Bulletin of Mathematical Biology (2009) 71: 1914–1933
DOI 10.1007/s11538-009-9431-9

ORIGINAL ARTICLE

A Geometric Buildup Algorithm for the Solution of the Distance Geometry Problem Using Least-Squares Approximation

Atilla Sit[a,*], Zhijun Wu[a], Yaxiang Yuan[b]

[a] *Department of Mathematics, Program on Bioinformatics and Computational Biology, Iowa State University, Ames, IA, USA*
[b] *Laboratory of Scientific and Engineering Computing, Chinese Academy of Science, Beijing, China*

Received: 11 March 2008 / Accepted: 7 May 2009 / Published online: 17 June 2009

Abstract We propose a new geometric buildup algorithm for the solution of the distance geometry problem in protein modeling, which can prevent the accumulation of the rounding errors in the buildup calculations successfully and also tolerate small errors in given distances. In this algorithm, we use all instead of a subset of available distances for the determination of each unknown atom and obtain the position of the atom by using a least-squares approximation instead of an exact solution to the system of distance equations. We show that the least-squares approximation can be obtained by using a special singular value decomposition method, which not only tolerates and minimizes small distance errors, but also prevents the rounding errors from propagation effectively, especially when the distance data is sparse. We describe the least-squares formulations and their solution methods, and present the test results from applying the new algorithm for the determination of a set of protein structures with varying degrees of availability and accuracy of the distances. We show that the new development of the algorithm increases the modeling ability, and improves stability and robustness of the geometric buildup approach significantly from both theoretical and practical points of view.

Keywords Biomolecular modeling · Protein structure determination · Distance geometry · Linear and nonlinear systems of equations · Linear and nonlinear optimization

1. Introduction

A well-known problem in protein modeling is the determination of the structure of a protein with a given set of interatomic or interresidue distances obtained from either physical

*Corresponding author.
E-mail address: atilla@iastate.edu (Atilla Sit).
Work supported by the NIH/NIGMS grant R01GM081680, and the NSF grant of China.

experiments or theoretical estimates. A more general and abstract form of the problem is known as the distance geometry problem in mathematics (Blumenthal, 1953), the graph embedding problem in computer science (Saxe, 1979), and the multidimensional scaling problem in statistics (Torgerson, 1958). In general, the problem can be stated as to find the coordinates for a set of points in some topological space given the distances for certain pairs of points. Therefore, in addition to protein modeling where everything is discussed only in three-dimensional Euclidean space, the problem has applications in many other scientific and engineering fields as well, such as sensor network localization (Biswas et al., 2006), image recognition (Klock and Buhmann, 1997), and protein classification (Hou et al., 2003), to name a few. In any case, the problem may or may not have a solution in a given topological space, and even if it does have a solution, the solution may not be easy to find, depending on the given distances.

Let n be the number of atoms in a given protein and $x_1, \ldots, x_n$ be the coordinate vectors for the atoms $1, \ldots, n$, where $x_i = (x_{i,1}, x_{i,2}, x_{i,3})^T$ and $x_{i,1}$, $x_{i,2}$, and $x_{i,3}$ are the first, second, and third coordinates of atom i. Let $d_{i,j}$ be the distance between atoms i and j, $d_{i,j} = \|x_i - x_j\|$, where $\|\cdot\|$ is the Euclidean norm. Then the distance geometry problem for a given set of distances $\{d_{i,j} : (i, j) \text{ in } S\}$ is to find the coordinates $x_1, \ldots, x_n$ for the atoms $1, \ldots, n$ so that the distances between atoms i and j are equal to the given distances $d_{i,j}$, i.e., $\|x_i - x_j\| = d_{i,j}$ where (i, j) is in S. In practice, the distances may have errors and, therefore, a more general yet practical form of the problem would be to find the coordinates $x_1, \ldots, x_n$ for the atoms given only a set of lower and upper bounds, $l_{i,j}$ and $u_{i,j}$, of the distances $d_{i,j}$ such that $l_{i,j} \le d_{i,j} \le u_{i,j}$ where (i, j) is in S.

The distance geometry problem is polynomial time solvable if the distances for all pairs of atoms are available (Havel, 1995). However, it has been proved to be NP-hard in general (Saxe, 1979). Even if errors are allowed for the distances, the problem is still hard, if only small errors are allowed (Moré and Wu, 1996). The existing approaches to the problem and their recent developments include, for example, the embed algorithm by Crippen and Havel (1988), Havel (1991), the alternating projection method by Glunt et al. (1990, 1993), the graph reduction approach by Hendrickson (1992, 1995), the global smoothing method by Moré and Wu (1997, 1999), the stochastic/perturbation method by Zou et al. (1997), the multidimensional scaling method by Kearsly et al. (1998), Trosset (1998), the dc programming method by Le Thi Hoai and Pham Dinh (2003), the semidefinite programming approach by Biswas et al. (2007), and the stochastic search method by Grosso et al. (2007).

We investigate the solution of the distance geometry problem within a so-called geometric buildup framework. Dong and Wu (2002, 2003) first implemented a geometric buildup algorithm for the solution of the distance geometry problem with exact distances and justified the linear computation time for the case when the distances required in every buildup step are always available. Central to the algorithm is the idea that whenever there are four determined atoms that are not in the same plane and there are distances from these atoms to an undetermined atom, the undetermined atom can immediately be determined uniquely by solving a system of four distance equations using the available distances. If for every atom, the required atoms and the distances can be found, the whole structure can be determined uniquely. The distance equations can in fact be reduced to a set of linear equations, and hence solved in constant time. Therefore, in ideal cases, a geometric buildup algorithm can solve a distance geometry problem with only $4n$ distances in $O(n)$ computing time, while the conventional singular value decomposition algorithm requires

all $n(n-1)/2$ distances and $O(n^2)$ computing time, where n is the number of atoms to be determined.

The geometric buildup algorithm can be sensitive to the numerical errors though, for the coordinates of the atoms are determined using the coordinates of previously determined atoms and the rounding errors in the previously determined atoms can be passed to and accumulated in later determined atoms, resulting in incorrect structural results. Wu and Wu (2007) proposed an updating scheme to prevent the accumulation of the numerical errors. The idea of the scheme is based on the fact that the coordinates of any four atoms can be determined without any other information if all the distances among them are given. Therefore, the coordinates of any four determined atoms can be recalculated whenever possible using the distances among them, before they are used as a basis set of atoms for the determination of other atoms. The recalculated coordinates do not depend on the coordinates of previously determined atoms and, therefore, do not inherit any errors from them. They are determined from "scratch" and will not pass errors to later atoms.

The geometric buildup algorithm cannot tolerate errors in given distances either, for the distances then may not be consistent and the systems of distance equations may not be solvable. However, in practice, the distances must have errors because they come from either experimental measures or theoretical estimates. In order for the algorithm to handle inexact distances (distances with errors), the general buildup procedure has to be modified. First, in every buildup step, if l distances are found from an undetermined atom to l determined atoms, $l \geq 4$, all l distances should be used for the determination of the unknown atom. The reason is that if the distances have errors, they can be inconsistent. Then the atom satisfying four of the distances may not necessarily satisfy the rest of the distances and, therefore, it should be determined with all its distance constraints. Second, if $l \geq 4$, an over-determined system of equations is obtained for the determination of the position of the unknown atom. If the distances have errors, the system may not be consistent. Therefore, we can only solve the system approximately by using for example a least-squares method. Third, a new updating scheme may be necessary to prevent the accumulation of the rounding errors. The previously developed updating scheme may not be practical any more for $l \gg 4$ because it requires all the distances available among l determined atoms.

