CHAPTER 2

THEORY

2.1 Introduction of molecular dynamics simulations

Molecular dynamics (MD) simulations are important tools for understanding the properties of assemblies of molecules in terms of their structure and the microscopic interactions between them. This serves as a complement to conventional experiments, enabling us to learn something new that cannot be found out in the other ways. The idea is to calculate the forces acting on the atoms in a molecular system and analyze their motion. When enough information on the motion of the individual atoms has been gathered, it is possible to condense it all using the methods of statistic mechanics to deduce the bulk properties of the material. These properties include the structure (e. g. crystal structure, predicted x-ray and neutron diffraction patterns), thermodynamics (e. g. enthalpy, temperature, pressure) and transport properties (e. g. thermal conductivity, viscosity, and diffusion). In addition, MD can be used to investigate the detailed atomistic mechanisms underlying these properties and compare them with theory. It is a valuable bridge[58] between experiment and theory as shown in Figure 2.1

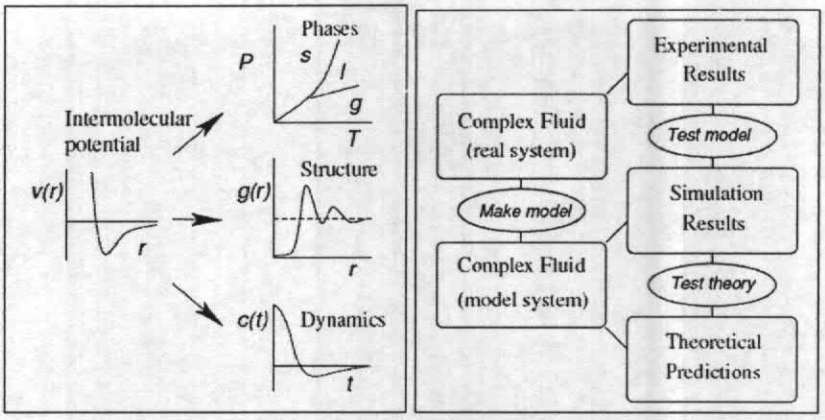

Figure 2.1 Simulations as a bridge between (a) microscopic and macroscopic; (b) theory and experiment[59].

It has been 25 years since the first MD simulations of a macromolecule of biological interest was published[60]. The simulation concerned the bovine pancreatic trypsin inhibitor (BPTI), which has served as the 'hydrogen molecule' of protein dynamics because of its small size, high stability and relatively accurate X-ray structure available in 1975. Two years after the BPTI simulation, it was recognized[61, 62] that thermal (B) factor calculated during X-ray crystallographic refinement could be used to infer the internal motions of proteins. During the following 10 years, a wide range of motional phenomena were investigated by MD simulations of proteins and nucleic acids. Most of these studies[63] focused on the physical aspect of the internal motions and the interpretation of experiments. They include the analysis of fluorescence depolarization of tryptophan residues[64], the role of dynamics in measured NMR parameters[65], the effect of solvent and temperature on protein structure and dynamics[66, 67], and now widely used simulated annealing methods for X-ray structure refinement[68] and NMR structure determination[69]. Currently, the increase in the number of studies using MD to simulate the properties of biological macromolecules has been fueled by the general

availability of programs and the computing power required for meaningful studied. Understanding of the molecular dynamics simulations step have been shown in Figure 2.2.

