2010

# Solving distance geometry problems for protein structure determination

Atilla Sit
*Iowa State University*

**Solving distance geometry problems for protein structure determination**

by

Atilla Sit

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Applied Mathematics

Program of Study Committee:
Zhijun Wu, Major Professor
Anastasios Matzavinos
Paul E. Sacks
Robert L. Jernigan
Guang Song
Peng Liu

Iowa State University

Ames, Iowa

2010

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere thanks and gratitude to my major professor, Dr. Zhijun Wu, for his guidance and endless support throughout this research and the writing of this thesis. It would be impossible to finish this work without the utmost understanding and patience he has demonstrated during the last three years. I would like to thank Dr. Robert Jernigan for his collaboration and valuable contributions to this work. I would also like to thank my other committee members: Dr. Paul Sacks, Dr. Guang Song, and Dr. Anastasios Matzavinos for their efforts and contributions to this work, and Dr. Peng Liu for being my minor representative for statistics. I also wish to thank the Department of Mathematics at Iowa State University for providing me the financial support throughout my PhD study. All my friends, colleagues and research group members, who have ever assisted me, are gratefully appreciated as well.

# ABSTRACT

A well-known problem in protein modeling is the determination of the structure of a protein with a given set of interatomic distances obtained from either physical experiments or theoretical estimates. A more general form of this problem is known as the distance geometry problem in mathematics, which can be solved in polynomial time if a complete set of exact distances is given, but is generally intractable for a general sparse set of distance data. We investigate the solution of the problem within a geometric buildup framework. We propose a new geometric buildup algorithm for solving the problem using special least-squares approximation techniques, which not only prevents the accumulation of the rounding errors in the buildup calculations successfully, but also tolerates small errors in given distances. In NMR spectroscopy, however, distances can only be obtained with their rough ranges, and hence an ensemble of solutions satisfying the given constraints becomes critical to find. We propose a new approach to the problem of determining an ensemble of protein structures with a set of interatomic distance bounds. Similar to X-ray crystallography, we assume that the protein has an equilibrium structure and the atoms fluctuate around their equilibrium positions. Then, the problem can be formulated as a generalized distance geometry problem to find the equilibrium positions and maximal possible fluctuation radii for the atoms in the protein, subject to the condition that the fluctuations should be within the given distance bounds. We describe the scientific background of the work, the motivation of the new approach and the formulation of the problem. We develop a geometric buildup algorithm for an approximate solution to the problem and present some preliminary test results. We also discuss related theoretical and computational issues and potential impacts of this work in NMR protein modeling.

# CHAPTER 1. INTRODUCTION

Proteins are an important class of biomolecules. They are key for biological systems to have certain functions, and most biological studies end up with studies on certain proteins. In order to understand proteins and their functions, it is necessary and critical to find their three-dimensional geometric structure. There are two major techniques for protein structure determination: One is X-ray crystallography [18] and the other one is nuclear magnetic resonance (NMR) spectroscopy [7]. In either case, a set of experimental data is collected and a mathematical problem needs to be solved to form the structure [55, 71].

In NMR spectroscopy, distances between certain pairs of atoms in a given protein can be detected. The related mathematical problem then to be solved is to find the coordinates of the atoms given a set of interatomic distances. A more general and abstract form of this problem is known as the distance geometry problem in mathematics [5]. It has other names in the literature as well, such as the graph embedding problem in computer science [54], the multidimensional scaling problem in statistics [65], and the graph realization problem in graph theory [32]. In general, the problem can be stated as to find the coordinates for a set of points in some topological space given the distances between 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 fields as well, such as sensor network localization [3], image recognition [39], and protein classification [36], to name a few. In any case, the problem may or may not have a solution depending on the given distance data or the space where the solution is to be found. Even though it has a solution, the solution may not be unique, or may not be easy to find depending on the given distances. These properties carry great theoretical and practical importance, but have not been well understood [71].

Throughout the thesis, we will consider the problem only in Euclidean space, in particular the 3D Euclidean space, where the problem for molecular modeling is defined. Our main motivation for studying the solution of the distance geometry problem is protein structure determination, so we will generally use the word *atom* to refer to points whose 3D coordinates are to be determined.

Suppose that we have a protein of $n$ atoms. Let $\{x_i : i = 1, \ldots, n\}$ be the set of coordinate vectors of these atoms, namely $x_i = (x_{i,1}, x_{i,2}, x_{i,3})^T$, where $x_{i,1}$, $x_{i,2}$, and $x_{i,3}$ are the first, second, and third coordinates of atom $i$, respectively. Let $\|\cdot\|$ be the Euclidean norm. If the coordinates $x_i$, $i = 1, \ldots, n$ are known, the distances $d_{i,j}$ between atoms $i$ and $j$ can easily be computed with $d_{i,j} = \|x_i - x_j\|$. Conversely, if the distances $d_{i,j}$ are given, the coordinates $x_1, \ldots, x_n$ for the atoms can also be obtained based on the distances $d_{i,j}$, but the computation is not as straightforward. The solution of a system of equations as can be stated in the following for $x_1, \ldots, x_n$ is required.

$$\|x_i - x_j\| = d_{i,j} \quad \text{for} \quad (i,j) \in S, \tag{1.1}$$

where $S$ is a subset of all atom pairs. The latter problem is called as the *distance geometry problem*. In practice, however, the distances come from physical experiments or theoretical estimates, hence may have errors. Therefore, a more general yet practical form of the problem would be to find the coordinates of the atoms $x_1, \ldots, x_n$ given only a set of lower and upper bounds, $l_{i,j}$ and $u_{i,j}$, of the distances $d_{i,j}$ such that

$$l_{i,j} \leq \|x_i - x_j\| \leq u_{i,j} \quad \text{for} \quad (i,j) \in S. \tag{1.2}$$

The distance geometry problem can be solved in polynomial time if the distances for all pairs of atoms are available [27]. However, it has been proved to be NP-hard in general [54]. Even if errors are allowed for the distances, the problem is still hard, when only small errors are allowed [47]. In practice, this means that it is unlikely to find a general algorithm which solves all instances of the problem efficiently. However, these facts have not discouraged scientists from searching and developing new algorithms for solving the distance geometry problem, because the theory related to NP-hardness of the problem given in [54] and [47] was based on very special graphs and distances, which are highly unlikely to occur in practical problems.

The existing approaches to the solution of the problem and their recent developments include, but not limited to, the embedding algorithm by Crippen and Havel [11, 28], the alternating projection method by Glunt and Hayden [22, 23], the graph reduction approach by Hendrickson [32, 33], the global optimization method by Moré and Wu [48, 49], the stochastic/perturbation method by Zou, Byrd, and Schnabel [76], the multidimensional scaling method by Kearsly, Tapia, and Trosset [38, 67], the dc programming method by Le Thi Hoai and Pham Dinh [42, 43], the semi-definite programming approach by Biswas, Liang, Toh, and Ye [4], the stochastic search method by Grosso, Locatelli, and Schoen [26], and the geometric buildup algorithm by Dong, Wu, and Wu [15, 16, 68].

The thesis is organized as follows. We provide a brief account on the biological background of this study in Chapter 2, and give a brief introduction to protein modeling. We describe the key steps of X-ray crystallography and NMR spectroscopy, and show how the protein structures are formed and represented differently in the two different approaches. We also mention briefly about NOE distance restraints coming from the NOESY experiment in NMR as well as their importance in protein modeling, and we formulate the fundamental problem arising in NMR spectroscopy.

In Chapter 3, we give an extensive introduction to the distance geometry problem, and examine the theory related to the problem under three categories with regards to the sparsity of the distance data given for the problem. For each of these categories, we discuss about the solution methods and the issues related to the computational complexity of the problem, and we provide some theoretical results. We close Chapter 3 by giving a brief description on the historical development of distance geometry, and reviewing some of the existing approaches to the solution of the distance geometry problem listed above.

In Chapter 4, we present a comprehensive review of the geometric buildup approach to the solution of the distance geometry problem with exact distances. We introduce three different geometric buildup algorithms proposed for solving the distance geometry problem; the general geometric buildup algorithm by Dong and Wu [15, 16], an updated algorithm by Wu and Wu [68], and a rigid geometric buildup algorithm by Wu, Wu, and Yuan [69].

In Chapter 5, we propose a new extended geometric buildup algorithm [60] for solving the distance geometry problem using least-squares approximations, which not only prevents the accumulation of the rounding errors in the buildup calculations successfully, but also tolerates small errors in given distances. 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 of the geometric buildup approach significantly from both theoretical and practical points of view.

The algorithm proposed in Chapter 5 works for solving the distance geometry problem with exact distances given in (1.1). However, in practice, for example in NMR experiments, the distances are not given in their exact values; only their rough ranges such as lower and upper bounds can be obtained. Then, the distance geometry problem with distance bounds given in (1.2) needs to be solved. The problem (1.2) is different from (1.1), and should be treated carefully, because it usually requires determining an ensemble of solution structures, all satisfying the given distance constraints.

In Chapter 6, we propose a new approach [61] to the problem of determining an ensemble of protein structures with a set of interatomic distance bounds in NMR protein modeling. Similar to X-ray crystallography, we assume that the protein has an equilibrium structure and the atoms fluctuate around their equilibrium positions. Then, the problem can be formulated as a generalized distance geometry problem, to find the equilibrium positions and maximal possible fluctuation radii for the atoms in the protein, subject to the condition that the fluctuations should be within the given distance bounds. We describe the scientific background of the work, the motivation of the new approach and the formulation of the problem. We develop a geometric buildup algorithm for an approximate solution to the problem and present some preliminary test results. We also discuss related theoretical and computational issues and potential impacts of this work in NMR protein modeling.

In Chapter 7, we summarize the entire thesis work. We describe our current work in

progress, and show some initial results from applying a modified version of the algorithm given in Chapter 6 to real a distance data coming from NMR experiments. We finish the thesis by discussing some important issues for future investigation.

# CHAPTER 2. BIOLOGICAL BACKGROUND

In this chapter, we give a brief introduction to protein modeling. We describe the key steps of X-ray crystallography and nuclear magnetic resonance (NMR) spectroscopy, and provide the details of how the structures are formed and represented differently in the two different approaches. We also mention briefly about NOE distance restraints coming from the NOESY experiment in NMR as well as their importance in protein modeling. Finally, we formulate the fundamental problem arising in NMR spectroscopy.

## 2.1 Protein Structure Determination

Proteins are an important class of biomolecules. They are encoded in genes and expressed in cells via genetic translation. Proteins are life supporting (or sometimes, destructing) ingredients and are indispensable for almost all biological processes [1, 10]. With the completion of the genomic sequencing of human and many other species, studies on proteins, the end products of gene expression, have become more important ever for the interpretation of the genes and their implications to life. However, to understand proteins and their functions, it is essential to know their three-dimensional structures, which, due to various technical reasons, are very difficult to determine [1, 10].

There is no direct physical means to observe the structure of a protein at a desired resolution, for example, at the residue level. Several experimental approaches have been used to obtain some indirect structural data upon which the structures may be deduced. For example, the diffraction data for a protein crystal can be obtained by X-ray crystallography and used to find the electron density distribution and hence the structure of the protein [18]; the magnetic resonance spectra of the nuclear spins in a protein can be detected by NMR experiments and

used to estimate the distances between certain pairs of atoms and subsequently, the coordinates of the atoms in the protein [7].

The experimental approaches have various limitations. For example, X-ray crystallography requires crystallizing the protein, which is time-consuming and often fails. To obtain accurate enough signals, NMR experiments can only be carried out for small proteins with less than a few hundred residues. Therefore, the number of structures that can be determined by these experimental approaches has been far from adequate for the increasing demands for structural information on the hundreds of thousands of proteins of biological and medical importance.

Surveys on the protein structures deposited into the PDB Data Bank [2] show that 80% of the structures are determined by X-ray crystallography, 15% by NMR, and 5% by other approaches. These structures, about several tens of thousands in total, contain a high percentage of replications (structures for the same protein determined with different techniques or under different conditions). Some structures are also very similar because there are only small mutations among them. Without counting the replications and genetically highly related structures, there may be only around several thousands of different proteins whose structures have been determined. However, there are at least several hundreds of thousands of different proteins in the human body alone. Most of their structures are still unknown [9, 53].

In X-ray crystallography, scientists obtain an electron density map for a protein crystal based on the crystal's diffraction data (by solving a so-called phase problem [18]) (see Fig. 2.1). They then assign the atoms of the protein to certain positions in the map according to the shapes and densities of the electron clouds around these positions. After further refinement, the structure of the protein is determined and documented in a structural file. Within the structural file, the atoms in the protein are listed in certain order, and their coordinates are recorded. In addition, there is a value called B-factor (or temperature factor) determined and assigned for each atom (see Fig. 2.2). Let $< r, r >$ be the mean-square fluctuation of an atom. Then the B-factor for this atom is defined to be $8\pi^2 < r, r >$ [18]. Therefore, the B-factor is an important indicator for how the atom fluctuates around its equilibrium position due to its special structural or physical condition. An atom with a high B-factor or a region with a high

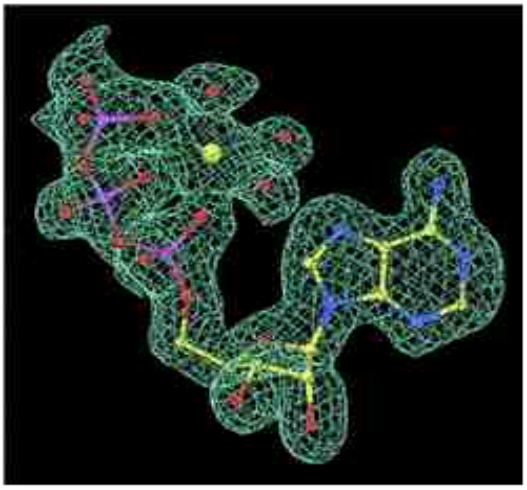


Figure 2.1: **Electron density map for a protein crystal:** The protein crystal diffraction data can be used to derive an electron density map for the protein. The atoms in the protein are then assigned with some positions in the map according to the shapes and densities of the electron clouds around the positions.

```
......

ATOM   11  CA  VAL A 121      12.250  12.310  23.376  1.00 50.46
ATOM   12  C   VAL A 121      12.675  12.621  24.805  1.00 49.93
ATOM   13  O   VAL A 121      11.901  12.429  25.744  1.00 51.04
ATOM   14  CB  VAL A 121      11.228  13.356  22.927  1.00 50.42
ATOM   15  CG1 VAL A 121      11.818  14.742  23.053  1.00 51.66
ATOM   16  CG2 VAL A 121      10.809  13.081  21.495  1.00 51.92
ATOM   17  N   VAL A 122      13.908  13.092  24.965  1.00 46.72
ATOM   18  CA  VAL A 122      14.423  13.438  26.280  1.00 43.86
ATOM   19  C   VAL A 122      13.614  14.629  26.797  1.00 41.49
ATOM   20  O   VAL A 122      13.104  15.429  26.010  1.00 39.42

......
```

Figure 2.2: **PDB file for an X-ray structure:** The atoms in the protein are listed in certain order, and their coordinates are recorded. In addition, there is a value called B-factor (or temperature factor) determined and assigned for each atom.

average B-factor is called a hot spot of the protein. It may correspond to an active functional site of the protein [18].

The advantage of using NMR is that the protein does not need to be crystalized (which is sometimes impossible) and the structure can be determined in solution. NMR can also be used to determine dynamic properties of proteins such as the flexibilities of the protein backbone or sidechains in solution [7]. The most common types of conformational distance constraints that can be obtained from NMR include the distances between hydrogen atoms estimated via Nuclear Overhauser Effects (NOE) and the dihedral angles around certain bonds through J-coupling [7].

The NOE intensity for two magnetically interacting hydrogen atoms is inversely proportional to the sixth power of the distance between the atoms and can therefore be detected only between atoms at very short distances ($< 5$ Å). The NOE can be reduced by other interactions such as spin diffusion, and therefore it can provide only a rough upper bound for a measured

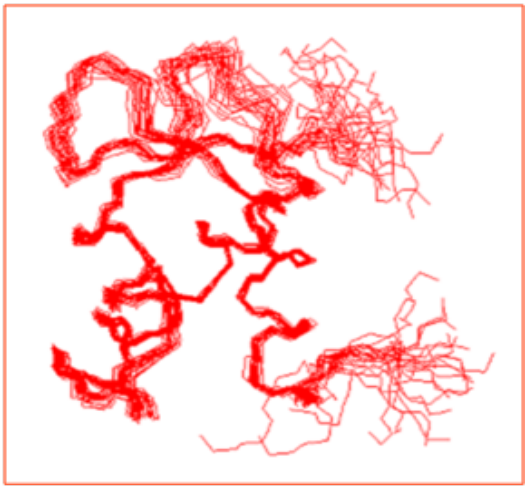


Figure 2.3: **NMR structural ensemble:** Given a set of lower and upper bounds of interatomic distances obtained from NMR experiments, an ensemble of structures, instead of a single one, can be determined for a protein. They can be aligned together to show the structure and variations.

```
MODEL1:

ATOM    1  N   LEU A 125     10.104 -0.797 -1.434  1.00  0.00
ATOM    2  CA  LEU A 125     10.516 -0.563 -0.027  1.00  0.00
ATOM    3  C   LEU A 125     11.782  0.287  0.043  1.00  0.00
------

MODEL2:

ATOM    1  N   LEU A 125      9.930 -0.836 -0.832  1.00  0.00
ATOM    2  CA  LEU A 125     10.500 -0.808  0.540  1.00  0.00
ATOM    3  C   LEU A 125     11.767  0.041  0.591  1.00  0.00
------

MODEL3:

ATOM    1  N   LEU A 125     10.236 -0.617 -1.300  1.00  0.00
ATOM    2  CA  LEU A 125     10.681 -0.544  0.117  1.00  0.00
ATOM    3  C   LEU A 125     11.938  0.308  0.257  1.00  0.00
------
```

Figure 2.4: **PDB file for an NMR structure:** Around 20 to 100 structural models are documented to represent the whole structural ensemble. There are not B-factors for atoms, but the atomic fluctuations can be estimated based on the average atomic positions and their root-mean-square deviations from those positions in the structural ensembles.

distance. The lower bound for the distance can be determined for example by using the van der Waals radii of the atoms. Given such a set of lower and upper bounds of the distances, an ensemble of structures, instead of a single one, can be determined by NMR [11, 72]. Usually, about 20 to 100 structures are determined to represent the whole structural ensemble. They are aligned together to show the overall structure and fluctuation (see Fig. 2.3). As such, the structural files for NMR determined structures typically contain multiple models as shown in Fig. 2.4. There are not B-factors for atoms, but the atomic fluctuations can be estimated based on the average atomic positions and their mean-square deviations from those positions in the structural ensembles [11, 72].

## 2.2 NOE Distance Restraints

The most important geometric information that is available from NMR spectroscopy comes in the form of contacts between pairs of hydrogen atoms, i.e. upper bounds on their distances. This information is obtained from a two-dimensional spectrum called NOESY (nuclear Overhauser effect spectroscopy), whose diagonal corresponds to the usual 1D spectrum, and whose cross-peaks occur at the frequency coordinates of spatially proximal pairs of protons [29].

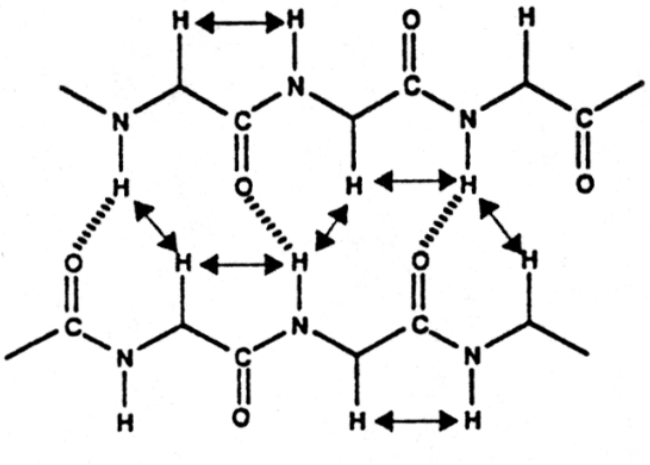


Figure 2.5: **Nuclear Overhauser effect:** When two protons are close in space, the resonance frequency of one proton will be affected by the presence of the other proton. The change in frequency is called the nuclear Overhauser effect and can be measured through NOESY experiment.

NOEs are the essential NMR data for defining the secondary and tertiary structures of a protein because they permit connection of pairs of hydrogen atoms in amino acid residues "through space" that may be far apart in the protein sequence, but close in space (less than about 5 Å apart, see Fig. 2.5). The NOE arises from the transfer of magnetization between spins coupled by the dipole-dipole interaction in a molecule undergoing Brownian motion in a liquid. The intensity of an NOE, i.e. the volume of the corresponding cross peak in a NOESY spectrum, is inversely proportional to the sixth power of the distance between two interacting $^1$H spins. Thus, if one interproton distance, $r_{\mathrm{ref}}$, is known (e.g. from covalent geometry), then another, unknown interproton distance, $r_i$, is determined by the relationship (ignoring internal mobility),

$$r_i = r_{\mathrm{ref}} \, (S_{\mathrm{ref}}/S_i)^{1/6}, \tag{2.1}$$

in which $S_{\mathrm{ref}}$ and $S_i$ are the cross-peak intensities. This way, NOE intensities are translated to

distance ranges. The lower bound is determined from the sum of the van der Waals' radii and the upper bound from the NOE intensity. NOEs are usually translated into upper bounds on interatomic distances rather than precise distance restraints because the presence of internal motions, spin diffusion and, possibly, chemical exchange may affect the intensity of an NOE. Since precise $^1$H-$^1$H separations cannot be determined from NOE intensities, NOE cross-peaks typically are grouped on the basis of their intensities into three categories, for example 2.7 Å (strong), 3.3 Å (medium), and 5.0 Å (weak). This calibration usually yields good results provided that there is a large number of restraints. However, if greater accuracy is required, for example when ligand-binding sites are being studied, a means of obtaining tighter distance restraints from NOE peak intensities becomes necessary [7, 44].

## 2.3 The Fundamental Problem

The determination of the coordinates of the atoms in the molecule once a set of distance data becomes available from the NMR experiments is essential to the NMR techniques for structure determination. Even when all the distances can be unambiguously determined, the problem of computing the conformation of biological macromolecules from the NMR data remains difficult for two reasons. One is the size of the molecules involved, which often exceeds 1000 atoms. The second lies in the sparsity of the distances, usually covering less than 1 % of the million or more different distances in such large molecules. This is a consequence of the fact that distances are short with lengths usually less than or equal to 5 Å (due to the weak NOE signals that can be detected only for nearby nuclei), and they are available only for pairs of hydrogen atoms a short distance apart. Such a set of data contains important structural information, but it is not sufficient for the complete determination of the structure. Fortunately, from the covalent geometry, certain bond lengths and bond angles can be obtained and an additional distance set can be formed. By combining the two sets of data, the coordinates of the atoms can then be determined, upon which a model for the molecule can be constructed [29, 71].

Let $x_i = (x_{i,1}, x_{i,2}, x_{i,3})^T$ be the coordinate vector for atom $i$, $i = 1, \ldots, n$. Let $\| \cdot \|$ be the

Euclidean norm. The fundamental problem then becomes to find $x_i$, $i = 1, \ldots, n$ such that

$$l_{i,j} \leq \|x_i - x_j\| \leq u_{i,j} \quad \text{for} \quad (i,j) \in S, \tag{2.2}$$

where $l_{i,j}$ and $u_{i,j}$ are the lower and upper bounds for the distances between the atoms $i$ and $j$, for $i, j = 1, \ldots, n$. The problem is called the *distance geometry problem* (see Fig. 2.6) which we will discuss further in Chapter 3.

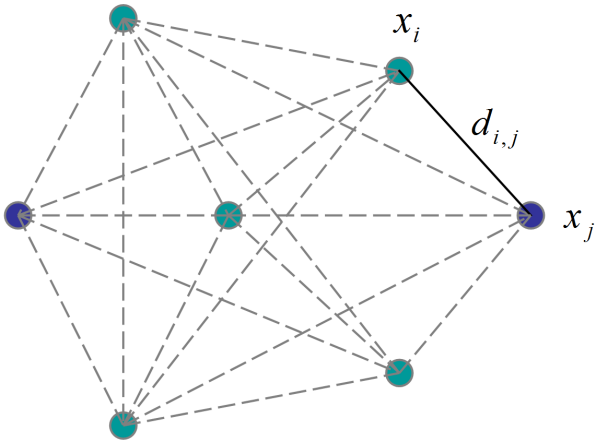



Figure 2.6: **The fundamental problem:** The problem is to find coordinates of the atoms in a molecule given a set of lower and upper bounds for distances.

This problem may have infinitely many possible solutions, corresponding to an ensemble of structures all satisfying the given distance constraints. In NMR, it turns out to be important to not just find one of these structures but the whole ensemble of structures, because the deviations of the structures from each other in the ensemble provide important information on how the protein structure may fluctuate dynamically around its equilibrium state. This dynamic property is often as critical as the structure itself for the understanding of the function of the protein [11, 72].

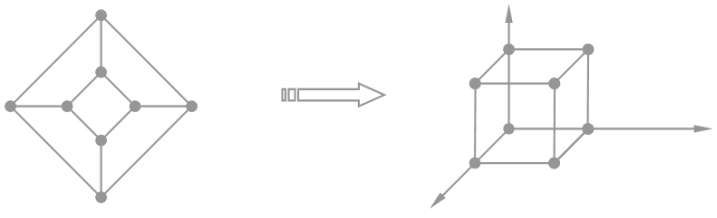
# CHAPTER 3. DISTANCE GEOMETRY PROBLEM

In this chapter, we study the distance geometry problem, which is a well-known problem in protein modeling. We give a short introduction, the definition and background of the problem. We examine the problem under three categories, depending on the sparsity of the given distance data. For each of these categories, we discuss about the solution methods and the issues related to the computational complexity of the problem, and we provide some theoretical results. After giving a brief description on the historical development of distance geometry, we review some of the existing approaches to the solution of the distance geometry problem.

