ON LEAST SQUARES EUCLIDEAN DISTANCE MATRIX APPROXIMATION AND COMPLETION

DAVID I. CHU[*], HUNTER C. BROWN[†], AND MOODY T. CHU[‡]

Abstract. The Euclidean distance matrix approximation problem as well as the completion problem have received a lot of attention in recent years because of their many important applications. In contrast to the many interesting but often sophisticated algorithms proposed in the literature, this paper offers a relatively straightforward procedure that can tackle both problems by the same framework. The location vectors whose relative distance squares form the entries of the Euclidean distance matrix are used directly as parameters in the least squares formulation. It is shown how both gradient and Hessian of the objective function can be calculated explicitly in block form and, thus, can be assembled block by block according to designated locations. Highly effective conventional optimization techniques can then utilized to obtain the least squares solution. The approach can be applied to a variety of problems arising in biological or engineering applications, including molecular structure analysis, protein folding problem, remote exploration and sensing, antenna array processing, and utility allocation problem.

Key words. distance geometry, least squares approximation, matrix completion, molecular structure, protein folding, utility allocation, conformational analysis.

1. Introduction. The endeavor that, "*given the distance and chirality constrains which define (our state of knowledge of) a mobile molecule, find one or more conformations which satisfy them, or else prove that no such conformations exists*," has been referred to as the *Fundamental Problem in Distance Geometry* in [7]. The notion of distance geometry, initiated by Menger and Schoenbeng in the 1930's, has been an area of active research because of its many important applications, including molecular conformation problems in chemistry [7], multidimensional scaling in behavioral sciences [10, 18], and multivariate analysis in statistics [20]. The article [13] is an excellent reference that expounds the framework of Euclidean distance geometry in general. More extensive discussion on the background and applications of distance geometry can be found in [7].

One of the most basic requisitions in the study of distance geometry is the information of inter-point relationships. Given n particles at locations $\mathbf{p}_1, \ldots, \mathbf{p}_n$ in the space $\mathbb{R}^m$, the corresponding *Euclidean distance matrix* $Q(\mathbf{p}_1, \ldots, \mathbf{p}_n) = [q_{ij}]$ is the $n \times n$ symmetric and nonnegative matrix whose entry q_{ij} is defined by

$$q_{ij} = \|\mathbf{p}_i - \mathbf{p}_j\|^2, \quad i, j = 1, \ldots, n, \tag{1.1}$$

[*]Department of Biomedical Engineering, Yale University, New Haven, CT 06520-8042. email: david.chu@yale.edu.

[†]Department of Mathematics, North Carolina State University, Raleigh, NC 27695-8205. email: hcbrown2@unity.ncsu.edu

[‡]Department of Mathematics, North Carolina State University, Raleigh, NC 27695-8205. email: chu@math.ncsu.edu. This research was supported in part by the National Science Foundation under grants DMS-9803759 and DMS-0073056.

where $\|\cdot\|$ denotes the Euclidean norm in $\mathbb{R}^m$. In other words, the distance matrix $Q(\mathbf{p}_1, \ldots, \mathbf{p}_n) = [q_{ij}]$ contains an exhaustive record of relative spacing between any two of the n particles in $\mathbb{R}^m$. For most real-world applications, we only need to be concerned about the case $m = 3$. Nonetheless, the theory presented hereinafter can be extended to a general dimension m.

Be aware that q_{ij} in (1.1) is defined to be the distance square. The corresponding problem to the following discussion but *without* the squares has a very different nature and an even longer history. It perhaps dates back to the 17th century when Fermat studied the problem of finding the shortest network interconnection points on a plane. Nowadays the problem without the squares is better known as the Euclidean facilities location problem and the Steiner minimal tree problem both of which will not be discussed in this paper.

Euclidean distance matrices enjoys many interesting properties. The thesis [8], for example, summarizes several equivalent forms of Euclidean distance matrices. The Euclidean distance matrices are closely connected to positive semidefinite matrices [16, 17]. Several other theoretical results can be found in articles such as [2, 9, 11, 14]. Among these, we find the rank structure of Euclidean distance matrices most peculiar. That is, regardless of the size n of the matrix, an Euclidean distance matrix is always rank deficient. The following result is well established in the literature.

THEOREM 1.1. *For any $n \geq m+2$, the rank of Q is no greater than $m+2$ and is generically $m+2$.*

The Euclidean distance matrix for a geometric entity embedded in $\mathbb{R}^3$ with at least 5 points, for instance, is generically of rank 5 regardless of the number n of points involved. Once all the inter-particle distances are in hand, it is a simple task to construct the (3-dimensional) geometric structure. The fact that a distance matrix is always rank deficient strongly suggests that many entries in the matrix provide redundant information. Indeed, not all the inter-particle distances are needed for the construction. To characterize when and what partial information of inter-particle distances will be sufficient for the determination of the entire geometric structure is an interesting and very challenging open question by itself. Such a task is referred to a *completion problem* in the literature.

Completion problems for special cases such as n-vertex polyhedrons are not difficult to answer. Theoretical results for more general configurations are known only in terms of abstract graph theory [2, 15] and is NP-hard. Despite the lack of a satisfactory theory, numerical method for the Euclidean distance matrix completion problem abound [1, 15, 23]. In this paper, we seek to delineate a similar notion to tackle this important completion problem along with a least squares distance matrix approximation formulation. Our contribution in this paper is that, in contrast to existing methods in the literature, we offer a much simpler framework which works for both approximation and completion problems. The procedure can be generalized to arbitrarily higher m. More importantly, our matrix calculus exploits the inherent block structure of the underlying objective function and its derivatives which, in turn, enable us to take advantage of large-scale optimization techniques. We shall be more specific in the sequel.

