresponding to intersections with the other two edges of the triangle are given by

$$\tan\frac{d}{2r} = \frac{l\sin\frac{s}{r}}{r + l\cos\frac{s}{r}}. \tag{3.1}$$

In this particular case, the esurf area is given by

$$A = \pi^2 r^2 - 4r\int_0^{\pi r} \arctan\left(\frac{l\sin\frac{s}{r}}{r + l\cos\frac{s}{r}}\right) ds. \tag{3.2}$$

3.4.2 Proximity

An E-set element implies proximity between different points on a curve. In detail, suppose $(s,t) \in E_d$. Let $\mathbf{a} = \mathbf{x}(s-d) - \mathbf{x}(s)$ and $\mathbf{b} = \mathbf{x}(s+d) - \mathbf{x}(s)$, and note that $\|\mathbf{a}\|, \|\mathbf{b}\| \le d$. The d-scale encirclement triangle T based at $\mathbf{x}(s)$ is given by

$$T = \{\mathbf{x}(s) + \alpha\mathbf{a} + \beta\mathbf{b} \mid \alpha, \beta > 0 \text{ and } \alpha + \beta < 1\}. \tag{3.3}$$

We are assuming that $(s,t) \in E_d$, and so it follows that $\mathbf{x}(t) \in T$ and that $\mathbf{x}(t) - \mathbf{x}(s) = \alpha\mathbf{a} + \beta\mathbf{b}$ for some α and β from (3.3). It then follows from the triangle inequality that

$$\|\mathbf{x}(t) - \mathbf{x}(s)\| \leq \alpha\|\mathbf{a}\| + \beta\|\mathbf{b}\| \leq (\alpha + \beta)d < d, \tag{3.4}$$

as we wished to show. We suspect that $\inf\{d \mid E_d \neq \varnothing\}$ is related to the normal injectivity radius r_i of a curve, however we have not proven this.

3.5 Topological Properties

Information about curve *topology* can be obtained from the intersection T of an Esurf with a plane perpendicular to the s-axis (see Figure 3.11). Holding s constant and varying d corresponds to moving the encirclement triangle edge with endpoints $\mathbf{x}(s-d)$ and $\mathbf{x}(s+d)$ through a ruled surface with boundary $\mathbf{x}$. The sum of the signed intersections of this surface with the curve $\mathbf{y}$ is the linking number of $\mathbf{x}$ and $\mathbf{y}$. These intersections correspond to a subset of the endpoints of strands in T, (like an E-set, T consists of a finite collection of open strands). In the following two subsections, we explain this in greater detail.

3.5.1 Definition of Several Esurf Related Objects

Points on the Esurf boundary correspond to the departure of a curve from the encirclement triangle. Such a departure can occur via the triangle face, or via one of the three triangle edges. A face departure for instance occurs when a loop is pulled through the triangle face as illustrated in Figure 3.13. In this case, the two points where the loop pierces the triangle move together and annihilate each other.

An edge departure can occur in one of three ways (corresponding to the three triangle edges). Our interest is in departure via the edge from $\mathbf{x}(s-d)$ to $\mathbf{x}(s+d)$, which we refer to as the *cutting edge*. With s fixed, let $\mathbf{e}(d)$ be the unit vector in the direction of $\mathbf{x}(s-d) - \mathbf{x}(s+d)$, and let p be the distance in the direction of this vector from $\mathbf{x}(s-d)$ to the departure point. Let $\mathbf{s}$ be the unit vector in the tangent plane to the spanning surface at the piercing point in the direction of increasing d, and let $\mathbf{t}$ be the unit tangent vector

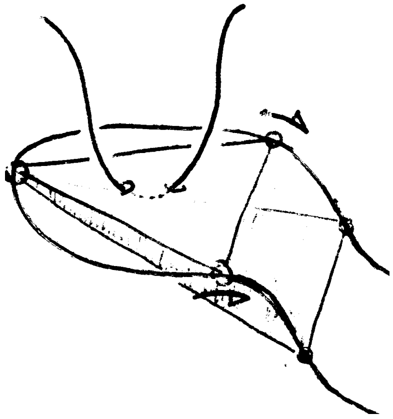


Figure 3.13. Curve departure from the encirclement triangle via the triangle face.

in the direction of the piercing curve. These vectors are illustrated in Figure 3.14. Let D (for departure) be a mapping from the Esurf boundary to $\{-1, 0, 1\}$, with -1 assigned if the triple $(\mathbf{s}, \mathbf{e}, \mathbf{t})$ is left handed, with $+1$ assigned if the triple is right handed, and with 0 assigned if the Esurf boundary point corresponds to another sort of departure (besides an intersection of the piercing curve by the cutting edge).

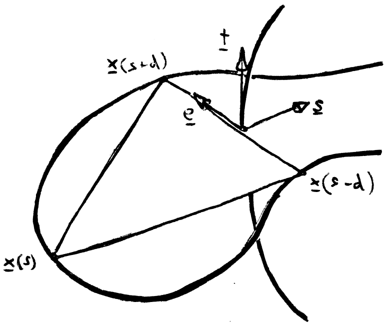


Figure 3.14. Unit vectors associated with a cutting edge departure.

3.5.2 From Esurf to Linking Number

The triple $(\mathbf{s}, \mathbf{e}, \mathbf{t})$ and the mapping D facilitate a formal expression for the linking number $Lk(\mathbf{x}, \mathbf{y})$ of two simple closed curves $\mathbf{x}$ and $\mathbf{y}$ in terms of $E_d(\mathbf{x}, \mathbf{y})$. The fixed s associated with the d indexed family of cutting edges from the previous subsection corresponds to a vertical slice of the Esurf in Figure 3.11. The linking number of $\mathbf{x}$ and $\mathbf{y}$ is the sum of the values D assigned to the endpoints of each of the line segments in this slice. We denote this sum by $B(s)$.

3.5.3 Further Observations

When $\mathbf{x}$ and $\mathbf{y}$ are different curves, $B(s)$ equals $Lk(\mathbf{x}, \mathbf{y})$ for almost every s, and so the integral

$$\frac{1}{L} \int B(s) ds \tag{3.5}$$

equals $Lk(\mathbf{x}, \mathbf{y})$. When $\mathbf{x} = \mathbf{y}$ however, $B(s)$ varies with s, and so this integral becomes nontrivial. We suspect (but haven't proved) that when $\mathbf{x} = \mathbf{y}$, (3.5) equals the self-linking number (2.5). We also suspect that the averaging accomplished by the integral corresponds to Fuller's observation that the writhing number is the mean of the directional writhing numbers. In particular, if $\mathbf{y} = \mathbf{x} + \epsilon\mathbf{d}$ for a vector $\mathbf{d}$ in general position[1] with respect to $\mathbf{x}$, then the linking number of $\mathbf{x}$ with respect to $\mathbf{y}$ equals one of the directional writhing numbers of $\mathbf{x}$. But these numbers vary dramatically with $\mathbf{d}$, implying that when $\mathbf{x} \longrightarrow \mathbf{y}$, the encirclement diagram of $\mathbf{x}$ with respect to $\mathbf{y}$ depends strongly on the way in which this limit is taken. We suspect that the number obtained from the well defined (self) encirclement diagram of a single strand $\mathbf{x}$ equals the Writhing number of $\mathbf{x}$.

3.6 Reducing Encirclement Information

An E-set is a 2D capture of space curves in 3D. Reducing a 2D E-set to something 1D (i.e., a list of numbers) or even to something 0D (i.e., a single number), is useful in applications such as curve comparison. It is a big win if we can get away with comparing lists of numbers, or even single numbers instead of directly comparing space curves in 3D. As an example, if curve A is a 5, curve B is a 300, and curve C is a 4, then curves A and C can be clustered together, away from curve B. This kind of analysis is the subject of Chapter 7.

[1] The vector $\mathbf{d}$ is in *general position* with respect to the curve $\mathbf{x}$ if $\mathbf{d}$ is not parallel to any tangent vector of $\mathbf{x}$.

3.6.1 Reduction Possibilities

One way to reduce an Eset to something 1D is to first assign a scalar to each $(s,t) \in E_d$. This can be the encirclement sign ($+1$ or -1), the angle at which $\mathbf{y}$ at t intersects the encirclement triangle based at s, or something more complex, such as the scalar triple product

$$e_d(s,t) = \left(\frac{\mathbf{x}(s-d) - \mathbf{x}(s)}{\|\mathbf{x}(s-d) - \mathbf{x}(s)\|} \times \frac{\mathbf{x}(s+d) - \mathbf{x}(s)}{\|\mathbf{x}(s+d) - \mathbf{x}(s)\|} \right) \cdot \mathbf{t}(t). \tag{3.6}$$

At each s there are almost always only a finite number of encirclement events; summing the scalars associated with these events give a 1D function in s. A further reduction (to a 0D scalar) can be accomplished by integrating the 1D function over s.

Chapter 4

Intersection with Disks

We now consider the intersections that one space curve makes with a set of disks centered on a second space curve. The disks on the second space curve form a tube, and so the appropriate image is of a small fishing line (the first curve) engaging with a thick fire hose (the tube around the second curve). This relates to current work on thick knots [31], and provides a second illustration (after triangles) of the umbrella strategy behind all our methods. (This strategy is to consider the intersections between one curve and a collection of manifolds associated with points on another curve.) For small d, the d scale encirclement triangle based at $\mathbf{x}(s)$ is roughly orthogonal to the disk centered at $\mathbf{x}(s)$, and so the intersection patterns associated with disks (this chapter) are fundamentally different from those associated with triangles (previous chapter).

4.1 Defining a D-set

Let $\mathbf{x}$ be an arc-length parameterized space curve, and let $\mathbf{u}(s)$ be the unit tangent vector to the curve $\mathbf{x}$ at the point $\mathbf{x}(s)$. Let the r-disk at $\mathbf{x}(s)$ refer to the open disk of radius r centered at $\mathbf{x}(s)$, and normal to $\mathbf{u}$. Let $\mathbf{y}$ be an arc-length parameterized space curve as well (possibly equal to $\mathbf{x}$). Define the r scale D-set of $\mathbf{x}$ with respect to $\mathbf{y}$ as the collection of ordered pairs (s, t) for which the r-disk at $\mathbf{x}(s)$ intersects $\mathbf{y}$ at the point

$\mathbf{y}(t)$. Let the sign of (s,t) be positive or negative depending on whether $\frac{d\mathbf{y}}{dt} \cdot \mathbf{u}$ is positive or negative. The D-set of $\mathbf{x}$ with respect to $\mathbf{y}$ can be plotted in the st-plane as shown in figure 4.1. As with E-sets, D-sets almost always consist of open curves (called *strands*) in the st-plane.

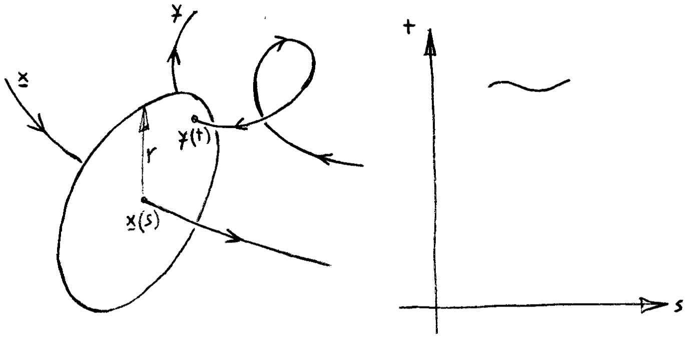


Figure 4.1. Definition of a D-set.

4.2 Symmetry

Typical D-sets for a large convoluted space curve are given in Figure 4.2. A close examination of the D-sets in Figure 4.2 reveals that none of the them are truly symmetric, however, they all exhibit a curious quasi-symmetry in which each D-set consists of symmetric *clusters* of small strands. Here we explore this phenomenon.

4.2.1 Straight Lines

Consider straight line segments $\mathbf{x}_1$ and $\mathbf{x}_2$ as shown in Figure 4.3. Assume the minimum distance d between $\mathbf{x}_1$ and $\mathbf{x}_2$ is given uniquely by $\|\mathbf{x}_1(\tilde{s}) - \mathbf{x}_2(\tilde{t})\|$. It follows that the unit tangent vectors $\mathbf{u}_1 = \frac{d\mathbf{x}_1}{ds}$ and $\mathbf{u}_2 = \frac{d\mathbf{x}_2}{dt}$ at $\mathbf{x}_1(\tilde{s})$ and $\mathbf{x}_2(\tilde{t})$ respectively define a plane. Let $\theta \in (0, \frac{\pi}{2}]$ be the angle between $\mathbf{u}_1$ and $\mathbf{u}_2$, (i.e., $\cos\theta = \mathbf{u}_1 \cdot \mathbf{u}_2$). If $r < d$, then r-disks with centers at $\mathbf{x}_1(s)$ for $s \in (\tilde{s} - l, \tilde{s} + l)$ will intersect $\mathbf{x}_2$ at $\mathbf{x}_2(t)$ for $t \in (\tilde{t} - L, \tilde{t} + L)$, where

$$l = \frac{\sqrt{r^2 - d^2}}{\tan^2\theta}, \qquad \text{and} \qquad L^2 = (r^2 - d^2)\left(\frac{1}{\tan^2\theta} + 1\right). \tag{4.1}$$

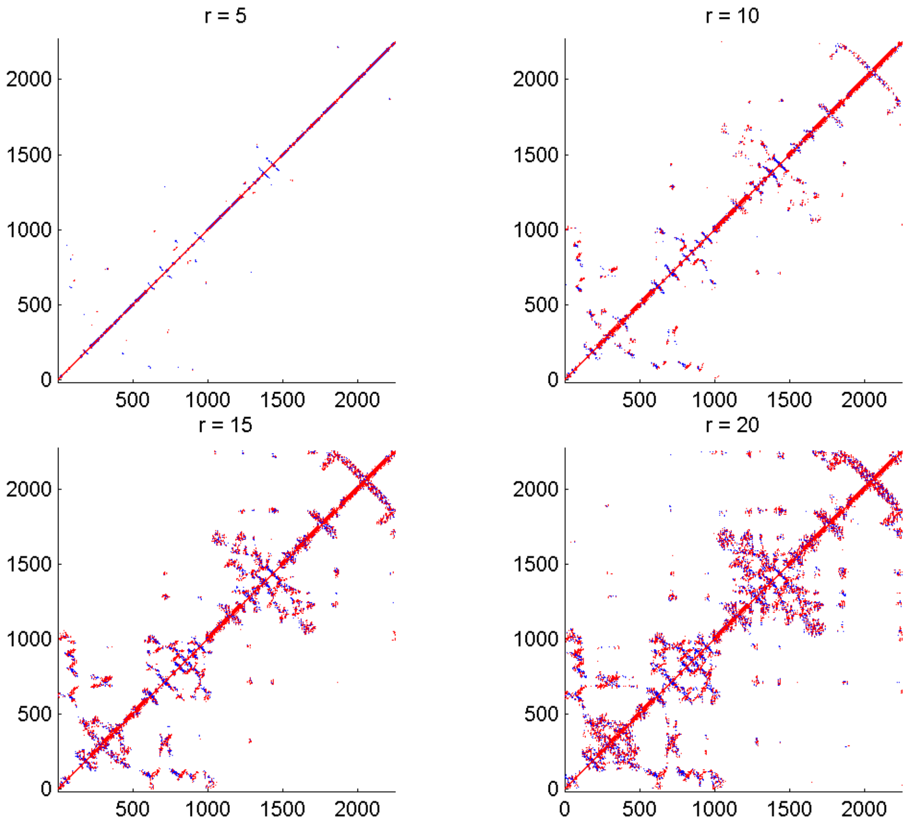


Figure 4.2. Representative D-sets for a large convoluted space curve (the protein molecule with PDB identifier 1K4Ta). Axes and r values are in units of Angstroms. Positive and negative D-set elements are colored red and blue respectively.

In fact, the D-set strand corresponding to these intersections will be the open straight line segment in the st-plane, with endpoints $(\tilde{s} - l, \tilde{t} - L)$ and $(\tilde{s} + l, \tilde{t} + L)$. In the same way, we can see that the D-set strand corresponding to the intersection of the r-disks about $\mathbf{y}$ with $\mathbf{x}$ will be the open straight line segment in the st-plane with endpoints $(\tilde{t} - l, \tilde{s} - L)$ and $(\tilde{t} + l, \tilde{s} + L)$. Usually, (when $\theta \neq \frac{\pi}{2}$), $l < L$, and so as shown in Figure 4.3, these two lines in the st-plane are *not* symmetric about the diagonal $s = t$. However, the reflection of one of these lines about this diagonal intersects the other at a point.

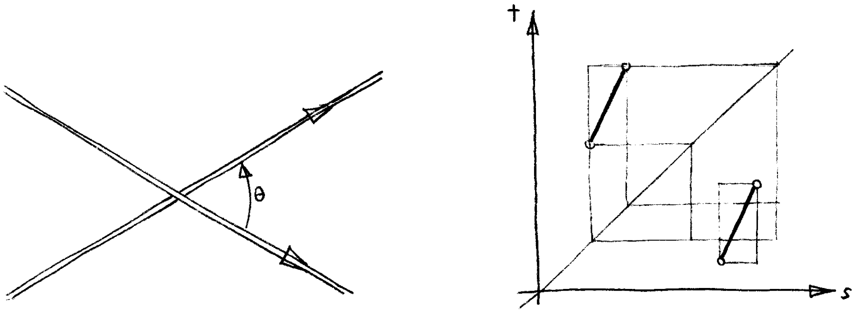


Figure 4.3. Straight line segments, both in the horizontal plane, seen from above.

4.2.2 The General Case

Let $\mathbf{x}_1$ and $\mathbf{x}_2$ be two (not necessarily straight) space curves. Here we establish the same quasi-symmetry as in this case where $\mathbf{x}_1$ and $\mathbf{x}_2$ are straight. Let $\mathcal{D}$ be the union of r-disks with centers on an open interval of $\mathbf{x}_2$

$$\mathcal{D} = \{\mathbf{x}_2(s) \mid s \in (s_a, s_b)\}. \tag{4.2}$$

Elements in the D-set of $\mathbf{x}_1$ with respect to $\mathbf{x}_2$ correspond to the intersection of $\mathcal{D}$ with $\mathbf{x}_2$. When r is sufficiently small, $\partial\mathcal{D}$ is the union of a tubular surface and the r-disks at $\mathbf{x}_2(s_a)$ and $\mathbf{x}_2(s_b)$, see Figure 4.4. As r grows, $\mathcal{D}$ can develop singularities in which a single point in $\mathcal{D}$ corresponds to the r-disks at multiple points along $\mathbf{x}_2$.

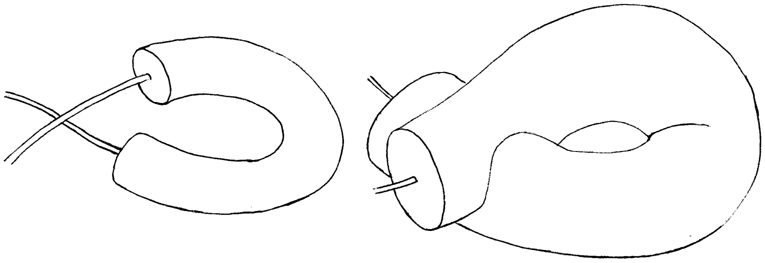


Figure 4.4. Singularities arise in the tube $\mathcal{D}$ associated with a space curve as the tube radius r increases. Recall (see 2.2) that the curve global injectivity radius r_i is the largest radius value for which $\mathcal{D}$ is singularity free.