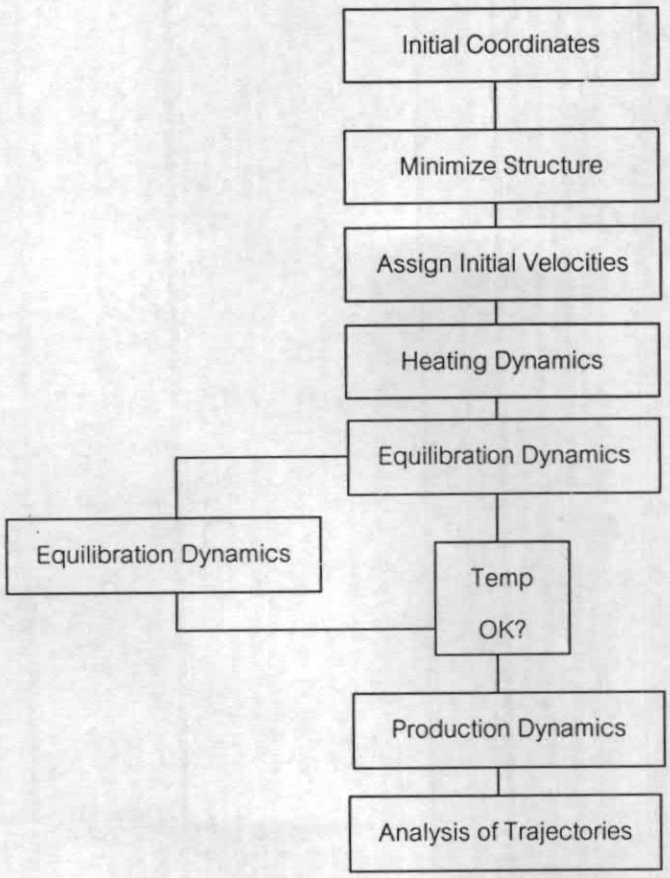

Figure 2.2 Steps of a molecular dynamics simulations.

2.1.1 Basic theory of molecular dynamics simulations

To calculate the dynamics of the system, that is the position of each atom as a function of time, Newton's classical equation is used in the molecular dynamics formalism to simulate atomic motion:

$$\text{Force} = \text{mass x acceleration } (F_i = m_i a_i) \qquad (2.1)$$

The rate and direction of motion (velocity) are governed by the forces that the atoms of the system exert on each other as described by Newton's equation. In practice, the atoms are assigned initial velocities that conform to the total kinetic energy of the system, which in turn, is dictated by the desired simulation temperature. This is carried out by slowly "heating" the system (initially at absolute zero) and then allowing the energy to equilibrate among the constituent atoms. The basic ingradients of molecular dynamics are the calculation of the force on each atom, and from that information, the position of each atom throughout a specified of time.

The force (F_i) on an atom can be calculated from the change in energy (E) between its current position and its position a small distance (r_i) away. This can be recognized as the derivative of the energy with respect to the change in the atom's position as Equation 2.2.

$$-\frac{dE}{dr_i} = F_i \qquad (2.2)$$

Knowledge of the atomic forces and masses can then be used to solve for the positions of each atom along a series of extremely small time steps (dt). The resulting series of snapshots of structural changes over time is called a trajectory. The use of this method to compute trajectories can be more easily seen when Newton's equation is expressed in the following form (Equation 2.3).

$$-\frac{dE}{dr_i} = m_i \frac{d^2 r_i}{dt^2} \qquad (2.3)$$

In practice, trajectories are not directly obtained from Newton's equation due to lack of an analytical solution. First, the atomic accelerations (a_i) are computed from the forces and masses. The velocities are next calculated from the accelerations based on the following relationship (Equation 2.4).

$$a_i = \frac{dv_i}{dt} \tag{2.4}$$

The positions are calculated from the velocities (v_i) (Equation 2.5).

$$v_i = \frac{dr_i}{dt} \tag{2.5}$$

The initial atomic positions at time "t" are used to predict the atomic positions at time "$t+\Delta t$". The positions at "$t+\Delta t$" are used to predict the positions at "$t+2*\Delta t$", and so on.

Therefore, to calculate a trajectory, the initial positions of the atoms, an initial distribution of velocities and the acceleration are need for determined by the gradient of the potential energy function. The initial distribution of velocities is usually computed randomly from a Maxwell-Boltzman or Gaussian distribution at given temperature, which gives the probability ($p(v_{ix})$) that an atom i has velocity v_i in the x direction at a temperature T.

$$p(v_{ix}) = \left(\frac{m_i}{2\pi k_B T}\right)^{1/2} \exp\left[-\frac{1}{2}\frac{m_i v_{ix}^2}{k_B T}\right] \tag{2.6}$$

The temperature can be related form the kinetic energy Equation 2.7:

$$T = \frac{1}{3Nk} \sum_{i=1}^{N} \frac{\left| p_i^2 \right|}{2m_i} \tag{2.7}$$

where N is the number of atoms in the system.

To generate molecular dynamics trajectories with continuous potential models, the algorithms for integrating (more details in 2.1.4) the equations of motion are used in molecular dynamic simulations.