## 3.1 Introduction

The problem of determining the coordinates of the atoms in a molecule given the distances between certain pairs of atoms can be studied in a general mathematical form, where the atoms can be placed in any metric space with the distances defined in terms of a general metric associated with the space (see Fig. 3.1). The problem may or may not have a solution, depending on the given distance data and the space where the solution is to be found. Even if it does have a solution, it may still be nonunique, or the solution may not be easy to

Figure 3.1: Distance geometry problem



Given $n$ atoms $a_1, a_2, \ldots, a_n$ and a set of distances $d_{i,j}$ between $a_i$ and $a_j$, find the positions $x_1, x_2, \ldots, x_n$ for $a_1, a_2, \ldots, a_n$ such that $\|x_i - x_j\| = d_{i,j}$.

find, depending on the given distances. These properties carry great theoretical and practical importance, but have not been well understood. We will consider the problem only in Euclidean space, in particular the 3D Euclidean space, where the problem for molecular modeling is defined.

## 3.2 The Distance Geometry Problem

Let $n$ be the number of atoms in a given protein and $x_1, \ldots, x_n$ be the coordinate vectors for the atoms, 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 $i$, $i = 1, \ldots, n$. Let $\|\cdot\|$ be the Euclidean norm. If the coordinates $x_1, \ldots, x_n$ are known, the distances $d_{i,j}$ between atoms $i$ and $j$ can be computed with $d_{i,j} = \|x_i - x_j\|$. Conversely, if the distances $d_{i,j}$ are given, the coordinates $x_1, \ldots, x_n$ for the atoms can also be obtained based on the distances $d_{i,j}$, but the computation is not as straightforward. The solution of a system of equations as can stated in the following for $x_1, \ldots, x_n$ is required.

$$\|x_i - x_j\| = d_{i,j} \quad \text{for} \quad (i,j) \in S, \tag{3.1}$$

where $S$ is a subset of all atom pairs. The latter problem is called as the *distance geometry problem* in mathematics [5]. It has different names in the literature, such as the graph embedding problem in computer science [54], the multidimensional scaling problem in statistics [65], and the graph realization problem in graph theory [32]. 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 in three-dimensional Euclidean space, the problem has applications in many other scientific and engineering fields as well, such as sensor network localization [3], image recognition [39], and protein classification [36], to name a few. In practice, the distances come from physical experiments or theoretical estimates, hence may have errors. Therefore, a more general yet practical form of the problem would be to find the coordinates of the atoms $x_1, \ldots, x_n$ given only a set of lower and upper bounds, $l_{i,j}$ and $u_{i,j}$, of the distances $d_{i,j}$ such that

$$l_{i,j} \leq \|x_i - x_j\| \leq u_{i,j} \quad \text{for} \quad (i,j) \in S. \tag{3.2}$$

The distance geometry problem is polynomial time solvable if the distances for all pairs of atoms are available [27]. However, it has been proved to be NP-hard in general [54]. Even if errors are allowed for the distances, the problem is still hard, if only small errors are allowed [47]. The set $S$ in (3.1) and (3.2) may not necessarily contain all possible $(i, j)$ pairs. We say that the problem has a sparse set of distances, or sparse distance data, if $S$ has only a subset of all $(i, j)$ pairs; otherwise, we say that it has a complete set of distances or dense distance data. The distances may not be provided as exact values and, in many cases, may be given in estimated ranges. When the exact distances are provided, we say that the problem has exact distances; otherwise, it has inexact distances or distance ranges or bounds. In the latter case, the solution is generally not unique, and there may in fact exist a set of solutions that may all be of interest in practice. Depending on the sparsity and type of distance data, the distance geometry problem can be examined in three different cases: Problem with exact distances, sparse distances, and distance bounds.

### 3.2.1 Exact Distances

We first consider the simple case when a complete set of exact distances is given. The distance geometry problem can then be solved in polynomial time. A solution with such a set of distance data can be obtained efficiently by using singular value decomposition (SVD) of an induced distance matrix.

Assume that a set of coordinates $x_1, \ldots, x_n$ can be found for a given set of distances $d_{i,j}$, where $i, j = 1, \ldots, n$. Then, $\|x_i - x_j\| = d_{i,j}$ for all $i, j = 1, \ldots, n$, and

$$\|x_i\|^2 - 2x_i^T x_j + \|x_j\|^2 = d_{i,j}^2, \quad i, j = 1, \ldots, n. \tag{3.3}$$

Since the molecular structure is invariant under any translation and rotation, we set a reference system so that the origin is located at the last atom or in other words, $x_n = (0, 0, 0)^T$. It follows that

$$d_{i,n}^2 - 2x_i^T x_j + d_{j,n}^2 = d_{i,j}^2, \quad i, j = 1, \ldots, n-1. \tag{3.4}$$

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

$$\begin{aligned} X &= \{x_{i,j} : i = 1, \ldots, n-1, j = 1, 2, 3\} \text{ and} \\ D &= \{(d_{i,n}^2 - d_{i,j}^2 + d_{j,n}^2)/2 : i, j = 1, \ldots, n-1\}. \end{aligned} \tag{3.5}$$

Then, $XX^T = D$ and $D$ must be of maximum rank 3.

The distance geometry problem can be defined in a general space $R^k$ with $x_1, \ldots, x_k$ in $R^k$ and $d_{i,j}$ the Euclidean distances between atoms $i$ and $j$. Then, the equation $XX^T = D$ still holds, and $D$ must be of maximum rank $k$, where $X = \{x_{i,j} : i = 1, \ldots, n, j = 1, \ldots, k\}$.

**Theorem 3.1.** *Let $\{d_{i,j} : i, j = 1, \ldots, n\}$ be a set of distances in $R^k$, for some $k \leq n$. Then, the induced matrix defined in (3.5) is of maximum rank $k$.*

*Proof.* It follows from the fact that $D = XX^T$ for a coordinate matrix $X$ in $R^{n-1} \times R^k$ and $X$ is of maximum rank $k$. □

The equation $XX^T = D$ can be solved using the singular value decomposition of $D$. Let $D = U\Sigma U^T$ be the singular value value decomposition of $D$, where $U$ is an orthogonal matrix and $\Sigma$ a diagonal matrix with singular values of $D$ along the diagonal. If $D$ is a matrix of rank less than or equal to $k$, the decomposition can be obtained with $U$ being $(n-1) \times k$ and $\Sigma$ being $k \times k$. Then, $X = U\Sigma^{1/2}$ solves the equation $XX^T = D$. Here the singular value decomposition of $D$ requires $O(kn^2)$ floating-point operations [24], and therefore, the distance geometry problem with a complete set of exact distances can be solved in polynomial time.

Note that although in practice, the distances may not be available for all the pairs of atoms, the solution of the problem with all exact distances can still be important for the solution of the general problem a sparse set of distances. For example, in the embed algorithm, a complete set of distances among all the atoms is generated after bound smoothing, and the solution of the distance geometry problem with all exact distances is always required afterwards [11, 27]. Also, if a subset of atoms has all the distances among the atoms, but the whole set of atoms does not, the coordinates of the subset of atoms can still be determined efficiently by solving a distance geometry problem with all exact distances for the subset of atoms. The procedure

may also be applied repeatedly as some of the atoms are determined and availability of the distances among them is changed, until no such subsets of atoms can be found [58, 59].

### 3.2.2 Sparse Distances

We now consider the problem with an incomplete set of exact distances. Let $S$ be a subset of all pairs of atoms such that $(i, j)$ is in $S$ if the distance $d_{i,j}$ between atoms $i$ and $j$ is given. Then, the problem is to find the coordinates $x_1, \ldots, x_n$ for the atoms so that

$$\|x_i - x_j\| = d_{i,j} \quad \text{for} \quad (i, j) \in S. \tag{3.6}$$

In computer science terminology, the distance geometry problem with sparse distance data can be proven to be NP-hard [54]. One can use a 1D version of the problem to demonstrate this property [71].

**Definition 3.1.** *Suppose that we have a molecule of $n$ atoms. Let $x_j$, $j = 1, \ldots, n$ be a set of positions on a real line, where $x_j$ is the position for atom $j$. Let $S$ be a set of index pairs $(i, j)$, with each corresponding to a given distance $d_{i,j}$ between atoms $i$ and $j$. Then, a 1D distance geometry problem is to determine the positions $x_j$ of the atoms in the molecule on a real line so that*

$$|x_i - x_j| = d_{i,j} \quad \textit{for} \quad (i, j) \in S. \tag{3.7}$$

**Definition 3.2.** *Let $A = \{s_1, \ldots, s_n\}$ be a set of positive integers. An integer set partition problem is to find two subsets of $A$ so that the sum of the integers in the subsets are equal, i.e.*

$$\sum_{j \in S_1} s_j = \sum_{j \in S_2} s_j, \tag{3.8}$$

*where $S_1$ and $S_2$ are the index sets of the integers in the first and second subsets, respectively.*

**Theorem 3.2.** *An integer set partition problem can be reduced to a 1D distance geometry problem.*

*Proof.* Construct the following 1D distance geometry problem for $n + 1$ atoms with a set of distances as given below:

$$d_{j,j+1} = s_j, \ j = 1, \ldots, n \quad \text{and} \quad d_{1,n+1} = 0. \tag{3.9}$$

If the distance geometry problem (3.9) has a solution, then the constraint $d_{1,n+1} = 0$ implies that $x_{n+1} = x_1$. Thus,

$$\sum_{j=1}^{n}(x_{j+1} - x_j) = x_{n+1} - x_1 = 0. \tag{3.10}$$

From (3.9), we have $|x_{j+1} - x_j| = s_j$ which means that $x_{j+1} - x_j$ is equal to either $s_j$ or $-s_j$. Let $S_1 = \{j : x_{j+1} - x_j = s_j\}$ and $S_2 = \{j : x_{j+1} - x_j = -s_j\}$. Then,

$$\sum_{j\in S_1} s_j - \sum_{j\in S_2} s_j = \sum_{j\in S_1}(x_{j+1} - x_j) + \sum_{j\in S_2}(x_{j+1} - x_j) = \sum_{j=1}^{n}(x_{j+1} - x_j) = 0$$

and

$$\sum_{j\in S_1} s_j = \sum_{j\in S_2} s_j$$

which shows that the two subsets of $A$ with indices in $S_1$ and $S_2$ solve the original integer set partition problem. □

Theorem 3.2 shows that the solution to a set partition problem can always be obtained by solving an equivalent 1D distance geometry problem. From computational theory, it is already known that the integer set partition problem is an NP-hard problem [21]. Therefore, the distance geometry problem cannot be solved in polynomial time; otherwise, the integer set partition problem would be polynomial time solvable, contradicting the fact that the latter is in fact NP-hard.

### 3.2.3 Distance Bounds

In practice, for example in protein modeling, the distances are often provided with some estimated bounds. The related distance geometry problem then becomes to find the coordinates $x_1, \ldots, x_n$ of the atoms, so that the distances $d_{i,j}$ between atoms $i$ and $j$ are within their estimated lower and upper bounds, $l_{i,j}$ and $u_{i,j}$, respectively, for all $(i, j)$ in a subset $S$ of all pairs of atoms. That is,

$$l_{i,j} \le \|x_i - x_j\| \le u_{i,j} \quad \text{for} \quad (i, j) \in S. \tag{3.11}$$

Let $d_{i,j} = (l_{i,j} + u_{i,j})/2$ and $\varepsilon_{i,j} = (u_{i,j} - l_{i,j})/2$. We can rewrite the problem (3.11) as

$$|\|x_i - x_j\| - d_{i,j}| \le \varepsilon_{i,j} \quad \text{for} \quad (i, j) \in S. \tag{3.12}$$

Then, the problem can be viewed as to find an approximate solution to the distance geometry problem for a set of exact distances $d_{i,j}$ with each distance $\|x_i - x_j\|$ allowed to have an error $\varepsilon_{i,j}$ from $d_{i,j}$. Such a solution is called an $\varepsilon$-approximation solution, or in short, $\varepsilon$-approximation.

If large errors are allowed, an approximate solution is certainly easier to obtain than an exact solution. However, when only small errors are allowed, the problem for finding an approximate solution can be as hard as finding an exact solution. To see this, we first consider the problem of finding an approximate solution to the integer set partition problem.

**Definition 3.3.** *Let $s_j$, $j = 1, \ldots, n$ be a set of positive integers. An approximate solution to the integer set partition problem for the given set of integers is to find two subsets of real numbers, $t_j$, $j = 1, \ldots, n$ such that*

$$\left| \sum_{j \in S_1} t_j - \sum_{j \in S_2} t_j \right| \leq \frac{1}{2}, \quad |t_j - s_j| \leq \varepsilon_j \quad \textit{for} \quad j = 1, \ldots, n, \tag{3.13}$$

*where $S_1$ and $S_2$ are the index sets of the numbers in the first and second subsets, respectively, and $\varepsilon_j$ are the differences allowed for the real numbers from the corresponding integers.*

**Theorem 3.3.** *The problem of obtaining an approximate solution to the integer set partition problem is equivalent to finding an exact solution to the problem when the errors allowed for the approximation are less than $1/(2n)$.*

*Proof.* Suppose that $S_1$ and $S_2$ give a partition for the numbers $t_j$, $j = 1, \ldots, n$ and hence an approximate solution to the corresponding integer set partition problem. Then,

$$\begin{aligned} \left| \sum_{j \in S_1} s_j - \sum_{j \in S_2} s_j \right| &\leq \left| \sum_{j \in S_1} (s_j - t_j) - \sum_{j \in S_2} (s_j - t_j) \right| + \left| \sum_{j \in S_1} t_j - \sum_{j \in S_2} t_j \right| \\ &\leq \sum_{j \in S_1} |s_j - t_j| + \sum_{j \in S_2} |s_j - t_j| + \frac{1}{2}. \end{aligned}$$

Assume that $\varepsilon_{i,j} < 1/(2n)$ for all $j = 1, \ldots, n$. The above result implies that

$$\left| \sum_{j \in S_1} s_j - \sum_{j \in S_2} s_j \right| < \frac{1}{2} + \frac{1}{2} = 1.$$

Note that the sums in the above inequality are over integers. Therefore, if the difference between the two sums is less than 1, the two sums must be equal. It follows that $S_1$ and $S_2$

give a partition for the integers $s_j$, $j = 1, \ldots, n$ and hence an exact solution to the integer set partition problem as well. □

The above discussion implies that the problem of obtaining an approximate solution to the integer set partition problem can be as hard as finding the exact solution to the problem. In other words, if the allowed errors are less than $1/(2n)$, the problem of obtaining an approximate solution must be NP-hard.

**Theorem 3.4.** *The problem of finding an approximate solution to the integer set partition problem can be reduced to the problem of finding an approximate solution of a 1D distance geometry problem.*

*Proof.* Suppose we want to find an approximate solution to an integer set partition problem, with $s_j$, $t_j$, and $\varepsilon_j$ as defined in (3.13). Construct a 1D distance geometry problem with the following distances:

$$d_{j,j+1} = s_j,\ j = 1, \ldots, n \quad \text{and} \quad d_{1,n+1} = 0. \tag{3.14}$$

Instead of solving this problem directly, we solve it approximately by allowing each distance $d_{j,j+1}$ to have an error $\varepsilon_j < 1/(2n)$ for all $j = 1, \ldots, n$. Suppose that we have found an approximate solution, $x_j$, $j = 1, \ldots, n+1$ such that

$$\begin{aligned} ||x_{j+1} - x_j| - d_{j,j+1}| &\le \varepsilon_j, \\ |x_1 - x_{n+1}| &\le \varepsilon_{n+1} \le \frac{1}{2}. \end{aligned} \tag{3.15}$$

Note that

$$\left| \sum_{j=1}^{n} (x_{j+1} - x_j) \right| = |x_{n+1} - x_1| \le \frac{1}{2}. \tag{3.16}$$

Let $t_j = |x_{j+1} - x_j|$. Then, $x_{j+1} - x_j = t_j$ or $-t_j$. Let $S_1 = \{j : x_{j+1} - x_j = t_j\}$ and $S_2 = \{j : x_{j+1} - x_j = -t_j\}$. It follows that

$$\begin{aligned} \left| \sum_{j=1}^{n} (x_{j+1} - x_j) \right| &= \left| \sum_{j \in S_1} (x_{j+1} - x_j) + \sum_{j \in S_2} (x_{j+1} - x_j) \right| \\ &= \left| \sum_{j \in S_1} t_j + \sum_{j \in S_2} t_j \right| \le \frac{1}{2}. \end{aligned}$$

Along with the fact that $|t_j - s_j| \leq \varepsilon_j$ for all $j = 1, \ldots, n$, $S_1$ and $S_2$ give a partition for the numbers $t_j$, $j = 1, \ldots, n$ and hence an approximate solution to the integer set partition problem. □

From Theorem 3.3 and 3.4, we conclude that, if only small errors are allowed for the distances, the problem of obtaining an approximate solution to a distance geometry problem is at least as hard as the problem of finding an approximate solution to the integer set partition problem. Since the latter is NP-hard, the former must be as well.

## 3.3 Review of Literature

The interatomic distances of a protein, which can be obtained from physical experiments and theoretical estimates, happen to be the most essential part of protein structure determination. However, after collecting the distance data, solving a challenging mathematical problem, the distance geometry problem, becomes necessary. The interatomic distances cannot take on arbitrary values; they must rather have particular combinations of some certain values, and according to Crippen and Havel [11], the general form of these combinations was first introduced by Cayley in 1841 [8]. However, it was not systematically studied until 1928, when Menger showed how convexity and many other geometric properties could be defined and studied in terms of pairwise distances between points [45, 46].

In 1935, Schoenberg found an equivalent characterization of Euclidean distances and realized the connection of the problem with bilinear forms [56]. Despite all these studies on the theory of distances and Euclidean geometry, it was Blumenthal, in 1953, who brought together and further clarified all previous work, and first stated the distance geometry problem as "When we have given a set of distances between pairs of points, the distance geometry can give a clue to find a correct set of coordinates for the points in three-dimensional Euclidean space satisfying the given distance constraints. [5]"

In 1979, Saxe [54] showed that the distance geometry problem is strongly NP-complete in one dimension and strongly NP-hard for higher dimensions. In practice, this means that one is unlikely to find a general algorithm to solve all instances of the problem efficiently. However,

the graphs and edge lengths (distances) that Saxe uses in his proofs are very special and are highly unlikely to occur in practical problems. Therefore, this fact has not discouraged scientists from searching and developing new algorithms for solving the distance geometry problem, especially for protein structure determination.

Perhaps, they were Crippen and Havel [11, 27], who have made the major contribution to the improvement of distance geometry. They were the first who applied the distance geometry to the area of protein modeling, using experimental distance data coming from X-ray crystallography and NMR spectroscopy. In 1988, they developed the embedding algorithm, which determines the coordinates of the atoms for a given set of interatomic distances or their ranges. This algorithm has been a very important work in the area of distance based protein modeling; it has been adopted in software like CNS, XPLOR and XPLOR-NIH, and is widely used in NMR modeling [6, 57].

After the first use of NMR spectroscopy for determining the 3D structure of biological macromolecules in solution [30, 40], finding molecular conformations via distance geometry became one of the important subjects to be explored. With the advancement of computational technology, the interest in solving the distance geometry problem has rapidly grown, and the problem is now being studied by many groups.

The existing approaches to the solution of the problem and their recent developments include, but not limited to, the embedding algorithm by Crippen and Havel [11, 28], the alternating projection method by Glunt and Hayden [22, 23], the graph reduction approach by Hendrickson [32, 33], the global optimization method by Moré and Wu [48, 49], the stochastic/perturbation method by Zou, Byrd, and Schnabel [76], the multidimensional scaling method by Kearsly, Tapia, and Trosset [38, 67], the dc programming method by Le Thi Hoai and Pham Dinh [42, 43], the semi-definite programming approach by Biswas, Liang, Toh, and Ye [4], the stochastic search method by Grosso, Locatelli, and Schoen [26], and the geometric buildup algorithm by Dong, Wu, and Wu [15, 16, 68]. In the rest of this section, we will review some of these approaches, and Chapter 4 will be particularly devoted to the geometric buildup approach.

### 3.3.1 The Embedding Algorithm

The embedding algorithm [11, 27], as implemented in CNS with certain extensions, has three successive stages, (1) bound smoothing, (2) metrication, and (3) actual embedding. Given a set of distance ranges, the bound smoothing procedure uses some certain geometric properties like triangle inequality to obtain an estimate of the missing distance ranges. For example, for the distances among any three atoms $i$, $j$, and $k$, if the distance ranges $(l_{i,k}, u_{i,k})$ between $i$ and $k$ as well as the bounds $(l_{k,j}, u_{k,j})$ between $k$ and $j$ are given, but those $(l_{i,j}, u_{i,j})$ between $i$ and $j$ are missing, then $u_{i,j}$ can be estimated via triangle inequality while $l_{i,j}$ via inverse triangle inequality. In other words,

$$\begin{aligned} u_{i,j} &\leq u_{i,k} + u_{k,j}, \\ l_{i,j} &\geq \max\{l_{i,k} - u_{k,j}, l_{k,j} - u_{i,k}\}. \end{aligned} \tag{3.17}$$

Once the bounds for all of the distances are estimated, en exact distance between each pair of bounds is generated. The generated distances may not necessarily be consistent, i.e. they may not even satisfy the triangle inequality. Thus, the metrication procedure is applied to correct the errors when they occur, for example, by regenerating distances between the corresponding bounds.

After metrication, a singular value decomposition algorithm is applied to the generated exact distances to obtain a set of coordinates for the atoms (see Section 3.2.1). If a rank 3 or less decomposition is obtained, the coordinates form a structure whose interatomic distances must satisfy all of the estimated distance ranges. Otherwise, the coordinates provide a structure that can be considered as an approximation to the true structure [29]. This structure can be refined by using optimization, which is typically done by an energy minimization procedure with the distance ranges as the constraints. The minimization may be done using some local or global optimization techniques, such as gradient methods or simulated annealing, and therefore a more accurate set of coordinates is obtained.

Unlike many other approaches, the embedding algorithm also handles chirality constraints, which are very crucial in recognizing chemically valid structures among feasible solutions to the distance geometry problem.

### 3.3.2 Graph Reduction

The distance geometry problem in (3.1) can be naturally formulated as a nonlinear unconstrained global optimization problem as follows:

$$\min_{\vec{x}=(x_1,\ldots,x_n)} \sum_{(i,j)\in S} (\|x_i - x_j\|^2 - d_{i,j}^2)^2. \tag{3.18}$$

This function is infinitely differentiable everywhere and has a number of local minimizers. It is clear that $\vec{x} = (x_1, \ldots, x_n)$ solves the problem (3.18) if and only if the objective function is minimized to zero.

In 1991, Hendrickson [34] proposed an approach to the solution of the distance geometry problem that replaces the large optimization problem in (3.18) by a sequence of smaller ones. He showed that the structure can be exploited by using a divide-and-conquer algorithm, which helps reduce the complexity of the problem.

The atoms in a molecule can be considered as nodes. Similarly, the distances between atoms can be viewed as edges. Then, the distance geometry problem can be described by a distance graph, and the solution to the problem by a realization of the distance graph in a Euclidean space. The graph may be sparse, therefore the embedding may not be unique. There may be more than one way to position the points, and all the distance constraints can still be satisfied. If some of the points can be moved continuously without violating any distance constraints, the graph is called flexible; otherwise, it is called a rigid graph. Note that flexibility of a graph leads to infinitely many solutions to the distance geometry problem [32, 34].

Rigidity and uniqueness of the distance graph can be important for the study of the distance geometry problem. In order for a graph to have a unique embedding, it is obvious that it must first be rigid. However, a rigid graph may still have multiple embeddings, for example, if it has partial reflections. Thus, another necessary condition for unique embeddability is that the graph does not have partial reflections. This is guaranteed in a 3D space if the graph is four-connected ($k$-connected in general, in a $k-1$ dimensional space). These properties can be used to exploit the structure of large graphs to find subgraphs that have unique embeddings. The embedding problem for a given distance graph can then be solved by dividing the graph

into such subgraphs. The solutions found for the subgraphs can finally be combined into a solution for the whole graph [32, 34].

Hendrickson also developed a software package called ABBIE [33, 34] for the determination of molecular structure with a given set of distances. The program first decomposes the graph recursively into subgraphs with unique 3D embeddings. The smaller embedding problem are then solved by minimizing the least-square error function given in (3.18). This method has several advantages. First, if there is not enough distance data to uniquely solve a given problem, the method will identify and solve unique subproblems. Second, the solution of a subproblem can be as important as the whole problem, because for many applications, only a small portion of a molecule, like a binding site, may be of interest. Third, the method can determine if there is sufficient data to the problem or not. Fourth, the distances coming from physical experiments can be erroneous. Such inconsistent data would then be indicated by the inability to solve a particular subproblem. Thus, if there are only a few bad data causing the confusion, identifying and discarding them could be extremely useful for the sake of the whole graph.

### 3.3.3 Alternating Projection Algorithm

Glunt, Hayden and Raydan [23] developed an alternating projection algorithm for solving the distance geometry problem with a given set of bounds on distances. The idea behind the algorithm is as follows: First, a set of distances are generated from the given distance bounds. Then a distance geometry problem with this set of distances is solved by minimizing an error function. If it gives the solution, the program is done; otherwise, the violated distance constraints are used to adjust the distances, and the algorithm is repeated for a new set of distances. In order to adjust the distances, they use a method similar to the one using alternating projections on convex sets for a matrix minimization problem [22].

The algorithm requires the bounds on all the distances available, or relies on a bound smoothing procedure as in the embedding algorithm to provide all the bounds. In every iteration, a least-squares problem needs to be solved, which may require a large amount of computation if a global optimization technique is used. Even with a local optimization tech-

nique, for example, a Newton's algorithm, the total cost can be as much as $O(n^3)$ floating point operations. When the problem size $n$ is large and the problem needs to iterate many times, the Newton's algorithm becomes too expensive to use. Therefore, a spectral gradient algorithm, which is much cheaper than conventional algorithms, is used in the alternating projection algorithm instead.

### 3.3.4 Global Smoothing and Continuation

Moré and Wu [48] formulated the distance geometry problem in terms of finding the global minimum of a weighted function similar to (3.18), i.e. solving

$$\min_{\vec{x}=(x_1,\ldots,x_n)} \sum_{(i,j)\in S} w_{i,j}(\|x_i - x_j\|^2 - d_{i,j}^2)^2, \tag{3.19}$$

where $w_{i,j}$ are positive weights.

