A Fast and Accurate Analytical Method for the Computation of Solvent Effects in Molecular Simulations

Thesis by

Georgios Zamanakos

In Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

California Institute of Technology

Pasadena, California

2002

(Defended December 14, 2001)

Acknowledgments

Good science is not done in vacuum and this thesis is an example of this rule. This work would not have been possible without the input of many people. Caltech and the Materials Simulation Center are full of brilliant people and I have been privileged to work with some of them. First of all my advisor, Bill Goddard, has been instrumental in my research, not only for accepting me in his group and providing me with the funds and instruments to carry on my work, but also for his continuous energy and passion for science, which is rare for a person in his sixties. Vaidehi Nagarajan has been a mentor for me, providing guidance through my graduate studies and showing personal interest and care for my progress. Joseph Danzer has shown to me what it really means to be a good programmer and his self-motivation and excitement for research was contagious. Our collaboration in the development of the software for the geometric calculations was one of the most exciting parts of my research. Mario Blanco provided me with many useful suggestions in the development of the AVGB model and his work in solvation types was indispensable for the development of the full AVGB/SAS solvation model. Daniel Mainz and Ryan Martin helped me with learning the mechanics of MPSim. Gregg Caldwell, Wely Floriano and Prabal Maiti were the early adopters of the solvation software and their tolerance for using the code in its early stages is much appreciated. The code is in a good state now because of their patience and I hope that it will prove to be useful for many more people in the future. Last, but definitely not least, Darryl Willick and Ryan Martin have been exceptional system managers and they did a great job making sure all those computers and printers are working properly so people like me can do their job without interruptions.

Abstract

The solvent environment of molecules plays a very important role in their structure and function. In biological systems it is well known that water has profound effects in the functions of proteins. Simulations assist us in microscopic studies of chemical and biological phenomena. It is important then to include solvation effects accurately and efficiently in molecular simulations. In this work we present a novel approximate analytical method for calculating the solvation energy for every atom of a molecular system and the forces that act on each atom because of the solvent. The solvation energy is partitioned into long-range and short-range contributions. The long-range contributions are due to polar interactions between the solvent and the solute and the short-range are due to van der Waals and entropic effects. We show how the calculation of these effects, under certain approximations, can be reduced to the calculation of the volume and exposed area of each atom, assuming a fused-sphere model for the solute. We demonstrate a fast method for the exact, analytical calculation of the volume and area of each atom in the fused-sphere model and their gradients with respect to the atom's position. We incorporate the fast geometric algorithms into the approximate formulas we derived for the calculation of the solvation energy, to get our solvation model, the Analytical Volume Generalized Born - Solvent Accessible Surface (AVGB-SAS) model.

The predictions of the polar part of the method (AVGB) are very good as compared to numerical solutions of the underlying physical model, the Poisson-Boltzman equation, for small and large molecular systems. AVGB does not depend on any fitting

parameters, which is common in the literature for such approximate methods. It is very fast compared to numerical solutions of the PB equation or other Generalized Born methods. Also, the method is parallelizable which allows us to study much larger systems. The AVGB-SAS method has been implemented in a parallel molecular dynamics software package and a molecular docking software package. We have demonstrated the quality of the results of the AVGB-SAS model in the dynamics of DNA and in rational drug design applications.

Contents

Figures

Tables

1 Solvation in Molecular Simulations

1.1 Molecular Simulations

Simulations play a key role in the determination of various physical and chemical properties of molecular systems. In science, they are the link between experiment and theory either by validating and challenging new theories or by probing experimental results on the atomic scale. In industry, they can serve as a great cost-cutting tool, for example by pre-screening molecular targets for a certain property and thus reducing the amount of experimental work needed to identify the appropriate compounds. For that reason, a great amount of work has taken place in the field of molecular simulations and great progress has been achieved, due to both algorithmic improvements and increase in computational power.

The recent discovery of the sequence of the human genome [1] has opened the way to understanding, on the molecular scale, many human diseases including possible methods for prevention and treatment. For this goal to be achieved, it is imperative to understand various properties of macromolecules, such as structure, binding and processes. Molecular simulations can address the above questions in different ways. Homology modeling and energy minimization can be used to determine the structure of a protein in water. Molecular docking is used to screen and identify ligands, potential drugs, which form energetically favorable complexes with proteins in water. Molecular

dynamics can provide atomistic detail on biological and chemical reactions and processes.

There are many issues that have to be resolved in order to accurately simulate a molecular system. These issues can range from the structure and geometry of the system, the nature of the intermolecular and intramolecular forces, the level of accuracy (quantum-mechanical or classical), the parameters used such as charges and atomic radii, the effect of external factors such as temperature and pressure and various other elements that can affect the quality of the simulation [2]. Also, computational efficiency is of paramount importance since typical studies of chemical and biological systems require the computation of many (on the order of hundreds of thousands or more) consecutive calculation steps in order to achieve the accuracy needed. Thus, fast methods, parallel algorithms, and hardware improvements are another area of focus for molecular simulations.

1.2 Solvent Effects

The definition of solvent is "a substance that is liquid under the conditions of application, in which other substances can be dissolved and from which they can be recovered unchanged on removal of the solvent" [3]. Water in particular is the environment in which all biological processes take place. Biological macromolecules like proteins perform complex functions, such as transport of substances, binding of ligands and catalyzing chemical reactions, in water. The effect of the water environment on those processes is profound: the solvent influences electronic properties, nuclear distribution,

spectroscopic functions, acidity/basicity, reactive processes and molecular association [4]. It is crucial to understand and accurately calculate solvation effects in molecular simulations.

The solvation process is defined as "the process in which a particle of the solute is transferred from a fixed position in the gas phase into a fixed position in solution at constant temperature" [5]. The key parameter to describe the effects of the solvent is the free energy of solvation, ΔG_{sol}, and is defined as the reversible work spent in the transfer of the solute under the aforementioned conditions at equal number densities in the gas phase and in solution [4]. Microscopically, the solvation effect is due to intermolecular interactions between the solute and the solvent, as well as a change in the intramolecular interactions of the solute and a reorganization of the solvent because of the solute. In general, the calculation of the solvent effects is partitioned into three separate parts: electrostatics, short-range effects and cavitation (see [6] and references therein).

1.2.1 Electrostatic Effects

Electrostatic forces dominate the interactions of molecules due to their strength and long range. Electron distributions around nuclei in molecules create an electrostatic field that interacts with that of other nuclei. The charge distributions of the solute and solvent play a fundamental role in the solvation process. The polar contribution to the solvation energy, ΔG_{polar}, includes the work necessary to create the solute's gas-phase charge distribution in solution and the work required to polarize the solute charge distribution. The solute charge distribution polarizes the solvent, which in turn induces an

electric field on the solute. This is called the reaction field and it changes the self-energy of the solute atoms. Also, the intramolecular coulomb interactions of the solute are screened because of the presence of the solvent (Figure 1).

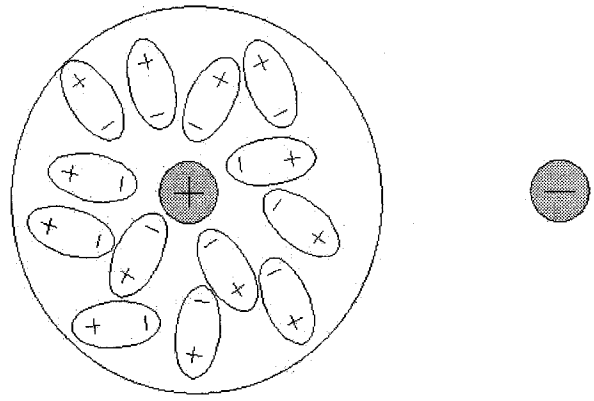

Figure 1. Reorganization of the solvent around a solute charge and dielectric screening of intramolecular interactions.

In addition, the presence of salt in the solvent affects the electrostatic energy of solvation and has a significant effect on conformational changes and binding. For example, DNA is known to go through a structural transition, from B to Z form as the salt concentration changes [7].

1.2.2 Short Range Effects

Besides the polar interaction, there is also the dispersion-repulsion (or van der Waals) interaction between the solvent and the solute that affects the solvation energy. These steric forces are of short-range nature and they are due to an effective dipole-dipole

interaction between solute and solvent [8]. Usually they are favorable to solvation since the dispersion forces are stronger than the repulsive forces around the solute cavity. Other contributions that take place are the hydrogen bonding between the solvent and the solute and charge transfer to or from the solute. This is particularly true for water. All these effects occur in a short range around the solute, the first solvation shell.

Cavitation, or "hydrophobic effect" is defined as the energetic cost of creating a cavity in the solvent for the solute to fit in. This term is entropic in nature. It accounts for the lowering of entropy due to the reorganization of water around non-polar solutes. For water in particular, it attributes the decrease in the number of ways that favorable hydrogen bonding can be achieved by solvent water because of the presence of a non-hydrogen bonding solute. Cavitation includes changes in solvent-solvent dispersion-repulsion due to the missing solvent in the cavity and changes in the local solvent structure. It is unfavorable to solvation because the entropy decreases.

1.2.3 Total Free Energy of Solvation

From the above, it is clear that the free energy of solvation has to take into account all sorts of effects: long range, short range and entropic. It is formally given by the formula:

$$\Delta G_{solv} = \Delta G_{polar} + \Delta G_{vdW} + \Delta G_{cav} \qquad (1)$$

Obviously, the different terms in equation (1) will contribute in different ways for various combinations of solute and solvent. For example, for a polar solvent like water the electrostatic term will dominate and the short-range steric interactions will be

moderate. On the other hand, for non-polar solvents the cavitation penalty and the electrostatic terms should be smaller and the steric interaction should dominate due to weaker interactions among the solvent molecules. For polar solutes in polar solvents the electrostatic term should dominate, whereas for non-polar solutes in non-polar solvents the steric interactions should dominate.