Before we formulate our problem and describe a numerical procedure for its solution, it might be fitting to ground the study through an interesting application in biochemistry. A protein molecule is a connected sequence of amino acid molecules. It is known that there are only twenty amino acids in nature. If we represent each amino acid by a letter from a twenty-letter alphabet, then a protein is a string of amino acids linked together like a word. Nowadays, most laboratories have the technology to find the ordered sequence of amino acids in a protein. However, merely knowing the long linear chains is not enough. To function properly, the one-dimensional primary amino acid sequence must fold into a particular three-dimensional conformation called its tertiary configuration. This protein then interacts three-dimensionally with other proteins or other groups of molecules called substrates in a manner much like a lock and key arrangement. It is the tertiary structure that mediates how a protein functions. For example, the final folding configuration determines whether a protein acts as an enzyme in chemical reactions or behaves as part of an antibody in the immune system, and so on. In short, how a protein folds determines how the protein works, which ultimately determines how our bodies work. Wrongly folded proteins do not behave normally, and their abnormal function can lead to disease. Understanding how proteins fold into three-dimensional structures given their primary linear sequence thus becomes an incredibly important task.

The literature on the various aspects of protein folding is enormous. A recent survey by Neumaier [21] gives a masterly account of the current state of this problem and contains 360 references, many of which also have extensive bibliographies. In this paper we intend to address the folding problem under one presumptuous scenario. Biologists try to see the three dimensional structure by techniques such as x-ray crystallography or nuclear magnetic resonance (NMR) to build molecular dynamics models [3]. Yet the noise in the sensing devices or the imperfection of the models often results in indefinite or incomplete images. Thus an imperative task is to retrieve a possible folding structure that is nearest to the observed but possibly inconsistent or imperfect configuration.

More specifically, suppose that $F \in \mathbb{R}^{n\times n}$ denotes an observation of (the squares of) the relative spacing among n particles in $\mathbb{R}^3$. In practice, it is very likely that some of the spacing has not been measured accurately. One prominent sign that something in the data is inconsistent would be that the rank of F is greater than 5. We therefore would like to retrieve whatever feasible information out of the matrix F so as to infer a realistic conformation. It is also likely that some of the spacing is unobservable and, consequently, some entries in the matrix F are missing. In this case, we want to complete the matrix F so that it becomes a distance matrix. In either case, the observed matrix F needs to be modified.

One idea of modification is to find an Euclidean distance matrix $Q \in \mathbb{R}^{n\times n}$ such that Q represents the minimal change of F. The criterion used in characterizing the changes affects the way in which the approximation is formulated. In [6], for example, it was suggested that low rank approximation techniques could be applied to find the nearest nonnegative, symmetric matrix of rank 5 to the given F. This approach, however, only partially corrects the problem because a nonnegative and symmetric matrices of rank 5 is not necessarily a distance matrix. The distance matrix has more structure that is yet to be exploited. In [11], efforts have been taken to cast the distance matrices as the intersection of two geometric entities. The Dykstra algorithm that alternates projections between two convex sets is used with the intention of minimizing a quadratic functional over the intersection which contains Euclidean distance matrices. This approach, however, suffers from its slow convergence and possible stagnation. We mention that there is a rich discussion for distance matrices without the squares. Gaussian smoothing followed by limited-memory variable-metric continuation techniques [19] and a transformation into a standard convex programming in conic form followed by a polynomial time interior point algorithm [25] are just two of many possible avenues to circumvent the non-smoothness. Similar approaches by using primal-dual interior point techniques that solves an equivalent semidefinite programming problem have been proposed in [1, 17, 18, 23]. In this paper, we propose to tackle the approximation problem by a least squares formulation where the Euclidean distance matrices are parameterized directly in terms the location vectors $\mathbf{p}_1, \ldots, \mathbf{p}_n$. One immediate advantage of this representation is that the resulting approximation of the imperfect F is guaranteed to be a distance matrix.

While we appreciate the efficiency and elegance of the many existing methods, especially the semidefinite programming techniques, usually there is a great deal of specific projections or transformations that must take place before the techniques can be applied. In contrast, our approach is much more straightforward. We work directly with the location vectors. Our idea is not new. Various least squares formulations have been discussed in [7, 12, 19, 23]. Our main contribution is to point out a highly organized way of explicitly computing the gradient and the Hessian of the objective function in block forms. Efficient classical optimization techniques therefore can be applied. Furthermore, the derivative information can be characterized block by block in accordance with the location vectors. This capacity gives us the additional flexibility to "assemble" the derivative information and, hence, facilitates the handling of the more complication situation where some position vectors are either fixed or missing. Our framework works for both approximation and completion problems.

Finally, we point that some recent work in [4, 5, 24] which actually suggests transforming semidefinite programming problems to standard convex programming problems via Choleskey-type factorizations. These approaches allows the possibility of exploiting rank structures. In some sense, this notion is close to ours in that the Euclidean distance matrix approximation problem is written as a smooth nonlinear programming problems as opposed to a semidefinite programming problem.

2. Basic Formulation. For generality, let entries in $F \in \mathbb{R}^{n\times n}$ represent henceforth the squares of observed distances among n particles in $\mathbb{R}^m$. As we begin to lay down the groundwork for discussion, we shall assume initially that *all* entries of F except the diagonal are to be approximated. Later we shall modify the basic formulation to tackle two special cases: the approximation problem where some location vectors of the n particles are known and fixed and the completion problem where a partial information of spacing among the n particles, but not necessarily from known location vectors, is known and fixed. Each problem requires different modifications and will be handled separately.

Our basic idea of the Euclidean distance matrix approximation of F is to minimize the objective function

$$f(\mathbf{p}_1, \ldots \mathbf{p}_n) = \frac{1}{2}\|F - [\langle \mathbf{p}_i - \mathbf{p}_j, \mathbf{p}_i - \mathbf{p}_j\rangle]\|_F^2 \tag{2.1}$$

where $\langle\cdot,\cdot\rangle$ denotes the usual Euclidean inner product and $\|\cdot\|_F$ denotes the Frobenius matrix norm. Thus far, it is obvious that the solution is not unique because any rigid body motion of a given conformation will produce the same relative spacing. Later we shall show how some of the locations must be fixed as reference points, so not every vector in the entire list of $\mathbf{p}_1, \ldots \mathbf{p}_n$ is to be determined numerically. Be noted that the function $f(\mathbf{p}_1, \ldots \mathbf{p}_n)$ is *not* convex. Unless global optimization technique is applied, generally we will find local solutions only.

