Protein Structure Elastic Network Models and the Positive Semidefinite Matrix Manifold

Xiao-Bo Li*, Forbes J. Burkowski, and Henry Wolkowicz

Abstract—Elastic network models (ENMs) assume pairs of α-carbons of a protein structure are connected by fictitious Hookean springs. A protein structure's potential energy is then a function of distance. If this distance is squared, the potential energy changes to a function on the positive semidefinite matrix manifold. This change is not trivial as it makes the relationship between ENMs and the rank 3 positive semidefinite (PSD) matrix manifold explicit; it suggests protein dynamics problems can be formulated as low rank semidefinite matrix manifold optimization problems. In this paper, we use normal mode analysis and elastic network interpolation to show the PSD matrix manifold is a reasonable tool for modelling protein dynamics.

Index Terms—elastic network model, Euclidean distance matrix, Gram matrix, positive semidefinite matrix manifold, protein structure, Riemannian manifold.

1 Introduction

Elastic network models (ENM) are a common and efficient tool for modelling protein dynamics. ENMs were first introduced by Tirion in the context of normal mode analysis (NMA) [25]. The author observed slow vibration normal modes produced by a simple Hookean potential energy function were in agreement with the modes found from complicated semi-empirical potentials. Subsequently, Kim et al. [13], [14], [15], [16], [17] and Jang [11] examined the generation of realistic transitional pathways between different protein conformations using ENMs. Kim [13] proposed elastic network interpolation (ENI) and showed ENI is more realistic than both Cartesian coordinate interpolation and internal coordinate (angle) interpolation by avoiding steric clashes. Many more authors have contributed to this direction of research, a detailed discussion is beyond the scope of this paper.

Under this modelling scheme, pairs of α-carbons are connected by fictitious Hookean springs. The protein structure's potential energy is consequently a function of distances between these α-carbons.

The potential energy can also be formulated using "distance-squared". This change is not trivial as it makes explicit the relation between ENMs and the rank 3 positive semidefinite (PSD) matrix manifold, a Riemannian manifold [12], [23], [27] . In classical mechanics, the potential energy, denoted U, is a function mapping a Riemannian manifold $\mathcal{M}$ to the set of real numbers $\mathbb{R}$, $U : \mathcal{M} \to \mathbb{R}$ [4], [7]. A linear vector space interpreted as a manifold is called a *linear manifold* [1]. While classical ENMs use the linear manifold, numerous mathematical properties of the PSD matrix manifold suggest it is a more natural choice for modelling proteins.

This paper is organized as follows. In Section 2, we review classical elastic network modelling, limiting the discussion to NMA and ENI. We then discuss PSD matrices, Euclidean distance matrices, and some mathematical properties that suggest they are a more natural choice for modelling protein structures in Sections 3 and 4. Finally, in Section 5 we present the potential energies for modelling protein structures on the PSD matrix manifold, again, limiting to NMA and ENI.

2 Elastic Network Models on Linear Manifolds

We will assume a protein is modelled by representing each amino acid residue by its α-carbon atom, and let n denote the number of α-carbons.

2.1 Normal Mode Analysis

The following potential energy for NMA was proposed by Tirion [25]to replace the more complicated semi-empirical potential energy:

$$\begin{aligned} U(\mathbf{y}) &= U(\mathbf{y}_0 + \delta) \\ &= \sum_{(i,j)\in\mathcal{D}} \frac{C}{2} \left(\|y_i - y_j\| - \|y_i^0 - y_j^0\| \right)^2 \\ &= \sum_{(i,j)\in\mathcal{D}} \frac{C}{2} \left(\|(y_i^0 + \delta_i) - (y_j^0 + \delta_j)\| - \|y_i^0 - y_j^0\| \right)^2 . \end{aligned} \tag{1}$$

C is a constant assumed to be the same for all interacting pairs [25]; we will assume $C = 1$. $y_i \in \mathbb{R}^3$ denote the vector of Cartesian coordinates of the i-th α-carbon atom. $y_i^0 \in \mathbb{R}^3$

- *Asterisk indicates corresponding author.*
- *XB Li is with the Cheriton School of Computer Science, University of Waterloo, ON Canada, N2L 3G1. Email: x22li@uwaterloo.ca.*
- *FJ Burkowski is with the Cheriton School of Computer Science, University of Waterloo, ON, Canada, N2L 3G1. E-mail: fjburkowski@uwaterloo.ca. Website: http://www.structuralbioinformatics.com/ and https://cs.uwaterloo.ca/about/people/fjburkowski.*
- *H. Wolkowicz is with the Department of Combinatorics and Optimization, University of Waterloo, ON, Canada, N2L 3G1. E-mail: hwolkowicz@uwaterloo.ca and website http://www.math.uwaterloo.ca/ hwolkowi/*

Manuscript received Month xx, 2016; revised Month xx, 2016.

denote the initial, equilibrium coordinates of α-carbon i, and $\delta_i \in \mathbb{R}^3$ is a perturbation vector. The set $\mathcal{D}$ contains pairs of indices indicating which α-carbons interact.

In order to extract the normal modes from equation (1), we need to find the Hessian matrix. Expanding the summand of equation (1) to second order gives:

$$U(\mathbf{y}_0 + \delta) \approx \sum_{(i,j)\in\mathcal{D}} (\delta_i - \delta_j)^T G_{ij} (\delta_i - \delta_j) \ . \tag{2}$$

Where:

$$G_{ij} = \frac{(y_i^0 - y_j^0)(y_i^0 - y_j^0)^T}{(y_i^0 - y_j^0)^T (y_i^0 - y_j^0)} \ . \tag{3}$$

When equation (2) is expressed using matrix multiplication, $\delta^T \mathcal{G} \delta$, $\mathcal{G}$ is the desired Hessian matrix. The set $\mathcal{D}$ is chosen to keep $\mathcal{G}$ sparse. For a protein with n α-carbons, $\mathcal{G}$ is a $3n \times 3n$ matrix with 3×3 blocks that have a Laplacian structure. The (i,j)-th block, for $i \neq j$ and $(i,j) \in \mathcal{D}$ is given by:

$$\mathcal{G}_{ij} = -G_{ij} \quad i \neq j \text{ and } (i,j) \in \mathcal{D} \ . \tag{4}$$

