CHAPTER III

POTENTIAL ENERGY, MINIMIZATION AND MOLECULAR DYNAMIC SIMULATIONS

3.1 Potential energy functions

The same for all approached, the theoretical studies to investigate the relationships between structure, function and dynamics at the atomic level are based on several assumptions. As many of the problems, especially in biological systems involve many atoms, quantum mechanics is too expensive to treat these systems. However, the problems become much more adaptable when turning to empirical potential energy functions. On the other hand, there are certain limitations of empirical force fields. One of the most important is that no drastic changes in electronic structure are allowed, for instance, no events like bond making or breaking can be modeled.

Up to date development potential energy functions (or force field) provide a reasonably good compromise between accuracy and computational efficiency. They are often calibrated to experimental results and quantum mechanical calculations of small model compounds. Their facility to reproduce physical properties calculable by experiment is tested; these properties include structural data obtained from x-ray crystallography and NMR, dynamic data obtained from spectroscopy and thermodynamic data. The most commonly used potential energy functions are AMBER [65], CHARMM [66], GROMOS [67] and OPLS/AMBER [68] force fields. The continuing development of force fields remains an intense area of research with implications for both fundamental researches as well as applied researches in the pharmaceutical industry.

We have briefly reviewed the variety of interactions which are important in bimolecular interactions and seen suitable simple mathematical forms for their representation. These are drawn together to form a potential energy function

$$V(R) = V_{\text{intra}}(R) + V_{\text{non-bonded}}(R) \quad . \tag{3.1}$$

3.1.1 Intramolecular interactions

The $V_{intra}(R)$ is sum of the following three terms

$$V_{intra}(R) = E_{bonds} + E_{bond-angles} + E_{torsion-angles}, \tag{3.2}$$

which correspond to three types of atom movement, r_{23}, θ_{234}, and ϕ_{1234} shown in Figure 3.1)

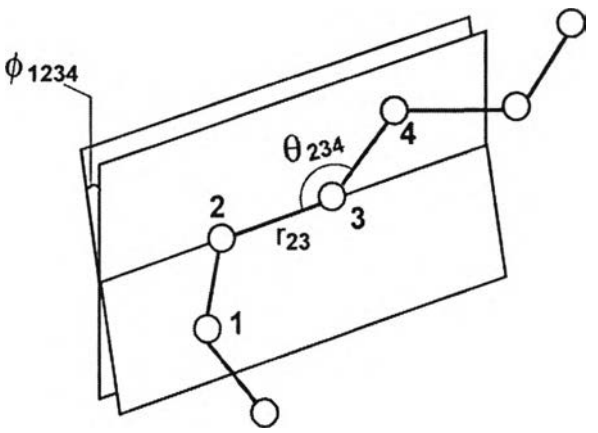


Figure 3.1 Geometry of a simple chain molecule, illustrating the definition of interatomic distance r_{23}, bend angle θ_{234}, and torsion angle ϕ_{1234}.

E_{bonds} is the energy function for stretching a bond between two atoms, atoms 2 and 3 in Figure 3.1. It is a harmonic potential representing the interaction between atomic pairs where atoms are separated by one covalent bond, i.e., 1, 2 pair. This is the approximation to the energy of a bond as a function of displacement from the ideal bond length, b_0. The force constant, K_b, determines the strength of the bond. Both ideal bond lengths and force constants are specific for each pair of bound atoms, i.e., their value only depend on chemical type of atoms-constituents,

$$E_{bonds} = \sum_{1,2-pairs} K_b (b - b_0)^2 . \tag{3 3}$$

Values of force constant are often evaluated from experimental data such as infrared stretching frequencies or from quantum mechanical calculations.

Values of bond length can be inferred from high resolution crystal structures or microwave spectroscopy data.

The second term in Equation 2.2 is associated with alteration of bond angles from ideal values θ_0 and from constant K_θ, which are also represented by a harmonic potential. Values of θ_0 and K_θ depend on chemical type of atoms constituting the angle,

$$E_{bonds-angles} = \sum_{angles} K_\theta (\theta - \theta_0)^2 . \tag{3.4}$$

These two terms describe the deviation from an ideal geometry. They are penalty functions and that in a perfectly optimized structure; the sum of them should be close to zero.

The last term in Equation 2.2 represents the torsion angle potential function which models the presence of steric barriers between atoms separated by 3 covalent bonds (1, 4 pair). The motion associated with this term is a rotation around 2-3 bond in a four atom sequence 1-2-3-4 (see Figure 3.1), described by a dihedral angle. This potential is assumed to be periodic and is often expressed as a cosine function.

$$E_{\text{torsion-angles}} = \sum_{1,4\,pair} K_\Phi (1 - cos(n\Phi)) \tag{3.5}$$

3.1.2 Non-bond interactions

The non-bonded interactions are contributed by two functions which are van der Waals interaction energy and the electrostatic interaction energy.

$$V_{\text{non-bonded}}(R) = E_{\text{vdw}} + E_{\text{electrostatic}} \tag{3.6}$$

The van der Waals energy, E_{vdw}, describes the repulsion or attraction between atoms that are not directly bonded. This term can be interpreted as the part of

the interaction which is not related to atomic charges. The van der Waals interaction between two atoms arises from a balance between repulsive and attractive forces. The repulsive force comes up at short distances because the electron-electron interaction is strong. The attractive force, also referred to as the dispersion force, arises from fluctuations in the charge distribution in the electron clouds. The fluctuation in the electron distribution on one atom or molecules gives rise to an instantaneous dipole which, in turn, induces a dipole in a second atom or molecule giving rise to an attractive interaction. Each of these two effects is equal to zero at infinite atomic separation R. The attractive interaction is longer range than the repulsion but as the distance become short, the repulsive interaction becomes dominant. This gives rise to a minimum in the energy. Both value of energy at the minimum E^* and the optimal separation of atoms r^* which is roughly equal to the sum of van der Waals radii of the atoms. It will depend on chemical type of these atoms.