We should point out immediately that in a demonstration of its large-scaled unconstrained nonlinear minimization capability, MATLAB features an M-file called MOLECULAR in the Optimization Toolbox that solves a two-dimensional molecule conformation problem [22]. However, that code appears to be loosely organized when compared to our block form. Our formulation easily extends that in MOLECULAR to any general dimensions. More importantly, we offer in the following a highly organized way of computing the derivative information that not only improves and generalizes the coding in MOLECULAR, but also gives us the great flexibility in assembling the derivative information location by location.

2.1. Analytic Gradient. The objection function $f(\mathbf{p}_1, \ldots \mathbf{p}_n)$ is differentiable. There are several ways to compute its derivatives. We suggest using matrix calculus to carry out the calculation as it naturally provides the block form. For convenience, denote

$$d_{ij} = f_{ij} - \langle \mathbf{p}_i - \mathbf{p}_j, \mathbf{p}_i - \mathbf{p}_j\rangle \tag{2.2}$$

for $i, j = 1, \ldots n$. We shall consider f as a function defined on the space $\mathbb{R}^m \times \ldots \times \mathbb{R}^m$ equipped with the product topology. In such a normed topological space, the Fréchet derivative of f at the point $(\mathbf{p}_1, \ldots \mathbf{p}_n)$ acting on an arbitrary *n-fold* vector $(\mathbf{z}_1, \ldots, \mathbf{z}_n) \in \mathbb{R}^m \times \ldots \times \mathbb{R}^m$ can be represented as the multi-lineal functional

$$f^{'}(\mathbf{p}_1, \ldots \mathbf{p}_n).(\mathbf{z}_1, \ldots, \mathbf{z}_n) = \sum_{k=1}^{n} \frac{\partial f}{\partial \mathbf{p_k}}(\mathbf{p}_1, \ldots \mathbf{p}_n).\mathbf{z}_k \tag{2.3}$$

where . denotes the action of the differential operator and the *partial gradient* $\frac{\partial f}{\partial \mathbf{p_k}}$ represents a "section" of the true gradient. For each k, the partial gradient can be expressed in the following theorem.

THEOREM 2.1. *For $k = 1, \ldots, n$, the partial gradient $\frac{\partial f}{\partial \mathbf{p_k}}$ is an m-dimensional vector given by*

$$\frac{\partial f}{\partial \mathbf{p}_k} = -4 \sum_{\substack{j=1 \\ j \neq k}}^{n} (\mathbf{p}_k - \mathbf{p}_j) d_{kj}. \tag{2.4}$$

Proof. By the Riesz representation theorem, the action of the partial gradient can be considered as the inner product

$$\frac{\partial f}{\partial \mathbf{p_k}}(\mathbf{p}_1, \ldots \mathbf{p}_n).\mathbf{z}_k = \langle \frac{\partial f}{\partial \mathbf{p_k}}(\mathbf{p}_1, \ldots \mathbf{p}_n), \mathbf{z}_k \rangle.$$

Observe that

$$\langle \frac{\partial f}{\partial \mathbf{p_k}}, \mathbf{z}_k \rangle = \langle \frac{\partial}{\partial \mathbf{p}_k} \left(F - [\langle \mathbf{p}_i - \mathbf{p}_j, \mathbf{p}_i - \mathbf{p}_j \rangle]\right).\mathbf{z}_k, F - [\langle \mathbf{p}_i - \mathbf{p}_j, \mathbf{p}_i - \mathbf{p}_j \rangle] \rangle,$$

whereas the first action on the right-hand side of the above expression will affect only the k-th row and column. More precisely,

$$\frac{\partial}{\partial \mathbf{p}_k} \left(F - [\langle \mathbf{p}_i - \mathbf{p}_j, \mathbf{p}_i - \mathbf{p}_j \rangle]\right).\mathbf{z}_k =$$

$$-2 \begin{pmatrix} 0 & \ldots & \langle \mathbf{z}_k, \mathbf{p}_k - \mathbf{p}_1 \rangle & & 0 \\ \vdots & & \vdots & & \vdots \\ \langle \mathbf{z}_k, \mathbf{p}_k - \mathbf{p}_1 \rangle & \ldots & 0 & \ldots & \langle \mathbf{z}_k, \mathbf{p}_k - \mathbf{p}_n \rangle \\ & & \vdots & & \\ 0 & & \langle \mathbf{z}_k, \mathbf{p}_k - \mathbf{p}_n \rangle & & 0 \end{pmatrix}.$$

It follows that

$$\frac{\partial f}{\partial \mathbf{p_k}}(\mathbf{p}_1, \ldots \mathbf{p}_n).\mathbf{z}_k = \langle \mathbf{z}_k, -4 \sum_{\substack{j=1 \\ j \neq k}}^{n} (\mathbf{p}_k - \mathbf{p}_j) d_{kj} \rangle$$

and the assertion is proved. $\square$

Using the product topology again, the gradient ∇f of the objective function f can easily be interpreted as the n-fold vector

$$\nabla f(\mathbf{p}_1, \ldots \mathbf{p}_n) = \left(\frac{\partial f}{\partial \mathbf{p}_1}, \ldots, \frac{\partial f}{\partial \mathbf{p}_n}\right) \in \mathbb{R}^m \times \ldots \times \mathbb{R}^m. \tag{2.5}$$

With the availability of the gradient, any numerical method utilizing the gradient information can now be employed to solve the approximation problem. For instance, the flow $(\mathbf{p}_1(t), \ldots \mathbf{p}_n(t))$ defined by the differential equation

$$\frac{d\mathbf{p}_k}{dt} = 4\sum_{\substack{j=1\\ j\neq k}}^{n} (\mathbf{p}_k - \mathbf{p}_j)(f_{kj} - \langle \mathbf{p}_k - \mathbf{p}_j, \mathbf{p}_k - \mathbf{p}_j\rangle), \quad k = 1, \ldots, n \tag{2.6}$$

moves in the steepest descent direction to reduce the values for the objective function f. As a descent flow bounded in a neighborhood of F, its limit point must exist and gives to a (local) least squares approximation of the given F.