Properly treating the solvation effect in simulations of biological systems is critical to obtaining accurate information. Because the effects of solvation are so complex, a number of assumptions and approximations need to be made in order to make such simulations computationally tractable. The key assumption is that we can partition the solvation energy into the different contributions, the short-range van der Waals and entropic effects and the long-range polar effects. Various methods that exist to calculate these contributions will be presented in the following.

1.3 Solvation Models

The existing models for the calculation of solvation in molecular simulations can be separated into two classes: explicit and implicit. The most obvious way of taking account of the solvent is by explicitly including solvent molecules in the simulation. This method has many drawbacks, the first of which is computational efficiency. Every atom that is explicitly included in a simulation adds 3 degrees of freedom. For 200 water molecules we add 1800 degrees of freedom, whereas for 200 1-octanol molecules we add 16200 degrees of freedom. In order to measure structural and dynamical properties of the system, we'll need first to equilibrate the system and then average over those additional

degrees of freedom. This implies that we need to perform the simulation for a larger number of steps where each additional step costs more CPU time. One might think that this additional cost comes at the benefit of a more accurate simulation, but this is not necessarily the case. Explicit solvent models are only as good as the simulation method and parameters used. Solute electronic polarization is usually not taken into account and the electrostatic interactions between solvent and solute are dependent on the charges of the forcefield parameter set used.

For the above reasons, the focus of most research has been on implicit solvation models. In such models, the solvent is implicitly included by assuming it is a continuous medium surrounding the solute. That way the effect of the solvent on the solute is already averaged and the solute is always in statistical equilibrium. The challenge then is to describe the effects of the solvent accurately under the continuum approximation. In the following we will present different ways that deal with this problem.

1.3.1 Surface Tension Models

Surface tension models were first introduced by Eisenberg in 1986 [9], [10]. In such models, the free energy of solvation is given as a product of the solvent accessible surface area and an empirically determined surface tension parameter, for each atom:

$$\Delta G_{solv} = \sum_{i=1}^{N} \sigma_i A_i \tag{2}$$

for a molecule of N atoms, where A_i is the solvent accessible surface area and σ_i is the surface tension of atom i.The solvent accessible surface area (SASA) is defined as the

surface traced by the center of a sphere of certain probe radius, as it rolls over a fused-sphere model of the solute [11]. An example of the SAS of 3 carbon atoms and the probe sphere that traces it is shown in Figure 2. Treating the solvent molecules as spheres is reasonable for molecules of spherical symmetry. The appropriate probe radius for solvents with different properties is addressed in [12]. For example, in a non-polar solvent like hexadecane, we expect the solvent effects to take place in the first solvation shell since dispersion interactions should dominate. Thus, for hexadecane a solvent probe radius of approximately 1.5 Å is more appropriate than a much larger radius that would result if one takes into account the size and shape of the solvent molecule.

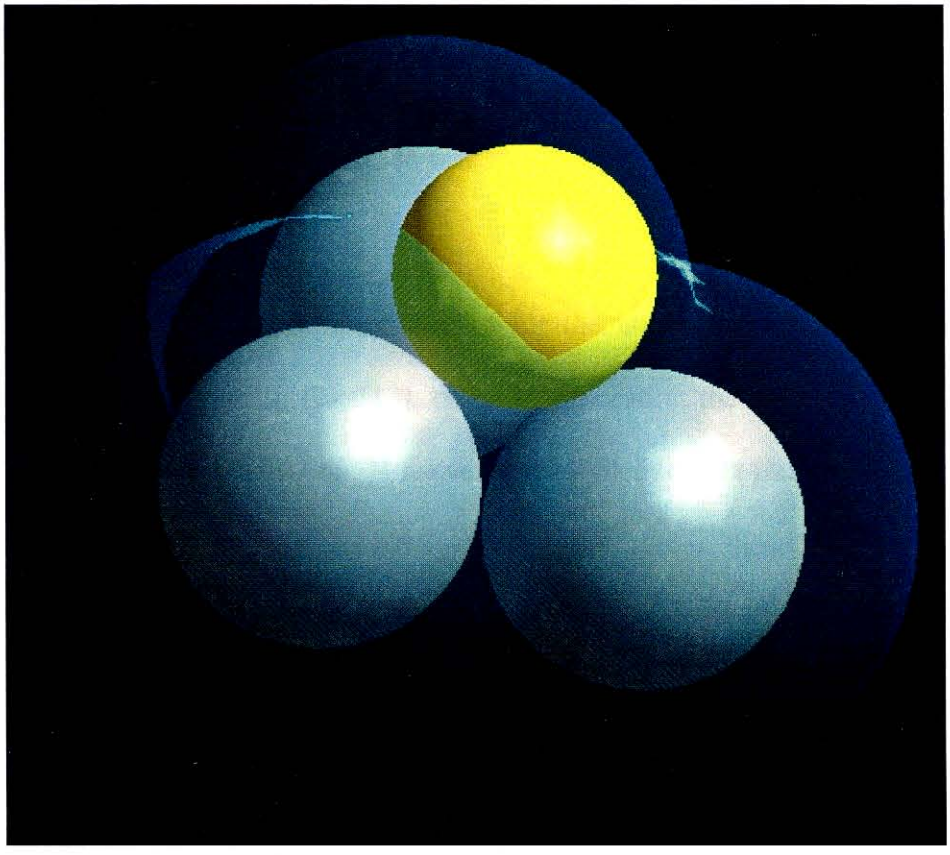

Figure 2. The Solvent Accessible Surface (transparent) of 3 carbon atoms with radius 1.7 Å (gray), as is traced by a probe of radius 1.4 Å (yellow.)

Surface tension models are conceptually simple and thus not very reliable. Their shortcomings are as follows:

- Accurate and efficient calculation of the SASA and its gradient with respect to atomic coordinates is necessary in order for this model to be practically useful in molecular simulations.
- The surface tension parameters are empirically obtained from a molecule data set. It is obvious that the accuracy of the model is heavily dependent on the training set used and the extension of it onto molecules out of that set is questionable.
- Since the model is purely based on the exposed surface, only atoms on the surface of the solute feel the effect of the solvent. It is thus not capable of calculating long-range effects such as dielectric screening which in polar solvents dominate the solvation process.

There have been attempts to correct for the above deficiencies, by modifying slightly the form of equation (2), [13], or by incorporating the occupied volume of the solute [14], [15] in the calculation. However, the problem of not treating the electrostatic contribution with any solid theoretical foundation remains in those theories, and that is where we will shift our focus in the following.

1.3.2 Continuum Dielectric Model: The Poisson-Boltzman Equation

Electrostatic interactions in macromolecules have been studied extensively due to their profound effects on the macromolecules' functions [16], [17]. Most important

biological phenomena involve changes in the interaction of various groups with the surrounding water environment and since polar effects are dominating those interactions, methods for the calculation of those effects have proliferated [18]. Continuum dielectric methods treat the solvent as a dielectric continuum surrounding the solute molecule. The calculation space is characterized by the dielectric permittivity $\varepsilon(\vec{r})$, and the solvent is separated from the solute cavity by a boundary surface. The solute is treated as a charge distribution, $\rho(\vec{r})$ (which could also be a set of point charges). The electrostatic potential, $\Phi(\vec{r})$, at every point in space would describe the interaction between the solute charge distribution and the solvent dielectric. It is clear, then, that in this model and in the absence of salt, the potential $\Phi(\vec{r})$ would be given by the Poisson equation:

$$-\nabla\cdot\left(\varepsilon(\vec{r})\nabla\Phi(\vec{r})\right)=4\pi\rho(\vec{r}) \tag{3}$$

In the presence of salt, we can employ the Debye- Huckel theory [19] to incorporate salt effects. In this theory, we assume that the ratio of the concentration of ion type i around the solute to its concentration far away from the solute is given by the Boltzman distribution, $\exp(-W_i(\vec{r})/k_BT)$, where k_B is Boltzman's constant, T the absolute temperature and $W_i(\vec{r})$ the work required to move the ion of type i from infinity (where $\Phi(\infty)=0$) to the point $\vec{r}$. We assume that we have only two types of ions, negative and positive (such that the total system is electrically neutral). Then, if e_c is the absolute charge of one electron, we must have for each ionic species:

$$W_1(\vec{r})=+e_c\Phi(\vec{r}) \qquad W_2(\vec{r})=-e_c\Phi(\vec{r}) \tag{4}$$

and if we assume that the concentration of each species M_+, M_- at infinity is M, the Boltzman distribution law gives

$$M_+ = M\exp(-e_c\Phi(\vec{r})/k_BT) \qquad M_- = M\exp(+e_c\Phi(\vec{r})/k_BT) \tag{5}$$

and, thus, the charge density of ions around the solute should be

$$\rho_{ion}(\vec{r}) = e_c(M_+ - M_-) = -2Me_c\sinh\left(\frac{e_c\Phi(\vec{r})}{k_BT}\right) \tag{6}$$

By applying equation (6) on (3) and using the ionic strength, I, instead of the salt concentration M, we get (since $I = 0.5\sum_{i=1}^{N} c_i z_i^2 = 1000M/N_A$, where N_A is Avogadro's number, c_i is the molar concentration and z_i the charge in electrons of ion species i)

$$-\nabla\cdot\left(\varepsilon(\vec{r})\nabla\Phi(\vec{r})\right) + \kappa(\vec{r})^2\sinh\left(\frac{e_c\Phi(\vec{r})}{k_BT}\right) = 4\pi\rho(\vec{r}) \tag{7}$$

where the constant κ is called the Debye-Huckel screening parameter, which is zero inside the solute and given by the formula

$$\kappa(\vec{r}) = \sqrt{\frac{8\pi N_A e_c^2 I}{1000\,\varepsilon(\vec{r})k_BT}} \tag{8}$$

outside of the solute. If we assume a constant dielectric permittivity value, ε_{out}, for the region outside of the solute, then the Debye-Huckel parameter is also a constant. Equation (7) is called the Poisson-Boltzman equation (PB) and it is a nonlinear, elliptic, second order, partial differential equation. In cases of low ionic strength, one could use the first term of a Taylor expansion of the exponential sign term to get the linearized PB equation. It is a very difficult equation to solve for arbitrary systems, but it describes very

accurately the polar effects of the solvent on the solute in the continuum dielectric approximation.