If $(i,j) \notin \mathcal{D}$, $\mathcal{G}(i,j)$ is a 3×3 zero matrix. The (i,i)-th diagonal block is given by,

$$\begin{aligned} \mathcal{G}_{ii} &= \sum_{k=1}^{i-1} G_{ki} + \sum_{k=i+1}^{n} G_{ik} \\ &= \sum_{k:k\neq i} G_{ki} \ . \end{aligned} \tag{5}$$

For example, $n = 3$ gives the following special 9×9 Laplacian matrix of 3×3 blocks of G_{ij}:

$$\mathcal{G} = \begin{pmatrix} G_{12} + G_{13} & -G_{12} & -G_{13} \\ -G_{12} & G_{12} + G_{23} & -G_{23} \\ -G_{13} & -G_{23} & G_{13} + G_{23} \end{pmatrix} \ . \tag{6}$$

Then, given a vector $\delta = (\delta_1^T, \delta_2^T, \delta_3^T)^T \in \mathbb{R}^9$, we have:

$$\delta^T G \delta = \sum_{i<j} (\delta_i - \delta_j)^T G_{ij} (\delta_i - \delta_j) \ . \tag{7}$$

Since $\delta = (\delta_1^T, \ldots, \delta_n^T)^T$ is an $\mathbb{R}^{3n}$ vector of all perturbations, both $\mathbf{y}$ and $\mathbf{y}_0$ are also vectors in $\mathbb{R}^{3n}$. Further, no additional requirements are placed on $\mathbf{y}$ and $\mathbf{y}_0$, thus classical ENMs are defined on the linear manifold $\mathbb{R}^{3n}$ [1].

2.2 Elastic Network Interpolation

In order to generate intermediate conformations between two given protein conformations, Kim et al. [13], [14], [15], [16], [17] proposed the following potential energy:

$$U_t(\delta) = \frac{1}{2} \sum_{(i,j)\in\mathcal{D}} (\parallel (y_i + \delta_i) - (y_j + \delta_j) \parallel - l_{ij}(t))^2 \ , \tag{8}$$

where δ_i is the optimal step size to arrive at the time t intermediate conformation from time $t-1$ α-carbon coordinates y_i's, and $l_{ij}(t)$ is the *linearly interpolated* targeted distance.

$$l_{ij}(t) = (1-t) \parallel y_i^0 - y_j^0 \parallel + t \parallel y_i^1 - y_j^1 \parallel \ . \tag{9}$$

The superscripts in y_i^0 and y_i^1 index the two end conformations.

As was the case for NMA, equation (8) can be expanded to second order:

$$\begin{aligned} U_t(\delta) \approx & \frac{1}{2} \sum_{(i,j)\in\mathcal{D}} (\delta_i - \delta_j)^T A_{ij} (\delta_i - \delta_j) \\ & + \sum_{(i,j)\in\mathcal{D}} B_{ij} (\delta_i - \delta_j) \\ & + \sum_{(i,j)\in\mathcal{D}} C_{ij} \ . \end{aligned} \tag{10}$$

Where:

$$\begin{aligned} A_{ij} &= I_3 - \frac{l_{ij}(t)}{\parallel y_i - y_j \parallel} \left(I_3 - \frac{(y_i - y_j)(y_i - y_j)^T}{\parallel y_i - y_j \parallel^2} \right) \ , \\ B_{ij} &= (\parallel y_i - y_j \parallel - l_{ij}(t)) \frac{(y_i - y_j)^T}{\parallel y_i - y_j \parallel} \ , \\ C_{ij} &= \frac{1}{2} (\parallel y_i - y_j \parallel - l_{ij}(t))^2 \ . \end{aligned} \tag{11}$$

I_3 is the 3×3 identity matrix. A_{ij} is a 3×3 matrix, B_{ij} is a 1×3 vector, and C_{ij} is a scalar. Whenever $(i,j) \notin \mathcal{D}$, A_{ij} will be a zero matrix, B_{ij} a zero vector, and C_{ij} a zero scalar. This can be expressed more concisely in matrix notation:

$$U_t(\delta) \approx \frac{1}{2} \delta^T \mathcal{A} \delta + \mathcal{B} \delta + c \ , \tag{12}$$

where $\mathcal{A}$ is a $3n \times 3n$ matrix with a Laplacian structure similar to equations (4) and (5) and $c \in \mathbb{R}$ is a constant. The vector $\mathcal{B}$ is given by:

$$\mathcal{B} = (\mathcal{B}_i, \ldots, \mathcal{B}_n) \in \mathbb{R}^{1\times 3n} \ , \tag{13}$$

where:

$$\mathcal{B}_i = -\sum_{k=1}^{i-1} B_{ki} + \sum_{k=i+1}^{n} B_{ik} = \sum_{k:k\neq i} B_{ki} \quad i = 1, \ldots n \ . \tag{14}$$

Equation (12) is minimized by the optimal displacement δ^* that solves the linear system:

$$\mathcal{A} \delta^* = \mathcal{B} \ , \tag{15}$$

which sets the derivative of equation (12) to zero. Note the similarity between equations (1) and (8). Both potential energies are the sum of squared difference of distances.

3 Positive Semidefinite Matrices and Euclidean Distance Matrices

In the above discussion on ENMs, we used the vector $y = (y_1^T, \ldots, y_n^T)^T \in \mathbb{R}^{3n}$ to represent all the α-carbon coordinates of the protein. Consider the following change, let Y be an $n \times 3$ matrix whose rows are 1×3 blocks containing α-carbon coordinates:

$$Y = \begin{pmatrix} y_1^T \\ \vdots \\ y_n^T \end{pmatrix} \in \mathbb{R}^{n\times 3} \ . \tag{16}$$

The **Gram matrix**, denoted X, for this set of α-carbons is then given by $X = YY^T$. A Gram matrix is *centered* if the centroid of all the α-carbons $y_1, \ldots, y_n$ is the origin. We will assume the Gram matrix is centered.

The **Euclidean Distance Matrix** (EDM), D, for this same set of α-carbon atoms is defined as the matrix whose ij-th entry is given by the distance-squared between α-carbon i and j,

$$D_{ij} = (y_i - y_j)^2 = (y_i - y_j)^T(y_i - y_j) \ . \tag{17}$$

Gram matrices and EDMs are important objects in semidefinite programming. The following papers are only a small sample of research using these objects: [2], [3], [6], [18], [22], [23].