2.2. Analytic Hessian. The critical point of $f(\mathbf{p}_1, \ldots \mathbf{p}_n)$ occurs at wherever the gradient vanishes. It will be more effective for finding such a critical point if the second-derivative information of f is available. By interpreting the gradient of f as an n-fold vector of $m \times 1$ blocks, we can formulate the Hessian of f as an $n \times n$ block matrix with $m \times m$ blocks. Let

$$g : \mathbb{R}^m \times \ldots \times \mathbb{R}^m \to \mathbb{R}^m \times \ldots \times \mathbb{R}^m$$

denote the nonlinear function whose k-th *component* $g_k : \mathbb{R}^m \times \ldots \times \mathbb{R}^m \to \mathbb{R}^m$ is precisely

$$g_k(\mathbf{p}_1, \ldots \mathbf{p}_n) = \frac{\partial f}{\partial \mathbf{p}_k}(\mathbf{p}_1, \ldots \mathbf{p}_n). \tag{2.7}$$

Then the Jacobian matrix of g_k constitutes precisely the k-th row block of the Hessian of f. Indeed, we can perform the block to block calculation as follows.

THEOREM 2.2. *For a fixed k and $i = 1, \ldots, n$, the (k,i)-block partial Jacobian $\frac{\partial g_k}{\partial \mathbf{p}_i}$ is given by the $m \times m$ matrix*

$$\frac{\partial g_k}{\partial \mathbf{p}_i} = \begin{cases} \sum_{j=1, j\neq k}^{n} \left[-4d_{kj}I_m + 8(\mathbf{p}_k - \mathbf{p}_j)(\mathbf{p}_k - \mathbf{p}_j)^T\right], & \text{if } i = k; \\ 4d_{ki}I_m - 8(\mathbf{p}_k - \mathbf{p}_i)(\mathbf{p}_k - \mathbf{p}_i)^T, & \text{if } i \neq k, \end{cases} \tag{2.8}$$

where I_m is the $m \times m$ identity matrix.

Proof. Under the product topology, the Fréchet derivative $g_k^{'}(\mathbf{p}_1, \ldots \mathbf{p}_n)$ acting at an arbitrary n-fold vector $(\mathbf{w}_1, \ldots \mathbf{w}_n) \in \mathbb{R}^m \times \ldots \times \mathbb{R}^m$ can be considered as the block to block operation

$$g_k^{'}(\mathbf{p}_1, \ldots \mathbf{p}_n).(\mathbf{w}_1, \ldots \mathbf{w}_n) = \sum_{i=1}^{n} \frac{\partial g_k}{\partial \mathbf{p}_i}.\mathbf{w}_i.$$

Based on (2.4), if $i = k$, then

$$\frac{\partial g_k}{\partial \mathbf{p}_k}.\mathbf{w}_k = -4\sum_{\substack{j=1\\ j\neq k}}^{n} \left[\mathbf{w}_k d_{kj} - 2(\mathbf{p}_k - \mathbf{p}_j)\langle \mathbf{w}_k, \mathbf{p}_k - \mathbf{p}_j\rangle\right]$$
$$= \sum_{\substack{j=1\\ j\neq k}}^{n} \left[-4d_{kj}I_m + 8(\mathbf{p}_k - \mathbf{p}_j)(\mathbf{p}_k - \mathbf{p}_j)^T\right] \mathbf{w}_k.$$

If $i \neq k$, then

$$\frac{\partial g_k}{\partial \mathbf{p}_i}.\mathbf{w}_i = -4\left[-\mathbf{w}_i d_{ki} - 2(\mathbf{p}_k - \mathbf{p}_i)\langle -\mathbf{w}_i, \mathbf{p}_k - \mathbf{p}_i\rangle\right]$$
$$= \left[4d_{ki}I_m - 8(\mathbf{p}_k - \mathbf{p}_i)(\mathbf{p}_k - \mathbf{p}_i)^T\right] \mathbf{w}_i.$$

The action of the Fréchet derivative of $g_k^{'}$ is precisely the multiplication of the Jacobian matrix with the n-fold vector $[\mathbf{w}_1, \ldots, \mathbf{w}_n]^T$. $\square$

3. Modification. In practice, usually some additional information is available in the geometric conformation. For example, the x-ray crystal structure of each of the twenty amino acids in nature is known. Together with the fact that most of the amino acid sequences of our proteins are also known, it is often the case that, once a certain amino acid is known to present in the protein, a certain block of the matrix F is already predetermined and fixed. The least squares formulation (2.1) should be modified accordingly to reflect this fact. We stress that the derivative information available in block form as described above is particularly convenient for the overall process of assembling the gradient and the Hessian which are essential for almost all efficient optimization technique to take effect. We briefly outline two situations in this section.

3.1. Approximation with Partially Fixed Locations. It is clear that any rotations, translations, or reflections of a given conformation will produce the same relative spacing and hence the same distance matrix. To shun rotation, translation, or reflection so as to exactly locate the positions of these particles from a given distance matrix, $m + 1$ positions of these n particles in the embedding space $\mathbb{R}^m$ must be specified and bound as reference points. Additionally, in the presence of some known amino acids for example, it is entirely possible that some additional location vectors among $\mathbf{p}_1, \ldots, \mathbf{p}_n$ are also known and fixed beforehand.

For convenience, let $\mathbf{q}$ denote the indices of known location vectors. Then entries f_{ij} of F where both $i, j \in \mathbf{q}$ correspond to the spacing among these known location vectors. These entries of F should be exact and kept constant. These known and fixed location vectors should not be altered. As such, derivatives at these points should be zero. In the process of assembling the derivative information for optimization software, the block form of derivatives facilitates the task of keeping these known position vectors invariant by simply nullifying any derivative information at the corresponding blocks.

On the other hand, any spacing information between particles that has been missed in the original observation can simply be substituted by zero at the corresponding entry in F. Our optimization scheme should automatically fill in the correct spacing information at the end of its iteration.

4. Completion with Partially Fixed F. In a completion problem, the matrix F represents a partially specified distance matrix. The task is to find the unspecified elements of F so that the completed F is a full distance matrix.