We propose a new geometric buildup algorithm which can prevent the accumulation of the rounding errors in the buildup calculations successfully and also tolerate small errors in the given distances. In this algorithm, we use all (instead of a subset of) the distances available for the determination of each unknown atom and obtain the position of the atom by using a least-squares approximation (instead of solving a system of equations exactly, see Fig. 1). The least-squares approximation can be implemented with either a linear or nonlinear formulation. The linear formulation can be obtained from the reduced linear system of equations for the determination of the coordinates of the unknown atom. The nonlinear formulation can be defined directly with the original system of distance equations. The linear least-squares problem can be solved using a standard method. The nonlinear least-squares problem may not be solved easily if an iterative method is used. However, we show that it can actually be solved by using a special singular value decomposition method, which can not only provide a good solution to the problem, but also prevent the accumulation of the rounding errors in the buildup procedure effectively. We describe these least-squares formulations and their solution methods. We present the test

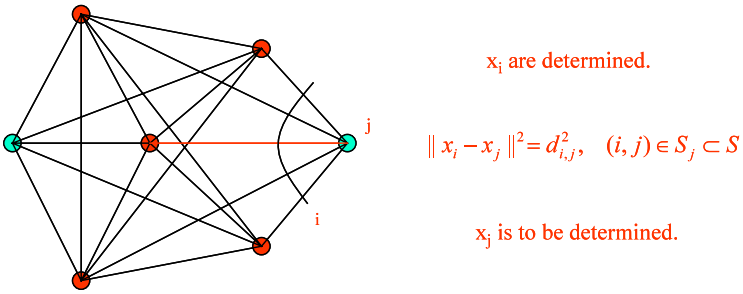

Fig. 1 Tolerance of distance errors The algorithm tries to determine the coordinates of each atom by taking all available distance constraints into account and by minimizing the errors for all the constraints. In this way, all the constraints are intended to be satisfied, and the algorithm is also more stable with possible errors in the distance data.

results from applying the new algorithm to the determination of a set of protein structures with varying degrees of availability and accuracy of the distances and show that the new development increases the modeling ability and improves stability and robustness of the geometric buildup approach significantly from both theoretical and practical point of views.

2. The general geometric buildup approach

Given an arbitrary set of distances, the general geometric buildup algorithm first finds four atoms that are not in the same plane and determines the coordinates for the four atoms with all the distances among them (assuming available). Then for any undetermined atom j, the algorithm repeatedly performs a procedure as follows: Find four determined atoms that are not in the same plane and have distances available to atom j, and determine the coordinates for atom j. Let $x_i = (x_{i,1}, x_{i,2}, x_{i,3})^T$, $i = 1, 2, 3, 4$, be the coordinate vectors of the four atoms. Then the coordinates $x_j = (x_{j,1}, x_{j,2}, x_{j,3})^T$ for atom j can be determined by using the distances $d_{i,j}$ from atoms $i = 1, 2, 3, 4$ to atom j. Indeed, x_j can be obtained from the solution of the following system of equations,

$$\|x_i\|^2 - 2x_i^T x_j + \|x_j\|^2 = d_{i,j}^2, \quad i = 1, 2, 3, 4. \tag{1}$$

By subtracting equation i from equation $i + 1$ for $i = 1, 2, 3$, we can eliminate the quadratic terms for x_j to obtain

$$\begin{aligned} &-2(x_{i+1} - x_i)^T x_j \\ &\quad = \left(d_{i+1,j}^2 - d_{i,j}^2\right) - \left(\|x_{i+1}\|^2 - \|x_i\|^2\right), \quad i = 1, 2, 3. \end{aligned} \tag{2}$$

Let A be a matrix and b a vector, and

$$A = -2\begin{bmatrix} (x_2 - x_1)^T \\ (x_3 - x_2)^T \\ (x_4 - x_3)^T \end{bmatrix}, \qquad b = \begin{bmatrix} \left(d_{2,j}^2 - d_{1,j}^2\right) - \left(\|x_2\|^2 - \|x_1\|^2\right) \\ \left(d_{3,j}^2 - d_{2,j}^2\right) - \left(\|x_3\|^2 - \|x_2\|^2\right) \\ \left(d_{4,j}^2 - d_{3,j}^2\right) - \left(\|x_4\|^2 - \|x_3\|^2\right) \end{bmatrix}. \tag{3}$$

We then have $Ax_j = b$. Since x_1, x_2, x_3, x_4 are not in the same plane, A must be nonsingular, and we can therefore solve the linear system to obtain a unique solution for x_j. Here, solving the linear system requires only constant time. Since we only need to solve $n-4$ such systems for $n-4$ coordinate vectors x_j, the total computation time is proportional to n, if in every step, the required coordinates x_i and distances $d_{i,j}$, $i = 1, 2, 3, 4$ are always available.

The General Geometric Buildup Algorithm

1. Find four atoms that are not in the same plane.
2. Determine the coordinates of the atoms with the distances among them.
3. Repeat:
   For each of the undetermined atoms,
   If the atom has 4 distances to 4 determined atoms that are not in the same plane,
   Determine the atom with the distances.
   End
   End
4. If no atom can be determined in the loop, stop.
5. All atoms are determined.

The theoretical basis of the geometric buildup approach can be traced back in the study of distance geometry in mathematics (Blumenthal, 1953). The earliest proposal for such an approach can be found in Sippl and Scheraga (1985, 1986). Huang et al. (2003) recently discussed some related theoretical issues in the context of distance matrix completion. Based on the distance geometry theory, any point in a Euclidean space can be determined in terms of the distances from this point to a special set of points.

Definition 2.1. A set of points B in a space S is a metric basis of S provided each point of S is uniquely determined by its distances from the points in B.

Definition 2.2. A set of $k+1$ points in R^k is called independent if it is not a set of points in R^{k-1}.

Theorem 2.1. *Any $k+1$ independent points in R^k form a metric basis for R^k.*

Proof: It follows directly by generalizing the basic geometric buildup step to the k-dimensional Euclidean space. Let $x_i = (x_{i,1}, \ldots, x_{i,k})^T$ be the coordinate vectors of an independent set of points $i = 1, \ldots, k+1$ in R^k. Let $x_j = (x_{j,1}, \ldots, x_{j,k})^T$ be the coordinate vector for any point j in R^k with distances $d_{i,j}$ from points $i = 1, \ldots, k+1$ to point j. Then

$$\|x_i\|^2 - 2x_i^T x_j + \|x_j\|^2 = d_{i,j}^2, \quad i = 1, \ldots, k+1, \tag{4}$$

and $Ax_j = b$, where

$$A = -2\begin{bmatrix} (x_2 - x_1)^T \\ (x_3 - x_2)^T \\ \vdots \\ (x_{k+1} - x_k)^T \end{bmatrix}, \qquad b = \begin{bmatrix} \left(d_{2,j}^2 - d_{1,j}^2\right) - \left(\|x_2\|^2 - \|x_1\|^2\right) \\ \left(d_{3,j}^2 - d_{2,j}^2\right) - \left(\|x_3\|^2 - \|x_2\|^2\right) \\ \vdots \\ \left(d_{k+1,j}^2 - d_{k,j}^2\right) - \left(\|x_{k+1}\|^2 - \|x_k\|^2\right) \end{bmatrix}. \quad (5)$$

Since the points $i = 1, \ldots, k+1$ are not in R^{k-1}, the matrix A must be nonsingular and x_j is determined uniquely. □

3. Control of numerical errors

The general geometric buildup algorithm can be sensitive to the numerical errors generated during the calculation of the coordinates of the atoms. With this algorithm, the coordinates of many atoms are determined by using the coordinates of previously determined atoms and, therefore, the errors in the previously determined atoms are passed to and accumulated in later determined atoms. As a result, the coordinates for later determined atoms may become completely incorrect, especially if there is a long sequence of atoms to be determined.