2.1.2 Potential energy functions

According to equation 2.3, the energy (E), a function of the atomic position (R), is obtained from summation between internal or bonded terms, and summation of external or non-bonded terms corresponding to equation 2.8.

$$E(R) = E_{bonded} + E_{non\text{-}bonded} \tag{2.8}$$

2.1.3.1 Bonded Terms

The bonded terms are an energy combination of bonds, angles, and torsional angles following equation 2.9, 2.10, and 2.11, respectively.

$$E_{bond} = \sum_{bonds} \frac{k_b}{2}(r - r_0)^2 \qquad (2.9)$$

$$E_{angle} = \sum_{angles} \frac{k_\theta}{2}(\theta - \theta_0)^2 \qquad (2.10)$$

$$E_{torsion} = \sum_{torsions} \frac{V_n}{2}(1 + \cos(n\omega - \gamma)) \qquad (2.11)$$

Where K_b and K_θ are present the force constant for bond and angle while V_n, n, ω, and γ are the barrier height, the number of wave, torsional angle, and the torsional angle at minima, respectively, as Figure 2.4. Bond length, bond angle, and dihedral angle are presented in r, and θ, respectively, while those subscripts are states as the ideal value.

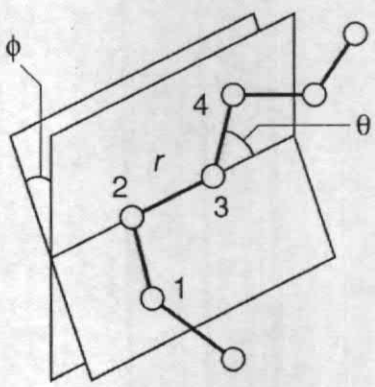

Figure 2.3 Geometry of a simple chain molecule, illustrating the definition of interatomic distance r_{23}, bend angle θ_{234}, and torsion angle ϕ_{1234}.

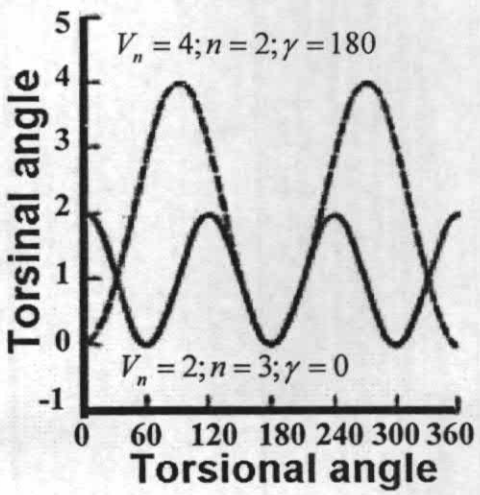

Figure 2.4 Tosional potential varies as shown for different values of V_n, n, and γ

2.1.3.2 Non-bonded Terms

The non-bonded terms are calculated from summation of van der Waals and electrostatic according to equations 2.12 and 2.13.

$$\text{Lennard-Jones potential} = \sum_{i=1}^{N}\sum_{j=i+1}^{N} 4\varepsilon_{ij}\left[\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}\right] \qquad (2.12)$$

$$\text{Electrostatic interaction} = \sum_{i=1}^{N}\sum_{j=i+1}^{N}\left(\frac{q_i q_j}{4\pi\varepsilon_0 r_{ij}}\right) \qquad (2.13)$$

The electrostatic contribution is calculated using Coulomb's law from partial charges (q_i or q_j) associated with each atom and van der Waals contribution as a Lennard-Jones potential with appropriate ε_{ij}, σ_{ij} parameters.

2.1.4 Integrating the equation of motion

2.1.4.1 The Verlet algorithm

In molecular dynamics, the most commonly used time integration algorithm is probably the so-called Verlet algorithm[70]. The basic idea is to write two third-order Taylor expansions for the positions $r(t)$, one forward and one

backward in time. Calling v the velocities, a the accelerations, and b the third derivatives of r with respect to t, one has:

$$r(t+\Delta t) = r(t) + v(t)\Delta t + \left(\frac{1}{2}\right)a(t)\Delta t^2 + \left(\frac{1}{6}\right)b(t)\Delta t^3 + O(\Delta t^4) \qquad (2.14)$$

$$r(t-\Delta t) = r(t) - v(t)\Delta t + \left(\frac{1}{2}\right)a(t)\Delta t^2 - \left(\frac{1}{6}\right)b(t)\Delta t^3 + O(\Delta t^4) \qquad (2.15)$$

Adding the two expressions gives

$$r(t+\Delta t) = 2r(t) - r(t-\Delta t) + a(t)\Delta t^2 + O(\Delta t^4) \qquad (2.16)$$

This is the basic form of the Verlet algorithm. Since we are integrating Newton's equations, $a(t)$ is just the force divided by the mass, and the force is in turn a function of the positions $r(t)$:

$$a(t) = -\left(\frac{1}{m}\right)\nabla V(r(t)) \qquad (2.17)$$

As one can immediately see, the truncation error of the algorithm when evolving the system by Δt is of the order of Δt^4, 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.