They proposed an algorithm, called DGSOL [48, 49], for solving the molecular distance geometry problem by using a global smoothing and continuation method. The method considers the least-squares formulation (3.19) of the distance geometry problem. It does not require all distances or bounds to be available.

The least-squares problem may have many local minimizers. In order to locate the global minimizer, the global smoothing and continuation method first transforms the least-squares function into a set of gradually deformed but smoother or easier functions with fewer local minimizers. The method then locate the minimizers of the transformed functions and trace their changes when the transformed functions are changed back to the original function. A global minimizer hopefully is located in the end.

The transformed function $\langle f \rangle_\lambda$, called the Gaussian transform, of a function $f : \mathbb{R}^n \to \mathbb{R}$ is defined by

$$\langle f \rangle_\lambda = \frac{1}{\pi^{n/2}\lambda^n} \int_{\mathbb{R}^n} f(y) \exp\left(-\frac{\|y-x\|^2}{\lambda^2}\right) dy, \tag{3.20}$$

where the parameter $\lambda$ controls the degree of smoothing. The value $\langle f \rangle_\lambda$ is a weighted average of $f(x)$ in a neighborhood of $x$. The size of the neighborhood decreases as $\lambda$ decreases, and thus as $\lambda \to 0$, $\langle f \rangle_\lambda$ will converge to $f(x)$, recovering the original function in the limit.

This method has been applied to some small to medium-sized test problems with around 200 atoms. The results showed that the method was able to find the global minimizer of the least-squares function with a very high probability while a conventional multi-start random search algorithm failed to find a single global minimizer of the function.

One of the advantages of this method is that it does not need all the distances or bounds. The cost for solving a distance geometry problem is cheaper in the sense that the least-squares function contains a smaller number of terms. The function, the gradient, as well as the Hessian, if required, can all be computed with less cost than for all the distances or bounds. The method is more practical as well since in practice only a very sparse set of distances or bounds are available.

### 3.3.5 D.C. Optimization

Le Thi Hoai and Pham Dinh [42, 43] developed an algorithm for solving the distance geometry problem, based on the d.c. (difference of convex functions) optimization technique. They worked in $\mathcal{M}_{n,3}(\mathbb{R})$, the space of real matrices of order $n \times 3$, where for $X \in \mathcal{M}_{n,3}(\mathbb{R})$, $X_i$ and $X^i$ are its $i$th row and $i$th column, respectively. Then, positions of atoms $x_1, \ldots, x_n$ in a molecule can be identified by $X$, where $X_i^T = x_i$ for $i = 1, \ldots, n$, and the distance geometry problem can be defined by

$$0 = \min \left\{ \sigma(X) = \frac{1}{2} \sum_{(i,j) \in S,\, i<j} w_{i,j} \theta_{i,j}(X) : X \in \mathcal{M}_{n,3}(\mathbb{R}) \right\}, \tag{3.21}$$

where $w_{i,j} > 0$ for $i \neq j$ and $w_{i,i} = 0$ for all $i$. The pairwise potentials $\theta_{i,j} : \mathcal{M}_{n,3}(\mathbb{R}) \to \mathbb{R}$ are defined for (3.1) by either

$$\theta_{i,j}(X) = (d_{i,j}^2 - \|X_i^T - X_j^T\|^2) \tag{3.22}$$

or

$$\theta_{i,j}(X) = (d_{i,j} - \|X_i^T - X_j^T\|). \tag{3.23}$$

For the distance geometry problem with bounds in (3.2), the corresponding $\theta_{i,j}$ will be defined by

$$\theta_{i,j}(X) = \min^2 \left\{ \frac{\|X_i^T - X_j^T\|^2 - l_{i,j}^2}{l_{i,j}^2}, 0 \right\} + \max^2 \left\{ \frac{\|X_i^T - X_j^T\|^2 - u_{i,j}^2}{u_{i,j}^2}, 0 \right\}. \tag{3.24}$$

Therefore, $X$ will be a solution to the distance geometry problem if and only if it is a global minimizer of (3.21) and $\sigma(X) = 0$. The problem (3.21) for $\theta_{i,j}$ given in (3.23) and (3.24) is a nondifferentiable optimization problem, but it is a d.c. optimization problem.

Le Thi Hoai and Pham Dinh showed that the d.c. algorithms can be adapted for developing efficient algorithms for solving large-scale distance geometry problems. They proposed various versions of d.c. algorithms that are based on different formulations for the problem. Due to its local character, the global optimality cannot be guaranteed for a general d.c. problem. However, they showed that the global optimality can be obtained with suitable starting points for the d.c algorithms.

# CHAPTER 4. THE GEOMETRIC BUILDUP APPROACH

In this chapter, we review one of the existing approaches for solving the distance geometry problem, the geometric buildup algorithm. Dong and Wu [15] first applied a geometric buildup algorithm to the solution of the distance geometry problem, and showed the algorithm can find a solution to the problem in linear time if the distances for all the pairs of atoms are available. The work was later extended to sparse distances [16] with an updating scheme proposed by Wu and Wu [68] to control the propagation of numerical errors in the buildup process. The recent development on the algorithm includes the enhancement of the algorithm on rigid vs. unique structure determination by Wu, Wu, and Yuan [69], and the extension of the algorithm to handling inexact or inconsistent distance data by Sit, Wu, and Yuan [60]. The latter work will be seen extensively in Chapter 5, thus we here continue with the reviews of former ones.

## 4.1 Introduction

As mentioned earlier, the major issue about finding the solution to the distance geometry problem is the computational cost of the algorithms developed for solving the problem. In the SVD algorithm with complete set of exact distances, the method requires $O(n^2)$ floating-point operations. If the distances are inconsistent, not only will the SVD fail, but it will also not be able to identify the place where it fails. The embedding algorithm can also be very costly, especially during its first two stages, bound smoothing and metrication due to the SVD algorithm used, as it may repeat the SVD step many times by determining different sets of exact distances satisfying the distance ranges. The global optimization techniques are also computationally expensive, as the objective functions to be minimized might have many local minimizers and finding the global one can be extremely difficult.

In order to improve on running time, Dong and Wu [15] have recently developed an efficient algorithm called the geometric buildup algorithm. Central to the algorithm is the idea to determine only a small group of atoms at the beginning and then complete the whole molecule by repeatedly determining one or more atoms every time using the available distances between the determined and undetermined atoms. The advantage of using a geometric buildup approach is that it works directly on the given distances and exploits the special structure of a given problem, and hence may be able to solve the problem more efficiently than a general approach.

## 4.2 The General Geometric Buildup Algorithm

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, using for example the SVD algorithm as described in Section 3.2.1, 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 $j$ (see Fig. 4.1). Indeed, $x_j$ can be obtained from the solution of the following system of equations,

$$\|x_i\|^2 - 2x_i^T x_j + \|x_j\|^2 = d_{i,j}^2, \qquad i = 1, 2, 3, 4. \tag{4.1}$$

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

$$-2(x_{i+1} - x_i)^T x_j = (d_{i+1,j}^2 - d_{i,j}^2) - (\|x_{i+1}\|^2 - \|x_i\|^2), \qquad i = 1, 2, 3. \tag{4.2}$$

Let $A$ be a matrix and $b$ a vector, and

$$A = -2 \begin{bmatrix} (x_2 - x_1)^T \\ (x_3 - x_2)^T \\ (x_4 - x_3)^T \end{bmatrix}, \quad b = \begin{bmatrix} (d_{2,j}^2 - d_{1,j}^2) - (\|x_2\|^2 - \|x_1\|^2) \\ (d_{3,j}^2 - d_{2,j}^2) - (\|x_3\|^2 - \|x_2\|^2) \\ (d_{4,j}^2 - d_{3,j}^2) - (\|x_4\|^2 - \|x_3\|^2) \end{bmatrix}. \tag{4.3}$$

We then have $Ax_j = b$. Since $x_1, x_2, x_3$, and $x_4$ 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 [15] (see Fig. 4.2).

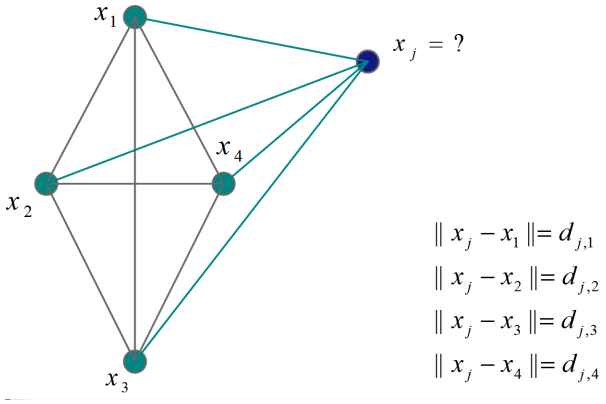



Figure 4.1: **Geometric buildup:** Central to the algorithm is the idea that whenever there are four determined atoms that are not in the same plane and there are distances from these atoms to an undetermined atom, the undetermined atom can immediately be determined uniquely using the distances. If for every atom, the required atoms and distances can be found, the whole structure can be determined uniquely.

The theoretical basis of the geometric buildup approach can be traced back in the study of distance geometry in mathematics [5]. The earliest proposal for such an approach can be found in Sippl and Scheraga [58, 59]. Huang, Liang, and Pardalos [37] 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 4.1.** *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 4.2.** *A set of $k+1$ points in $\mathbb{R}^k$ is called independent if it is not a set of points in $\mathbb{R}^{k-1}$.*

**Theorem 4.1.** *Any $k+1$ independent points in $\mathbb{R}^k$ form a metric basis for $\mathbb{R}^k$.*

*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 $\mathbb{R}^k$. Let $x_j = (x_{j,1}, \ldots, x_{j,k})^T$ be the coordinate vector for any point $j$ in $\mathbb{R}^k$ with distances $d_{i,j}$ from points $i = 1, \ldots, k+1$ to point $j$. Then,

$$\|x_i\|^2 - 2x_i^T x_j + \|x_j\|^2 = d_{i,j}^2, \qquad i = 1, \ldots, k+1, \tag{4.4}$$

and $Ax_j = b$, where

$$A = -2 \begin{bmatrix} (x_2 - x_1)^T \\ (x_3 - x_2)^T \\ \cdots \\ (x_{k+1} - x_k)^T \end{bmatrix}, \quad b = \begin{bmatrix} (d_{2,j}^2 - d_{1,j}^2) - (\|x_2\|^2 - \|x_1\|^2) \\ (d_{3,j}^2 - d_{2,j}^2) - (\|x_3\|^2 - \|x_2\|^2) \\ \cdots \\ (d_{k+1,j}^2 - d_{k,j}^2) - (\|x_{k+1}\|^2 - \|x_k\|^2) \end{bmatrix}. \tag{4.5}$$

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

Figure 4.2: 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.

Given the above properties, we can easily see that a necessary condition for uniquely determining the coordinates of the atoms with a given set of distances is that each atom must have at least four distances to other atoms, and a sufficient condition is that in every step of the geometric buildup algorithm, there is an undetermined atom and the atom has four distances from four determined atoms which are not in the same plane. In general, we have the following results [69]:

**Theorem 4.2.** *A necessary condition for the unique determination of the coordinates of a group of points $x_1, \ldots, x_n$ in $\mathbb{R}^k$ with a given set of distances among the points is that each point must have at least $k+1$ distances from other $k+1$ points, assuming that this point is not in $\mathbb{R}^{k-1}$ with any $k$ of the $k+1$ points.*

*Proof.* It follows immediately from the fact that in $\mathbb{R}^k$, a point can be defined uniquely only if it has $k+1$ distances from $k+1$ independent points, assuming it is not in $\mathbb{R}^{k-1}$ with any $k$ of the $k+1$ points. If it has only $k$ distances from $k$ points, the point will have at least two reflective positions. □

**Theorem 4.3.** *A sufficient condition for the unique determination of the coordinates of a group of points $x_1, \ldots, x_n$ in $\mathbb{R}^k$ with a given set of distances among the points is that in every step of the geometric buildup algorithm, there is an undetermined point with $k+1$ distances from $k+1$ independent and determined points.*

*Proof.* Follows from the construction of the geometric buildup algorithm, because if the condition holds in every step of the algorithm, it will be able to determine the coordinates of all the points uniquely. □

Fig. 4.3 shows an example protein structure determined by using the general geometric buildup algorithm, with the distances for all the pairs of atoms in the protein, as demonstrated in Dong and Wu [15]. The structure is determined accurately and uniquely. The RMSD value of the structure compared with its X-ray reference structure is 1.0e-04 Å. The computation time is much more efficient than the conventional SVD algorithm described in Section 3.2.1.


Figure 4.3: **Structure determination with geometric buildup:** The X-ray crystal structure (left) of the HIV-1 RT p66 protein (4,200 atoms) and the structure (right) determined by the geometric buildup algorithm using the distances for all pairs of atoms in the protein. The algorithm took only 188,859 floating-point operations, while a conventional SVD algorithm required 1,268,200,000 floating-point operations.

## 4.3 An Updated Geometric Buildup Algorithm

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.

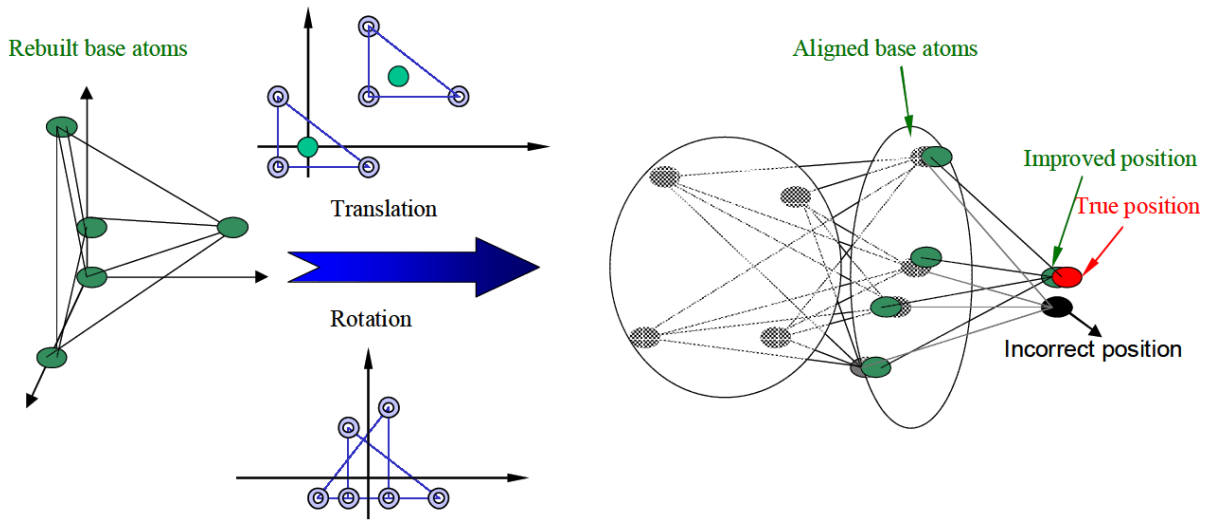
Wu and Wu [68] proposed an updating scheme to prevent the accumulation of the numerical errors. The idea of the scheme is based on the fact that the coordinates of any four atoms can be determined without any other information if all the distances among them are given. Therefore, the coordinates of any 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 (see Fig. 4.4).

Let $y_1, y_2, y_3, y_4$ be the coordinate vectors of the four atoms calculated in the regular geometric buildup process, and $x_1, x_2, x_3, x_4$ the recalculated coordinate vectors. Let $Y$ and $X$ be the corresponding coordinate matrices, i.e.

$$Y = \{y_{i,k} : i = 1, 2, 3, 4,\ k = 1, 2, 3\} \quad \text{and} \quad X = \{x_{i,k} : i = 1, 2, 3, 4,\ k = 1, 2, 3\}. \tag{4.6}$$

Figure 4.4: Redetermination of base atoms




The four base atoms are redetermined if the distances among them are given. The atoms are then moved to and aligned with their original positions, and used to determine other atoms.

If the distances among all the four atoms are available, $X$ can be obtained for example using the SVD algorithm described in Section 3.2.1. 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:

$$x_c^T = \sum_{i=1}^{4} X(i,:)/4, \quad y_c^T = \sum_{i=1}^{4} Y(i,:)/4. \tag{4.7}$$

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

$$X \Leftarrow X + e(y_c - x_c)^T, \tag{4.8}$$

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 (see Fig. 4.5). In fact, the calculation of such a deviation can be done by solving an optimization problem,

$$\min_Q \|Y - XQ\|_F, \quad QQ^T = I, \tag{4.9}$$

where $\|\cdot\|_F$ is the matrix Frobenius norm and $Q$ the rotation matrix. Let $C = X^TY$, 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 [24].

Figure 4.5: The updated geometric buildup algorithm

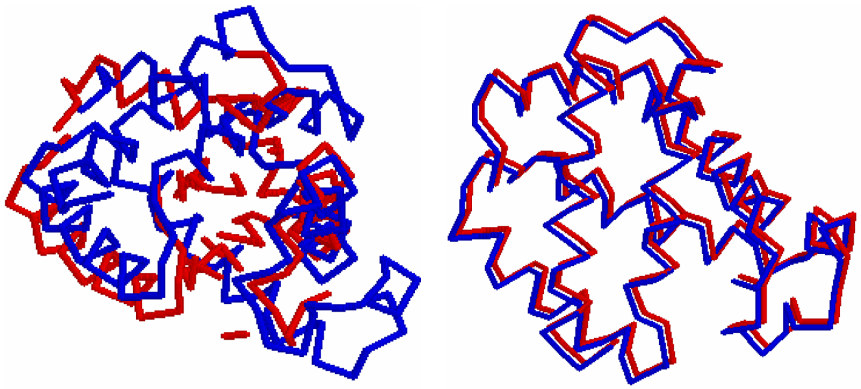
1. Find four atoms that are not in the same plane.
2. Determine the coordinates of the atoms with the distances among them.
3. Repeat:
    - For each of the undetermined atoms,
        - If the atom has 4 distances to 4 determined atoms that are not in the same plane,
            - Determine the atom with its distances to the base atoms.
            - If the base atoms have all distances among them,
                - Recalculate their coordinates with these distances.
                - 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.

Fig. 4.6 demonstrates in some scenarios for how the structure determined by a geometric buildup algorithm can be affected by the accumulated numerical errors and how they can be corrected by using the updating scheme, as given in Wu and Wu [68]. The figure shows the structures (red lines) of protein 4MBA (1086 atoms) determined using $\leq 5$ Å distances, first by the general geometric buildup algorithm (see Fig. 4.6(a)) and then by the updating algorithm (see Fig. 4.6(b)). The graphs show that the general algorithm results in a structure that disagrees with the X-ray reference structure (blue lines) in many regions, while the updating algorithm generates a structure that agrees with the X-ray reference structure (blue lines) almost completely.

## 4.4 Rigid vs. Unique Geometric Buildup Algorithm

For the unique determination of a structure, it is necessary that every atom has at least four distances from other atoms. Further, the general geometric buildup algorithm requires four distances from four determined atoms to the atom to be determined in every buildup step. These conditions may not be satisfied by a given set of distances in practice. If the first

Figure 4.6: Control of rounding errors



(a) (b)

(a) The structure (red lines) of 4MBA determined by using a general geometric buildup algorithm and compared with the original structure of 4MBA (blue lines). (b) The structure (red lines) of 4MBA determined by using an updating geometric buildup algorithm and compared with the original structure of 4MBA (blue lines).

condition is not satisfied, the structure will not be guaranteed unique. If the second condition is not satisfied, the general geometric buildup algorithm will not be able to determine the structure, even if the first condition is satisfied and the structure is unique.

In order to handle more sparse distance data, Wu, Wu, and Yuan [69] proposed a rigid geometric buildup algorithm which can determine the structures only rigidly instead of uniquely. The necessary condition to have a rigid structure requires only three distances for each atom. Therefore, in every buildup step, the geometric buildup algorithm can be modified to require only three distances from three determined atoms to the atom to be determined. The atom can then be determined rigidly, although with two possible positions. In the end, the algorithm may produce multiple structures, due to the multiple choices of the positions of the atoms, but the structures are rigid and in finite number.

More formally, in any buildup step, let $x_i = (x_{i,1}, x_{i,2}, x_{i,3})^T$, $i = 1, 2, 3$, be the coordinate vectors of three determined atoms that are not in a line. Let $x_j = (x_{j,1}, x_{j,2}, x_{j,3})^T$ be the coordinate vector for an undetermined atom $j$ and $d_{i,j}$ the distances from atoms $i = 1, 2, 3$ to atom $j$. Then, $x_j$ can be obtained from the solution of the following system of equations,

$$\|x_i\|^2 - 2x_i^T x_j + \|x_j\|^2 = d_{i,j}^2, \qquad i = 1, 2, 3. \tag{4.10}$$

By subtracting equation $i$ from equation $i+1$ for $i = 1, 2$, the quadratic terms for $x_j$ can be eliminated to obtain

$$-2(x_{i+1} - x_i)^T x_j = (d_{i+1,j}^2 - d_{i,j}^2) - (\|x_{i+1}\|^2 - \|x_i\|^2), \quad i = 1, 2. \tag{4.11}$$

Let $A$ be a matrix and $b$ a vector, and

$$A = -2 \begin{bmatrix} (x_2 - x_1)^T \\ (x_3 - x_2)^T \end{bmatrix}, \quad b = \begin{bmatrix} (d_{2,j}^2 - d_{1,j}^2) - (\|x_2\|^2 - \|x_1\|^2) \\ (d_{3,j}^2 - d_{2,j}^2) - (\|x_3\|^2 - \|x_2\|^2) \end{bmatrix}. \tag{4.12}$$

We then have $Ax_j = b$. Let $x_j = A^T y_j$, where $y_j = (y_{j,1}, y_{j,2})^T$. Then, $AA^T y_j = b$. Since $x_1, x_2, x_3$ are not in the same line, $A$ must be full rank and $AA^T$ be nonsingular. We can therefore solve the linear system $AA^T y_j = b$ to obtain a unique solution for $y_j$. Let $x'_j = (x_{j,1}, x_{j,2})^T$ and $A' = A(1:2, 1:2)$. Then, $x'_j = [A']^T y_j$. By using one of the equations in (4.10), we can obtain two possible values for $x_{j,3}$, assuming that the equation has real solutions. In the end, we obtain two solutions for (4.10).

The advantage of using the modified buildup algorithm is that the algorithm requires fewer distance constraints than the general buildup algorithm. It can handle even more sparse distance data, yet determine meaningful structures. The modified algorithm may find multiple structures, but they all are rigid, and in some cases, it can find a unique structure as well, because the requirement by the general buildup algorithm on the availability of the special four distances in every buildup step is sufficient for the determination of a unique structure, but not necessary.

However, a problem with the modified buildup algorithm is that it may produce too many possible structures: Since in every step, an atom is only determined rigidly, there may be at least two possible positions for it. We have to keep both positions unless later on we find that one of them can be excluded with other distance constraints. Moreover, the three determined atoms may also have multiple positions. Let the $i$th determined atom have $l_i$ possible positions, $i = 1, 2, 3$. Then, in the worst case, there can be $2 \times l_1 \times l_2 \times l_3$ possible positions for the atom to be determined. Therefore, as the algorithm proceeds, the total number of possible positions for an atom to be determined may grow into exponentially many.

To reduce the number of possible positions for an atom, we can allow the algorithm to determine the atom uniquely first if there are more than three required distances available, and determine it rigidly otherwise. Also, in every buildup step, after the atom is determined, either rigidly or uniquely, we can examine all given distances from this atom to other determined atoms for their possible positions. If some positions have violated their distance constraints, they can be removed for further consideration. In this way, the structures generated in the end are guaranteed to satisfy all available distance constraints among the atoms, and they may be reduced to a unique structure after all infeasible structures are identified and removed (see Fig. 4.7).

Figure 4.7: The rigid geometric buildup algorithm

1. Find at least three atoms that are not in the same line.
2. Determine the coordinates of the atoms with the distances among them.
3. Repeat:
   - For each of the undetermined atoms,
     - If the atom has $> 3$ distances to the determined atoms,
       - Determine the atom uniquely.
       - Check multiple structures with all these distances.
       - Remove structures that violate the distance constraints.
     - End
     - If the atoms has 3 distances to 3 determined atoms,
       - Determine the atom rigidly.
       - Record multiple structures generated from reflections.
     - End
   - End
4. If no atom can be determined in the loop, stop.
5. All atoms are determined.

Similar to the general geometric buildup algorithm, the theoretical basis for the rigid geometric buildup algorithm can be established and generalized to any $k$-dimensional Euclidean space.