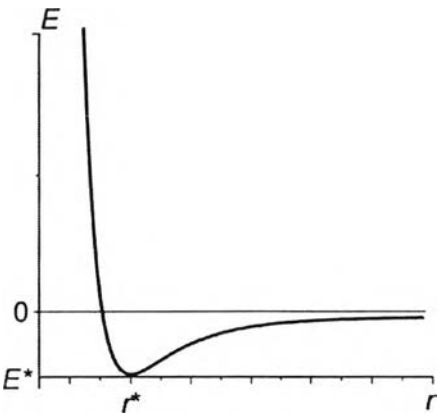
The Lennard-Jones potential is the most commonly used form. It can be written as

$$E_{LJ}(R) = \varepsilon\left[\left(\frac{r_0}{r^{12}}\right)^{12} - 2\left(\frac{r_0}{r^6}\right)^6\right], \tag{3.7}$$

with two parameters: r_0, the diameter, and ε, the well depth. r^{-6} represents the attraction interaction and the repulsive part is given by r^{-12}. The van der Waals interactions are one of the most important for the stability of the biological macromolecules.

Another part of the non-bonded interaction is $E_{\text{electrostatic}}$ which is due to internal distribution of the electron. It is created by positive and negative parts of the molecule. The electrostatic interaction between a pair of atoms is represented by Coulomb potential; D is the effective dielectric function for the medium and r is the distance between two atoms having charges q_i and qk.

$$E_{\text{electrostatic}} = \sum_{\substack{nonbonded \\ pair}} \frac{q_i q_k}{D r_{ik}} \tag{3.8}$$



The empirical potential energy function is differentiable with respect to the atomic coordinates; this gives the value and the direction of the force acting on an atom and thus it can be used in a molecular dynamics simulation.

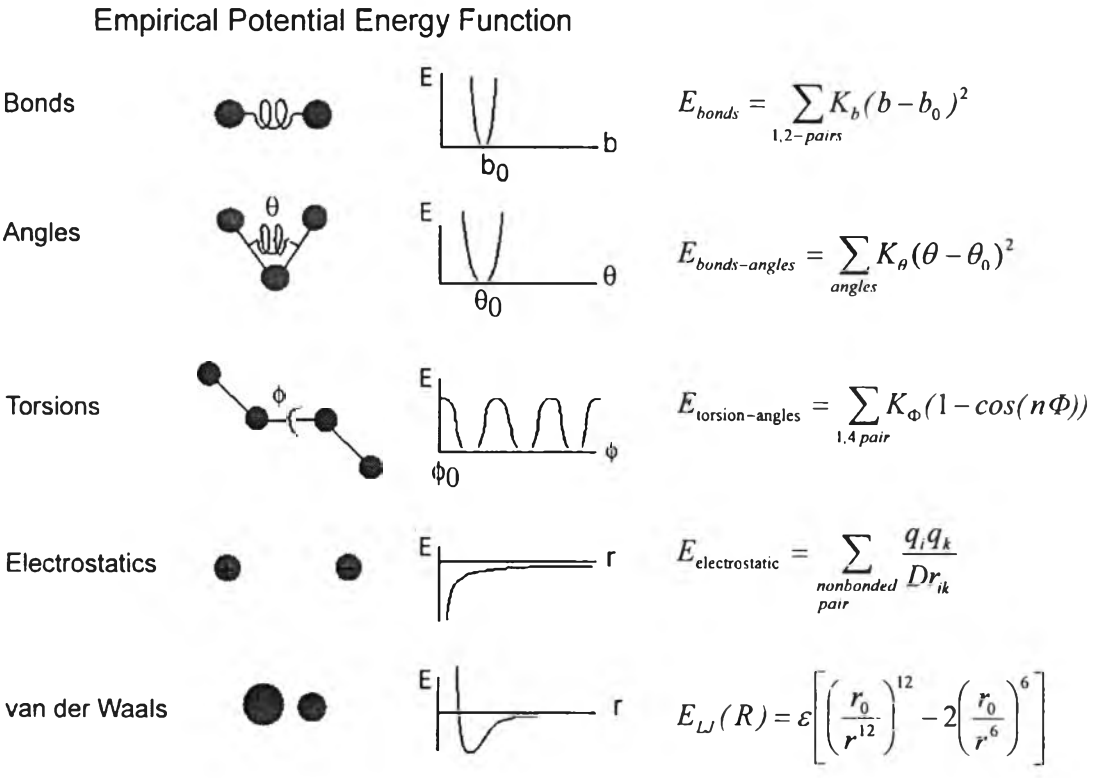


Figure 3.2 Interactions included in representative potential energy function for MD simulation.

3.2 Minimization methods

Using the force field that has been assigned to the atoms in the system it is essential to find a stable point or a minimum on the potential energy surface by adjusting the atomic coordinates, in order to begin dynamics. Derivatives provide information that can be very useful in minimization procedure. At the minimum the net force on each atom vanishes, *i.e.* the derivative or gradient $\nabla V(\mathbf{R}) = 0$. There can be more than one minimum for a large molecule. The minima are called *local* minima. The ideal solution of geometry minimization is the *global minimum*. Due to numerical limitations, however, it is impossible to exactly reach the global minimum or even the local minimum. In practice, local minimum refers to a point on the potential energy surface where the applied minimization procedure cannot further reduce the function value. Mostly, the magnitude of the first derivative is a rigorous way to characterize convergence. The minimum has converged when the derivatives are close to zero. The typical tolerance, for example, in AMBER program [69] is in the range of 10^{-5} to 10^{-6} kcal·mol^{-1}·Å^{-1}. To reach the minimum the structure must be successively updated by changing the coordinates (taking a step) and checking for convergence. Each complete cycle of differentiation and stepping is known as minimization iteration. The efficiency of minimization can be judged by both the number of iterations required to converge and number of function evaluations needed per iteration. Typically, thousands of iterations are required for macromolecules to reach the convergence.

Two first-order minimization methods, which are frequently used in molecular modeling, are *steepest descents* and *conjugate gradient* methods. Both techniques use the first derivatives of the potential function. Additionally, the *Newton-Raphson* method which uses both the first and the second derivatives to locate the minimum, namely the second-order method, is also widely used.