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 evaluation 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 by using

$$v(t) = \frac{r(t+\Delta t) - r(t-\Delta t)}{2\Delta t} \tag{2.18}$$

However, the error associated to this expression is of order Δt^2 rather than Δt^4.

To overcome this difficulty, some variants of the Verlet algorithm have been developed. They give rise to exactly the same trajectory, and differ in what variables are stored in memory and at what times. The *leap-frog* algorithm, not reported here, is one of such variants[71] where velocities are handled somewhat better.

An even better implementation of the same basic algorithm is the so-called *velocity Verlet* scheme, where positions, velocities and accelerations at time $t+\Delta t$ are obtained from the same quantities at time t in the following way:

$$r(t+\Delta t) = r(t) + v(t)\Delta t + \left(\frac{1}{2}\right)a(t)\Delta t^2 \tag{2.19}$$

$$r\left(t + {}^{\Delta t}/_{2}\right) = v(t) + \left(\frac{1}{2}\right)a(t)\Delta t \tag{2.20}$$

$$a\left(t+\Delta t\right) = -\left({}^{1}/_{m}\right)\nabla V(r(t+\Delta t)) \tag{2.21}$$

$$v(t+\Delta t) = v\left(t + {}^{\Delta t}/_{2}\right) + \left({}^{1}/_{2}\right)a(t+\Delta t)\Delta t \tag{2.22}$$

2.1.4.2 Leap-frog algorithm

The first method used is the *leapfrog* algorithm, which is a modified version of the *Verlet* algorithm. The Verlet algorithm uses the positions and accelerations at the time t and the positions at the time $t-\Delta t$ to predict the positions at the time $t+\Delta t$, where Δt is the integration step. From a Taylor expansion of the 3-rd order, we obtain

$$r_i(t+\Delta t)=2r_i(t)-r_i(t-\Delta t)+\ddot{r}_i(t)\Delta t^2 \qquad (2.23)$$

The error in the atomic positions is of the order of Δt^4. The velocities are obtained from the basic definition of differentiation

$$r_i(t)=\frac{r_i(t+\Delta t)-r_i(t-\Delta t)}{2\Delta t}, \qquad (2.24)$$

with an error of the order of Δt^2. To obtain more accurate velocities, the leapfrog algorithm is used, using velocities at half time step

$$r_i\left(t+\frac{\Delta t}{2}\right)=r_i\left(t-\frac{\Delta t}{2}\right)+r_i(t)\Delta t \qquad (2.25)$$

The velocities at time t can be also computed from

$$r_i(t)=\frac{r_i\left(t+\frac{\Delta t}{2}\right)+r_i\left(t-\frac{\Delta t}{2}\right)}{2} \qquad (2.26)$$

This is useful when the kinetic energy is needed at time t, as for example in the case where velocity rescaling must be carried out (see below). The atomic positions are then obtained from

$$r_i(t+\Delta t) = r_i(t) + r_i\left(t+\frac{\Delta t}{2}\right)\Delta t \qquad (2.27)$$

The leapfrog algorithm is computationally less expensive than the Predictor-Corrector approach for example, and requires less storage. This could be an important advantage in the case of large scale calculations. Moreover, the conservation of energy is respected, even at large time steps. Therefore, the computation time could be greatly decreased when this algorithm is used. However, when more accurate velocities and positions are needed, another algorithm should be implemented, like the Predictor-Corrector algorithm.

2.1.5 Periodic Boundary Conditions

To eliminate surface effect from the computation, the periodic boundary is applied for neglect all system sizes. The cubical simulation box is replicated throughout space to form an infinite lattice for this condition. A molecule moves in the central box and its periodic image in every one of the other boxes moves with exactly the same orientation in exactly the same way for the simulation. Thus, as a molecule leaves the central box, one of its images will enter through the opposite face. There are no walls at the boundary of the central box, and the system has no surface. The central box simply forms a convenient coordinate system for measuring locations of the N molecules.