Wu and Wu (2007) proposed an updating scheme to prevent the accumulation of the numerical errors. The idea of the scheme is based on the fact that the coordinates of any four atoms can be determined without any other information if all the distances among them are given. Therefore, the coordinates of any four determined atoms should be recalculated whenever possible using the distances among them, before they are used as a basis set of atoms for the determination of other atoms. The recalculated coordinates do not depend on the coordinates of previously determined atoms and, therefore, do not inherit any errors from them. They are determined from "scratch" and will not pass previous errors to later atoms as well. In this way, the coordinates of many atoms can be "corrected," and the errors in the calculated coordinates can be prevented from growing into incorrect structural results.

The recalculation of the coordinates of the four atoms in the above algorithm usually is done in an independent coordinate system, which is not related to the overall structure already constructed by the algorithm. However, they can be moved back to the original structure by aligning them to their original locations with an appropriate translation and rotation. In other words, the new coordinates of the four atoms can be translated and rotated so that the root-mean-square-deviation (RMSD) between the new coordinates and the old ones is minimized.

Let $y_1, \ldots, y_4$ be the coordinate vectors of the four atoms calculated in the regular geometric buildup process, and $x_1, \ldots, x_4$ the recalculated coordinate vectors. Let Y and X be the corresponding coordinate matrices, i.e.,

$$Y = \{y_{i,k} : i = 1, \ldots, 4, k = 1, 2, 3\} \quad \text{and} \quad X = \{x_{i,k} : i = 1, \ldots, 4, k = 1, 2, 3\}. \quad (6)$$

In order to move X to the position where Y is located in the molecule, the geometric centers of X and Y are calculated first:

$$x_c^T = \sum_{i=1}^{4} X(i,:)/4, \qquad y_c^T = \sum_{i=1}^{4} Y(i,:)/4. \tag{7}$$

Then X is translated so that the geometric centers of X and Y are at the same location,

$$X \leqslant X + e(y_c - x_c)^T, \tag{8}$$

where $e = (1, 1, 1, 1)^T$. After the translation, a rotation for X is selected so that the root-mean-square-deviation of X and Y is minimized. In fact, the calculation of such a deviation can be done by solving an optimization problem,

$$\min_Q \|Y - XQ\|_F, \quad QQ^T = I, \tag{9}$$

where $\|\|_F$ is the matrix Frobenius norm and Q the rotation matrix. Let $C = X^T Y$, and let $C = U\Sigma V^T$ be the singular-value decomposition of C. Then it is not difficult to verify that $Q = UV^T$ solves the above optimization problem (Golub and van Loan, 1989).

4. Tolerance of distance errors

In practice, the distance data often contains errors. As a result, the distances may become inconsistent or have violated some basic rules such as the triangle inequality. In terms of graph embedding, the distance graph may not be realizable in a given space for such a set of distances. Generally, the geometric buildup algorithm assumes that the distances are consistent and, therefore, in every step, only four distances are required for the determination of the coordinates of an atom uniquely, although there may be more available. However, this will not be the case if the distances are not consistent.

The geometric buildup algorithm can be extended in a straightforward manner to handling the possible errors from the distance data. For example, in every buildup step, in addition to the four required distances, we can include all the available distances, say l distances, from the determined atoms to the one to be determined. Let $x_i = (x_{i,1}, x_{i,2}, x_{i,3})^T$, $i = 1, \ldots, l$, be the coordinate vectors of the l determined atoms and $d_{i,j}$ the distances from atoms $i = 1, \ldots, l$ to the undetermined atom j. Then the coordinates $x_j = (x_{j,1}, x_{j,2}, x_{j,3})^T$ for atom j can be obtained from the solution of the following system of equations,

$$\|x_i\|^2 - 2x_i^T x_j + \|x_j\|^2 = d_{i,j}^2, \quad i = 1, \ldots, l. \tag{10}$$

By subtracting equation i from equation $i + 1$ for $i = 1, \ldots, l - 1$, we can eliminate the quadratic terms for x_j to obtain

$$\begin{aligned} &-2(x_{i+1} - x_i)^T x_j \\ &\quad = \left(d_{i+1,j}^2 - d_{i,j}^2\right) - \left(\|x_{i+1}\|^2 - \|x_i\|^2\right), \quad i = 1, \ldots, l-1. \end{aligned} \tag{11}$$

Let A be a matrix and b a vector, and

$$A = -2\begin{bmatrix} (x_2 - x_1)^T \\ (x_3 - x_2)^T \\ \vdots \\ (x_l - x_{l-1})^T \end{bmatrix}, \qquad b = \begin{bmatrix} (d_{2,j}^2 - d_{1,j}^2) - (\|x_2\|^2 - \|x_1\|^2) \\ (d_{3,j}^2 - d_{2,j}^2) - (\|x_3\|^2 - \|x_2\|^2) \\ \vdots \\ (d_{l,j}^2 - d_{l-1,j}^2) - (\|x_l\|^2 - \|x_{l-1}\|^2) \end{bmatrix}. \quad (12)$$

We then have $Ax_j = b$. This system is certainly over-determined if $l > 4$. However, it can be solved by using a standard linear least-squares method. For example, we can compute the QR factorization of A to obtain an equation $QRx_j = b$, where Q is $(l-1) \times 3$ and R is 3×3. If at least four of the l determined atoms are not in the same plane, A must be full rank and R be nonsingular. We can solve the linear system $QRx_j = b$ to obtain an unique solution $x_j = R^{-1}Q^T b$. Here, solving the linear system $QRx_j = b$ requires $O(l)$ computing time, but QR factorization may take $O(l^2)$ time. We can also take another so-called normal equation method, although it may not be as stable as the QR method: We can first multiply the equation $Ax_j = b$ by A^T to obtain $A^T Ax_j = A^T b$. If at least four of the l determined atoms are not in the same plane, A must be full rank and $A^T A$ be nonsingular. We can then solve the linear system $A^T Ax_j = A^T b$ to obtain a unique solution $x_j = [A^T A]^{-1} A^T b$. Here, solving the linear system $A^T Ax_j = A^T b$ requires only constant time, but $A^T A$ may take $O(l)$ time. In either case, since we only need to solve $\sim n$ linear least-squares problems for $\sim n$ coordinate vectors x_j, the total computation time must be in order of either $l_m^2 n$ or $l_m n$, if in every step, the required coordinates x_i and distances $d_{i,j}$ are always available, where $l_m = \max_j\{|S_j|\}$, $S_j = \{i : (i, j) \text{ in } S\}$.

The above solution to the system $Ax_j = b$ can be exact, if the system is consistent or in other words, if the original distance are consistent and do not have errors. However, it still provides the best approximation to the solution of the system, even if the system is inconsistent or in other words, if the original distances are inconsistent or have errors. In this sense, the extended geometric buildup algorithm should be more robust and stable than the general algorithm, in addition to being able to tolerate small errors in the distance data.