**Definition 4.3.** *A set of points $B$ in a space $S$ is a reduced metric basis of $S$ provided any*

*point in $S$ can be determined rigidly by its distances to the points in $B$.*

**Definition 4.4.** *A set of $k$ points in $\mathbb{R}^k$ is said to be an independent set of points if it is not a set of points in $\mathbb{R}^{k-2}$.*

**Theorem 4.4.** *A set of $k$ independent points in $\mathbb{R}^k$ form a reduced metric basis for $\mathbb{R}^k$.*

*Proof.* It follows directly by generalizing the modified 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 $i = 1, \ldots, k$ in $\mathbb{R}^k$. Let $x_j = (x_{j,1}, \ldots, x_{j,k})^T$ be the coordinate vector for any point $j$ in $\mathbb{R}^k$ with distances $d_{i,j}$ from points $i = 1, \ldots, k$ to point $j$. Then

$$\|x_i\|^2 - 2x_i^T x_j + \|x_j\|^2 = d_{i,j}^2, \qquad i = 1, \ldots, k, \tag{4.13}$$

and $Ax_j = b$, where

$$A = -2 \begin{bmatrix} (x_2 - x_1)^T \\ (x_3 - x_2)^T \\ \cdots \\ (x_k - x_{k-1})^T \end{bmatrix}, \quad b = \begin{bmatrix} (d_{2,j}^2 - d_{1,j}^2) - (\|x_2\|^2 - \|x_1\|^2) \\ (d_{3,j}^2 - d_{2,j}^2) - (\|x_3\|^2 - \|x_2\|^2) \\ \cdots \\ (d_{k,j}^2 - d_{k-1,j}^2) - (\|x_k\|^2 - \|x_{k-1}\|^2) \end{bmatrix}. \tag{4.14}$$

Let $x_j = A^T y_j$, where $y_j = (y_{j,1}, \ldots, y_{j,k-1})^T$. Then, $AA^T y_j = b$. Since $x_1, \ldots, x_k$ are not in $\mathbb{R}^{k-2}$, $A$ must be full rank and $AA^T$ be nonsingular. We can therefore solve the linear system $AA^T y_j = b$ to obtain a unique solution for $y_j$. Let $x_j' = (x_{j,1}, \ldots, x_{j,k-1})^T$ and $A' = A(1 : k-1, 1 : k-1)$. Then, $x_j' = [A']^T y_j$. By using one of the equations in (4.13), we can obtain two possible values for $x_{j,k}$, assuming that the equation has real solutions. In the end, we obtain two solutions for (4.13), and the positions for point $j$ are determined rigidly. □

Given the above properties, we can easily see that a necessary condition for rigidly determining the coordinates of the atoms with a given set of distances is that each atom must have at least three distances to other atoms, and a sufficient condition is that in every step of the geometric buildup algorithm, there is an undetermined atom and the atom has three distances from three determined atoms which are not in the same line. In general, we have the following results [69]:

**Theorem 4.5.** *A necessary condition for the rigid determination of the coordinates of a group of points $x_1, \ldots, x_n$ in $\mathbb{R}^k$ with a given set of distances among the points is that each point must have at least $k$ distances from other $k$ points, assuming that this point is not in $\mathbb{R}^{k-2}$ with any $k-1$ of the $k$ points.*

*Proof.* It follows immediately from the fact that in $\mathbb{R}^k$, a point can be defined rigidly only if it has $k$ distances to $k$ independent points, assuming it is not in $\mathbb{R}^{k-2}$ with any $k-1$ of the $k$ points. If it has only $k-1$ distances from $k-1$ points, the position of the point will be flexible. □

**Theorem 4.6.** *A sufficient condition for the rigid determination of the coordinates of a group of points $x_1, \ldots, x_n$ in $\mathbb{R}^k$ with a given set of distances among the points is that in every step of the geometric buildup algorithm, there is an undetermined point with $k$ distances from $k$ independent and determined points.*

*Proof.* Follows from the construction of the modified geometric buildup algorithm, because if the condition holds in every step of the algorithm, it will be able to determine the coordinates of all the points rigidly. □

Fig. 4.8 demonstrates the application of the rigid geometric buildup algorithm to a small protein, 1AKG, and the nature of the multiple structures it can generate, as given along with other examples in [69]. The protein 1AKG is a small polypeptide with 16 amino acids and 110 atoms. The general geometric buildup algorithm is able to determine to the structure for this protein completely, with distances $\leq 4.5$ Å, and the RMSD value of the structure is 8.3e-07 Å against the original structure. Here, the number of distances used is 1638, which is about 14% of all the distances. However, with distances $\leq 3.5$ Å, the general geometric buildup algorithm fails, but the rigid algorithm is still able to find a reasonable number of rigid structures. Here, the number of distances used is 898, which is only 7.5% of all the distances. There are total 8192 multiple conformations found by the rigid algorithm. The one closest to the original structure has the RMSD value equal to 4.3e-07 Å. Note that $8192 = 2^{13}$, and therefore, the multiple structures are perhaps generated just from a sequence of 13 reflections

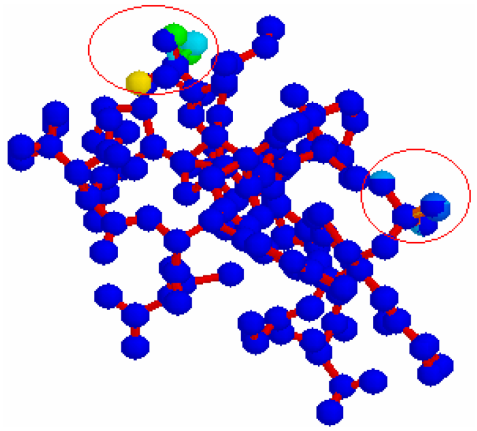


Figure 4.8: **Rigid structure determination:** Shown is structure of protein 1AKG, with 16 residues, 110 atoms. The distances $< 3.5$ Å were used. Total 8192 rigid structures were determined. They all were almost identical except for the circled small regions.

of the atomic positions. In fact, as can be observed in the figure, most of the reflections happen for the side-chain atoms when they are in the surface of the protein, and they only affect the determination of a small part of the structure. On the other hand, the major parts of the protein with the backbone atoms and the atoms in the interior of the protein are all uniquely determined.

# CHAPTER 5. A GEOMETRIC BUILDUP ALGORITHM USING LEAST-SQUARES APPROXIMATIONS[1]

In this chapter, 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. 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 of the geometric buildup approach significantly from both theoretical and practical points of view.

## 5.1 Introduction

We investigate the solution of the distance geometry problem within a so-called geometric buildup framework. Dong and Wu [15, 16] 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

[1] **Modified from a paper published in the *Bulletin of Mathematical Biology* [60].**

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 [68] 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,

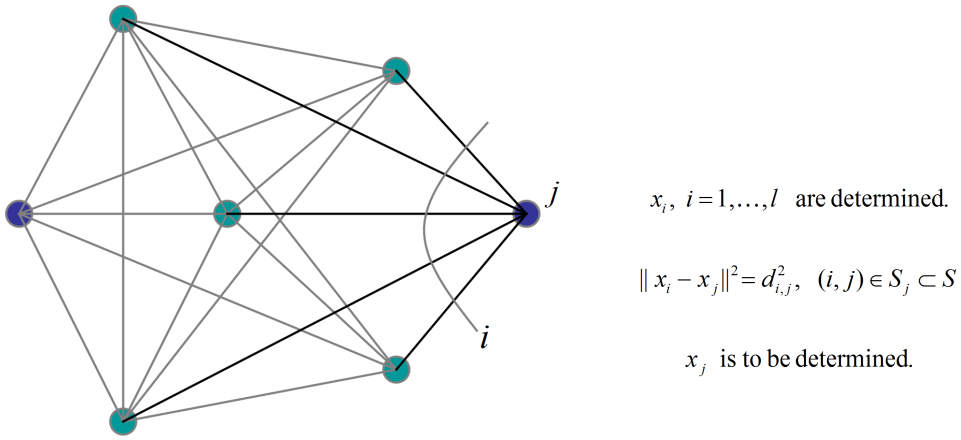
$l \geq 4$, all $l$ distances should be used for the determination of the unknown atom. The reason is that if the distances have errors, they can be inconsistent. Then, the atom satisfying four of the distances may not necessarily satisfy the rest of the distances and therefore, it should be determined 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 [68] 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). 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 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 of the geometric buildup approach significantly from both theoretical and practical point of views.

## 5.2 Geometric Buildup with Linear Least-Squares

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.

Figure 5.1: A buildup step with linear least-squares




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.

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 (see Fig. 5.1). 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,

$$\|x_i\|^2 - 2x_i^T x_j + \|x_j\|^2 = d_{i,j}^2, \qquad i = 1, \ldots, l. \tag{5.1}$$

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

$$-2(x_{i+1} - x_i)^T x_j = (d_{i+1,j}^2 - d_{i,j}^2) - (\|x_{i+1}\|^2 - \|x_i\|^2), \quad i = 1, \ldots, l-1. \tag{5.2}$$

Let $A$ be a matrix and $b$ a vector, and

$$A = -2 \begin{bmatrix} (x_2 - x_1)^T \\ (x_3 - x_2)^T \\ \cdots \\ (x_l - x_{l-1})^T \end{bmatrix}, \quad b = \begin{bmatrix} (d_{2,j}^2 - d_{1,j}^2) - (\|x_2\|^2 - \|x_1\|^2) \\ (d_{3,j}^2 - d_{2,j}^2) - (\|x_3\|^2 - \|x_2\|^2) \\ \cdots \\ (d_{l,j}^2 - d_{l-1,j}^2) - (\|x_l\|^2 - \|x_{l-1}\|^2) \end{bmatrix}. \tag{5.3}$$

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 $3 \times 3$. 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 a 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 $l_m = \max_j\{|S_j|\}$, $S_j = \{i : (i,j) \in S\}$.

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.

Figure 5.2: 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 metric basis for a space and an extended set of independent points in $\mathbb{R}^k$.

**Definition 5.1.** *A set of points $B$ in a space $S$ is an extended metric basis of $S$ provided any point in $S$ can be determined uniquely by its distances from the points in $B$.*

**Definition 5.2.** *A set of $l$ points is said to be an extended set of independent points in $\mathbb{R}^k$ if it contains $k+1$ independent points.*

**Theorem 5.1.** *An extended set of $l$ independent points in $\mathbb{R}^k$ forms a metric basis for $\mathbb{R}^k$.*

*Proof.* It follows directly by generalizing the extended geometric buildup step to the $k$-dimensional Euclidean space. Let $x_i = (x_{i,1}, \ldots, x_{i,k})^T$ be the coordinate vectors for an extended set of independent points $i = 1, \ldots, l$ in $\mathbb{R}^k$. Let $x_j = (x_{j,1}, \ldots, x_{j,k})^T$ be the coordinate vector for

any point $j$ in $\mathbb{R}^k$ with distances $d_{i,j}$ from points $i = 1, \ldots, l$ to point $j$. Then

$$\|x_i\|^2 - 2x_i^T x_j + \|x_j\|^2 = d_{i,j}^2, \qquad i = 1, \ldots, l, \tag{5.4}$$

and $Ax_j = b$, where

$$A = -2 \begin{bmatrix} (x_2 - x_1)^T \\ (x_3 - x_2)^T \\ \cdots \\ (x_l - x_{l-1})^T \end{bmatrix}, \quad b = \begin{bmatrix} (d_{2,j}^2 - d_{1,j}^2) - (\|x_2\|^2 - \|x_1\|^2) \\ (d_{3,j}^2 - d_{2,j}^2) - (\|x_3\|^2 - \|x_2\|^2) \\ \cdots \\ (d_{l,j}^2 - d_{l-1,j}^2) - (\|x_l\|^2 - \|x_{l-1}\|^2) \end{bmatrix}. \tag{5.5}$$

Multiply the equation by $A^T$ to obtain $A^T A x_j = A^T b$. Since $k + 1$ of the $l$ determined points are independent, $A$ must be full rank and $A^T A$ be nonsingular. We can then solve the linear system $A^T A x_j = A^T b$ to obtain a unique solution $x_j = [A^T A]^{-1} A^T b$. □

### 5.3 Geometric Buildup with Nonlinear Least-Squares

The algorithm described in Section 5.2 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 4.3 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

$$\|x_i\|^2 - 2x_i^T x_j + \|x_j\|^2 = d_{i,j}^2, \quad i = 1, \ldots, l+1. \tag{5.6}$$

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

$$d_{i,l+1}^2 - 2x_i^T x_j + d_{j,l+1}^2 = d_{i,j}^2, \quad i = 1, \ldots, l. \tag{5.7}$$

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

$$\begin{aligned} X &= \{x_{i,k} : i = 1, \ldots, l,\ k = 1,2,3\} \quad \text{and} \\ D &= \{(d_{i,l+1}^2 - d_{i,j}^2 + d_{j,l+1}^2)/2 : i, j = 1, \ldots, l\}. \end{aligned} \tag{5.8}$$

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 $\Sigma$ 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 $V = U(:, 1:3)$ 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 5.2.** *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 (5.8) 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 5.3.** *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 $\Sigma$ a $k \times k$ 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 5.4.** *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 $\|\cdot\|_F$ is the matrix Frobenius norm.*

*Proof.* [29] 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

$$f(X) = trace(D^2) - trace(2DXX^T - XX^TXX^T) = trace(D^2) - trace(XX^TXX^T).$$

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

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

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,

$$D = \{(d_{i,l+1}^2 - d_{i,j}^2 + d_{j,l+1}^2)/2 : i, j = 1, \ldots, l\}, \tag{5.9}$$

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 $V = U(:, 1:3)$ 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

Figure 5.3: Geometric buildup with nonlinear least-squares

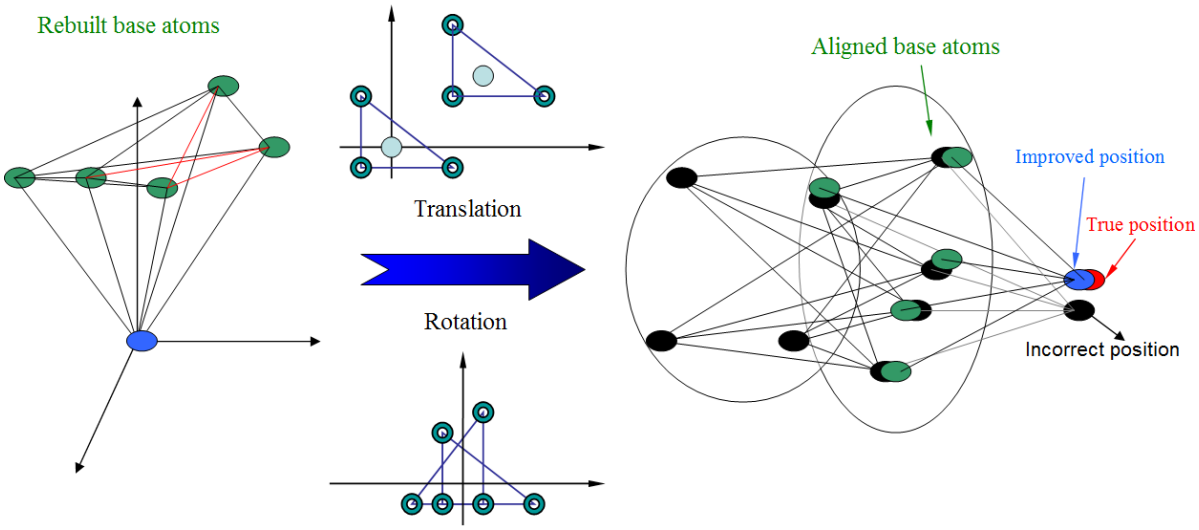
1. Find four atoms that are not in the same plane.
2. Determine the coordinates of the atoms with the distances among them.
3. Repeat:
   - For each of the undetermined atoms,
     - If the atom has $l$ distances to $l$ determined atoms that are not in the same plane,
       - Determine the $l+1$ atoms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

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 5.4, 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 Section 5.2.

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 undetermined atom in the original structure, we need to make a proper

Figure 5.4: A buildup step with nonlinear least-squares




The base atoms and the new atom are redetermined in a new reference system using the distances among them. The distances among base atoms may not be provided, but they can be calculated as the base atoms are already determined. The base atoms are then moved to and aligned with their original positions, in order to find the new position of the undetermined atom.

translation and rotation for the coordinates just like we need to do in the updating scheme for the general geometric buildup algorithm (see Fig. 5.3 and 5.4). 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 $Qx_i$.

## 5.4 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 [2] 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 Å, 6 Å, 7 Å, 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 Section 5.2 and 5.3, respectively. The programs were written in MATLAB and run on a standard desktop workstation. Table 6.1 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 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 Section 5.3, 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 5.1: Available distances for different cutoff values*

| | | | ≤ 5 Å | | ≤ 6 Å | | ≤ 7 Å | | ≤ 8 Å | |
|---|---|---|---|---|---|---|---|---|---|---|
| ID | TA | TD | 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 5.2 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 6.1. The RMSD values show that the algorithm solved almost all the problems with cutoff distances equal to 6 Å, 7 Å, 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 5.2: RMSD values of structures computed with linear least-squares*

| | | ≤ 5 Å | | ≤ 6 Å | | ≤ 7 Å | | ≤ 8 Å | |
|---|---|---|---|---|---|---|---|---|---|
| ID | TA | 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 between the original and computed structure (in Å)

Table 5.3 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 6.1. The RMSD values show that the algorithm solved all the test problems perfectly. The largest RMSD value in Table 5.3 is in the order of $10^{-11}$, which is considered as being almost zero for most of the scientific applications. For RMSD being zero means that the two structures are completely identical, and this shows that the buildup algorithm with nonlinear least-squares is much more powerful and reliable for determining structures with exact distance data.

Table 5.4 presents the performance results for the same test cases as shown in Table 5.2 and 5.3, 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 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 a time range from less than a second to about two minutes for all the test cases.

Table 5.3: RMSD values of structures computed with nonlinear least-squares*

| | | ≤ 5 Å | | ≤ 6 Å | | ≤ 7 Å | | ≤ 8 Å | |
|---|---|---|---|---|---|---|---|---|---|
| ID | TA | DA | RMSD | DA | RMSD | DA | RMSD | DA | RMSD |
| 1PTQ | 402 | 402 | 1.6e−14 | 402 | 3.0e−14 | 402 | 1.8e−14 | 402 | 1.5e−14 |
| 1HOE | 558 | 558 | 8.2e−14 | 558 | 5.3e−14 | 558 | 3.6e−14 | 558 | 3.3e−14 |
| 1LFB | 641 | 641 | 6.0e−14 | 641 | 1.8e−14 | 641 | 2.0e−14 | 641 | 1.6e−14 |
| 1PHT | 814 | 809 | 6.2e−14 | 814 | 5.0e−14 | 814 | 5.0e−14 | 814 | 4.7e−14 |
| 1POA | 914 | 914 | 2.1e−13 | 914 | 5.5e−14 | 914 | 5.0e−14 | 914 | 5.2e−14 |
| 1AX8 | 1003 | 1003 | 1.1e−13 | 1003 | 7.6e−14 | 1003 | 7.2e−14 | 1003 | 7.8e−14 |
| 4MBA | 1086 | 1083 | 2.6e−13 | 1086 | 1.4e−13 | 1086 | 1.3e−13 | 1086 | 1.3e−13 |
| 1F39 | 1534 | 1534 | 7.1e−13 | 1534 | 9.4e−14 | 1534 | 7.6e−14 | 1534 | 6.8e−14 |
| 1RGS | 2015 | 2010 | 5.9e−13 | 2015 | 2.7e−13 | 2015 | 1.9e−13 | 2015 | 1.9e−13 |
| 1BPM | 3672 | 3669 | 4.3e−13 | 3672 | 6.9e−14 | 3672 | 9.8e−14 | 3672 | 4.8e−14 |
| 1HMV | 7398 | 7389 | 2.4e−11 | 7398 | 6.4e−13 | 7398 | 3.0e−13 | 7398 | 2.9e−13 |

* ID: Protein ID, TA: Total number of atoms, DA: Total number of determined atoms, RMSD: RMSD between the original and computed structure (in Å)

In Table 5.5, we compare the new geometric buildup algorithms with previous approaches. The table contains only 6 of the proteins because they were the only ones commonly used by all buildup approaches. The general geometric buildup algorithm (GGBU) introduced by Dong and Wu [15, 16] is capable to solve the distance geometry problem when the distance cutoff is at least 8 Å and not able to provide solution for more sparse data. The updated geometric buildup algorithm (UGBU) presented by Wu and Wu [68] performs better than the general algorithm, but it still fails to determine some structures (e.g. the protein 1AX8). From Table 5.5, we observe that, although the buildup algorithm with linear least-squares (LNLS) doesn't work well with sparse distance data ($\leq 5$ Å), it behaves perfectly and outperforms the previous buildup approaches with larger distance cutoffs ($\leq 8$ Å). However, the buildup algorithm with nonlinear least-squares (NLLS) surpasses the previous buildup approaches and the algorithm

with linear least-squares in all test cases, as can be seen from Table 5.5.

Table 5.4: Total CPU times elapsed during structure determination (in seconds)*

| | | ≤ 5 Å | | ≤ 6 Å | | ≤ 7 Å | | ≤ 8 Å | |
|---|---|---|---|---|---|---|---|---|---|
| ID | TA | LNLS | NLLS | LNLS | NLLS | LNLS | NLLS | LNLS | NLLS |
| 1PTQ | 402 | 0.33 | 0.56 | 0.53 | 0.87 | 0.34 | 1.44 | 0.44 | 2.51 |
| 1HOE | 558 | 0.64 | 0.92 | 0.58 | 1.36 | 0.62 | 2.01 | 0.70 | 3.76 |
| 1LFB | 641 | 0.76 | 0.97 | 0.67 | 1.34 | 0.69 | 2.07 | 0.86 | 3.93 |
| 1PHT | 814 | 1.20 | 1.67 | 1.12 | 2.40 | 1.22 | 4.15 | 1.40 | 8.35 |
| 1POA | 914 | 1.42 | 1.83 | 1.36 | 2.42 | 1.39 | 3.73 | 1.37 | 6.55 |
| 1AX8 | 1003 | 1.65 | 2.32 | 1.64 | 2.89 | 1.61 | 4.65 | 1.58 | 6.97 |
| 4MBA | 1086 | 1.58 | 2.23 | 1.72 | 3.06 | 1.75 | 4.60 | 1.97 | 7.78 |
| 1F39 | 1534 | 3.29 | 3.84 | 3.26 | 4.98 | 3.43 | 7.33 | 3.48 | 12.82 |
| 1RGS | 2015 | 4.98 | 6.01 | 5.24 | 7.68 | 5.07 | 11.23 | 5.46 | 18.14 |
| 1BPM | 3672 | 16.26 | 17.53 | 16.04 | 20.97 | 15.63 | 29.05 | 16.69 | 47.95 |
| 1HMV | 7398 | 64.51 | 66.11 | 64.72 | 74.68 | 63.59 | 86.11 | 67.00 | 113.74 |

* ID: Protein ID, TA: Total number of atoms, 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.

Table 5.6 further demonstrates 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 Å and 6 Å and perturbed them with some small random errors. More specifically, we perturbed every generated distance $d$ by using an update formula

$$d_{i,j} \Leftarrow d_{i,j} + 2 * RE * (0.5 - rand) * d_{i,j},$$

where RE are the maximum relative errors and RE = $10^{-8}, 10^{-7}, 10^{-6}, 10^{-5}$, and $10^{-4}$, 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 Å 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 5.6 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

Table 5.5: Comparison with the previous buildup algorithms*

| | | ≤ 5 Å | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | GGBU | | UGBU | | LNLS | | NLLS | |
| ID | TA | DA | RMSD | DA | RMSD | DA | RMSD | DA | RMSD |
| 1HOE | 558 | – | – | 558 | 8.2e−13 | 558 | 5.8e−02 | 558 | 8.2e−14 |
| 1LFB | 641 | – | – | 641 | 9.5e−12 | 641 | 2.0e−02 | 641 | 6.0e−14 |
| 1PHT | 814 | – | – | 809 | 7.9e−09 | 809 | 1.2e+01 | 809 | 6.2e−14 |
| 1POA | 914 | – | – | 914 | 6.8e−10 | 914 | 6.6e+00 | 914 | 2.1e−13 |
| 1AX8 | 1003 | – | – | – | – | 1003 | 5.2e+00 | 1003 | 1.1e−13 |
| 1RGS | 2015 | – | – | 2010 | 7.4e−08 | 2010 | 2.0e+01 | 2010 | 5.9e−13 |
| | | ≤ 8 Å | | | | | | | |
| | | GGBU | | UGBU | | LNLS | | NLLS | |
| ID | TA | DA | RMSD | DA | RMSD | DA | RMSD | DA | RMSD |
| 1HOE | 558 | 558 | 9.4e−06 | 558 | 1.0e−11 | 558 | 1.8e−13 | 558 | 3.3e−14 |
| 1LFB | 641 | – | – | 641 | 3.9e−12 | 641 | 1.3e−13 | 641 | 1.6e−14 |
| 1PHT | 814 | 814 | 4.4e−05 | 814 | 1.8e−12 | 814 | 1.8e−13 | 814 | 4.7e−14 |
| 1POA | 914 | – | – | 914 | 1.7e−11 | 914 | 4.9e−13 | 914 | 5.2e−14 |
| 1AX8 | 1003 | 1003 | 1.5e−06 | 998 | 3.5e−12 | 1003 | 7.7e−13 | 1003 | 7.8e−14 |
| 1RGS | 2015 | – | – | 2015 | 1.1e−09 | 2015 | 1.7e−12 | 2015 | 1.9e−13 |

* ID: Protein ID, TA: Total number of atoms, DA: Total number of determined atoms, RMSD: RMSD between the original and computed structure (in Å)

distances were perturbed with RE = $10^{-8}, 10^{-7}, 10^{-6}, 10^{-5}$, and $10^{-4}$. 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, especially when the problem sizes are small. The proposed algorithms failed to produce accurate structures for some of the test cases when the problem sizes are large and therefore, for those cases the accumulated rounding errors are still large. However, in either case, we observed that the algorithm using nonlinear least-squares always outperformed the one using linear least-squares.