3.2.1 Steepest descents method

The steepest descents method uses the first derivatives to determine the direction to move towards the minimum. This direction is defined by the negative first derivative of the potential energy, $-\nabla V(\mathbf{R})$. However, the derivative (gradient) merely points downhill of potential energy surface but not necessarily direct the minimum.

Thus the technique so-called *line search*, which is used to locate the minimum along the gradient direction, is required to decide how far to move along the direction or to determine the step size. Figure 3.3 shows a contour of the potential energy surface for a simple quadratic function. The gradient direction from the arbitrarily starting point *a* is along the line *ac*. Using the line search (lower curve), the minimum along this line is obtained (point *b*). Thus, minimization path (thick solid line in Figure 3.3) moves from point *a* to *b*. For the next minimization step, then, the new gradient direction is defined at point ***b*** and perpendicular to the previous gradient. The iteration processes until the convergence is reached.

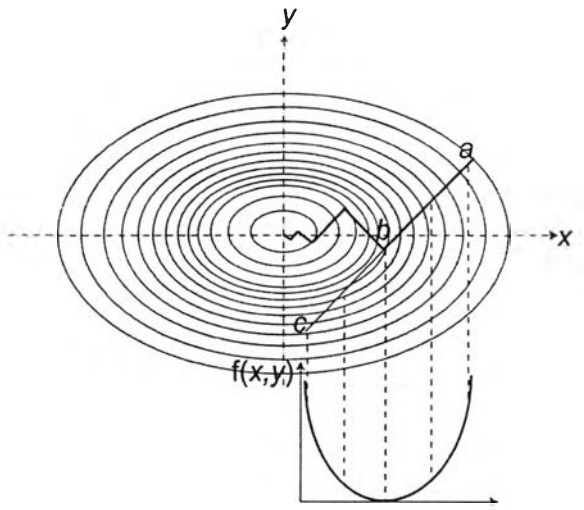


Figure 3.3 Minimization path (thick solid line) given by the steepest descents approach combining with the line search (lower curve) for the simple quadratic function.

However, because each gradient direction must be orthogonal to the previous direction, each segment of the minimization path tends to reverse progress made in an earlier iteration. Consequently, the steepest descents is an insufficient method which gives the oscillating directions along the way to the minimum converges slowly near the minimum, especially on the potential energy surface having narrow valleys. In contrast, the advantage of this method is that it is extremely robust and most likely to generate a lower energy structure regardless of what potential function is or where the process begins. Therefore, this method is often used when the gradients are

large and the configurations are far from the minimum, *e.g.* the configuration obtains from poorly refined crystallographic data or from graphically built models.

3.2.2 Conjugate gradient method

The conjugate gradient method also uses the first derivatives of the potential energy. But instead of using local gradient for going downhill as in the steepest descents method, the conjugate gradient technique defines the new gradient direction for each iteration by using information from previous gradient directions to determine the optimum direction for the line search. Using an algorithm that produces a complete basis set of mutually conjugate directions, each successive step continually refines the direction toward the minimum. Therefore, the conjugate gradient method is more efficient and gives smaller number of iterations to reach the convergence, comparing to the steepest descents method (see Figure 3.4).

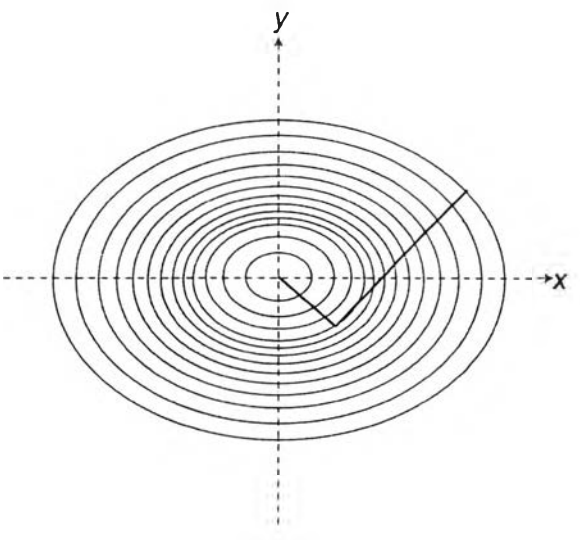


Figure 3.4 Minimization path given by the conjugate gradient method.

Generally, this method converges in approximately M steps for a quadratic function, where M is the number of degrees of freedom of the function. Note that several terms in the potential energy are quadratic. Nevertheless, the disadvantage is that the line minimizations need to be performed accurately in order to ensure that

the conjugate direction is set up correctly and thus time consuming. In addition, the method can be unstable if conformation is so far away from a local minimum.

3.2.3 Newton-Raphson method

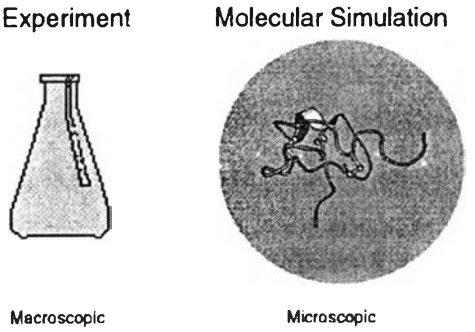
The Newton-Raphson method uses the second derivatives providing information about curvature of the function, as well as the first derivatives providing the gradient. In addition to the use of the gradient to identify a search direction, the curvature of the function is also applied to predict where the function passes through a minimum along that direction, *i.e.* to predict where along the gradient the function will change direction. This gives rise to the Newton-Raphson method to converge faster (if the starting geometry is not too far from the minimum) and more accurate with a tolerance up to 10^{-9} kcal·mol^{-1}·Å^{-1} in comparing to that of 10^{-6} kcal·mol^{-1}·Å^{-1} of the conjugate gradient method. However, for a molecule of N atoms it requires not only the vector of $3N$ first derivatives but also the Hessian matrix of $(3N)^2$ second derivatives, which must be inverted. The need to calculate the Hessian matrix, the iteration makes this algorithm computationally expensive and large storage requirement. In particular, calculating, inverting and storing the second derivative matrix are prohibitive for large systems (usually more than 100 atoms). Furthermore, when a structure is far from the minimum, the minimization can become unstable.