Theorem: If the region $\mathcal{D}$ centered about $\mathbf{x}_1$ on the arc-length interval (s_a, s_b) contains the open segment $\{\mathbf{x}_2(t) \mid t \in (t_a, t_b)\}$ of $\mathbf{x}_2$, and if $\mathbf{x}_2(t_a)$ and $\mathbf{x}_2(t_b)$ belong to the tubular portion of $\partial\mathcal{D}$, then there exists a $t \in (t_a, t_b)$ and an $s \in (s_a, s_b)$ such that the r-disk centered at $\mathbf{x}_2(t)$ contains $\mathbf{x}_1(s)$.

Proof: Consider the function $d : (t_a, t_b) \longrightarrow (0, r)$ which gives the distance from $\mathbf{x}_2(t)$ to the segment $\{\mathbf{x}_1(s) \mid s \in (s_a, s_b)\}$. Both $d(t_a)$ and $d(t_b)$ equal r, and so there exists a $\tilde{t} \in (t_a, t_b)$ at which d is minimized. Let $\tilde{s}$ be a value for which $d(\tilde{t}) = \|\mathbf{x}_1(\tilde{s}) - \mathbf{x}_2(\tilde{t})\|$. Assuming differentiability, we note that the unit tangent vector at $\mathbf{x}_2(\tilde{t})$ is orthogonal to $\mathbf{x}_1(\tilde{s}) - \mathbf{x}_2(\tilde{t})$, and so $\mathbf{x}_1(\tilde{s})$ is within the normal plane to $\mathbf{x}_2$ at $\mathbf{x}_2(\tilde{t})$. But $\|\mathbf{x}_1(\tilde{s}) - \mathbf{x}_2(\tilde{t})\| < r$, and so in fact, $\mathbf{x}_1(\tilde{s})$ is in the r-disk centered at $\mathbf{x}_2(\tilde{t})$.□

By continuity, s and t from the theorem almost always belong to open subintervals of (s_a, s_b) and (t_a, t_b) respectively, which are subsets of the projection of a strand in the st-plane onto the s and t axes. It therefore follows as a corollary that every open strand in the st-plane has a reflection about the main diagonal which intersects another open strand in the st-plane.

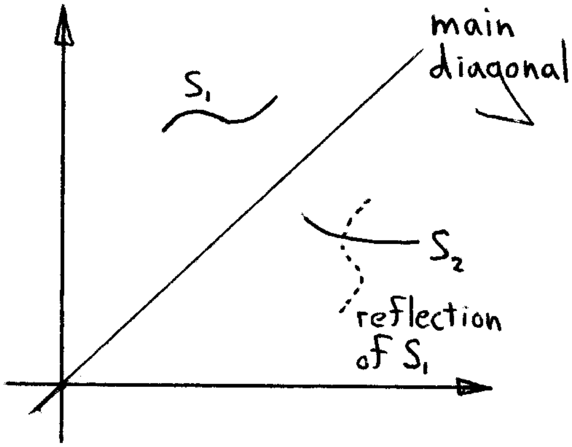


Figure 4.5. Given the D-set strand S_1, there exists a strand S_2 that intersects the reflection of S_1 about the main diagonal.

4.2.3 Breakdown of Quasi-Symmetry

In certain pathological cases, the hypotheses of the previous theorem are not satisfied, and D-set strands exist which do not have reflections about the main diagonal that intersect other strands. An example of this is given in Figure 4.6, in which an intersection asymmetry occurs between curves $\mathbf{x}$ and $\mathbf{y}$. The curve $\mathbf{y}$ is within the r-tube about $\mathbf{x}$, but the r-tube about $\mathbf{y}$ never intersects $\mathbf{x}$.

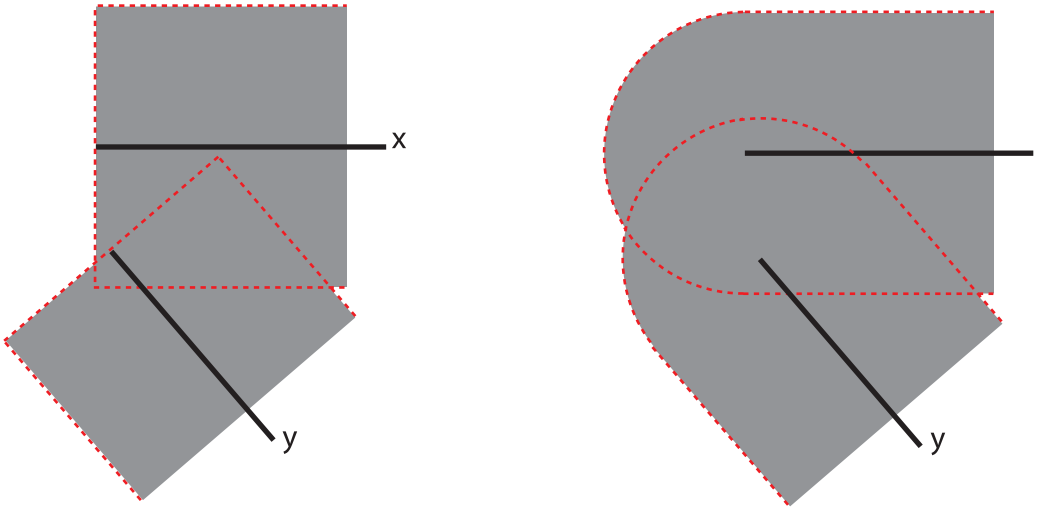


Figure 4.6. At left are two curves for which the quasi-symmetry with disks breaks down. As shown at right, this pathology vanishes when a sphere (instead of a disk) is associated with each point on the curves; a radius r sphere centered at $\mathbf{x}(s)$ contains $\mathbf{y}(t)$ iff a radius r sphere centered at $\mathbf{y}(t)$ contains $\mathbf{x}(s)$.

From the previous theorem, we know that this type of asymmetry can only occur at the end point of one of the curves involved (and hence it never occurs for closed curves). Although we don't prove this, the left image in Figure 4.6 suggests that the length of curve $\mathbf{x}$ ($\mathbf{z}$) over which this asymmetry can occur is bounded above by r ($\frac{\pi r}{2}$); to see this, let C and D coincide, and let $\mathbf{z}$ trace out a quarter circle from A to B. Thus D-set strands enjoy the quasi-symmetry property, except possibly those associated with the curve ends. These end strands will have lengths on the order of r, and will have ends that contact the (rectangular) boundary of the D-set.

As the disk radius increases, the D-set for a pair of space curves gains elements. When $\mathbf{x}$ and $\mathbf{y}$ are the same closed curve, this growth continues until the D-set strands (now living on a torus) connect with one another so that every strand is a closed loop. As with the similar construction in Figure 3.9, the totality of these strands is rich with structure, but bewilderingly complex; we leave an investigation of these objects to future workers.

4.3 Prototypical D-set Patterns

The D-set shown in Figure 4.2 is rich with patterns that correspond to structures in the generating curve (which happens to be the backbone of a protein molecule). Generally, different curve structures such as parallel segments of curve and helices give rise to distinctive D-set markings. The number and type of these structures can be immediately determined from the D-set. As shown in Figure 4.7, parallel segments generate a criss-cross pattern in the D-set. In fact, the width of the sheet of parallel segments, as well as the twist of the sheet about its axis can be determined from the D-set pattern. Also shown in Figure 4.7 is a helix and its corresponding D-set. The larger arrangement of helices generates large scale criss-cross patterns just like sheets of individual strands. Additional curve structures and their corresponding D-set patterns are shown in Figure 4.8.

4.4 Relation to Contact Maps

Contact maps are regions in the st-plane, consisting of points (s, t) for which $\|\mathbf{x}(s) - \mathbf{x}(t)\| < \epsilon$, for some $\epsilon > 0$. An ϵ contact map is a superset of a radius ϵ D-set. The two are related by a transition in which the (asymmetric) D-set strands increase in thickness so as to fill out the contact map area in the st-plane. The asymmetry of a D-set endows it with more information than the corresponding contact map.

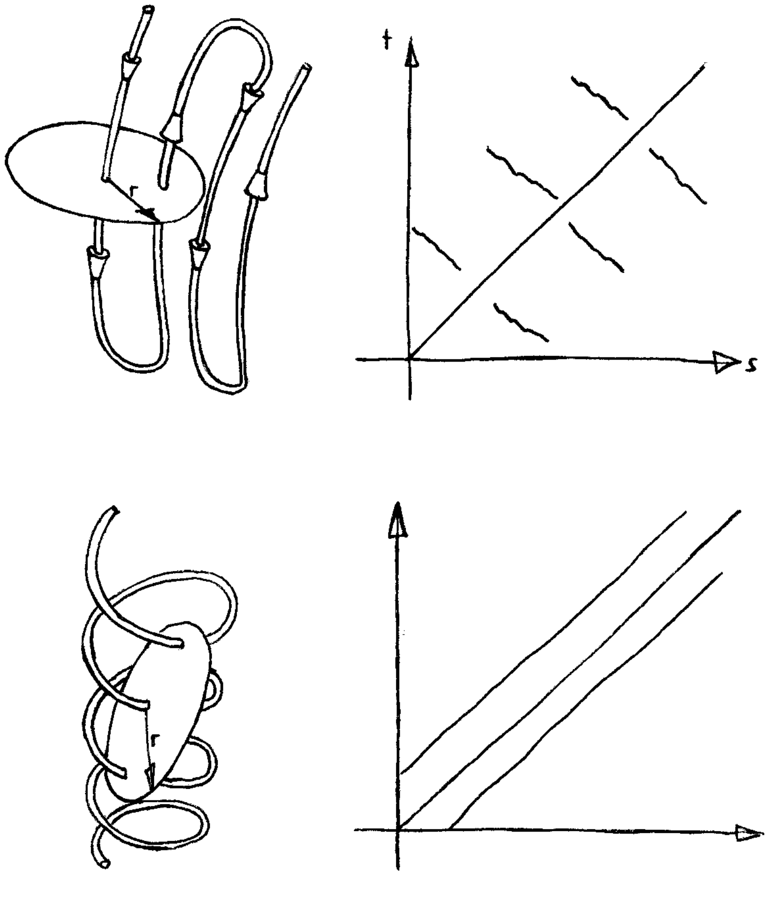


Figure 4.7. Curve structures and their corresponding D-set patterns.

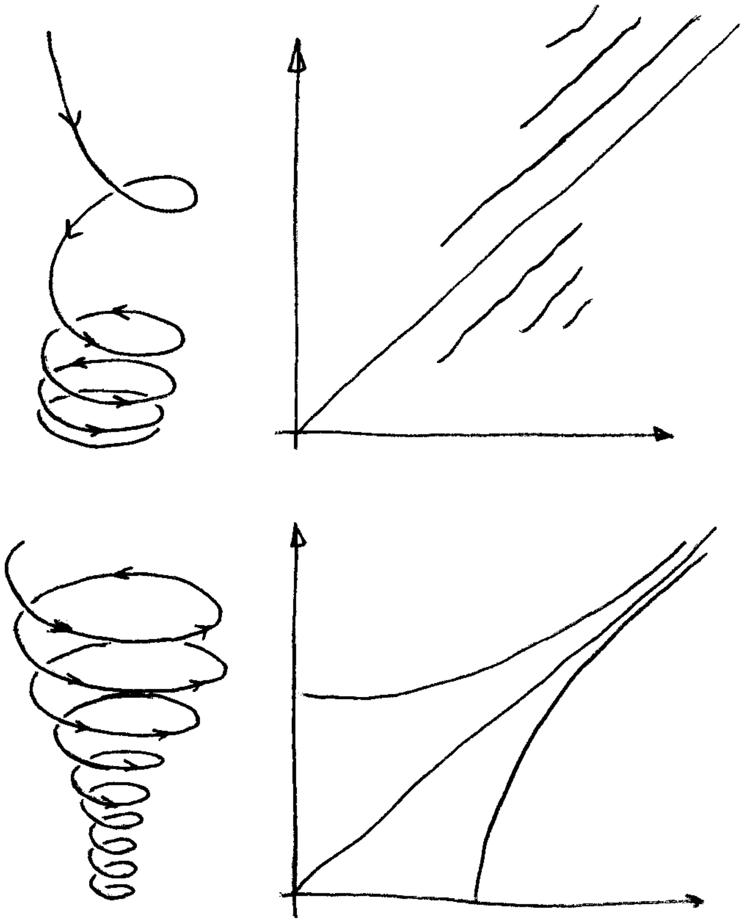


Figure 4.8. Additional D-set patterns; different curves leave signature D-set markings.

Chapter 5

Distance

Here we consider the (2D) set of distances between pairs of points on a space curve. Patterns in this set correspond to different curve structures; for instance, the distribution of these distances can be used as a shape identifier, (Funkhouser [30]).

One of our initial motivating questions was whether or not a curve can be constructed (modulo a translation and rotation) given only the distances between its points. We answer in the affirmative, and provide an algorithm for doing this section 5.2.

5.1 Distance Surfaces and Matrices

We define the *distance surface* associated with a space curve $\mathbf{x}$ to be the mapping given by $D(s,t) = \|\mathbf{x}(s) - \mathbf{x}(t)\|$. The distances between the nodes ($\mathbf{x}_i$) of a discrete space curve comprise a *distance matrix*, with entries $D_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$. Distance surfaces are pictured in Figure 5.1.

As discussed in Chapter 4, we define an ϵ contact map of a curve $\mathbf{x}$ to be the collection of points (s,t) in the st-plane for which $\|\mathbf{x}(s) - \mathbf{x}(t)\| < \epsilon$. Also, we define N_ϵ to be the number of unconnected regions in an ϵ contact map.

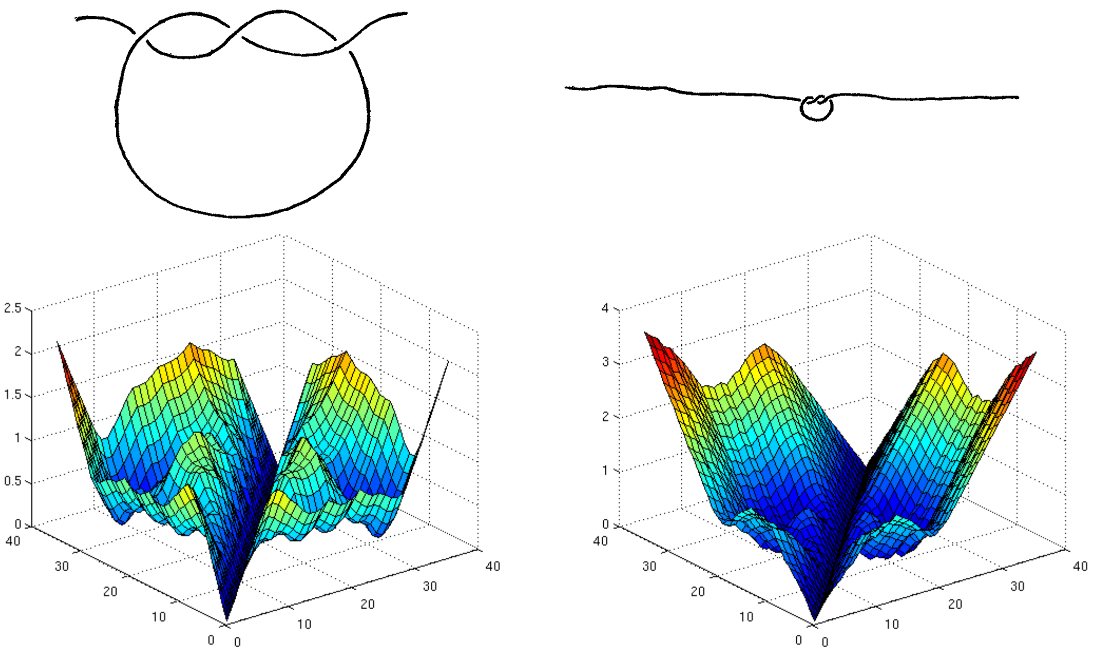


Figure 5.1. Distance surfaces for a loose knot (left), and for a tighter knot (right). The knoted curves in both cases have the same lengths. At right, the curve is largely straight, with a small knot at its midsection. In this case the distance surface consists of large planar sheets (the distance surface of a straight line is a single planar sheet).

5.2 Building Curves from Distance Surfaces

It is possible to reconstruct a curve in $\mathbb{E}^n$ given only the set of distances between points on the curve. One way to do this is to choose $n + 1$ points on the curve corresponding to the vertices of an n-dimensional tetrahedron, and then to find the positions of other points on the curve with respect to this tetrahedron. The same task is accomplished using a procedure known as Multi-Dimensional Scaling, that we discuss in Appendix A.

5.2.1 Tetrahedra

The $n + 1$ points $\mathbf{x}_0, \ldots, \mathbf{x}_n$ are the vertices of an n-dimensional tetrahedron iff the n edges $\mathbf{e}_i = \mathbf{x}_i - \mathbf{x}_0$ $(i = 1, 2, \ldots, n)$ are linearly independent. Supposing that such a tetrahedron is given in $\mathbb{R}^n$, we note that Gram-Schmidt can be used to construct an orthonormal basis $\{\mathbf{E}_i\}$ from the edges $\{\mathbf{e}_i\}$. The two sets of vectors are related by

$$\begin{bmatrix}\mathbf{e}_1\\ \mathbf{e}_2\\ \vdots\\ \mathbf{e}_n\end{bmatrix} = \begin{bmatrix}\alpha_{11} & & & \\ \alpha_{21} & \alpha_{22} & & \\ \vdots & \vdots & \ddots & \\ \alpha_{n1} & \alpha_{n2} & \dots & \alpha_{nn}\end{bmatrix}\begin{bmatrix}\mathbf{E}_1\\ \mathbf{E}_2\\ \vdots\\ \mathbf{E}_n\end{bmatrix} \quad\text{and}\quad \begin{bmatrix}\mathbf{E}_1\\ \mathbf{E}_2\\ \vdots\\ \mathbf{E}_n\end{bmatrix} = \begin{bmatrix}\beta_{11} & & & \\ \beta_{21} & \beta_{22} & & \\ \vdots & \vdots & \ddots & \\ \beta_{n1} & \beta_{n2} & \dots & \beta_{nn}\end{bmatrix}\begin{bmatrix}\mathbf{e}_1\\ \mathbf{e}_2\\ \vdots\\ \mathbf{e}_n\end{bmatrix} \tag{5.1}$$

where to find the α_{ij}'s and the $\mathbf{E}_i$'s, we first compute $\alpha_{11} = \|\mathbf{e}_1\|$ and $\mathbf{E}_1 = \alpha_{11}^{-1}\mathbf{e}_1$. Once $\mathbf{E}_1$ to $\mathbf{E}_{k-1}$ and the first $k-1$ rows of $[\alpha_{ij}]$ are known, we compute $\alpha_{ki} = \mathbf{e}_k \cdot \mathbf{E}_i$ for $i = 1, \dots, k-1$, and use

$$\alpha_{kk}\mathbf{E}_k = \mathbf{e}_k - (\alpha_{k1}\mathbf{E}_1 + \cdots + \alpha_{k\,k-1}\mathbf{E}_{k-1}) \tag{5.2}$$

to compute α_{kk} and $\mathbf{E}_k$. To compute the β_{ij}'s, we first note that $\beta_{11} = \alpha_{11}^{-1}$. Next, given the first $k-1$ rows of $[\beta_{ij}]$, we obtain the kth row from

$$[\beta_{k1}\ \beta_{k2}\ \dots\ \beta_{k\,k-1}] = -\alpha_{kk}^{-1}\begin{bmatrix}\alpha_{k1} & \dots & \alpha_{k\,k-1}\end{bmatrix}\begin{bmatrix}\beta_{11} & & & \\ \beta_{21} & \beta_{22} & & \\ \vdots & \vdots & \ddots & \\ \beta_{k-1\,1} & \beta_{k-1\,2} & \dots & \beta_{k-1\,k-1}\end{bmatrix} \tag{5.3}$$

and from $\beta_{kk} = \alpha_{kk}^{-1}$.