Table 5.6: RMSD values of structures computed with perturbed distances

| | | ≤ 5 Å | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | RE: 1.0e-08 | | RE: 1.0e-07 | | RE: 1.0e-06 | | RE: 1.0e-05 | | RE: 1.0e-04 | |
| ID | TA | LNLS | NLLS | LNLS | NLLS | LNLS | NLLS | LNLS | NLLS | LNLS | NLLS |
| 1PTQ | 402 | 7.8e+00 | 6.2e−06 | 6.9e+00 | 6.2e−05 | 1.5e+01 | 5.9e−04 | 1.1e+01 | 4.6e−03 | 9.9e+00 | 2.2e−02 |
| 1HOE | 558 | 8.2e+00 | 2.4e−06 | 8.7e+00 | 2.4e−05 | 8.3e+00 | 2.6e−04 | 9.3e+00 | 4.7e−03 | 1.0e+01 | 1.1e−02 |
| 1LFB | 641 | 1.8e+01 | 5.9e−07 | 8.6e+00 | 5.9e−06 | 1.5e+01 | 5.9e−05 | 1.6e+01 | 8.7e−04 | 1.0e+01 | 1.9e−02 |
| 1PHT | 814 | 2.4e+01 | 1.8e−06 | 9.7e+00 | 1.8e−05 | 1.1e+01 | 1.8e−04 | 9.1e+00 | 1.5e−03 | 1.2e+01 | 6.2e−02 |
| 1POA | 914 | 9.6e+00 | 6.0e−06 | 9.2e+00 | 6.0e−05 | 1.2e+01 | 6.0e−04 | 1.1e+01 | 1.5e−03 | 3.6e+01 | 1.7e−02 |
| 1AX8 | 1003 | 2.2e+03 | 1.3e−06 | 1.2e+01 | 1.3e−05 | 1.4e+01 | 1.3e−04 | 1.5e+01 | 1.3e−03 | 1.5e+06 | 1.2e−02 |
| 4MBA | 1086 | 1.0e+01 | 8.4e−06 | 1.3e+01 | 8.4e−05 | 3.0e+01 | 8.4e−04 | 1.2e+01 | 8.3e−03 | 1.0e+01 | 6.6e−02 |
| 1F39 | 1534 | 2.5e+02 | 1.3e−05 | 2.4e+04 | 1.3e−04 | 9.6e+01 | 9.8e−04 | 2.4e+02 | 8.3e−03 | 1.1e+02 | 1.4e+01 |
| 1RGS | 2015 | 6.6e+06 | 1.7e−05 | 6.2e+02 | 1.7e−04 | 3.9e+01 | 1.7e−03 | 2.2e+01 | 1.5e−02 | 1.6e+01 | 2.7e−01 |
| 1BPM | 3672 | 2.1e+01 | 1.0e−05 | 3.3e+02 | 1.0e−04 | 2.1e+01 | 1.0e−03 | 3.2e+04 | 1.0e−02 | 2.6e+01 | 1.1e−01 |
| 1HMV | 7398 | 3.6e+12 | 4.0e−04 | 4.5e+03 | 3.3e−03 | 5.9e+02 | 2.5e+00 | 5.8e+06 | 2.8e+01 | 7.3e+07 | 3.2e+01 |
| | | ≤ 6 Å | | | | | | | | | |
| | | RE: 1.0e-08 | | RE: 1.0e-07 | | RE: 1.0e-06 | | RE: 1.0e-05 | | RE: 1.0e-04 | |
| ID | TA | LNLS | NLLS | LNLS | NLLS | LNLS | NLLS | LNLS | NLLS | LNLS | NLLS |
| 1PTQ | 402 | 3.1e−04 | 7.9e−07 | 2.9e−03 | 7.9e−06 | 1.7e−02 | 7.8e−05 | 6.3e−01 | 3.7e−04 | 4.7e+00 | 3.8e−03 |
| 1HOE | 558 | 2.3e−04 | 1.7e−06 | 3.5e−03 | 1.7e−05 | 2.0e−01 | 1.6e−04 | 2.1e+00 | 1.0e−03 | 2.2e+00 | 4.7e−03 |
| 1LFB | 641 | 1.1e−02 | 2.4e−07 | 1.2e−01 | 2.4e−06 | 3.7e−01 | 2.4e−05 | 3.8e+01 | 2.4e−04 | 9.2e+00 | 2.4e−03 |
| 1PHT | 814 | 4.3e−02 | 4.8e−07 | 1.2e+00 | 4.8e−06 | 2.2e−01 | 4.8e−05 | 1.6e+00 | 4.8e−04 | 4.3e+00 | 4.7e−03 |
| 1POA | 914 | 6.1e−03 | 8.6e−07 | 5.8e−02 | 8.6e−06 | 1.1e+00 | 8.4e−05 | 3.9e+00 | 8.8e−04 | 4.3e+00 | 3.9e−03 |
| 1AX8 | 1003 | 1.6e+00 | 1.2e−06 | 1.8e+00 | 1.2e−05 | 2.6e+00 | 1.2e−04 | 4.4e+00 | 1.2e−03 | 9.3e+00 | 1.2e−02 |
| 4MBA | 1086 | 3.4e+00 | 5.2e−07 | 4.0e+00 | 5.3e−06 | 7.9e+00 | 5.6e−05 | 1.1e+01 | 4.7e−04 | 1.1e+01 | 1.8e−03 |
| 1F39 | 1534 | 7.5e−01 | 1.5e−06 | 5.0e+00 | 1.5e−05 | 7.4e+00 | 1.5e−04 | 7.9e+00 | 1.6e−03 | 1.8e+01 | 1.5e−02 |
| 1RGS | 2015 | 1.3e+01 | 1.8e−06 | 1.2e+01 | 1.8e−05 | 1.3e+01 | 1.8e−04 | 1.7e+01 | 1.8e−03 | 1.4e+01 | 1.9e−02 |
| 1BPM | 3672 | 1.5e+01 | 1.3e−06 | 1.5e+01 | 1.3e−05 | 5.5e+01 | 1.3e−04 | 2.0e+01 | 1.3e−03 | 2.3e+01 | 1.3e−02 |
| 1HMV | 7398 | 2.8e+01 | 2.2e−05 | 9.9e+03 | 2.2e−04 | 3.0e+01 | 2.2e−03 | 3.0e+01 | 1.7e+01 | 2.6e+01 | 1.7e+01 |

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

## 5.5 Concluding Remarks

In this chapter, 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 introduced in Section 4.2 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 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 updating scheme presented in Section 4.3 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 previously, 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

$$l_{i,j} \leq \|x_i - x_j\| \leq u_{i,j}, \qquad (i,j) \in S. \tag{5.10}$$

The algorithm presented in this chapter may not be applied directly to this kind 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 we will investigate this topic in Chapter 6.

# CHAPTER 6. SOLVING A GENERALIZED DISTANCE GEOMETRY PROBLEM [1]

In this chapter, we propose a new approach to the problem of determining an ensemble of protein structures with a set of interatomic distance bounds in NMR protein modeling. Similar to X-ray crystallography, we assume that the protein has an equilibrium structure and the atoms fluctuate around their equilibrium positions. Then, the problem can be formulated as a generalized distance geometry problem, to find the equilibrium positions and maximal possible fluctuation radii for the atoms in the protein, subject to the condition that the fluctuations should be within the given distance bounds. We describe the scientific background of the work, the motivation of the new approach and the formulation of the problem. We develop a geometric buildup algorithm for an approximate solution to the problem and present some preliminary test results. We also discuss related theoretical and computational issues and potential impacts of this work in NMR protein modeling.

## 6.1 Introduction

Biological studies often end up with studies on certain proteins that are key for a biological system to have certain functions. In order to study a protein, it is necessary and critical to find its geometric structure. There are two principal techniques for protein structure determination: One is X-ray crystallography [18] and another the nuclear magnetic resonance spectroscopy (NMR) [7]. In either case, a set of experimental data is collected and a mathematical problem needs to be solved to form the structure [55, 71]. In this chapter, we study the solution of a mathematical problem for the determination of a protein structure in NMR.

[1] **Modified from a paper submitted to the *Bulletin of Mathematical Biology* [61].**

In NMR, the distances between certain pairs of atoms in a given protein can be detected. The mathematical problem then to be solved is to find the coordinates of the atoms given a set of interatomic distances [11, 72]. This problem is called in mathematics a distance geometry problem [5, 65]. Let $n$ be the number of atoms in a given protein. Let $x_i = (x_{i1}, x_{i2}, x_{i3})^T$ be the coordinate vector for atom $i$, $i = 1, \ldots, n$. Let $\|\cdot\|$ be the Euclidean norm. Then, a distance geometry problem can be formulated as to find $x_i$, $i = 1, \ldots, n$ such that

$$\|x_i - x_j\| = d_{ij} \quad \text{for} \quad (i,j) \in S, \tag{6.1}$$

where $d_{ij}$ is the distance between atoms $i$ and $j$ and $S$ is a given set of $(i,j)$ pairs.

The distance geometry problem in (6.1) can be solved in polynomial time if a complete set of exact distances is available, but is NP-hard for a general sparse set of distances [54, 47]. In NMR, not all the distances can be obtained: Only the distances between hydrogen atoms in short distance ($< 5$ Å) can be detected [11, 72]. The distances are not given in their exact values either: Only their rough ranges such as lower and upper bounds can be obtained because the structure fluctuates [11, 72]. The problem then becomes to find $x_i$, $i = 1, \ldots, n$ such that

$$l_{ij} \leq \|x_i - x_j\| \leq u_{ij} \quad \text{for} \quad (i,j) \in S, \tag{6.2}$$

where $l_{ij}$ and $u_{ij}$ are the lower and upper bounds of $d_{ij}$. We call this problem the distance geometry problem with distance bounds. This problem may have infinitely many possible solutions, corresponding to an ensemble of structures all satisfying the given distance constraints. In NMR, it turns out to be important to not just find one of these structures but the whole ensemble of structures, because the deviations of the structures from each other in the ensemble provide important information on how the protein structure may fluctuate dynamically around its equilibrium state. This dynamic property is often as critical as the structure itself for the understanding of the function of the protein [11, 72].

## 6.2 Least Squares Method

For a given set of distances or distance ranges, a straightforward method to find the coordinates of the atoms is to minimize the total distance errors. The method can be implemented

through the solution to a least squares problem for the distances. For example, given a set of distances $d_{i,j}$, for all $(i,j)$ in $S$, the coordinates $x = \{x_1, \ldots, x_n\}$ of the atoms satisfying (6.1) can be obtained by solving the following problem:

$$\min \sum_{(i,j)\in S} (\|x_i - x_j\|^2 - d_{i,j}^2)^2. \tag{6.3}$$

If a set of distance ranges as in (6.2) is given instead, a similar least squares formula can also be obtained so that the sum of the squares of the errors is minimized when a structure can be found to fit in all of the distance ranges:

$$\min \sum_{(i,j)\in S} (\|x_i - x_j\|^2 - u_{i,j}^2)_+^2 + (l_{i,j}^2 - \|x_i - x_j\|^2)_+^2. \tag{6.4}$$

where for any function $g$, $g_+ = g$ when $g > 0$ and $g_+ = 0$ otherwise.

Other formulations similar to formulas (6.3) and (6.4) have also been used, which simply remove the squares on the distances. Therefore, for exact distances, the problem becomes

$$\min \sum_{(i,j)\in S} (\|x_i - x_j\| - d_{i,j})^2, \tag{6.5}$$

and for distance bounds,

$$\min \sum_{(i,j)\in S} (\|x_i - x_j\| - u_{i,j})_+^2 + (l_{i,j} - \|x_i - x_j\|)_+^2. \tag{6.6}$$

The objective functions (6.5) and (6.6) calculate the errors of the distances directly rather than the errors of the squares of the distances, and therefore may be numerically more stable. However, they are not continuously differentiable, and need to be treated with caution when minimized with a conventional optimization method such as the steepest descent direction method, which requires the continuous differentiability of the objective function to converge [71].

In any case, the advantage of using least squares formulation for the solution of the distance geometry problem is that it does not require all of the distances; in other words, it does not require estimating the missing distances as done in the bound smoothing stage of the embedding algorithm (see Section 3.3.1). This avoids not only introducing additional possible

errors but also overdetermining the solution, because the determination of the coordinates does not necessarily require all of the distances. Note also that for the above least squares problems, the global minimum of the objective function is known to be zero when the solution to the problem exists. Therefore, in contrast to other global optimization problems, the global minimum for the least squares formulation of the distance geometry problem can be verified relatively easily. Nevertheless, the global minimum is generally still difficult to achieve because the objective function is highly nonconvex and has many local minima [71].

## 6.3 A Generalized Distance Geometry Problem

Methods have been proposed for solving the problem in (6.2). Many of them use an optimization method, but often end with false minimizers, and the uniqueness can only be evidenced by having a good number of threads converge to highly similar structures. A common procedure is to generate repeatedly a set of distances within the given distance bounds, and solve a distance geometry problem (6.1) for the generated distances [11, 12, 22, 41]. In every step, if a solution to the distance geometry problem is obtained, it must satisfy all the constraints in (6.2) and hence be a solution to the corresponding distance geometry problem with bounds. In the end, a set of solutions to the distance geometry problem with bounds (6.2) is obtained and used to represent the whole solution set and hence the whole ensemble of structures of the protein [11, 12, 22, 41]. A long-standing issue with this approach is that the solution set of the problem is often under-determined or not well represented by the obtained solutions, and the structures, when aligned together, may not be able to fully recover the dynamic fluctuation behaviors of the protein [17, 35, 64, 51, 63].

We propose a new approach to the problem of determining an ensemble of protein structures for a given set of interatomic distance bounds. We assume that a protein has an equilibrium structure and the atoms fluctuate around their equilibrium positions (as described by the B-factors in X-ray crystallography [18]). Then, different from (6.2), we formulate the problem for determining an ensemble of protein structures for a given set of interatomic distance bounds as an optimization problem, to find the equilibrium positions and maximal possible fluctuation

radii for the atoms in the protein, subject to the condition that the fluctuations should be within the given distance bounds (see Fig. 6.1). Let $x_i$ be the coordinate vector and $r_i$ the fluctuation radius of atom $i$, $i = 1, \ldots, n$. Then, the problem can be written as to find $x_i$ and $r_i$, $i = 1, \ldots, n$ such that

$$\begin{aligned} &\max_{x_i, r_i} \sum_{i=1}^{n} r_i \\ &\text{subject to } \|x_i - x_j\| + r_i + r_j \leq u_{i,j} \\ &\qquad\qquad \|x_i - x_j\| - r_i - r_j \geq l_{i,j}, \quad (i,j) \in S \\ &\qquad\qquad r_i \geq 0, \quad i = 1, \ldots, n. \end{aligned} \tag{6.7}$$

We call this problem a *generalized distance geometry problem*, because here the distances are in some sense generalized to distance ranges and the positions to spheres around them. The problem is reduced to the regular distance geometry problem (6.1) if the exact distances are given, when the lower bounds are the same as their upper bounds. The generalized distance geometry problem has not been posed and studied before. It can be an interesting class of problems from a mathematical as well as biological perspective.

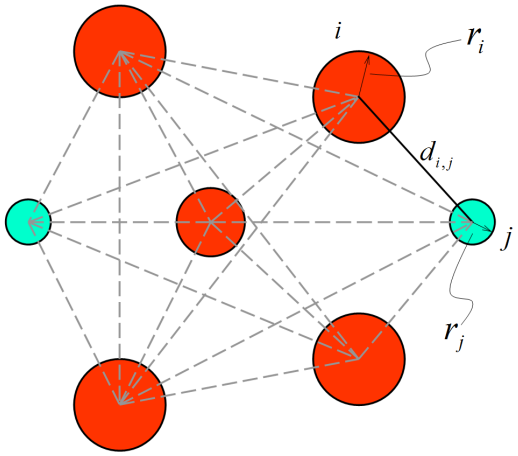



Figure 6.1: **A generalized distance geometry problem:** A position $x_i$ and a maximum possible sphere of radius $r_i$ are to be determined for each atom $i$ such that the distances $d_{ij}$ between atoms $i$ and $j$, when the atoms are restricted in their spheres, are within their given lower and upper bounds, for a subset of all $(i, j)$ pairs.

The problem in (6.7) is not exactly equivalent to that in (6.2), but the solution of the problem can provide a meaningful description on the structure and its dynamic behavior of a given protein. Moreover, the formulation has several advantages over that in (6.2): First, it is a much better defined problem because it requires only a single solution rather than a solution set. Second, it is a constrained optimization problem, which can be approached by an

optimization method, while the problem in (6.2) is system of nonlinear inequalities. There is no effective method for solving a system of nonlinear inequalities, even for obtaining a subset of solutions. Third, the solution of the problem can deliver an NMR structure in a form similar to an X-ray crystal structure, with a single structural file containing the coordinates and fluctuation radii (or B-factors) for the atoms. These advantages make it possible to develop an efficient algorithm for the determination of a structure and its dynamic behaviors using a set of interatomic distance bounds and to improve the way to represent a structural ensemble in NMR protein modeling (Note that the ensemble of structures determined by a set of NMR data should be a continuously connected structures, while a finite number of samples is most likely to under-represent it as concerned by several scientists in the field recently [17, 35, 51, 63, 64]).

In practice, there can be more than tens of thousands of variables and constraints for the problem in (6.7). For example, a protein of 100 residues may have 1000 atoms and at least $4 \times 1000$ pairs of distance bounds. The problem will then have 1000 variables for the fluctuation radii and $3 \times 1000$ variables for the coordinates of the atoms and $2 \times 4 \times 1000$ possible constraints. A constrained optimization problem of this complexity can be very difficult to solve [13, 20, 50, 52]. We develop a so-called geometric buildup algorithm for an approximate solution to the problem. Such an algorithm has been developed for the solution of the distance geometry problem in (6.1) with either exact or inexact distances [15, 16, 60, 68, 69], and can be extended to obtaining an approximate solution to the generalized distance geometry problem in (6.7).

The idea of the geometric buildup algorithm for the solution of a generalized distance geometry problem is to determine the positions and fluctuation radii of the atoms, one at a time, using the distance constraints from the determined atoms to the undetermined ones. For an undetermined atom, if distance bounds between this atom and $l$ determined atoms $x_i$, $i = 1, \ldots, l$ are given, then this atom can immediately be determined. Let us call this atom the $(l+1)$th atom, and let the coordinate vector and fluctuation radius of the atom be denoted as $x_{l+1}$ and $r_{l+1}$, respectively. Then, a subproblem for determining the atom $l + 1$ can be

formulated as:

$$\begin{aligned} \max_{x_{l+1}, r_{l+1}} \quad & r_{l+1} \\ \text{subject to} \quad & \|x_i - x_{l+1}\| + r_i + r_{l+1} \leq u_{i,l+1} \\ & \|x_i - x_{l+1}\| - r_i - r_{l+1} \geq l_{i,l+1}, \quad i = 1, \ldots, l, \\ & r_{l+1} \geq 0. \end{aligned} \tag{6.8}$$

This subproblem has only four variables and $2l+1$ constraints, where, in practice, an average $l$ number is usually small, in the range of 10 to 15, and the maximum $l$ number does not exceed 40. It can therefore be solved relatively easily. By repeatedly solving such a subproblem for an undetermined atom, the coordinate vectors and fluctuation radii of all the atoms can be determined subsequently (see Fig. 6.2).

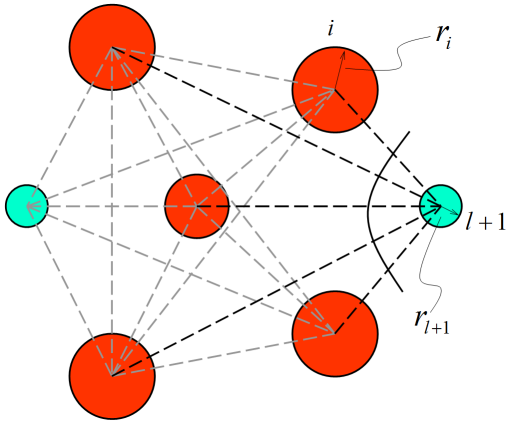



Figure 6.2: **A generalized subproblem:** The idea is to determine the positions and fluctuation radii of the atoms, one at a time, using the distance constraints from the determined atoms to the undetermined ones. For an undetermined atom $l+1$, if distance bounds between this atom and $l$ determined atoms, $i = 1, \ldots, l$ are given, then the atom $l+1$ can immediately be determined.

Note that the generalized distance geometry problem is a hard problem, if a global optimal solution is to be found. In fact, even if we just want to find a feasible solution, the problem is still equivalent to a distance geometry problem with distance bounds which has been proven to be NP-hard if the bounds are tight enough [47]. We will consider only a local optimal solution to the generalized distance geometry problem, and expect that such a solution, or even an approximation, may be adequate for the description of a structural ensemble in terms of its equilibrium structure and possible fluctuation range. Therefore, the geometric buildup algorithm is not meant to be able to solve the generalized distance geometry problem exactly. Indeed, it can provide only an approximate solution to the problem.

## 6.4 Algorithm for Solving a Generalized Distance Geometry Problem Approximately

In this section, we present the main algorithm for obtaining an approximate solution to a generalized distance geometry problem (6.7). The algorithm uses a similar buildup framework as the algorithm described in previous section, but attempts to solve a constrained optimization problem (6.7) instead of a system of distance equations (6.1). In every step, it also attempts to solve a constrained optimization subproblem (6.8). We describe the algorithm and discuss the conditions for the solution of the optimization subproblems. We also describe the procedures for obtaining the initial solutions for these problems.

The goal of this algorithm is to obtain an approximate solution to a generalized distance geometry problem (6.7). Instead of directly solving this problem, the algorithm solves a sequence of generalized subproblems (6.8). These problems, (6.7) or (6.8), are all constrained optimization problems. We only consider their local optimal solutions at this stage, hoping that they are adequate for the description of a required structural model. The problems and in particular, the subproblems (because of their small sizes) can thus be solved by using standard local optimization techniques [20, 52].

In the first stage, the algorithm finds a small set of (usually 4) atoms to start. It first selects an atom in the protein with the most number of connections, i.e. having the maximum number of available distances to other atoms, and then finds three more atoms connected to this atom so that the distance bounds for every pair of the initial four atoms are available. There are two reasons for starting from the core part of the protein; first to increase the probability that the algorithm gets started, and second to ensure that later determined atoms have as many available distance bounds as possible, to the previously determined ones. Since the distance bounds for every pair of these atoms are known, the algorithm tries to find the positions and fluctuation radii of these atoms by solving a generalized distance geometry problem (6.7) approximately with the available distance bounds. Here, the positions are determined by using SVD with a set of distances within their bounds, and the fluctuation radii are assigned arbitrarily as long as the distance bound constraints are satisfied. Hopefully, the atoms are

not in the same plane. Otherwise, another set of atoms will be tried.

In the second stage, the algorithm goes into a loop as a general buildup procedure does. In each step of the loop, the algorithm determines the position and fluctuation radius of an undetermined atom by solving a generalized distance geometry subproblem (6.8), if $l \geq 4$ distance bounds are known from this atom to $l$ determined atoms that are not in the same plane. See Fig. 6.3 for an outline of the algorithm.

Figure 6.3: Algorithm for solving a generalized distance geometry problem approximately

1. Find four atoms that are not in the same plane.
2. Determine the coordinates and fluctuation radii of the atoms by solving a problem (6.8) with the distance bounds among them.
3. Repeat:
   - For each of the undetermined atoms,
     - If the atom has $l$ distance bounds to $l$ determined atoms that are not in the same plane ($l \geq 4$),
       - Determine the position and fluctuation radius for the undetermined atom by solving a problem (6.8) with the above $l$ distance bounds.
     - End
   - End
4. If no atom can be determined in the loop, stop.
5. All atoms are determined.

As we have pointed out in Section 6.1, the reason we use a buildup algorithm to solve a generalized distance geometry problem (6.7) is that in practice, there can be more than tens of thousands of variables and constraints for the problem. A constrained optimization problem of this size can be very difficult to solve [20, 52]. On the other hand, each generalized subproblem (6.8) has only a few variables and a small number of constraints (usually in the range of 10 to 15) and can be solved relatively easily.

Note that the solution obtained by solving a sequence of generalized subproblems using the buildup algorithm is not necessarily optimal for the original generalized distance geometry problem. It is NOT in general. However, it is certainly feasible, and may be good enough as an approximately optimal solution in practice. We state some of the feasibility or opti-

mality properties of the generalized distance geometry problem/subproblems in the following theorems.

**Theorem 6.1.** *If the bounds in (6.8) are finite and the feasible region is nonempty, the generalized subproblem must have an optimal solution.*

*Proof.* Because the bounds are finite, the feasible set of solutions of the problem is bounded as well as closed. The objective function of the problem is also continuous. Therefore, by the standard theory in optimization, the problem must have an optimal solution. □

**Theorem 6.2.** *Let $(x_i, r_i)$ be a feasible solution to the generalized subproblem (6.8) in one of the iterative steps of the buildup algorithm. Let $x = \{x_i : \ i = 1, \ldots, n\}$ and $r = \{r_i : \ i = 1, \ldots, n\}$. Then, $(x, r)$ is a feasible solution to the generalized distance geometry problem (6.7).*

*Proof.* Suppose that $(x, r)$ is not feasible for the problem (6.7). Then, there must be a pair of atoms $i$ and $j$ such that $(x_i, r_i)$ and $(x_j, r_j)$ do not satisfy the corresponding distance constraints, i.e.,

$$\begin{aligned} \|x_i - x_j\| + r_i + r_j &\leq u_{i,j}, \\ \|x_i - x_j\| - r_i - r_j &\geq l_{i,j}, \end{aligned} \tag{6.9}$$

for some given distance bounds $u_{i,j}$ and $l_{i,j}$. Assume that atom $i$ is determined before atom $j$ in the buildup algorithm. Then, the above violated constraints (6.9) must be two of the constraints in the generalized subproblem (6.8) for determining atom $j$. Then, $(x_j, r_j)$ must satisfy these constraints because it is a feasible solution for this subproblem. This is a contradiction. Therefore, $(x, r)$ must be feasible for the generalized distance geometry problem (6.7). □

By Theorem 6.2, we see that if we can find a sequence of feasible solutions $\{(x_i, r_i)\}$ for the generalized subproblems (6.8) in the buildup algorithm, by collecting all of them together, we then obtain a feasible solution $(x, r)$ for the generalized distance geometry problem (6.7). If every solution $(x_i, r_i)$ is optimal, $(x, r)$ may not necessarily be optimal, but $r_i$ are maximized in their corresponding subproblems and should provide good, if not optimal overall, estimates on atomic fluctuations, as we will show numerically in next section.

The generalized subproblem (6.8) has a linear objective function, but its constraints are still nonlinear and the second set is even nonconvex. A general optimization algorithm is needed to solve the problem, and for such an algorithm to start, an initial solution to the problem is required. In order to find such an initial solution, we take two steps: First, we use a procedure similar to the buildup algorithm for the solution of a regular distance geometry problem as described in Section 5.3. Let $X = [x_1^T; \ldots; x_l^T]$ be the coordinate matrix for the $l$ determined atoms. Let $x_{l+1}$ be the coordinate vector for the atom to be determined. Following the algorithm in Section 5.3, we first set $y_{l+1} = (0, 0, 0)^T$ and then use the singular value decomposition to solve an equation $D = YY^T$ for $Y$, where $D$ is defined by

$$D = \{(d_{i,l+1}^2 - d_{i,j}^2 + d_{j,l+1}^2)/2 : \ i, j = 1, \ldots, l\}, \tag{6.10}$$

with the following distances,

$$\begin{aligned} d_{i,j} &= \|x_i - x_j\|, \quad i, j = 1, \ldots, l, \quad \text{and} \\ d_{i,j} &\in [l_{i,j}, u_{i,j}], \quad i = 1, \ldots, l, \ j = l + 1. \end{aligned} \tag{6.11}$$

After solving $D = YY^T$ for $Y$ as described in Section 5.3, we obtain a set of coordinate vectors for the atoms with those for atoms 1 to $l$ in $Y$ and that for atom $l + 1$ at the origin. We then translate and rotate all the atoms together so that $X$ and $Y$ are aligned (i.e., RMSD of $X$ and $Y$ is minimized). In the end, we set $x_{l+1} = y_{l+1}$ with the updated $y_{l+1}$.

