THE SOLUTION OF THE BOUNDARY-VALUE PROBLEMS FOR THE SIMULATION OF TRANSITION OF PROTEIN CONFORMATION

By

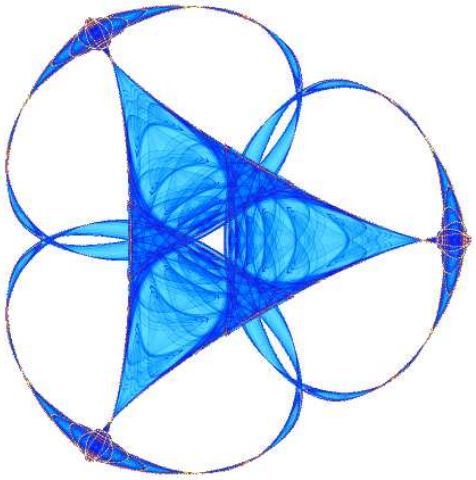
Peter Vedell

and

Zhijun Wu

IMA Preprint Series # 2191

(March 2008)



INSTITUTE FOR MATHEMATICS AND ITS APPLICATIONS

UNIVERSITY OF MINNESOTA

400 Lind Hall
207 Church Street S.E.
Minneapolis, Minnesota 55455–0436

Phone: 612-624-6066 Fax: 612-626-7370
URL: http://www.ima.umn.edu

INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 1, Number 1, Pages 1–24

THE SOLUTION OF THE BOUNDARY-VALUE PROBLEMS FOR THE SIMULATION OF TRANSITION OF PROTEIN CONFORMATION

PETER VEDELL[1] AND ZHIJUN WU[2]

Abstract. Under certain kinetic or thermodynamic conditions, proteins make large conformational changes, formally called state transitions, resulting in significant changes in their chemical or biological functions. These dynamic properties of proteins can be studied through molecular dynamics simulation. However, in contrast to conventional dynamics simulation protocols where an initial-value problem is solved, the simulation of transition of protein conformation can be done by solving a boundary-value problem, with the beginning and ending states of the protein as the boundary conditions. While a boundary-value problem is generally more difficult to solve, it provides a more realistic model for transition of protein conformation and has certain computational advantages as well, especially for long-time simulations. Here we study the solution of the boundary-value problems for the simulation of transition of protein conformation using a standard class of numerical methods called the multiple shooting methods. We describe the methods and discuss the issues related to their implementations for our specific applications, including the definition of the boundary conditions, the formation of the initial trajectories, and the convergence of the solutions. We present the results from using the multiple shooting methods for the study of the conformational transition of a small molecular cluster and an alanine dipeptide, and show the potential extension of the methods to larger biomolecular systems.

Key Words. Macromolecular modeling, protein folding and misfolding, molecular dynamics simulation, initial-value problems, boundary-value problems, finite difference methods, multiple shooting methods

1. Introduction

Under certain kinetic or thermodynamic conditions, proteins make large conformational changes, formally called state transitions, resulting in significant changes in their chemical or biological functions. Of various types, protein folding or misfolding may be those of the most important conformational transitions of proteins. Folding accounts for the whole process of a one-dimensional polypeptide chain folding into a stable three-dimensional protein. The process can be considered as the protein making a conformational transition, or a series of conformational transitions, from an arbitrary state to its native state [36]. Instead, misfolding, as the word implies, is a process that the protein folds to a nonnative state, either from an

Received by the editors May 1, 2007 and, in revised form, January 1, 2008.
2000 *Mathematics Subject Classification.* 65K10, 65L05, 65L10, 65L80, 90C30, 92B05.
This research was supported partially by the NIH/NIGMS grant R01GM081680 and the Institute of Mathematics and Its Applications with funds provided by the NSF of America.

arbitrary state or its native state. Proper folding is necessary for a protein to assume its normal function, while misfolding often leads to an abnormal protein. The latter could alter the normal behaviors of a biological system and cause complex diseases [32].

The dynamic properties of proteins can be studied through molecular dynamics simulation. A conventional molecular dynamics simulation protocol solves an initial-value problem for the equation of motion defined for the molecule, with the positions and velocities of the atoms in the molecule as the initial conditions [38]. Such a procedure can be used to track protein conformational transitions, but the simulation has to be carried out for a very long time to make it possible for the expected transition to occur. Alternatively, the conformational transition of a molecule can be formulated more naturally as a boundary-value problem, with the beginning and ending positions of the atoms in the molecule as the boundary conditions. This latter approach has been adopted by several research groups [35, 16, 4, 8, 20]. However, an accurate solution to the boundary-value problem has not been fully developed.

In this paper, we study the solution of the boundary-value problems for the simulation of transition of protein conformation using a standard class of numerical methods called the multiple shooting methods. We describe the methods and discuss the issues related to their implementations for our specific applications, including the definition of the boundary conditions, the formation of the initial trajectories, and the convergence of the solutions. We present the results from using the multiple shooting methods for the study of the conformational transition of a small molecular cluster and an alanine dipeptide, and show the potential extension of the methods to larger biomolecular systems.

2. Classical Simulation

A protein is composed of a sequence of amino acids. The neighboring amino acids in the sequence are connected by strong chemical bonds and form an amino acid chain called a polypeptide. The polypeptide chain, once generated in the cell, quickly folds into a three dimensional structure and then becomes a live and functional protein (see Figure 1). The folding of a protein has been one of the most fundamental yet challenging scientific problems in the past several decades. Today, it is still unclear why and how a given sequence of amino acids can fold into a specific three dimensional protein, and it is still not possible to predict the folding pathway and the folded structure for an arbitrarily given protein [9, 30].

A protein may misfold to a "wrong" structure and lose its normal function. This may happen when the folding process makes a "wrong" turn or the native structure unfolds to another structure under certain conditions [14, 37]. The study of protein misfolding is as difficult as the study of protein folding. Many important diseases are in fact caused by or related to protein misfolding. For example, the well-known mad cow disease is believed to be the result of the misfolding of the prion protein, which damages the neuron cells in the brain [1, 32] (see Figure 2).

Both folding and misfolding can be viewed as special types of conformational transitions: During folding, a protein changes its conformation from an initial state to the native state. During misfolding, a protein makes a conformational transition from an initial (and perhaps the native) state to a misfolded state. In either case, the process can be considered as a transition of protein conformation between two conformational states. Without loss of generality, we assume that the two states correspond to two energy minima of the protein. Then, the problem of finding a

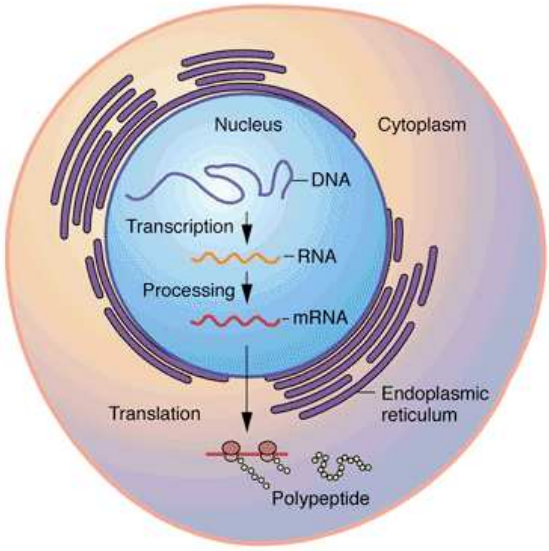


FIGURE 1. **Central dogma of molecular biology** In cell, a gene, as a DNA sequence, is transcribed to a RNA sequence. The RNA sequence is then translated to an amino acid sequence. The amino acids are connected with strong chemical bonds, forming a polypeptide chain. The polypeptide chain folds into a three-dimensional structure and becomes a live protein.

folding or misfolding pathway is to find a conformational trajectory from one energy minimum to another in the protein conformational space.

A protein conformation (or structure) is difficult to determine experimentally. The tracking of a conformational trajectory is certainly even harder. Theoretically, it may be approached through molecular dynamics simulation, which basically tries to follow the conformational changes along the trajectory based on the known physical interactions in the protein. Mathematically, a system of equations, defined by the Newton's Second Law of Motion for the atoms in the protein, needs to be solved to obtain the changes of the positions of the atoms and hence the changes of the conformation of the protein over the time [38, 5].