Our interest is in placing $n+1$ points $\mathbf{x}_i$ $(i = 0, 1, \dots, n)$ in $\mathbb{R}^n$ so that $\|\mathbf{x}_i - \mathbf{x}_j\| = d_{ij}$, where the $\frac{1}{2}(n+1)(n+2)$ positive numbers d_{ij} are given. The placement of these points is unique aside from an isometry. Let $\{\mathbf{E}_i\}$ be an ortho-normal basis for $\mathbb{R}^n$. Put $\mathbf{x}_0$ at the origin, and $\mathbf{x}_1$ at $\pm d_{10}\mathbf{E}_1$, (so that $\alpha_{11} = \pm d_{10}$), where it is our choice whether to use plus or minus. Now suppose the first $k-1$ rows of $[\alpha_{ij}]$ have been determined. The coefficients α_{ki} in the next row can be calculated for successive values of i from 1 to $k-1$ with

$$\alpha_{ki} = \frac{1}{2\alpha_{ii}}\left(d_{k0}^2 - d_{ki}^2 + (\alpha_{i1}^2 + \alpha_{i2}^2 + \cdots + \alpha_{ii}^2) - 2(\alpha_{k1}\alpha_{i1} + \alpha_{k2}\alpha_{i2} + \cdots + \alpha_{k\,i-1}\alpha_{i\,i-1})\right) \tag{5.4}$$

Then, α_{kk} is given by

$$\alpha_{kk} = \pm\sqrt{d_{k0}^2 - (\alpha_{k1}^2 + \alpha_{k2}^2 + \cdots + \alpha_{k\,k-1}^2)} \tag{5.5}$$

where again it is our choice whether to use plus or minus.

5.2.2 Curve Content

The construction technique from the previous section relies on the fact that if four points from a curve in $\mathbb{E}^3$ can be chosen so that they comprise the vertices of a tetrahedron with nonzero volume, then this tetrahedron can be used to exactly locate every other point on the curve. The corresponding fact for curves in $\mathbb{E}^2$ involves a triangle, and in general, for curves in $\mathbb{E}^N$ it involves a N-dimensional tetrahedron (often called a simplex). Sometimes however, a curve sits in $\mathbb{E}^N$ in such a way that no collection of its points comprise the vertices of an N-dimensional tetrahedron. In these cases we say that a curve has *diminished content.*

In general, the *content* of a curve is the dimension of the space spanned by the vectors between all pairs of curve points. If a curve lies entirely within a 2D plane, we say that it has content 2; if it lies on a line, we say that it has content 1. The content of a curve can be determined by picking curve points one after another so that they comprise the vertices of a tetrahedron with increasing dimension. If k is the greatest number of these points that can be chosen, then the curve has content k, and it exists in a k-dimensional slice of its original space.

Every curve has a unique distance surface, but not the other way around. A distance surface only corresponds to a curve that is unique modulo a translation and isometry. This corresponds to the fact that choices must be made in the construction of a curve from its distance surface.

Chapter 6

Using I-sets to Compare Space Curves

Space curves with similar shapes have self I-sets that look the same. When *sub*sections of two space curves are similar, the I-*sub*sets corresponding to these sections also are similar. In this chapter, we introduce a (quasi) metric on space curves that is based on a comparison of I-subsets. The metric depends on the manifold M used to generate the I-sets (e.g., triangles versus disks), so that two space curves that are different in one sense may be similar in another. Background material on metrics can be found in Munkres [28].

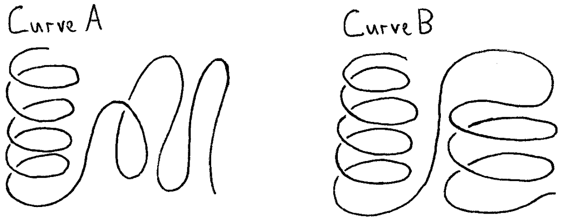


Figure 6.1. How different are curves A and B? We use I-sets to establish a notion of *qualitative distance* between space curves, (this is an abstraction of the usual notion of physical distance).

Our metric is motivated by applications across a range of fields. In structural bioinformatics, it is important to be able to compare the shapes of protein molecules, which are chains of atoms coiled into complicated space curves. Additional families of space curves that could be compared include other biological fibers such as plant tendrils, as well as solutions of the dynamical system $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$.

We find that for protein molecules, an I-set based shape comparison code is fast, accurate, and competitive with comparable state-of-the-art schemes used in structural bioinformatics. Moving from a space curve (3D) to an I-set (2D) reduces the dimension of the comparison problem, and so increases the speed of solution. In certain instances, a further reduction (to 1D) is possible, with massive computational savings.

6.1 Comparison Paradigms

Suppose that space curves A and B match reasonably well (but not exactly) over their entire lengths, and that a third space curve C has one segment that is identical to a segment of A, but also other segments which do not have any resemblance at all to any part of A. How should the distance between A and B compare to the distance between A and C? A fourth space curve D may have several segments that match segments of A, but these segments may be ordered in D differently from the order of the corresponding segments in A. These possibilities are shown in Figure 6.2.

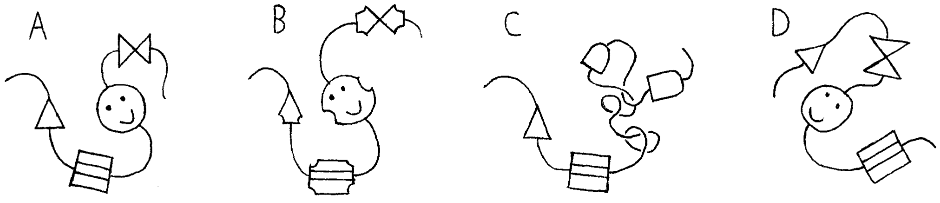


Figure 6.2. Symbolic representation of ways in which space curves A, B, C, and D may be similar. The four glyphs on each curve represent segments of curve with a particular structure.

Several decisions need to be made when comparing two space curves. Should a matching that involves numerous small gaps count for less than a matching which involves fewer but larger gaps? What if matching curve segments have to be translated and rotated different amounts in $\mathbb{E}^3$ to simultaneously coincide? There are many different ways in which two space curves can be similar, and thus many different possible metrics on the set of space curves. The appropriate metric to use depends on the problem being considered. The application guiding us as we develop our metric is the comparison of space curves associated with protein molecules.

6.2 Existing Comparison Schemes

Shape comparison and pattern recognition are fundamental tasks which have given rise to a variety of ingenious schemes [25, 30, 36]. There are many different ways to assign a *distance* to two space curves based on shape; here we mention only a few of them.

In the context of comparing protein molecules, one of the most popular algorithms for assigning an overall distance to two space curves is DALI [18], which uses Monte Carlo methods to match sub-matrices of curve distance matrices.

If a correspondence has been established between the nodes $\mathbf{x}_i$ and $\mathbf{y}_i$ of two space curves, then the distance between the two curves can be taken to be the minimum value of $\sum \|\mathbf{x}_i - (\mathbf{v} + \mathbf{T}\mathbf{y}_i)\|$ over all translations $\mathbf{v}$ and rotations $\mathbf{T}$. Finding $\mathbf{v}$ and $\mathbf{T}$ is known as the Procrustes problem, and has several well known fast solutions [35].

One of the most ingenious methods we have seen for comparing space curves involves Voroni cells [4]. Given a space curve as a collection of n nodes $\mathbf{x}_i$, the method begins with a partition of the ambient space into n Voroni cells (one containing each node). Let u_k be the number of faces on the Voroni cell that contains node $\mathbf{x}_k$. This number reflects the degree to which node $\mathbf{x}_k$ is surrounded by other nodes. Using this construction, the original 3D space curve is associated with a 1D list of scalars (u_k). Two space curves are compared by comparing these corresponding lists. Because the lists are 1D, this comparison can be accomplished with a (fast) dynamic programming algorithm, such as the one in Appendix C.

6.3 Our Scheme

In our scheme, we chop space curves A and B into small segments a_i and b_j, and compare all possible pairs of I-subsets for these segments. Essentially, a non-repeating list of segment pairs (a_i, b_j) is chosen so that the sum of the associated distances is a minimum. A flowchart of this scheme is given in Figure 6.8. The idea for non-pathological cases is that agreement in shape on a local scale (in the sense of which ever I-set we happen to be using) is used to string together the curve segments a_i and b_j into larger pieces that have the same shape. Because the method is based on comparing small adjacent segments, it is insensitive to *low frequency* differences between two curves, in which small coilings may agree, but the larger structures drift apart, (i.e., it is a high pass filter for shape.)

6.4 Parameterization Invariance

The I-set of a space curve is affected by the way in which the curve is parameterized. Our interest is in constructing a distance function for space curves that is insensitive to this. In general, a shift ($s \mapsto s + \Delta$) in the arc-length parameterization of curves $\mathbf{x}$ and $\mathbf{y}$ causes the I-set of curve $\mathbf{x}$ with respect to curve $\mathbf{y}$ to translate in the st-plane. In the case of self I-sets, this translation is only in one direction (that of the main diagonal $s = t$). A change in the sign of an arc-length parameter ($s \mapsto -s$) corresponds to a change in the direction associated with a space curve. This causes the associated I-set to undergo two reflections (i.e., a 180° rotation) in the st-plane.

In Section 6.5.1 and B we present two I-set metrics, and different methods for making them insensitive to translation. Insensitivity to reflection can be accomplished by checking the small finite number of possible cases. The protein molecules that we investigate in Chapter 7 have directed backbone curves, and so we don't bother correcting for reflections. (Also, D can be evaluated separately for the positive and negative parts of two I-sets, however we don't persue this.)

6.5 I-set Metrics

We now provide two metrics for quantifying the difference between I-sets. Later in this chapter, we apply these to I-subsets (note that an I-subset is itself an I-set) corresponding to segments of two curves, and then bootstrap our way to a single distance value, together with alignment information for the two curves. We are looking now for local matches; we want to know if the curve $\mathbf{x}$ in a small but not too small neighborhood of $\mathbf{x}(s)$ has the same shape as the curve $\mathbf{y}$ in a neighborhood of $\mathbf{y}(t)$.

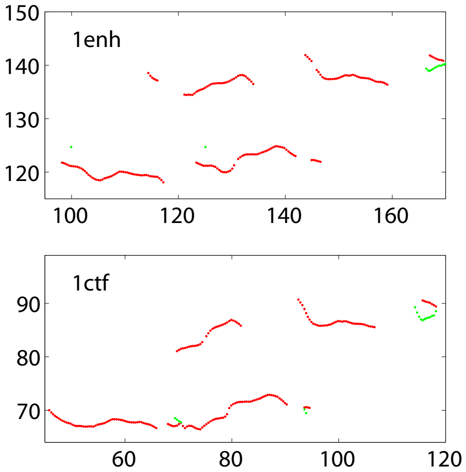


Figure 6.3. Encirclement subsets of backbone curves for protein molecules with PDB identifiers 1ENH and 1CTF, (these images are magnifications of the boxed regions in Figure 7.3.) The similarity in these E-sets corresponds to a similarity between the associated protein substructures. The axes indicate position (in Angstroms) along the protein backbone curves.

6.5.1 Metric A

Let A and B be two I-sets, and let $f_B : A \longrightarrow \mathbb{R}$ map $\mathbf{a} \in A$ to

$$f_B(\mathbf{a}) = \inf\{\|\mathbf{a} - \mathbf{b}\| \mid \mathbf{b} \in B\}. \tag{6.1}$$

This function tells how far a single element of A is from all the elements of B. A distance function $D(A,B)$ is obtained by integrating $f_B(\mathbf{a})$ along the strands comprising A and by integrating $f_A(\mathbf{b})$ along the strands comprising B, with respect to arc-length parameters along these strands:

$$D(A,B) = \frac{1}{L_A + L_B} \left(\int_A f_B(\mathbf{a}(\tau))\, d\tau + \int_B f_A(\mathbf{b}(\tau))\, d\tau \right). \tag{6.2}$$

The result is normalized by $L_A + L_B$, where L_A and L_B are the total lengths of the strands in A and B respectively. $D(A,B)$ partitions the strands in A and B into (vanishingly) small segments of equal length. Each segment in A (B) is assigned the distance from that segment to the nearest element in B (A). The function $D(A,B)$ returns the mean of these distances.

If A and B are unequal, then $D(A,B) > 0$. For instance, if A consists of the vertical line l_A from $(0,-1)$ to $(0,1)$, and if B consists of the horizontal line l_B from $(-1,0)$ to $(1,0)$, then $D(A,B) = 1/2$. As A and B get closer together, for instance as l_A rotates about the origin and becomes increasingly aligned with l_B, $D(A,B)$ gets smaller. If A and B are equal then $D(A,B) = 0$, and conversely.

If the I-sets A and B are contained in disks in the st-plane, then $d_{min} < D(A,B) < d_{max}$, where d_{min} is the diameter of the largest circle that can pass between the disks, and where d_{max} is the diameter of the smallest circle that contains the disks.

Unfortunately, although D is positive definite (and trivially symmetric), it is not a true metric because it sometimes violates the triangle inequality. For instance, when A, B, and C are I-sets consisting of the open intervals on $\mathbb{R}$ given by $(-1,0)$, $(-1,1)$, and $(0,1)$ respectively, $D(A,B) + D(B,C) = 1/3$, and $D(A,C) = 1/2$. It is instead a *quasi*-metric.

6.5.2 Minimizing the Metric A over Translations

Here we discuss finding the minimum value of $D(A, B + \mathbf{x})$, where $B + \mathbf{x}$ stands for a translation (but no rotation) of B in the st-plane. We think of the corresponding parts of A and $B + \mathbf{x}$ as connected by springs. This motivates an ODE for $\mathbf{x}(\gamma)$ which causes

$D(A, B + \mathbf{x})$ to approach a local minimum as γ increases. In particular, let

$$\frac{d\mathbf{x}}{dt} = \frac{1}{L_A + L_B}\left(\int_A \mathbf{g}_B(\mathbf{a}(\tau))d\tau - \int_B \mathbf{g}_A(\mathbf{b}(\tau))d\tau\right), \tag{6.3}$$

where $\mathbf{g}_A : B \longrightarrow \mathbb{R}^2$ such that $\mathbf{b} \in B$ is mapped to the average of the vectors from $\mathbf{b}$ to the points $\mathbf{a} \in A$ for which $\|\mathbf{b} - \mathbf{a}\|$ equals $f_A(\mathbf{b})$. The flow associated with this ODE has basins of attraction with boundaries given by the ridge lines of the function $D(A, B + \mathbf{x})$ of $\mathbf{x}$. A strategy for exploring all basins would be to estimate the minimum size of the basins based on properties of A and B, and then to place a grid small enough over $\mathcal{D} \subset \mathbb{R}^2$ (where $\mathcal{D}$ corresponds to $\mathbf{x}$ for which $A \cap (B + \mathbf{x}) \neq \varnothing$) to ensure that there is at least one trajectory in every basin.

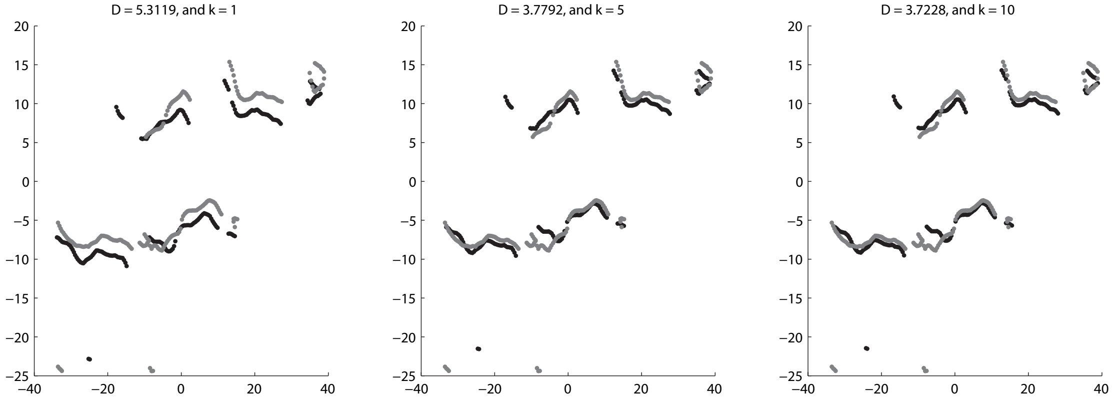


Figure 6.4. Algorithm translates two E-sets (from Figure 6.3) so that the distance D between them is minimized. We start the algorithm with the E-sets centered about the origin. The D values asymptote to approximately 3.72 Å.

It is only appropriate to minimize over arbitrary I-set translations if each of the two I-sets being compared is for two *different* curves, (i.e., neither I-set is a *self* I-set). If A is the I-set for curve $\mathbf{x}_A$ with respect to curve $\mathbf{y}_A$, and if B is the I-set for curve $\mathbf{x}_B$ with respect to curve $\mathbf{y}_B$, then a match between A and B indicates that $\mathbf{x}_A$ is to $\mathbf{y}_A$ as $\mathbf{x}_B$ is to $\mathbf{y}_B$. As mentioned in 6.4, a shift in the arc-length parameterizations of these four curves can cause A and B to translate arbitrarily in the *st*-plane. Making the distance between A and B invariant to these shifts involves minimizing over arbitrary translations. Sign changes in the curve parameterizations cause the corresponding I-sets to reflect in the *st*-plane, further complicating the problem. Four curves have a total of $2^4 = 16$ reflection

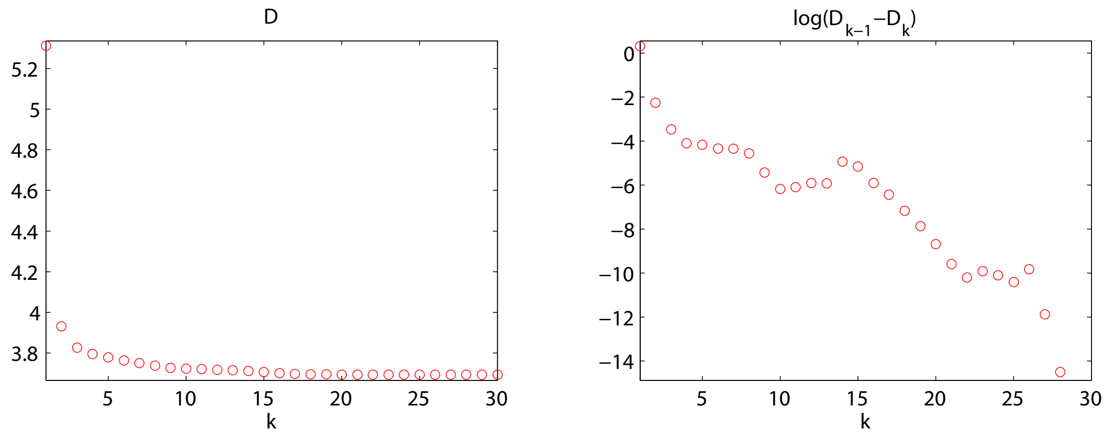


Figure 6.5. It seems that the algorithm for minimizing D for E-sets over all translations always reduces the value of D with each iteration (we have not proven this yet). In practice, we run the algorithm for 10 iterations. This data is from the E-sets in Figure 6.4.

possibilities, and so in general, we would have to minimize over arbitrary translations for each of these separately, and then take the minimum of these 16 minima.

The situation is considerably simpler in the case that interests us, in which each I-set is a *self* I-set. In this case, shifts in the arc-length parameterization of a curve translate its self I-set along the main diagonal of the st-plane. With $\mathbf{e}$ a unit vector in this direction, our search is for an optimal translation of the form $\gamma\mathbf{e}$. An ODE which provides γ follows from the projection of (6.3) in the direction of $\mathbf{e}$. In particular,

$$\frac{d\gamma}{dt} = \frac{1}{L_A + L_B}\left(\int_A \mathbf{g}_B(\mathbf{a}(\tau))d\tau - \int_B \mathbf{g}_A(\mathbf{b}(\tau))d\tau\right)\cdot\mathbf{e}. \tag{6.4}$$

A change in the sign of an arc-length parameter causes the associated I-set to undergo two reflections, equivalent to a 180° rotation in the st-plane. The number of possibilities is only two.