One must distinguish the completion problem from the approximation problem carefully. In the approximation problem, every entry f_{ij} of F except those corresponding to the known location vectors in $\mathbf{q}$ are allowed to be approximated. The completion problem differs from the approximation problem in that the specified entries in F do not necessarily correspond to any known location vectors while still being required to remain the same throughout the whole matrix completion process. In both cases, one fundamental issue must be addressed first. That is, the specified entries in F must be consistent by themselves to begin with. If any of these entries also carries error, then we are facing a more complicated problem which requires both completion and approximation. To determine whether the specified entries in F are consistent so that F indeed can be completed as a distance matrix is another challenging problem which has not completely understood [2, 15, 23].

Let Δ denote the index set of those specified entries of F, that is, let

$$\Delta := \{(i,j) \in \mathbb{Z} \times \mathbb{Z} \,|\, d_{ij} = 0\}.$$

A reasonable least squares formulation of the completion problem would be to minimize the same objective function $f(\mathbf{p}_1, \ldots \mathbf{p}_n)$ as in (2.1), where $d_{ij} = 0$ whenever $(i,j) \in \Delta$, subject to the additional equality constraints

$$\langle \mathbf{p}_i - \mathbf{p}_j, \mathbf{p}_i - \mathbf{p}_j \rangle = f_{ij} \quad \text{for all } (i,j) \in \Delta. \tag{4.1}$$

In other words, the completion problem is now cast as an equality constrained optimization problem.

5. Numerical Examples. The formulas of gradient (2.4) and Hessian (2.8) are valid for arbitrary dimension m. The discussion stipulated in this paper can be applied to the most general formulation (2.1). In order that we can visualize the effect of the least squares approximation, we shall demonstrate some numerical examples by limiting the experiments to the cases where $m = 2$ or 3.

Example 1. The parametric equation

$$\begin{cases} x = -10\cos(t) - 2\cos(5t) + 15\sin(2t); \\ y = -15\cos(2t) + 10\sin(t) - 2\sin(5t); \qquad 0 \le t \le 2\pi \\ z = 10\cos(3t); \end{cases}$$

defines a knot in $\mathbb{R}^3$. See the upper left of Figure 5.1 for its drawing in $\mathbb{R}^3$ and the lower right for its projection onto the xy-plane. To simulate the distance geometry, we represent the knot by finitely many, say n, discrete points marked by "$*$" in Figure 5.1. These n discrete points define a true $n \times n$ distance matrix $Q = [q_{ij}]$. Let $\mathbf{q}$ denote the indices of known location vectors. To avoid rotation, translation, or reflection, the size of $\mathbf{q}$ should be at least four.

We perturb the square roots of elements of Q symmetrically at those locations not belonging to $\mathbf{q}$ by a random noise with mean zero and variance σ^2, that is, let $F = [f_{ij}]$, where

$$f_{ij} = f_{ji} = \begin{cases} 0, & \text{if } i = j, \\ q_{ij}, & \text{if both } i \text{ and } j \text{ are in } \mathbf{q}, \\ \left(\sqrt{q_{ij}} + \sigma * randn(1)\right)^2, & \text{if either } i \text{ or } j \text{ is not in } \mathbf{q}, \end{cases}$$

and $randn$ denotes the normally distributed random variable with mean zero and variance one. Then F represents (the squares of) the observed and imperfect spacing among the n location vectors of the knot. The standard deviation σ provides a measurement of how far F is away from a distance matrix. In this example, we take $\sigma = 2$ which implies that the original distance matrix Q is buried in a significant amount of random noise. We want to find the least squares distance matrix approximation of F and to plot, with the $\mathbf{q}$ location vectors fixed, the resulting configuration.

For illustration purpose, we employ an existing routine FMINUNC in the Optimization Toolbox, Version 2.2, of MATLAB to solve the least squares problem (2.1), while keeping those location vectors identified by $\mathbf{q}$ fixed throughout the iteration. Obviously, many other software packages can be utilized as well. As the starting value, we perturb each entry of the true location vectors by adding a random variable with uniform distribution over [-20,20]. This would amount to a fairly deviate initial guess. See the upper right and lower right drawings in Figure 5.1 for comparison.

Depicted in the lower left of Figure 5.1 by "o" is the numerical result of the case where $n = 51$ and $\mathbf{q} = 1 : 5 : 51$. (So totally there are 120 variables to be determined.) Its projection onto the xy-plane is compared with that of the original knot in the lower right of Figure 5.1. In this particular example, the value of the objective function is reduced from an original order of 10^9 to 10^6, indicating that F is far from being a distance matrix. Nevertheless, the drawing in Figure 5.1 shows a remarkable likeness between the recovered and the original configurations.