From the above description, it can be seen that different minimization algorithms have different advantages and disadvantages. To optimize the minimization procedure it is usually best to combine several algorithms in the minimization scheme. For instance, the steepest descents method should be used for the first 10-1000 steps of minimization procedure to obtain the configuration close to the local minimum, then the conjugate gradient or the Newton-Raphson method (depending on how large the system) is employed to reach the minimum.

3.3 Molecular dynamics simulations

Computer simulations serve as a complement to conventional experiments, enabling us to learn. The two main families of simulation technique are molecular

dynamics (MD) and Monte Carlo (MC). The obvious advantage of MD over MC is that it gives a route to dynamical properties of the system.



A molecular dynamics simulation (MD) is one of the principal tools in the theoretical study of biological molecules such as molecule of life or proteins. This computational method calculates the time dependent behavior of a molecular system. MD simulations have presented detailed information on the fluctuations and conformational changes of nucleic acids and proteins. These methods are now routinely used to investigate the structure, dynamics and thermodynamics of biological molecules and their complexes. Molecular dynamics simulation techniques are widely used in experimental procedures such as X-ray crystallography and NMR structure determination.

MD methods date back to the 1950's, when Alder and Wainwright [70·71] studied the interactions of hard and elastic spheres leading to important insights into the behavior of simple liquids. The first molecular dynamics simulation of a realistic system was done by Rahman and Stillinger.[72] They investigated the simulation of liquid water in 1974. The first protein simulations performed in 1977 with the simulation of the bovine pancreatic trypsin inhibitor (BPTI). [73] Nowadays, one usually finds molecular dynamics simulations of solvated proteins, solvated DNA, protein-DNA complexes. The number of simulation techniques has seriously extended; the exist now many specialized techniques for particular problems, including mixed quantum mechanical classical simulations, which are being employed to study enzymatic reactions in the context of the full protein and also drug design development.

3.3.1 Basic theory of molecular dynamics

Molecular dynamics simulation consists of the numerical, step-by-step, solution of Newton's second law or the equation of motion, which for a simple atomic system may be written as

$$F_i = m_i a_i = m_i \frac{d^2 r_i}{dt^2}. \tag{3.9}$$

Where F_i is the force acting on atom i, m_i is its mass and a_i is its acceleration (the second derivative of coordinate r_i with respect to time, t).

From Equation 3.9, the acceleration a_i is thus expressed as

$$a_i = \frac{F_i}{m_i} \tag{3.10}$$

$$a_i = \frac{1}{m_i} \frac{dV(R)}{dr_i}, \tag{3.11}$$

which F_i is the first derivative of potential function with respect to coordinate r_i.

$$F_i = \frac{dV(R)}{dr_i}. \tag{3.12}$$

For this purpose we need to be able to carry out the forces F_i acting on the atoms, and these are usually derived from a potential energy $V(R)$, where $R = (r_1; r_2; \ldots, r_N)$ represents the complete set of $3N$ atomic coordinates. The potential energy is a function of the atomic positions ($3N$). Because of the complicated nature of this function, there is no analytical solution to the equations of motion. They must be solved numerically. Numerous numerical algorithms have been developed for integrating the equations of motion.

Time integration algorithms are required to integrate the equation of motion of the interacting particles and follow their trajectories. They are the engine of a molecular dynamics program, which are based on *finite difference methods*. The

equations are solved step-by-step in discrete time interval that is called *time step,* Δt. Knowing the positions and some of their time derivatives at time t, the integration scheme gives the same quantities at a later time $t + \Delta t$. By iterating the procedure, the time evolution of the system can be followed for long times. There are many integration methods widely used in a molecular dynamics simulation.

3.3.2 Integration algorithms

The simplest and most straight forward way to construct an integrator is by expanding the positions, velocities and accelerations in a Taylor series,

$$r(t+\Delta t) = r(t) + v(t)\Delta t + \frac{1}{2}a(t)\Delta t^2 + \frac{1}{6}b(t)\Delta t^3 + \frac{1}{24}c(t)\Delta t^4 + \ldots \tag{3.13}$$

$$v(t+\Delta t) = v(t) + a(t)\Delta t + \frac{1}{2}b(t)\Delta t^2 + \frac{1}{6}c(t)\Delta t^3 + \ldots \tag{3.14}$$

$$a(t+\Delta t) = a(t) + b(t)\Delta t + \frac{1}{2}c(t)\Delta t^2 + \ldots, \tag{3.15}$$

where v is the velocity (the first derivative with respect to time).

In molecular dynamics, the most commonly used time integration algorithm is probably the so-called *Verlet algorithm.* [74] The basic idea is to use two third-order Taylor expansions for the positions $r(t)$, one forward $r(t+\Delta t)$ and one backward $r(t-\Delta t)$ in time. The relation can be written as,

$$r(t+\Delta t) = r(t) + v(t)\Delta t + \frac{1}{2}a(t)\Delta t^2 + \ldots \tag{3.16}$$

$$r(t-\Delta t) = r(t) - v(t)\Delta t + \frac{1}{2}a(t)\Delta t^2 - \ldots, \tag{3.17}$$

Adding the two expressions give

$$r(t+\Delta t)=2r(t)-r(t-\Delta t)+a(t)\Delta t^2). \tag{3.18}$$

This is the basic form of the Verlet algorithm. Because we are integrating Newton's equations, acceleration, $a(t)$, is just the force divided by the mass (Equation 2.11). As one can see, the truncation error of the algorithm when evolving the system by Δt is of the order of four, even if third derivatives do not appear explicitly. This algorithm is at the same time simple to implement, accurate and stable, explaining its large popularity among molecular dynamics simulators.