The most common assumption for solving the PB equation is that the dielectric permittivity takes two values, ε_{in} in the solute cavity and ε_{out} outside. For water environment, ε_{out} is 78.2. The electrostatic free energy of solvation ΔG_{polar} can be obtained by solving this equation twice, once with the solute inside the solvent dielectric and once with the solute in vacuum ($\varepsilon_{out} = 1$). The polar solvation energy, assuming that the solute charge density is a set of N point charges q_i at positions $\vec{r}_i$, is

$$\Delta G_{polar} = \frac{1}{2}\sum_{i=1}^{N} q_i \left(\Phi^{solv}(\vec{r}_i) - \Phi^{vac}(\vec{r}_i)\right) \tag{9}$$

Exact analytical solutions of the PB or in the absence of salt, the Poisson equation, are not possible except for very few simple cases. In order to get analytical solutions, we must make crude simplifications in the shape of the solute. For example, small molecules and globular proteins are treated as spherical cavities, whereas DNA is modeled as a charged cylinder. Kirkwood [20] introduced an analytical solution for equation (3) for a set of point charges inside a spherical cavity and Jayaram [21] has given the analytical solution for the intermolecular problem of two ions embedded in a dielectric continuum. Although analytical, these solutions are of little value for any practical purpose because the simplifications needed in the shape of the solute limit severely the applicability of the model to realistic systems. For that reason, numerical solutions of the PB and Poisson equations have been developed instead.

A number of numerical methods have been developed for the numerical solution of the PB equation either using finite differences or boundary element methods. The best known in the literature are the DelPhi program [22], UHBD [23] and PBF [24]. A major problem with numerical methods is that they do not calculate the solvation electrostatic forces along with the energies. These forces can be computed but only at a great computational expense since they would have to be calculated from numerical differences. This makes numerical methods useless for molecular mechanics simulations. At the same time, since a spatial grid is used in order to solve the PB equation, the accuracy of the solution and the CPU time needed for its calculation are highly dependent on the density of the grid. Too sparse a grid would result in fast calculations with inaccurate solutions. Too fine a grid would result in accurate solutions but slow calculations. For example, DelPhi would take about 25 minutes on a 195MHz SGI processor to solve problems on a 185x185x185 grid for a system of 600 atoms. Also, since the solution is based on a grid, the algorithm used will scale with the size of the solute as $O(N^3)$, making it impractical for studying very large systems. Finally, parallelization of such algorithms has met with limited success.

Regardless of their computational efficiency shortcomings, numerical solutions of the PB equation have been successful at different applications [18]. This is a proof that the dielectric continuum approximation, despite its conceptual simplicity (the PB equation ignores the molecular nature of the solvent, the finite size of the ions and ion-ion correlation effects), is a valid approximation. In order to get an accurate description of the

polar solvation effects in molecular systems, but with a certain computational efficiency, approximate analytical solutions to equations (3) and (7) have to be found. The goal is to find solutions that correlate well to numerical solutions of the PB equation qualitatively and quantitatively, with analytical formulas that are derived using approximations that capture the physics of the PB equations. Such theories will be described in section 1.3.3.

1.3.3 Approximate Solutions for the Continuum Dielectric Model

- Multipole Expansions

In the multipole expansion approach the electrostatic potential is determined by assuming very simple shapes for the solute cavity and using limited multipole expansions to represent the solute charge distribution. The electrostatic potential can be written as a series of spherical harmonic terms. This method was first introduced by Kirkwood [20] as an analytical solution to the Poisson equation but has since then been extended for more complex cavity shapes [25]. Although faster than the numerical solution of the PB equation, it still is a slow method since the series must converge for the results to be valuable, which means many terms have to be included. There have been attempts for a faster calculation, [26], [27], but the inherent problems of inaccurate description of the molecular cavity and the need to truncate the series at some point limit the suitability of these methods to simple systems or qualitative studies.

- Distance Dependent Dielectric (DDD) methods

As was already described above, the solvent molecules surrounding the solute are polarized due to the solute charge distribution. This generates a reaction field, which in

turn polarizes the solute. The intramolecular coulombic interactions are screened because of the surrounding solvent molecules (Figure 1). This effect of dielectric screening on the polar energy of two atoms, E_{pol}^{ij}, can be represented quantitatively by the dielectric permittivity ε:

$$E_{pol}^{ij} = \frac{q_i q_j}{\varepsilon\, r_{ij}} \tag{10}$$

However, this formula cannot be accurate for small distances r_{ij} since when two atoms are close together there is not enough space for the solvent to screen the interaction. For large distances though, we expect that there will be enough solvent and the screening will be significant. This motivates us to assume that the dielectric permittivity should be dependent on the distance by a sigmoid profile (Figure 3).

Such sigmoid profile can be described mathematically by an equation of the form:

$$\varepsilon(r_{ij}) = A + \frac{B}{1 + k\exp(-\lambda B r_{ij})} \tag{11}$$

In practice, one would calibrate the results obtained from such model by assigning different parameters in (11), according to the type of atoms involved in the interaction, [28], [29]. Comparisons with experimental solvation energies or numerical solutions of the PB equation will determine the exact values of the parameters.

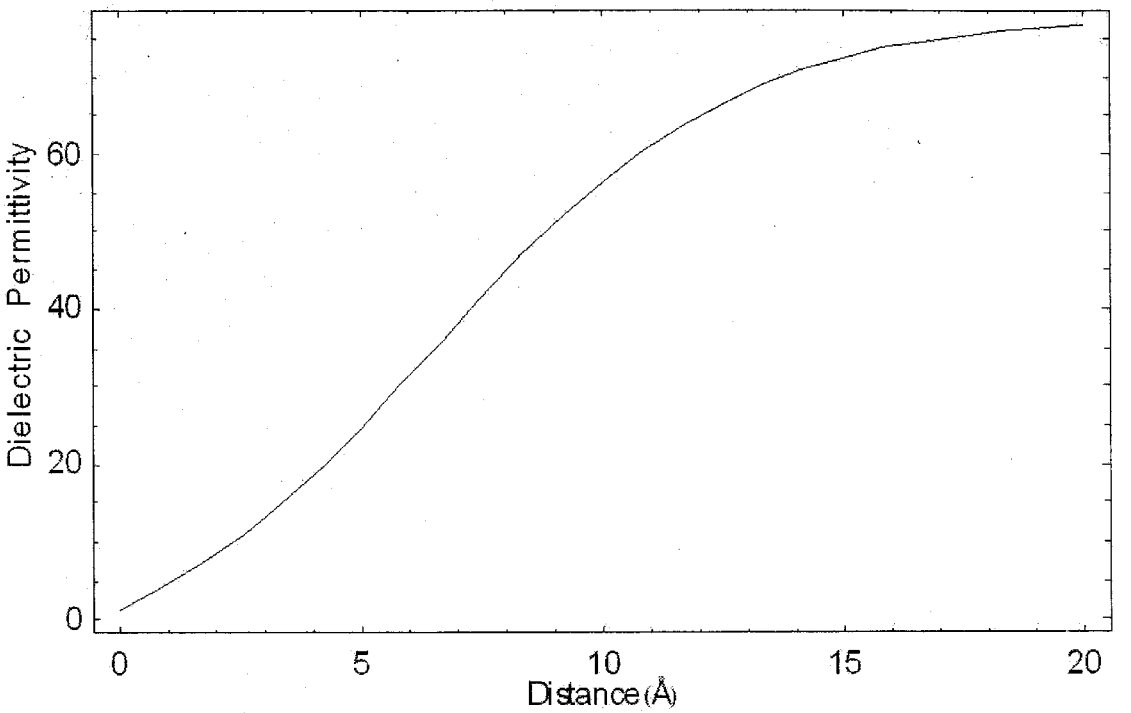

Figure 3. Sigmoid permittivity profile for the distance dependent dielectric model.

Other formulas more or less complex have been proposed; however, all these models are ad-hoc in nature. Sigmoid permittivity profiles are predicted by the Lorentz-Debye-Sack (LDS) theory of polar solvation [30], with reaction field corrections included [31]. Thus, although qualitatively equation (11) should be able to describe dielectric screening, there is little formal justification for it. On the other hand, studies on the intermolecular screening of the polar energy due to the solvent have been done with the PB equation [32]. It is shown there that the dielectric screening only qualitatively is described by a sigmoid behavior. The phenomenon is just too complex to be described by such a simple formula (Figure 4). This method is also heavily dependent on the molecule set used to train the parameters that describe the exact form of the sigmoid permittivity, for each atom type. This would make the extension of the method in different systems

questionable. Nevertheless, DDD models are still extensively used because they are easy to implement and computationally very efficient. DDD models, however, should always be used with caution in applications, since comparisons with other solvation electrostatic models have shown that the results from these models can be qualitatively erroneous [33], [34].