4 Mathematical Significance

In this section, we discuss some mathematical properties of PSD matrices and EDMs not available to the linear manifold currently used by classical ENMs. These properties suggest PSD matrices may be a more "natural" choice for modelling proteins.

4.1 The Gram Matrix is Invariant to Rotation

A rotation matrix Q is a 3×3 orthogonal matrix, $QQ^T = Q^TQ = I$. When all the atomic coordinates of a protein are rotated together, the protein structure has no net change. This property is exactly reflected in the Gram matrix since:

$$(YQ)(YQ)^T = YQQ^TY^T = YY^T \ . \tag{18}$$

The linear manifold does not capture this invariance.

4.2 Each Gram Matrix can be Linearly Mapped to an Unique Euclidean Distance Matrix

The centered Gram matrix and its corresponding EDM are intimately related via the linear *bijective* mapping, often denoted $\mathcal{K}(\cdot)$ [8], [9], [18]. Thus, each Gram matrix $X = YY^T$ uniquely maps to an EDM, D, $\mathcal{K}(X) = D$. Let $e \in \mathbb{R}^n$ be the vector of all ones. Let $\text{diag}(\cdot)$ be the operator extracting the diagonal of a matrix. The mapping $\mathcal{K}(X)$ is given by:

$$\mathcal{K}(X) = \text{diag}(X)e^T + e\text{diag}(X)^T - 2X \ . \tag{19}$$

No such bijective linear map is known between linear manifolds and the matrix of distances.

4.3 The Set of $n \times n$ EDMs is a Convex Cone

The set of $n \times n$ Gram matrices forms a convex cone, the PSD cone. Further, for a given $n > 0$, the set of $n \times n$ EDMs also forms a convex cone. See [9], [18] for a more detailed discussion.

However, when we take the square root of the entries of these EDMs to get matrices containing distances, this set of $n \times n$ $\sqrt{EDM}$ matrices is no longer convex for $n > 3$ [9]. Equation (9) attempts to interpolate between the $\sqrt{EDM}$ matrices of the two end conformations, dispite the cone being nonconvex.

4.4 The Set of Fixed Rank PSD Matrices Form a Riemannian Manifold

A protein structure lies in 3 dimensional space, therefore its Gram matrix is rank 3. The set of such fixed rank PSD matrices is a Riemannian manifold. This manifold structure was first discussed in the context of optimization algorithms on matrix manifolds, for example: [12], [22], [23], [27]. The geometry of a fixed rank PSD matrix manifold is not unique, see the discussion in [27]. Since the Gram matrix is invariant to rotation, the geometry relevant to this paper's discussion is the quotient geometry discussed in Section 6.6.2 of [27].

This fact is important because classical mechanics requires the potential energy to be defined on a Riemmannian manifold [4], [7]. That the set of rank 3 PSD matrices is also a Riemannian manifold, implies this mathematical structure and the corresponding EDMs may be considered as tools for modelling proteins.

4.4.1 Rank Constraint is not Convex

Although the PSD cone and EDM cone are convex, when we require solutions of a certain rank, for proteins this is 3, the problem to be solved is no longer convex. Rank constraint is often left out in semidefinite optimization or treated with heuristics; matrix manifold algorithms are one such heuristic. It is beyond the scope of this paper to discuss these heuristics in detail.

4.5 The PSD Cone has Faces

A set of atoms whose mutual distances stay fixed regardless of the protein's conformation is called a *rigid cluster* in the context of ENMs [11], [17] and a *clique* in the context of sensor network localization [18], [19].

Krisklock and Wolkowicz [18], [19] showed how such cliques can be used to find the face of the Gram matrix, this process is called *facial reduction* and allows semidefinite problems to be strictly feasible and also more efficient to solve.

Faces of a Gram matrix are relevant to proteins since proteins contain many rigid clusters. These rigid clusters restrict the protein's Gram matrix to be on a certain face of the rank 3 PSD matrix manifold; this property is further suggesting that the PSD matrix manifold can "naturally" express a protein's structural properties.

5 Elastic Network Models on the Positive Semidefinite Matrix Manifold

5.1 Normal Mode Analysis

By squaring the distances in equation (1), we get the following potential energy:

$$\begin{aligned} U(Y) &= U(Y_0 + \Delta) \\ &= \sum_{(i,j)\in\mathcal{D}} (y_i - y_j)^T(y_i - y_j) - (y_i^0 - y_j^0)^T(y_i^0 - y_j^0))^2 \\ &= \sum_{(i,j)\in\mathcal{D}} ((y_i^0 + \delta_i) - (y_j^0 + \delta_j))^T((y_i^0 + \delta_i) - (y_j^0 + \delta_j)) - D_{ij}^0)^2 \\ &= \sum_{(i,j)\in\mathcal{D}} ((e_i - e_j)^T(Y_0 + \Delta)(Y_0 + \Delta)^T(e_i - e_j) - D_{ij}^0) \ , \end{aligned} \tag{20}$$

where:

$$Y_0 = \begin{pmatrix} (y_1^0)^T \\ \vdots \\ (y_n^0)^T \end{pmatrix} \in \mathbb{R}^{n\times 3} \ , \tag{21}$$

and:

$$\Delta = \begin{pmatrix} (\delta_1)^T \\ \vdots \\ (\delta_n)^T \end{pmatrix} \in \mathbb{R}^{n\times 3} \ , \tag{22}$$

Following the convention of equation (17),

$$D_{ij}^0 = (y_i^0 - y_j^0)^T (y_i^0 - y_j^0) \ , \tag{23}$$

are the entries of the equilibrium EDM. $e_i \in \mathbb{R}^n$ has a 1 at position i and zero otherwise. This is actually the same format as the objective function seen in Meyer [22] for solving low rank distance matrix completion. Due to the $\mathcal{K}(X)$ relation, equation (20) can be expressed concisely in terms of the rank 3 PSD matrix as seen in [23], and in fact is a well-known objective function for Euclidean distance matrix completion first seen in [2].

$$U(\Delta) = \| \ H \odot (\mathcal{K}((Y_0 + \Delta)(Y_0 + \Delta)^T) - D_0) \ \|_F^2 \ . \tag{24}$$