Note that the generated distances may not be consistent. Therefore, the solution $Y$ can only be an approximate solution to the equation $D = YY^T$ and it may not even satisfy the distance bounds and in particular, the bounds for the distances from the determined atoms to the undetermined one. Therefore, in the second step, we use the coordinate vector $x_{l+1}$ for the undetermined atom in the first step as an initial solution, and solve another optimization subproblem,

$$\min_{x_{l+1}} \sum_{i=1}^{l} (\|x_i - x_{l+1}\|^2 - u_{i,l+1}^2)_+^2 + (l_{i,l+1}^2 - \|x_i - x_{l+1}\|^2)_+^2 \tag{6.12}$$

where $l_{i,l+1}$ and $u_{i,l+1}$ are lower and upper bounds on distance $d_{i,l+1}$, $i = 1, \ldots, l$ and for any function $g$, $g_+ = g$ when $g > 0$ and $g_+ = 0$ otherwise. It is easy to see that $x_{l+1}$ satisfies all

the bounds for the distances from the determined atoms to the undetermined one if and only if the objective function in (6.12) is minimized to zero. Therefore, if $x_{l+1}$ is infeasible for the bounds in the first step, we can solve the problem in (6.12) to make it feasible if possible.

Of course, in order to obtain an initial solution to the subproblem (6.8), we also need to find an initial feasible radius $r_{l+1}$ for atom $l+1$. Let the distance constraints in (6.8) be written in the following form,

$$\begin{aligned} \|x_i - x_{l+1}\| + r_i + r_{l+1} &\leq u_{i,l+1} \\ \|x_i - x_{l+1}\| - r_i - r_{l+1} &\geq l_{i,l+1}. \end{aligned} \tag{6.13}$$

By solving this system of inequalities for $r_{l+1}$, a feasible value for the radius $r_{l+1}$ can then be obtained as

$$r_{l+1} = \min_{1 \leq i \leq l} \min\{u_{i,l+1} - \|x_i - x_{l+1}\| - r_i, \|x_i - x_{l+1}\| - l_{i,l+1} - r_i\}. \tag{6.14}$$

With the above obtained $(x_{l+1}, r_{l+1})$ as an initial solution, a standard optimization procedure is then ready to apply to find an optimal solution for the generalized subproblem (6.8).

## 6.5 Test Results

In this section, we present the test results from applying the buildup algorithm to a set of generalized distance geometry problems. Given the complexity of the generalized distance geometry problem, a buildup algorithm is not always guaranteed to provide a solution, even a feasible solution to it. Therefore, the numerical tests presented here basically serve as a first step concept proofing for obtaining an approximate solution to the generalized distance geometry problem with buildup. They do not necessarily imply that the algorithm is ready to apply to real NMR data. They only show that the algorithm converges reasonably for carefully constructed test problems. Further development of the algorithm and application to real NMR data are needed and will certainly be pursued in our next step work.

We have constructed two sets of test problems using some known protein structures (we generated a set of distance bounds from each of these structures and use it to define a generalized distance geometry problem). The first set of problems use exactly the generated distance

bounds. The problems are considered to be relatively simple because they have solutions for sure. The second set of problems use the generated distance bounds with some small perturbations. They may pose more difficulties. As we will show, our algorithm has obtained an approximate solution for all these problems successfully. Each time, the algorithm was able to find an equilibrium structure for the protein, which is very close to the original structure, and also produce a set of atomic fluctuation radii, which correlate very well with the B-factors of the atoms in the original structure.

Our test problems are constructed as follows. We first downloaded eleven protein structures from the PDB Data Bank [2]. They were determined by X-ray crystallography with the number of atoms in the structures ranging from several hundreds to several thousands. With each of these structures, we generated two sets of distances with a cutoff distance equal to 5 Å and 6 Å, respectively. For each set of distances, we generated artificial upper and lower bounds for the distances according to the following rules:

$$\begin{aligned} l_{ij} &= \|x_i - x_j\| - f_i - f_j \\ u_{ij} &= \|x_i - x_j\| + f_i + f_j \end{aligned} \tag{6.15}$$

where $f_i$ and $f_j$ are proportional to the root-mean-square fluctuations of atoms $i$ and $j$, respectively (extracted from the B-factors of the atoms in the structures). By this we mean that $f_i$ is proportional to $r_i$ in the formula $B_i = 8\pi^2 < r_i, r_i >$. In fact $f_i$ is a fraction of $r_i$, i.e., $f_i = \eta r_i$, for some constant $\eta$ in (0,1), $i = 1, \ldots, n$. By using a small value for $\eta$, we can avoid $f_i$ being too big and the lower bound being negative. However, this $\eta$ value is rather arbitrary. It is so selected just to generate a reasonable set of test data. Fig. 6.4 demonstrates the relationships between fluctuation radii and distance bounds.

Note that the distance constraints we have generated are not so realistic as NMR distance constraints. First, they are generated from X-ray structures, and therefore, are only constraints on distances of heavy atoms, while the NMR constraints are for distances between some hydrogen atoms. Second, they contain the constraints for all the distances within a cutoff distance, while the NMR constraints contain only those for a subset of all cutoff distances. The generated distance constraints are similar to NMR distance constraints in two respects:

They are constraints for short distances (5 or 6 Å). They are also lower and upper bounds on the distances. They can therefore be used to construct a set of generalized distance geometry problems so that the geometric buildup algorithm can be tested. Yet, they are intended to show potential applications of generalized distance geometry problems to the determination of NMR structural ensembles as well.

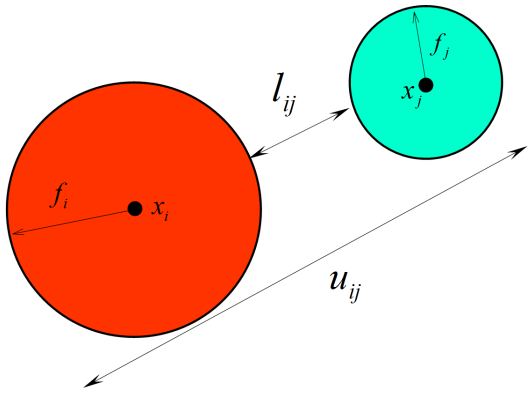



Figure 6.4: Lower and upper bounds for the distance between atoms $i$ and $j$

With the generated distance bounds, we determined an equilibrium structure and a set of atomic fluctuation radii for each protein by solving a generalized distance geometry problem approximately using the buildup algorithm. For each new structure, we evaluated its coordinate RMSD against its original structure. We also analyzed the correlation between the calculated atomic fluctuation radii and the root-mean-square deviations of the atoms extracted from their B-factors in the original structure. In practice, where we are not able to exploit the information about the original structure, we can recalculate the distances after determining the structure, and compare them with the given distance bounds using the following SDME (sparse distance matrix error) and MDE (maximum distance error) values:

$$\begin{aligned} \text{SDME} &= \text{sqrt}\left[\frac{1}{|S|}\sum_{(i,j)\in S}(\|x_i - x_j\| - u_{ij})_+^2 + (l_{ij} - \|x_i - x_j\|)_+^2\right] \\ \text{MDE} &= \max_{(i,j)\in S}(\|x_i - x_j\| - u_{ij})_+ + (l_{ij} - \|x_i - x_j\|)_+, \end{aligned} \tag{6.16}$$

where $|S|$ is the normalizing factor for the SDME value, i.e., the number of distance bounds in the given data. Note that MDE value is the maximal deviation of the distances from their

given intervals, while SDME is similar to distance matrix error (DME), the root-mean-square deviations of the distances from their given intervals.

Table 6.1 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 distance bounds between all pairs of atoms, the number of bounds generated under the cutoff distances 5 Å and 6 Å. In our work, we did not consider the cases of larger cutoffs as for realistic cases, for instance in NMR experiments, the available distances are usually shorter that 5 or 6 Å. From this table, we can see that for each of the structures, a very sparse set of distances (ranging from 0.32% to 8.79%) 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 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 is to obtain different sets of distance data with different sparsity so we can test the algorithm for problems with varying degrees of availability of the distance data.

Table 6.1: Available distance bounds for different cutoff values *

| ID | TA | TD | $\leq 5$ Å | | $\leq 6$ Å | |
|---|---|---|---|---|---|---|
| | | | AD | AD/TD | AD | AD/TD |
| 1PTQ | 402 | 80601 | 4399 | 5.46% | 7088 | 8.79% |
| 1HOE | 558 | 155403 | 6299 | 4.05% | 10178 | 6.55% |
| 1LFB | 641 | 205120 | 6974 | 3.40% | 11435 | 5.57% |
| 1PHT | 814 | 330891 | 11033 | 3.33% | 17695 | 5.35% |
| 1POA | 914 | 417241 | 10468 | 2.51% | 16983 | 4.07% |
| 1AX8 | 1003 | 502503 | 11542 | 2.30% | 18795 | 3.74% |
| 4MBA | 1086 | 589155 | 12761 | 2.17% | 20905 | 3.55% |
| 1F39 | 1534 | 1175811 | 17300 | 1.47% | 28532 | 2.43% |
| 1RGS | 2015 | 2029105 | 22784 | 1.12% | 38020 | 1.87% |
| 1BPM | 3672 | 6739956 | 44789 | 0.66% | 75152 | 1.12% |
| 1HMV | 7398 | 27361503 | 86288 | 0.32% | 143196 | 0.52% |

* ID: Protein ID, TA: Total number of atoms, TD: Total number of distance bounds, AD: Number of available distance bounds

We have implemented our algorithm in MATLAB and run on a standard desktop workstation. The main computation of the algorithm is to solve a sequence of generalized distance geometry subproblems (6.8). For each subproblem (6.8), the algorithm needs to apply SVD to

a metric matrix for the distances in (6.11) to find an initial solution for (6.8). The initial solution is further adjusted in an unconstrained optimization subproblem in (6.12). The algorithm then uses it to solve the constrained optimization subproblem in (6.8). Here, the computations for SVD and for constrained and unconstrained optimization are all done with direct MATLAB routine calls. The initial solution may be infeasible, but the MATLAB routine can still use it to start the constrained optimization procedure. The final solution may or may not be feasible then. Also, the number $l$ used for solving the subproblems (6.8) is usually small. For example, with the 5 Å distance cutoff, an average $l$ number is in the range of 10 to 15, and the maximum $l$ does not exceed 40, while with the 6 Å cutoff, the maximum $l$ can be at most 65, and the average such number still remains in the range of 17 to 22. Thus, the number of constraints in subproblems (6.8) is small, and they can be solved relatively easily. In practice, the real NMR data is even sparser than the data sets generated in Table 6.1, so we should expect $l$ to be even smaller in real applications.

Table 6.2 contains the RMSD (root-mean-square deviation) values of the obtained structures (compared with their original structures), correlation coefficients between the original and calculated atomic fluctuation radii, SDME and MDE values between the original and recalculated distance data. All these error measures are obtained by using the new buildup algorithm on the data sets listed in Table 6.1. From Table 6.2, we observe that as the size of the problem or the number of available distances increase, RMSD values increase. This is quite natural because of the rounding errors accumulated during the buildup process. All the RMSD values are less than $10^{-3}$ and almost all of the SDME and MDE values turn out to be exactly zero (except for 1HOE which is almost zero as well). While keeping all the errors small, the algorithm also determines atomic fluctuation radius for each atom very accurately. From Table 6.2 we see that the smallest correlation value is 0.9692 which means that original and calculated atomic fluctuation radii are almost perfectly correlated (see Fig. 6.5 and Fig. 6.6 for graphics displays of an equilibrium structure and the fluctuation correlation).

Table 6.2 also contains the performance results with the time required by the algorithm for each test case. The program was run in MATLAB R2006a version 7.2 on a desktop workstation,

with 2.40 GHz CPU and 2.00 GB memory. From the table, we see that the computing times of the algorithm are very short. It is a very efficient algorithm that it can finish the calculations in only several seconds to a few minutes for all the test cases.

Table 6.2: Error measures of determined structures *

| | | $\leq 5$ Å | | | | | |
|---|---|---|---|---|---|---|---|
| ID | TA | DA | RMSD | CORR | SDME | MDE | CPU |
| 1PTQ | 402 | 402 | 1.0e-13 | 0.9857 | 0.0e+00 | 0.0e+00 | 7.3 |
| 1HOE | 558 | 558 | 7.1e-14 | 0.9692 | 1.4e-14 | 3.6e-14 | 8.5 |
| 1LFB | 641 | 641 | 4.6e-12 | 0.9960 | 0.0e+00 | 0.0e+00 | 10.5 |
| 1PHT | 814 | 809 | 2.5e-13 | 0.9903 | 0.0e+00 | 0.0e+00 | 14.1 |
| 1POA | 914 | 914 | 1.5e-12 | 0.9741 | 0.0e+00 | 0.0e+00 | 15.5 |
| 1AX8 | 1003 | 1003 | 2.1e-11 | 0.9927 | 0.0e+00 | 0.0e+00 | 19.2 |
| 4MBA | 1086 | 1083 | 7.8e-12 | 0.9815 | 0.0e+00 | 0.0e+00 | 16.6 |
| 1F39 | 1534 | 1534 | 1.3e-12 | 0.9976 | 0.0e+00 | 0.0e+00 | 26.7 |
| 1RGS | 2015 | 2010 | 1.0e-09 | 0.9786 | 0.0e+00 | 0.0e+00 | 37.0 |
| 1BPM | 3672 | 3669 | 1.3e-11 | 0.9781 | 0.0e+00 | 0.0e+00 | 71.5 |
| 1HMV | 7398 | 7389 | 1.2e-04 | 0.9796 | 0.0e+00 | 0.0e+00 | 196.9 |
| | | $\leq 6$ Å | | | | | |
| ID | TA | DA | RMSD | CORR | SDME | MDE | CPU |
| 1PTQ | 402 | 402 | 1.8e-14 | 0.9857 | 0.0e+00 | 0.0e+00 | 6.5 |
| 1HOE | 558 | 558 | 3.3e-14 | 0.9692 | 2.4e-14 | 4.8e-14 | 10.4 |
| 1LFB | 641 | 641 | 2.5e-14 | 0.9960 | 0.0e+00 | 0.0e+00 | 12.3 |
| 1PHT | 814 | 814 | 5.7e-14 | 0.9904 | 0.0e+00 | 0.0e+00 | 17.5 |
| 1POA | 914 | 914 | 5.7e-14 | 0.9741 | 0.0e+00 | 0.0e+00 | 17.6 |
| 1AX8 | 1003 | 1003 | 6.6e-14 | 0.9927 | 0.0e+00 | 0.0e+00 | 22.8 |
| 4MBA | 1086 | 1086 | 1.3e-13 | 0.9815 | 0.0e+00 | 0.0e+00 | 19.9 |
| 1F39 | 1534 | 1534 | 2.1e-13 | 0.9976 | 0.0e+00 | 0.0e+00 | 32.4 |
| 1RGS | 2015 | 2015 | 2.4e-13 | 0.9787 | 0.0e+00 | 0.0e+00 | 39.8 |
| 1BPM | 3672 | 3672 | 3.3e-13 | 0.9781 | 0.0e+00 | 0.0e+00 | 80.1 |
| 1HMV | 7398 | 7398 | 1.4e-10 | 0.9796 | 0.0e+00 | 0.0e+00 | 223.9 |

* ID: Protein ID, TA: Total number of atoms, DA: Total number of determined atoms, RMSD: RMSD between the original and computed structure (in Å), CORR: Correlation between the original and calculated atomic fluctuation radii, SDME: Sparse distance matrix error (in Å), MDE: Maximum distance error (in Å), CPU: Total CPU time elapsed during structure determination (in seconds)

Note that the problems in Table 6.2 all have solutions which take the midpoints of the lower and upper distance bounds and the original fluctuation radii. If we start from these solutions, we can obtain the solutions to the generalized distance geometry problems immediately. However, we solved the problems without assuming any knowledge of these solutions. We applied the buildup algorithm to each problem and solved a sequence of generalized subproblems for

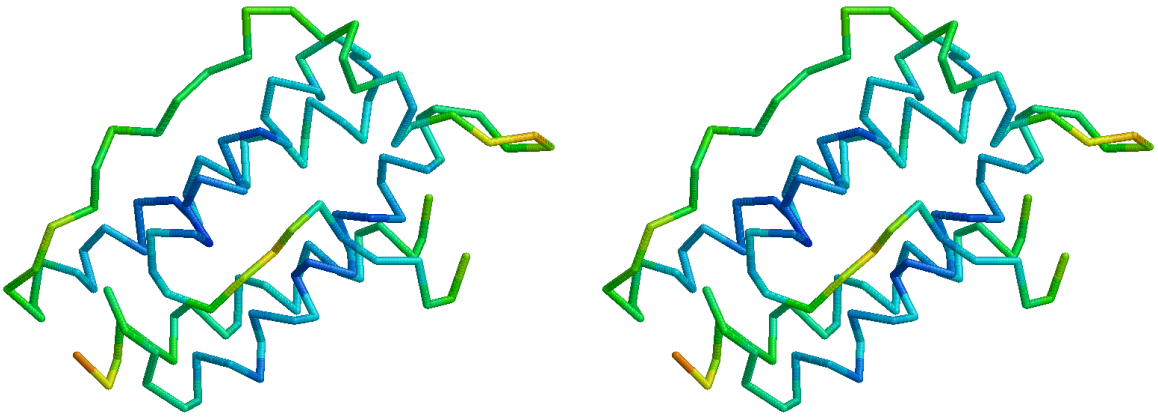


Figure 6.5: **Equilibrium structure vs. original structure:** The structure for 1AX8 on the left is obtained by solving a generalized distance geometry problem using a set of distance bounds. The one on the right is the crystal structure for protein 1AX8. The RMSD value of the two structures is 2.1e-11 Å. The colors in the pictures only represent different temperature regions.

it. What we wanted to see is if the buildup algorithm can indeed end up with a solution or an approximate solution to each of the test problems. It turned out that perfect solutions to these problems were all found because the problems were well defined and the convergence to the optimal solution to each subproblem was achieved in the buildup process. In general, this is not guaranteed as shown in our second set of test cases.

The results in Table 6.2 are for the test problems with exactly the generated distance bounds. In practice, however, distances come from either physical experiments or theoretical estimates, and must be noisy and have errors. Hence, in order to analyze the effectiveness of the algorithm, we have also tested it on more noisy data. Table 6.3 and 6.4 further demonstrate the behavior of the algorithm for distance bounds with small perturbation errors. In order to obtain these results, we have perturbed previously generated distance bounds with some small random errors. More specifically, we perturbed every generated distance bound pair, $l_{ij}$ and $u_{ij}$, by using the following formulas:

$$\begin{aligned} l_{ij} &\Leftarrow l_{ij} + 2 * \mathrm{RE} * (0.5 - \mathit{rand}) * l_{ij} \\ u_{ij} &\Leftarrow u_{ij} + 2 * \mathrm{RE} * (0.5 - \mathit{rand}) * u_{ij} \end{aligned} \tag{6.17}$$

where RE is the maximum relative error and RE = $10^{-8}$, $10^{-7}$, $10^{-6}$, $10^{-5}$, $10^{-4}$, and $10^{-3}$, and *rand* is a function which returns a number in $[0, 1]$. We have then obtained a new set of

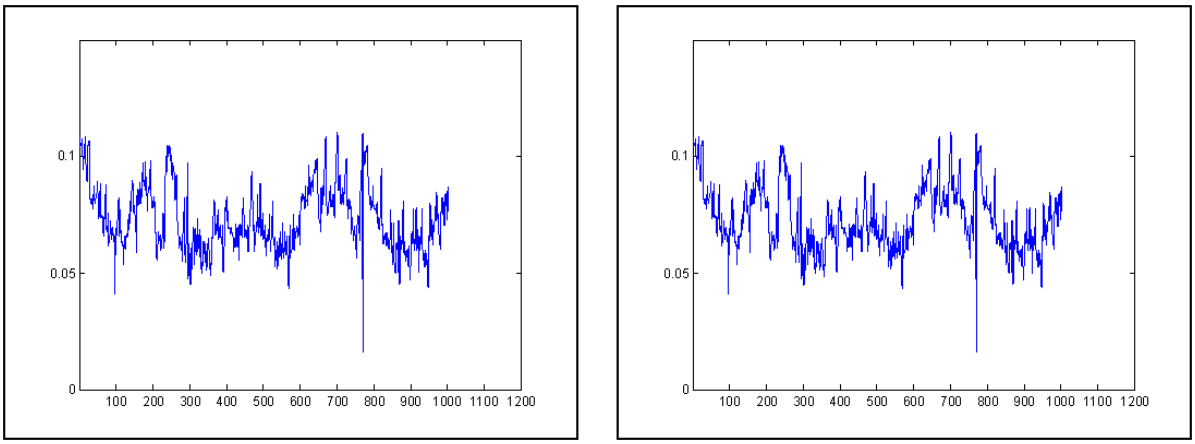



Figure 6.6: **Fluctuation radii vs. B-factors:** The graph on the left shows the fluctuation radii of the atoms extracted from the B-factors of the original crystal structure of protein 1AX8. The one on the right shows the fluctuation radii obtained by solving a generalized distance geometry problem. The two sets of values correlate very well.

upper and lower bounds for the distances. They are more erroneous and can be inconsistent. For each of these data sets, we applied the buildup algorithm again and also calculated the four error measures as done in Table 6.2.

Note that the errors are just perturbations on the distance bounds generated in the first set of test cases. The gaps between the lower and upper bounds remained large to contain corresponding atomic fluctuations. We introduced small perturbations on those bounds, just to make the solutions to the problems not as certain as the first set of test cases and to see some limits of the algorithm. Indeed, once the perturbations are increased to $10^{-3}$, we have observed large errors in some of the structures. The algorithm certainly needs further development before it can apply to real NMR data.

Table 6.3 and 6.4 show that the algorithm performed on the second set of test cases similarly as on the first set, except when either the distance data was too sparse or the relative perturbation error was increased to $10^{-3}$. They may due to the accumulation of the rounding errors or some inconsistent distance bounds. In both tables, these instances are shown in red.

More specifically, while the above test results are generally good, they reveal that the proposed algorithm can behave poorly in the presence of relative errors on the order of $10^{-3}$ and even $10^{-4}$. This can be seen in the last column of Table 6.3 for 1POA, 1AX8, and 1F39, and in the last two columns of Table 6.3 for 1PHT, 1RGS, and 1BPM, and for every column

for 1HMV. In all of these cases the structures generated by the build-up algorithm have large violations of the distance bounds, with the maximum violations in the range of 2-74 Å. The behavior is less pronounced for the tests reported in Table 6.4, where more distance information has been included. Even here, however, the buildup approach yields unacceptable structures in some cases, as seen in the last column of Table 6.4 for 1F39, 1RGS, and 1BPM, and the last two columns for 1HMV, where the maximum distance restraint violations range from 1 Å to 7.5 Å. This behavior is more pronounced as the size of the protein increases. This is most likely because of the accretion of errors and infeasibilities in the buildup process.

It is notable that the correlations between the original and computed atomic fluctuation radii are high in both tables. For all of the proteins, there is a good correlation between RMSD and SDME values. This means that when we do not know the original structure, we can still use SDME as a measure for computed structures. When compared to Table 6.3, Table 6.4 has structures determined with more accuracy. This is because of increasing density of the distance data. The more the distances, the more accurate the structure determined and the more capable to correct itself. In general, even when the problem size is large and distance data is sparse (e.g. the protein 1HMV with 7398 atoms), the deviations are still quite reasonable, which implies that the rounding errors are under control and the algorithm is relatively stable. It is also noteworthy to mention that protein structures determined by NMR are not this big in size, which increases hopes for the geometric buildup algorithm to work for real NMR distance data.

We have also tested the algorithm with distance cutoffs of 7 and 8 Å. Here we have only presented the results for distance cutoffs of 5 and 6 Å because they are close to the real NMR distance range. The distance cutoffs of 7 and 8 Å or longer allow us to generate more distances and the problems become relatively easier to solve. Indeed, in our results, the problems with these larger distance cutoffs have all been solved with a RMSD value less than $10^{-4}$ Å. It is therefore not so informative to include those results in our tables.

## 6.6 Summary and Discussion

In this chapter, we have proposed a new approach to the problem of determining an ensemble of protein structures for a given set of interatomic distance bounds. We assumed that a protein has an equilibrium structure and the atoms fluctuate around their equilibrium positions (as described by the B-factors in X-ray crystallography). Then, we formulated the problem for determining an ensemble of protein structures for a given set of interatomic distance bounds as a so-called generalized distance geometry problem as given in (6.7). The problem then becomes to find the equilibrium positions and maximal possible fluctuation radii for the atoms in the protein, subject to the condition that the fluctuations should be within the given distance bounds.

The new formulation of the problem has several advantages over those in the conventional approaches such as that in (6.2), which requires to obtain a solution set for a system of nonlinear inequalities: First, it is a much better defined problem because it requires only a single solution rather than a solution set. Second, it is a constrained optimization problem, which can be approached by an optimization method, while the problem in (6.2) is a system of nonlinear inequalities. There is no effective method for solving a system of nonlinear inequalities, even for obtaining a subset of solutions. Third, the solution of the problem can deliver an NMR structure in a form similar to an X-ray crystal structure, with a single structural file containing the coordinates and fluctuation radii (or B-factors) for the atoms. These advantages make it possible to develop an efficient algorithm for the determination of a structure using a set of interatomic distance bounds and to improve the way to represent a structural ensemble in NMR protein modeling.