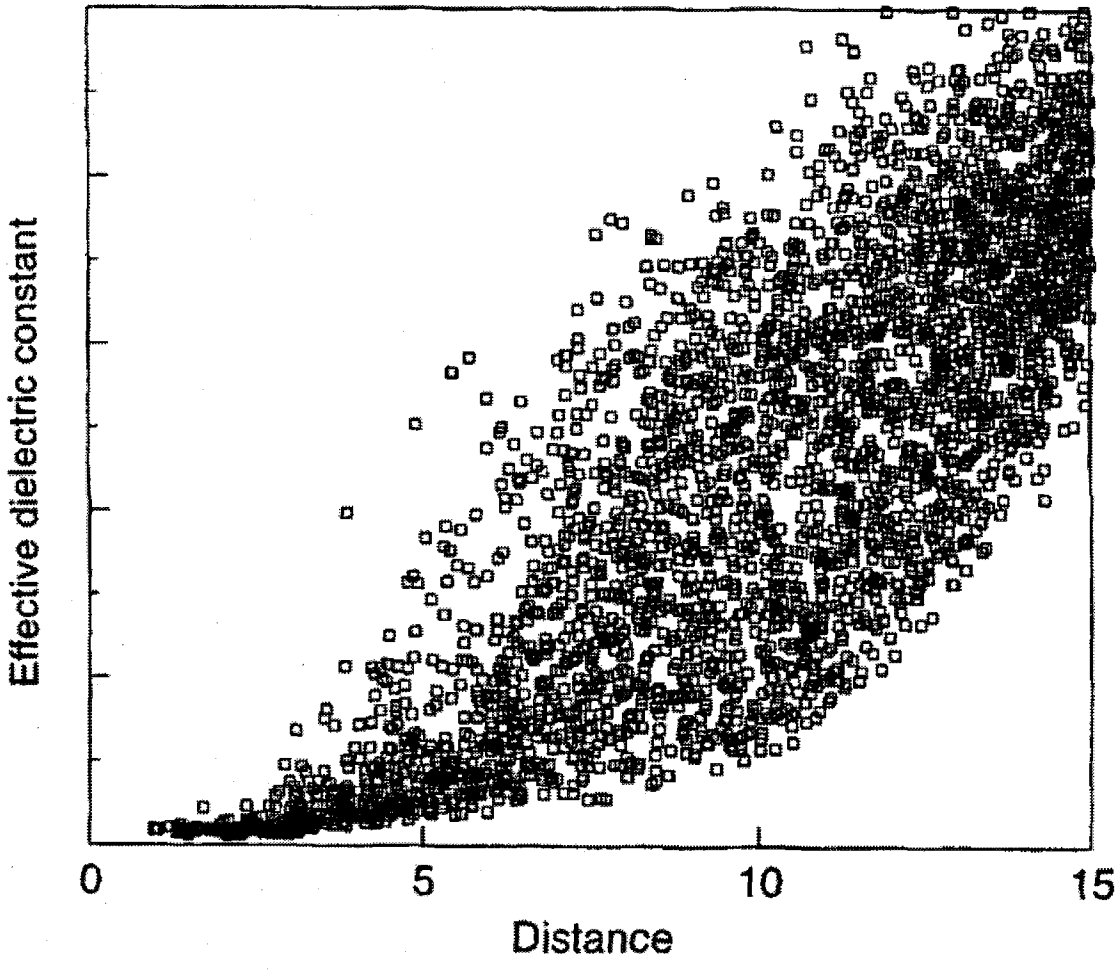

Figure 4. Effective dielectric permittivity for protein A calculated using equation (10), where the pair energies where calculated from numerical solutions of the PB equation [22]. The sigmoid profile is only qualitatively described by an equation of the form (11). The phenomenon is more complex. (Figure from reference [32].)

- Generalized Born Model

In the Generalized Born (GB) model the goal is to find an expression for the polar solvation energy of the form

$$\Delta G_{polar} = -\frac{1}{2}\left(\frac{1}{\varepsilon_{in}} - \frac{1}{\varepsilon_{out}}\right)\sum_{i=1}^{N}\sum_{j=1}^{N} q_i q_j \gamma_{ij} \tag{12}$$

where γ_{ij} is an ad-hoc function that somehow describes the effects of polarization and dielectric screening. The GB model has been proven to be very successful in predicting polar solvation energies and has received considerable attention in the literature [33], [34], [35]. It comes in different variations that differ in the functional form of γ_{ij}. A certain variation of the GB model is employed in this work and the theoretical foundations and approximations behind the GB theory will be discussed in detail in chapter 2.

- Other methods

An issue that arises with the application of the PB equation on the solvation electrostatics is the value of the dielectric permittivity inside the solute, ε_{in}. A value of 1 is appropriate only for small molecules but is probably not right for macromolecules. Electronic polarization and field-induced nuclear reorientation effects affect the value of ε_{in}. Typically, values that range from 2 to 8 have been used, but even then the assumption that the solute dielectric permittivity should be isotropic is questionable. In general, the definition of the dielectric permittivity in proteins can be ambiguous [17].

For that reason, the Langevin Dipoles (LD) model was developed [36] in order to avoid the use of ε_{in}. In this model, the solute is placed in the center of a cubic grid. Langevin dipoles are placed on the grid points and the polarization of the solvent with the solute is accounted for by reorienting the solvent dipoles, which generates the reaction field. The shortcomings of this model are mainly due to the assumption that the electrostatic potential is represented by a dipole term only; the accuracy and speed depend on the resolution of the grid used and the results may not be rotationally invariant. However, it is an interesting idea and an alternative to the continuum dielectric methods.

Another method is the conducting screening model (COSMO) that assumes that the surrounding medium is well modeled as a conductor [37]. The dielectric behavior is derived using analytical formulas that allow for the calculation of gradients, which are necessary for molecular dynamics simulations.

1.3.4 Including First Solvation Shell Effects

Due to the biological importance of the effect of water on macromolecules, the focus of the calculation of solvation effects in the literature has been on the calculation of polar effects. This is because water is a highly polar solvent and proteins have polar groups. In such situations the polar effect dominates dispersion-repulsion and cavitation effects. Short-range effects, though, can play a significant role in non-polar solvents and should be included in order to have a complete solvation model. The nature of van der Waals and cavity effects was discussed in section 1.2.2.

The most obvious way to include these effects in solvation calculations would be by explicitly including solvent molecules in a short range around the solute. However this would lead to the same problems of the explicit solvent calculations, namely the averaging in time of many more degrees of freedom and the lack of polarization effects between solute-solvent. Instead, the surface tension models (1.3.1) are an attractive alternative. In this model, the free energy of solvation is determined by equation (2) and a set of surface tension parameters that are empirically determined. Although this model cannot describe accurately long-range polar effects, it should be able to reproduce short-range effects, like dispersion-repulsion and entropic effects like cavitation. Thus, the first solvation shell effects will be described by the formula:

$$\Delta G_{vdW} + \Delta G_{cav} = \sum_{i=1}^{N} \sigma_i A_i \tag{13}$$

where A_i is the solvent accessible surface area [11] and σ_i the surface tension parameter for atom i of the solute.

The justification behind this approximation is that the magnitude of the free energy of solvation due to first solvation shell effects can be considered proportional to the number of solvent molecules in the first solvation shell. One expects that entropic terms like cavitation, along with the averaging of short-range steric interactions between solute and solvent should be a function of the geometry of the solute and correlate statistically with the exposed area. The exposed area can be thought of as a non-integer average (ensemble or time average) number of solvent molecules in the first solvation shell. The assumption in equation (13) is that the energy is proportional to the solvent accessible surface area (SASA), as defined by Richards [11] and weighted by parameters

that are empirically determined. The quality of the fitting to experimental results and its predictive ability will be the ultimate judge of the success of the model.

The fact that surface tension models, despite their simplicity, have yielded qualitatively accurate results prompts us to accept their validity for short-range effects. Thus, a model that has an accurate method for predicting electrostatic contributions to the solvation energy accompanied by a surface tension model for including first solvation shell effects should be capable of predicting solvation energies for a multitude of solvents and solutes. Of course, the quality of the first solvation shell contribution will depend on the surface tension parameters used.

2 The Generalized Born (GB) Model

2.1 The Born Model

The electrostatic energy G_{pol} of a charged dielectric sphere of radius R and charge q, embedded in a dielectric of permittivity ε, can be easily calculated from Gauss's law $\left(\int \vec{E} \cdot d\vec{s} = 4\pi q\right)$ by use of spherical symmetry and the relation between the electrostatic energy and the electric field $\vec{E}$, $G_{pol} = \frac{1}{4\pi}\int E^2 dv$. It is given by the formula:

$$G_{pol} = \frac{q^2}{2\varepsilon R} \tag{14}$$

Thus, if we assume that the sphere has interior dielectric of 1, and it is reversibly transferred from the vacuum to a medium of dielectric permittivity ε, the change in the electrostatic free energy change should be

$$\Delta G_{pol} = -\frac{1}{2}\left(1 - \frac{1}{\varepsilon}\right)\frac{q^2}{R} \tag{15}$$

In general, if the interior dielectric of the sphere is ε_{in} and the surrounding medium has permittivity ε_{out}, then the electrostatic free energy change is

$$\Delta G_{pol} = -\frac{1}{2}\left(\frac{1}{\varepsilon_{in}} - \frac{1}{\varepsilon_{out}}\right)\frac{q^2}{R} \tag{16}$$

If we assume now that the sphere is actually an ion and the exterior dielectric is the solvent environment, we can see that equation (16) actually describes the solvation

energy of an ion of radius R. This model was first introduced by Born [38] and has been used successfully for the calculation of the solvation effects on ions [39], [40] and ion pairs [41].

The success of the Born model on ions has given the impetus to generalize the Born equation (16) and create an analytical, approximate model for the description of electrostatic solvation effects on multi-atom systems such as macromolecules, with arbitrary size and shape. In the following, the formalism of the GB model will be derived and the validity of the approximations made will be discussed.