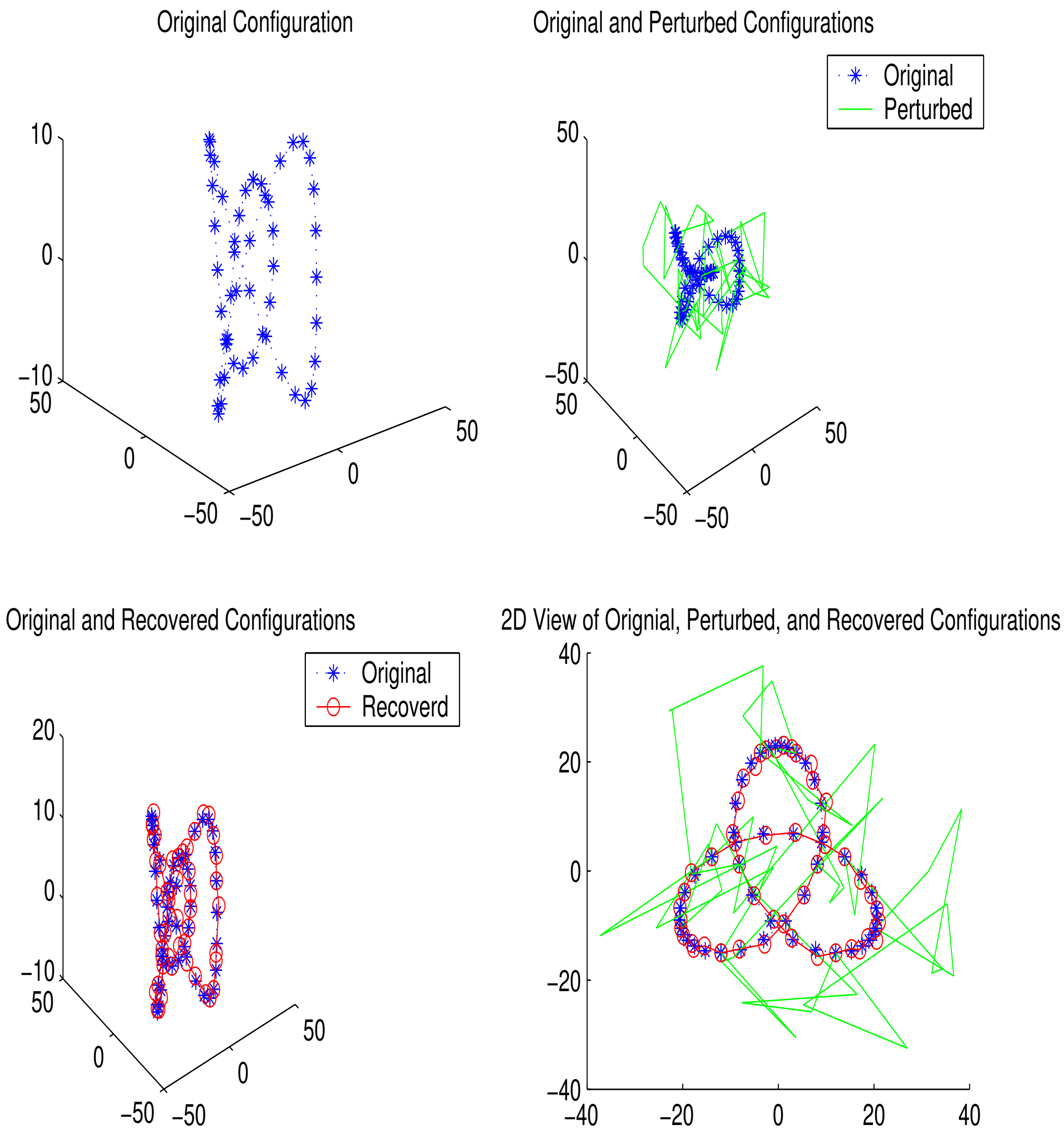

FIG. 5.1. *Approximation of a knot.*

Example 2. Consider the helix defined by

$$\begin{cases} x = 4\cos(3t); \\ y = 4\sin(3t); \quad 0 \leq t \leq 2\pi; \\ z = 2t. \end{cases}$$

We repeat the above experiment except using $n = 101$ and $\mathbf{q} = 1:5:101$. So totally there are 240 variables to be determined. The original and the recovered

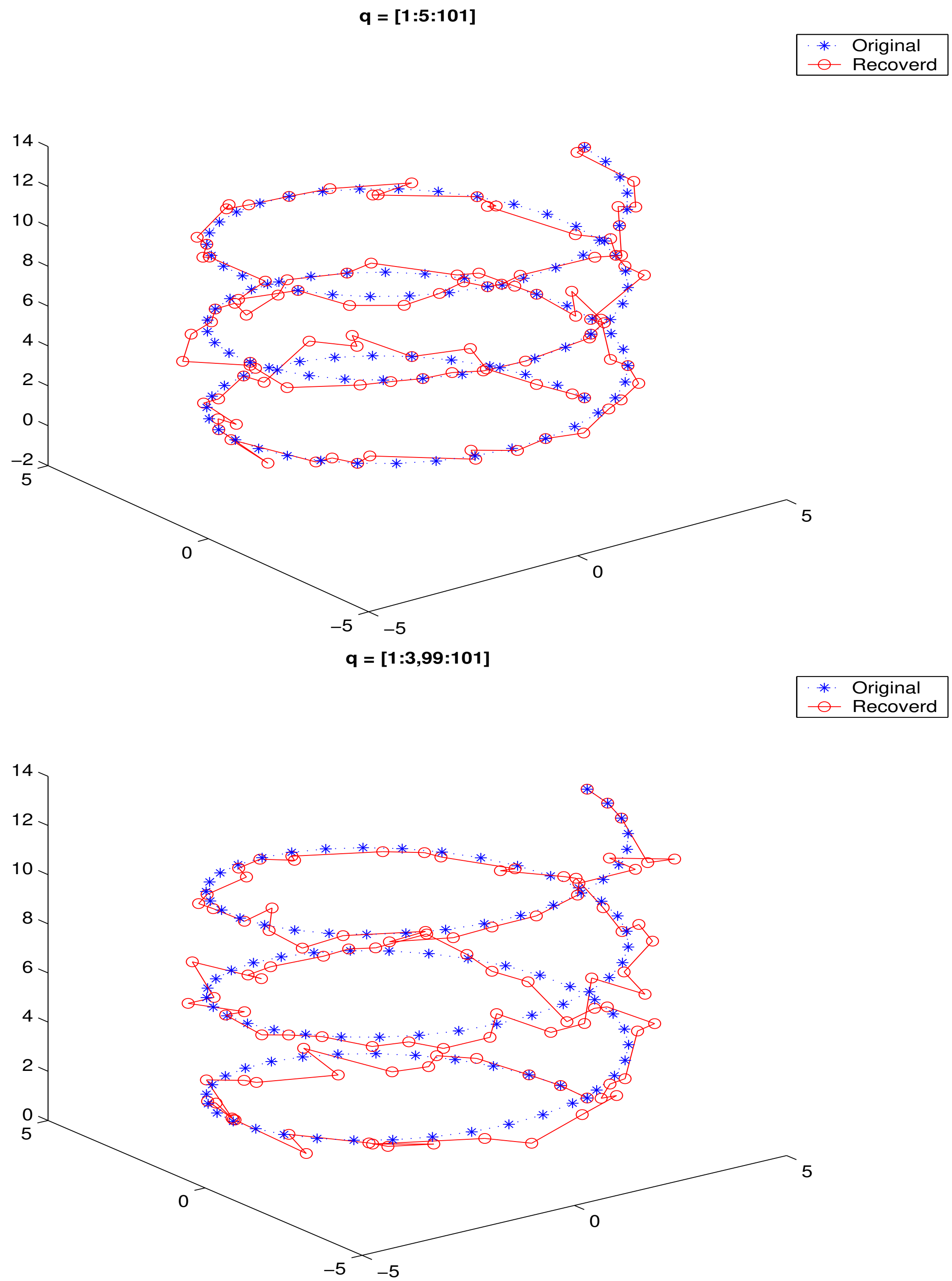

Fig. 5.2. *Approximation of a helix.*

configurations are plotted in the upper drawing in Figure 5.2. Although the numerical solution is not as smooth, it captures the essential feature of a helix from afar with a fairly deviate initial guess and a fraudulent F.