A two dimensional version of such a periodic system is shown in Figure 2.5. As a particle moves through a boundary, all its corresponding images move across their corresponding boundaries. The number of particles in the central box in the entire system is conserved.

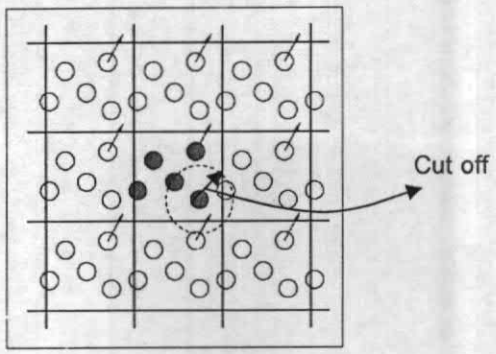

Figure 2.5 Periodic boundary conditions. As a particle moves out of the simulation box, an image particle moves in to replace it.

2.1.6 Treatment of non-bonded interactions

The most time-consuming part of molecular dynamics simulations is the computation of the non-bonded energies or forces. The numbers of bond, angle, and torsional angle terms are all proportion of the number of atoms while non-bonded terms require to be estimated increases as the square of the number of atoms (N^2). To deal with the problem of non-bonded interaction, using a non-bonded cutoff applying the minimum image convention is the most popular way. The energy is computed with the closet atom or image, as shown in Figure 2.5 (dot circle). The all pair interactions, further apart that the cut off are set to zero when the non-bonded cut off is applied.

The simplest of non-bonded interaction is van der Waals. The cut-off distance is used to truncate the van der Waals interactions as shown in Figure 2.5. While the electrostatic interactions is slightly more complicated due to the multiple time-steping for full electrostatics interactions. The Ewald summation method is used for truncating this long-range interaction. The summing interaction between an ion and all its periodic image. The construction of the periodic systems in Ewald summation method is illustrated in Figure 2.6.

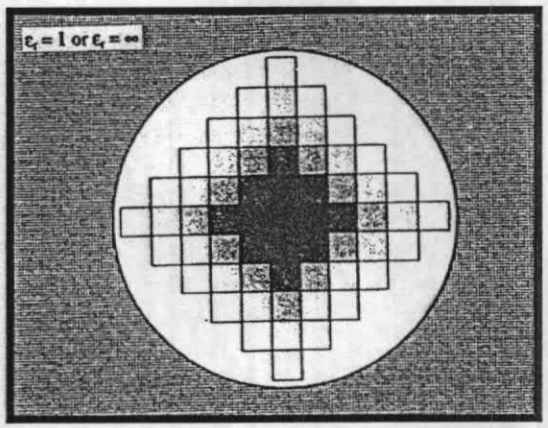

Figure 2.6 Construction of a system of periodic cells in the Ewald method[72].

2.1.7 Energy minimization

In molecular modeling, the minimum points on the energy surface is essentially interested. The arrangements of minimum energy for the atoms correspond to the stable system. There are a large number of the minima on the energy surface. The minimum with the lowest energy is known as the global energy minimum. Most of minimization algorithms can only go downhill on the energy surface and locate the minimum that is the nearest to the starting point.

The minimization problem can be formally found that we require the values of variable, x_1, x_2, x_3, ..., x_i, for the function (f) where f has a minimum value. At the minimum point the first derivative of the function with respect to each of the variables is zero and the second derivatives are all positive (Equation 2.28).

$$\frac{\partial f}{\partial x_i} = 0; \frac{\partial^2 f}{\partial x_i^2} > 0 \qquad (2.28)$$

The minimization algorithms can be divided into two groups, which are non-derivative (*i. e.* the simplex, and sequential univariate method) and derivative minimization (i. e. the steepest descents, line search in one dimension, conjugate gradient, and Newton-Raphson method). The first derivative minimization algorithm, teepest descents and conjugate gradient, frequently used in the molecular modeling.

2.1.7.1 Steepest descent method

The energy is calculated for the initial geometry and then again after one of the atoms has been moved in a small increment in one of directions of the coordinates system. This process is repeated for all atoms which finally are moved to a new position downhill on the energy surface. The procedure stops when a predetermined threshold condition is fulfilled. The optimization process is slow near the minimum, and consequently, the steepest descent method is often used for structures far from the minimum as a first, rough and introductory run followed by subsequent minimization employing a more advanced algorithm like the conjugate gradient.