2.2 The Generalized Born Approximation

We will now try to generalize the Born equation (16) for a system of N atoms. As in section 2.1, we assume that every atom i is a conducting sphere of radius α_i and charge q_i. If the spheres are at a very large distance away from each other, then it is a safe approximation that the interaction energy between every atom will follow Coulomb's law. This is because the spheres are very far away and thus the finite size of them has no effect on the energy. Effectively, the spheres interact as point charges, as long as the separation distances are much larger than the spheres' radius. At the same time we have to include the electrostatic self-energy of solvation for every atom, which is given by equation (16). Thus, for the system of N atoms very far away from each other, the solvation energy is

$$\Delta G_{pol} = -\frac{1}{2}\left(\frac{1}{\varepsilon_{in}} - \frac{1}{\varepsilon_{out}}\right)\left[\sum_{i=1}^{N}\frac{q_i^2}{\alpha_i} + \sum_{i\neq j=1}^{N}\sum_{j=1}^{N}\frac{q_i q_j}{r_{ij}}\right] \tag{17}$$

where r_{ij} is the distance between atoms i and j. We would like to generalize this formula to be applicable to molecular systems of arbitrary shape and size. For this, we have to correct equation (17) for when the spheres get closer to and possibly intersect each other. We seek for an analytical formula that has the form of equation (12) or, rewritten to resemble more of coulomb's law,

$$\Delta G_{pol} = -\frac{1}{2}\left(\frac{1}{\varepsilon_{in}} - \frac{1}{\varepsilon_{out}}\right)\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{q_i q_j}{f_{ij}} \tag{18}$$

The functional form of f_{ij} can be determined only by an ad-hoc way, as long as it makes physical sense and satisfies appropriate boundary conditions. The most common form used is [42]:

$$f_{ij} = \sqrt{r_{ij}^2 + \alpha_i \alpha_j \exp\left(-r_{ij}^2 / 4\alpha_i \alpha_j\right)} \tag{19}$$

The parameters α_i and α_j are called the Born radii and they represent an effective radius for the respective atoms. This functional form is chosen because it reproduces the right solvation energies at the two limits:

- when there is only one atom, $N = 1$ and $r_{ij} = 0$, then $f_{ii} = \alpha_i$, which means that the solvation polarization energy is

$$\Delta G_{pol} = -\frac{1}{2}\left(\frac{1}{\varepsilon_{in}} - \frac{1}{\varepsilon_{out}}\right)\frac{q_i^2}{\alpha_i} \tag{20}$$

  which is what is expected from the Born model.

- At very large interatomic distances compared to the radii, $r_{ij} \to \infty$, the exponential in equation (19) falls fast to zero, and $f_{ij} \to r_{ij}$. Then, the intramolecular solvation energy between atoms i, j, becomes

$$\Delta G^{ij}{}_{pol} \to -\frac{1}{2}\left(\frac{1}{\varepsilon_{in}} - \frac{1}{\varepsilon_{out}}\right)\frac{q_i q_j}{r_{ij}} \tag{21}$$

which is, as expected, Coulomb's law.

Equation (19) is basically an interpolation formula between the Born and Coulomb limits, as is shown in Figure 5.

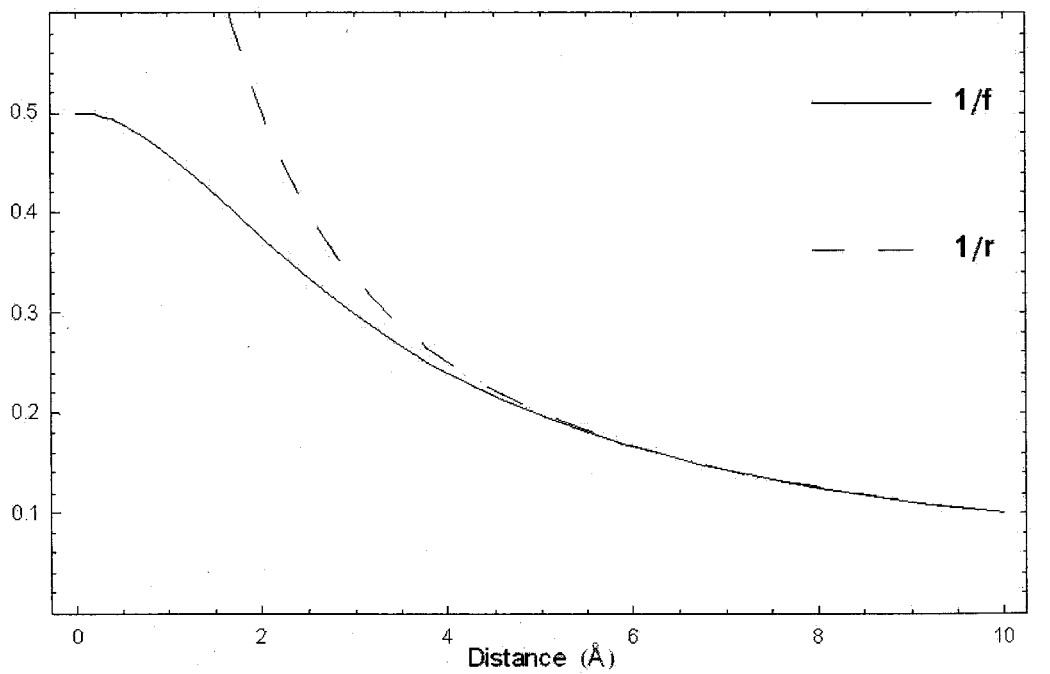

Figure 5. The functional form of equation $1/f$ (from equation (19)) with Born radii $\alpha_i = \alpha_j = 2$, compared to the coulombic behavior $1/r$.

The key parameters in this model are the Born radii α_i. The physical meaning of the Born radii becomes obvious when one sets all charges $q_i = 0$ except for atom k. Then, the solvation energy of the system becomes

$$\Delta G_{pol} = -\frac{1}{2}\left(\frac{1}{\varepsilon_{in}} - \frac{1}{\varepsilon_{out}}\right)\frac{q_k^2}{\alpha_k} \tag{22}$$

For a molecule that only atom k is charged, the solvation energy is effectively the self-energy of polarization for atom k. From equation (16) we can interpret the Born radius parameter α_k as the effective radius of an ion of charge q_k, whose solvation energy is equal to the self-energy of polarization of atom k in the molecule. This means that in order to use the GB model we will have to already know the self-energy of polarization, since that is the only way to know the values of the Born radii. Obviously this procedure would be of no practical use since we would need to know the answer in order to solve the problem. We will have to introduce some approximations in order to predict the values of the Born radii, and these will be discussed in section 2.3.

Nevertheless, it is instructive to examine the accuracy of the approximations made already at this point. The interpolation formula (19) along with equation (18) provides an analytical formula to get the solvation energy for an arbitrary molecular system. Although we do not have a formal way to calculate the Born radii yet, we can still use numerical solutions of the PB equation in order to predict these parameters. Specifically, for a molecule of N atoms, we do N numerical calculations of the solvation energy where each time all charges are set to zero except for one atom, i. The numerical answer can be plugged into equation (22) and thus calculate the Born radius α_i. Then, we repeat

the numerical solution for the real, fully charged molecule and compare the solvation energy to the predicted energy from the GB model with the numerically derived Born radii. We performed these calculations on a set of 376 small molecules, not larger than 40 atoms each (see reference [43] and the Appendix for the list of molecules). The numerical calculations were performed using the PBF solvation code [24] and the results are shown in Figure 6.

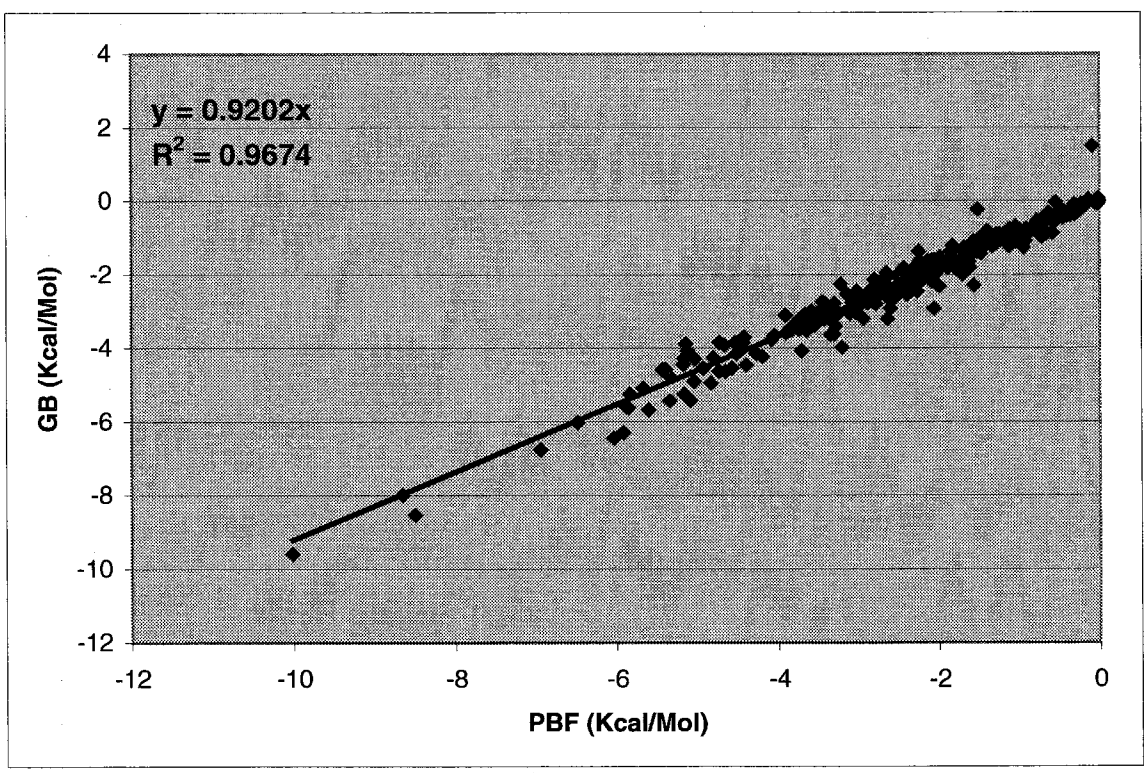

Figure 6. Comparison between numerical solutions of the PB equation and the predictions of the GB model with PBF-derived Born radii, for 376 small molecules. Linear regression fit and correlation coefficient shown.