To illustrate that the folding of the numerical solution into a helix is *not* due to the fact that fixed locations vectors identified by the current $\mathbf{q}$, i.e., $1:5:101$, have already roughly outlined a crude helix, we rerun the experiment with the index set $\mathbf{q} = [1:3, 99:101]$, i.e., with only the first and the last three location vectors are fixed while all the intermediate locations are subject to change. In this case, there are a total 291 variables to be determined. The numerical result is plotted in the lower drawing in Figure 5.2. A close examination shows that "$*$" and "o" coincide only at these points and that the computed "helix" deviates further from the true helix than the previous case. However, the feature of spirality is evident.

Example 3. The 2-dimensional twist defined by

$$\begin{cases} x = t + 3\sin(2t); \\ y = t + 2\sin(3t); \quad -2\pi \leq t \leq 2\pi; \end{cases}$$

has many critical turns which are close to each other. A more careful observation of the spacing among the location vectors is necessary. To affect this scenario, we assume a smaller standard deviation $\sigma = 1$ of noise in the simulation. Depicted in Figure 5.3 is the numerical result with $n = 61$ and only the first and last three location vectors are fixed. We find that even with moderately wild initial guesses (not shown but refer to the setting explained in Example 1), the numerical result is able to pick up the folds of the original curve. Of course, the smaller the σ is and the more known positions are prescribed, the easier and more accurately the least squares approximation can be accomplished.

Example 4. Consider the 6×6 partially specified matrix

$$F = \begin{bmatrix} 0 & 3 & 4 & 3 & 4 & 3 \\ 3 & 0 & 1 & x_1 & 5 & x_3 \\ 4 & 1 & 0 & 5 & x_2 & 5 \\ 3 & x_1 & 5 & 0 & 1 & x_4 \\ 4 & 5 & x_2 & 1 & 0 & 5 \\ 3 & x_3 & 5 & x_4 & 5 & 0 \end{bmatrix}$$

where x_i, $i = 1, \ldots, 4$, denotes the unspecified entries. To complete this matrix means to find six location vectors in $\mathbb{R}^m$ whose relative spacing agrees with those already specified in F. It is difficult to tell at a glance of this matrix what the dimension m of the embedding space should be. It is suggested in [23] that no conformation could give rise to this matrix F if $m = 2$. Note that the first 3×3 principal submatrix is completely known, suggesting that three location vectors could have been self-determined. Using $\mathbf{p}_1 = (0, 0)$, $\mathbf{p}_2 = (\sqrt{3}, 0)$, and $\mathbf{p}_3 = (\sqrt{3}, 1)$ as reference

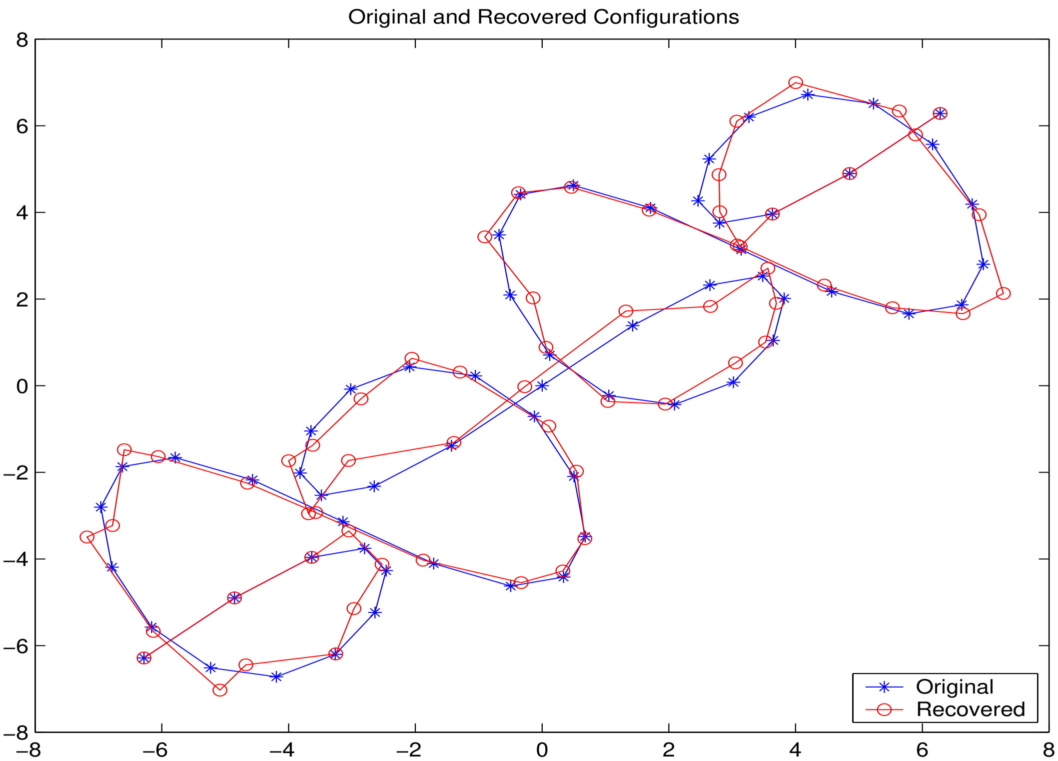

FIG. 5.3. *Approximation of a twist.*

points in $\mathbb{R}^2$, there are eight equality constraints in the form of (4.1) for the remaining three location vectors. If the embedding space is $\mathbb{R}^2$, then the unknown location vectors $\mathbf{p}_j$, $j = 4, 5, 6$, constitute only six unknowns. There are more constraints than unknowns. It is perhaps for this reason that the overdetermined system has no feasible solution at all in $\mathbb{R}^2$.