The advantages of the Verlet algorithm are, *i)* it is straightforward, and *ii)* the storage requirements are modest. The disadvantage is that the algorithm is moderate precision. A problem with this version of the Verlet algorithm is that velocities are not directly generated. While they are not needed for the time evolution, their knowledge is sometimes necessary. Moreover, they are required to compute the kinetic energy K, whose estimate is necessary to test the conservation of the total energy $E=K+V$. This is one of the most important tests to verify that a MD simulation is proceeding correctly. One could compute the velocities from the positions using

$$v(t)=\frac{r(t+\Delta t)-r(t-\Delta t)}{2\Delta t}. \tag{3.19}$$

To overcome this difficulty, some variants of the Verlet algorithm have been developed. They give rise to exactly the same trajectory, and the stored variables are different in memory and at what times. The *leap-frog algorithm* [75] is one of such variants where velocities are handled somewhat better. In this algorithm, the velocities are first calculated at time $t+(1/2)\Delta t$; these are used to calculate the positions, r, at time $r+\Delta t$.

$$v\left(t+\frac{1}{2}\Delta t\right)=v\left(t-\frac{1}{2}\Delta t\right)+a(t)\Delta t \tag{3.20}$$

$$r(t+\Delta t)=r(t)+v\left(t+\frac{1}{2}\Delta t\right)\Delta t. \tag{3.21}$$

The advantage of this algorithm is that the velocities are explicitly calculated, however, the disadvantage is that they are not calculated at the same time as the positions. The velocities at time t can be approximated by the relationship:

$$v(t) = \frac{1}{2}\left[v(t + \frac{1}{2}\Delta t) + v(t - \frac{1}{2}\Delta t) \right]. \tag{3.22}$$

An improved integrator, which was also used in simulations, is *the Velocity Verlet algorithm* [76] which is designed to further improve on the velocity evaluations. The positions, velocities and accelerations at time $t + \Delta t$ are obtained from the same quantities at time t in the following way:

$$r(t + \Delta t) = r(t) + \Delta t v(t) + \frac{1}{2} dt^2 a(t) \tag{3.23}$$

$$v(t + \Delta t) = v(t) + \frac{1}{2}\Delta t\left[a(t) + a(t + \Delta t)\right]. \tag{3.24}$$

The advantage is the best evaluation of velocities but there is also the disadvantage which is computationally more expensive than other integration algorithms (Verlet, Leap-Frog).

3.3.3 Bond constrained

According to biological model, the atoms of macromolecules are linked by covalent bonds. Such bonds vibrate at a very high frequency, so that a typical timestep for the simulation is very small, which is around one femtosecond. This is the cause of the total simulation time. The Bond constrained is the effective techniques to be proposed to increase the timestep.

A common method for the application of bond constraints in molecular simulations employing Cartesian coordinates is the SHAKE procedure invented by Ryckaert *et al.* [77] This algorithm is applied for the fastest processes which are the stretching vibrations, especially those involving hydrogen. Theses degrees of freedom have relatively small influence on interested properties. Therefore it is the advantage to

constrain all these bond lengths, which consequently longer simulation times can be obtained for the same computational cost. However, the SHAKE algorithm works well for time steps 1-2 fs, it is commonly used with SHAKE on the hydrogen atoms. The constraint equation is expressed as

$$g(r) = \left\| r_{ik}(t) - r_{jk}(t) \right\|^2 - d_k^2, k = 1,2...,l, \tag{3.25}$$

where k labels the rigid bond connecting atom i_k and atom j_k of length d_k and the norm $\|\cdot\|$ is the distance.

We need the smallest deviation. However, satisfying one constraint may cause other constraints to be violated, and so it is necessary to iterate round the constraints until they are all satisfied to within some tolerance, *e.g.* 0.00001 Å^2.

3.3.4 Periodic boundary condition

MD simulations are used to investigate on macroscopic behavior and to obtain information that is not easily got from experiments. Nevertheless, computers still cannot model more than a few million atoms at one time, despite the rapid advancement of computer power. These numbers are still far below the real size of most systems. In order to model a macroscopic system in terms of a finite simulation system of N particles, the concept of periodic boundary conditions is often employed.

This idea is represented in Figure 3.5. The original box contains a solute and solvent molecules which is surrounded with identical images of itself. The positions and velocities of corresponding particles in all of the boxes are identical. The common approach is to use a cubic or rectangular parallelepiped box, but other shapes are also possible, *e.g.*, truncated octahedron. By this approach, we obtain what is in effect an infinite sized system.

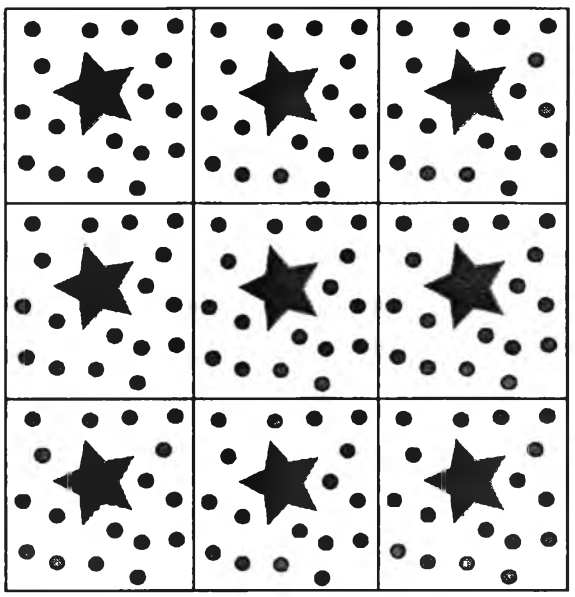


Figure 3.5 Two dimension of periodic boundary condition.

In the way of the simulation, when a molecule moves in the central box, its periodic image in every one of the other boxes moves with exactly the same orientation in exactly the same way. Thus, as a molecule leaves the central box through the right wall, its image will enter the box through the left wall from the neighboring box. The number of particles in the central box (and hence in the entire system) is conserved.

3.3.5 Treatment of long-range electrostatic forces

Currently simulations of bimolecular systems include a very large numbers of atoms ten to hundred of thousands atoms. They are involved over timescales of many nanoseconds. The accurate computation of electrostatic and van der Waals interactions is the most difficult task in computer modeling. The most time consuming part of any molecular dynamics simulation is the calculation of the electrostatic interactions. These interactions fall off as $1/r$, where r is the separation between charges, and have consequently to be considered as long-range.