The results show excellent correlation (correlation coefficient 0.97) between the GB model and the numerical solutions of the PB equation. The linear regression fit shows

that the results from the two methods are very close to each other although there is a small systematic error; the relation is not exactly of the form $y = x$, but instead the slope is 0.92. This is proof that as long as we have an accurate description of the Born radii parameters for the system, the interpolation formula (19) in the GB model (18) describes remarkably well electrostatic solvation energies, at least for small molecules.

Despite its success, there have been attempts to modify equation (19). Different formulas have been proposed that satisfy the same limiting conditions already discussed, but perform better for specific applications [44]. However, there is no systematic way to introduce an interpolation formula. It is always an approximation that is put to the test by direct comparisons with numerical solutions to the PB equation.

2.3 Born Radii and the Coulombic Approximation

In order to calculate the Born radii for the GB model, we need to come up with an analytical approximate solution to the polarization self-energy of an atom in the molecule. Such a solution was first given in [45] by use of the electrostatic energy density. Instead, we will present a novel, formal proof, inspired by [35], that clearly shows the physical meaning of the assumptions made in this calculation.

As was already described in 1.2.1, the electrostatic self-energy of solvation of an atom is due to the interaction of the solute charge distribution, $\rho(\vec{r})$ with the induced dipoles of the solvent. This is called the reaction field $\Phi_{reac}(\vec{r})$ and it is responsible for polarizing

the solute atoms. The polarization is described by the induced surface charge, $\sigma_{pol}(\vec{r})$, on the surface of the solute atoms. Then according to electrostatic theory [46], the reaction field is given by

$$\Phi_{reac}(\vec{r}) = \oint_S \frac{\sigma_{pol}(\vec{R})}{|\vec{r} - \vec{R}|} d^2R \tag{23}$$

where S is the solvent accessible surface of the solute. The polar free energy of solvation is a functional of the reaction field:

$$\Delta G_{pol} = \frac{1}{2}\int \rho(\vec{r})\Phi_{reac}(\vec{r}) d^3r \tag{24}$$

Assuming that the solute charge distribution is a set of N point charges q_k located at points $\vec{r}_k$, $\rho(\vec{r}) = \sum_{k=1}^{N} q_k \delta(\vec{r} - \vec{r}_k)$, the polar free energy of solvation becomes

$$\Delta G_{pol} = \frac{1}{2}\sum_{k=1}^{N} q_k \oint_S \frac{\sigma_{pol}(\vec{r})}{|\vec{r} - \vec{r}_k|} d^2r \tag{25}$$

By applying Gauss's law on an infinitesimal pillbox of surface ΔS on the boundary surface that separates the two dielectrics, ε_{in} and ε_{out}, we can calculate the discontinuity of the reaction field on the surface of the solute:

$$\oint_S \vec{E} \cdot d\vec{s} = 4\pi q$$

or,

$$\left(\vec{E}_{out} - \vec{E}_{in}\right) \cdot \hat{n} = 4\pi\sigma_{pol}\Delta S \tag{26}$$

where $\hat{n}$ is the normal to the boundary surface and $\vec{E}$ is the electric field due to the local polarization charge density. According to the boundary condition for the dielectric

displacement $\vec{D}(\vec{r})$ on the interface between the two dielectrics, the normal component of the dielectric displacement has a discontinuity that is proportional to the bare surface charge density σ (which does not include the polarization charge) [46].

$$\left(\vec{D}_{out} - \vec{D}_{in}\right) \cdot \hat{n} = 4\pi\sigma \tag{27}$$

But the bare charge density σ is zero since we assume point charges and the dielectric displacement is proportional to the dielectric constant, $\vec{D} = \varepsilon \vec{E}$. Thus, from equations (26) and (27) we have an expression for the polarization charge density $\sigma_{pol}(\vec{r})$ as a function of the normal component of the electrostatic field on the surface of the molecule, $\vec{E}_{in}(\vec{r}) \cdot \hat{n}$:

$$\sigma_{pol}(\vec{r}) = \frac{1}{4\pi}\left(\frac{\varepsilon_{in}}{\varepsilon_{out}} - 1\right)\vec{E}_{in}(\vec{r}) \cdot \hat{n} \tag{28}$$

By plugging equation (28) into equation (25) and setting all charges equal to zero except for atom k, which is located at position $\vec{r}_k$, we get an expression for the self-energy of polarization for atom k, $\Delta G_{pol,k}$, in the molecular cavity:

$$\Delta G_{pol,k} = \frac{1}{8\pi} q_k \left(\frac{\varepsilon_{in}}{\varepsilon_{out}} - 1\right) \oint_S \frac{\vec{E}_{in}(\vec{r}).\hat{n}}{|\vec{r} - \vec{r}_k|} d^2 r \tag{29}$$

Equation (29) is exact, but not very useful since the functional form of the electric field is not known. We will need to introduce an approximation for the electric field in order to get a formula for the Born radii. We know from Gauss's law that for a point charge q_k located at point $\vec{r}_k$ in a spherical cavity with dielectric constant ε_{in}, the electric field is

$$\vec{E}_{in}(\vec{r}) = \frac{1}{\varepsilon_{in}} q_k \frac{\vec{r} - \vec{r}_k}{\left|\vec{r} - \vec{r}_k\right|^3} \tag{30}$$

which is Coulomb's law. If we use equation (30) in equation (29), we can have an expression for the self-energy of polarization for atom k, only in terms of the geometry of the system:

$$\Delta G_{pol,k} = -\frac{1}{8\pi}\left(\frac{1}{\varepsilon_{in}} - \frac{1}{\varepsilon_{out}}\right) q_k^2 \oint_S \frac{(\vec{r} - \vec{r}_k)\cdot \hat{n}}{\left|\vec{r} - \vec{r}_k\right|^4} d^2 r \tag{31}$$

The use of Coulomb's law for the electric field is exact only in the case of a single charge q_k in the center of a spherically symmetric cavity. We can expect that this approximation for the local electrostatic field will be valid for cases that the molecule's surface is locally convex. This approximation is known as the "Coulombic approximation" and its validity has been examined in [45] and [47]. For cases that the surface is not locally convex, we cannot be sure how well this approximation will hold. However, our tests from chapter 4 showed that the Coulombic approximation works very well for a diverse set of molecules.

We will attempt to reformulate equation (31) from a surface to a volume integral since this will allow us an analytical calculation of the Born radius. For this, we employ Green's theorem for a vector field $\vec{A}$, from vector analysis:

$$\oint_{\partial V = S} \vec{A} \cdot d\vec{s} = \int_V \nabla \cdot \vec{A}\, d^3 r \tag{32}$$

We will set the vector field $\vec{A}$ equal to the expression inside the surface integral in (31), $\vec{A} = \dfrac{\vec{r} - \vec{r}_k}{\left|\vec{r} - \vec{r}_k\right|^4}$, and in order to avoid the singularity at the center, $\vec{r} = \vec{r}_k$, we break the integration over the solute volume V into two regions: the volume of a sphere of radius R_k centered at $\vec{r}_k$, $\left|\vec{r} - \vec{r}_k\right| \le R_k$, and the volume of the solute excluding the sphere of radius R_k, $\Omega_k = \left\{\vec{r} \in \Re^3 : \vec{r} \in V \wedge \left|\vec{r} - \vec{r}_k\right| > R_k\right\}$, where R_k is an arbitrary radius. Then we get

$$\begin{aligned} \oint_{\partial V = S} \frac{\vec{r} - \vec{r}_k}{\left|\vec{r} - \vec{r}_k\right|^4} \cdot d\vec{s} &= \int_V \nabla \cdot \frac{\vec{r} - \vec{r}_k}{\left|\vec{r} - \vec{r}_k\right|^4} d^3 r \\ &= \int_{\Omega_k} \nabla \cdot \frac{\vec{r} - \vec{r}_k}{\left|\vec{r} - \vec{r}_k\right|^4} d^3 r + \int_{\left|\vec{r} - \vec{r}_k\right| \le R_k} \nabla \cdot \frac{\vec{r} - \vec{r}_k}{\left|\vec{r} - \vec{r}_k\right|^4} d^3 r \end{aligned} \tag{33}$$

The integral over the volume of the sphere of radius R_k can be rewritten, as a surface integral over the surface of the sphere, and it yields $4\pi / R_k$:

$$\begin{aligned} \int_{\left|\vec{r} - \vec{r}_k\right| \le R_k} \nabla \cdot \frac{\vec{r} - \vec{r}_k}{\left|\vec{r} - \vec{r}_k\right|^4} d^3 r &= \oint_{\left|\vec{r} - \vec{r}_k\right| = R_k} \frac{\vec{r} - \vec{r}_k}{\left|\vec{r} - \vec{r}_k\right|^4} \cdot d\vec{s} \\ &= \frac{R_k}{R_k^4} 4\pi R_k^2 \\ &= \frac{4\pi}{R_k} \end{aligned} \tag{34}$$

For the integral over the volume Ω_k, we need to calculate the divergence of the vector field $\vec{r} - \vec{r}_k \big/ \left|\vec{r} - \vec{r}_k\right|^4$. If we use partial differentiation and the identity $\nabla \cdot \left(\vec{r} - \vec{r}_k\right) = 3$, we get

$$\nabla \cdot \left(\frac{\vec{r} - \vec{r}_k}{\left|\vec{r} - \vec{r}_k\right|^4} \right) = -\frac{1}{\left|\vec{r} - \vec{r}_k\right|^4} \tag{35}$$

Hence, by applying equations (34) and (35) in (33) we get an expression for the surface integral:

$$\oint_{\partial V=S} \frac{\vec{r}-\vec{r}_k}{\left|\vec{r}-\vec{r}_k\right|^4}\cdot d\vec{s} = \frac{4\pi}{R_k} - \frac{1}{4\pi}\int_{\Omega_k}\frac{1}{\left|\vec{r}-\vec{r}_k\right|^4}d^3r \tag{36}$$

If we apply equation (36) into (31), we can get an expression for the self-energy of polarization of atom k, $\Delta G_{pol,k}$. By comparing this expression to equation (22) we get an analytical formula for the Born radius of atom k, α_k:

$$\frac{1}{\alpha_k} = \frac{1}{R} - \frac{1}{4\pi}\int_{\Omega_k}\frac{1}{\left|\vec{r}-\vec{r}_k\right|^4}d^3r \tag{37}$$

The value of the radius R_k can be determined if we examine the case of the solute being just one ion. Then, the GB formula (18) becomes the well-known Born expression (16) and hence the radius R_k will be the ionic radius of that atom. In general, we will take this to be the van der Waals radius of the atom. The van der Waals radii are atomic parameters that are usually dependent on the forcefield parameter set used.