Geometric Buildup with Linear Least-Squares

1. Find four atoms that are not in the same plane.
2. Determine the coordinates of the atoms with the distances among them.
3. Repeat:
   For each of the undetermined atoms,
   If the atom has l distances to l determined atoms that are not in the same plane,
   Determine the atom with the least-squares fit to the distances.
   End
   End
4. If no atom can be determined in the loop, stop.
5. All atoms are determined.

Again, the theory for the extended geometric buildup algorithm can be established and generalized to any k-dimensional Euclidean space in a similar fashion as that for the general geometric buildup algorithm. For this purpose, we define an extended set of independent points in R^k.

Definition 4.1. A set of l points is said to be an extended set of independent points in R^k if it contains $k+1$ independent points.

The following result is a trivial generalization of Theorem 2.1 and we state it without proof.

Theorem 4.1. *An extended set of l independent points in R^k forms a metric basis for R^k.*

5. Singular value decomposition

The algorithm described in the previous section may not necessarily be stable for preventing rounding errors from growing, because in every step, the coordinates of the unknown atom must have rounding errors, which can still be propagated and accumulated into later calculations. Different from the general algorithm, it is difficult to apply an updating scheme as described in Section 3 in the new algorithm, because the scheme requires the availability of the distances among all l determined atoms, which is not so realistic when l is large. Here, we describe another buildup procedure that may resolve this problem. The idea is to determine the unknown atom in each buildup step by using not only the l distances from l determined atoms to the unknown atom, but also the distances among all the l determined atoms. The l distances from l determined atoms to the unknown atom must be given. The distances among the l determined atoms may not necessarily be provided, but they can be calculated. In any case, once all these distances become available, the coordinates for the unknown atom and the l known atoms can all be calculated (or recalculated) using these distances.

In general, let $x_1, \ldots, x_l$ and x_{l+1} be the coordinate vectors of atoms $1, \ldots, l+1$. If the distances among all these atoms, $d_{i,j}$, $i, j = 1, \ldots, l+1$, are available, then $\|x_i - x_j\| = d_{i,j}$ for all $i, j = 1, \ldots, l+1$, and

$$\|x_i\|^2 - 2x_i^T x_j + \|x_j\|^2 = d_{i,j}^2, \quad i, j = 1, \ldots, l+1. \tag{13}$$

Since the structure formed by these atoms is invariant under any translation or rotation, we can set a reference system so that the origin is located at the last atom or in other words, $x_{l+1} = (0, 0, 0)^T$. It follows that $\|x_i\| = d_{i,l+1}$, $\|x_j\| = d_{j,l+1}$, and

$$d_{i,l+1}^2 - 2x_i^T x_j + d_{j,l+1}^2 = d_{i,j}^2, \quad i, j = 1, \ldots, l. \tag{14}$$

Define a coordinate matrix X and an induced distance matrix D,

$$\begin{aligned} &X = \{x_{i,k} : i = 1, \ldots, l, k = 1, 2, 3\} \quad \text{and} \\ &D = \left\{\left(d_{i,l+1}^2 - d_{i,j}^2 + d_{j,l+1}^2\right)/2 : i, j = 1, \ldots, l\right\}. \end{aligned} \tag{15}$$

Then it is easy to verify that $XX^T = D$ and D must be of maximum rank 3.

Let $D = U\Sigma U^T$ be the singular value decomposition of D, where U is an orthogonal matrix and Σ a diagonal matrix with the singular values of D along the diagonal. If D is a matrix of rank less than or equal to 3, $X = V\Lambda^{1/2}$ solves the equation $XX^T = D$, where $V = U(:, l:3)$ and $\Lambda = \Sigma(1:3, 1:3)$. In other words, if the distances $d_{i,j}$ are available for all $i, j = 1, \ldots, l+1$, we can always construct an induced matrix D for the distances and then, based on the singular value decomposition of D, obtain the coordinates for all the atoms $1, \ldots, l$ as given in X with atom $l+1$ fixed at $(0, 0, 0)^T$.

The above procedure can in fact be applied to any $l+1$ atoms, and is one of the standard algorithms for the solution of the distance geometry problems, when the distances for all pairs of atoms in the molecule are given. The algorithm can also be generalized to problems in any k-dimensional Euclidean space, with X being an $l \times k$ matrix and D being an $l \times l$ matrix. In general,

Theorem 5.1. *Let $\{d_{i,j} : i, j = 1, \ldots, l+1\}$ be a set of distances in R^k, for some $k < l$. Then the matrix D as induced in (15) is of maximum rank k.*

Proof: It follows from the facts that $D = XX^T$ and X is an $l \times k$ matrix with maximum rank k when $k < l$. □

Theorem 5.2. *Let $D = U\Sigma U^T$ be the singular value decomposition of D. If D is a matrix of rank less than or equal to k, $X = V\Lambda^{1/2}$ solves the equation $XX^T = D$, where $V = U(:, 1:k)$ and $\Lambda = \Sigma(1:k, 1:k)$.*

Proof: If D is of maximum rank k, D can be decomposed into $U\Sigma U^T$ with U being an $l \times k$ orthogonal matrix and Σ an $k \times k$ diagonal matrix. It follows that $XX^T = D$, if $X = V\Lambda^{1/2}$. □

Note that the distances may have errors. Then the matrix D may have a higher rank than k or in other words, the equation $XX^T = D$ may not have an exact solution. However, $X = V\Lambda^{1/2}$ as defined above is still a good approximation to the solution of the equation in the following nonlinear least-squares sense.

Theorem 5.3. *Let $D = U\Sigma U^T$ be the singular value decomposition of D. Let $V = U(:, 1:k)$ and $\Lambda = \Sigma(1:k, 1:k)$. Then $X = V\Lambda^{1/2}$ minimizes $\|D - XX^T\|_F$, where $\|\|_F$ is the matrix Frobenius norm.*

Proof: (Havel, 1998) Let $f(X) = \|D - XX^T\|^2$. Then $(D - XX^T)X = 0$ for any stationary point X of f. It follows that $(D - XX^T)X = (D - XX^T)XX^T = 0$ and

$$f(X) = \operatorname{trace}\left(D^2\right) - \operatorname{trace}\left(2DXX^T - XX^TXX^T\right) = \operatorname{trace}\left(D^2\right) - \operatorname{trace}\left(XX^TXX^T\right).$$

Let $\sigma_1 \geq \cdots \geq \sigma_l \geq 0$ be the singular values of D and $\lambda_1 \geq \cdots \geq \lambda_k > 0$ be the singular values of XX^T. Then

$$f(X) = \operatorname{trace}\left(D^2\right) - \operatorname{trace}\left(XX^TXX^T\right) = \sum_{j=1}^{l} \sigma_j^2 - \sum_{j=1}^{k} \lambda_j^2.$$

Let $XX^T = V\Lambda V^T$ be the singular value decomposition of XX^T, where V is an $l \times k$ orthogonal matrix and $\Lambda = \text{diag}\{\lambda_1, \ldots, \lambda_k\}$. Since $DXX^T = XX^TXX^T$, $V^TDV = \Lambda$ and, therefore, $\{\lambda_j : j = 1, \ldots, k\} \subset \{\sigma_j : j = 1, \ldots, n\}$. It follows that $f(X)$ is minimized when $\lambda_j = \sigma_j$ for $j = 1, \ldots, k$. □

Based on the above discussion, a buildup procedure can immediately be implemented as follows. In every buildup step, construct an induced matrix D from the distances $d_{i,j}, i, j = 1, \ldots, l+1$ among $l+1$ atoms,

$$D = \left\{\left(d_{i,l+1}^2 - d_{i,j}^2 + d_{j,l+1}^2\right)/2 : i, j = 1, \ldots, l\right\}, \tag{16}$$

where $d_{i,j}, i, j = 1, \ldots, l$ are the distances among l determined atoms and $d_{i,l+1}, i = 1, \ldots, l$ are the distances from the determined atoms to the undetermined one. The former are either given in the original distance data or calculated using the determined coordinates of the related atoms. The latter must be given and cannot be calculated because atom $l+1$ is undetermined. Assuming the availability of all these distances, we can then compute the singular value decomposition of $D = U\Sigma U^T$, and obtain $X = V\Lambda^{1/2}$ with $V = U(:, l : 3)$ and $\Lambda = \Sigma(1 : 3, 1 : 3)$, and hence the coordinates of all the atoms $1, \ldots, l+1$, with the coordinates of atom $l+1$, the undetermined atom, at $(0, 0, 0)^T$.