6.5.3 Metric B

A natural metric for regions A and B in the plane is given by

$$D(A,B) = \frac{\text{Area}(A\cup B)}{\text{Area}(A\cap B)} - 1, \tag{6.5}$$

which returns a zero if and only if $A = B$, and which is greater than zero otherwise. If we wish to detect when a small region exactly matches some piece of a larger region, we might consider using

$$D(A,B) = \frac{\min\{\text{Area}(A), \text{Area}(B)\}}{\text{Area}(A \cap B)} - 1, \tag{6.6}$$

which is zero if $A \subset B$ or $B \subset A$. Metrics (6.5) and (6.6) apply to I-sets by way of the construction shown in Figure 6.6.

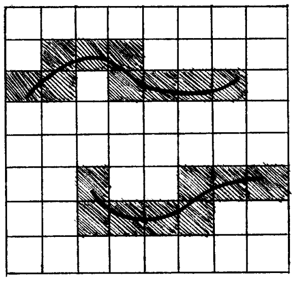


Figure 6.6. Turning curves in the st-plane into matrices.

In detail, let $(s_0, s_1, \ldots, s_m)$ and $(t_0, t_1, \ldots, t_n)$ be equally spaced points on the intervals $[s_0, s_m]$ and $[t_0, t_n]$ respectively, (where all I-set points have a first element on the s-interval, and a second element on the t-interval). Let the (i, j) entry of the $n \times m$ matrix M equal 1 if any I-set element is in the box $[s_{j-1}, s_j) \times [t_{i-1}, t_i)$, and let it equal 0 otherwise. This construction turns a collection of curves in the plane into a region with area. This region can be compared to other such regions using (6.5) and (6.6). In practice, we store these regions as large sparse logical arrays which take up very little space and are easy to work with.

As with metric A, we can introduce a fine tuning alignment of (sparse matrices) A and B so as to obtain the minimum possible distance between them. With self I-sets this is a matter of shifting the indicies (row and column) of one of the matrices by the same integer amount. We start so that the mass centers of the matrices coincide, and then search some distance in both directions away from this.

6.6 Scaled Distance

The distance D between two I-sets gives either the average length of the line connecting an element from one I-set to the nearest element of the other I-set (in the case of metric A), or a ratio of overlapping areas (in the case of metric B). We wish to reduce the distance between two I-sets that occupy a large region in the *st*-plane; the larger this region, the greater the potential for I-set mismatch with respect to metrics A and B, and thus the greater the similarity between I-sets that don't mismatch. A scaled distance D_s that accomplishes this is given by

$$D_s(A, B) = \frac{D(A, B)}{box(A \cup B)} \tag{6.7}$$

where $box(A \cup B)$ reflects the size of the smallest rectangle in the *st*-plane that contains $A \cup B$. In the case of metric A, $box(A \cup B)$ is the length of the diagonal of the smallest rectangle that contains $A \cup B$. In the case of metric B, $box(A \cup B)$ is the area of the smallest rectangle that contains $A \cup B$, *which is bounded by gridlines of the partition used to define* D.

Consider the upper left and lower right images in Figure 6.7. There is a stronger match between the black and gray I-sets in the upper left case than in the lower right case. This is reflected in the scaled distance values D_s.

6.7 Overall Alignment

Here we construct an overall match between two curves by using I-subsets to match curve subsections. By matching I-subsets that contain *enough* information about structural correspondance on a scale that is small but not too small, we obtain an overall match of the two structures. Our measure of the global match between two curves tolerates drift in the relative position of curve pieces that are far apart with respect to arc-length.

At this point we have a matrix of pairwise comparison scores. Each entry reflects the similarity between two segments of protein. From this matrix we construct an overall pairing for two proteins. Parts that match are paired, and parts that do not match are left free.

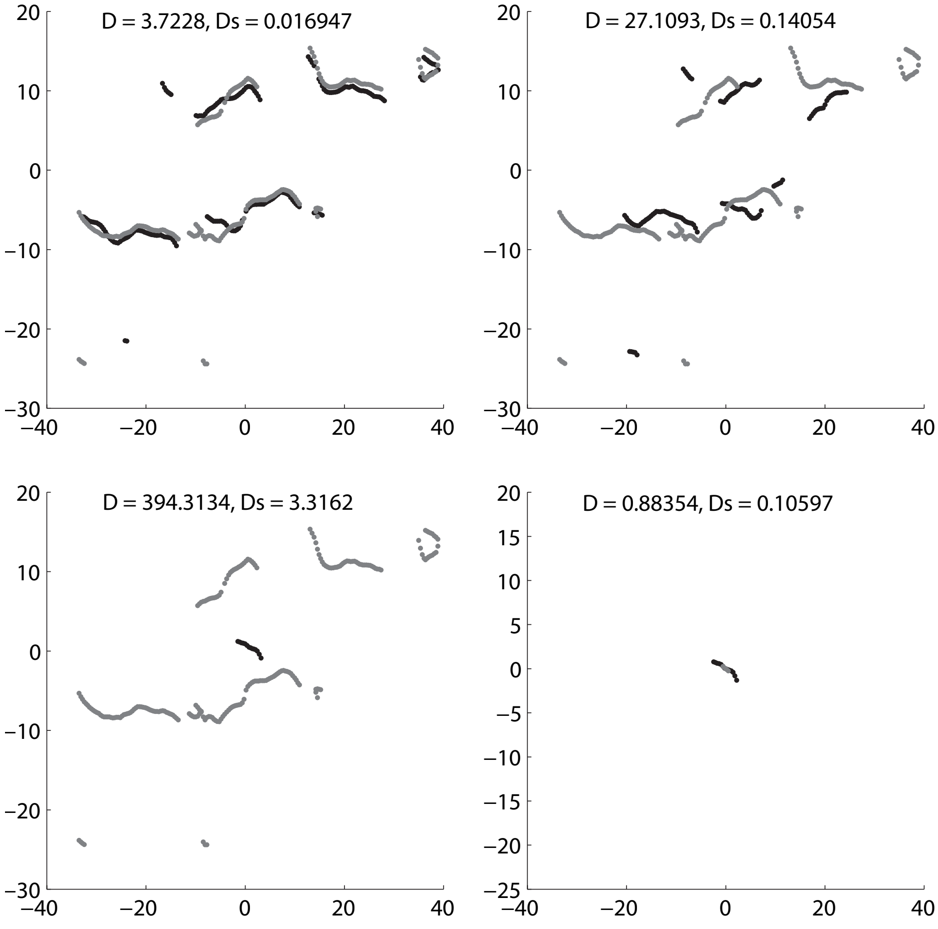


Figure 6.7. Examples of distance values D and scaled distance values D_s for different I-set pairs.

6.7.1 Sequential Alignment

Sequential alignment is discussed in Appendix C. In general, if the direction in which two curves match is not known ahead of time, sequential alignment would be performed on A and B, and then on A and $\tilde{B}$, where $\tilde{B}$ is B taken in the reverse direction. Sequential alignment is what we seek in our work with protein molecules, (protein molecules are rarely aligned non-sequentially). The assumption of sequential alignment greatly reduces the cost of computing the alignment.

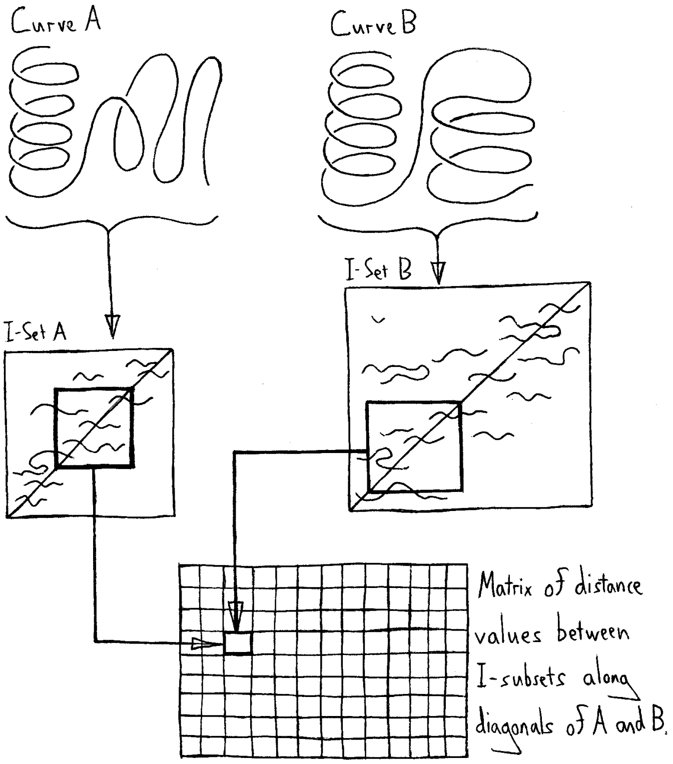

Curve A
Curve B
I-Set A
I-Set B
Matrix of distance values between I-subsets along diagonals of A and B.

Figure 6.8. Flow chart for comparing two space curves.

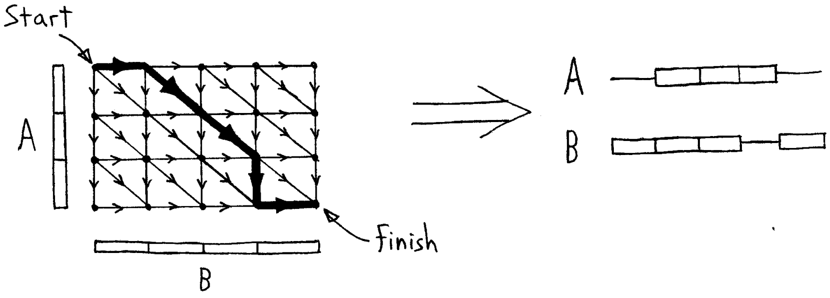


Figure 6.9. Distance values from the end matrix in Figure 6.8 are weights in a path optimization scheme.

6.7.2 Non-Sequential Alignment

Non-sequential alignment is expensive, however it accommodates a wider variety of possibilities than sequential alignment. For instance it can accommodate reversals in direction, as in [18]. In the case of two curves that are sequentially aligned, this matching gives the same results as the sequential alignment matching. The Kuhn-Munkres algorithm (see Appendix B) solves for this kind of alignment. We suggest the following overall strategy:

1. compute two matrices of pairwise comparison scores, $M(A, B)$ and $M(A, \tilde{B})$. Then, put the element-wise minima of these two matrices into a single matrix called M.

2. remove rows and columns from M in which all entries are larger than a threshold value.

3. run the Kuhn-Munkres algorithm on M.

4. define *matching segments* be continuous diagonal sets of (two or more) elements in M. two such sets in M with different orientations should have this reflected in their individual elements. one set should be for A and B, while the other should be for A and $\tilde{B}$. this alternation should hold for *all* matches.

6.8 Overall Distance

An alignment between curves A and B establishes a correspondence between points on A and points on B. Every point of A (B) is connected either to nothing, or to a unique point on B (A). Given an alignment between two curves, there are several associated notions of

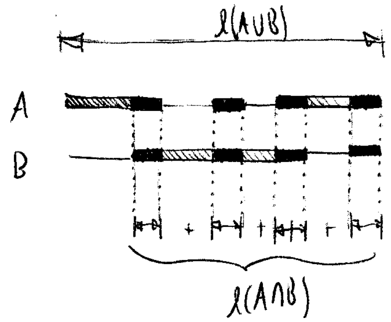


Figure 6.10. This diagram represents curves A and B in the case of a sequential alignment. The curves are uncoiled, and gaps are introduced to illustrate the alignment. $l(A \cup B)$ and $l(A \cap B)$ are measures of arc-length along the curves.

overall distance. The most obvious of these is simply the score S_A associated with the alignment. If we wish to measure alignment *quality per overall length*, we could use

$$D(A,B) = \frac{S_A}{l(A \cup B)}, \text{ or } D(A,B) = \frac{S_A}{l_A + l_B}, \tag{6.8}$$

where S_A is the score associated with the alignment, where l_A and l_B are the lengths associated with A and B respectively, and where $l(A \cup B)$ is the overall alignment length, as shown in Figure 6.10. If we wish to ignore pieces of curve that are unmatched in the alignment, we could evaluate the score associated with matched segments divided by the segment lengths. Finally, if we wish to measure the proportion of curve length which is involved in the alignment, we could use

$$D(A,B) = \frac{l(A \cup B)}{l(A \cap B)} - 1 = \frac{l_g}{l(A \cap B)}, \tag{6.9}$$

where $l(A \cap B)$ is the net overlap length (see Figure 6.10), and where $l_g = l(A \cup B) - l(A \cap B)$ is the gap length associated with the alignment.

These different notions of distance (as well as many others) give different relative distances between curves. The appropriate distance function to use depends on what needs to be measured. In our work with proteins (in Chapter 7), we wish simply to verify that I-sets enable curves in different SCOP superfamilies to be distinguished. For this purpose we use (6.8b).

Chapter 7

Protein Molecules

As a real world application of our novel shape analysis methods, we now use I-sets to distinguish between different protein molecules. Protein molecules are the building blocks of life, performing myriad functions in living organisms. Understanding these molecules is a fundamental challenge which underpins work in medicine, such as the development of new drugs. The first protein molecule to have its structure solved was Sperm Whale myoglobin, in 1953. Since then, thousands of solved protein structures have been added to an online repository called the Protein Data Bank (PDB). Being able to navigate the 40,000 structures now residing in the PDB is essential to further developments in the field. For instance, the process of solving for new protein shapes has come to rely heavily on comparison of an emerging structure with existing solved structures.

Navigating a set of objects depends on having a meaningful notion of distance. This comes from a metric, which tells which objects are close together, and which objects are far apart. It is often useful to have more than one metric on the same set of objects; two things may be close together in one sense, and far apart in another. With respect to a particular metric, objects can be collected into groups and therefore classified, [32, 38].

A multitude of classification schemes deepens our understanding of a set of objects. There are three major schemes for classifying proteins by their structure. These differ mainly in their level of automation (especially for distinguishing different protein domains). The fully automated scheme is FSSP (Families of Structurally Similar Proteins) based on DALI, the semi-automated scheme is CATH, and the largely manual scheme is SCOP (Structural Classification of Proteins). The references in [11] include papers on these schemes, as well as on many additional innovative approaches to protein classification.

In this chapter we show that protein structures give rise to different I-set patterns. We use the comparison algorithm from Chapter 6 to find *distances* between protein molecules, and thus to classify them into groups. The protein structure space is filled with different patterns and possible methods of grouping. Different organizational schemes are appropriate for accomplishing different tasks. Given a new protein structure, we wish to quickly find existing proteins that are structurally similar.

7.1 Protein Structure

Protein molecules consist of chains of atoms coiled into compact hierarchically structured curves in $\mathbb{E}^3$. The *primary structure* of a protein refers to the order along the protein chain of groups of atoms called residues. Each residue has the following form

$$C - N - C^{\alpha} \tag{7.1}$$

with a distinguishing appendage linked to N. There are 20 naturally occuring residues, and like letters in a sentence, their order along a protein chain determines the shape and function of the protein molecule. The local structural patterns that arise within the protein are called *secondary structures.* These include helices (called α-helices), as well as parallel (and anti-parallel) portions of chain that form what are called β-sheets. Additional secondary structures include *turns* and *loops.* So called *super-secondary structures* consist of common combinations of α-helices and β-sheets. These secondary structures in turn are arranged in space into *tertiary structures.* Frequently, tertiary structures combine to form even bigger complexes, called *quatenary structures.* Proteins with quatenary structure are often modular, with the same subunits repeated multiple times.

This hierarchy in proteins is analogous to the levels of structure in written text. Just as

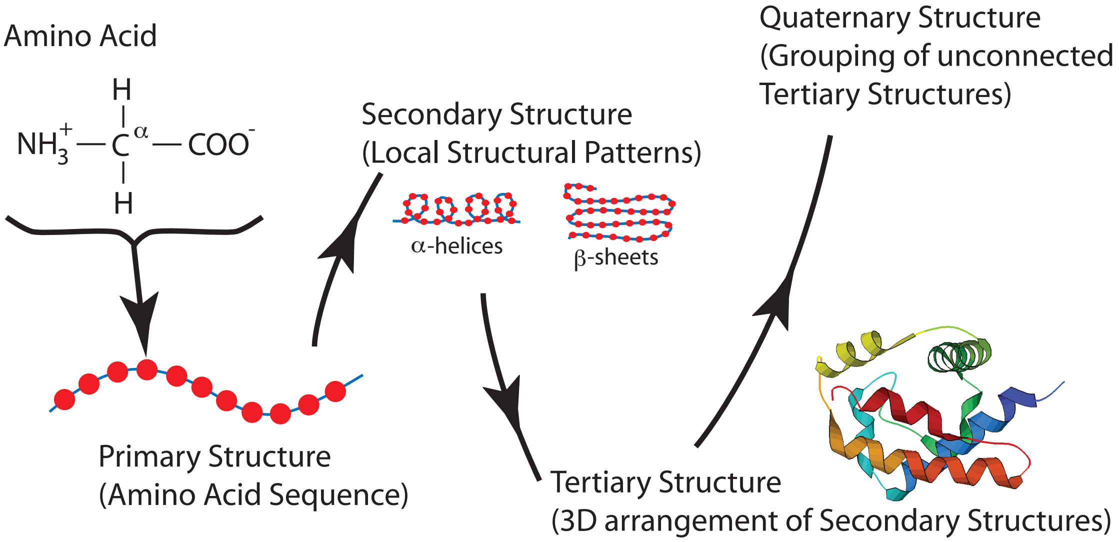


Figure 7.1. Proteins exhibit a hiearchy of organizational levels.

the meaning of a sentence or paragraph can often be deciphered even if numerous letters are changed, the structure and thus function of a protein molecule is (usually) robust to changes in individual residues. Furthermore, just as there are often many ways to say the same thing in text, there are often different residue sequences that result in the same protein structure. The rules of protein grammar and sentence construction allow protein identity to be deciphered with only partial information. Although protein curves appear to be hopelessly convoluted (see Figure 1.1), they are actually rich with supporting patterns. The functional identity of a protein largely depends on its shape. Also, the same primary structure can lead to different stable final structures (although this is rare). Further background can be found in [24, 34].

7.2 Organizing Proteins by Shape

Patterns have emerged at all levels of the protein structure hierarchy, (see Schlick [34]). A largely manual categorization of proteins by structure is provided by SCOP, (Structural Classification of Proteins). The categories in this scheme are

$$\text{Class} \supset \text{Fold} \supset \text{Superfamily} \supset \text{Family}$$

Protein molecules are arranged within a Family according to the species of organism they come from. The four main classes in SCOP (Structural Classification of Protein) are

- all-α: proteins with only α-helices.
- all-β: proteins with only β-sheets.
- α/β: proteins with both α-helices and mainly parallel β-sheets, (as beta-alpha-beta units).
- α+β: (proteins with both α-helices and mainly antiparallel β-sheets, as separate alpha and beta domains).

7.3 Our Goal

We want to be able to tell whether or not two protein molecules belong to the same superfamily. When comparing molecules in the same superfamily, we are careful to consider molecules from different families, (molecules in the same family are essentially identical, and it is no test of method quality to verify this). Our proof of concept analysis involves the twelve molecules from the all-α class shown in Figure 7.2. Each row in the table corresponds to a different all-α superfamily. The molecules in each row come from different families.