2.1.7.2 Conjugate gradient method

The conjugate gradient algorithm accumulates the information about the function from the one iteration to the next. With this proceeding the reverse of the progress made in an earlier iteration can be avoided. For each minimization step the gradient is calculated and used as additional information for computing the new direction vector of the minimization procedure. Thus, each successive step refines the direction towards the minimum. The computational effort and the storage requirements are greater than for steepest descent, but conjugate gradients is the method of choice for larger systems. The greater total computational expense and the longer time per iteration is more than compensated by the more efficient convergence to minimum achieved by conjugate gradients. This method is more efficient for the structure near the minimum.

2.2 Binding free energy of protein-ligand complexes

"Free energy is arguably the most important general concept in chemistry"[73] The free energy for the formation of a protein-ligand complexes is equal to:

$$\Delta G_{bind} = \Delta H - T\Delta S \qquad (2.29)$$

that, under equilibrium conditions, is equal to:

$$\Delta G^{\circ} = \Delta H^{\circ} - T\Delta S^{\circ} = -RT\ln(K_{eq}) \qquad (2.30)$$

The binding free energy (ΔG_{bind}) contains both enthalpic (ΔH) and entropic (ΔS) contributions under temperature (T), that in many reactions of biological systems compensate each other[74]. Where K_{eq} represents equilibrium constant. The thermodynamic rationale for enthalpy-entropy compensation is based on the fact that, as the binding becomes stronger, enthalpy becomes more negative and entropy

concomitantly tends to decrease due to formation of a tight complex. On the contrary, as the binding becomes weaker, enthalpy become less negative and entropy tends to increase due to the formation of a loose complex[51].

The recently reported molecular mechanics Poisson-Boltzman surface area (MM-PBSA) and molecular mechanics generalized Born surface area (MM-GBSA) methods[75-77] for free energy calculation of protein-ligand complex (see next section).

2.2.1 Molecular mechanics Poisson-Boltzman(Generalized Born) surface area (MM-PB(GB)SA)

A hybrid method termed MM-PB(GB)SA combining molecular mechanics and continuum solvent calculations has recently been developed to analyze the free energies of binding and relative free energies of different conformations[78]. The MM-PB(GB)SA method extracts solute conformations or snapshots from a MD trajectory carried out with explicit solvent, typically a periodic box with water and counter-ions. For each snapshot, solvent molecules are removed to obtain the molecular mechanics potential as in the simulation, but in the absence of cut-offs in order to evaluate the non-bonded interactions. The free energy of the complex is calculated using the following thermodynamic cycle (Figure 2.7):

In gas

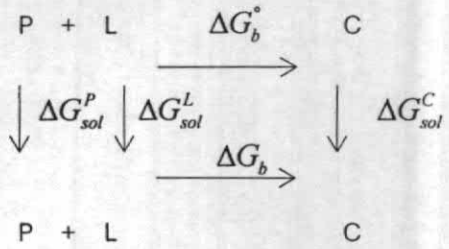

Figure 2.7 Thermodynamic cycle.

The free energy of each molecular species (ΔG_b^X) is approximated by

$$\Delta G_b^X = \Delta G_b^{o,X} + \Delta E_{sol}^X \qquad (2.31)$$

where $\Delta G_b^{\circ,X}$ and ΔE_{sol}^X are the molecular mechanics free energy and the solvation free energy of species X which are complex (C), protein (P), and ligand (L), respectively. A further partitioning of these energy components is defined by the Equations (2.32 and 2.33):

$$\Delta G_b^{o,X} = \Delta H - T\Delta S = \Delta E_{\text{int}}^{ele} + \Delta E_{\text{int}}^{vdw} - T\Delta S \qquad (2.32)$$

where enthalpic (ΔH) equal to the combination between $\Delta E_{\text{int}}^{ele}$ and $\Delta E_{\text{int}}^{vdw}$ represented the electrostatic and the van der Waals interactions, respectively. The solvation free energy is comprised of two components:

$$\Delta E_{sol}^X = \Delta E_{sol}^{ele} + \Delta E_{sol}^{nonpolar} \qquad (2.33)$$

where ΔE_{sol}^{ele} is the polar contribution to solvation, and $\Delta E_{sol}^{nonpolar}$ is the nonpolar solvation term. The former component was calculated using the PB or GB calculation, whereas the latter term is determined using Equation 2.34.:

$$\Delta E_{sol}^{nonpolar} = \gamma SASA + b \qquad (2.34)$$

where γ represents surface tension is set to 0.0072 or 0.0542 and b was set equal to 0.92 as in the work of Still and coworker[79].