The results from the above calculations have several folds. First, the coordinates of the unknown atom are determined by using l previously determined atoms, to which the unknown atom has distances given. Second, the coordinates are determined by solving a system of distance equations approximately. They are the best possible estimations in a nonlinear least-squares sense as stated in Theorem 5.3, and can therefore be evaluated even if the distances have errors. Third, the calculations not only determine the coordinates of the unknown atom, but also recalculate the coordinates of all the involved atoms including the determined ones. Most importantly, these coordinates do not depend completely on the results from previous calculations. Rather, they are determined by using the provided distances among the atoms (determined and undetermined) as much as possible, thereby reducing the risk of large error propagation and accumulation. In this sense, the method should be more stable numerically than the one described in the previous section.

Of course, the calculations of the coordinates are conducted in an independent reference system with its origin at the position of the atom to be determined. In order to recover the coordinates of the atoms in their original structure, we need to make a proper translation and rotation for the coordinates just like we need to do in the updating scheme for the general geometric buildup algorithm. More specifically, let Y be an $l \times 3$ matrix having the original coordinates of the l determined atoms. Let X be an $l \times 3$ matrix with the recalculated coordinates of the determined atoms. First, we translate X to Y with a translation vector $y_c - x_c$, where x_c and y_c are the geometric centers of X and Y, respectively. Then we can rotate the coordinates of all the atoms by using a rotation matrix $Q = UV^T$, where U and V are obtained from the singular value decomposition, $X^TY = U\Sigma V^T$. That is, if x_i is the coordinate vector of atom $i, i = 1, \ldots, l+1$, then we set x_i to Qx_i.

Geometric Buildup with Nonlinear Least-Squares

1. Find four atoms that are not in the same plane.
2. Determine the coordinates of the atoms with the distances among them.

3. Repeat: For each of the undetermined atoms,
   If the atom has l distances to l determined atoms that are not in the same plane,
   Determine the $l + 1$ atoms with the distances among them.
   Put the atoms back to their original positions by proper translation and rotation
   End
   End
4. If no atom can be determined in the loop, stop.
5. All atoms are determined.

6. Test results

In this section, we present the test results from applying the new geometric buildup algorithm to the determination of a set of protein structures with varying degrees of availability and accuracy of the distances. We first downloaded eleven protein structures from the PDB databank with the number of atoms ranging from 402 to 7398. With each of these structures, we generated four sets of distance data with the cutoff distances correspondingly equal to 5, 6, 7, and 8 Å. For each generated distance set, we applied the new algorithm to obtain a structure. The obtained structure was then evaluated with the coordinate RMSD against its original structure.

We have implemented the new algorithm with both linear and nonlinear least-squares buildup strategies as described in Sections 4 and 5, respectively. The programs were written in MATLAB and run on a standard desktop workstation. Table 1 contains information for the distance data generated from each of the downloaded structures including the number of atoms in the structure, the total number of distances between all pairs of atoms, and the numbers of distances generated under specified cutoff distances. From this table, we can see that for each of the structures, a very sparse set of distances (ranging from 0.32% to 17.40%) was generated with specified cutoff distances. The distances became denser when a larger cutoff distance was used (as can be observed from each row of the table). However, as the number of atoms in the structure increases, the sparsity of the generated

Table 1 Available distances for different cutoff values*

ID	TA	TD	≤5 Å		≤6 Å		≤7 Å		≤8 Å	
			AD	AD/TD	AD	AD/TD	AD	AD/TD	AD	AD/TD
1PTQ	402	80601	4399	5.46%	7088	8.79%	10302	12.78%	14023	17.40%
1HOE	558	155403	6299	4.05%	10178	6.55%	14936	9.63%	20423	13.14%
1LFB	641	205120	6974	3.40%	11435	5.57%	16602	8.09%	22519	10.98%
1PHT	814	330891	11033	3.33%	17695	5.35%	26299	7.95%	36077	10.90%
1POA	914	417241	10468	2.51%	16983	4.07%	24984	5.99%	34485	8.27%
1AX8	1003	502503	11542	2.30%	18795	3.74%	27286	5.43%	37130	7.39%
4MBA	1086	589155	12761	2.17%	20905	3.55%	30706	5.21%	42151	7.15%
1F39	1534	1175811	17300	1.47%	28532	2.43%	42678	3.63%	59551	5.06%
1RGS	2015	2029105	22784	1.12%	38020	1.87%	56298	2.77%	77513	3.82%
1BPM	3672	6739956	44789	0.66%	75152	1.12%	112940	1.68%	159303	2.36%
1HMV	7398	27361503	86288	0.32%	143196	0.52%	214498	0.78%	299939	1.10%

*ID—protein ID, TA—total number of atoms, TD—total number of distances, AD—available distances

Table 2 RMSD values of structures computed with linear least-squares

ID	TA	≤5 Å		≤6 Å		≤7 Å		≤8 Å	
		DA	RMSD	DA	RMSD	DA	RMSD	DA	RMSD
1PTQ	402	402	1.4E+00	402	2.6E−09	402	1.7E−13	402	1.3E−13
1HOE	558	558	5.8E−02	558	3.1E−09	558	1.6E−13	558	1.8E−13
1LFB	641	641	2.0E−02	641	2.1E−10	641	6.7E−13	641	1.3E−13
1PHT	814	809	1.2E+01	814	8.2E−09	814	3.1E−13	814	1.8E−13
1POA	914	914	6.6E+00	914	1.9E−09	914	5.3E−13	914	4.9E−13
1AX8	1003	1003	5.2E+00	1003	1.8E−05	1003	6.7E−12	1003	7.7E−13
4MBA	1086	1083	4.9E+00	1086	3.8E−06	1086	1.1E−10	1086	3.7E−12
1F39	1534	1534	1.4E+01	1534	6.3E−08	1534	4.6E−11	1534	1.6E−10
1RGS	2015	2010	2.0E+01	2015	1.1E−01	2015	5.5E−10	2015	1.7E−12
1BPM	3672	3669	6.4E+04	3672	3.6E−02	3672	3.4E−09	3672	5.5E−12
1HMV	7398	7389	1.2E+03	7398	3.5E+01	7398	1.1E−04	7398	5.5E−10

ID—protein ID, TA—total number of atoms, DA—total number of determined atoms, RMSD—RMSD values of the computed structure against the original structures

distances also increases for a fixed cutoff distance (as can be observed from each column of the table). The purpose of using different cutoff distances was to obtain different sets of distance data with different sparsities so we can test the algorithm for problems with varying degrees of availability of the distances. As we have discussed in the previous section, the problem becomes usually unrealistic for practical cases when the number of available distances is large. For realistic cases, for instance in NMR experiments, the number of available distances is always small since the distance cutoff is about 5 or 6 Å. In our work, we also considered the cases of larger cutoffs like 7 and 8 Å for the purpose of numerical study. These results are listed for purely mathematical and numerical purposes, and they will not affect practicality of the algorithm because it behaves very well for sparse distance data.