7.4 Recognizing Different Folds

In Figure 7.3, we show that a qualitative difference exists between I-sets for different protein molecules, (in particular for the protein molecules with PDB identifiers 1ENH, 1CTF, and 1ABO). The proteins with PDB entries 1ENH and 1CTF both contain three α-helices, while the protein with PDB entry 1ABO contains none. The similarities and differences in these curve structures are reflected in their corresponding ($d = 25$Å) E-sets; the E-sets for these molecules at other scales also reflect these similarities and differences.

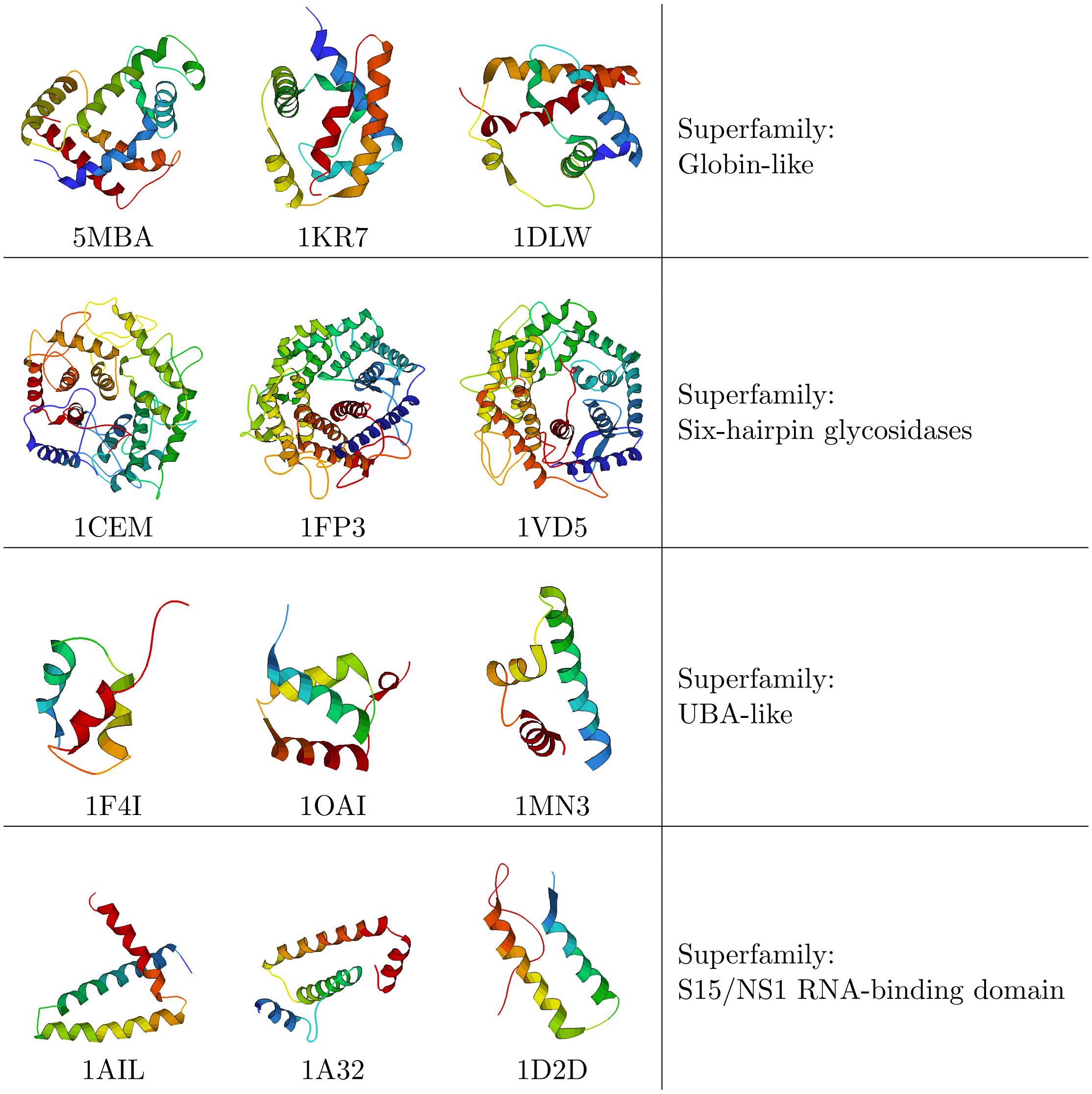


Figure 7.2. Twelve protein molecules from the all-α class. Each row contains proteins from the same superfamily; no two proteins are from the same family.

Because different I-sets correspond to different protein structures, we can use them to navigate the space of protein structures, using the notion of distance developed in chapter 6. The algorithm that we develop is like DALI, but with a similarity score based on comparing intersection subsets rather than distance matrices. The data structure associated with each protein molecule in our scheme is a sparse $n \times n$ logical array, where n is the number of residues in the protein backbone chain. As a sparse array, the actual storage size is far less than n^2. These small easily manipulated data structures compare quite favorably to the

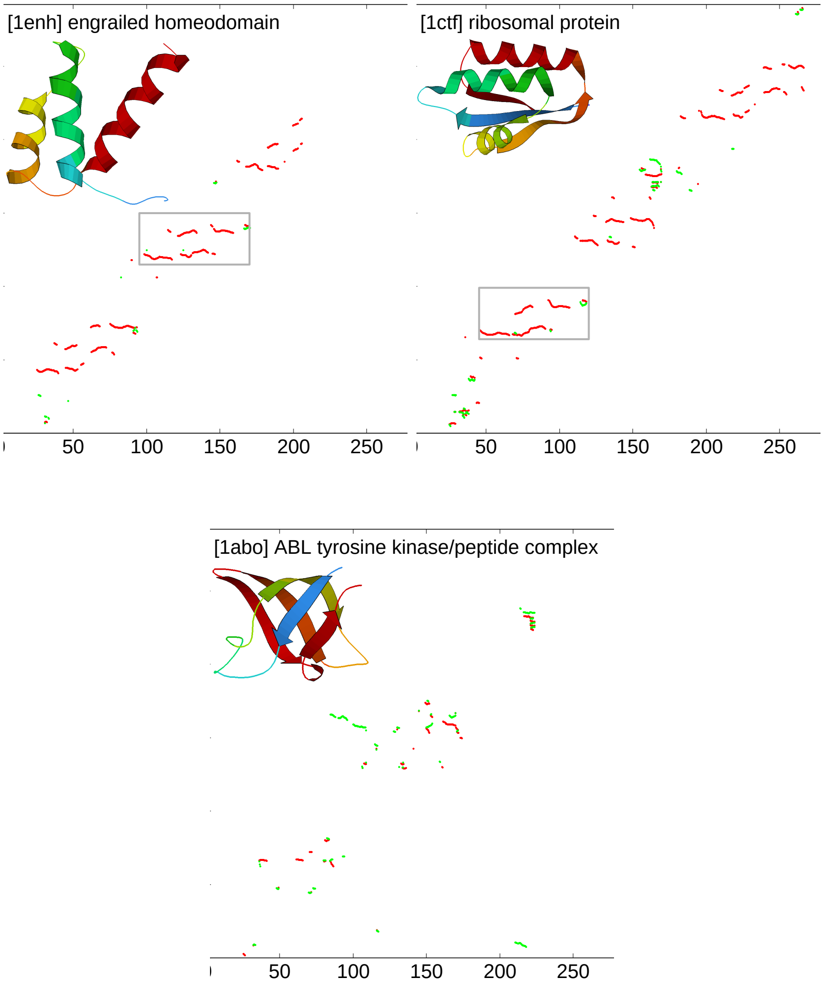


Figure 7.3. Self encirclement sets for three different proteins, with $d = 25\text{Å}$. The axes indicate position (in Angstroms) along the protein backbone. There is an especially good correspondence between certain E-subsets of the proteins with PDB entries 1ENH and 1CTF, such as those boxed in gray. These particular two subsets are compared with greater magnification in Figure 6.3. (Protein images generated with KiNG Version 1.39, see http://kinemage.biochem.duke.edu/)

much larger distance matrices used in DALI. Given a similarity score, it is straightforward to organize protein molecules into trees, clusters, and other structural families [18, 20, 19].

7.5 Protein Alignment

The same range of possibilities exists for comparing protein molecules that exists for comparing space curves in general, (discussed in Section 6.1). Corresponding protein segments can be at different locations along their respective chains, and the order of these segments along their respective chains can differ. Finding which segments of one protein best correspond to which segments of another involves sifting through a huge number of possible pairings. One of the most successful algorithms for doing this is DALI [18], which uses a Monte Carlo approach based on comparing the distances between curve nodes to slowly assemble a collection of pairings for which an overall similarity score is high.

Although an I-set based comparison scheme using the Khun-Munkres algorithm (see 6.7.2) can deal with the difficult case of non-sequential alignment, we explore a cheaper scheme which makes use of the fact that most similar protein molecules have corresponding pieces that are in the same order and direction. This is referred to as *sequential alignment* (see Appendix C).

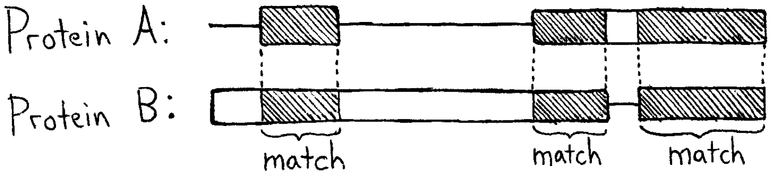


Figure 7.4. Structurally similar proteins usually have corresponding pieces that are in the same order and direction.

7.6 Protein Comparison: Practical Details

We align and compute the distance between protein molecules by following the procedure outlined in Chapter 6. Here we provide detail on the steps involved in this process. We start by downloading the protein PDB files from www.pdb.org. Next, we read atom coordinates into a $3 \times N$ double array in Matlab, (each column of the array gives the position of a backbone chain atom in $\mathbb{E}^3$). An I-set is then computed from this array.

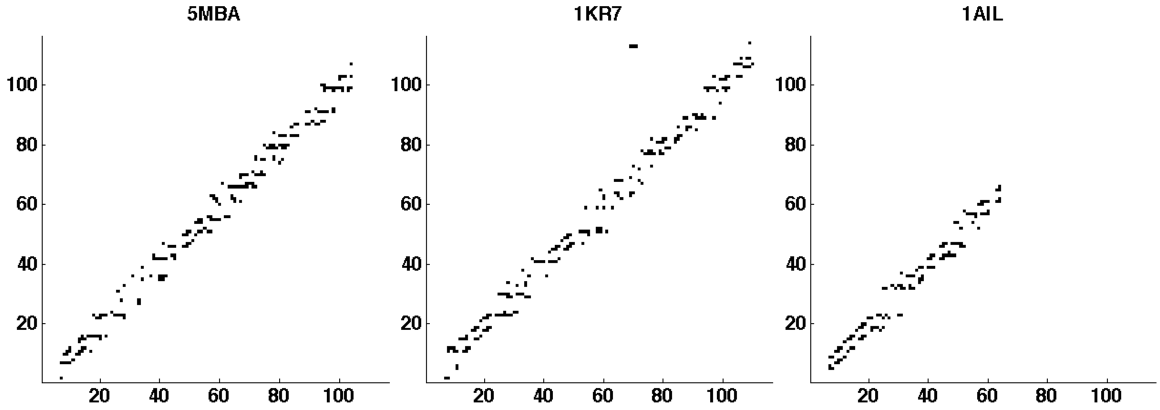


Figure 7.5. E-sets for three of the molecules from Section 7.2, with $d = 25$Å. Axes are in units of residues.

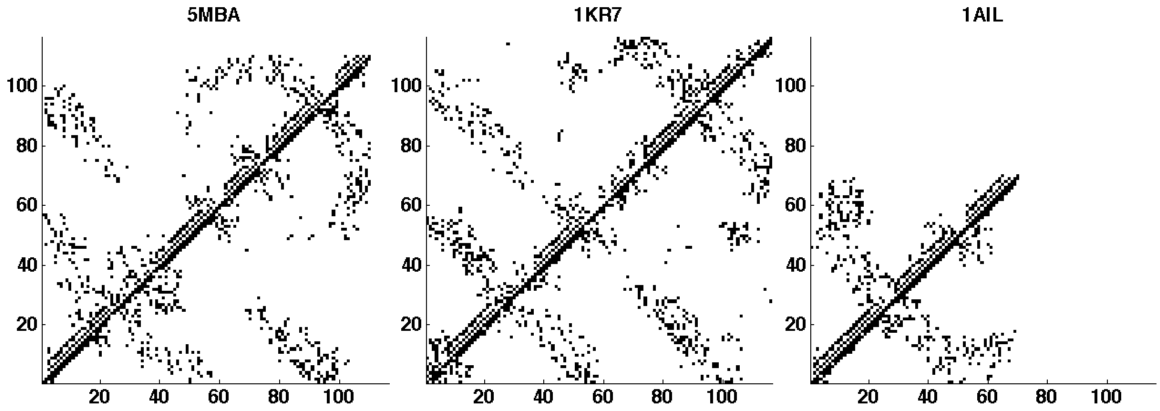


Figure 7.6. D-sets for three of the molecules from Section 7.2, with $r = 15$Å. Axes are in units of residues.

Although an I-set is a collection of curves in the *st*-plane (as in Figure 7.3), we find that protein I-sets consist of so many distinct curves that they appear more as sprinklings of points. On the scale of the whole set, the individual strands are less important than the patterns of these point clusters. When constructing an I-set for a curve, we therefor ignore the fine features of individual strands, and store the I-set as a sparse logical array (as discussed in Section 6.6). Each row and column of the array corresponds to a protein *residue* (i.e., three atoms along the backbone chain). Representative E-sets and D-sets for three protein molecules from Section 7.2 are shown in Figures 7.5 and 7.6 respectively.

In our investigations, I-subsets are 20×20 subarrays along the main diagonal of the I-set

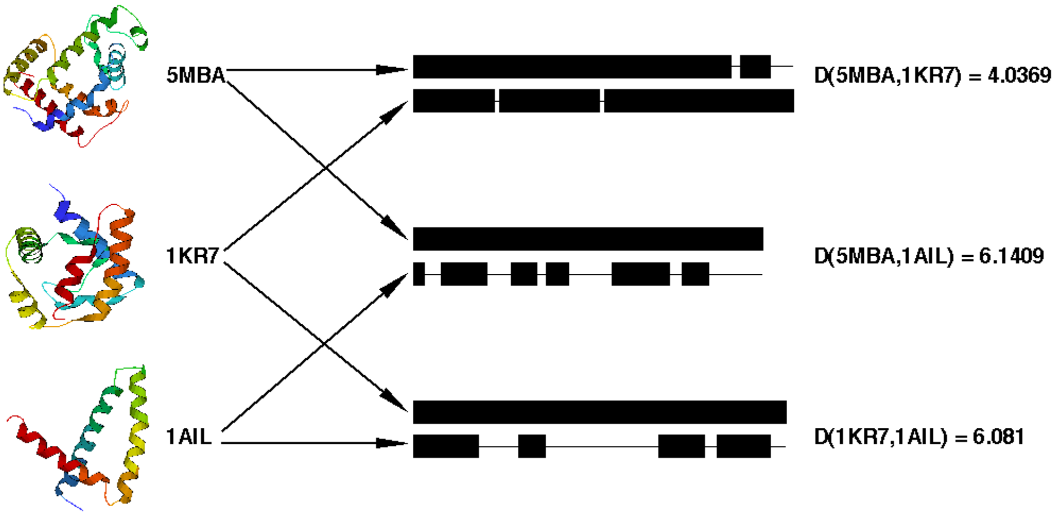


Figure 7.7. Alignments and distances for the molecules from Figures 7.5 and 7.6. (These molecules are from Table 1.

array. Using (6.5), we compute the distances between all possible pairs of I-subsets of the two proteins. Then, dynamic programing with a gap penalty of 10 (see Appendix C) is used to produce an alignment for the two proteins. An overall score is then computed using (6.8a). Representative results are given in Figure 7.7.

7.7 Overall Distribution of Molecules

We now run proof-of-concept computations on the selection of twelve all-α molecules shown in Table 1. An alignment and overall distance is computed for every pair of molecules. The molecules are then visualized using multidimensional scaling [5] as abstract points in the plane (see Appendix A). These abstract points are far apart or close together depending on the I-set based distance between the corresponding molecules.

With both the E-set and D-set data, (Figures 7.8 and 7.9 respectively), we find that molecules in the same SCOP superfamilies are generally clumped together, away from molecules from other superfamilies. Different distance functions (i.e., E-sets versus D-sets) give rise to different clusterings of the protein molecules.

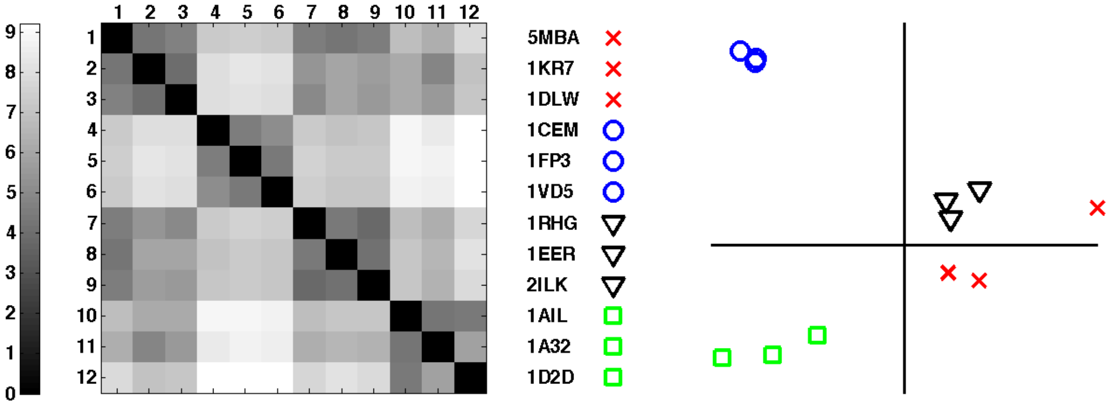


Figure 7.8. E-set comparison. All against all table of E-set based distances for the proteins from Table 1. Protein molecules are listed at center in the order in which they correspond to the rows (and columns) in the table at left. Next to each protein is a glyph; proteins in the same superfamily have the same glyph. An MDS plot for this data is shown on the right.

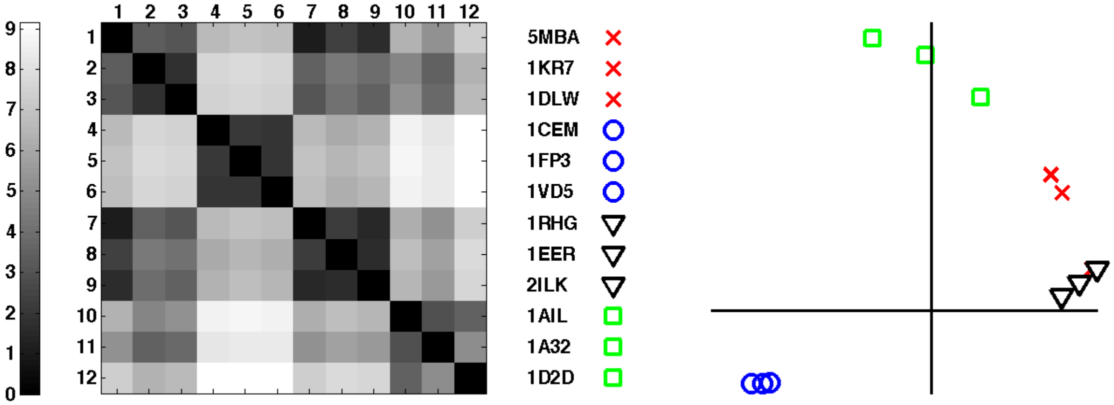


Figure 7.9. D-set comparison. All against all table of D-set based distances for the proteins from Table 1. As in Figure 7.8, protein molecules are listed at center in the order in which they correspond to the rows (and columns) in the table at left. Next to each protein is a glyph; proteins in the same superfamily have the same glyph. An MDS plot for this data is shown on the right.