Let $x(t)$ be the configuration of a molecule at time t,

$$x = \{x_i : \ x_i = (x_{i,1}, x_{i,2}, x_{i,3})^T, \ i = 1, \ldots, n\},$$

where x_i is the position vector of atom i and n the total number of atoms in the molecule. Then, the equations of motion for the molecule can be written as

$$m_i \frac{d^2 x_i}{dt^2} = f_i(x_1, \ldots, x_n), \qquad i = 1, \ldots, n, \tag{2.1}$$

where m_i is the mass and f_i the force for atom i. If the potential energy function $E(x_1, \ldots, x_n)$ of the molecule is known, $f_i = -\partial E / \partial x_i$. Let

$$f = \{f_i : \ f_i = (f_{i,1}, f_{i,2}, f_{i,3})^T, \ i = 1, \ldots, n\}.$$

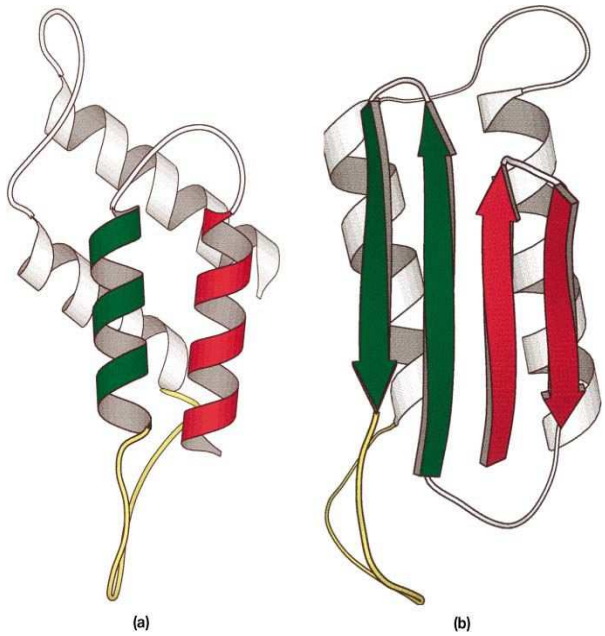


FIGURE 2. **Protein misfolding** A normal prion protein PrPC (a) may misfold to an abnormal form PrPSc (b). The abnormal prions make large aggregates and damage neuron cells, causing serious diseases such as the mad cow disease.

Then, the equations of motion can be put into a more compact form:

$$m\frac{d^2x}{dt^2} = f(x), \tag{2.2}$$

where m is the mass matrix, $m = \mathrm{diag}[m_1, \ldots, m_n]$, and f is the force field, $f(x) = -\nabla E(x)$. If we write the equation in a linear form,

$$\frac{dx}{dt} = v, \qquad \frac{dv}{dt} = m^{-1}f(x), \tag{2.3}$$

we can see that the system is Hamiltonian with the Hamiltonian

$$H(x, v) = \frac{v^T m v}{2} + E(x).$$

In classical molecular dynamics simulation, the above equations of motion are solved with a set of initial conditions as can be defined as follows:

$$x_i(t_0) = x_i^0, \qquad v_i(t_0) = v_i^0, \qquad i = 1, \ldots, n, \tag{2.4}$$

where x_i and v_i are position and velocity of atom i, and x_i^0 and v_i^0 are initial position and velocity of atom i at time t_0. In other words, an initial value problem,

$$\begin{aligned} m_i\frac{d^2x_i}{dt^2} &= f_i(x_1, \ldots, x_n), \\ x_i(t_0) = x_i^0, &\qquad v_i(t_0) = v_i^0, \qquad i = 1, \ldots, n, \end{aligned} \tag{2.5}$$

needs to be solved. In a more compact form, the problem can be written equivalently as,

$$m\frac{d^2x}{dt^2} = f(x), \qquad x(t_0) = x^0, \qquad v(t_0) = v^0. \tag{2.6}$$

Let x_i^k be the function value of x_i at time t_k. Then, the second derivative d^2x_i/dt^2 at time t_k can be approximated with

$$\frac{x_i^{k+1} - 2x_i^k + x_i^{k-1}}{\Delta t^2},$$

and the equations of motion in (2.5)are reduced to

$$\frac{x_i^{k+1} - 2x_i^k + x_i^{k-1}}{\Delta t^2} = \frac{f_i(x_1, \ldots, x_n)}{m_i}, \qquad i = 1, \ldots, n. \tag{2.7}$$

By rearranging the above equations, we obtain,

$$x_i^{k+1} = 2x_i^k - x_i^{k-1} + \frac{f_i(x_1, \ldots, x_n)}{m_i}\Delta t^2, \qquad i = 1, \ldots, n. \tag{2.8}$$

By applying the above formulas iteratively with $x_i^1 = x_i^0 + v_i^0\Delta t$, we can obtain the values x_i^k for all i and any arbitrary k, and by connecting all the values of x_i^k for a sequence of k, we can obtain an approximate trajectory for x_i. The algorithm with such a scheme is called a Verlet algorithm due to Verlet's early work in molecular dynamics simulation [41].

Let $h = \Delta t$. Then, two types of Verlet algorithms are in fact used in practice: One called the position Verlet and the other the velocity Verlet, with two slightly different iterative formulas, respectively:

Position Verlet :

$$x_i^{k+1} = 2x_i^k - x_i^{k-1} + h^2\frac{f_i^k}{m_i} \tag{2.9}$$

$$v_i^{k+1} = v_i^k + h\frac{f_i^k}{m_i} \qquad i = 1, \ldots, n, \qquad k = 1, \ldots \tag{2.10}$$

Velocity Verlet :

$$x_i^{k+1} = x_i^k + hv_i^k + h^2\frac{f_i^k}{2m_i} \tag{2.11}$$

$$v_i^{k+1} = v_i^k + h\frac{f_i^k + f_i^{k+1}}{2m_i} \qquad i = 1, \ldots, n, \qquad k = 1, \ldots \tag{2.12}$$

The position Verlet has a higher accuracy than the velocity Verlet for the calculation of the positions, but the velocity Verlet is more popular in practice because it is simplectic and preserves energy and volume of the system, which is important for obtaining an relatively accurate simulation after a long sequence of iterations. We write these numerical properties of the Verlet algorithms formally in the following theorems.

Theorem 2.1. *The position Verlet has the fourth order accuracy for the calculation of the positions.*

Proof Based on the Taylor theory, the function values $x_i(t_{k-1})$ and $x_i(t_{k+1})$ can be expressed in the following forms.

$$x_i(t_{k-1}) = x_i(t_k) - hx_i'(t_k) + h^2x_i''(t_k)/2! - h^3x_i'''(t_k)/3! + O(h^4)$$

$$x_i(t_{k+1}) = x_i(t_k) + hx_i'(t_k) + h^2x_i''(t_k)/2! + h^3x_i'''(t_k)/3! + O(h^4)$$

The sum of the two equations is

$$x_i(t_{k+1}) = 2x_i(t_k) - x_i(t_{k-1}) + h^2x_i''(t_k)/2! + O(h^4)$$

By comparing the above formula with the one used in the position Verlet algorithm, we can see that the difference between x_i^{k+1} calculated using the position Verlet and the true value of x_i at time t_{k+1} is in the order of h^4 if x_i^k and x_i^{k-1} are accurate. □

Theorem 2.2. *The velocity Verlet has the third order accuracy for the calculation of the positions.*

Proof Based on the Taylor theory, the function values $x_i(t_{k+1})$ can be expressed in the following form.

$$x_i(t_{k+1}) = x_i(t_k) + hx_i'(t_k) + h^2x_i''(t_k)/2! + O(h^3)$$

By comparing the above formula with the one used in the velocity Verlet algorithm, we can see that the difference between x_i^{k+1} calculated using the velocity Verlet and the true value of x_i at time t_{k+1} is in the order of h^3 if x_i^k is assumed to be accurate. □

Theorem 2.3. *The velocity Verlet is simplectic.*

Proof See [38] for a proof. □

The classical molecular dynamics simulation can be used to estimate the macroscopic properties such as temperature, pressure, or volume of a given molecular system, or to observe the microscopic motions of the molecule such as atomic fluctuations, local interactions, and conformational transitions. However, there are several limitations in classical molecular dynamics simulation. First, the time step of the simulation is very small in an order of femto (1.0E-15) seconds – a time scale for atomic vibrations in a molecule, and therefore, the simulation is limited to short time motions such as those in pico to nano seconds, while the time frame for many biologically interesting motions such as protein folding or misfolding are usually in the order of seconds. Second, for the transition of protein conformation, the simulation is nondeterministic in the sense that it has to be carried out for a long time period so that the transition may occur, and the waiting time is unknown priori for a given system. Finally, the numerical methods used for the solution of the initial-value problem in the classical simulation are all sequential methods because the calculations in each iteration have to be completed before they can be continued in the next iteration, and therefore, the methods cannot be parallelized for massively parallel computation, which further limits the use of available computing power for long time simulations.

3. The Boundary-Value Formulation

To simulate the conformational transition of a molecule, a more natural way is to model the problem as a boundary-value problem, namely, solving the equations of motion of the molecule with a set of conditions on the beginning and ending

states of the molecule. Elber et al [35, 16, 17, 18] are among the pioneers who have investigated such an approach for the study of conformational transitions of macromolecules such as proteins.