Subsequently, the absolute binding free energy (ΔG_b) between protein and ligand is calculated as

$$\Delta G_b = \Delta G_b^C - \Delta G_b^P - \Delta G_b^L \qquad (2.35)$$

where ΔG_b^C, ΔG_b^P, and ΔG_b^L are the free energy of the complex, protein and ligand, respectively.

2.2.1.1 Poisson-Boltzmman (PB) model

Poisson-Boltzmann approach is performed as a low dielectric cavity with embedded atomic partial charges for the macromolecule. The dielectric constant of the cavity is typically set between 2 and 4 to take into account electronic polarization and the limited flexibility of the macromolecule[80]. The effected motions of the solvent molecules are much faster than those of the molecule and the ions, taken into account on average through a continuum of high dielectric constant[81].

The average electrostatic potential ($\overline{U}$) is obtained from the charge density embedded in the molecule (ρ^f) and by the average charge density due to the mobile ions ρ^{-m}, via the Poisson equation:

$$\vec{\nabla} \cdot (\varepsilon \vec{\nabla} \overline{U}) = -4\pi\rho^{-m} - 4\pi\rho^f \qquad (2.36)$$

The unit expression is centimeter gram second-electrostatic for the position-dependent dielectric constant, ε, and all terms. The charge density ρ^{-m} can be computed in terms of the bulk concentrations and a potential of mean force:

$$\rho^{-m} = \sum_i c_i^{\infty} z_i q \exp\left(\frac{-w_i}{kT}\right) \tag{2.37}$$

where c_i^{∞} is the concentration of ion i at an infinite distance from the molecule (or at any reference position where the potential of mean force w_i is set to zero), z_i is its charge number, q is the proton charge, k is the Boltzmann constant and T is the temperature.

The key assumptions to obtain the PBE are that the potentials of mean force are given by $w_i = z_i q U$ and that U is equal to the average electrostatic potential $\overline{U}$:

$$\vec{\nabla} \cdot (\varepsilon \vec{\nabla} \overline{U}) = -4\pi \sum_i c_i^{\infty} z_i q \exp\left(\frac{-z_i q U}{kT}\right) - 4\pi\rho^f \tag{2.38}$$

When the term $(\frac{z_i q U}{kT}) << 1$ the exponential can be expanded in Taylor series, retaining only the first two terms. Due to electroneutrality, $\sum_i c_i^{\infty} z_i q = 0$, the LPBE is obtained:

$$\vec{\nabla} \cdot (\varepsilon \vec{\nabla} \overline{U}) = \left(\sum_i 4\pi c_i^{\infty} \frac{z_i^2 q^2}{kT}\right) U - 4\pi\rho^f \tag{2.39}$$

2.2.1.2 Generalized Born model

As it is known that Poisson calculation is a very time demanding method, therefore, generalized Born model is developed to mimic Posion energy. The original Born theory[82] describes the solvation energy of a spherical charge q locating in the center of spherical cavity of radius R following equation 2.40

$$E_B = -\frac{k'q^2}{R}, \qquad (2.40)$$
$$k' = k(\varepsilon_{solute}^{-1} - \varepsilon_{solvent}^{-1}),$$

As k = -166.0 (Å kcal)/[mol(atomic charge)^2], the charge (q) is in the atomic units while, the radius R has units of Å. The ε_{solute} and $\varepsilon_{solvent}$ are the dielectric constants of the solute and solvent, respectively. *Generalized* Born theory explains the electrostatic energy of two or more atomic charges in a cavity of arbitrary shape. The most successful of GB formulation is Still and co-workers[79]:

$$E_{GB} = -k'\sum_{i,j}\frac{q_i q_j}{\sqrt{r_{ij}^2 + \alpha_i\alpha_j \exp(-r_{ij}^2/4\alpha_i\alpha_j)}}, \qquad (2.41)$$

The r_{ij} is the distance between atoms i and j, while α_i is the atomic Born radius of the i th atom.

2.2.1.3 Solvent accessible surface area (SASA)

Lee and Richards[83] introduced originally the concept of the solvent accessible surface of a protein molecule that is a way of quantifying hydrophobic burial. The solvent accessible area (ASA) explains the area over which contact between protein and solvent as shown in Figure 2.8.