D-sets are richer with data than E-sets, however this extra information does not seem necessary for the task of distinguishing between different protein molecules.

7.7.1 Shape Distribution Results

Here we compare the collection of molecules in Table 1 using Funckhouser's method of shape distributions. Distributions for three of the proteins from Table 1 are shown in Figure 7.10.

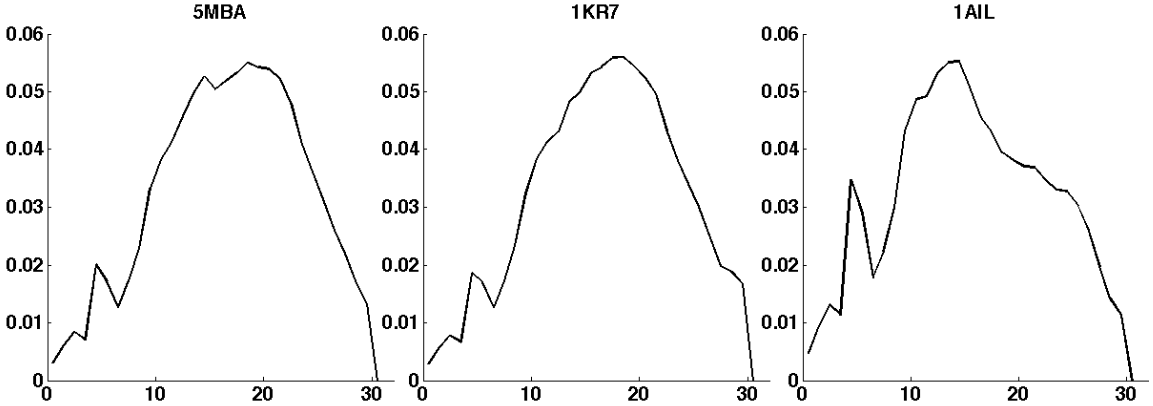


Figure 7.10. Representative probability distributions of pairwise distances (over all pairs of backbone atoms) for the protein molecules with PDB identifiers 5MBA, 1KR7, and 1AIL. The x-axis in each case is in Angstroms.

We use $\langle f, g\rangle = \int f g \; ds$ as an inner product between shape distributions f and g. The corresponding metric on shape distributions is given by $\|f - g\|$ where $\|f - g\|^2 = \langle f - g, f - g\rangle$. In Figure 7.11 we show the collection of pairwise distances between the shape distributions of the 12 molecules in Table 1. As in Figures 7.8 and 7.9, we use MDS to transform these distances into a collection of 12 abstract points in the plane. The grouping of these points is mildly consistent with the corresponding SCOP superfamilies, however the effect is weaker than for the corresponding E-set and D-set analyses shown in Figures 7.8 and 7.9. This is likely due to the fact that E-sets and D-sets contain more structural information than histograms.

7.8 A Faster Implementation

Instead of comparing 2D curve I-sets, it is possible to compare 1D objects constructed from these I-sets. A 1D object associated with the n nodes of a discrete space curve for

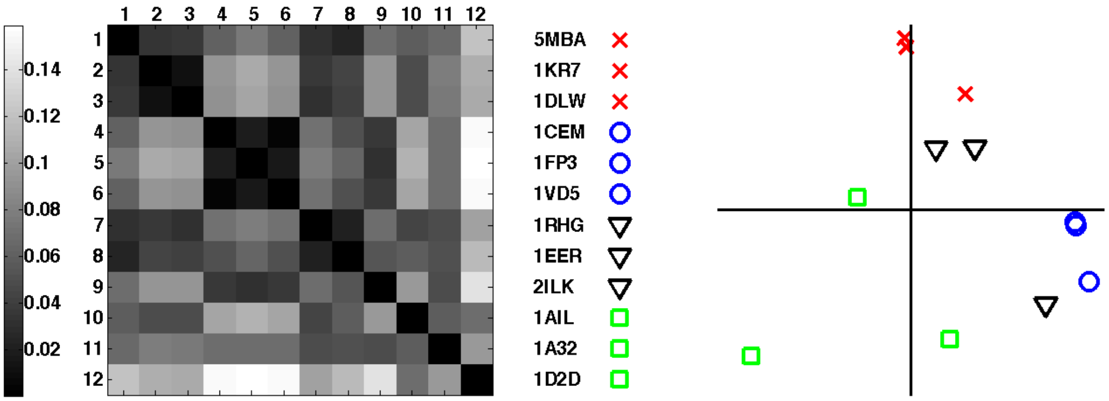


Figure 7.11. All against all table of shape distribution based distances for the proteins from Table 1 together with a corresponding MDS plot. The distances and MDS data are displayed in the same way as in Figures 7.8 and 7.9.

instance is given by a $1 \times n$ array; each node of a space curve can be assigned a number related to the intersections associated with the manifold based at the node. For instance, each node of a space curve can be assigned an integer equal to the number of times the manifold based at this node is intersected.

Comparing 1D vectors is cheaper than comparing 2D I-sets; in an ongoing project with workers in bioinformatics, a scheme based on comparing 1D arrays constructed from I-sets has been generated that performs with a speed and accuracy on the order of the best schemes from that field. In fact, the best scheme is one based on Voroni cells, that we describe in Section 6.2, and that is remarkably similar to our scheme.

7.9 Lack of Encirclement

Protein E-sets show distinctive patterns for different protein structures, however protein chains generally do not experience the kind of encirclement that first motivated our construction of E-sets, in which one curve wraps many times around another. This kind of encirclement does occur between a protein chain and a smoother version of the chain, as we show in Figure 7.12. Let $\mathbf{X}$ be a $3 \times N$ array with columns containing the xyz coordinates of the atoms comprising a protein's backbone chain, and let $A_k(\mathbf{X})$ denote the k^{th} spatial average of $\mathbf{X}$, where a sequence of spatial averages is defined by

$$\begin{aligned} &A_0(\mathbf{X}) = \mathbf{X}, \\ &[A_{k+1}(\mathbf{X})]_1 = [\mathbf{X}]_1, \;\; [A_{k+1}(\mathbf{X})]_N = [\mathbf{X}]_N, \\ &[A_{k+1}(\mathbf{X})]_i = \frac{[A_k(\mathbf{X})]_{i-1} + [A_k(\mathbf{X})]_{i+1}}{2}, \end{aligned}$$

for $i = 2, \ldots, N-1$. As k grows, $A_k(\mathbf{X})$ approaches the straight line connecting the endpoints of $\mathbf{X}$.

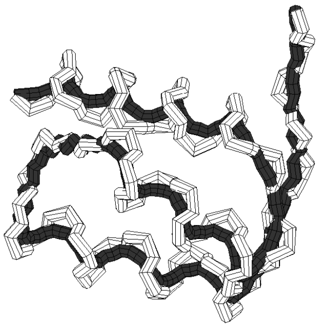


Figure 7.12. The backbone $\mathbf{X}$ of the protein with PDB identifier 1ENH is shown in white, and the smoother spatial average $A_5(\mathbf{X})$ of this curve is shown in black. The encirclement of $A_5(\mathbf{X})$ by α-helices in $\mathbf{X}$ shows up clearly in the $d = 4\text{Å}$ E-set for these curves.

For small values of k, $A_k(\mathbf{X})$ is encircled by the α-helices in $\mathbf{X}$. For larger values of k, $A_k(\mathbf{X})$ engages with the higher order coiling structures in $\mathbf{X}$ (e.g., barrels), and the corresponding E-sets reflect this with signature markings. It is interesting to consider *coiling* (the adding of coils to a given curve) as an inverse operation to averaging. Just as a smooth version of $\mathbf{X}$ is encircled by the α-helices in the protein, the β-sheet strands in a protein are encircled by a coiled version of $\mathbf{X}$. Thus inverse operations reveal the two secondary protein structures!

Chapter 8

Closing Comments

There are many additional families of space curves that can be investigated using I-sets. These include biological fibers such as root tendrils and arteries, as well as solutions of the dynamical system $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$. An example of such a system is given by Lorenz's equations [26], which give rise to a strange attractor and chaos. In the context of a dynamical system we could associate to every point $\mathbf{x}$ in the domain of the field $\mathbf{f}$ a manifold M. We could consider intersections between M and the forward (and backward) images of $\mathbf{x}$ under the flow induced by $\mathbf{f}$. This extension is similar of the extension of Gauss's formula to vector fields, (introduced by Lodewijk Woltjer in 1958, and called *helicity* [6]).

The method of I-sets extends to surfaces and higher dimensional manifolds. In general, let M be an m-dimensional manifold in some larger space, and associate to each point of M an n-dimensional manifold N. For instance, let M be a 2-dimensional manifold in $\mathbb{R}^3$, as shown in Figure 8.1, and let the manifold N associated with point $p \in M$ be given by the tangent space T_pM. The line through p normal to the tangent space and spheres centered at p are additional possibilities for N. The manifold M can be characterized by the way in which it intersects N. For instance if N is the line through p perpendicular to T_pM, then the length along N from p to the nearest point of intersection gives an idea of how close M at p is to other pieces of M.

Recently, we discovered that this approach to characterizing surfaces has been investi-

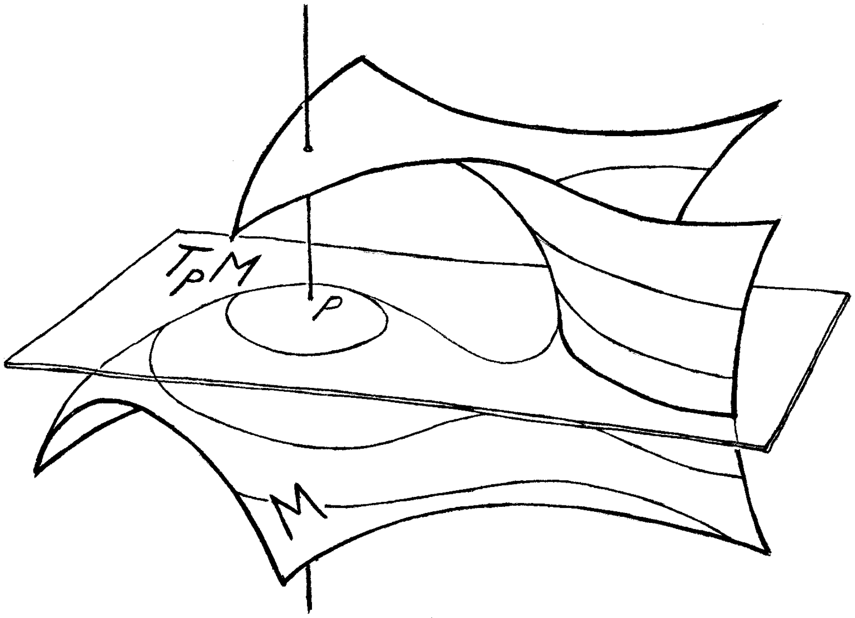


Figure 8.1. Extension of the I-set construction to a 2D surface M. Manifolds associated with the point $p \in M$ include the tangent space T_pM, and the line through p perpendicular to T_pM. The intersection of these manifolds with M give information about the shape of M.

gated for the case where the intersecting manifold N is a sphere; Mortara and co-workers [27] have studied how intersections made with a family of different sized spheres centered at a point convey information about the shape of a surface near that point.

Our work is based on the idea of exploring the shape of a manifold in a scale d region surrounding a given point on the manifold. This general idea can be implemented in a variety of ways. For instance, in the case of a space curve $\mathbf{x}$, we can consider the distance from $\mathbf{x}(s)$ to the center of mass of the strand segment from $s - d$ to $s + d$, or we could consider the radius of the smallest ball containing the strand segment from $s - d$ to $s + d$. Another construction is a d-scale tangent indicatrix, (i.e., the set of unit vectors pointing from $\mathbf{x}(s)$ to $\mathbf{x}(s + d)$). These are semi-global measures; they depend on more than what goes on in a (vanishingly small) region around a point $\mathbf{x}(s)$, however they do not depend on what happens very far away from $\mathbf{x}(s)$. There are many exciting ways to present this data graphically, for instance see [37].

Bibliography

[1] W. Albrecht, W. Kittelmann, and H. Fuchs, editors. *Nonwoven Fabrics.* Wiley-VCH, 2003.

[2] S. Axler. *Linear Algebra Done Right.* Springer, 2004.

[3] Jean-Baptiste Biot and Felix Savart. Note sur le magnetisme de la pile de Volta. *Annales de chimie et de physique, 2nd ser.*, 15:222–223, 1820.

[4] F. Birzele, J.E. Gewehr, G. Csaba, and R. Zimmer. Vorolign- fast structural alignment using Voronoi contacts. *Structural Bioinformatics*, 23(2):e205–e211, 2007.

[5] I. Borg and P.J.F. Groenan. *Modern Multidimensional Scaling, Theory and Applications.* Springer Series on Statistics. Springer, 2nd edition, 2005.

[6] Jason Canarella, Dennis Deturck, and Herman Gluck. The writhing number of a space curve. *International Press, AMS/IP Series on Advanced Mathematics*, 2000. Proc. of Conerence in Honor of the 70th Birthday of Joan Birman, ed. by J. Gilman, X.-S. Lin and W. Menasco.

[7] James Casey. *Exploring Curvature.* Friedrich Vieweg & Sohn, 1996.

[8] G. Călugăreanu. L'integral de Gauss et l'analyse des noeuds tridimensionnels. *Rev. Math. Pures Appl.*, 4:5–20, 1959.

[9] G. Călugăreanu. Sur les classes d'isotopie des noeuds tridimensionnels et leurs invariants. *Czechoslovak Math. J.*, 11(86):588–625, 1961.

[10] G. Călugăreanu. Sur les enlacements tridimensionnels des courbes fermees. *Comm. Acad. R. P. Romine*, 11:829–832, 1961.

[11] P. Daras, D. Zarpalas, A. Axenopoulos, D. Tzovaras, and M. G. Strintzis. Three-dimensional shape-structure comparison method for protein classification. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 3(3):193–207, July-September 2006.

[12] James W. Demmel. *Applied Numerical Linear Algebra.* SIAM, 1997.

[13] Theodore Frankel. *The Geometry of Physics.* Cambridge University Press, 1997.

[14] F. Brock Fuller. The writhing number of a space curve. *Proc. Nat. Acad. Sci. USA*, 68(4):815–819, April 1971.

[15] Oscar Gonzalez. Website: http://www.ma.utexas.edu/ og/curvature.html, 2006.

[16] Oscar Gonzalez and J.H. Maddocks. Global curvature, thickness and the ideal shapes of knots. *Proc. Nat. Acad. Sci. USA*, 96:4769–4773, 1999. See also Gonzalez's website: http://www.ma.utexas.edu/ og/curvature.html.

[17] T.D. Gottschalk. Concurrent implementation of Munkres algorithm. *IEEE, 0-8186-2113-3*, 1990.

[18] L. Holm and C. Sander. Protein structure comparison by alignment of distance matrices. *J. Mol. Biol*, 233:123–138, 1993.

[19] J. Hou, S.R. Jun, C. Zhang, and S.H. Kim. Global mapping of the protein structure space and application in structure-based inference of protein function. *Proc. Nat. Acad. Sci. USA*, 102:3651–3656, March 2005.

[20] J. Hou, G.E. Sims, C. Zhang, and S.H. Kim. A global representation of the protein fold space. *Proc. Nat. Acad. Sci. USA*, 100:2386–2390, March 2003.

[21] P. Kessler and O. M. O'Reilly. Curve encirclement and protein structure. *PRL*, January 2007. To appear in PRL.

[22] E. Kreyszig. *Differential Geometry.* Dover, 1991. Originally published by the University of Toronto Press, Toronto, 1959.

[23] H.W. Kuhn. The Hungarian method of the assignment problem. *Naval Research Logistics Quarterly*, 2-83, 1955.

[24] A. M. Lesk. *Introduction to Protein Architecture.* Oxford University Press, 2001.

[25] S. Loncaric. A survey of shape analysis techniques. *Pattern Recognition*, 31(8):983–1001, 1998.

[26] E.N. Lorenz. Deterministic nonperiodic flow. *J. Atmos. Sci.*, 20:130–141, 1963.

[27] M. Mortara, G. Patané, M. Spagnuolo, B. Falcidieno, and J. Rossignac. Blowing bubbles for the multi-scale analysis and decomposition of triangle-meshes. *Algorithmica*, 38(1):227–248, 2003. Available online at http://www.gvu.gatech.edu/ jarek/papers/-Taylor.pdf.

[28] J.R. Munkres. *Topology.* Prentice Hall, Second edition, 2000.

[29] T. Needham. *Visual Complex Analysis.* Cambridge University Press, 1997.

[30] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Shape distributions. *ACM Transactions on Graphics*, 21(4):807–832, October 2002.

[31] E.J. Rawdon and J. Simon. Moebius energy of thick knots. *arXiv*, August 2001.

[32] P. Rogen and B. Fain. Automatic classification of protein structure by using Gauss integrals. *Proc. Nat. Acad. Sci. USA*, 100:119–124, January 2007.

[33] R. Sarma. *Ramachandran: A Biography of Gopalasamudram Narayana Ramachandran, the Famous Indian Biophysicist.* Adenine Press, Schenectady, NY, 1998.

[34] T. Schlick. *Molecular Modeling and Simulation: An Interdisciplinary Guide.* Springer Verlag, 2002.

[35] P. H. Schönemann. A generalized solution of the orthogonal Procrustes problem. *Psychometrika*, 31(1):1–10, 1966.

[36] J.W.H. Tangelder and R.C. Veltkamp. A survey of content based 3D shape retrieval methods. *Proceedings of the Shape Modeling International*, August 2004. IEEE 0-7695-2075-8/04.

[37] L. Wilkinson. *The Grammar of Graphics.* Springer Verlag, 2000.

[38] D. Zhi, S. S. Krishna, H. Cao, P. Pevzner, and A. Godzik. Representing and comparing protein structures as paths in three-dimensional space. *BMC Bioinformatics*, 7:460, October 2006. article is available from http://www.biomedcentral.com/1471-2105/7/460.

Appendix A

Multi-Dimensional Scaling

Multi-Dimensional Scaling (MDS) is a technique for positioning points in an abstract space so that the pairwise distances between them have prescribed values. It is easier to digest a planar (or 3D) plot of n abstract points than it is to digest a table of $\frac{(n+2)(n+1)}{2}$ distance values, and so MDS is useful as a communication tool- it increases the information content of a graph. MDS is a well established technique, and for a review of its history as well as a detailed mathematical analysis of its various incarnations we recommend [5].

The name "Multidimensional Scaling" reveals little about the associated mathematical technique; we suspect that this name continues to be used mainly because of cultural inertia, and because "MDS" is a pleasant TLA (Three Letter Acronym). As far as we can tell, the "multidimensional" in MDS refers to the multidimensionality of the abstract space, while the "scale" refers to our ability to adjust the scale of the data in any of these dimensions, (something not discussed below).

A.1 Uses

We use MDS to visualize the clustering of protein molecules that corresponds to I-set based distances between them. MDS is used in this way in [19, 20] to understand the DALI based distances between molecules. Different metrics give rise to different clusterings of molecules.

In addition to visualizing the clustering of molecules, MDS allows for the reconstruction of a curve given only the collection of pairwise distances between curve points, (this problem was considered in Chapter 5).