The conclusion drawn from equation (37) is that the Born radius, in the Coulombic approximation at least, is dependent only on the geometry of the solute. However, the volume integral in (37) is still very difficult to be calculated analytically for cases of arbitrary molecular geometry. Clearly, we will need to introduce additional approximations in order to arrive to an analytical formula for the calculation of the polar part of the solvation energy.

2.4 *Calculation of the Born Radii*

The integral in equation (37) cannot be calculated analytically for all but the simplest case of a spherical solute. For that reason there has been considerable interest in the literature for the calculation of this integral, and we will present in the following different methods that have been proposed.

- Numerical integration

The most obvious way to calculate such integral for arbitrary geometries is by numerical integration. The integration domain is divided by a cubic grid, elements of which are assigned as being inside or outside the solute. Then each grid element (of volume ΔV and center coordinate $\vec{r}$) contributes $\Delta V/|\vec{r}-\vec{r}_k|$ to the integral of the Born radius for atom k. Comparisons of the results of this method with numerical solutions of the PB equation on a set of small molecules and molecular complexes show that the GB results correlate very well to the PB answers, although there is a systematic error [48]. This is encouraging since it proves that the Coulombic approximation that was introduced in 2.3 is valid, at least for the molecular systems tested. However, the numerical integration is not practical for molecular simulations since it lacks derivatives (which are necessary for the calculation of forces), it is very slow and the accuracy and speed depend on the resolution of the grid used.

- The asymptotic model

Since the numerical solution is not practical and we need an analytical formula to calculate gradients, the asymptotic model attempts to provide an ad-hoc analytical solution. The model assumes that the Born radius of atom i is given by [49]:

$$\frac{1}{\alpha_i} = \frac{1}{R_i + \phi + P_1} - \left\{ \sum_{j \in \text{stretch}} \frac{P_2 V_j}{r_{ij}^4} + \sum_{j \in \text{bend}} \frac{P_3 V_j}{r_{ij}^4} + \sum_{j \in \text{nonbond}} \frac{P_4 V_j C}{r_{ij}^4} \right\} \tag{38}$$

The parameters ϕ, P_1, P_2, P_3, P_4, are scaling factors that are determined by fitting the predicted solvation energies to the numerical solutions of the PB equation of a set of small molecules. CCF is a "close contact function" that adjusts radii for nonbonded atoms that are too close to the central atom i and V_j is the van der Waals volume of atom j.

The similarity of equation (38) to equation (37) is obvious. Equation (38) is an ad-hoc formula with not much formal justification besides the fact that the term V_j / r_{ij}^4 corresponds to the energy loss of a classical charge-induced dipole interaction between the charge of atom i and the dielectric medium that is displaced by atom j [49]. One can think this as a first order approximation to the exact integral of equation (37). In fact, the V_j / r_{ij}^4 relationship for the contribution of atom j to the self-energy of atom i will hold only asymptotically, for large distances r_{ij}. At the same time, the true volume of atom j in the solute is different than the van der Waals volume V_j since the atoms intersect each other. This is the reason that the parameters of the model need to be fitted to numerical solutions of the PB equation, and this makes questionable the application of (38) to larger molecules. For example, in order to use this model for proteins and nucleic acids, a re-

parameterization was necessary [50]. Again, those parameters will be applicable only for the molecule set that were trained on and the forcefield parameters used.

- The pairwise descreening approximation (PDA)

In this model the polar solvation energy is calculated using a slightly different but equivalent formula for the Born radii [47], [51]:

$$\alpha_k^{-1} = \int_{R_k}^{\infty} \frac{A\left(r, \{r_{kk'}, R_{k'}\}_{all\ k'}\right)}{4\pi r^4} dr \tag{39}$$

where $A\left(r, \{r_{kk'}, R_{k'}\}_{all\ k'}\right)$ is the exposed surface area of a sphere of radius r centered at atom k and is intersected by all the other spheres k' of radius $R_{k'}$, centered at the locations of all the other atoms k' at distance $r_{kk'}$ from atom k [51]. Again, the integral in equation (39) is not possible to be calculated analytically because the atomic spheres k' overlap with each other. In order to account for this error, the radius of each atom k' is scaled by a factor $S_{k'}$. These factors are less than one so that each sphere is reduced to an effective volume. The final formula for the Born radii in this model has the form:

$$\alpha_k^{-1} = R_k^{-1} - \sum_{k'} H(r_{kk'}, S_{k'} R_k) \tag{40}$$

where $H(r_{kk'}, S_{k'} R_k)$ is a complex expression [51]. The scaling factors $S_{k'}$ are determined from fitting the solvation energies predicted by the PDA model to the numerical solutions of the PB equation, for a set of small molecules.

Although this method provides for an analytical approximation to the Born radii, the dependence on scaling factors and fittings to numerical solutions cause it to be not

easily applied to other molecular systems. Different parameterizations of the model have to be developed for it to be applicable to different systems [52], [53]. There have been attempts to improve on the results of the PDA model [54], but the main drawback of the dependence of the results on empirically determined scaling factors remains.

- The surface generalized Born model (SGB)

In the SGB model [55] the Born radii are calculated using the surface integral formula for the polarization self-energy of solvation, equation (31), instead of the volume integral (37). The two expressions are formally equivalent but the surface integral has the advantage of being faster to calculate numerically than the volume integral. By creating a triangulation of the surface of the solute, we can calculate the contribution to the Born radius at each surface element and add up to get the value of the integral over the solute. The advantage of this method is that the CPU time for the surface integration scales better than the volume numerical integration, as a function of the size of the solute, and there is no need for any parameterization or scaling factors. In practice, however, empirical short-range (involving atom-pairs whose spheres overlap) and long-range corrections (based on the amount of invagination of the molecular surface) are added to improve agreement with numerical solutions of the PB equation. Also, the accuracy of the method depends on the resolution of the grid used and derivatives of the Born radii are not readily available.

- The overlapping spheres approximation and the analytical volume model

The integration region Ω_k in the volume integral of equation (37) is the volume of the solute that is inaccessible to the solvent, V, minus the volume of a sphere of radius R_k centered at $\vec{r}_k$. The solute volume is defined as the interior of the solvent accessible surface, which in turn is defined by the van der Waals spheres of each atom extended by the probe radius of the solvent, r_p, as is shown in Figure 2 [11]. We can partition the integration region into the set of sub-volumes $V_{k'}$ that each neighboring atom k' occupies and then rewrite the integral as a sum of the integrals over each sub-volume:

$$\int_{\Omega_k} \frac{1}{\left|\vec{r}-\vec{r}_k\right|^4} d^3r = \sum_{k'\neq k} \int_{V_{k'}} \frac{1}{\left|\vec{r}-\vec{r}_k\right|^4} d^3r \tag{41}$$

Equation (41) is exact as long as the partition of the integration region into sub-volumes is consistent, i.e., $\sum_{k'\neq k} V_{k'} = \Omega_k$. However, the shape of the sub-volumes is highly irregular and the volume integrals in (41) cannot be solved analytically except in the simple case that $V_{k'}$ is a sphere (it's not intersected by the neighboring atoms).

In particular, the integral of the quantity $1/\left|\vec{r}-\vec{r}_k\right|^4$ over the volume of a sphere of radius $R_{k'}$, centered at $\vec{r}_{k'}$, minus a possible overlap with the sphere $(R_k, \vec{r}_k)$, can be solved analytically [45]:

$$\int_{V_k-(V_k\cap V_{k'})} \frac{1}{\left|\vec{r}-\vec{r}_k\right|^4} d^3r =$$

$$= \begin{cases} \dfrac{2\pi R_{k'}}{r_{kk'}^2 - R_{k'}^2} + \dfrac{\pi}{r_{kk'}} \ln \dfrac{r_{kk'} - R_{k'}}{r_{kk'} + R_{k'}}, & \text{if } r_{kk'} \geq R_k + R_{k'} \\ \dfrac{\pi}{R_k}(2-\theta) - \dfrac{\pi}{r_{kk'} + R_{k'}} + \dfrac{\pi}{r_{kk'}} \ln \dfrac{R_k}{r_{kk'} + R_{k'}}, & \text{if } \left|R_k - R_{k'}\right| \leq r_{kk'} \leq R_k + R_{k'} \\ \dfrac{2\pi R_{k'}}{r_{kk'}^2 - R_{k'}^2} + \dfrac{4\pi}{R_k} + \dfrac{\pi}{r_{kk'}} \ln \dfrac{R_{k'} - r_{kk'}}{R_{k'} + r_{kk'}}, & \text{if } r_{kk'} < \left|R_k - R_{k'}\right| \wedge R_k \leq R_{k'} \\ \dfrac{4\pi}{R_k} - \dfrac{4\pi}{R_{k'}}, & \text{if } r_{kk'} = 0 \wedge R_k \leq R_{k'} \\ 0, & \text{if } r_{kk'} < \left|R_k - R_{k'}\right| \wedge R_k \geq R_{k'} \end{cases} \quad (42)$$

$$\text{where } \theta = \frac{r_{kk'}^2 + R_k^2 - R_{k'}^2}{2 r_{kk'} R_k}$$

The five cases result from the different topologies that arise between the two spheres, $V_k = (R_k, \vec{r}_k)$ and $V_{k'} = (R_{k'}, \vec{r}_{k'})$. They correspond, respectively, to the cases of no overlap, partial overlap, k' completely swallows k but the spheres are not concentric, k' completely swallows k and the spheres are concentric, and k completely swallows k'.