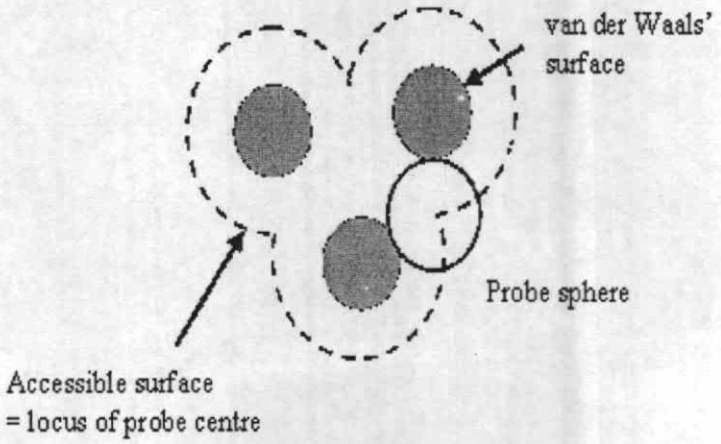

Figure 2.8 Accessible surface of a molecule, defined as the location of the center of a solvent molecule as it rolls over the van der Waals surface of the protein.

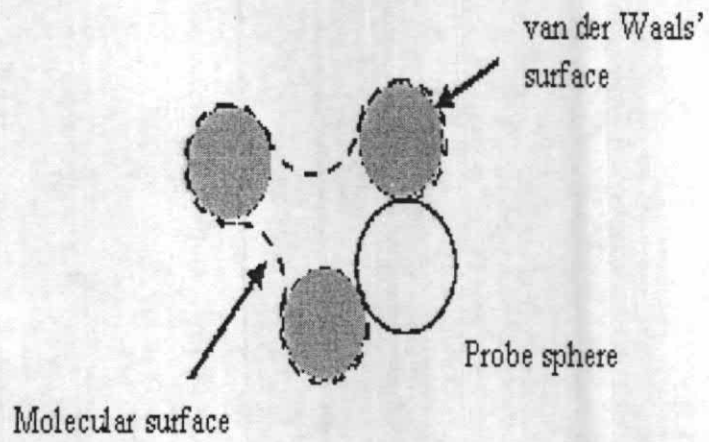

Figure 2.9 Molecular surface of a molecule, defined as the locus of the inward-facing probe sphere.

The definition of the solvent accessible surface is the location of the center of a sphere (representing the solvent molecule) as it rolls over the van der Waals surface of the protein (Figure 2.8). It is important to realize that this is different to the molecular surface, which is defined as the locus of the inward-facing probe sphere[83] (Figure 2.9)

2.2.1.4 Entropic Term (*TS*)

Entropy contributions arising from changes in the degree of freedom (translational, rotational, vibrational) of the solute molecules were included applying classical statistical thermodynamics[84]. There are three terms account for the entropy changes that are due to the transformation (S_{trans}), rotation (S_{rot}), and vibration (S_{vib}) calculating as Equation 2.61 to 2.63.

$$S_{trans} = R\left[\frac{3}{2}\ln\left(\frac{2\pi m}{h^2}\right)+\left(\frac{5}{2}\right)\ln(kT)-\ln\rho+\frac{5}{2}\right] \quad (2.61)$$

$$S_{rot} = R\left[\ln\left(\frac{8\pi^2}{\sigma}\right)+\frac{3}{2}\ln\left(\frac{2\pi kT}{h^2}\right)+\frac{1}{2}\ln(I_A I_B I_C+\frac{3}{2}\right] \quad (2.62)$$

$$S_{vib} = -R\sum_i \ln(1-e^{-h\nu_i/kT})+R\sum_i \frac{h\nu_i}{kT}\left(\frac{e^{-h\nu_i/kT}}{1-e^{-h\nu_i/kT}}\right) \quad (2.63)$$

The entropy of translational and rotational depend on known quantities only, namely the total mass m (ρ is the number density at 1 mol/L) and the principle moments of inertia ($I_A I_B I_C$) and σ is 1 for this case. The entropy of vibrational S_{vib} depends on the normal mode frequencies, ν_i. Therefore, ΔS obtains from (Equation 2.64):

$$\Delta S = (S_{trans}+S_{rot}+S_{vib})^C-(S_{trans}+S_{rot}+S_{vib})^P-(S_{trans}+S_{rot}+S_{vib})^L \quad (2.64)$$

where superscript C, P, and L represent complex, protein, and ligand, respectively.