The need for efficiency treatment of long-range electrostatic interactions in the molecular size (*e.g.*, DNA strands, proteins, membranes, enzymes,

etc.) has been clearly well-known in the last decade. $E_{electrostatic}$ represents a sum over $N(N-1)/2$ pairs; *i.e.*, it is an $O(N^2)$ calculation necessary to evaluate all nonbonded pairs in macromolecular systems. The treatment long-range forces were ignored in the macromolecular simulations, being with the use of 'truncation' or 'cutoff', r_{off}. [78] These methods were developed to limit the computational effort needed by the evaluation of the long-range forces. However, there is problem to select the truncation technique. One can use a straight truncation method, which the electrostatic interactions are zero at r_{off}. The truncation simulations can perform in the old version of AMBER package. [79] Another truncation method, the shifting functions $S(r)$, scales the interaction potential to zero at the specific distance

$$S(r) = \begin{cases} (1-(r/r_{off})^2)^2, r \le r_{off} \\ 0, r > r_{off}. \end{cases} \tag{3.26}$$

One can see that the short-range interactions are disturbed. The distortion and overestimation of the short-range interactions are the drawbacks of the shifting function scheme. The switching function is other smoothing scheme, which brings the potential to zero between a $switches_{on}$ and $switches_{off}$ distance. [80] With a suitable $switches_{on}$ the short-range interactions are not distorted, giving continuous force or potential energy. Although the truncation methods can significantly reduce the amount of computational time for evaluating the electrostatic interactions, these methods are inaccuracy because of finite cutoff distance which restricts severely the infinite character of the system. This may result an unstable geometry for a long simulation.

In order to improve the treatment of the long-range electrostatic interactions, the Ewald summation methods, [81,82] including the particle mesh Ewald (PME) summation method, have been implemented in MD packages. [83] These methods impose a crystal-like periodicity on the system and allow one to calculate all non-bonded electrostatic forces without truncation.

Ewald summation was developed in 1921 to study energetic of ionic crystals. In this method a particle interacts with all the other particles in the simulation box and with all their images in an infinite array of periodic cells. Coulomb's Law

cannot be used to compute electrostatic energy as it is poorly converging conditionally convergent. The Ewald sum splits up the slowly converging sum over the Coulomb potential into three contributions which converge exponentially,

$$U = Ur + Uf + Us. \tag{3.27}$$

Real space part:

$$U_r = \sum_{|n|=0}^{\infty} \frac{q_i q_j}{4\pi\varepsilon_0} \frac{erfc(\alpha|r_{ij}+n|}{|r_{ij}+n|} \tag{3.28}$$

$$erfc(x) = \frac{2}{\sqrt{\pi}} \int_x^{\infty} \exp(-t^2)dt, \tag{3.29}$$

Reciprocal part:

$$U_f = \frac{1}{2L^3\varepsilon_0} \sum_{k\neq 0}^{\infty} \frac{\exp[-k^2/4\alpha^2]}{k^2} \left| \sum_{i=1}^{N} \exp[-ik \bullet r_i] \right|^2. \tag{3.30}$$

Self interaction:

$$U_s = -\frac{1}{4\pi\varepsilon_0} \sum_{i=1}^{N} \frac{q_i^2 \alpha}{\sqrt{\pi}}. \tag{3.31}$$

r_{ij} is the minimum distance between the charges *i* and *j*. The position of each image box is assumed for simplification to be a cube of side L containing N particles. Each particle *i* at position r_i within the cell has a number of image particles at position r_i+n, with $n = n_x L + n_y L + n_z L$, and n_x, n_y, n_z integers. The vector k is reciprocal vector. Here, α is the splitting parameter of the real and reciprocal part. For an optimal α the Ewald summation scale as $O(N^{3/2})$ However, even with these optimizations, Ewald summation remains costly compared to conventional cutoff schemes.

The particle mesh Ewald methods (PME) [84] was has been developed that approximate the reciprocal space term of the standard Ewald summation by a discrete convolution on an interpolation grid, using the discrete Fast-Fourier transforms (FFT). By choosing an appropriate α, the computational cost can be reduce from $O(N^{3/2})$ to $O(NlogN)$. The advantages of the PME are the method for the treatment of long-range forces in macromolecular systems. The high accuracy can be obtained with relatively little increase in computational cost and efficiently implemented into usual MD algorithms such as AMBER package.[69,84]

In molecular dynamics simulations of bimolecular, accurate treatments of the long-range interactions are crucial for achieving stable nanosecond trajectory. The importance of the treatment and also comparison of the methods have been reviewed. [85-88] The particle mesh Ewald method is recommended to use for the long-range electrostatic interactions.[88]

3.4 Procedure to analyze base step parameters program

In this section, we briefly present a basic approach to the calculation of step parameters. The parameters describe the mutual position and orientation of two fragments. The step parameters (see Figure 3.6) are tilt, roll, twist, shift, slide and rise and are determined as rotations and displacements about the *x*, *y* and *z* axes of the base-pair reference tried.

The program is based on the Cambridge University Engineering Department Helix Computation Scheme (CEHS) [89,90] parameters. A detailed algorithm for the evaluation has been already described.[91,92]

Calculation scheme

Definitions of structural parameters. PDB files (Brookhaven Protein Data Bank) will be used to analyze that define residue names, atom names, and atomic coordinates. Local coordinate system is used for each fragment to derive base parameters. The corresponding transformation of coordinates has to be carried out. The

atomic positions for reference bases are the same as defined for Watson-Crick pairs (i.e. A, T, G and C).