In practice, there can be more than tens of thousands of variables and constraints for the problem in (6.7). A large-scale constrained optimization problem can still be difficult to solve. We have developed a geometric buildup algorithm for an approximate solution to the problem. The idea of the algorithm is to determine the positions and fluctuation radii of the atoms, one at a time, using the distance constraints from the determined atoms to the undetermined ones. In every step, only a small generalized DG subproblem needs to be solved, to find the

equilibrium position and fluctuation radius of one atom, which can be done relatively easily. By repeatedly solving such a subproblem for an undetermined atom, the coordinate vectors and fluctuation radii of all the atoms can be determined, and a solution or more accurately, an approximate solution to the generalized DG problem can be obtained.

While exciting with a novel approach and its successful implementation and testing, the work is still in an initial stage, and many issues are yet to be addressed. First, the generalized distance geometry problem (6.7) has not included the equality constraints or tightly bounded constraints, which may occur in practice when for example some pairs of atoms are connected with strong chemical bonds of almost fixed lengths. If we add such constraints to the problem, we will restrict the movement of related atoms so tightly that there is no room for the atoms to have a reasonable fluctuation radius. In order to incorporate these types of distance constraints, we need to treat them differently from those in (6.7). For example, for a pair of atoms $i$ and $j$, if there is a chemical bond of length tightly bounded by $l_{ij}$ and $u_{ij}$, we may use the following constraints for the atoms:

$$\begin{aligned} \|x_i - x_j\| &\leq u_{i,j}, \\ \|x_i - x_j\| &\geq l_{i,j}, \end{aligned} \tag{6.18}$$

without the fluctuation radii $r_i$ and $r_j$ in the formulas. In this way, the tight distance constraints for chemically bonded atoms can be satisfied while the determination of the fluctuation radii of the atoms are not affected. We have done some initial work on this issue and will report the results in Chapter 7.

Second, the generalized distance geometry problem is still a hard problem, if a global optimal solution is to be found. In fact, even if we just wanted to find a feasible solution, the problem is equivalent to a distance geometry problem with distance bounds which has been proven to be NP-hard if the bounds are tight enough [47]. We have considered only a local optimal solution to the generalized distance geometry problem for possibly reduced computational complexities. We expect that such a solution, or even an approximation, may be adequate for the description of a structural ensemble in terms of its equilibrium structure and possible fluctuation range. In theory, we have provided some results for the existence of an

optimal solution of a generalized subproblem and for the convergence of a sequence of feasible solutions for the subproblems to a feasible solution for the general problem. These results are rather weak. It would be interesting, at least theoretically, to prove a condition under which a sequence of optimal solution to the subproblems can converge to an optimal solution to the general problem.

Third, an ultimate goal of this work is to provide an effective computational tool for NMR protein structure determination. Therefore, we hope that the algorithm developed can be applied to real modeling problems, for example, to determine the structure or the structural ensemble of a novel protein using a given set of NMR distance data. The algorithm is not ready for a real application yet. The reason is that the real NMR distance data may be even sparser than that in our test cases. Then, some parts of the structures may not be determined uniquely. The distance ranges may also be much larger. It remains to be tested whether or not a meaningful approximate solution to the generalized distance geometry problem can always be found for such distance ranges. There are also known distances or angles such as the bond lengths or bond angles necessary for determining a structure but not given directly. We may need to combine our algorithm with existing modeling software such as CNS [6] or CONCOORD [12]. It can then be applied to the real data. We may need to improve the algorithm so it can deal with arbitrary (other than artificially generated) distance constraints. These all require further efforts, which we are planning to make. A simpler step is to recompute the existing NMR structures and evaluate the results. This can be done relatively easily because the distance data for these proteins are all available in public domain. The existing structures can also serve as initial structures for the solution of the corresponding generalized distance geometry problems. While the existing structures are all documented in multi-model formats, the new structures, in a form similar to that for X-ray crystal structures, may provide a very different perspective for viewing and analyzing these proteins.

Table 6.3: Error measures of structures computed with perturbed distances ($\leq 5$ Å)*

| ID | DA | | RE: $10^{-8}$ | RE: $10^{-7}$ | RE: $10^{-6}$ | RE: $10^{-5}$ | RE: $10^{-4}$ | RE: $10^{-3}$ |
|---|---|---|---|---|---|---|---|---|
| 1PTQ | 402 | RMSD | 6.1e-06 | 7.5e-05 | 4.1e-04 | 8.2e-03 | 7.0e-03 | 5.0e-02 |
| | | CORR | 0.9857 | 0.9857 | 0.9854 | 0.9651 | 0.9654 | 0.7396 |
| | | SDME | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 1.4e-02 |
| | | MDE | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 2.0e-02 |
| 1HOE | 558 | RMSD | 1.4e-06 | 1.4e-05 | 2.3e-04 | 1.8e-03 | 9.2e-03 | 2.0e-01 |
| | | CORR | 0.9692 | 0.9692 | 0.9693 | 0.9695 | 0.9608 | 0.6810 |
| | | SDME | 1.4e-07 | 2.1e-06 | 2.4e-05 | 7.8e-05 | 1.2e-03 | 1.6e-01 |
| | | MDE | 2.6e-07 | 4.2e-06 | 5.5e-05 | 1.8e-04 | 2.3e-03 | 7.3e-01 |
| 1LFB | 641 | RMSD | 4.2e-05 | 3.7e-04 | 5.5e-03 | 2.6e-02 | 8.6e-02 | 1.4e-01 |
| | | CORR | 0.9960 | 0.9959 | 0.9824 | 0.8248 | 0.6305 | 0.5698 |
| | | SDME | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 4.4e-02 | 4.7e-02 |
| | | MDE | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 1.5e-01 | 2.1e-01 |
| 1PHT | 809 | RMSD | 1.8e-05 | 1.6e-04 | 1.5e-03 | 8.8e-03 | 6.1e-01 | 3.5e+00 |
| | | CORR | 0.9903 | 0.9904 | 0.9904 | 0.9820 | 0.8179 | 0.7330 |
| | | SDME | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 4.3e-01 | 1.6e+00 |
| | | MDE | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 2.4e+00 | 8.1e+00 |
| 1POA | 914 | RMSD | 1.7e-05 | 1.8e-04 | 2.0e-03 | 2.6e-03 | 1.6e-02 | 4.0e-01 |
| | | CORR | 0.9741 | 0.9742 | 0.9749 | 0.9744 | 0.9531 | 0.8348 |
| | | SDME | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 5.1e-01 |
| | | MDE | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 2.5e+00 |
| 1AX8 | 1003 | RMSD | 2.7e-05 | 8.8e-04 | 9.9e-03 | 1.0e-02 | 6.5e-02 | 3.0e+00 |
| | | CORR | 0.9927 | 0.9928 | 0.9859 | 0.9846 | 0.8153 | 0.6649 |
| | | SDME | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 3.4e-02 | 3.2e+00 |
| | | MDE | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 7.4e-02 | 1.6e+01 |
| 4MBA | 1083 | RMSD | 3.5e-05 | 1.4e-04 | 1.6e-03 | 3.2e-03 | 1.4e-02 | 6.2e-02 |
| | | CORR | 0.9815 | 0.9815 | 0.9811 | 0.9794 | 0.9538 | 0.8274 |
| | | SDME | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 2.1e-02 |
| | | MDE | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 9.1e-02 |
| 1F39 | 1534 | RMSD | 3.7e-05 | 2.2e-04 | 2.6e-03 | 3.3e-02 | 8.4e-02 | 1.6e+01 |
| | | CORR | 0.9976 | 0.9976 | 0.9966 | 0.8936 | 0.7036 | 0.5683 |
| | | SDME | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 3.0e-02 | 4.6e+00 |
| | | MDE | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 4.9e-02 | 2.1e+01 |
| 1RGS | 2010 | RMSD | 1.4e-03 | 2.0e-03 | 6.2e-02 | 5.4e-02 | 1.0e+01 | 1.4e+01 |
| | | CORR | 0.9787 | 0.9787 | 0.9331 | 0.8734 | 0.8106 | 0.9052 |
| | | SDME | 0.0e+00 | 0.0e+00 | 7.9e-02 | 4.7e-02 | 2.6e+00 | 8.5e+00 |
| | | MDE | 0.0e+00 | 0.0e+00 | 2.3e-01 | 9.6e-02 | 2.0e+01 | 4.0e+01 |
| 1BPM | 3669 | RMSD | 2.0e-04 | 3.6e-04 | 3.4e-03 | 9.6e-03 | 1.9e+00 | 7.4e-01 |
| | | CORR | 0.9780 | 0.9780 | 0.9769 | 0.9741 | 0.8329 | 0.8770 |
| | | SDME | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 3.5e+00 | 5.2e-01 |
| | | MDE | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 1.8e+01 | 4.5e+00 |
| 1HMV | 7389 | RMSD | 1.1e+00 | 2.4e+00 | 2.3e+00 | 2.1e+00 | 2.6e+00 | 1.3e+01 |
| | | CORR | 0.9194 | 0.8137 | 0.8870 | 0.8070 | 0.7789 | 0.7770 |
| | | SDME | 1.6e+00 | 3.1e+00 | 2.2e+00 | 4.0e+00 | 4.2e+00 | 1.4e+01 |
| | | MDE | 7.8e+00 | 2.3e+01 | 1.3e+01 | 2.2e+01 | 2.4e+01 | 7.4e+01 |

* ID: Protein ID, DA: Total number of determined atoms, RE: Maximum relative error, RMSD: RMSD between the original and computed structure (in Å), CORR: Correlation between the original and calculated atomic fluctuation radii, SDME: Sparse distance matrix error (in Å), MDE: Maximum distance error (in Å)

Table 6.4: Error measures of structures computed with perturbed distances ($\leq 6$ Å)*

| ID | DA | | RE: $10^{-8}$ | RE: $10^{-7}$ | RE: $10^{-6}$ | RE: $10^{-5}$ | RE: $10^{-4}$ | RE: $10^{-3}$ |
|---|---|---|---|---|---|---|---|---|
| 1PTQ | 402 | RMSD | 2.7e-06 | 2.7e-05 | 7.3e-05 | 9.0e-04 | 6.5e-03 | 2.9e-02 |
| | | CORR | 0.9857 | 0.9857 | 0.9857 | 0.9851 | 0.9679 | 0.7805 |
| | | SDME | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 1.6e-02 |
| | | MDE | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 2.4e-02 |
| 1HOE | 558 | RMSD | 1.6e-06 | 1.3e-05 | 1.5e-04 | 4.5e-04 | 4.7e-03 | 3.3e-02 |
| | | CORR | 0.9692 | 0.9692 | 0.9693 | 0.9693 | 0.9669 | 0.8407 |
| | | SDME | 1.4e-06 | 8.9e-06 | 1.1e-04 | 4.0e-04 | 3.5e-03 | 1.9e-02 |
| | | MDE | 3.7e-06 | 2.1e-05 | 2.6e-04 | 9.6e-04 | 9.2e-03 | 4.2e-02 |
| 1LFB | 641 | RMSD | 6.8e-07 | 8.2e-06 | 6.4e-05 | 3.6e-04 | 3.4e-03 | 2.1e-02 |
| | | CORR | 0.9960 | 0.9960 | 0.9960 | 0.9959 | 0.9920 | 0.8179 |
| | | SDME | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 2.7e-02 |
| | | MDE | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 2.7e-02 |
| 1PHT | 814 | RMSD | 1.3e-06 | 1.8e-05 | 8.7e-05 | 6.4e-04 | 1.1e-02 | 1.3e-01 |
| | | CORR | 0.9904 | 0.9904 | 0.9904 | 0.9903 | 0.9741 | 0.8412 |
| | | SDME | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 2.1e-01 |
| | | MDE | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 9.5e-01 |
| 1POA | 914 | RMSD | 3.7e-07 | 5.3e-06 | 7.4e-05 | 8.8e-04 | 5.8e-03 | 3.7e-02 |
| | | CORR | 0.9741 | 0.9741 | 0.9741 | 0.9736 | 0.9656 | 0.8483 |
| | | SDME | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 4.1e-02 |
| | | MDE | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 7.7e-02 |
| 1AX8 | 1003 | RMSD | 2.0e-06 | 5.5e-06 | 8.0e-05 | 7.8e-04 | 1.6e-02 | 1.1e-01 |
| | | CORR | 0.9927 | 0.9927 | 0.9928 | 0.9928 | 0.9602 | 0.6426 |
| | | SDME | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 7.0e-02 |
| | | MDE | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 2.3e-01 |
| 4MBA | 1086 | RMSD | 1.2e-06 | 1.2e-05 | 2.0e-04 | 1.7e-03 | 1.3e-02 | 5.3e-02 |
| | | CORR | 0.9815 | 0.9815 | 0.9816 | 0.9811 | 0.9355 | 0.7942 |
| | | SDME | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 3.1e-02 |
| | | MDE | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 1.1e-01 |
| 1F39 | 1534 | RMSD | 2.1e-06 | 1.9e-05 | 6.2e-04 | 4.7e-03 | 4.4e-02 | 1.4e-01 |
| | | CORR | 0.9976 | 0.9976 | 0.9975 | 0.9917 | 0.8012 | 0.4863 |
| | | SDME | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 2.3e-02 | 2.0e-01 |
| | | MDE | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 3.3e-02 | 1.1e+00 |
| 1RGS | 2015 | RMSD | 3.6e-06 | 4.0e-05 | 4.0e-04 | 6.1e-03 | 1.5e-02 | 8.0e+00 |
| | | CORR | 0.9787 | 0.9787 | 0.9788 | 0.9759 | 0.9535 | 0.7946 |
| | | SDME | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 6.5e+00 |
| | | MDE | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 3.0e+01 |
| 1BPM | 3672 | RMSD | 7.4e-06 | 4.0e-05 | 2.0e-04 | 5.9e-03 | 8.9e-03 | 2.9e-01 |
| | | CORR | 0.9781 | 0.9781 | 0.9781 | 0.9758 | 0.9716 | 0.8663 |
| | | SDME | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 3.5e-01 |
| | | MDE | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 0.0e+00 | 5.3e+00 |
| 1HMV | 7398 | RMSD | 9.5e-04 | 9.4e-03 | 1.1e-02 | 6.6e-02 | 3.2e-01 | 1.4e+00 |
| | | CORR | 0.9797 | 0.9737 | 0.9706 | 0.8693 | 0.8316 | 0.7581 |
| | | SDME | 0.0e+00 | 0.0e+00 | 0.0e+00 | 8.8e-02 | 3.3e-01 | 1.2e+00 |
| | | MDE | 0.0e+00 | 0.0e+00 | 0.0e+00 | 4.6e-01 | 2.6e+00 | 7.5e+00 |

* ID: Protein ID, DA: Total number of determined atoms, RE: Maximum relative error, RMSD: RMSD between the original and computed structure (in Å), CORR: Correlation between the original and calculated atomic fluctuation radii, SDME: Sparse distance matrix error (in Å), MDE: Maximum distance error (in Å)

# CHAPTER 7. CONCLUSIONS AND FUTURE WORK

## 7.1 Summary

In this thesis, we have studied a well-known problem in protein modeling, the determination of the structure of a protein with a given set of interatomic distances obtained from either physical experiments (e.g. NMR spectroscopy) or theoretical estimates. A more general and abstract form of this problem is known as the distance geometry problem in mathematics [5], but it has other names in the literature as well [32, 54, 65].

Let $n$ be the number of atoms in a given protein, and $x_i = (x_{i,1}, x_{i,2}, x_{i,3})^T$ the coordinate vector for atom $i$, $i = 1, \ldots, n$. Let $\|\cdot\|$ be the Euclidean norm. Then, a distance geometry problem can be formulated as to find $x_i$, $i = 1, \ldots, n$ such that

$$\|x_i - x_j\| = d_{i,j} \quad \text{for} \quad (i,j) \in S, \tag{7.1}$$

where $d_{i,j}$ is the distance between atoms $i$ and $j$, and $S$ is a given set of $(i,j)$ pairs. The distance geometry problem can be solved in polynomial time if a complete set of exact distances is available [27]. However, it is generally intractable for a general sparse set of distances [54], and especially difficult when only sparse and inexact distance data is available [47].

We have studied the solution of the distance geometry problem within a so-called geometric buildup framework. Dong and Wu [15, 16] 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 [68] 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 [68] 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.

In practice, however, the distances are not given in their exact values: Only their rough ranges such as lower and upper bounds can be obtained because the structure fluctuates. The distance geometry 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

$$l_{i,j} \leq \|x_i - x_j\| \leq u_{i,j}, \qquad (i,j) \in S. \tag{7.2}$$

This problem may have infinitely many possible solutions, corresponding to an ensemble of structures all satisfying the given distance constraints. In NMR, it turns out to be important to not just find one of these structures but the whole ensemble of structures, because the deviations of the structures from each other in the ensemble provide important information on how the protein structure may fluctuate dynamically around its equilibrium state. This dynamic property is often as critical as the structure itself for the understanding of the function of the protein [11, 72].

The algorithm mentioned above may not be applied directly to the problem with bounds (7.2). 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.

We have developed a new approach to the problem of determining an ensemble of protein structures for a given set of interatomic distance bounds. We assumed that a protein has an equilibrium structure and the atoms fluctuate around their equilibrium positions (as described by the B-factors in X-ray crystallography). Then, we formulated the problem for determining an ensemble of protein structures for a given set of interatomic distance bounds as a so-called generalized distance geometry problem. The problem then becomes to find the equilibrium positions and maximal possible fluctuation radii for the atoms in the protein, subject to the condition that the fluctuations should be within the given distance bounds.

The new formulation of the problem has several advantages over those in the conventional approaches such as that in (7.2), which requires to obtain a solution set for a system of nonlinear inequalities: First, it is a much better defined problem because it requires only a single solution

rather than a solution set. Second, it is computationally more tractable because there are well-developed methods for solving optimization problems. Third, the solution of the problem can deliver an NMR structure in a form similar to an X-ray crystal structure, with a single structural file containing the coordinates and fluctuation radii (or B-factors) for the atoms. These advantages make it possible to develop an efficient algorithm for the determination of a structure using a set of interatomic distance bounds and to improve the way to represent a structural ensemble in NMR protein modeling.

In practice, there can be more than tens of thousands of variables and constraints for the generalized distance geometry problem. A large-scale constrained optimization problem can still be difficult to solve. We have developed a geometric buildup algorithm for the solution of the problem. The idea of the algorithm is to determine the positions and fluctuation radii of the atoms, one at a time, using the distance constraints from the determined atoms to the undetermined ones. In every step, only a small generalized distance geometry subproblem needs to be solved, to find the equilibrium position and fluctuation radius of one atom, which can be done relatively easily. By repeatedly solving such a subproblem for an undetermined atom, the coordinate vectors and fluctuation radii of all the atoms can be determined, and a solution or more accurately, an approximate solution to the generalized distance geometry problem can be obtained.

## 7.2 Recent Progress and Future Directions

The generalized distance geometry problem (6.7) has not included the equality constraints or tightly bounded constraints, which may occur in practice when for example some pairs of atoms are connected with strong chemical bonds of almost fixed lengths. If we add such constraints to the problem, we will restrict the movement of related atoms so tightly that there is no room for the atoms to have a reasonable fluctuation radius. In order to incorporate these types of distance constraints, we need to treat them differently from those in (6.7). For example, for a pair of atoms $i$ and $j$, if there is a chemical bond of length tightly bounded by

$l_{i,j}$ and $u_{i,j}$, we may use the following constraints for the atoms:

$$\begin{aligned} \|x_i - x_j\| &\leq u_{i,j}, \\ \|x_i - x_j\| &\geq l_{i,j}, \end{aligned} \tag{7.3}$$

without the fluctuation radii $r_i$ and $r_j$ in the formulas. In this way, the tight distance constraints for chemically bonded atoms can be satisfied while the determination of the fluctuation radii of the atoms are not affected.

We have modified our algorithm so that it can also handle the tightly bounded distance constraints. For example, if there is a chemical bond between two atoms, we can consider the corresponding distance constraint as an equality constraint, and the generalized distance geometry problem then becomes

$$\begin{aligned} \max_{x_i, r_i} \ & \sum_{i=1}^{n} r_i \\ \text{subject to } & \|x_i - x_j\| + r_i + r_j \leq u_{i,j} \\ & \|x_i - x_j\| - r_i - r_j \geq l_{i,j}, \quad (i,j) \in S_1, \\ & \|x_i - x_j\| = d_{i,j}, \quad (i,j) \in S_2, \\ & r_i \geq 0, \ \ i = 1, \ldots, n. \end{aligned} \tag{7.4}$$

Similarly, the generalized subproblems for determining the atom $l+1$ can be defined as

$$\begin{aligned} \max_{x_{l+1}, r_{l+1}} \ & r_{l+1} \\ \text{subject to } & \|x_i - x_{l+1}\| + r_i + r_{l+1} \leq u_{i,l+1} \\ & \|x_i - x_{l+1}\| - r_i - r_{l+1} \geq l_{i,l+1}, \quad i = 1, \ldots, k, \\ & \|x_i - x_{l+1}\| = d_{i,l+1}, \quad i = k+1, \ldots, l, \\ & r_{l+1} \geq 0. \end{aligned} \tag{7.5}$$

The subproblem (7.5) is different than (6.8) and needs to be treated with caution. In order to obtain reasonable initial fluctuation radius at each step of the buildup, the algorithm should choose the atoms with more nonbonded connections first, as the initial radii will always be determined by the inequality constraints. This means that there should be no chemical

bond among the first four atoms to be determined, and during the geometric buildup process, the atoms with many bonded connections should not be determined until they have enough number of distance bounds from the determined ones.

We have tested our modified algorithm with the same set of protein structures used in Chapter 6. Table 7.1 contains the RMSD values of the obtained structures (compared with their original structures), correlation coefficients between the original and calculated atomic fluctuation radii, SDME and MDE values between the original and recalculated distance data.

Table 7.1: Error measures of structures computed with mixed constraints *

| | | $\leq 5$ Å | | | | |
|---|---|---|---|---|---|---|
| ID | TA | DA | RMSD | CORR | SDME | MDE |
| 1PTQ | 402 | 402 | 2.4e-14 | 0.9857 | 3.9e-15 | 1.7e-14 |
| 1HOE | 558 | 558 | 3.7e-14 | 0.9692 | 8.7e-15 | 8.6e-14 |
| 1LFB | 641 | 641 | 2.1e-12 | 0.9960 | 3.0e-13 | 2.8e-12 |
| 1PHT | 814 | 809 | 4.2e-12 | 0.9903 | 6.2e-13 | 5.5e-12 |
| 1POA | 914 | 914 | 2.0e-13 | 0.9592 | 4.6e-14 | 5.4e-13 |
| 1AX8 | 1003 | 1003 | 1.3e-13 | 0.9940 | 2.2e-14 | 1.8e-13 |
| 4MBA | 1086 | 1083 | 1.4e-12 | 0.9815 | 3.2e-13 | 3.5e-12 |
| 1F39 | 1534 | 1534 | 2.6e-13 | 0.9976 | 3.8e-14 | 5.5e-13 |
| 1RGS | 2015 | 2010 | 4.7e-13 | 0.9786 | 9.0e-14 | 1.5e-12 |
| 1BPM | 3672 | 3669 | 4.2e-13 | 0.9781 | 5.4e-14 | 6.5e-13 |
| 1HMV | 7398 | 7389 | 8.5e-11 | 0.9904 | 1.9e-11 | 4.7e-10 |
| | | $\leq 6$ Å | | | | |
| ID | TA | DA | RMSD | CORR | SDME | MDE |
| 1PTQ | 402 | 402 | 1.6e-14 | 0.9857 | 2.7e-15 | 1.2e-14 |
| 1HOE | 558 | 558 | 4.9e-14 | 0.9692 | 8.1e-15 | 5.9e-14 |
| 1LFB | 641 | 641 | 2.9e-14 | 0.9960 | 3.9e-15 | 3.4e-14 |
| 1PHT | 814 | 814 | 6.3e-14 | 0.9904 | 3.4e-15 | 1.5e-14 |
| 1POA | 914 | 914 | 6.5e-14 | 0.9592 | 3.7e-15 | 2.2e-14 |
| 1AX8 | 1003 | 1003 | 5.1e-14 | 0.9940 | 3.5e-15 | 1.6e-14 |
| 4MBA | 1086 | 1086 | 1.3e-13 | 0.9815 | 6.1e-15 | 4.7e-14 |
| 1F39 | 1534 | 1534 | 8.3e-13 | 0.9976 | 1.6e-13 | 2.1e-12 |
| 1RGS | 2015 | 2015 | 2.3e-13 | 0.9787 | 2.0e-14 | 2.5e-13 |
| 1BPM | 3672 | 3672 | 1.5e-13 | 0.9781 | 2.1e-14 | 3.5e-13 |
| 1HMV | 7398 | 7398 | 8.0e-13 | 0.9904 | 1.0e-13 | 1.5e-12 |

* ID: Protein ID, TA: Total number of atoms, DA: Total number of determined atoms, RMSD: RMSD between the original and computed structure (in Å), CORR: Correlation between the original and calculated atomic fluctuation radii, SDME: Sparse distance matrix error (in Å), MDE: Maximum distance error (in Å)

All these error measures are obtained by using the modified algorithm on the mixed distance constraints, namely the equality and inequality constraints corresponding to the chemical bonds

and nonbonded pairs, respectively. From Table 7.1, we observe that the modified algorithm behaves as good as the original algorithm. The RMSD, SDME and MDE values turn out to be very small, in the order of $10^{-10}$ at most. The correlation values (CORR) in Table 7.1 show that the original and the calculated fluctuation radii are perfectly correlated.

In practice, however, there are some other tightly bounded constraints in proteins, for example, the distances corresponding to bond angles or aromatic rings in the side chains. It would definitely be interesting to treat those constraints too as tightly bounded constraints in problems (7.4) and (7.5). This work is still in progress, and we will show some initial results in a future paper.