H is a matrix such that $H_{ij} = 1$ if $(i, j) \in \mathcal{D}$ and $H_{ij} = 0$ otherwise. $\odot$ is elementwise multiplication. This potential energy gives the following value for G_{ij} in the second order Taylor expansion:

$$G_{ij} = 4(y_i^0 - y_j^0)(y_i^0 - y_j^0)^T \ . \tag{25}$$

Compared to equation (3), equation (25) is missing the division by $(y_i^0 - y_j^0)^T (y_i^0 - y_j^0)$. We first discussed the potential for NMA on the PSD matrix manifold in [21]. There, we referred to "distance-squared" as "quadrance" following terminology from [28]

5.1.1 The Potential Energy Defined using Distance and Distance-Squared Agrees with Each Other

Tirion [25] justified the use of the Hookean potential energy by showing the density of eigenvalues and the root mean square (rms) fluctuations given by equation (1) agreed with the semi-empirical potential. We first made the same observation for the potential energy on the positive semidefinite matrix manifold, equation (20), in [21]. Tirion's analysis mainly focused on the G-Actin protein, pdb id 1ATN, therefore we will use this same protein in the current discussion. In [21], we gave examples of other proteins, and the conclusion is very similar. We will not repeat those figures here due to space limitations.

Ben-Avraham [5] observed the eigenvalues have a density graph that is similar for many globular proteins, a shape he called the "universal curve". Tirion showed 1ATN's lower modes given by the Hookean potential energy matched the universal curve. In Figure 1, we show that for 1ATN, this universal curve shape is seen for both distance and distance-squared eigenvalue densities. These histograms were generated using pyplot [10].

Tirion [25] also examined the root mean square (rms) deviations of α-carbons from equilibrium. Two measures of deviation were given. The first is the rms fluctuation of all α-carbons as as function of mode, the second is the rms fluctuation per residue for all modes. We denote these σ_k, σ^i respectively following [13]. See also [26].

σ_k drops off because lower modes represent high amplitude motions. The same shape is given by both distance and distance-squared potential energies in Figure 2.

The σ^i graph is given in Figure 3, as can be seen, distance and distance-square potential energies give the same shape.

The formulas for σ_k and σ^i been described previously in for example [13], [26].

$$\sigma_k = \left(\sum_{i=1}^{n} \frac{(\sigma_k^i)^2}{n} \right)^{\frac{1}{2}} \ , \tag{26}$$

where

$$\sigma_k^i = \left\| v_k^i \frac{\alpha_k}{\sqrt{2}} \right\| \ , \tag{27}$$

and $v_k = ((v_k^1)^T, \ldots, (v_k^n)^T)^T \in \mathbb{R}^{3n}$ is the eigenvector for mode k. The authors in [13], [26] have used an α_k value of:

$$\alpha_k = \left(\frac{2k_B T}{\lambda_k} \right)^{\frac{1}{2}} \ , \tag{28}$$

where λ_k is the k-th eigenvalue, k_B is the Boltzmann constant, and T is temperature. However, since the constants do not affect the shape of the RMS plots, we have ignored them and used an α_k value of:

$$\alpha_k = \frac{1}{\sqrt{\lambda_k}} \ . \tag{29}$$

The σ^i is given by:

$$\sigma^i = \left(\sum_{k=7}^{3n} (\sigma_k^i)^2 \right)^{\frac{1}{2}} \ . \tag{30}$$

The first 6 eigenvalues our zero, so the summation ignores them.

The agreement in the shape of the rms fluctuation curve is consistent for many proteins. In Figure 4, we present some more protein examples.

5.2 Elastic Network Interpolation

Squaring the distance in equation (8) gives:

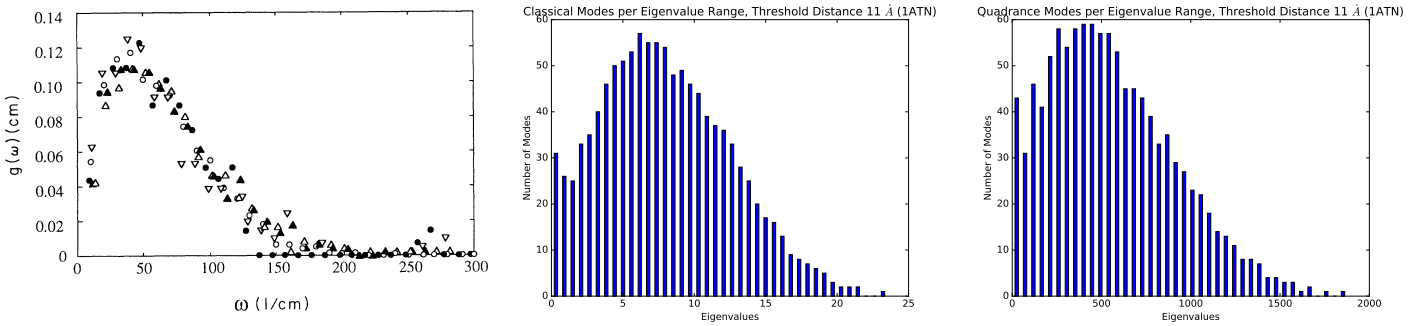
$$\begin{aligned} &U_t(\delta) \\ &= \frac{1}{2} \sum_{(i,j)\in\mathcal{D}} \left(\| \ (y_i + \delta_i) - (y_j + \delta_j) \ \|^2 - D_{ij}(t) \right)^2 \ , \end{aligned} \tag{31}$$

with:

$$\begin{aligned} &\| \ (y_i + \delta_i) - (y_j + \delta_j) \ \|^2 \\ &= ((y_i + \delta_i) - (y_j + \delta_j))^T ((y_i + \delta_i) - (y_j + \delta_j)) \ . \end{aligned} \tag{32}$$