The boundary-value problem for the simulation of the conformational transition of a molecule can be stated as a system of equations describing the movement of the atoms in the molecule, along with a set of beginning and ending positions for the atoms. Similar to the definitions in the previous section, let $x(t)$ be the configuration of the molecule at time t,

$$x = \{x_i : \ x_i = (x_{i,1}, x_{i,2}, x_{i,3})^T, \ i = 1, \ldots, n\},$$

where x_i is the position vector of atom i and n the total number of atoms in the molecule. Let x_i^0 and x_i^e be the beginning and ending positions of atom i at time t_0 and t_e, respectively. Then, the boundary-value problem for the conformational transition of the molecule between its beginning and ending configurations at t_0 and t_e can be written as

$$m_i \frac{d^2 x_i}{dt^2} = f_i(x_1, \ldots, x_n), \tag{3.1}$$

$$x_i(t_0) = x_i^0, \qquad x_i(t_e) = x_i^e, \qquad i = 1, \ldots, n,$$

where m_i is the mass and f_i the force for atom i, $f_i = -\partial E/\partial x_i$, where E is the function for the potential energy of the molecule. Again, let

$$f = \{f_i : \ f_i = (f_{i,1}, f_{i,2}, f_{i,3})^T, \ i = 1, \ldots, n\}.$$

In a more compact form, the problem can be written equivalently as,

$$m \frac{d^2 x}{dt^2} = f(x), \tag{3.2}$$

$$x(t_0) = x^0, \qquad x(t_e) = x^e.$$

where m is the mass matrix, $m = \mathrm{diag}[m_1, \ldots, m_n]$, f the force field, $f = -\nabla E$, and $x^0 = \{x_i^0 : \ i = 1, \ldots, n\}$ and $x^e = \{x_i^e : \ i = 1, \ldots, n\}$.

Based on the theory of classical mechanics [2, 21], the trajectory of molecular motion between two molecular states minimizes the total action of the motion. Then, given the beginning and ending time t_0 and t_e, $x(t)$ in $[t_0, t_e]$ defines a trajectory connecting two molecular states $x_0 = x(t_0)$ and $x_e = x(t_e)$. Let $L(x, x', t)$ be the difference of the kinetic and potential energy of the molecule at time t. The functional L is called the Lagrangian of the molecule. Let S be the action of the molecule in $[t_0, t_e]$. Then, S is defined as the integral of the Lagrangian L in $[t_0, t_e]$, and according to the least action principle, the trajectory $x(t)$ as t changes from t_0 to t_e minimizes the action S in $[t_0, t_e]$, i.e.,

$$\min S(x) = \int_{t_0}^{t_e} L(x, x', t) dt, \tag{3.3}$$

$$x(t_0) = x^0, \qquad x(t_e) = x^e.$$

Theorem 3.1. *Let L be a continuously differentiable functional. Let $\{x(t) : \ t \in [t_0, t_e]\}$ be a solution of the least action problem in (3.3). Then, $x(t)$ satisfies the following Euler-Lagrange Equation,*

$$\frac{\partial L(x, x', t)}{\partial x'} - \frac{d}{dt}\left[\frac{\partial L(x, x', t)}{\partial x}\right] = 0 \tag{3.4}$$

Proof: Let δx be a small variation of x and $\delta x(t_0) = \delta x(t_e) = 0$. By the principle of variation, the necessary condition for x to be a solution of the least-action problem in (3.3) is that,

$$\delta S = \int_{t_0}^{t_e} \left(\frac{\partial L(x, x', t)}{\partial x} \delta x + \frac{\partial L(x, x', t)}{\partial x'} \delta x' \right) dt = 0$$

Since $\delta x' = \delta \left(\frac{dx}{dt} \right) = d \left(\frac{\delta x}{dt} \right)$, we obtain, after integrating the second term in the integrand by parts,

$$\delta S = \int_{t_0}^{t_e} \left(\frac{\partial L(x, x', t)}{\partial x} - \frac{d}{dt} \left[\frac{\partial L(x, x', t)}{\partial x'} \right] \right) \delta x dt = 0$$

Since δS should be zero for all δx, the integrand must be zero and (3.4) follows. □

Theorem 3.2. *Let* $L = \frac{x'^T m x'}{2} - E(x)$*, where* m *is the mass matrix of the molecule and* E *the potential energy function. Then, a necessary condition for* x *to minimize the action* S *is that,*

$$m \frac{d^2 x}{dt^2} = f(x), \qquad f(x) = -\nabla E(x) \tag{3.5}$$

$$x(t_0) = x^0, \qquad x(t_e) = x^e.$$

Proof: It follows from Theorem 3.1 and the facts that $\frac{d}{dt} \frac{\partial L}{\partial x'} = m \frac{d^2 x}{dt^2}$ and $\frac{\partial L}{\partial x} = -\nabla E$. □

Note that Theorem 3.1 and 3.2 show that a molecular trajectory x that minimizes the action S between two molecular states necessarily satisfies the equations of motion and the boundary conditions in (3.1). In other words, a solution x to the boundary-value problem (3.1) has a meaningful physical basis: It satisfies a necessary condition to be a solution to the least-action problem (3.3). Of course, not all trajectories x that solve the boundary-value problem (3.1) necessarily minimize the action S. Some may be maxima and others may be saddle points. Even if it is a minimizer, it may not necessarily the global minimum as the function S is nonconvex in general [19, 33].

4. Multiple Shooting

The idea of the shooting methods for the solution of the boundary-value problem in (3.1) is to find a molecular trajectory between two molecular states by mimicking the process of shooting a basketball from a given position to a target position by choosing a correct initial speed and direction for the ball. Let $x(t)$ be the position of a basketball at time t, m the mass, and $f(x)$ the force at position x. Then, x can be obtained as the solution to a boundary-value problem,

$$m \frac{d^2 x}{dt^2} = f(x), \qquad x(t_0) = x^0, \qquad x(t_e) = x^e, \tag{4.1}$$

where t_0 and t_e are the beginning and ending times, and x^0 and x^e the beginning and ending positions. In order to find a solution to the above problem, first, let x be the solution to the following initial value problem,

$$m \frac{d^2 x}{dt^2} = f(x), \qquad x(t_0) = x^0, \qquad v(t_0) = v^0. \tag{4.2}$$

Then, $x(t)$ depends on the initial velocity v^0, and can be written as $x(t) = x(t; v^0)$ and considered as a function of v^0. For an arbitrary v^0, $x(t)$ may not necessarily satisfy the ending condition, $x(t_e) = x^e$. In order to find such a solution or in other words, to find a solution to the original boundary-value problem, we need to find an appropriate v^0 so that $x(t_e) = x(t_e; v^0) = x^e$. Let $\phi(v^0) = x(t_e) - x^e = x(t_e; v^0) - x^e$. Then, the problem becomes finding v^0 so that $\phi(v^0) = 0$.

In general, let x be a vector, m a mass matrix, and f the force field. We then have a nonlinear system of equations,

$$\phi(v^0) = x(t_e; v^0) - x^e = 0, \tag{4.3}$$

where v^0 and x^e are all vectors, and

$$m\frac{d^2x}{dt^2} = f(x), \qquad x(t_0) = x^0, \qquad v(t_0) = v^0. \tag{4.4}$$

By solving the nonlinear system of equations, we can find a solution to the general boundary value problem,

$$m\frac{d^2x}{dt^2} = f(x), \qquad x(t_0) = x^0, \qquad x(t_e) = x^e. \tag{4.5}$$

Such a method is called the single shooting method [3, 12].

The single shooting method is not stable, as we can imagine that if the time is long or in other words, the distance is long, the shot will be very sensitive to the changes in the initial velocity and difficult to reach the target accurately. To overcome this difficulty, we can divide the time interval into smaller subintervals, and make the shootings separately within the subintervals. Then, the positions and velocities of the solution trajectory at the intermediate points are of course all unknown. We need to determine them all so that the solution trajectories obtained in the subintervals can be connected into a smooth trajectory over the entire interval. The method for obtaining such a trajectory is called the multiple-shooting method [3, 12].