A.2 Getting Positions from Distances

The goal in MDS is to construct n position vectors $\mathbf{x}_k$ $(k = 1, \ldots, n)$ from a given set of $\frac{(n+2)(n+1)}{2}$ scalar distances $d_{ij} = d_{ji}$ $(i, j = 0, 1, \ldots, n)$, so that

$$\|\mathbf{x}_i - \mathbf{x}_j\| = d_{ij} \qquad \text{and} \qquad \|\mathbf{x}_i\| = d_{0i}. \tag{A.1}$$

The MDS problem is rich and interesting, especially when the equalities in (A.1) only need to be satisfied approximately, (such as when the vector space of positions is so small that no exact MDS solution is possible). We avoid this complexity in the following derivation. For concreteness, we discuss the MDS problem in the context of the vector space $\mathbb{R}^L$ (the set of L-tuples of real numbers), endowed with the standard inner product $\langle\cdot,\cdot\rangle$ and corresponding norm $\|\cdot\|$. We use a set of n vectors $\mathbf{y}_k \in \mathbb{R}^L$ to establish a consistent set of initial distances d_{ij}, and from this set of distances, we construct a set of vectors $\mathbf{x}_k \in \mathbb{R}^L$ satisfying (A.1).

Let $\mathbf{y}_1, \ldots, \mathbf{y}_n$ be n vectors in $\mathbb{R}^L$. Think of the k^{th} vector $\mathbf{y}_k$ as an $L \times 1$ array, and define $\mathbf{Y} = [\mathbf{y}_1 \cdots \mathbf{y}_n]$ to be the $L \times n$ array which contains the $\mathbf{y}_k$'s as columns. The $\mathbf{y}_k$'s give rise to a set of distances $d_{ij} = \|\mathbf{y}_i - \mathbf{y}_j\|$, and $d_{i0} = \|\mathbf{y}_i\|$, which we use as starting data for constructing a set of vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$, that satisfy (A.1).

Our first step is to use the d_{ij}'s to construct the $n \times n$ array $\mathbf{P}$ of inner products $\langle \mathbf{x}_i, \mathbf{x}_j\rangle$. We find the $\langle \mathbf{x}_i, \mathbf{x}_j\rangle$'s in terms of the d_{ij}'s *before* finding the $\mathbf{x}_k$'s. This can be done using $\langle \mathbf{x}_i, \mathbf{x}_i\rangle = d_{i0}^2$, and $2\langle \mathbf{x}_i, \mathbf{x}_j\rangle = d_{i0}^2 + d_{j0}^2 - d_{ij}^2$. (Conversely, these relations can be used to obtain the d_{ij}'s from the $\langle \mathbf{x}_i, \mathbf{x}_j\rangle$'s.) Once we have $\mathbf{P}$, our job is to compute a collection of $\mathbf{x}_k$'s that satisfies (A.1), that is, to find an $L \times n$ array $\mathbf{X} = [\mathbf{x}_1 \cdots \mathbf{x}_n]$ which satisfies $\mathbf{X}^T\mathbf{X} = \mathbf{P}$.

In this problem, $\mathbf{P} = \mathbf{Y}^T\mathbf{Y}$ is symmetric, and so from the spectral theorem [2], it follows that an orthonormal basis $(\mathbf{e}_i)$ for $\mathbb{R}^n$ exists with respect to which $\mathbf{P}$ is diagonal. In particular, with $\mathbf{E} = [\mathbf{e}_1 \cdots \mathbf{e}_n]$, we can write

$$\mathbf{P} = \mathbf{E}\mathbf{D}\mathbf{E}^T, \tag{A.2}$$

where $\mathbf{D}$ is diagonal. Because $\mathbf{v}^T\mathbf{P}\mathbf{v} \geq 0$ for all vectors $\mathbf{v}$ (i.e., because $\mathbf{P}$ is positive semidefinite), the $\mathbf{e}_i$'s can be ordered in $\mathbf{E}$ so that $\mathbf{D} = \mathrm{diag}(d_1, d_2, \ldots, d_n)$, with $d_1 \geq d_2 \geq \cdots \geq d_n \geq 0$. If only the first k diagonal terms of $\mathbf{D}$ are nonzero, then (A.2) can be written as

$$\mathbf{P} = \tilde{\mathbf{E}}\tilde{\mathbf{D}}\tilde{\mathbf{E}}^T, \tag{A.3}$$

where $\tilde{\mathbf{E}} = \mathbf{E}(:, 1\!:\!k)$, and where $\tilde{\mathbf{D}} = \mathrm{diag}(d_1, \ldots, d_k)$. With $\tilde{\mathbf{D}}^{\frac{1}{2}} = \mathrm{diag}(\sqrt{d_1}, \ldots, \sqrt{d_k})$, a $k \times n$ array $\mathbf{X}$ satisfying $\mathbf{X}^T\mathbf{X} = \mathbf{P}$ is given by

$$\mathbf{X} = \tilde{\mathbf{D}}^{\frac{1}{2}}\tilde{\mathbf{E}}^T. \tag{A.4}$$

When $n \geq L$, $\dim(\mathrm{null}(\mathbf{P} = \mathbf{Y}^T\mathbf{Y})) \geq n - L$, and so $k \leq L$. The columns of $\mathbf{X}$ from (A.4) therefore reside in $\mathbb{R}^k \subset \mathbb{R}^L$ as desired. When $k < L$, the columns of $\mathbf{X}$ occupy a k-dimensional hyperplane in $\mathbb{R}^L$. For real data this almost never happens, (i.e., usually $k = L$). When $n < L$, we immediately have $k \leq n$, and so the columns of $\mathbf{X}$ occupy at most an n-dimensional hyperplane in $\mathbb{R}^L$.

A.2.1 Approximate Solutions

Because the $\mathbf{e}_i$'s are orthogonal, $\mathbf{P}$ can be written as the following sum of outer products

$$\mathbf{P} = d_1\mathbf{e}_1\mathbf{e}_1^T + d_2\mathbf{e}_2\mathbf{e}_2^T + \cdots + d_n\mathbf{e}_n\mathbf{e}_n^T, \tag{A.5}$$

where $(\mathbf{a}\mathbf{b}^T)\mathbf{c} = \gamma\mathbf{a}$, with $\gamma = \langle \mathbf{b}, \mathbf{c}\rangle$. The tailing terms can be dropped without seriously affecting the sum. In particular, let $\mathbf{P}_k$ denote the sum of the first k terms in the sum. The relative error ϵ associated with approximating $\mathbf{P}$ by $\mathbf{P}_k$ is given by

$$\epsilon = \frac{\|\mathbf{P} - \mathbf{P}_k\|}{\|\mathbf{P}\|} = \frac{d_{k+1}}{d_1} \tag{A.6}$$

where $\|\mathbf{P}\|$ is the induced 2-norm on matrices. In the case of (A.3), $d_{k+1} = 0$, and so $\epsilon = 0$. Generally, $\mathbf{P}_k$ is the best rank k approximation to $\mathbf{P}$, in the sense that it gives the lowest value of the relative error ϵ in (A.6). As with the construction of $\mathbf{X}$ in (A.4), a rank k approximation to the $n \times n$ array $\mathbf{P}$ gives rise to a set of n vectors in $\mathbb{R}^k$ with relative inner products that are a good approximation to those in $\mathbf{P}$; as k increases, so does the quality of the approximation.

A.2.2 Orientation of the Data

Consider the quantity $X(\mathbf{u}) = \sum_{j=1}^{n} \langle \mathbf{u}, \mathbf{x}_j \rangle^2$ where $\mathbf{u}$ is a unit vector in $\mathbb{R}^k$. Note that

$$X(\mathbf{u}) = \mathbf{u}^T \mathbf{X} \mathbf{X}^T \mathbf{u} = \mathbf{u}^T \tilde{\mathbf{D}} \mathbf{u} = u_1^2 d_1 + u_2^2 d_2 + \cdots + u_k^2 d_k, \tag{A.7}$$

and so from the conditions $\sum u_i^2 = 1$ and $d_1 \geq d_2 \geq \cdots \geq d_k \geq 0$, it follows that the maximum value of $X(\mathbf{u})$ is d_1. This value is attained when $u_1 = 1$ and when $u_i = 0$ for $i \neq 1$; we denote the associated vector by $\mathbf{u}_1$. The components of the $\mathbf{x}_i$'s (i.e., the columns of $\mathbf{X}$) in this direction are furthest from the origin in the sense that the sum of the squares of their values is greatest.

Let $\mathbf{u}_i$ denote the unit vector in $\mathbb{R}^k$ with a 1 in the i^{th} slot, and with 0's everywhere else, (note that $\mathbf{u}_i = \tilde{\mathbf{E}}^T \mathbf{e}_i$, and that the $\mathbf{u}_i$'s comprise an orthonormal basis for $\mathbb{R}^k$). Generally, the maximum value of $X(\mathbf{u})$ over all unit vectors in $\mathbf{u} \in \text{span}(\mathbf{u}_i, \mathbf{u}_{i+1}, \ldots, \mathbf{u}_k)$ is d_i, and $X(\mathbf{u}_i) = d_i$. Thus the data is preferentially aligned with the natural basis $(\mathbf{u}_i)$ for $\mathbb{R}^k$.

A.3 A Matlab Example

The eigenvalues in $\mathbf{D}$ and the eigenvectors in $\mathbf{E}$ can be computed using the `eig` command in Matlab (which uses QR factorization as described in [12]). The following code demonstrates MDS by reconstructing a set of vectors (in $\mathbb{R}^2$) given only the inner products between them. In Figure A.1, we show original and reconstructed vectors as points in the plane; the two only differ by an isometry and so the reconstruction is a success.

```
Y=rand(2,15)-0.5;   %Establish 15 vectors in R^2, (i.e., the columns of Y).
P=Y'*Y;             %Find the pairwise inner products between the vectors.
[E,D]=eig(P);       %Given P, compute eigenvectors E and eigenvalues D.

%Arrange E and D so that the values in D are in increasing order:
[D,per]=sort(diag(D),1,'descend');  E=E(:,per);

%Create X
X=diag(sqrt(D(1:2)))*E(:,1:2)';
```

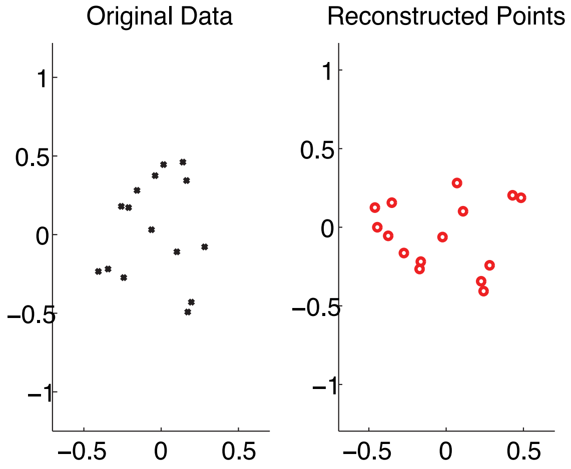


Figure A.1. Original and reconstructed vectors from the above Matlab code.

Appendix B

The Assignment Problem

Given $A \in \mathbb{R}^{n \times n}$, we wish to find a permutation $P = (p_1, p_2, \ldots, p_n)$ of $(1, 2, \ldots, n)$ for which the sum $\sum_{k=1}^{n} A(p_k, k)$ is a minimum.

There are $n!$ possible permutations P, and so directly checking the associated sums is out of the question. We present our own version of the Munkres-Kuhn algorithm [17, 23], which determines a minimizing P in only $O(n^3)$ operations. The algorithm extends to the case in which A is rectangular, (simply append rows or columns to A to make it square), and also to the case in which the sum must be maximized (replace A with $-A$).

The main observation behind the algorithm is that P is an optimal permutation for A if and only if P is an optimal permutation for B, where B is obtained by adding a constant to all the elements of a row or column of A. Our strategy is to repeatedly add and subtract from the rows and columns of A until finding P is trivial.

B.1 Preprocessing

Start by subtracting the minimum value in each column from that column. Next, subtract the minimum value in each row from that row. These operations cause $A_{ij} \geq 0$, and cause every row and column to contain at least one 0. We add to rows and subtract from columns so as to increase the size of a special selection of the resulting zeros in A; these special zeros are referred to as $\bigstar$'s. The core routine in our algorithm inevitably increases the number of $\bigstar$'s in A. This routine is repeated until the number of $\bigstar$'s equals n. At this point, the $\bigstar$'s correspond to an optimizing permutation and we are done. Each cycle of the core routine involves a possible change in the location of the existing $\bigstar$'s; we keep track of things by *blocking* and *unblocking* the rows and columns of A.

B.2 Preliminary Selection

Cycle through the 0's of A. If no $\bigstar$'s exist in the row or column of a 0, convert the 0 to a $\bigstar$.

B.3 Core Routine

Only enter into the following routine if there are fewer than n ★'s.

Remove all blocks (rows and columns) from A, and then establish a new block on every column with a ★. There are fewer than n ★'s (this is the condition for calling the routine), and every column has at least one 0. It follows that there is at least one unblocked 0. Note that the only reason that this 0 is not a ★ is that some other ★ exists in its row. Think of each such unblocked 0 as the root of a plant-like tendril that will wander over the matrix in a rectilinear path, in search of an unblocked 0 with no ★ in its row. The tendril starts at an unblocked 0, moves horizontally to a ★, moves vertically to a 0, horizontally to another ★, and so on. After the tendril moves horizontally in a row, the row is blocked, and before the tendril moves vertically in a column, the column is unblocked. Thus, as a tendril snakes its way through a matrix, rows become blocked and columns become unblocked. Ideally, a tendril terminates at a 0 with no ★ in its row, as in Figure B.1.

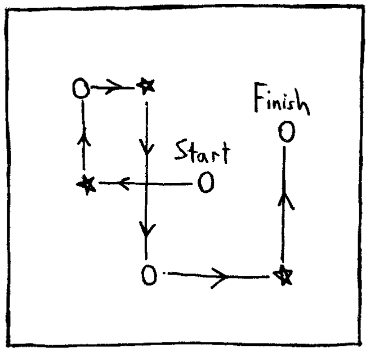


Figure B.1. Tendril terminating at a 0 with no ★ in its row.

When this happens, the ★'s along the tendril are changed to 0's, and the 0's are changed to ★'s, giving one more ★ than when the core routine started, as in Figure B.2. This ends the current cycle of the routine. If there are now n ★'s we are done; if there are still fewer than n ★'s, we return to the begining of this section for another cycle of the core routine.

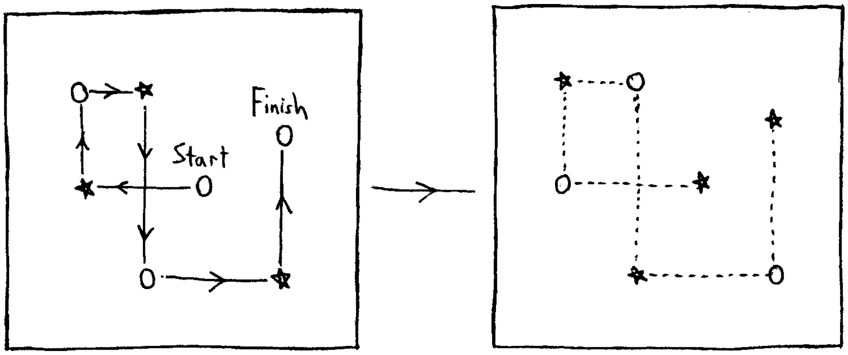


Figure B.2. We end with one more ★ than when the core routine started.

Generally, a single tendril won't be successful in finding a terminal 0. Instead, it will terminate with a ★ that has no unblocked 0's in its column. When this happens, this particular tendril is abandoned, and another is started from any of the unblocked 0's in the matrix (some of which may have become unblocked by another tendril). Thus the general picture is of a dense system of intertwining tendrils, each constructed one after the other.

Frequently, there will be no unblocked 0's at all, and so having multiple tendrils is not enough. If $\alpha > 0$ is the minimum unblocked value in A, we create a new unblocked 0 by subtracting α from the unblocked columns of A and adding α to the blocked rows of A. There are four categories of element to consider:

- An element is completely unblocked.
  The elements in this category decrease in value, and one or more of them becomes equal to 0. (These are the new 0's that we wanted to create.)
- An element is in a blocked row and a blocked column.
  The elements in this category increase in value, and so it could be that some 0's disappear. This is fine though because no elements in this category are marked as ★'s. (When a row is blocked, the single ★ that it contains has its column unblocked.)
- An element is in a blocked row but not a blocked column.
  Elements in this category are unchanged in value.
- An element is in a blocked column but not a blocked row.
  Elements in this category are unchanged in value.

Eventually, an unblocked 0 appears in A with no ★ in its row, (if this didn't happen, it would be possible for A to be completely blocked, but with fewer than n ★'s). When this 0 appears, return to the beginning of Section B.3 for another cycle of the core routine.

B.4 Matlab Implementation

```
function P=munkres(A)
%munkres.m solves the assignment problem given a square matrix A with non INF values

n=size(A,1);
A=A−ones(n,1)*min(A,[],1);
A=A−min(A,[],2)*ones(1,n);

P=zeros(1,n);  %P is a 1xn array, with each entry a nonnegative integer.
                %P(i)>0 indicates a * in row P(i) of column i.
                %P(i)=0 indicates that no * has been assigned for column i.
BR=zeros(n,1);  %BR (for Block Rows) is a 1xn array of 1's and 0's.
                   %A 1 indicates that the corresponding row is blocked.

%Preliminary Star Selection:
for k=1:n
    star=(A(:,k)==0 & ¬BR);
    if any(star)
        index=find(star);  index=index(1);
        P(k)=index;  BR(index)=1;
    end
end

%Main Algorithm:
while ¬all(P>0)  %perform a cycle if not all columns have stars.
    %reset all blocks: BC (Block Columns) is like BR but for columns.
    BR=0*BR;  BC=P>0;
    ZP=false(n);
    %send out tendrils in search of terminal 0's.
    %unblocked is an nxn logical array with 1's indicating unblocked elements
    unblocked=(¬logical(ones(n,1)*BC) & ¬logical(BR*ones(1,n)));
    Z=(abs(A)<1e-10 & unblocked);  %unblocked Z's
    rt=0;  %terminal 0 row index.
    %perform the following loop until a terminal 0 is found.
    while rt==0
        if all(all(¬Z)) %there are no unblocked Zs; so create at least one
            h=min(min(A(unblocked)));
            A=A-h*(ones(n,1)*(¬BC)-BR*ones(1,n));
        Z=(abs(A)<1e-10 & unblocked);
        end
        %now grow tendrils from the elements of Z
        while any(any(Z)) && rt==0
            [r,c]=find(Z); r=r(1); c=c(1); %get indecies for the beginning of a tendril
            flag=1;
            while flag %propagate a tendril
                if all(P≠r)
                    rt=r;  ct=c; %row and column indecies of terminal 0
                    break
                end
                ZP(r,c)=1;  c=find(P==r);
                BR(r)=1;  BC(c)=0;
                %if this column contains no unblocked 0s, the tendril ends.
                if ¬any(Z(:,c)&¬BR)
                    Ir=find(¬BR);  sr=size(Ir,1);
                    Ic=find(¬BC);  sc=size(Ic,2);
                    unblocked=false(n);
                    unblocked(Ir*ones(1,sc)+ones(sr,1)*(n*(Ic-1)))=true;
                    Z=(abs(A)<1e-10 & unblocked);  %unblocked Z's
                    flag=0;
                    continue
                end
                r=find(Z(:,c)&¬BR);  r=r(1);
            end
        end
    end
    %Having found a terminal 0, swap 0's and *'s
    while P(ct)≠0
        rtold=rt;
        rt=P(ct);  P(ct)=rtold;
        ct=find(ZP(rt,:));
    end
    P(ct)=rt;
end
```

Appendix C

Sequential List Alignment

Definitions:

- A *list* is a finite ordered collection of objects, such as $A = (A_1, A_2, \ldots, A_m)$.
- A *sublist* is a subset of a list, with order inherited from the list; if β comes after α in a sublist of A, then β comes after α in A.
- An *inflation* of $(1, 2, \ldots, m)$ is a list of non-negative integers for which $(1, 2, \ldots, m)$ is the sublist obtained by selecting all positive entries; $(1, 0, 2, 0, 0, 3, 4, 0)$ is an inflation of $(1, 2, 3, 4)$.
- A *sequential alignment* of lists $A = (A_1, A_2, \ldots, A_m)$ and $B = (B_1, B_2, \ldots, B_n)$ is a length $L \geq \max\{m, n\}$ inflation α of $(1, 2, \ldots, m)$ together with a length L inflation β of $(1, 2, \ldots, n)$, such that α_k and β_k are never both equal to zero. (So in fact, $L \leq m + n$.)