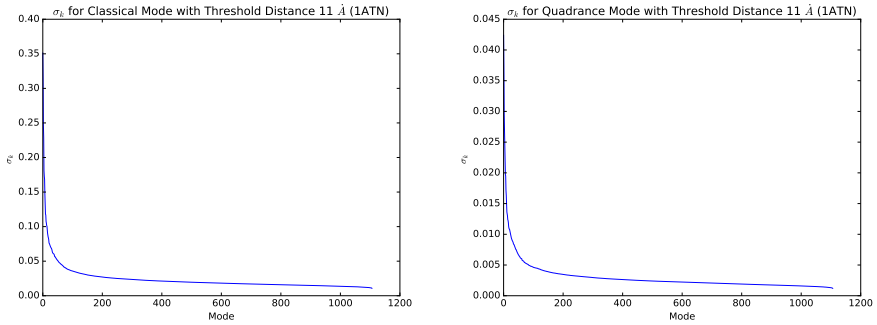
In the NMA case, the equilibrium coordinates, Y_0, were perturbed. In the ENI case, the Y matrix are the coordinates from the previous time period, $t-1$, and we seek the best perturbation to get as close as possible to the time t $D_{ij}(t)$ value. We redefine equation (9) to be the targeted distance-squared $D_{ij}(t)$:

$$D_{ij}(t) = (1-t)(y_i^0 - y_j^0)^T (y_i^0 - y_j^0) + t(y_i^1 - y_j^1)^T (y_i^1 - y_j^1) \ , \tag{33}$$



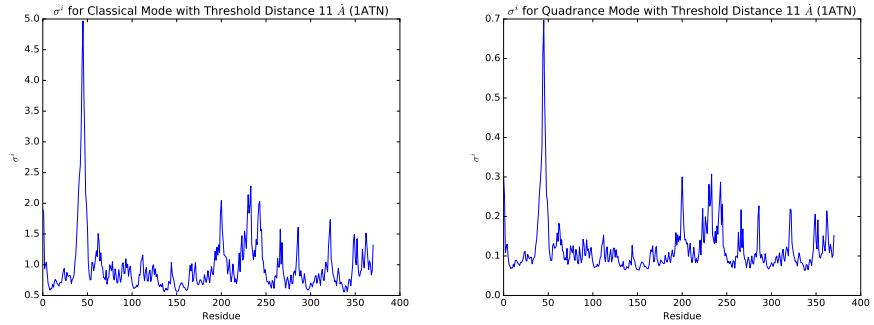
(a) "Universal curve" taken from [5]. (b) Density for distance. (c) Density for distance-squared.

Fig. 1. The density of eigenvalues for 1ATN. Note that distance-squared follow the "universal curve".



(a) Distance potential. (b) Distance-squared potential.

Fig. 2. RMS fluctuation per mode for 1ATN has the same shape for distance and distance-squared potentials.



(a) Distance potential. (b) Distance-squared potential.

Fig. 3. RMS fluctuation per residue for 1ATN has the same shape for distance and distance-squared potentials.

The superscripts in y_i^0 and y_i^1 index the two end conformations as before. As in the case of normal mode analysis, we can use the $\mathcal{K}(\cdot)$ map to concisely express equation (31) in matrix form:

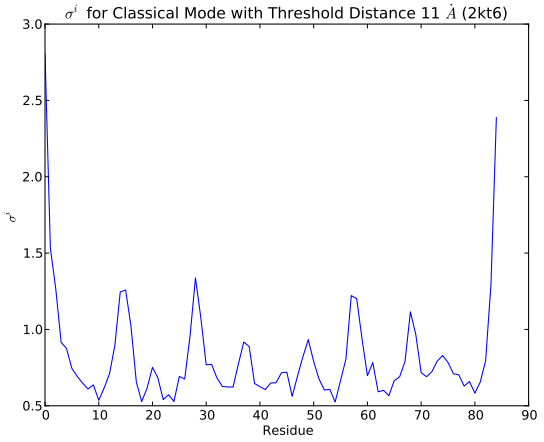
$$U_t(\Delta) = \| H \odot (\mathcal{K}((Y+\Delta)(Y+\Delta)^T) - D_t) \|_F^2 \ . \quad (34)$$

D_t is now a convex combination of the two end EDMs, D_0 and D_1:

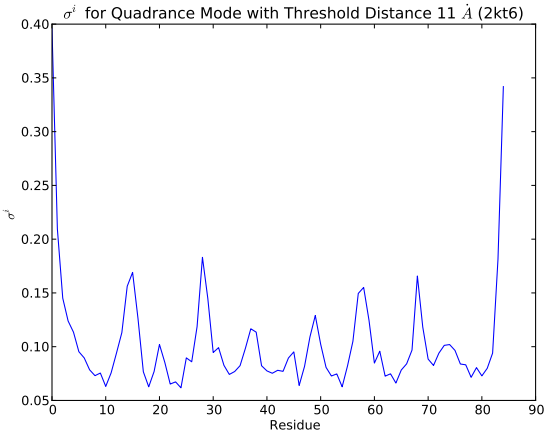
$$D_t = (1-t)D_0 + tD_1 \ . \quad (35)$$

Since the EDM cone is convex, D_t will always lie in the cone. As mentioned in Section 4.4.1, rank constraint is not convex. This means the matrix D_t might not be representing a 3 dimension EDM. However, the potential energy given by equation (31) and (34) will finding the best rank 3 approximation. We first discussed the potential for ENI on the PSD matrix manifold in [20].

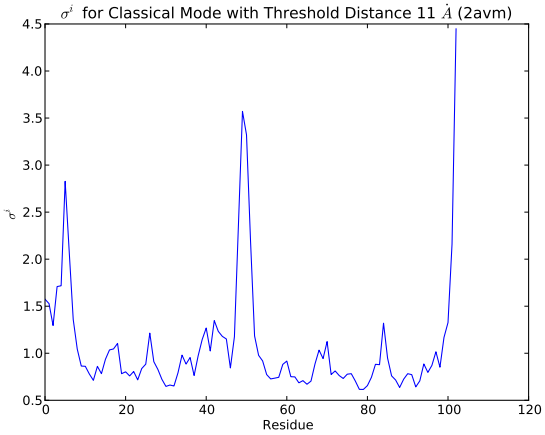
This potential energy gives the following values for A_{ij}, B_{ij}, and C_{ij} in the second order Taylor expansion:



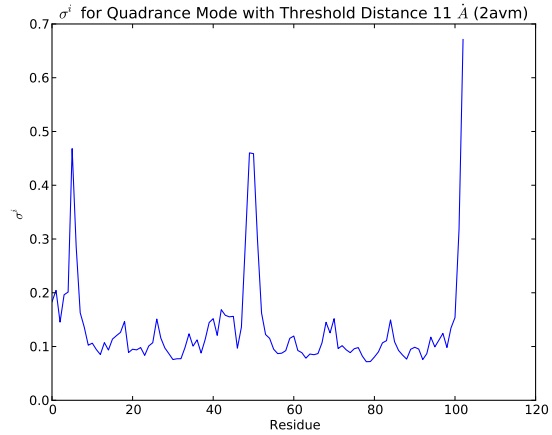
(a) 2KT6 σ^i, distance potential.