Table 2 contains the RMSD (root-mean-square deviation) values of the structures (compared with their original structures) obtained by using the new buildup algorithm with linear least-squares on the data sets listed in Table 1. The RMSD values show that the algorithm solved almost all the problems with cutoff distances equal to 6, 7, and 8 Å, but failed for those with cutoff distance equal to 5 Å. The last cutoff value is critical because in NMR modeling, usually only less than or equal 5 Å distances can be estimated. In any case, the results show that with linear least-squares, the new buildup algorithm performed well in general if the distance data was not too sparse. The reason that it did not work well for very sparse data was that a long sequence of buildup steps had to be carried out and a large amount of rounding errors was accumulated.

Table 3 contains the RMSD (root-mean-square deviation) values of the structures (compared with their original structures) obtained by using the new buildup algorithm with nonlinear least-squares on the data sets listed in Table 1. The RMSD values show that the algorithm solved almost all the problems with cutoff distances equal to 5, 6, and 7 Å, but failed for those with cutoff distance equal to 8 Å. The large cutoff values are in fact not so important because in practice, usually only shorter distances can be estimated. Therefore, the results indicated that with nonlinear least-squares, the new buildup algorithm performed well in general. The reason it worked well for very sparse data was that it calculated the coordinates of the undetermined as well as determined atoms in every

Table 3 RMSD values of structures computed with nonlinear least-squares

ID	TA	≤5 Å		≤6 Å		≤7 Å		≤8 Å	
		DA	RMSD	DA	RMSD	DA	RMSD	DA	RMSD
1PTQ	402	402	5.5E−14	402	5.0E−14	402	2.5E−12	402	5.6E−11
1HOE	558	558	1.6E−13	558	2.7E−13	558	9.2E−13	558	3.6E−07
1LFB	641	641	9.5E−14	641	5.5E−14	641	2.5E−13	641	3.1E−09
1PHT	814	809	1.1E−13	814	1.8E−13	814	2.2E−08	814	3.0E+08
1POA	914	914	3.2E−13	914	1.5E−13	914	2.5E−10	914	8.3E−03
1AX8	1003	1003	4.0E−13	1003	4.6E−12	1003	2.2E−08	1003	8.7E+10
4MBA	1086	1083	1.8E−13	1086	2.6E−13	1086	3.3E−10	1086	7.1E+02
1F39	1534	1534	7.9E−13	1534	1.9E−13	1534	8.2E−08	1534	1.0E+34
1RGS	2015	2010	8.3E−12	2015	2.4E−12	2015	5.3E−07	2015	4.6E+28
1BPM	3672	3669	8.1E−11	3672	1.0E−11	3672	7.0E+26	–	–
1HMV	7398	7389	1.1E−08	7398	5.5E−07	–	–	–	–

ID—protein ID, TA—total number of atoms, DA—total number of determined atoms, RMSD—RMSD values of the computed structure against the original structures

Table 4 Total CPU times elapsed during structure determination (in seconds)

ID	≤5 Å		≤6 Å		≤7 Å		≤8 Å	
	LNLS	NLLS	LNLS	NLLS	LNLS	NLLS	LNLS	NLLS
1PTQ	0.312	0.390	0.484	0.920	0.437	1.201	0.499	2.137
1HOE	0.577	0.889	0.593	1.295	0.749	2.075	0.608	3.370
1LFB	0.733	1.014	0.718	1.295	0.780	2.028	0.733	3.526
1PHT	1.092	1.716	1.154	2.309	1.076	3.806	1.310	7.769
1POA	1.326	1.576	1.217	2.278	1.217	3.494	1.498	5.912
1AX8	1.420	1.981	1.544	2.480	1.404	3.853	1.778	6.193
4MBA	1.685	2.090	1.778	2.761	1.669	4.337	1.950	7.316
1F39	2.933	3.604	3.104	4.758	3.104	7.192	3.120	11.809
1RGS	4.976	5.741	4.820	7.457	4.914	10.312	5.320	16.630
1BPM	15.038	16.848	15.179	20.327	15.600	26.863	15.975	–
1HMV	61.464	64.772	61.464	69.280	61.402	–	62.900	–

ID—protein ID, LNLS—total CPU time elapsed during structure determination using the linear least-squares method, NLLS—total CPU time elapsed during structure determination using the nonlinear least-squares method

buildup step using the distances among them (most presumably given in the original distance data) and therefore, stopped the propagation of the rounding errors. It did not work well for larger cutoff distances because in those cases, most of the distances among a group of atoms to be determined were probably calculated instead of given and, therefore, the recalculated coordinates of the atoms would still inherit the errors produced in previous calculations through these distances.

Table 4 presents the performance results for the same test cases as shown in Tables 2, 3, with the times required by both algorithms, linear least-squares (LNLS) and nonlinear least-squares (NLLS). The programs were run in Matlab R2008b version 7.7 on a Dell Laptop, with 1.86 GHz CPU and 2.00 GB memory. From the table, we can see that the computing times of both algorithms were comparable, with the nonlinear one requiring

slightly longer time. However, both turned out to be very efficient, and were able to finish the calculations in only a few seconds to a few minutes for almost all the test cases.

Tables 5 and 6 further demonstrate the behaviors of the new algorithm for distances with some small magnitudes of errors. In order to obtain these results, we have first used the distances generated for the proteins with the cutoff distance equal to 5 and 6 Å and perturbed them with some small random errors. More specifically, we perturbed every generated distance d by using an update formula $d \leqslant d + 2 * \mathrm{RE} * (0.5 - \mathrm{rand}) * d$, where RE are the maximum relative errors and RE = 1.0E−08, 1.0E−07, 1.0E−06, 1.0E−05, and 1.0E−04, and rand is a function which returns a random number in [0, 1]. We have then obtained a new set of distance data for each protein, with the cutoff distance equal to 5 or 6 Å. The distances have errors and can be inconsistent. For each of these data sets, we applied the new algorithm again to obtain a structure for the corresponding protein and also calculated the RMSD value of the structure against its original structure. Table V shows that for very sparse distances with cutoff distance equal to 5 Å, the algorithm with a nonlinear least-squares buildup procedure was able to obtain a good approximated structure for almost all the tested proteins, after the distances were perturbed with RE = 1.0E−08, 1.0E−07, 1.0E−06, 1.0E−05 and 1.0E−04. The algorithm with a linear least-squares buildup procedure did not work well because of an obvious reason of rounding error accumulation. However, when the distances were increased, the latter was able to produce reasonable results as well, while the nonlinear least-squares buildup started having problems for larger test cases with larger distance errors (as can be observed in Table 6). The proposed algorithms failed to produce accurate structures for some of the test cases when the problem sizes are large or the distances are relatively dense (with large distance cutoff values) and, therefore, the accumulated rounding errors or the distance errors are still too large. However, in either case, we observed that the algorithm using nonlinear least-squares always outperformed the one using linear least-squares.