The results in Table 7.1 have motivated us to apply the modified algorithm to a real distance data coming from NMR. We have modified the algorithm further so that it can handle the NOE distance restraints. We have downloaded an NMR file of a protein (2KNX) of 661 atoms from BioMagResBank. The available NOE restraints were only 1.03% of all distances. These distances, however, were among certain hydrogen atoms only, and hence not sufficient for the determination of all atoms. We have also added the distances corresponding to bond lengths and angles as equality constraints, which were calculated from the coordinates in the PDB file of the protein. The total number of available distances then reached 1.94% of all distances, which is yet a very sparse distance set. By using all these available distances, we have applied our modified algorithm to the determination of this protein. We have determined 448 of 661 atoms with their fluctuation radii, with an average NOE distance violation (SDME) of 0.98 Å. The algorithm failed to determine 213 of the atoms though, because the distance data is very sparse that some of the atoms have only two or three connections to other atoms. Therefore, they will not be determined by a general geometric buildup approach, since it assumes the availability of four determined atoms at every step of the buildup process. This work is in progress too, so in order to make a fair judgment, we need to find the ways to determine those remaining atoms in the protein with connections less than four. We hope to complete this work soon and show the results in a future paper as well [62].

An ultimate goal of this work is to provide an effective computational tool for NMR protein

structure determination. Therefore, we hope that the algorithm developed can work for real modeling problems, for example, to determine the structure of a novel protein using a given set of NMR distance constraints. We need to combine our algorithm with an existing modeling software such as CNS [6] or CONCOORD [12] so it can be applied to real NMR data, followed by energy minimization. Energy minimization can also be implemented by including proper energy terms in the objective function of the generalized distance geometry problem.

A short term goal is to recompute the existing NMR structures using our method as described above. This can be done relatively easily because the existing structures can serve as initial solutions for the generalized distance geometry problems. While the existing structures are all documented in a multi-model format, the new structures, in a similar form to that for X-ray crystal structures, may provide an alternative perspective for viewing and analyzing these proteins [62].

# APPENDIX. BACKGROUND MATERIAL

## Vectors and Matrices

In this thesis, we work with vectors and matrices whose components are real numbers. Vectors are denoted by lowercase letters, and matrices by uppercase letters. The space of real vectors of length $n$ is denoted by $\mathbb{R}^n$, and the space of real $m \times n$ matrices is denoted by $\mathbb{R}^{m\times n}$.

Given a vector $x \in \mathbb{R}^n$, we use $x_i$ to denote its $i$th component, and assume that $x$ is a column vector such that its transpose, denoted by $x^T$, is a row vector, i.e. $x = (x_1, \ldots, x_n)^T$. Now, let $x = (x_1, \ldots, x_n)^T$ and $y = (y_1, \ldots, y_n)^T$ be two vectors of the same dimension. Then,

$$x \pm y = (x_1 \pm y_1, \ldots, x_n \pm y_n)^T. \tag{A.1}$$

If $\alpha$ is a scalar, then

$$\alpha x = (\alpha x_1, \ldots, \alpha x_n)^T. \tag{A.2}$$

If $x, y \in \mathbb{R}^n$, the standard inner product is

$$x \cdot y = x^T y = \sum_{i=1}^{n} x_i y_i. \tag{A.3}$$

Given a matrix $A \in \mathbb{R}^{m\times n}$, we specify its components by double scripts as $a_{ij}$, $i = 1, \ldots, m$ and $j = 1, \ldots, n$. Let $x \in \mathbb{R}^n$ be a vector and let $A \in \mathbb{R}^{m\times n}$ be a matrix. Then the matrix-vector product $b = Ax$ is the $m$-dimensional column vector defined as follows:

$$b_i = \sum_{j=1}^{n} a_{ij}\, x_j, \quad i = 1, \ldots, m. \tag{A.4}$$

If $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{n\times l}$ are two matrices, the matrix-matrix product $C = AB$ is a matrix in $\mathbb{R}^{m\times l}$ defined by

$$c_{ij} = \sum_{k=1}^{n} a_{ik}\, b_{kj}, \quad i = 1, \ldots, m,\ j = 1, \ldots, l. \tag{A.5}$$

The transpose of an $m \times n$ matrix $A$, denoted by $A^T$, is the $n \times m$ matrix such that $a_{ij}^T = a_{ji}$, $i = 1, \ldots, n$, $j = 1, \ldots, m$. A matrix $A$ is said to be *square* if $m = n$. A square matrix $A$ is *symmetric* if $A^T = A$. The diagonal of the matrix $A \in \mathbb{R}^{m \times n}$ consists of the elements $a_{ii}$, for $i = 1, \ldots, \min(m, n)$. $A$ is called *diagonal* if $a_{ij} = 0$ whenever $i \neq j$. The *identity* matrix, denoted by $I$, is the square diagonal matrix whose diagonal elements are all 1.

A square matrix $A \in \mathbb{R}^{n \times n}$ is called *nonsingular* if there exists an $n \times n$ matrix $B$ such that $AB = BA = I$. We denote $B$ by $A^{-1}$ and call it the *inverse* of $A$. For a nonsingular matrix $A \in \mathbb{R}^{n \times n}$ and for any vector $b \in \mathbb{R}^n$, there exists $x \in \mathbb{R}^n$ such that $Ax = b$. A square matrix $Q$ is *orthogonal* if it satisfies $QQ^T = Q^TQ = I$. In other words, the inverse of an orthogonal matrix is its transpose.

A scalar value $\lambda$ is an *eigenvalue* of the $n \times n$ matrix $A$ if there is a nonzero vector $v \in \mathbb{R}^n$ such that

$$Av = \lambda v. \tag{A.6}$$

The vector $v$ is called an *eigenvector* of $A$. The matrix $A$ is nonsingular if none of its eigenvalues are zero. The eigenvalues of symmetric matrices are all real numbers, while nonsymmetric matrices may have imaginary eigenvalues. If the matrix is positive definite as well as symmetric, then its eigenvalues are all positive real numbers.

The trace of an $n \times n$ matrix $A$ is defined by

$$\text{trace}(A) = \sum_{i=1}^{n} a_{ii}. \tag{A.7}$$

If the eigenvalues of $A$ are denoted by $\lambda_1, \ldots, \lambda_n$, it can be shown that

$$\text{trace}(A) = \sum_{i=1}^{n} \lambda_i, \tag{A.8}$$

that is, the trace of the matrix is the sum of its eigenvalues.

## Norms

A *norm* is a mapping $\| \cdot \|$ from $\mathbb{R}^n$ to the nonnegative real numbers that satisfies the following:

$$\begin{aligned} &\text{(i)} \quad \|x\| = 0 \;\Leftrightarrow\; x = 0 \;\text{ for all }\; x \in \mathbb{R}^n, \\ &\text{(ii)} \quad \|\alpha x\| = |\alpha|\|x\| \;\text{ for all }\; \alpha \in \mathbb{R} \;\text{ and }\; x \in \mathbb{R}^n, \\ &\text{(iii)} \quad \|x+y\| \le \|x\| + \|y\| \;\text{ for all } x, y \in \mathbb{R}^n. \end{aligned} \tag{A.9}$$

For any vector $x \in \mathbb{R}^n$, one can define the following norms:

$$\begin{aligned} \|x\|_1 &= \sum_{i=1}^{n} |x_i|, \\ \|x\|_2 &= \left( \sum_{i=1}^{n} |x_i|^2 \right)^{1/2}, \\ \|x\|_\infty &= \max_{i=1,\ldots,n} |x_i|. \end{aligned} \tag{A.10}$$

The norm $\|\cdot\|_2$ is often called the *Euclidean* norm, and it satisfies the Cauchy-Schwarz inequality

$$|x^T y| \le \|x\|_2 \|y\|_2, \tag{A.11}$$

with equality if and only if one of these vectors is a nonnegative multiple of the other.

We can also derive definitions for certain matrix norms from these vector norm definitions. If we let $\|\cdot\|$ be one of the three norms listed in (A.10), we define the corresponding matrix norm as

$$\|A\| = \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|}. \tag{A.12}$$

The matrix norms defined in this way are said to be *consistent* with the vector norms. Explicit formulas for these norms are as follows:

$$\begin{aligned} \|A\|_1 &= \max_{j=1,\ldots,n} \sum_{i=1}^{m} |a_{ij}|, \\ \|A\|_2 &= \{\text{largest eigenvalue of } A^T A\}^{1/2}, \\ \|A\|_\infty &= \max_{i=1,\ldots,m} \sum_{j=1}^{n} |a_{ij}|. \end{aligned} \tag{A.13}$$

The Frobenius norm $\|A\|_F$ of $A$ is defined by

$$\|A\|_F = \left( \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij}^2 \right)^{1/2} = \text{trace}(A^T A). \tag{A.14}$$

This norm is useful for many purposes, but it is not consistent with any vector norm.

## Singular Value Decomposition

Let $A \in \mathbb{R}^{m \times n}$ and assume that $m \leq n$. Then, $AA^T$ is a symmetric and positive semidefinite matrix. Let $\sigma_1^2, \ldots, \sigma_m^2$ be the eigenvalues of $AA^T$ and $\sigma_1 \geq \ldots \geq \sigma_m$. Let $u_i$ be the eigenvectors of $AA^T$ corresponding to the eigenvalues $\sigma_i^2$, $i = 1, \ldots, m$. Then,

$$AA^T u_i \;=\; \sigma_i^2 u_i, \quad i = 1, \ldots, m. \tag{A.15}$$

Let $A^T u_i = \sigma_i v_i$. Then,

$$Av_i \;=\; \sigma_i u_i, \quad i = 1, \ldots, m. \tag{A.16}$$

Define an $m \times m$ orthogonal matrix $U = [u_1, \ldots, u_m]$ and an $n \times n$ orthogonal matrix $V = [v_1, \ldots, v_m, v_{m+1}, \ldots, v_n]$ with additional orthogonal vectors $v_{m+1}, \ldots, v_n$. Let $\Sigma$ be an $m \times n$ diagonal matrix with $m$ diagonal elements $\sigma_1, \ldots, \sigma_m$. Then,

$$AV \;=\; U\Sigma \quad \text{or} \quad A = U\Sigma V^T. \tag{A.17}$$

Here, $A = U\Sigma V^T$ is called a *singular value decomposition (SVD)* of $A$. The diagonal elements $\sigma_1, \ldots, \sigma_m$ are called the *singular values* of $A$. Several important properties related to the SVD of a matrix can be stated in the following theorems [71].

**Theorem A.1.** *An matrix $A \in \mathbb{R}^{m \times n}$ can always be factorized as $A = U\Sigma V^T$, where $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ are orthogonal matrices, and $\Sigma \in \mathbb{R}^{m \times n}$ is a diagonal matrix with $m$ nonnegative diagonal elements $\sigma_1, \ldots, \sigma_m$.*

**Theorem A.2.** *Assume that the singular values of $A \in \mathbb{R}^{m \times n}$ can be ordered in such a way that $\sigma_1 \geq \ldots \geq \sigma_r > \sigma_{r+1} = \ldots = \sigma_m = 0$. Then, the rank of $A$ is equal to $r$, that is, the number of positive singular values of $A$.*

**Theorem A.3.** *Let $A \in \mathbb{R}^{m \times n}$ be a matrix with singular value decomposition $A = U\Sigma V^T$. Define $A^+ = V\Sigma^{-1}U^T$. Then, $A^+$ is called the pseudoinverse of $A$, and $x = A^+ b$ minimizes $\|Ax - b\|_2$ or, in other words, solves a least squares problem for the equation $Ax = b$.*

**Theorem A.4.** *Let $A \in \mathbb{R}^{m \times n}$ be a matrix with singular value decomposition $A = U\Sigma V^T$. Define $\Sigma_k$ to be an $m \times n$ diagonal matrix with only first $k$ nonzero diagonal elements of $\Sigma$.*

*Then, $B = U\Sigma_k V^T$ minimizes $\|A - B\|_F$ for all matrices $B$ of rank $k$ or, in other words, makes the best approximation to $A$ by a matrix of rank $k$.*

## The Coordinate Root-Mean-Square Deviation

The coordinate root-mean-square deviation (RMSD) has been widely used in protein modeling, for comparing and validating protein structures. It has also been an important tool for structural classification, motif recognition, and structure prediction, where a large number of different proteins must be aligned and compared [71].

Let $X = [x_1^T; \ldots; x_n^T]$ and $Y = [y_1^T; \ldots; y_n^T]$ be two $n \times 3$ coordinate matrices for two lists of atoms in proteins $A$ and $B$, respectively, where $x_i = (x_{i,1}, x_{i,2}, x_{i,3})^T$ is the coordinate vector of the $i$th atom selected from protein $A$ to be compared with $y_i = (y_{i,1}, y_{i,2}, y_{i,3})^T$, the coordinate vector of the $i$th atom selected from protein $B$. Assume that $X$ and $Y$ have been translated so that their geometric centers are located at the same position, say at the origin. Then, the structural similarity between the two proteins can be measured by using the coordinate RMSD of the structures as defined by the following:

$$\text{RMSD} \;=\; \min_{Q} \|X - YQ\|_F / \sqrt{n}, \tag{A.18}$$

where $Q$ is a $3 \times 3$ orthogonal rotation matrix, and $\|\cdot\|_F$ is the matrix Frobenius norm. Based on this definition, the RMSD of two structures $X$ and $Y$ is essentially the smallest average coordinate error of the structures for all possible rotations $Q$ of structure $Y$ to fit structure $X$ (see Fig. A.1).

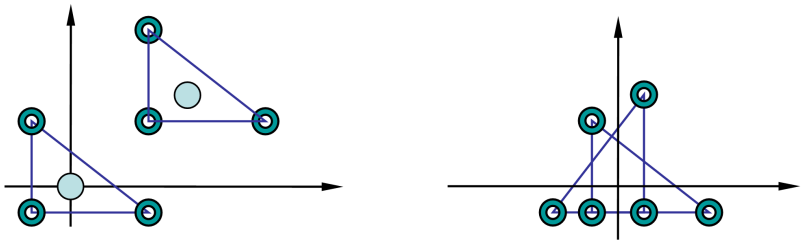


Figure A.1: **Translation and rotation:** The coordinate root-mean-square deviation (RMSD) can be calculated only after aligning the structures with proper translation and rotation.

The RMSD calculation requires the solution of an optimization problem, as suggested in its definition. The optimization problem is not trivial to solve if a conventional optimization method is to be used [71]. Fortunately, an analytical solution to the problem can actually be obtained with some simple linear algebraic calculations as follows.

Note that

$$\|X - YQ\|_F^2 \;=\; \operatorname{trace}(X^T X) + \operatorname{trace}(Y^T Y) - 2\operatorname{trace}(Q^T Y^T X). \tag{A.19}$$

Therefore, minimizing the square of $\|X - YQ\|_F$ is equivalent to maximizing $\operatorname{trace}(Q^T Y^T X)$. Let $C = Y^T X$. Let $C = U\Sigma V^T$ be the singular value decomposition of $C$. Then,

$$\operatorname{trace}(Q^T Y^T X) = \operatorname{trace}(V^T Q^T U\Sigma) \leq \operatorname{trace}(\Sigma). \tag{A.20}$$

It follows that $Q = UV^T$ maximizes $\operatorname{trace}(Q^T Y^T X)$ and therefore minimizes the square of $\|X - YQ\|_F$ [71].

# BIBLIOGRAPHY

[1] Berg J. M., Tymoczko J. L. and Stryer L., *Biochemistry*, W. H. Freeman, 2006.

[2] Berman H. M., Westbrook J., Feng Z., Gilliland G., Bhat T. N., Weissig H., Shindyalov I. N. and Bourne P. E., *The Protein Data Bank*, Nucleic Acids Research 28, 235-242, 2000.

[3] Biswas P., Liang T., Wang T. and Ye Y., *Semidefinite programming based algorithms for sensor network localization*, ACM J on Transactions on Sensor Networks, 2, 188-220, 2006.

[4] Biswas P., Toh K. and Ye Y., *A distributed SDP approach for large scale noisy anchor-free graph realization with applications to molecular conformation*, SIAM J. on Sci. Comp., 30, 1251-1277, 2008.

[5] Blumenthal L. M., *Theory and Applications of Distance Geometry*, Oxford Clarendon Press, 1953.

[6] Brünger A. T., Adams P. D., Clore G. M., Gros P., Grosse-Kunstleve R. W., Jiang J. -S., Kuszewski J., Nilges N., Pannu N. S., Read R. J., Rice L. M., Simonson T. and Warren G. L., *Crystallography and NMR System (CNS), A new software suite for macromolecular structure determination*, Acta Cryst. D54, 905-921, 1998.

[7] Cavanagh J., Fairbrother W. J., Palmer A. G. and Skelton N. J., *Protein NMR Spectroscopy: Principals and Practice*, Academic Press, 2006.

[8] Cayley A., *A theorem in the geometry of position*, Cambridge Math. J., II, 267-271, 1841.

[9] Chandonia J. M. and Brenner S. E., *The impact of structural genomics: expectations and outcomes*, Science 311, 347-351, 2006.

[10] Creighton T. E., *Proteins: Structures and Molecular Properties*, 2nd Edition. Freeman and Company, 1993.

[11] Crippen G. M. and Havel T. F., *Distance Geometry and Molecular Conformation*, John Wiley & Sons, 1988.

[12] de Groot B. L., van Aalten D. M. F., Scheek R. M., Amadei A., Vriend G. and Berendsen H. J. C., *Prediction of protein conformational freedom from distance constraints*, Proteins: Structure, Function, and Genetics, 29, 240-251, 1997.

[13] Dennis J. E. and Schnabel R. B., *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, SIAM, 1996.

[14] Deza E. and Deza M., *Dictionary of Distances*, Elsevier, 2006.

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

[16] Dong Q. and Wu Z., *A geometric buildup algorithm for solving the molecular distance geometry problem with sparse distance data*, J. Global Optim., 26, 321-333, 2003.

[17] Doreleijers J. F., Rulmann J. A. C. and Katein R., *Quality assessment of NMR structures: A statistical survey*, J. Mol. Biol. 281, 149-164, 1998.

[18] Drenth J., *Principals of Protein X-ray Crystallography*, Springer, 2006.

[19] Eckart C. and Young G., *The approximation of one matrix by another of lower rank*, Psychometrika 1, 211-218, 1936.

[20] Fletcher R., *Practical Methods of Optimization*, Wiley, 2000.

[21] Garey M. R. and Johnson D. S., *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman & Co., 1979.

[22] Glunt W. , Hayden T. L., Hong S. and Wells J., *An alternating projection algorithm for computing the nearest Euclidean distance matrix*, SIAM J. Mat. Anal. Appl., 11, 589-600, 1990.

[23] Glunt W., Hayden T. L. and Raydan M. , *Molecular conformations from distance matrices*, J. Comput. Chem., 14, 114-120, 1993.

[24] Golub G. H. and van Loan C. F., *Matrix Computations*, Johns Hopkins University Press, 1989.

[25] Gower J. C., *Euclidean distance geometry*, Math. Sci., 7, 1-14, 1982.

[26] Grosso A., Locatelli M. and Schoen F., *Solving molecular distance geometry problems by global optimization algorithms*, J. Comput. Opt. and Appl., 43, 23-37, 2009.

[27] Havel T. F., *Distance geometry*, in Encyclopedia of Nuclear Magnetic Resonance, D. M. Grant and R. K. Harris, eds., John Wiley & Sons, 1701-1710, 1995.

[28] Havel T., *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, 1991.

[29] Havel T. F., *Distance geometry: Theory, algorithms, and chemical applications*, in Encyclopedia of Computational Chemistry, John Wiley & Sons, 1-20, 1998.

[30] Havel T. and Wüthrich K., *An evaluation of the combined use of nuclear magnetic resonance and distance geometry for the determination of protein conformations in solution*, J. Mol. Biol., 182, 281-294, 1985.

[31] Havel T., Knutz I. and Crippen G., *The theory and practice of distance geometry*, Bull. Math. Biol., 45, 665-720, 1983.

[32] Hendrickson B., *Conditions for unique graph realizations*, SIAM J. Comput., 21, 65-84, 1992.

[33] Hendrickson B., *The molecule problem: Exploiting structure in global optimization*, SIAM J. Optim., 5, 835-857, 1995.

[34] Hendrickson B., *The Molecular Problem: Determining Conformation from Pairwise Distances*, Ph.D. thesis, Cornell University, 1991.

[35] Hooft R. W., Vriend G., Sander C. and Abola E. E., *Errors in protein structures*, Nature 381, 272, 1996.

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

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

[38] Kearsly A., Tapia R. and Trosset M., *Solution of the metric STRESS and SSTRESS problems in multidimensional scaling by Newton's method*, Computational Statistics 13, 369-396, 1998.

[39] Klock H. and Buhmann J. M., *Multidimensional scaling with deterministic annealing*, in Lecture Notes in Computer Science 1223: Energy Minimization Methods in Computer Vision and Pattern Recognition, M Pilillo and E. R. Hancock, eds., Springer-Verlag, 246-260, 1997.

[40] Kumar A., Ernst R. R. and Wüthrich K., *A two-dimensional nuclear Overhauser enhancement (2D NOE) experiment for the elucidation of complete proton-proton cross-relaxation networks in biological macromolecules*, Biochem. Biophys. Res. Commun. 95, 1-6, 1980.

[41] Kuszewski J., Niles M. and Brünger A. T., *Sampling and efficiency of metric matrix distance geometry: A novel partial metrization algorithm*, J. Biomol. NMR 2, 33-56, 1992.

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

[43] Le Thi Hoai A., *Solving large-scale molecular distance geometry problems by a smoothing technique via the Gaussian transform and d.c. programming*, Journal of Global Optimization, 27, 375-397, 2003.

[44] Mal T. K., Bagby S. and Ikura M., *Protein Structure Calculation from NMR Data*, Methods in Molecular Biology, Book Title: Calcium-Binding Protein Protocols, Vol 2: Methods and Techniques, 173, 267-283, 2002.

[45] Menger K., *Untersuchungen ueber allgemeine Metrik*, Math. Ann., 100, 75-163, 1928.

[46] Menger K., *New foundation of Euclidean geometry*, Am. J. Math., 53, 721-745, 1931.

[47] Moré J. and Wu Z., *$\epsilon$-Optimal solutions to distance geometry problems via global continuation*, in Global Minimization of Non-Convex Energy Functions: Molecular Conformation and Protein Folding, P. M. Pardalos, D. Shalloway, and G. Xue, eds., American Mathematical Society, 151-168, 1996.

[48] Moré J. and Wu Z., *Global continuation for distance geometry problems*, SIAM J. Optim., 7, 814-836, 1997.

[49] Moré J. and Wu Z., *Distance geometry optimization for protein structures*, J. Global Optim. 15, 219-234, 1999.

[50] Moré J. J. and Wright S. J., *Optimization Software Guide*, SIAM, 1993.

[51] Nabuurs S. B., Spronk C. A., Vuister G. W. and Vriend G., *Traditional biomolecular structure determination by NMR spectroscopy allows for major errors*, PLoS Comput. Biol. 2, 71-79, 2006.

[52] Nocedal J. and Wright S. J., *Numerical Optimization*, Springer, 2002.

[53] Russell R. B. and Eggleston D. S., *New roles for structure in biology and drug discovery*, Nat. Struct. Biol. 7,928-930, 2000.

[54] Saxe J. B., *Embeddability of weighted graphs in k-space is strongly NP-hard*, in Proc. 17th Allerton Conference in Communications, Control and Computing, 480-489, 1979.

[55] Schlick T., *Molecular Modeling and Simulation: An Interdisciplinary Guide*, Springer, 2003.

[56] Schoenberg I. J., *Remarks to Maurice Fréchet's Article "Sur la définition axiomatique d'une classe d'espace distanciés vectoriellement applicable sur l'espace de Hilbert"*, Annals Math., 36, 724-732, 1935.

[57] Schwieters C. D., Kuszewski J. J., Tjandra N. and Clore G. M., *The Xplor-NIH NMR Molecular Structure Determination Package*, J. Magn. Res., 160, 66-74, 2003.

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

[59] Sippl M. and Scheraga H., *Cayley-Menger coordinates*, Proc. Natl. Acad. Sci. USA, 83, 2283-2287, 1986.

[60] Sit A., Wu Z. and Yuan Y., *A geometric buildup algorithm for the solution of the distance geometry problem using least-squares approximation*, Bull. Math. Biol. 71, 1914-1933, 2009.

[61] Sit A. and Wu Z., *Solving a generalized distance geometry problem for protein structure determination*, Bull. Math. Biol., submitted, 2010.

[62] Sit A., Kloczkowski A., Jernigan R. L. and Wu Z., *Refinement of protein NMR structural ensembles*, in preparation, 2010.

[63] Snyder D. A., Bhattacharya A., Huang Y. J. and Montelione G. T., *Assessing precision and accuracy of protein structures derived from NMR data*, Proteins 59, 655-661, 2005.

[64] Spronk C, A. E. M., Natuurs S. B., Bonvin A. M. J. J., Krieger E., Vuister G. W. and Vriend G., *The precision of NMR structure ensembles revisited*, J. Biomol. NMR 25, 225-234, 2003.

[65] Torgerson W. S., *Theory and Method of Scaling*, John Wiley & Sons, 1958.

[66] Trefethen L. N. and Bau D., *Numerical Linear Algebra*, SIAM, Philadelphia, 1997.

[67] Trosset M., *Applications of multidimensional scaling to molecular conformation*, Computing Sciences and Statistics 29, 148-152, 1998.

[68] Wu D. and Wu Z., *An updated geometric buildup algorithm for solving the molecular distance geometry problem with sparse distance data*, J. Global Optim., 37, 661-673, 2007.

[69] Wu D., Wu Z. and Yuan Y., *Generating rigid protein structures with sparse sets of inter-atomic and inter-residual distances*, Optimization Letters 2, 319-331, 2008.

[70] Wu D., Wu Z. and Yuan Y., *The solution of the distance geometry problem in protein modeling via geometric build-up*, Institute for Mathematics and its Applications.

[71] Wu Z., *Lecture Notes on Computational Structural Biology*, World Scientific Publising Company, 2008.

[72] Wüthrich K., *NMR in Structural Biology*, World Scientific Publishing Company, 1995.

[73] Wüthrich K., *NMR of Proteins and Nucleic Acids*, Wiley, New York, 1986.

[74] Yoon J. M., Gad Y. and Wu Z., *Mathematical modeling of protein structure using distance geometry*, Technical report, Department of Computational & Applied Mathematics, Rice University, 2000.

[75] Young G. and Householder A. S., *Discussion of a set of points in terms of their mutual distances*, Psychometrika, 3, 19-22, 1938.

[76] Zou Z., Byrd R. H. and Schnabel R. B., *A stochastic/perturbation global optimization algorithm for distance geometry problems*, J. Global Optim., 11, 91-105, 1997.