(b) 2KT6 σ^i, distance-squared potential.



(c) 2AVM σ^i, distance potential.



(d) 2AVM σ^i, distance-squared potential.

Fig. 4. More examples of σ^i graph shapes agreeing. "Quadrance" refers to distance-squared.

$$
\begin{aligned}
A_{ij} &= 2((y_i - y_j)^T(y_i - y_j) - D_{ij}(t))I_3 \\
&\quad + 4(y_i - y_j)(y_i - y_j)^T \,, \\
B_{ij} &= 2\left((y_i - y_j)^T(y_i - y_j) - D_{ij}(t)\right)(y_i - y_j)^T \,, \\
C_{ij} &= \frac{1}{2}\left((y_i - y_j)^T(y_i - y_j) - D_{ij}(t)\right)^2 \,.
\end{aligned} \tag{36}
$$

5.3 Sample Transition Comparison

Consider the two lattice protein structures in Figure 5 for illustrative purposes. We interpolated from the initial to the final conformation by solving the linear system in equation (15) using both distance and distance-squared formulas from Section 2.2 and 5.2. To ensure the bond length is preserved, we explicitly ensured the vector difference between α carbon i and $i+1$ with coordinates y_i and y_{i+1}, given by $t_i = y_{i+1} - y_i$, always has length one, the original lattice structure bond length. The vector t_i is called the tangent vector.

We observe that the distance-squared transition pathway in Figure 7 is in agreement with the distance transition pathway in Figure 6.

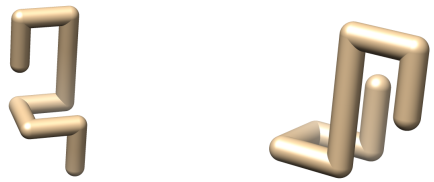
In our previous publication [20], the tCG algorithm produced a different pathway when we did not restrict the tangent vector length. However, when we did restrict the tangent length, we found the tCG algorithm also gave the same pathway for this case, but the time for these transitions is different. This is shown in Figure 8. The tCG algorithm is found in [1] with the Hessian and gradient formulas given by Meyer [22].

Finally, we note that all of the above ending conformations are chirally different from the targeted conformation.

Consider now the following two conformations given by figure 9. We interpolated these structures using the linear manifold and the PSD matrix manifold just as we done for the structures in Figure 5. The results are given in Figure 10 and 11 respectively. In this case, the figures are not the same. However the linear manifold has ended up with the correct chirality.

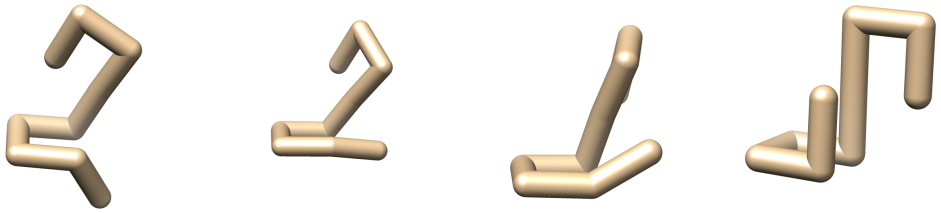
5.4 Remarks and Considerations for Futures Research

In this current paper, we generated the lattice transitions, for both the linear and PSD matrix manifold, using the python scripting environment in UCSF Chimera [24]. Matlab was



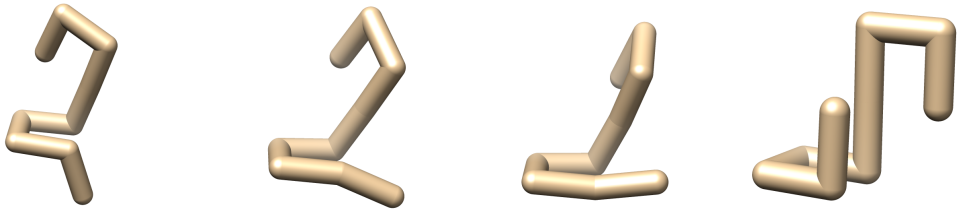
(a) Initial conformation. (b) Targeted conformation.

Fig. 5. First pair of lattice structures for illustrating ENI.



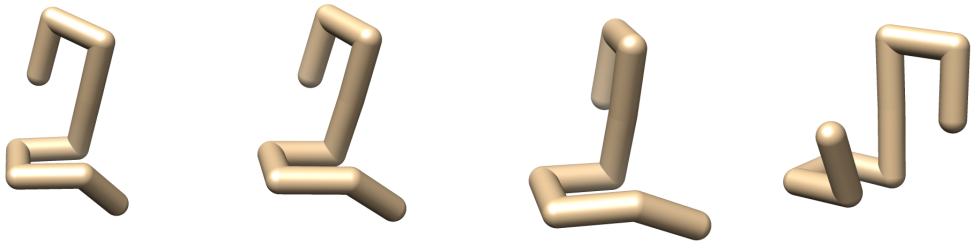
(a) $t = 0.15$. (b) $t = 0.26$. (c) $t = 0.33$. (d) Stable ending conformation reached by $t = 0.49$.

Fig. 6. Transition on the linear manifold. Figure 6 and 7 are consistent. The ending conformation is chirally different from the targeted one.



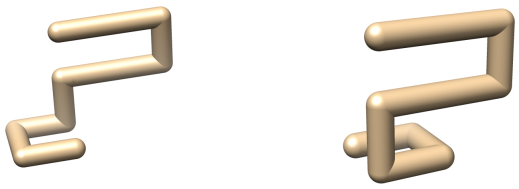
(a) $t = 0.15$. (b) $t = 0.26$. (c) $t = 0.33$. (d) Stable ending conformation reached by $t = 0.49$.

Fig. 7. Transition on the rank 3 PSD matrix manifold. Figure 6 and 7 are consistent. The ending conformation is chirally different from the targeted one.