For a general description of the multiple-shooting method, we write the problem in (4.5) in the following form, without the mass matrix,

$$\frac{d^2x}{dt^2} = f(x), \qquad x(t_0) = x^0, \qquad x(t_e) = x^e. \tag{4.6}$$

We first divide the time interval $[t_0, t_e]$ uniformly into N subintervals $[t_0, t_1], \ldots, [t_{N-1}, t_N]$, with $t_N = t_e$. Then, on each subinterval $[t_i, t_{i+1}]$, $0 \le i < N$, we solve an initial value problem,

$$\frac{d^2x}{dt^2} = f(x), \qquad x(t_i) = r^i, \qquad v(t_i) = s^i, \qquad t_i \le t \le t_{i+1}. \tag{4.7}$$

Let the solution be denoted by $x^{(i)}(t; r^{(i)}; s^{(i)})$. Then, in the entire interval,

$$x(t) = x^{(i)}(t; r^{(i)}; s^{(i)}), \qquad t_i \le t \le t_{i+1}, \qquad 0 \le i < N. \tag{4.8}$$

Here, we need to find $r = [r^{(0)}; \ldots; r^{(N-1)}]$ and $s = [s^{(0)}; \ldots; s^{(N-1)}]$ such that

$$\begin{aligned} &x^{(i)}(t_{i+1}; r^{(i)}; s^{(i)}) = r^{(i+1)}, \\ &v^{(i)}(t_{i+1}; r^{(i)}; s^{(i)}) = s^{(i+1)}, \\ &0 \le i < N-1, \end{aligned} \tag{4.9}$$

and $x^{(0)}(t_0; r^{(0)}; s^{(0)}) = x^0,\ x^{(N-1)}(t_N; r^{(N-1)}; s^{(N-1)}) = x^e$. If we define a vector function F such that

$$F(r;s) = \begin{pmatrix} x^{(0)}(t_1; r^{(0)}; s^{(0)}) - r^{(1)} \\ v^{(0)}(t_1; r^{(0)}; s^{(0)}) - s^{(1)} \\ \cdots \\ \cdots \\ x^{(N-2)}(t_{N-1}; r^{(N-2)}; s^{(N-2)}) - r^{(N-1)} \\ v^{(N-2)}(t_{N-1}; r^{(N-2)}; s^{(N-2)}) - s^{(N-1)} \\ x^{(0)}(t_0; r^{(0)}; s^{(0)}) - x^{(0)} \\ x^{(N-1)}(t_N; r^{(N-1)}; s^{(N-1)}) - x^{(e)} \end{pmatrix}, \tag{4.10}$$

then, we essentially need to determine $(r; s)$ such that $F(r; s) = 0$. The latter can be solved by using a conventional nonlinear equation solver, say, the Newton method [11, 39, 13].

Note that the system of equations $F(r; s) = 0$ has N subsystems,

$$\begin{aligned} & x^{(i)}(t_{i+1}; r^{(i)}; s^{(i)}) - r^{(i+1)} = 0, \\ & v^{(i)}(t_{i+1}; r^{(i)}; s^{(i)}) - s^{(i+1)} = 0, \\ & 0 \le i < N-1, \end{aligned} \tag{4.11}$$

and

$$\begin{aligned} & x^{(0)}(t_0; r^{(0)}; s^{(0)}) - x^0 = 0, \\ & x^{(N-1)}(t_N; r^{(N-1)}; s^{(N-1)}) - x^e = 0, \end{aligned} \tag{4.12}$$

where each subsystem in turn has $6n$ equations. There are a total of $6nN$ equations. The variables include the initial position and velocity vectors of the solution trajectory in all the subintervals. Therefore, the total number of variables is also $6nN$.

For proteins, n is typically in the order of 10,000. Therefore, if the interval is divided into 100 subintervals, there will be a system of one million equations with one million variables to solve. Suppose that each subinterval has a length of ten pico-seconds. A nanosecond trajectory would require the solution of a system of roughly this size.

However, the system can be solved more efficiently than it looks. First, each of the equations in the system has only a small subset of all variables, and therefore, the Jacobian of F is very sparse. By exploiting the sparse structure of the problem, the calculations can be significantly reduced. Second, the evaluation of the equations in each subsystem requires the solution of an initial value problem in the corresponding time subinterval. While evaluating the equations for all the subsystems would take a substantial amount of computing time, the evaluations of the equations of the subsystems are independent of each other, and can be carried out in parallel on their own subintervals. The latter property makes the multiple-shooting method more scalable and hence more efficient on parallel computers than conventional molecular dynamics simulation schemes.

Note also that the system of equations $F(r; s) = 0$ may not necessarily have an exact solution. There are at least three possible reasons. First, the boundary conditions may have errors and hence are hard to satisfy. Second, the boundary value problem may not have a solution at the first place. Third, the system is so large and complex that a solution is hard to approach if a good initial guess is not provided. A general approach to this problem is to solve the system approximately by using a least-squares method [11, 33]. One of such approaches is to separate

the system into two parts as in (4.11) and (4.12). Let the subsystem in (4.11) be denoted as $\bar{F}(r;s) = 0$. We can try to solve this subsystem relatively accurately, while solving the equations in (4.12) approximately by minimizing the squares of the norms,

$$\begin{aligned} &||x^{(0)}(t_0; r^{(0)}; s^{(0)}) - x^0||^2 \text{ and} \\ &||x^{(N-1)}(t_N; r^{(N-1)}; s^{(N-1)}) - x^e||^2. \end{aligned} \tag{4.13}$$

Here, the norms measure the coordinate differences between the two molecular structures. Therefore, the structures have to be translated and rotated properly before the the structures are compared. Also, the structures are usually flexible with different parts fluctuating differently. Therefore, the coordinates in more fluctuating parts of the structures should be compared with lower accuracy requirements. In general, the following formulation may serve the purpose even better,

$$\begin{aligned} \min\ &||x^{(0)}(t_0; r^{(0)}; s^{(0)}) - x^0||^2_{H_0} \\ &+ ||x^{(N-1)}(t_N; r^{(N-1)}; s^{(N-1)}) - x^e||^2_{H_e} \\ &+ k\bar{F}^T(r;s)\bar{F}(r;s), \end{aligned} \tag{4.14}$$

where k is a constant, and H_0 and H_e are the Hessian matrices of the energy function for the molecule evaluated at the beginning and ending positions of the trajectory, and $||x||_H = x^T H x$. The eigenvalues of the Hessian matrices correspond to the vibrational modes of the conformations [31, 10]. Therefore, minimizing the Hessian-weighted norms of the beginning and ending structures automatically takes the fluctuations of structures into account.

5. Implementation

We have implemented a multiple shooting algorithm in MATLAB for the solution of the boundary-value problem in (3.1). We have coded a Newton method with trust-region for the solution of the system of equations in (4.9) and directly called the initial-value solver in MATLAB for the solution of the initial-value problems in the subintervals. We have applied the algorithm to two test problems: a six atom argon cluster and a twenty-two atom alanine dipeptide. The argon clusters have been studied widely in physical chemistry. Each cluster has many local energy minima and the transition of the cluster from one energy minimum to another has been an important topic of research [24, 34]. The alanine dipeptide is a small polypeptide with only two residues. We chose this molecule because it is perhaps the smallest possible protein fragment that can be studied yet has almost all the necessary local interactions in a protein. This molecule has several energy minima with respect to two specific torsion angles called φ-ψ-angles [8]. We are interested in using the multiple shooting algorithm to find the molecular trajectories among these energy minima.

The multiple shooting algorithm is essentially solving a special nonlinear system of equations. In order for the algorithm to converge, a good initial guess for the solution is necessary and even critical, given the fact that the system is large and unstable. The initial guess for the conformational transition problem has to be a molecular trajectory, which is not easy to construct in general. We have used a so-called distance interpolation method [27, 28, 29] to obtain the initial trajectory, which is essentially a rough approximation to the true trajectory. The idea is to use the interatomic distances in the beginning and ending structures to generate a set of intermediate distances and then use the intermediate distances to construct a set

of intermediate structures (see Figure 3). By connecting these structures together, a piecewise linear molecular trajectory can be obtained as the initial trajectory for the shooting algorithm. More specifically, let $d^0_{i,j}$ and $d^e_{i,j}$ be two corresponding distances between atoms i and j in the beginning and ending structures. Then, an intermediate distance between atoms i and j can be generated by using the following formula:

$$d^k_{i,j} = \lambda_k d^0_{i,j} + (1 - \lambda_k) d^e_{i,j}, \tag{5.1}$$

where $\lambda_k = k/M$, $k = 1, \ldots, M-1$, if M intermediate distance sets are to be generated.

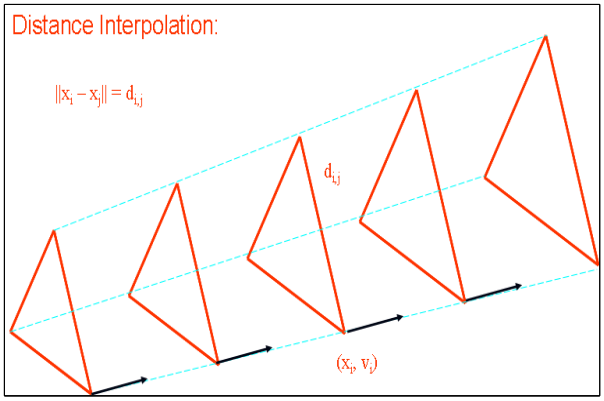


FIGURE 3. **Distance interpolation** For every pairs of distances in the beginning and ending structures, a set of intermediate distances is generated. The generated distances can then be used to form a set of intermediate structures, via the solution of a set of distance geometry problems. An initial trajectory can be obtained by connecting the generated intermediate structures, along with the beginning and ending structures.