The equations for calculating the position and orientation of the coordinate frame of each base in a structure are convenient to express in the matrix form. To derive these equations, consider an atom with absolute x, y and z-coordinates (p_x, p_y, p_z), which can be represented as a column vector p:

$$p = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}.$$

The position of the i^{th} base-pair (or base) is defined by a (1*3) vector, o_i, representing the origin of the triad (**Figure 3.7**);

$$o_i = [o_{ix} \quad o_{iy} \quad o_{iz}].$$

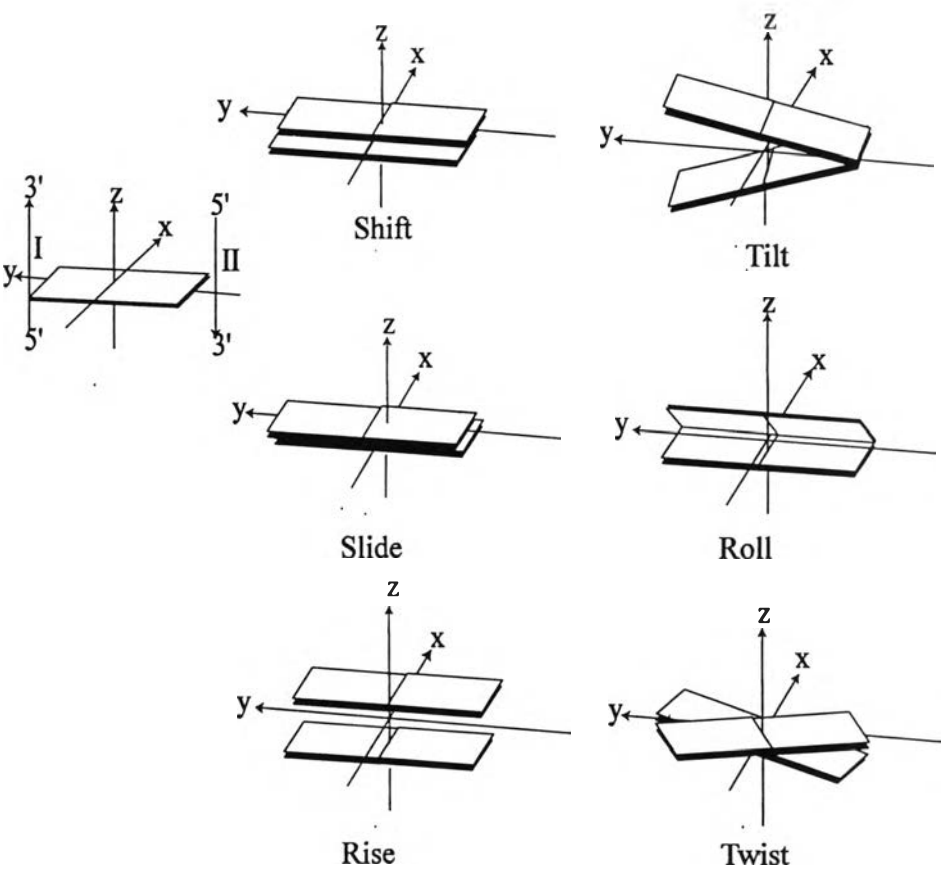


Figure 3.6 Coordinate frame and base step parameters.

The orientation is defined as following:

$$x = \begin{bmatrix} x_{ix} \\ x_{iy} \\ x_{iz} \end{bmatrix}, \qquad y = \begin{bmatrix} y_{ix} \\ y_{iy} \\ y_{iz} \end{bmatrix}, \qquad z = \begin{bmatrix} z_{ix} \\ z_{iy} \\ z_{iz} \end{bmatrix}.$$

They are the absolute x, y and z-unit vector, respectively, that define the x, y and z-axes of their reference triad of base-pair i. The relationship between the local coordinate frame of base i and the absolute coordinate frame is illustrated in two dimensions in **Figure 3.7**.

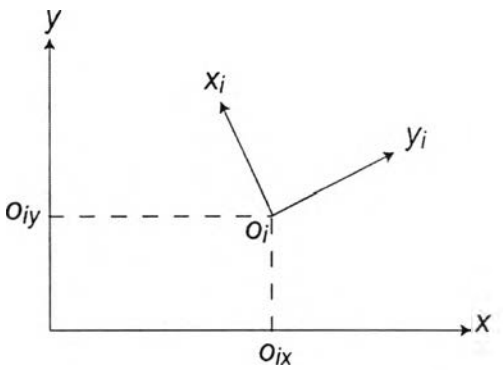


Figure 3.7 The x and y-coordinates of an origin as measured in the absolute coordinate frame (o_x,o_y) of base-pair i.

It can be seen from Figure 3.8 that in order to define the reference triads of the bases and base-pairs, one strand needs to be arbitrarily assigned strand I and the other strand II. Briefly, the reference triads are defined as follows:

Base-pair triad: The origin is the midpoint of the line connection C6 for pyrimidines and C8 for purines. The C6-C8 line gives the y-axis. Its positive direction points from strand II to strand I. The z-axis is defined as the normal to the least-squares plane through atoms of the base pair and its positive direction is along the 5′→3′ direction of strand I. The x-axis completes a right-handed triad with the y and z-axes. The positive x-axis direction points along the short axis of the base-pair form the minor groove side to the major groove side.

Base triad: The *y*-axis always point from strand II to strand I, and it lies along N1-C4 (purine) and N3-C6 (pyrimidine). The origin is given by the midpoint of the N1-C4 for purine and N3-C6 for pyrimidine. It should be clear from Figure 3.8. For the *z*-axis, it is defined as the normal of the least-squares plane through all atoms in the base excluding hydrogen and C1 atom. It is positive direction is along the 5′→ 3′ direction.