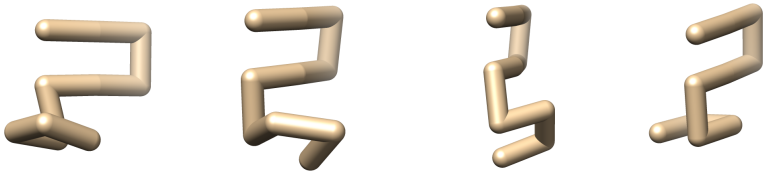
(a) $t = 0.39$. (b) $t = 0.45$. (c) $t = 0.50$. (d) Stable ending conformation reached by $t = 0.80$.

Fig. 8. Transition on the rank 3 PSD matrix manifold using the tCG algorithm as in [20] with bond length constraints. Consistent with Figure 6 and 7, but times are very different. The ending conformation is chirally different from the targeted one.



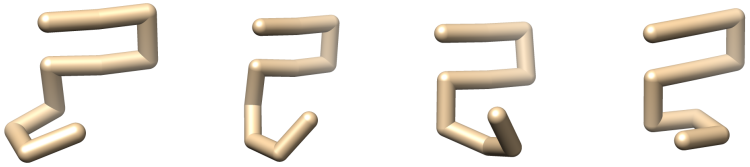
(a) Initial conformation. (b) Targeted conformation.

Fig. 9. Second pair of lattice structures for illustrating ENI.



(a) $t = 0.20$. (b) $t = 0.28$. (c) $t = 0.37$. (d) Stable ending conformation reached by $t = 0.47$.

Fig. 10. Transition on the linear manifold. Figure 10 and 11 are **not** consistent. The ending conformation is chirally the same as the targeted one.



(a) $t = 0.20$. (b) $t = 0.28$. (c) $t = 0.37$. (d) Stable ending conformation reached by $t = 0.47$.

Fig. 11. Transition on the rank 3 PSD matrix manifold. Figure 10 and 11 are **not** consistent. The ending conformation is chirally different from the targeted one.

not used, and we did not observe some of the anomalies for using the linear manifold as described in our previous publication [20] when using the code provided by Kim [1]. Thus, it seems the anomalies were due to the linear algebra libraries and not because of the use of distance in the potential energy.

However, as the previous section showed, we still observed chirality differences and the possibility of a different pathway.

Since lattice proteins are not realistic proteins, we leave as the subject of future research the investigation of any differences in interpolating real protein pathways and any improvements to these algorithms.

The close relationship between ENMs and Semidefinite programming is the focus of this current paper.

6 Conclusion

The mathematical properties of the PSD matrix manifold suggests that this matrix manifold is suitable for modelling protein dynamics. Working with this manifold is equivalent to using distance-squared instead of distance in the currently popular ENMs. We presented a potential energy on this manifold, which has already been used in semidefinite optimization as an objective function for distance matrix completion. This potential energy has the general form:

$$U(X) = \| \mathcal{K}(X) - D \|_F^2 \quad . \tag{37}$$

X is a PSD matrix. Both $\mathcal{K}(X)$ and D are EDMs. $\| \cdot \|_F^2$ denote the Frobenius norm. The entries of D depend on the application.

When applied to NMA, this potential energy is in agreement with the Hookean potential energy introduced by Tirion. When applied to ENI, D is an interpolated EDM, a convex combination on a convex EDM cone. This potential energy may propose a different transitional pathway than the one using distances introduced by Kim. However, we have only looked at unrealistic lattice structures in this paper; further investigation of real protein structures will be the subject of future research. For this, we require rigid clusters, groups of atoms in a protein that move concurrently, to be accommodated by the model; this will also be the subject of future research.

1. Matlab code for ENI is available from the KOSMOS website http://bioengineering.skku.ac.kr/kosmos/tutorial.php. This code does not handle rigid clusters.

ENMs are closely related to semidefinite matrix manifolds; we are still in the early stages of examining and exploiting this relationship.

APPENDIX

DERIVATIVE OF DISTANCE

Let $x \in \mathbb{R}^n$.

$$\frac{d \parallel x \parallel}{dx} = \frac{x}{\parallel x \parallel} . \tag{38}$$

DERIVATIVE OF DISTANCE SQUARED

Let $x \in \mathbb{R}^n$.

$$\frac{d \parallel x \parallel^2}{dx} = \frac{d(x^T x)}{dx} = 2x . \tag{39}$$

DERIVATIVE OF DISTANCE TIMES VECTOR

Let $x \in \mathbb{R}^n$ and I_n be the $n \times n$ identity matrix. Note that $\parallel x \parallel$ is a scalar so the order of multiplication does not matter. The following expression uses to product rule for taking derivatives.

$$\frac{d(x \parallel x \parallel)}{dx} = \frac{d(\parallel x \parallel x)}{dx} = \parallel x \parallel I_n + \frac{xx^T}{\parallel x \parallel^2} . \tag{40}$$

SECOND ORDER EXPANSION FOR DISTANCES

Let $d \in \mathbb{R}$ be a scalar and $x \in \mathbb{R}^n$ be a vector, and the function to be expanded be:

$$f(\delta) = \frac{1}{2} \left(\parallel x + \delta \parallel - d \right)^2 . \tag{41}$$

The second order expansion is:

$$f(\delta) \approx f(0) + \text{grad} f(0)^T \delta + \frac{1}{2} \delta^T \text{Hess} f(0) \delta . \tag{42}$$

The constant term $f(0)$ is given by:

$$f(0) = \frac{1}{2} \left(\parallel x \parallel - d \right)^2 . \tag{43}$$

$\text{grad} f(0)$ is an $n \times 1$ vector given by:

$$\text{grad} f(0) = \left(\parallel x \parallel - d \right) \frac{x}{\parallel x \parallel} . \tag{44}$$

$\text{Hess} f(0)$ is an $n \times n$ matrix given by, using the product rule on $\text{grad} f(0)$:

$$\begin{aligned} \text{Hess} f(0) &= \left(\parallel x \parallel - d \right) \left(\frac{\parallel x \parallel I_n - \frac{xx^T}{\|x\|^2}}{\parallel x \parallel^2} \right) + \frac{xx^T}{\parallel x \parallel^2} \\ &= I_n - \frac{d}{\parallel x \parallel} \left(I_n - \frac{xx^T}{\parallel x \parallel^2} \right) . \end{aligned} \tag{45}$$