A sequential alignment pairs *some* of the elements of A and B. If α_k and β_k are non-zero, then A_{α_k} is paired with B_{β_k}. If $\alpha_k \neq 0$ and $\beta_k = 0$, then A_{α_k} isn't paired to any element in B; likewise if $\alpha_k = 0$ and $\beta_k \neq 0$, then B_{β_k} isn't paired to any element in A. For instance if $A = (A_1, A_2, A_3, A_4)$ and $B = (B_1, B_2, B_3, B_4, B_5, B_6)$, and if $\alpha = (1, 2, 3, 0, 0, 4, 0)$ and $\beta = (0, 1, 2, 3, 4, 5, 6)$, then A and B are aligned as follows

$$\begin{array}{ccccccc} A_1 & A_2 & A_3 & - & - & A_4 & - \\ - & B_1 & B_2 & B_3 & B_4 & B_5 & B_6 \end{array}$$

This alignment of A and B corresponds to a path through a rectangular grid of nodes, as in Figure C.1. Every alignment of A and B corresponds to a path from the upper left node to the lower right node. When moving from one node to another, it is possible to move, down, diagonally, or to the right.

Consider the case in which each of the segments in Figure X has an associated cost. We wish to find an alignment for which the associated net cost is a minimum. There are too many paths to consider the cost associated with each one, and so we employ an indirect approach which returns an optimum path in only $O(mn)$. The strategy is to start at the end point, and then for every node in the grid to determine the minimum cost to reach the end point, as well as the direction (right, down, diagonal) in which to travel.

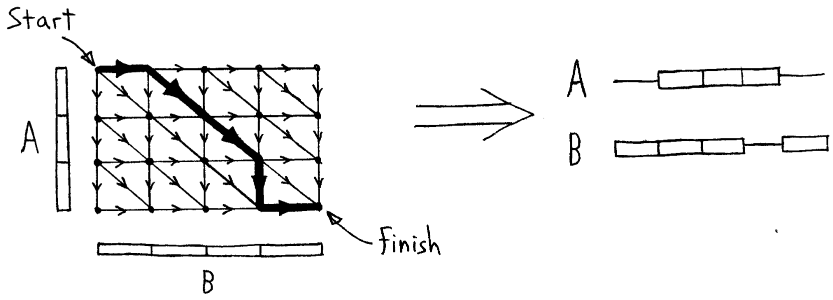


Figure C.1. Rectangular grid of nodes.

C.1 Matlab Implementation

```
function [cost,A]=seqalign(Dcost,Rcost,Ccost)
%seqalign finds the minimal cost alignment of two sequences.
%
% >>D=rand(10,10); R=2*rand(10,10); C=2*rand(10,10);
% >>[cost,A]=seqalign(D,R,C)

[R,C]=size(Dcost);
 optpath11=struct('cost',cell(R,C),'dir',cell(R,C));
 optpath12=struct('cost',num2cell(flipud(cumsum(flipud(Rcost(:,C))))),...
                  'dir',cell(R,1));
[optpath12.dir]=deal([1 0]);
 optpath21=struct('cost',num2cell(fliplr(cumsum(fliplr(Ccost(R,:))))),...
                  'dir',cell(1,C));
[optpath21.dir]=deal([0 1]);
optpath22=struct('cost',{0},'dir',{[0 0]});
optpath=[optpath11 optpath12; optpath21 optpath22];

for col=C:-1:1
    for row=R:-1:1
        Cdown= Rcost(row,col)+optpath(row+1,col  ).cost;
        Cright=Ccost(row,col)+optpath(row  ,col+1).cost;
        Cdiag= Dcost(row,col)+optpath(row+1,col+1).cost;
        if Cdown≤Cright & Cdown≤Cdiag
            optpath(row,col).cost=Cdown;
            optpath(row,col).dir =[1 0];
        elseif Cdiag≤Cright
            optpath(row,col).cost=Cdiag;
            optpath(row,col).dir =[1 1];
        else
            optpath(row,col).cost=Cright;
            optpath(row,col).dir =[0 1];
        end
    end
end
cost=optpath(1,1).cost;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if nargout==2  %Only compute A if it has been requested.
A=zeros(2,R+C);
row=1; col=1; index=1;
```

```
while row≠R+1 | col≠C+1
    Dr=optpath(row,col).dir(1);
    Dc=optpath(row,col).dir(2);
    A(:,index)=[row⋆Dr;col⋆Dc];
    row=row+Dr; col=col+Dc; index=index+1;
end
A=A(:,1:index−1);
end
```

Appendix D

The Linking Number

Let $\mathbf{x} : [0, l_x) \longrightarrow \mathbb{R}^3$ and $\mathbf{y} : [0, l_y) \longrightarrow \mathbb{R}^3$ be two closed curves in $\mathbb{R}^3$. The closed curve $\mathbf{x}$ is the boundary of an oriented surface D, over which there is a field of positive unit normal vectors $\mathbf{n}$. Generally, $\mathbf{y}$ intersects D transversally at a finite number of points $\mathbf{y}(t_k)$. The linking number of $\mathbf{x}$ with respect to $\mathbf{y}$ is given by

$$Lk(\mathbf{x}, \mathbf{y}) = \sum_k sign(\mathbf{n} \cdot \frac{\partial \mathbf{y}}{\partial t}), \tag{D.1}$$

where $\mathbf{n}$ and $\frac{\partial \mathbf{y}}{\partial t}$ are both evaluated at $\mathbf{y}(t_k)$. When $\mathbf{x}$ is a circle, one of the possible corresponding oriented surfaces D is a disk. Several basic examples are shown in Figure D.1. Self-intersections in D have no effect on the definition of $Lk(\mathbf{x}, \mathbf{y})$.

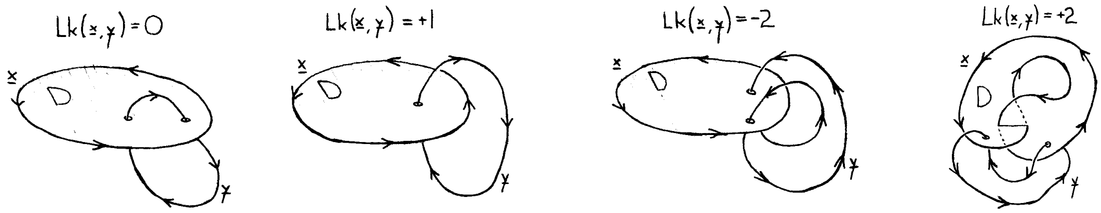


Figure D.1. Basic examples of the linking number.

D.1 Gauss's Integral Formula

One of the most intriguing things about $Lk(\mathbf{x}, \mathbf{y})$ is that it can be expressed as

$$Lk(\mathbf{x}, \mathbf{y}) = \frac{1}{4\pi} \int_T \left(\frac{\partial \mathbf{x}}{\partial s} \times \frac{\partial \mathbf{y}}{\partial t} \right) \cdot \frac{\mathbf{x}(s) - \mathbf{y}(t)}{\|\mathbf{x}(s) - \mathbf{y}(t)\|^3} \, ds \, dt, \tag{D.2}$$

where T is the torus $[0, l_x) \times [0, l_y)$. Gauss gave (D.2) with no derivation in a half page paper dated 1833. Our understanding of (D.2) follows from an analysis of the mapping $\mathbf{u} : T \longrightarrow S^2$, which assigns to (s, t) the unit vector from $\mathbf{x}(s)$ to $\mathbf{y}(t)$.

$$\mathbf{u}(s, t) = \frac{\mathbf{y}(t) - \mathbf{x}(s)}{\|\mathbf{y}(t) - \mathbf{x}(s)\|}. \tag{D.3}$$

We will see that $\mathbf{u}$ maps T into the unit sphere S^2 in a way that reflects the topological character of $\mathbf{x}$ and $\mathbf{y}$. The normalized area of the image of T is the linking number of $\mathbf{x}$ with respect to $\mathbf{y}$; (D.2) is simply a formula for this area.

Let $\mathbf{u}(s, [0, l_y))$ denote the image of the curve $\{s\} \times [0, t) \subset T$. This image is $\mathbf{y}$ as it would appear in the celestial sphere of a miniscule insect at $\mathbf{x}(s)$, (i.e., the projection of $\mathbf{y}$ onto the unit sphere centered at $\mathbf{x}(s)$). As s varies, so does $\mathbf{u}(s, [0, l_y))$. If increase in s moves $\mathbf{x}(s)$ up through the closed curve $\mathbf{y}$, (as in Figure D.2), then the images $\mathbf{u}(s, [0, l_y))$ comprise a tube that has collapsed around S^2. In fact, this tube is the image of a section of the torus T.

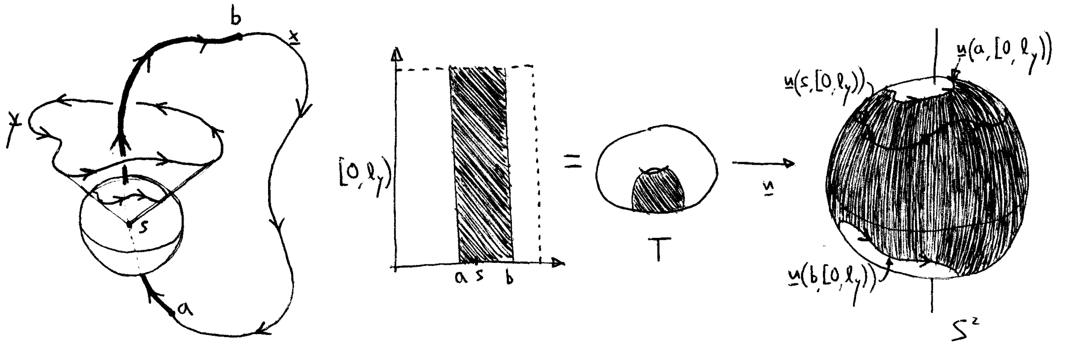


Figure D.2. When $\mathbf{x}$ passes through $\mathbf{y}$, the corresponding section of the torus T collapses around the unit sphere S^2.

In Figure D.3, we illustrate the case in which an increase in s moves $\mathbf{x}(s)$ around the outside of $\mathbf{y}$. In this case, the images $\mathbf{u}(s, [0, l_y))$ comprise a tube that has been squashed onto one side of S^2. As before, this tube is the image of a section of T.

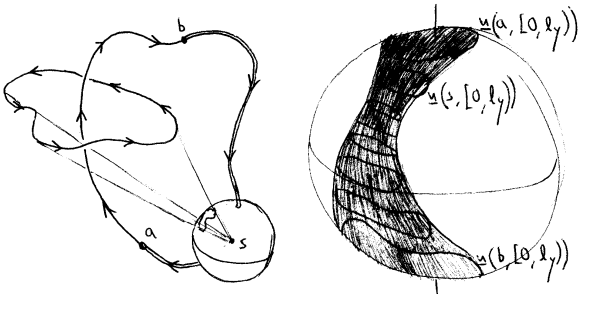


Figure D.3. When $\mathbf{x}$ passes around the outside of $\mathbf{y}$, the corresponding section of the torus T is squashed onto one side of the unit sphere S^2.

Qualitative differences in the images of sections of T are reflected in the (signed) areas of these images. A vanishingly small rectangle on T, with vertices (s, t), $(s + ds, t)$, $(s, t + dt)$, and $(s + ds, t + dt)$ has an image on S^2 with area dA given by

$$dA = \left(\frac{\partial \mathbf{u}}{\partial s} ds \times \frac{\partial \mathbf{u}}{\partial t} dt \right) \cdot \mathbf{u}. \tag{D.4}$$

The sign differences in dA cause the areas of the overlapping tube pieces from Figure D.3 to almost completely cancel one another. In contrast, all pieces of the tube from Figure D.2 have the same sign. Normalizing by 4π, (the area of S^2), we obtain the following measure on sections R of the torus T

$$A(R) = \frac{1}{4\pi}\int_R dA = \frac{1}{4\pi}\int_R \left(\frac{\partial \mathbf{u}}{\partial s} \times \frac{\partial \mathbf{u}}{\partial t}\right) \cdot \mathbf{u}\, ds\, dt. \tag{D.5}$$

As illustrated in Figures D.3 and D.2, $A(R)$ reflects whether $\mathbf{x}$ passes through a closed curve $\mathbf{y}$ or around it. When applied to the domain of $\mathbf{u}$ as a whole, A equals the linking number of $\mathbf{x}$ with respect to $\mathbf{y}$.

$$A(T) = Lk(\mathbf{x}, \mathbf{y}) = \frac{1}{4\pi}\int_T \left(\frac{\partial \mathbf{u}}{\partial s} \times \frac{\partial \mathbf{u}}{\partial t}\right) \cdot \mathbf{u}\, ds\, dt. \tag{D.6}$$

Because of the continuity of $\mathbf{u}$, T is mapped onto S^2 like a flexible rubber inner-tube squashed around a hard metallic ball. Either T covers all of S^2 (perhaps multiple times), or T doubles back on itself, and pieces of it have areas that exactly cancel. This is illustrated in Figure D.4.

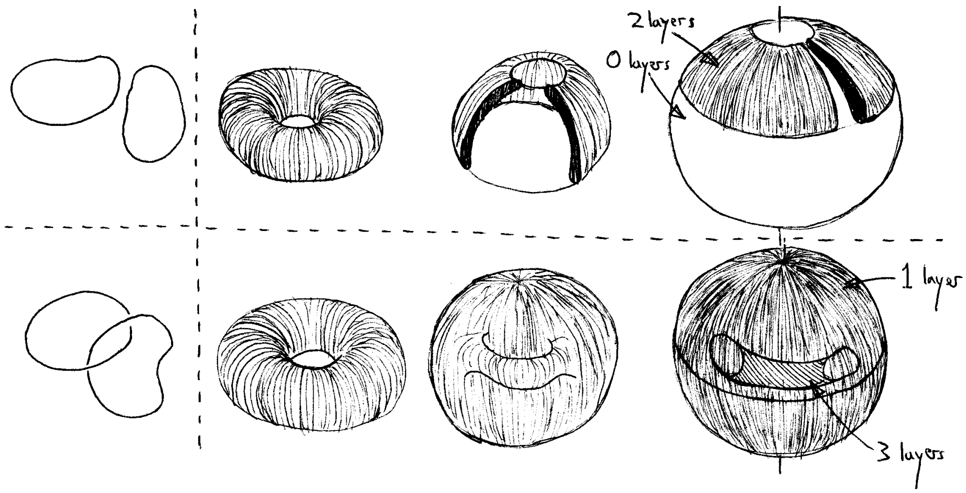


Figure D.4. Ways in which T can be mapped onto S^2.

Gauss's integral formula in its usual form (D.2) can be obtained from (D.6) by expanding the partials of $\mathbf{u}$.

$$\begin{aligned}
\frac{\partial \mathbf{u}}{\partial s} &= -\frac{\partial \mathbf{x}}{\partial s}\frac{1}{\|\mathbf{y}-\mathbf{x}\|} - \mathbf{u}\left(\frac{\partial \mathbf{x}}{\partial s}\cdot \mathbf{u}\right)\frac{1}{\|\mathbf{y}-\mathbf{x}\|}\\
\frac{\partial \mathbf{u}}{\partial t} &= \frac{\partial \mathbf{y}}{\partial t}\frac{1}{\|\mathbf{y}-\mathbf{x}\|} + \mathbf{u}\left(\frac{\partial \mathbf{y}}{\partial t}\cdot \mathbf{u}\right)\frac{1}{\|\mathbf{y}-\mathbf{x}\|}
\end{aligned} \tag{D.7}$$

A double occurance of a vector in a scalar triple product results in a zero, and so

$$\left(\frac{\partial \mathbf{u}}{\partial s} \times \frac{\partial \mathbf{u}}{\partial t}\right) \cdot \mathbf{u} = \left(\frac{\partial \mathbf{x}}{\partial s} \times \frac{\partial \mathbf{y}}{\partial t}\right) \cdot \frac{\mathbf{x}-\mathbf{y}}{\|\mathbf{x}-\mathbf{y}\|^3}, \tag{D.8}$$

showing that (D.6) is equivalent to (D.2). Note that s and t need not be arc-length parameters for $\mathbf{x}$ and $\mathbf{y}$.

Our understanding of Gauss's integral formula (D.2) follows what is often called a "degree of map" argument. A more formal derivation along these lines is given in [13], in which $Lk(\mathbf{x}, \mathbf{y})$ is defined as the Brouwer degree of the mapping (D.3). The Gauss integral integrates over T. An alternative is to work directly on S^2. $Lk(\mathbf{x}, \mathbf{y})$ can be found by picking a point on S^2 and counting the signs associated with the layers above this point. This corresponds to finding all crossings of two curves when they are projected onto the plane normal to the direction associated with the point chosen on S^2.

D.2 Writhing Number

The writhing number of a closed curve $\mathbf{x}$ is often called the *self-linking number* of $\mathbf{x}$ because it is given by

$$Wr(\mathbf{x}) = \frac{1}{4\pi} \int_M \left(\frac{\partial \mathbf{x}}{\partial s} \times \frac{\partial \mathbf{x}}{\partial t} \right) \cdot \frac{\mathbf{x}(s) - \mathbf{x}(t)}{\|\mathbf{x}(s) - \mathbf{x}(t)\|^3} \, ds \, dt, \tag{D.9}$$

that is, by exactly the same formula as (D.2), but with $\mathbf{y}(t)$ replaced by $\mathbf{x}(t)$. The torus $T = [0, l_x) \times [0, l_x)$ is mapped onto S^2 as it was before, except of course for points of the form $(s, s) \in T$ where the mapping (D.3) is undefined. These don't end up destroying the integral however; consider the set $U \subset T$ consisting of points (s, t) for which the arc-length along $\mathbf{x}$ from $\mathbf{x}(s)$ to $\mathbf{x}(t)$ is less than ϵ. As ϵ shrinks, $\mathbf{u}(s, t)$ becomes increasingly well aligned with the tangent vectors to $\mathbf{x}$ at $\mathbf{x}(s)$ and $\mathbf{x}(t)$. This causes the integrand of (D.9) to go to zero over U in spite of the $\|\mathbf{x}(s) - \mathbf{x}(t)\|^3$ term in the denominator. It follows that the value of $\mathrm{Wr}(\mathbf{x})$ can be understood in terms of the set $V = T - U$. Topologically, V is a doubly twisted band, as illustrated in Figure D.5; the writhing number of $\mathbf{x}$ is the normalized area of the image of this band on S^2.

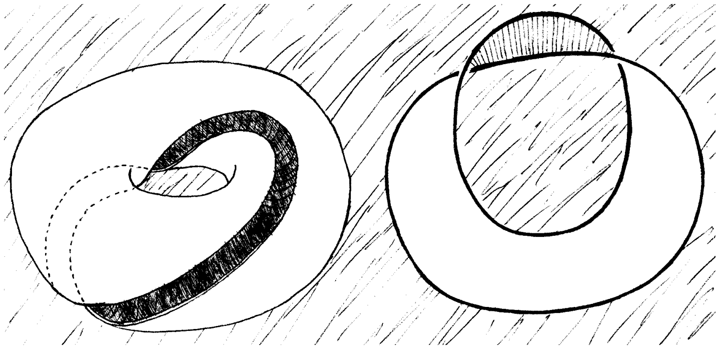


Figure D.5. Illustrations of the subset V of the torus T. Topologically, this subset is a doubly twisted band; the writhing number of a closed curve is the normalized area of the image of this band on S^2.