For the kth intermediate structure, a set of distances $\{d^k_{i,j} : \ i, j = 1, \ldots, n\}$ is generated. Such a set of distances may not define a three-dimensional structure, but we can still obtain an approximated one by approximately solving a so-called distance geometry problem as can be given in the following,

$$||x^k_i - x^k_j|| = d^k_{i,j}, \qquad i, j = 1, \ldots, n. \tag{5.2}$$

One of the approximation methods is based on singular value decomposition [22]. First, set x^k_n to the origin. Then, let $x^k = \{x^k_i : \ i = 1, \ldots, n-1\}$ and $d^k = \{([d^k_{i,n}]^2 - [d^k_{i,j}]^2 + [d^k_{j,n}]^2)/2 : \ i, j = 1, \ldots, n-1\}$. Then, it is easy to verify that $[x^k][x^k]^T = d^k$. Let the singular value decomposition of $d^k = usu^T$. Then, an approximate solution to the equation $[x^k][x^k]^T = d^k$ can be obtained by setting $x^k = \bar{u}\bar{s}^{1/2}$, where $\bar{u} = u(:, 1:3)$ and $\bar{s} = s(1:3, 1:3)$. Here, if the rank of d^k is less than or equal to three, x^k in fact solves the equation exactly. If the rank of d^k is greater than three, x^k is the best possible approximation to the solution of the equation in the following sense.

Theorem 5.1. *$x^k = \bar{u}\bar{s}^{1/2}$ is an approximate solution to the equation $[x^k][x^k]^T = d^k$ in the sense it minimizes $||[x^k][x^k]^T - d^k||_F$.*

Proof: See a proof in [23] □

6. Test Results on Argon Cluster

Let x_i be the coordinate vector of argon atom i. Then, the energy function for a cluster of n argon atoms can be defined by using the following formula:

$$E(x) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} h_{i,j}(||x_i - x_j||), \tag{6.1}$$

where $x = \{x_i : \ i = 1, \ldots, n\}$ is the set of coordinate vectors of the atoms in the cluster, which fully defines the configuration of the cluster, and $h_{i,j}$ is the pairwise potential for atoms i and j defined on the distance between the two atoms $d_{i,j} = ||x_i - x_j||$:

$$h_{i,j}(d_{i,j}) = \epsilon_{i,j} \left[\frac{\sigma_{i,j}}{d_{i,j}^{12}} - 2\frac{\sigma_{i,j}}{d_{i,j}^{6}} \right], \tag{6.2}$$

where $\epsilon_{i,j}$ and $\sigma_{i,j}$ are appropriately chosen parameters. The parameters are usually different for different pairwise interactions, but for homogeneous clusters, they should be the same for all the (i, j)-pairs and can be denoted simply as ϵ and σ. For argon atoms, ϵ = 661.6E-23 J and σ = 3.405E-10 m [41].

Let ϵ be the unit for energy and σ the unit for length. We can then reduce the above formula for $h_{i,j}$ to:

$$h_{i,j}(d_{i,j}) = \frac{1}{d_{i,j}^{12}} - \frac{2}{d_{i,j}^{6}}, \tag{6.3}$$

which achieves a minimum energy -1 at 1 unit length. For simplicity, we have used this formula in all our calculations and we have only considered the cluster of 6 argon atoms. According to previous studies [24, 34], the cluster of 6 argon atoms has two local minima, among many others. One of the two minima, considered as the global minimum, is equal to -12.7 energy units, and is achieved when the cluster forms a symmetric octahedra as shown in Figure 4. Another one, very close to the global minimum, is equal to -12.3 energy units, and is achieved when the cluster forms a stretched structure as if there are three tetrahedra connected in sequence as shown in Figure 5. The energy difference between the two minima is small, but it seems that there is a big energy barrier in between them, because a local optimization routine can often be trapped into the -12.3 minimum and cannot find the other one easily.

We are interested in finding out the condition under which the cluster of 6 argon atoms changes its configuration between the above two energy minima. Let the configuration corresponding to the -12.7 energy minimum be denoted by x^0 and the one corresponding to the -12.3 energy minimum by x^e. Then, $x^0 = \{x_i^0, \ i = 1, \ldots, 6\}$ and $x^e = \{x_i^e, \ i = 1, \ldots, 6\}$, and the problem can be formulated as a boundary-value problem as shown below.

$$\begin{aligned} m\frac{d^2x}{dt^2} &= -\nabla E(x) \\ x(t_0) = x^0, &\qquad x(t_e) = x^e \end{aligned} \tag{6.4}$$

where $E(x)$ is given in (6.1) and m an $n \times n$ mass matrix with $n = 6$.

We have applied the multiple shooting algorithm discussed in previous section to the above problem. Our goal was to determine if the algorithm converges (for

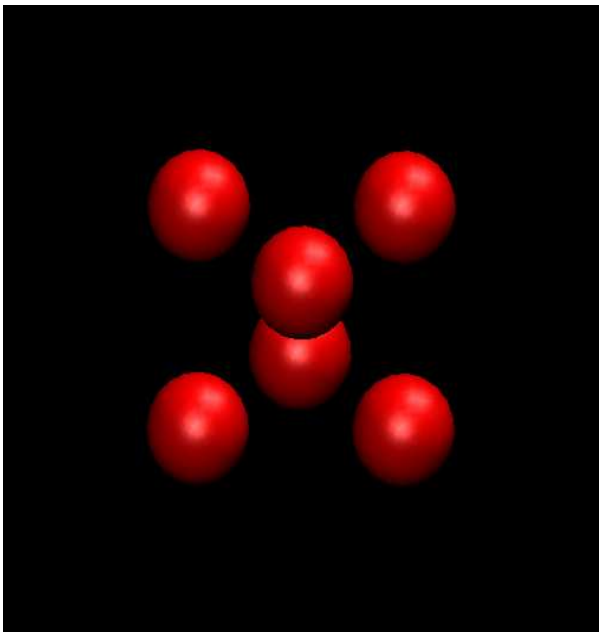


FIGURE 4. **Argon cluster configuration 1** A octahedra is formed with six argon atoms. It corresponds to the global potential energy minimum of the cluster.

different numbers of subintervals) and is able to find the correct solution for the problem, when given a reasonably constructed initial solution.

In order to generate an initial solution, we ran a conventional molecular dynamics simulation algorithm for 453 steps with a (very small!) time step of 0.0049 femto seconds. We divided the entire time interval into 1, 3 or 6 subintervals, and over these intervals, we perturbed the initial positions and velocities to obtain an initial trajectory. With such an initial trajectory, we ran a maximum of 25 iterations of two different algorithms, the damped Newton and the Newton with trust region, to find the correct trajectory. The convergence results are summarized in Table 1. For either choice for the algorithm, if the algorithm converged, the convergence seemed linear until the last few steps when the super-linear convergence was observed. In fact, for small perturbations, convergence could be rapid and super-linear. The data in Table 1 shows that 5 of 6 simulations converged super-linearly in less than 12 iterations. One simulation did not converge before the maximum number of iterations was reached.

The results of the simulations revealed a converged trajectory of the argon cluster moving from the octahedra configuration to the connected tetrahedra configuration, as shown in Figure 6. If the cluster makes a direct configuration change by moving the middle two atoms strictly down and stretching the bottom two atoms strictly apart, it will have to overcome some high energy barriers. Instead, it seemed that the cluster first pull the middle two atoms and the bottom two atoms apart at the same time and then pushed the middle two atoms down. Since the atoms were made far apart, the energy barriers became small when the middle two atoms went down, and the cluster was able to make an "easy" transition path between the two energy states.

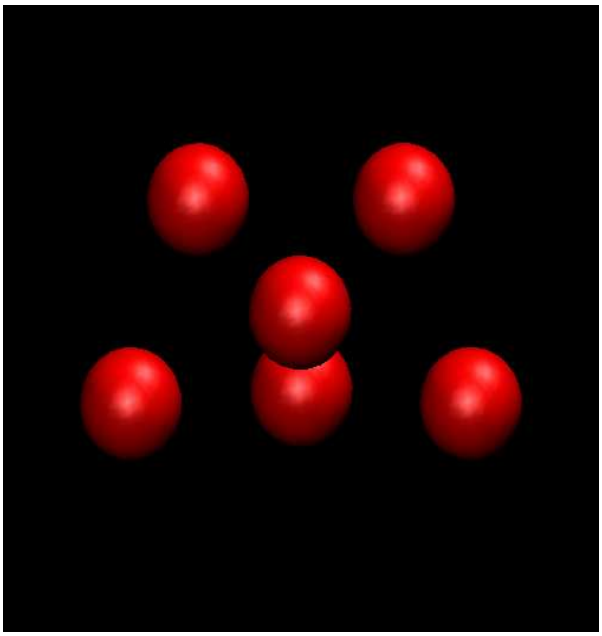


FIGURE 5. **Argon cluster configuration 2** Three tetrahedra are connected to form a structure for the six atom argon cluster. It has a local minimal potential energy.