Suppose we embed $\mathbf{p}_i$, $i = 1, 2, 3$, in $\mathbb{R}^3$ by adding zeros to their third components and seek to complete the matrix F by location vectors in $\mathbb{R}^3$. We employ an existing routine FMINCON in the Optimization Toolbox of MATLAB by supplying the equality constraints (4.1). Each row in Table 5.1 represents (up to the fifth significant digits) a completed distance matrix of F obtained by a different starting value, a clear indication that the completion problem has multiple solutions. It is interesting to note that vectors in Table 5.2 denote a set of location vectors that complete the same matrix F in $\mathbb{R}^4$.

6. Conclusion. In contrast to the many algorithms developed specifically for the distance matrix approximation problem, we cast the problem under a least squares approximation in terms of the location vectors directly and propose using conventional large-scale optimization techniques instead. We manage the resulting

x_1	x_2	x_3	x_4
6.6883	3.9512	2.0187	7.3255
1.7434	9.1772	2.2007	2.2006
2.2800	9.4157	2.3913	4.7487
2.7971	5.7203	7.2315	7.2315
2.2723	9.4208	2.3964	4.7398

Table 5.1
Examples of entries for completed distance matrix.

$\mathbf{p}_1$	$\mathbf{p}_2$	$\mathbf{p}_3$	$\mathbf{p}_4$	$\mathbf{p}_5$	$\mathbf{p}_6$
0	1.7321	1.7321	0.9014	0.5774	1.1359
0	0	1.0000	-0.5613	-0.4223	-0.9675
0	0	0	1.3335	1.7980	-0.2711
0	0	0	-0.3067	0.5057	0.8367

Table 5.2
Example of location vectors in $\mathbb{R}^4$ for completed distance matrix.

complexity by organizing the gradient and Hessian information in block forms. The matrix calculus makes it particularly easy to assemble the derivatives for existing software packages when some locations vectors are known and fixed. The theory is simple, yet numerical experiments seem to suggest that this straightforward approach is just as efficient and robust in the reconstruction.

REFERENCES

[1] A. Alfakih, A. Khandani, and H. Wolkowicz, Solving Euclidean distance matrix completion problems via semidefinite programming, Comput. Optim. Appl., 12(1999), pp.13-30.

[2] M. Bakonyi and C. R. Johnson, The Euclidean distance matrix completion problem, SIAM J. Matrix Anal. Appl., 16(1995), 645-654.

[3] A. T. Brünger and M. Nilges, Computational challenges for macromolecular structure determination by x-ray crystallography and solution NMR-spectroscopy, Q. Rev. Biophysics, 26(1993), pp.49-125.

[4] S. Burer, R.D.C. Monteiro, and Y. Zhang, Solving semidefinite programs via nonlinear programming, Part 1: Transformations and derivative, Technical report, Department of Computational and Applied Mathematics, Rice University, 1999.

[5] S. Burer and R.D.C. Monteiro, A projected gradient algorithm for solving the maxcut SDP relaxation, Optim. Methods Softw., 15(2001), 175-200.

[6] M. T. Chu, R. E. Funderlic, and R. J. Plemmons, Structured low rank approximation, Linear Alg. Appl., 366(2003), 157-172.

[7] G. M. Crippen and T. F. Havel, Distance Geometry and Molecular Confirmation, John Wiley & Sons, 1988.

[8] J. Dattorro, Euclidean Distance Matrix, http://www.stanford.edu/~dattorro/EDM.pdf.

[9] R. W. Farebrother, Three theorems with applications to Euclidean distance matrices, Linear

Alg. Appl., 95(1987), 11-16.
[10] J. de Leeuwand and W. Heiser, Theory of multidimensional scaling, in Handbook of Statistics, Vol. 2, P. R. Krishnaiah and L. N. Kanal, ed., North Holland, Amsterdam, 1982, pp.285-316.
[11] W. Glunt, T. L. Hayden, S. Hong, and J. Wells, An alternating projection algorithm for computing the nearest Euclidean distance matrix, SIAM J. Matrix Anal. Appl., 11(1990), pp.589-600.
[12] W. Glunt, T. L. Hayden, and M Raydan, Molecular conformations from distance matrices, J. Comput. Chemistry, 14(1993), 114-120.
[13] J. C. Gower, Euclidean distance geometry, Math. Scientist, 7(1982), pp.1-14.
[14] J. C. Gower, Properties of Euclidean and non-Euclidean distance matrices, Linear Alg. Appl. 67(1985), 81-97.
[15] H. X. Huang, Z. A. Liang, and P. M Pardalos, Some properties for the Euclidean distance matrix and positive semidefinite matrix completion problems, J. Global Optim., 25(2003), 3-21.
[16] C. R. Johnson and P. Tarazaga, Connections between the real positive semidefinite and distance matrix completion problems, Linear Alg. Appl., 223/224(1995), 375-391.
[17] M. Laurent, A connection between positive semidefinite and Euclidean distance matrix completion problem, Linear Alg. Appl., 273(1998), 9-22.
[18] R. Mathar, The best Euclidian fit to a given distance matrix in prescribed dimensions, Linear Alg. Appl., 67(1985), pp.1-6.
[19] J. J. Moré and Z. Wu, Distance geometry optimization for protein structures, Preprint MCS-P628-1296, Argonne National Laboratory, 1996.
[20] J. Meulman, A Distance Approach to Nonlinear Multivariate Analysis, DSWO Press, Leiden, Netherlands, 1986.
[21] A. Neumaier, Molecular modeling of proteins and mathematical prediction of protein structure, SIAM Rev., 39(1997), pp.407-460.
[22] J. Pena, molecule.m, rev. 1.10, in Optimization Toolbox, MATLAB 6.5, The MathWorks, Inc., 2002.
[23] M. W. Trosset, Distance matrix completion by numerical optimizaion, TR95-31, CAAM, Rice University, Houston, TX, 1997.
[24] R. J. Vanderbei and H. Y. Benson, On formulating semidefinte programming problems as smooth convex nonlinear optimization problems, Technical report, Department of Statistics and Operation Research, Princeton University, 1999.
[25] G. Xue and Y. Ye, An efficient algorithm for minimizing a sum of Euclidean norm with applications, SIAM J. Optim., 7(1997), 1017-1036.