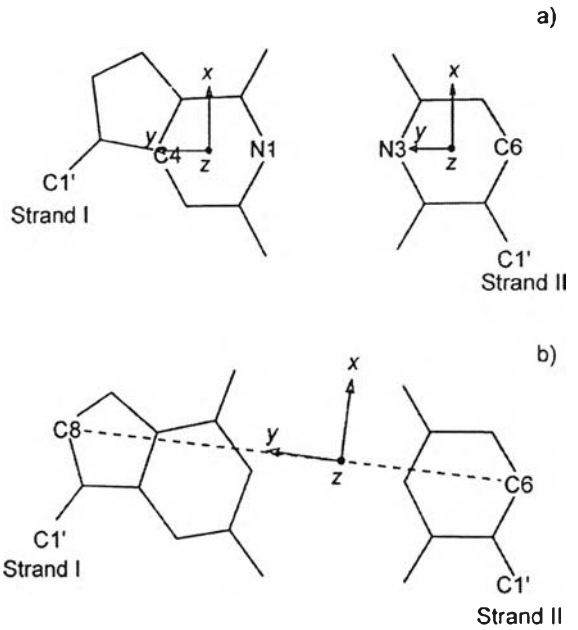
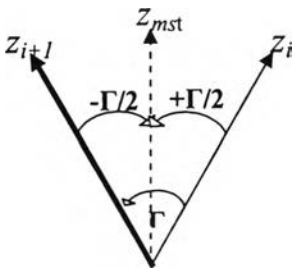


Figure 3.8 Schematic description of the base and base-pair reference frames with respect to the actual bases: a) give the base reference frames and b) gives the base-pair reference frame.

Defining and calculating base-step parameters.

1) We now apply the angle Roll-Tilt (Γ) which is defined as the magnitude of the angle between z_i and z_{i+1},

$$\Gamma = \cos^{-1}(z_i \bullet z_{i+1}). \tag{3.32}$$



2) Next we calculate the Roll-Tilt axis (*rt*), which need to take the cross product of *z*-axis of the two successive base-pair. This immediately gives the direction of Roll-Tilt axis,

$$rt = z_i \times z_{i+1}. \tag{3.33}$$

3) We now rotate the two base-pair triads: one base pair triad is rotated about Roll-Tilt axis by angle $\left(+\Gamma/2\right)$, and the other is rotated $\left(-\Gamma/2\right)$ also about Roll-Tilt axis:

$$\omega = \left(\pm\Gamma/2\right), \tag{3.34}$$

$$\begin{bmatrix} x'_{ix} & y'_{ix} & z'_{ix} \\ x'_{iy} & y'_{iy} & z'_{iy} \\ x'_{iz} & y'_{iz} & z'_{iz} \end{bmatrix} = R_{rt}(\omega) \begin{bmatrix} x_{ix} & y_{ix} & z_{ix} \\ x_{iy} & y_{iy} & z_{iy} \\ x_{iz} & y_{iz} & z_{iz} \end{bmatrix}, \tag{3.35}$$

which $R_{rt}(\omega)$ [93] describing a rotation of magnitude ω about and arbitrary unit vector $\hat{rt} = rt_x\hat{i} + rt_y\hat{j} + rt_z\hat{k}$:

$$R_{rt}(\omega) = \begin{bmatrix} \cos\omega + (1-\cos\omega)rt_x^2 & (1-\cos\omega)rt_x rt_y - rt_z\sin\omega & (1-\cos\omega)rt_x rt_z + rt_y\sin\omega \\ (1-\cos\omega)rt_x rt_y + rt_z\sin\omega & \cos\omega + (1-\cos\omega)rt_y^2 & (1-\cos\omega)rt_y rt_z - rt_x\sin\omega \\ (1-\cos\omega)rt_x rt_z - rt_y\sin\omega & (1-\cos\omega)rt_y rt_z + rt_x\sin\omega & \cos\omega + (1-\cos\omega)rt_z^2 \end{bmatrix}$$

(3.36)

We then rotate two base-pair triads about Roll-Tilt axis until their x-y planes are precisely parallel; the current z-axis is the axis of the mid-step triad (MST). The x- and y-axes of MST lies along the bisector of the angle between the transformed x and y axis of the i^{th} and $(i+1)^{th}$ base-pair. The Roll-Tilt axis, in general, will not bisect the angle between the two transform y-axis. It is in the x-y plane of the MST, which is inclined at an angle (ϕ) to the y axis (See Figure 3.9).

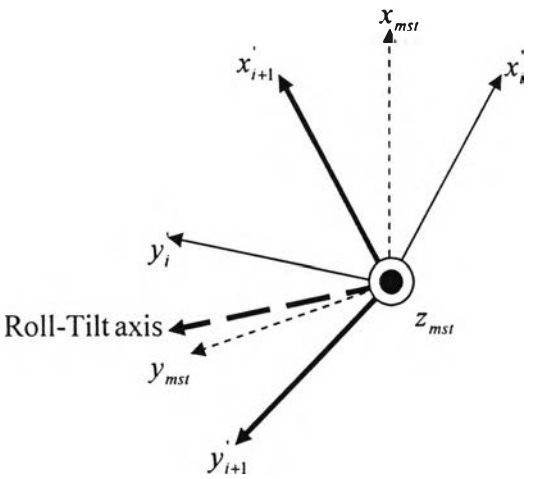


Figure 3.9 The Transformed base-pair reference triads and MST

4) The angle between the two transformed y- (or x-) axis of the base-pair is Twist (Ω), which is given by:

$$\Omega = \cos^{-1}(y'_i + y'_{i+1}). \tag{3.37}$$

5) To determine the angle between the Roll-Tilt axis and the MST y-axis

$$\phi = \cos^{-1}(\hat{rt} \cdot y_{mst}). \tag{3.38}$$

6) Roll and Tilt, which are define as the component of RollTilt along the y- and x-axes of the MST, respectively, are given as following:

$$\rho = \Gamma\cos(\phi), \qquad \tau = \Gamma\sin(\phi). \tag{3.39}$$

ต้นฉบับ หน้าขาดหาย

Test calculations were performed on 12 base pairs shown in Table 3.1. It lists the step parameters (translations in Å, rotations in degree) analyzed by new code. The most considerable difference is found for the rotation parameters. The rise parameter of the third pair is 27.73°, while carried out by 3DNA program is 27.96°. For the tilt and roll parameters, the most different are found about 0.15° and -0.28° for the third and eighth pair, respectively. The code for calculation step parameters of base pair almost reproduces the step parameters given by X3DNA program.

This program is modified for the analysis the step parameters of DNA-chromophore complex. It can describe the position and orientation of chromophore relative with base in the complex.