TABLE 1. Convergence Results for the Shooting Methods

iteration	trust region Newton			damped Newton		
	$N = 1$	$N = 3$	$N = 6$	$N = 1$	$N = 3$	$N = 6$
1	1.59E-01	1.42E+00	2.76E+00	1.42E+00	2.76E+00	5.50E-02
2	1.52E-02	1.28E+00	1.95E+00	7.04E-01	1.24E+00	9.86E-03
3	6.98E-04	1.15E+00	1.38E+00	2.51E-01	3.03E-01	6.63E-04
4	3.50E-06	1.03E+00	1.00E+00	1.17E-01	2.35E-01	6.35E-06
5	3.07E-10	8.22E-01	9.46E-01	1.08E-02	6.76E-02	6.88E-10
6	-	4.84E-01	9.93E-01	3.72E-04	9.37E-03	1.98E-15
7	-	9.91E-02	7.73E-01	4.58E-07	1.49E-04	-
8	-	2.01E-02	9.36E-01	8.17E-13	5.61E-08	-
9	-	3.60E-04	3.57E-01	-	3.71E-14	-
10	-	1.08E-06	3.51E-01	-	-	-
11	-	4.69E-12	3.51E-01	-	-	-
12	-	-	3.51E-01	-	-	-

7. Test Results on Alanine Dipeptide

Let x_i be the coordinate vector for atom i in an alanine dipeptide capped with an acetyl group at the N-terminus and amine (amide) and methyl groups at the C-terminus as shown in Figure 7. Let $x = \{x_i : \ i = 1, \ldots, n\}$ be the set of coordinate vectors for all the atoms in the molecule, where $n = 22$. Then, the potential energy

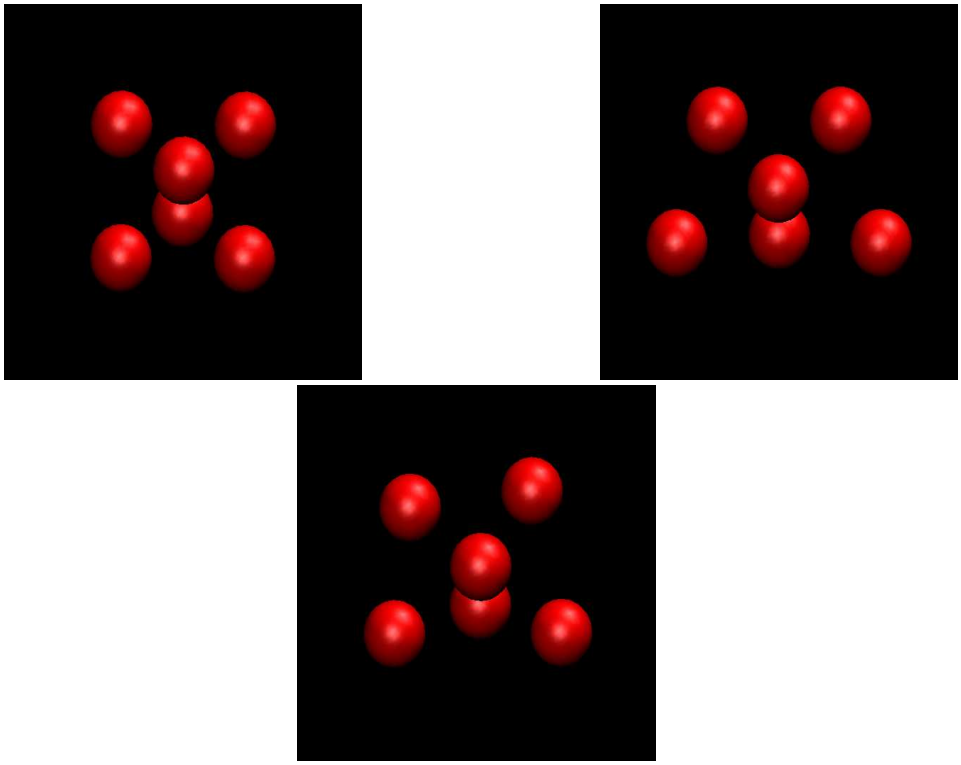


FIGURE 6. **Conformational transition of argon cluster** The cluster first pulls the middle two atoms and the bottom two atoms apart at the same time and then pushes the middle two atoms down. Since the atoms are made far apart, the energy barriers are small when the middle two atoms move down, and the cluster is able to make an "easy" transition path between the two energy states.

function for this molecule can be defined in a general form as follows.

$$
\begin{aligned}
E(x) = &\sum_{bonds} K_b(b-b_0)^2 + \sum_{bond\ angles} K_\theta(\theta-\theta_0)^2 \\
&+ \sum_{dihedral\ angles} K_\phi \cos(n\phi-\delta) + \sum_{improper\ dihedral\ angles} K_\omega(\omega-\omega_0)^2 \\
&+ \sum_{nonbonded} \frac{A_{i,j}}{r_{i,j}^{12}} - \frac{B_{i,j}}{r_{i,j}^{6}} + \frac{q_i q_j}{\varepsilon r_{i,j}}
\end{aligned}
\tag{7.1}
$$

where K_b is the bond stretching force constant; b, the bond length; b_0, the ideal bond length; K_θ, the bond angle bending force constant; θ, the bond angle; θ_0, the ideal bond angle; K_ϕ, the proper dihedral angle bending force constant; ϕ, the proper dihedral angle; n, the dihedral angle multiplicity term; δ, the phase factor; K_ω, the improper dihedral angle bending force constant; ω, the improper dihedral angle; ω_0, the ideal improper dihedral angle; $A_{i,j}$, the Lennard-Jones repulsion parameters; $B_{i,j}$, the Lennard-Jones attraction parameters; $r_{i,j}$, the interatomic distances; q_i, the atomic electrostatic charges; ε, the dielectric constant.

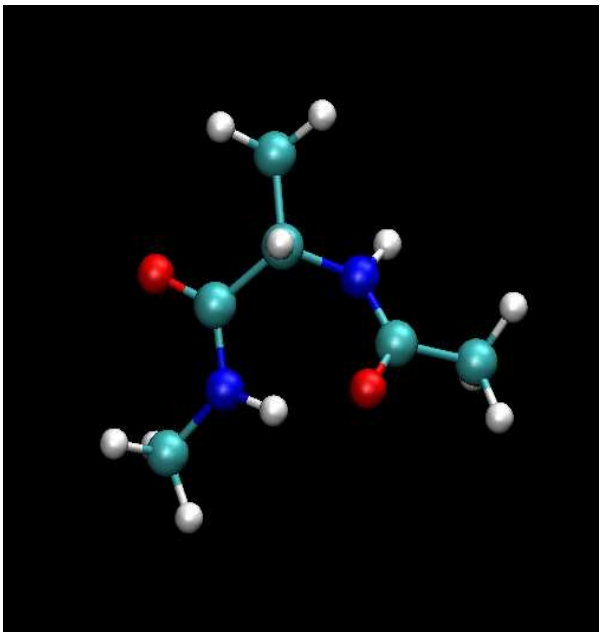


FIGURE 7. **Alanine dipeptide** The dipeptide has two alanine peptides capped with an acetyl group at the N-terminus and amine (amide) and methyl groups at the C-terminus. There are twenty-two atoms in the molecule.

The above function can be used to compute the energy for any protein. Of course, for each different protein, the parameters in the function need to be properly assigned. There are several software packages providing these parameters, including AMBER [7] and CHARMM [6]. The software MOIL developed by Elber et al. [18] combines the parameters in AMBER with some in CHARMM. We have followed the implementation of MOIL and made a simplified version of MOIL in MATLAB for our calculations. We call our code MAM as an abreviation for MATLAB AMBER MOIL.

Alanine dipeptide has a single sidechain CH3 branched from the C_β position and two blocked amide planes. The $C - N - C - C_\alpha$ dihedral angle φ and the $N - C - C_\alpha - C$ dihedral angle ψ are the two so-called soft degrees of freedom and these angles are thought to be of primary importance in classification of the molecular shape of alanine dipeptide. The other internal degrees of freedom are thought to deviate relatively slightly from mean values as a function of time.

Because of the relative flexibility and importance of these φ and ψ dihedral angles, it is common to use a projection onto a two-dimensional subspace determined by the values of φ and ψ as a way to visualize the potential energy surface and the conformational change. We have constructed a two dimensional adiabatic energy map using the MAM energy function. We applied the MATLAB optimization rotuine FMINUNC to the energy function of the molecule, fixing 1369 different combinations of values of φ and ψ. We then used the energy minima to obtain an energy map with varying φ and ψ values. The obtained map is shown in Figure 8, which is similar to the plots in [35, 4] produced with similar methods in similar force fields. In our calculations, in most time, the optimization routine terminated with

one of the seven minimal energy conformations as shown in Figure 8 and described in Table 2.