For most of the problems we generated in this work, the previous geometric buildup methods will not work since they do not tolerate any error in the distance data. They do not work either, even for some of the cases with exact distance data. However, in the new algorithm, we include both exact and inexact distance data. Different from previous works done on geometric buildup, our method can also be applied to the problems with distance errors. In order to suppress the test results, we refer the reader to Dong and Wu (2002, 2003), and Wu and Wu (2007), and we will not include test results from previous geometric buildup approaches.

7. Concluding remarks

In this paper, we have described a new extension of the general geometric buildup algorithm to determining protein structures with sparse and possibly inconsistent distances. The general geometric buildup algorithm can be sensitive to the numerical errors, for the coordinates of the atoms are determined using the coordinates of previously determined atoms and the rounding errors in the previously determined atoms can be passed to and accumulated in later determined atoms, resulting in incorrect structural results. The general geometric buildup algorithm cannot tolerate errors in given distances either, for the distances then may not be consistent and the systems of distance equations may not be solvable. However, in practice, the distances must have errors because they come from

Table 5 RMSD values of structures computed with perturbed distances (≤5 Å)

ID	TA	RE: 1.0E−08		RE: 1.0E−07		RE: 1.0E−06		RE: 1.0E−05		RE: 1.0E−04	
		LNLS	NLLS	LNLS	NLLS	LNLS	NLLS	LNLS	NLLS	LNLS	NLLS
1PTQ	402	7.8E+00	1.8E−06	6.9E+00	1.8E−05	1.5E+01	1.7E−04	1.1E+01	2.1E−03	9.9E+00	1.1E−02
1HOE	558	8.2E+00	6.1E−06	8.7E+00	6.2E−05	8.3E+00	6.4E−04	9.3E+00	1.6E−02	1.0E+01	5.7E−03
1LFB	641	1.8E+01	9.5E−07	8.6E+00	9.5E−06	1.5E+01	9.5E−05	1.6E+01	9.9E−03	1.0E+01	3.1E−02
1PHT	814	2.4E+01	1.4E−06	9.7E+00	1.4E−05	1.1E+01	1.7E−04	9.1E+00	1.8E−03	1.2E+01	4.1E−02
1POA	914	9.6E+00	5.9E−06	9.2E+00	5.8E−05	1.2E+01	5.6E−04	1.1E+01	2.2E−03	3.6E+01	3.1E−02
1AX8	1003	2.2E+03	4.1E−06	1.2E+01	4.1E−05	1.4E+01	4.1E−04	1.5E+01	4.2E−03	1.5E+06	4.4E−02
4MBA	1086	1.0E+01	2.5E−06	1.3E+01	2.5E−05	3.0E+01	2.5E−04	1.2E+01	2.4E−03	1.0E+01	1.5E−02
1F39	1534	2.5E+02	2.6E−05	2.4E+04	3.0E−04	9.6E+01	2.2E−03	2.4E+02	1.8E−02	1.1E+02	1.6E+01
1RGS	2015	6.6E+06	8.2E−05	6.2E+02	8.2E−04	3.9E+01	8.3E−03	2.2E+01	1.3E−01	1.6E+01	3.0E+01
1BPM	3672	2.1E+01	2.1E−03	3.3E+02	6.2E−03	2.1E+01	7.9E−02	3.2E+04	2.8E+02	2.6E+01	2.9E+03
1HMV	7398	3.6E+12	9.0E−03	4.5E+03	2.1E+01	5.9E+02	3.7E+04	5.8E+06	4.6E+03	7.3E+07	2.1E+04

ID—protein ID, TA—total number of atoms, RE—relative errors, LNLS—RMSD values obtained using the linear least-squares method, NLLS—RMSD values obtained using the nonlinear least-squares method

Table 6 RMSD values of structures computed with perturbed distances (≤6 Å)

ID	TA	RE: 1.0E−08		RE: 1.0E−07		RE: 1.0E−06		RE: 1.0E−05		RE: 1.0E−04	
		LNLS	NLLS	LNLS	NLLS	LNLS	NLLS	LNLS	NLLS	LNLS	NLLS
1PTQ	402	3.1E−04	1.7E−06	2.9E−03	1.7E−05	1.7E−02	1.7E−04	6.3E−01	7.7E−04	4.7E+00	8.3E−03
1HOE	558	2.3E−04	7.7E−06	3.5E−03	7.6E−05	2.0E−01	7.0E−04	2.1E+00	4.9E−03	2.2E+00	2.6E−02
1LFB	641	1.1E−02	2.3E−07	1.2E−01	2.3E−06	3.7E−01	2.3E−05	3.8E+01	2.3E−04	9.2E+00	2.2E−03
1PHT	814	4.3E−02	3.4E−06	1.2E+00	3.4E−05	2.2E−01	3.4E−04	1.6E+00	3.4E−03	4.3E+00	3.8E−02
1POA	914	6.1E−03	2.4E−06	5.8E−02	2.4E−05	1.1E+00	2.4E−04	3.9E+00	2.3E−03	4.3E+00	1.3E−02
1AX8	1003	1.6E+00	1.3E−05	1.8E+00	1.3E−04	2.6E+00	1.3E−03	4.4E+00	1.2E−02	9.3E+00	8.9E−02
4MBA	1086	3.4E+00	1.3E−06	4.0E+00	1.3E−05	7.9E+00	1.3E−04	1.1E+01	1.3E−03	1.1E+01	1.3E−02
1F39	1534	7.5E−01	4.6E−06	5.0E+00	4.6E−05	7.4E+00	4.8E−04	7.9E+00	6.2E−03	1.8E+01	3.9E−02
1RGS	2015	1.3E+01	1.6E−05	1.2E+01	1.6E−04	1.3E+01	1.6E−03	1.7E+01	1.7E−02	1.4E+01	7.6E−01
1BPM	3672	1.5E+01	8.8E−05	1.5E+01	8.8E−04	5.5E+01	9.0E−03	2.0E+01	3.9E+02	2.3E+01	9.4E+22
1HMV	7398	2.8E+01	5.7E+24	9.9E+03	2.4E+37	3.0E+01	4.7E+46	3.0E+01	3.3E+53	2.6E+01	2.2E+76

ID—protein ID, TA—total number of atoms, RE—relative errors, LNLS—RMSD values obtained using the linear least-squares method, NLLS—RMSD values obtained using the nonlinear least-squares method

either experimental measures or theoretical estimates. In order for the algorithm to handle inexact distances (distances with errors), the general buildup procedure has to be modified. First, in every buildup step, if l distances are found from an undetermined atom to l determined atoms, $l \geq 4$, all l distances should be used for the determination of the unknown atom. The reason is that if the distances have errors, they can be inconsistent. Then the atom satisfying four of the distances may not necessarily satisfy the rest of the distances and, therefore, it should be determined with all its distance constraints. Second, if $l \geq 4$, an over-determined system of equations is obtained for the determination of the position of the unknown atom. If the distances have errors, the system may not be consistent. Therefore, we can only solve the system approximately by using for example a least-squares method. Third, a new updating scheme may be necessary to prevent the accumulation of the rounding errors. The previously developed updating scheme may not be practical any more for $l \gg 4$ because it requires all the distances available among l determined atoms.