SECOND ORDER EXPANSION FOR DISTANCES-SQUARED

Let $d \in \mathbb{R}$ be a scalar and $x \in \mathbb{R}^n$ be a vector, and the function to be expanded be:

$$f(\delta) = \frac{1}{2} \left((x + \delta)^T (x + \delta) - d \right)^2 . \tag{46}$$

The second order expansion is:

$$f(\delta) \approx f(0) + \text{grad} f(0)^T \delta + \frac{1}{2} \delta^T \text{Hess} f(0) \delta . \tag{47}$$

The constant term is given by:

$$f(0) = \frac{1}{2} \left(x^T x - d \right)^2 . \tag{48}$$

The first derivative term is an $n \times 1$ vector given by:

$$\text{grad} f(0) = 2 \left(x^T x - d \right) x . \tag{49}$$

The second derivative term is an $n \times n$ matrix given by:

$$\text{Hess} f(0) = 2 \left(x^T x - d \right) I_n + 4xx^T . \tag{50}$$

REFERENCES

[1] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*. Princeton, NJ: Princeton University Press, 2008.

[2] A. Alfakih, A. Khandani, and H. Wolkowicz, "Solving Euclidean distance matrix completion problems via semidefinite programming," *Computational Optimization and Applications*, vol. 12, no. 1-3, pp. 13–30, January 1999.

[3] B. Alipanahi, N. Krislock, A. Ghodsi, H. Wolkowicz, L. Donaldson, and M. Li, "Determining protein structures from noesy distance constraints by semidefinite programming," *Journal of Computational Biology*, vol. 20, no. 4, pp. 296–310, 2013.

[4] V. Arnold, *Mathematical Methods of Classical Mechanics*. Springer-Verlag, 1978.

[5] D. ben-Avraham, "Vibrational normal-mode spectrum of globular proteins," *Physical Review B*, vol. 47, no. 21, p. 14559, 1993.

[6] P. Biswas, K.-C. Toh, and Y. Ye, "A distributed SDP approach for large-scale noisy anchor-free graph realization with applications to molecular conformation," *SIAM Journal on Scientific Computing*, vol. 30, no. 3, pp. 1251–1277, 2008.

[7] O. Calin and D.-C. Chang, *Geometric mechanics on Riemannian manifolds: applications to partial differential equations*. Springer Science & Business Media, 2006.

[8] F. Critchley, "On certain linear mappings between inner-product and squared-distance matrices," *Linear Algebra and its Applications*, vol. 105, pp. 91–107, 1988.

[9] J. Dattorro, *Convex optimization and Euclidean distance geometry*. MeBoo, 2014.

[10] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing In Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[11] Y. Jang, "Hybrid elastic network model for macromolecular dynamics," Ph.D. dissertation, University of Massachusetts Amherst, 2008.

[12] M. Journée, F. Bach, P.-A. Absil, and R. Sepulchre, "Low-rank optimization on the cone of positive semidefinite matrices," *SIAM J. OPTIM*, vol. 20, no. 5, pp. 2327–2351, May 2010.

[13] M. K. Kim, "Elastic network models of biomolecular structure and dynamics," Ph.D. dissertation, The Johns Hopkins University, 2004.

[14] M. K. Kim, G. S. Chirikjian, and R. L. Jernigan, "Elastic models of conformational transitions in macromolecules," *Journal of Molecular Graphics and Modelling*, vol. 21, no. 2, pp. 151–160, 2002.

[15] M. K. Kim, Y. Jang, and J. I. Jeong, "Using harmonic analysis and optimization to study macromolecular dynamics," *International Journal of Control Automation and Systems*, vol. 4, no. 3, pp. 382–393, 2006.

[16] M. K. Kim, R. L. Jernigan, and G. S. Chirikjian, "Efficient generation of feasible pathways for protein conformational transitions," *Biophysical Journal*, vol. 83, no. 3, pp. 1620–1630, 2002.

[17] ——, "Rigid-cluster models of conformational transitions in macromolecular machines and assemblies," *Biophysical journal*, vol. 89, no. 1, pp. 43–55, 2005.
[18] N. Krislock, "Semidefinite facial reduction for low-rank Euclidean distance matrix completion," Ph.D. dissertation, School of Computer Science, University of Waterloo, 2010.
[19] N. Krislock and H.Wolkowicz, "Explicit sensor network localization using semidefinite representations and facial reductions," *SIAM Journal on Optimization*, vol. 20, no. 5, pp. 2679–2708, 2010.
[20] X. Li and F. Burkowski, "Generating conformational transitions using the Euclidean distance matrix." *IEEE transactions on nanobioscience*, 2015.
[21] X. LI, B. Forbes, and H. Wolkowicz, "Protein structure normal mode analysis on the positive semidefinite matrix manifold," in *8th International Conference on Bioinformatics and Computational Biology (BICOB 2016)*, 2016.
[22] G. Meyer, "Geometric optimization algorithms for linear regression on fixed-rank matrices," Ph.D. dissertation, University of Liège, 2011.
[23] B. Mishra, G. Meyer, and R. Sepulchre, "Low-rank optimization for distance matrix completion," in *2011 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, Orlando, FL, USA, December 12-15 2011, pp. 4455–4460.
[24] E. F. Pettersen, T. D. Goddard, C. C. Huang, G. S. Couch, D. M. Greenblatt, E. C. Meng, and T. E. Ferrin, "UCSF chimera a visualization system for exploratory research and analysis," *J Comp Chem*, vol. 25, no. 13, pp. 1605–1612, 2004.
[25] M. Tirion, "Large amplitude elastic motions in proteins from a single-parameter, atomic analysis," *Physical Review Letters*, vol. 77, no. 9, pp. 1905–1908, August 1996.
[26] M. M. Tirion and D. ben-Avraham, "Normal mode analysis of g-actin," *Journal of molecular biology*, vol. 230, no. 1, pp. 186–195, 1993.
[27] B. Vandereycken, "Riemannian and multilevel optimization for rank-constrained matrix problems," Ph.D. dissertation, Department of Computer Science, KU Leuven, 2010.
[28] N. J. Wildberger, *Divine Proportions: Rational Trigonometry to Universal Geometry*. Wild Egg, 2005.