TABLE 2. Local Minima of Alanine Dipeptide

id	conformation	φ	ψ	energy
1	$C7_{eq}$	-72	40	-22.7
2	$C5_\beta$ I	-152	166	-21.8
3	$C7_{ax}$	59	-31	-20.4
4	C6	-133	23	-20.1
5	$C5_\beta$ II	-150	157	-18.6
6	α_R	-72	-13	-17.6
7	α_L	56	20	-13.6

As mentioned in [8], this molecule has been a model system for many computational studies of biomolecules. And, with respect to the location of local energy minima, these studies exhibit some variations due to differences in the details of the effective potential model. However, in [8], it is asserted that the local minima are typically found in five primary regions: $C7_{eq}$, $C5_\beta$, $C7_{ax}$, α_R, and α_L in the φ-ψ plot (see Figure 8).

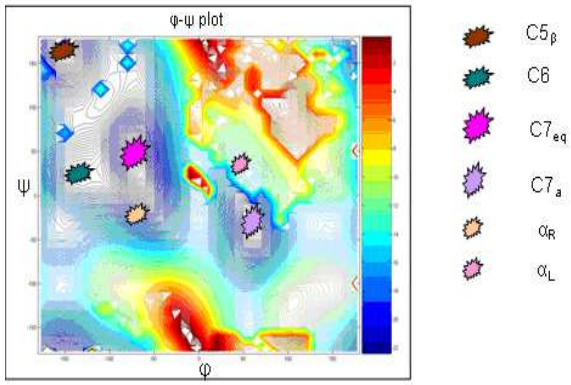


FIGURE 8. **Energy minima of alanine dipeptide** The alanine dipeptide has seven major energy minima, in terms of two torsional angle variables, φ and ψ, in the backbone, with all other variables fixed. They are labeled as $C7_{eq}$, $C5_\beta$ I, $C5_\beta$ II, $C7_{ax}$, $C6$, α_R, and α_L.

With respect to the alanine dipeptide potential energy surface and the frequencies of the different ending structures for the unconstrained minimization, α_L and lower α_R regions seem to have higher energy and lower frequencies than what is suggested to be common in studies of alanine dipeptide in solution. This may be due to the fact that this energy surface corresponds to alanine dipeptide in vacuum rather than in solution. Also, it could be partly due to the peculiarities of the optimization methods used to model the potential energy surface. Similar explanations may apply to the existence of the C6 'hybrid' local minimum on this energy surface. For the purpose of testing the multiple shooting methods on alanine dipeptide, these differences do not seem to be critical. In terms of φ-ψ plot regions, we have considered 3 types of conformational transitions:

(1) transitions between primary local minima of the $C7_{eq}$ and $C6$ potential energy wells.
(2) transitions between primary local minima of the $C7_{eq}$ and $C5_{\beta}$ potential energy wells.
(3) transitions between primary local minima of the $C7_{eq}$ and $C7_{ax}$ potential energy wells.

We are interested in finding out the condition under which the alanine dipeptide makes its conformational transitions between the above three pairs of energy minima. Let the conformation corresponding to the energy minimum $C7_{eq}$ be denoted by x^0 and the one corresponding to one of the other three energy minima, $C6$, $C5_{\beta}$, and $C7_{ax}$ by x^e. Then, $x^0 = \{x_i^0,\ i = 1, \ldots, 22\}$ and $x^e = \{x_i^e,\ i = 1, \ldots, 22\}$, and the problem can be formulated as a boundary-value problem as shown below.

$$\begin{aligned} m\frac{d^2x}{dt^2} &= -\nabla E(x) \\ x(t_0) = x^0, &\qquad x(t_e) = x^e \end{aligned} \tag{7.2}$$

where m is an $n \times n$ mass matrix with $n = 22$, and $E(x)$ is given in (7.1) and calculated with MAM.

We have applied the multiple shooting algorithm discussed in previous section to the above problem. Our goal was to determine if the algorithm converges (for different numbers of subintervals) and is able to find the correct solution for the problem, when given a reasonably constructed initial solution. Our study is en vacuo, so our simulations do not explicitly include the interactions with water molecules which in general help define and promote the stability of the global minimum and other local minima on the potential energy surface. However, a model in vacuum is certainly a reasonable point to start with.

We generated the initial trajectories with the distance interpolation method described in Section 5. Multiple trajectories were generated for each of three transitions for testing purposes. For simplicity, we will not discuss the detailed process of computing the initial trajectories, but refer the readers to the work described in [40]. Here, we focus on the application of the multiple shooting algorithm. The algorithm was applied with different numbers of subintervals and with different energy levels. In particular, for N =1, 2, 3, and 6 subintervals, the algorithm using the Newton with trust-region was applied at nine different energy levels. We defined a weak convergence for the algorithm if the norm of the system of equations was less than 0.25 in the final iteration of the algorithm. We also defined a strong convergence for the algorithm if the norm of the system of equations was less than 1.0E-06. From the point of view of numerical computing, the weak convergence indicated an approximate solution. However, in practice, it is already enough for obtaining a meaningful trajectory. In any case, a super-linear convergence was observed in most of our test cases when a strong convergence was achieved. Table 3 shows the frequency of convergence, energy level upon convergence, and transition time in our testing. For the ensemble of solution trajectories for each transition, the distribution of total energy is also shown in Figure 9 and the φ-ψ plots are shown in Figure 10.

8. Concluding Remarks

In this paper, we have formulated the problems of protein conformation transitions as boundary-value problems and developed multiple shooting methods for

TABLE 3. **Convergence of Trajectories**

transition	weak convergence	strong convergence	transition time	energy
$C7_{eq}$ to $C6$	118 / 120	55 / 120	0.1 2.3 ps	-13.6 kCal
$C7_{eq}$ to $C5_beta$	88 / 120	10 / 120	0.1 3.0 ps	-6 kCal
$C7_{eq}$ to $C7_{ax}$	74 / 120	9 / 120	0.1 3.0 ps	7 kCal

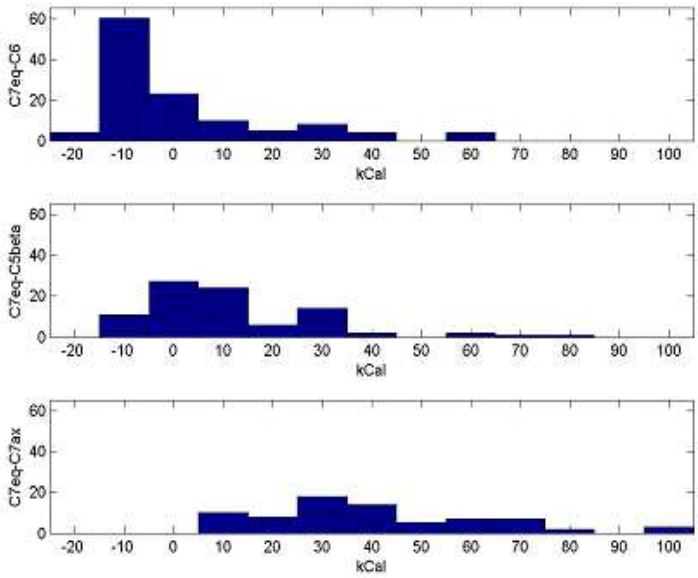


FIGURE 9. **Energy distribution** Shown in the graphs are the distributions of total energy for the ensembles of solution trajectories for three computed transitions. The trajectories from $C7_{eq}$ to $C6$ seem to have lower energies.

the solution of the problems. We described the methods and discussed related implementation issues, and presented results from using the methods for the study of conformational transitions of a small molecular cluster and an alanine dipeptide, and showed the potential extension of the methods to larger biomolecular systems.

The boundary-value approach to the transition of protein conformation in this work has considered all the atoms in the protein. It is possible, and perhaps more advantageous, to apply the same approach to more coarse-grained, or reduced, models of proteins. In [32], interesting results were obtained from molecular dynamics simulation of prion-like proteins using medium resolution lattice models. We plan to investigate this possibility in detail in future.

In future work, we also hope to consider the interaction between the molecule and the surrounding solvent since this interaction is thought to be critical to the folding process. A simple way to attempt to model the interactions with solvent is to adjust the dielectric constant in the potential energy function. Another approach

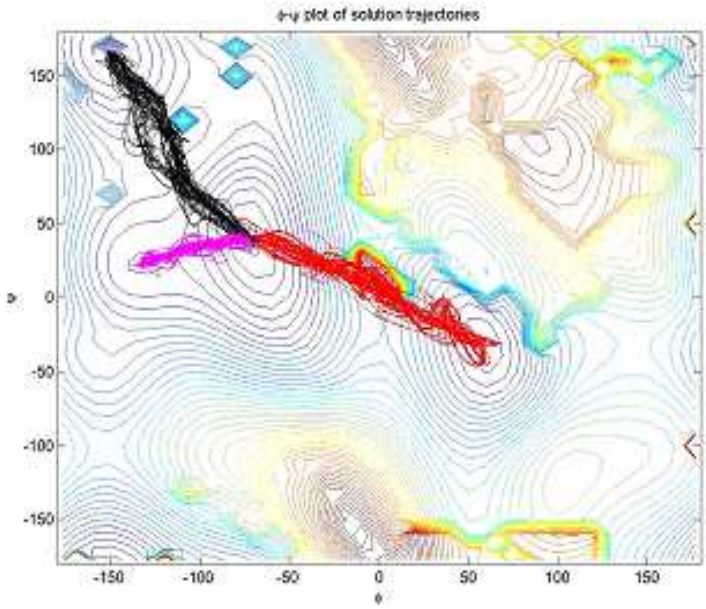


FIGURE 10. **Transition paths** Ensembles of solution trajectories or transition paths are obtained for each of the transitions: $C7_{eq}$ to $C6$ (pink), $C7_{eq}$ to $C5_{\beta}$ (brown), and $C7_{eq}$ to $C7_{ax}$ (red).

to simulating interaction with water is to use a stochastic approach and apply Brownian or Langevin dynamics.