We have developed a new geometric buildup algorithm which can prevent the accumulation of the rounding errors in the buildup calculations successfully and also tolerate small errors in the given distances. In this algorithm, we use all (instead of a subset of) the distances available for the determination of each unknown atom and obtain the position of the atom by using a least-squares approximation (instead of solving a system of equations exactly). The least-squares approximation can be implemented with either a linear or nonlinear formulation. The linear formulation can be obtained from the reduced linear system of equations for the determination of the coordinates of the unknown atom. The nonlinear formulation can be defined directly with the original system of distance equations. The linear least-squares problem can be solved using a standard method. The nonlinear least-squares problem may not be solved easily if an iterative method is used. However, we have shown that it could actually be solved by using a special singular value decomposition method, which could not only provide a good solution to the problem, but also prevent the accumulation of the rounding errors in the buildup procedure effectively. We have described these least-squares formulations and their solution methods. We have presented the test results from applying the new algorithm to the determination of a set of protein structures with varying degrees of availability and accuracy of the distances and showed that the new development increases the modeling ability of the geometric buildup approach significantly from both theoretical and practical point of views.

As we have discussed in the paper, a further complicated yet practical case of the distance geometry problem is when the distances are given with only their lower and upper bounds. The problem then becomes to find the coordinates $x_1, \ldots, x_n$ for the atoms for a given set of lower and upper bounds, $l_{i,j}$ and $u_{i,j}$, of the distances $d_{i,j}$ such that

$$l_{i,j} \leq \|x_i - x_j\| \leq u_{i,j}, \quad (i, j) \in S.$$

The algorithm presented in this paper may not be applied directly to these kinds of problems. However, its general procedure can still be adopted for the solution of such a problem. The only difference is that in every buildup step, an atom will be determined by satisfying a set of distance bounds instead of exact distances. The computation will certainly be more involved and subject to even more arbitrary errors. The solution to the problem will not be unique, either. In fact, there can be an ensemble of solutions all satisfying the given distance inequalities. On the other hand, in practice, it is actually preferred

to obtain the entire ensemble of solutions instead of a few samples. How to implement a buildup algorithm for the solution of such a problem can be challenging and will be the topic of our next step of investigation.

Acknowledgements

The authors would like to thank Vlad Sukhoy for helpful discussions and suggestions. They would also like to acknowledge the support from the Department of Mathematics and the Baker Center for Bioinformatics and Biological Statistics, Iowa State University and the Laboratory for Scientific and Engineering Computing, Chinese Academy of Sciences. The work is funded partially by the NIH/NIGMS grant R01GM081680 and by the NSF of China. The authors would also like to thank the anonymous referees for reading the paper and providing helpful suggestions.

References

Biswas, P., Liang, T., Wang, T., Ye, Y., 2006. Semidefinite programming based algorithms for sensor network localization. ACM J. Trans. Sensor Netw. 2, 188–220.

Biswas, P., Liang, T., Toh, K., Ye, Y., 2007. A SDP based approach to anchor-free 3D graph realization. Department of Management Science and Engineering, Electrical Engineering, Stanford University, Stanford, California.

Blumenthal, L.M., 1953. Theory and Applications of Distance Geometry. Clarendon, Oxford.

Crippen, G.M., Havel, T.F., 1988. Distance Geometry and Molecular Conformation. Wiley, New York.

Dong, Q., Wu, Z., 2002. A linear-time algorithm for solving the molecular distance geometry problem with exact inter-atomic distances. J. Global Optim. 22, 365–375.

Dong, Q., Wu, Z., 2003. A geometric buildup algorithm for solving the molecular distance geometry problem with sparse distance data. J. Global Optim. 26, 321–333.

Glunt, W., Hayden, T.L., Hong, S., Wells, J., 1990. An alternating projection algorithm for computing the nearest Euclidean distance matrix. SIAM J. Matrix Anal. Appl. 11, 589–600.

Glunt, W., Hayden, T.L., Raydan, M., 1993. Molecular conformations from distance matrices. J. Comput. Chem. 14, 114–120.

Golub, G.H., van Loan, C.F., 1989. Matrix Computations. Johns Hopkins Press, Baltimore.

Grosso, A., Locatelli, M., Schoen, F., 2007. Solving molecular distance geometry problems by global optimization algorithms. J. Comput. Opt. Appl. 43, 22–37.

Havel, T., 1991. An evaluation of computational strategies for use in the determination of protein structure from distance constraints obtained by nuclear magnetic resonance. Prog. Biophys. Molec. Biol. 56, 43–78.

Havel, T.F., 1995. Distance geometry. In: Grant, D.M., Harris, R.K. (Eds.), Encyclopedia of Nuclear Magnetic Resonance, pp. 1701–1710. Wiley, New York.

Havel, T.F., 1998. Distance geometry: Theory, algorithms, and chemical applications. In: Encyclopedia of Computational Chemistry, pp. 1–20. Wiley, New York.

Hendrickson, B., 1992. Conditions for unique graph realizations. SIAM J. Comput. 21, 65–84.

Hendrickson, B., 1995. The molecule problem: Exploiting structure in global optimization. SIAM J. Optim. 5, 835–857.

Hou, J.T., Sims, G.E., Zhang, C., Kim, S.H., 2003. A global representation of the protein fold space. Proc. Natl. Acad. Sci. USA 100, 2386–2390.

Huang, H.X., Liang, Z.A., Pardalos, P., 2003. Some properties for the Euclidean distance matrix and positive semi-definite matrix completion problems. J. Global Optim. 25, 3–21.

Kearsly, A., Tapia, R., Trosset, M., 1998. Solution of the metric STRESS and SSTRESS problems in multidimensional scaling by Newton's method. Comput. Stat. 13, 369–396.

Klock, H., Buhmann, J.M., 1997. Multidimensional scaling with deterministic annealing. In: Pilillo, M., Hancock, E.R. (Eds.), Energy Minimization Methods in Computer Vision and Pattern Recognition, Lecture Notes in Computer Science, vol. 1223, pp. 246–260. Springer, Berlin.

Le Thi Hoai, A., Pham Dinh, T., 2003. Large scale molecular optimization from distance matrices by a d.c. optimization approach. SIAM J. Optim. 4, 77–116.

Moré, J., Wu, Z., 1996. ε-Optimal solutions to distance geometry problems via global continuation. In: Pardalos, P.M., Shalloway, D., Xue, G. (Eds.), Global Minimization of Non-Convex Energy Functions: Molecular Conformation and Protein Folding, pp. 151–168. Am. Math. Soc., Providence.

Moré, J., Wu, Z., 1997. Global continuation for distance geometry problems. SIAM J. Optim. 7, 814–836.

Moré, J., Wu, Z., 1999. Distance geometry optimization for protein structures. J. Global Optim. 15, 219–234.

Saxe, J.B., 1979. Embeddability of weighted graphs in k-space is strongly NP-hard. In: Proc. 17th Allerton Conference in Communications, Control and Computing, pp. 480–489.

Sippl, M., Scheraga, H., 1985. Solution of the embedding problem and decomposition of symmetric matrices. Proc. Natl. Acad. Sci. USA 82, 2197–2201.

Sippl, M., Scheraga, H., 1986. Cayley-Menger coordinates. Proc. Natl. Acad. Sci. USA 83, 2283–2287.

Torgerson, W.S., 1958. Theory and Method of Scaling. Wiley, New York.

Trosset, M., 1998. Applications of multidimensional scaling to molecular conformation. Comput. Sci. Stat. 29, 148–152.

Wu, D., Wu, Z., 2007. An updated geometric buildup algorithm for solving the molecular distance geometry problem with sparse distance data. J. Global Optim. 37, 661–673.

Zou, Z., Byrd, R.H., Schnabel, R.B., 1997. A stochastic/perturbation global optimization algorithm for distance geometry problems. J. Global Optim. 11, 91–105.