It is reasonable to question the physical significance of the deterministic solutions to the specific boundary-value problems presented here. Do their pathways represent pathways that occur in nature? If so, do they represent common pathways? Before comparing different pathways, we must be able to define in some way the different pathways being compared and develop a method of categorizing trajectories by pathway. We intend to explore these issues in future.

The actual transition time between local minima can be very fast. While transition can be somewhat longer in solution than in vacuum, it is still worthwhile to note that, in vacuum, the transition times are usually 0.5 to 3.0 picoseconds. The boundary value approach can be more efficient since it is deterministic and can find a solution trajectory much faster than the initial value approach in conventional simulation. The waiting time in the latter approach is usually not known priori and can be much longer than necessary.

Acknowledgments

We would like to thank Di Wu and Ajith Gunaratne for helpful comments and suggestions on the paper, and Program on Bioinformatics and Computational Biology and Department of Mathematics, Iowa State University for their support on this work.

References

[1] Aguzzi A, Montrasio F, and Kaeser P, Prions: Health scare and biological challenge, Nature Rev. Mol. Cell Biol., 2, 2001, pp. 118-125.

[2] Arnold V, Mathematical Methods of Classical Mechanics, Springer-Verlag, 1975.

[3] Ascher U, Mattheij R, and Russell R, Numerical Solution of Boundary Value Problems for Ordinary Differential Equations, SIAM, 1995.

[4] Bolhuis P, Dellago C, and Chandler D, Reaction coordinates of biomolecular isomerization, Proc Natl Acad Sci USA, 97, 2000, pp. 58775882.

[5] Brooks B, Karplus M, and Pettitt M, Proteins: A Theoretical Perspective of Dynamics, Structure and Thermodynamics, John Wiley & Sons, 2004.

[6] Brooks B, Bruccoleri R, Olafson B, States D, Swaminathan S, and Karplus M, CHARMM: A program for macromolecular energy minimization and dynamics calculations, Journal of Computational Chemistry, 4, 1983, pp. 187-217.

[7] Case D, Darden T, Cheatham T, Simmerling C, Wang J, Duke R, Luo R, Merz K, Wang B, Pearlman B, Crowley M, Brozell S, Tsui V, Gohlke H, Mongan J, Hornak V, Cui G, Beroza P, Schafmeister C, Caldwell J, Ross W, and Kollman P, AMBER 8, University of California, San Francisco, 2004.

[8] Chekmarev D, Ishida D, and Levy R, Long time conformational transitions of alanine dipeptide in aqueous solution: continuous and discrete state kinetic models, J. Phys. Chem. B, 108, 2004, pp. 19487-19495.

[9] Creighton T, Proteins: Structures and Molecular Properties, 2nd ed., Freeman and Company, 1993.

[10] Cui Q and Bahar I, Normal Mode Analysis: Theory and Applications to Biological and Chemical Systems, Chapman & Hall CRC, 2005.

[11] Dennis J and Schnabel R, Numerical Methods for Unconstrained Optimization and Nonlinear Equations, SIAM, 1996.

[12] Deuflhard P and Bornemann F, Scientific Computing with Ordinary Differential Equations, Springer, 2002.

[13] Deuflhard P, Newton Methods for Nonlinear Problems, Springer, 2004.

[14] Dobson C and Fersht A, Protein Folding, Cambridge University Press, 1996.

[15] Elber R, Roitberg A, Simmerling C, Goldstein R, Li H, Verkhiver G, Keasar C, Zhang J, and Ulitsky A, MOIL: A program for simulation of macromolecules, Comp. Phys. Comm. 91, 1994, pp. 159-189.

[16] Elber R, Meller J, and Olender R, Stochastic path approach to compute atomically detailed trajectories: application to the folding of C peptide, Journal of Physical Chemistry B, 103, 1999, pp. 899-911.

[17] Elber R, Ghosh A, and Cardenas A, Long time dynamics of complex systems, Acc. Chem. Res, 35, 2002, 396-403.

[18] Elber R, Ghosh A, and Crdenas A, and Stern H, Bridging the gap between reaction pathways, long time dynamics and calculation of rates, Advances in Chemical Physics, 126, 2003, pp. 93-129.

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

[20] Gladwin B and Huber T, Long time scale molecular dynamics using least action, Anziam J., 45E, 2004, pp. C534C550.

[21] Goldstein H, Classical Mechanics, Addison-Wesley, 1980.

[22] Golub G and van Loan C, Matrix Computation, Johns Hopkins University Press, 1989.

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

[24] Hoare M, Structure and dynamics of simple microclusters, Advances in Chemical Physics, 40, 1979, pp. 49-135.

[25] Hu H, Elstner M, and Hermans J, Comparision of a QM/MM force field and molecular mechanics force fields in simulations of alanine and glycine dipeptides (Ace-Ala-Nme and Ace-Gly-Nme) in water in relation to the problem of modeling the unfolded peptide backbone in solution, Proteins: Structure, Function, and Genetics, 50, 2003, pp. 451463.

[26] Humphrey W, Dalke A. and Schulten K, VMD - Visual Molecular Dynamics, J. Molec. Graphics, 14, 1996, pp. 33-38.

[27] Kim M, Chirikjian G, and Jernigan R, Elastic models of conformational transitions in macromolecules, Journal of Molecular Graphics and Modelling, 21, 2002, pp. 151160.

[28] Kim M, Jernigan R, and Chirikjian G, Efficient generation of feasible pathways for protein conformational transitions, Biophysical Journal, 83, 2002, pp. 1620-1630.

[29] Kim M, Li W, Shapiro B, and Chirikjian G, A comparison between elastic network interpolation and MD simulation of 16S ribosomal RNA. Journal of Biomolecular Structure & Dynamics, 21, 2003, pp. 1-12.
[30] Lesk A, Introduction to Protein Architecture: The Structural Biology of Proteins, Oxford University Press, 2001.
[31] Levitt M, Sander, C, and Stern P, Protein normal-mode dynamics: trypsin inhibitor, crambin, ribonuclease and lysozyme, J. Mol. Biol., 181, 1985, pp. 423-447.
[32] Malolepsza E, Boniecki M, Kolinski A, and Piela L, Theoretical model of prion propagation: A misfolded protein induces misfolding, PNAS 102, 2005, pp. 78357840.
[33] Nocedal J and Wright S, Numerical Optimization, Springer-Verlag, 2006.
[34] Northby J, Structure and binding of Lennard-Jones clusters: 13 147, Journal of Chemical Physics, 87, 1987, pp. 6166-6177.
[35] Olender R and Elber R, Calculation of classical trajectories with a very large time step: formalism and numerical examples, J. Chem. Phys., 105, 1996, pp. 9299-9315.
[36] Onuchic J and Wolynes P, Theory of protein folding, Current Opinion in Structural Biology, 14, 2004, pp. 7075.
[37] Pain R, Mechanism of Protein Folding, Oxford University Press, 2000.
[38] Schlick T, Molecular Modeling and Simulation: An Interdisciplinary Guide, Springer, 2003.
[39] Stoer J and Bulirsch R, Introduction to Numerical Analysis, 3rd edition, Springer, 2002.
[40] Vedell P, Multiple Shooting Methods for the Solution of the Bourday-Value Problems in Molecular Dynamics Simulation, Ph.D. Thesis, Department of Mathematics, Program on Bioinformatics and Computational Biology, Iowa State University, 2006.
[41] Verlet L, Computer "experiments" on classical fluids I. Thermodynamical properties of Lennard-Jones molecules, Physical Review, 159, 1967, pp. 98-103.
[42] Wille L, Minimum-energy configurations of atomic clusters: New results obtained by simulated annealing, Chemical Physics Letter, 133, 1987, pp. 405 - 410.

Program on Bioinformatics and Computational Biology, Department of Mathematics, Iowa State University, Ames, IA 50010, USA
E-mail: `vedell@iastate.edu`

Program on Bioinformatics and Computational Biology, Department of Mathematics, Iowa State University, Ames, IA 50010, USA
E-mail: `zhijun@iastate.edu`