Equation (42) would allow for an analytical calculation of the Born radii, if the solute were a set of non-overlapping spheres. In reality though the atoms intersect each other so we cannot use (42) without introducing some approximations. In the overlapping spheres approximation, we attempt to represent the sub-volumes $V_{k'}$ as spheres and use equation (42) for the calculation of the Born radius of atom k. We cannot simply use the radii $R_{k'}$ (which are equal to the van der Waals radius of atom k', $R_{k'}^{vdW}$ plus the solvent's probe radius r_p, $R_{k'} = R_{k'}^{vdW} + r_p$) for the sub-volumes because we would overestimate the solute volume due to the interatomic overlap. An attempt to use

effective radii that depend on atom types [45], derived from studies of crystallographic protein structures [56], lead to moderate correlation between the calculated atomic self-energies of polarization and the results of numerical calculations [57]. This is probably due to an inadequate description of the molecular geometry. Standard effective volumes that are dependent only on atom types cannot distinguish between atoms on the surface of the solute and atoms that are deeply buried inside the molecular cavity. Secondly, these effective volumes were derived from studies on a finite number of proteins, and the application of those values to other systems is not obvious.

If we had, however, a fast and accurate way of calculating the true sub-volume of each atom k in the molecular cavity by using a fused-sphere model of the solute, V_k, we could define for each atom k an effective radius R_k^{eff} such that the volume of a sphere of radius R_k^{eff} is equal to the true sub-volume V_k:

$$R_k^{eff} = \sqrt[3]{\frac{3V_k}{4\pi}} \tag{43}$$

Then, we can perform the volume integration in equation (41) analytically by use of (42), where each integration region is a sphere of radius R_k^{eff}. Effectively, in this approximation we assume that the atoms are not overlapping, but the spherical volumes that we assign to each atom have been corrected for the overlap.

If the method for the calculation of the true sub-volumes V_k is analytical, then the final formula for the Born radii will be also analytical and we will have the capability to calculate the full gradient of the polar solvation energy. Obviously, for the method to be

practically applicable in molecular simulations, the volume calculations have to be computationally very efficient. In the "Analytical Volume Generalized Born" model (AVGB) that we propose here, the volumes are calculated accurately and efficiently by means of analytical algorithms. The algorithms for the volume calculations are described in detail in chapter 3.

2.5 Improvements on the Generalized Born Model

The GB model as described here has two limitations: the effect of salt on the solvation energy is not taken into account and the charge density of the solute is assumed to be a set of point charges. However, there are improvements that can be done on the GB formalism that address these problems.

- Inclusion of salt effects

The GB model was derived as an approximation between two limits, the case of two widely separated spheres and the case of a spherical ion. When we include salt effects we can get analytical solutions for these two special cases, in the limit of low salt concentration from the linearized form of the PB equation (7). If κ is the Debye-Huckel dielectric screening parameter, then for two widely separated spheres i and j, the free energy of polarization is [32]:

$$\Delta G_{pol} = -\left(\frac{1}{\varepsilon_{in}} - \frac{e^{-\kappa r_{ij}}}{\varepsilon_{out}}\right)\frac{q_i q_j}{r_{ij}} \tag{44}$$

and for the case of a spherical ion of radius α it is [19]:

$$\Delta G_{pol} = -\frac{1}{2}\left(\frac{1}{\varepsilon_{in}} - \frac{e^{-\kappa r_{ij}}}{\varepsilon_{out}}\right)\frac{q^2}{\alpha} - \frac{q^2\kappa}{2\varepsilon_{out}\left(1+\kappa r_p\right)} \tag{45}$$

where r_p is the probe radius of the solvent. To a close extend, these two limits can be obtained by the simple substitution [32]:

$$\left(\frac{1}{\varepsilon_{in}} - \frac{1}{\varepsilon_{out}}\right) \rightarrow \left(\frac{1}{\varepsilon_{in}} - \frac{e^{-\kappa f_{ij}}}{\varepsilon_{out}}\right) \tag{46}$$

where f_{ij} is the interpolation formula (19). Although this is a very simple approximation, this model can reproduce well the salt contribution to the solvation energy, as is shown in Figure 7.

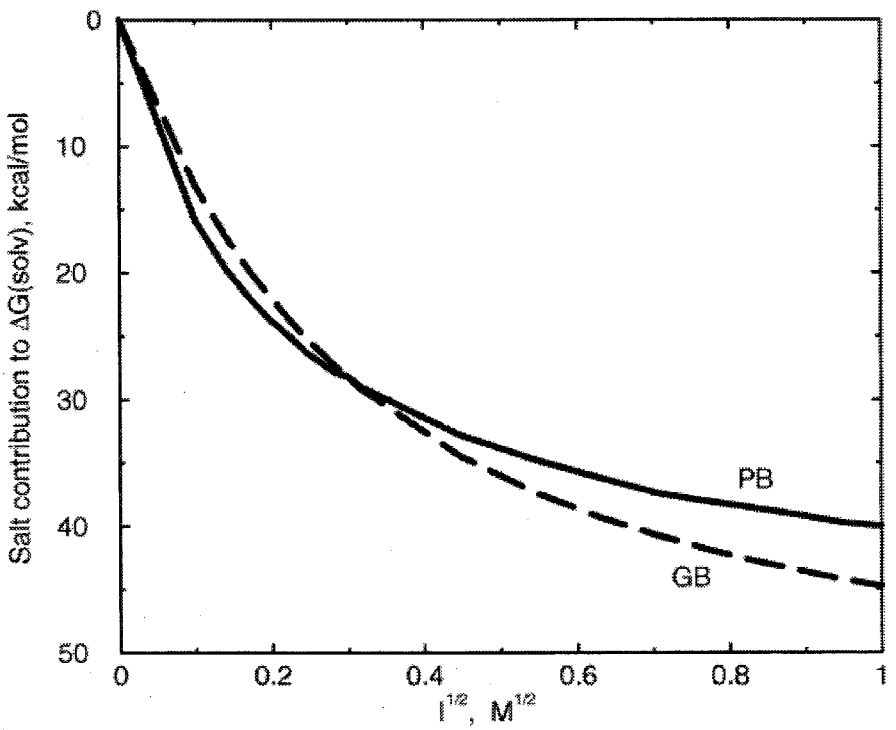

Figure 7. Comparison between PB and GB predictions of the salt contribution to the solvation energy for a B-DNA structure, as a function of the square root of the concentration of added monovalent salt. (Figure from reference [32].)

- Gaussian charge distributions

In the derivation of the Born radii of atom k, equation (37), we made the assumption that the solute charge distribution was a set of point charges. If instead we assume that the charge density for every atom has a gaussian shape,

$$\rho_k(\vec{r}) = q_k \pi a_k^{3/2} \exp\left(-a_k^2 (\vec{r} - \vec{r}_k)^2\right), \quad a_k = \frac{(\pi/2)^{1/2}}{R_k} \tag{47}$$

we can re-derive the formalism of section 2.3 and get a modified expression for the Born radii. This theory was described in [57] and [58], along with a modification for the partition of the integral in the overlapping spheres approximation by assuming partial atomic densities of also gaussian shape. The resulting formulation is more complicated than the original GB theory (and potentially more CPU intensive), but this is an interesting addition to the GB model.

3 *Geometric Algorithms for the Fused-Sphere Model*

In chapter 1 we showed how the short-range contributions to the solvation energy are dependent on the exposed area of the solute. In chapter 2, using the Generalized Born approximation, we reduced the calculation of the polar contribution to the solvation energy to the computation of the occupied volume of the solute. In particular, for every atom i in the solute (consisting of N atoms total), the exposed area A_i and the occupied volume V_i of that atom need to be calculated, along with their gradients with respect to the atomic coordinates. The sub-volumes V_i are needed for the polar part of the calculation and the areas A_i for the short-range cavity-van der Waals term. The definition of the exposed area and the solvent excluded volume for the solute is, as defined by Richards, the solvent accessible surface area (SASA) and solvent excluded volume [11]. We assume a fused-sphere model for the solute where each sphere has radius r_i equal to the van der Waals radius of the atom it represents, r_i^{vdW} extended by the probe radius of the solute, r_p, i.e., $r_i = r_i^{vdW} + r_p$. We are extending the radii of the atoms by the probe radius, according to the definition of the SASA as the surface traced by the center of a spherical solvent probe, as it rolls around the van der Waals spheres of the solute (Figure 2). Since many atoms intersect each other, the spheres are fused into each other, as is shown in Figure 8.

The calculation of the volumes and areas has to be analytical and fast, in order for this model to be practical for molecular simulations. At the same time the algorithms used will have to be applicable to all different topologies that may arise between an atom and its neighbors. It is clear from Figure 8 that these topologies can vary wildly from atom to atom. Therefore, it is crucial to have very robust algorithms.

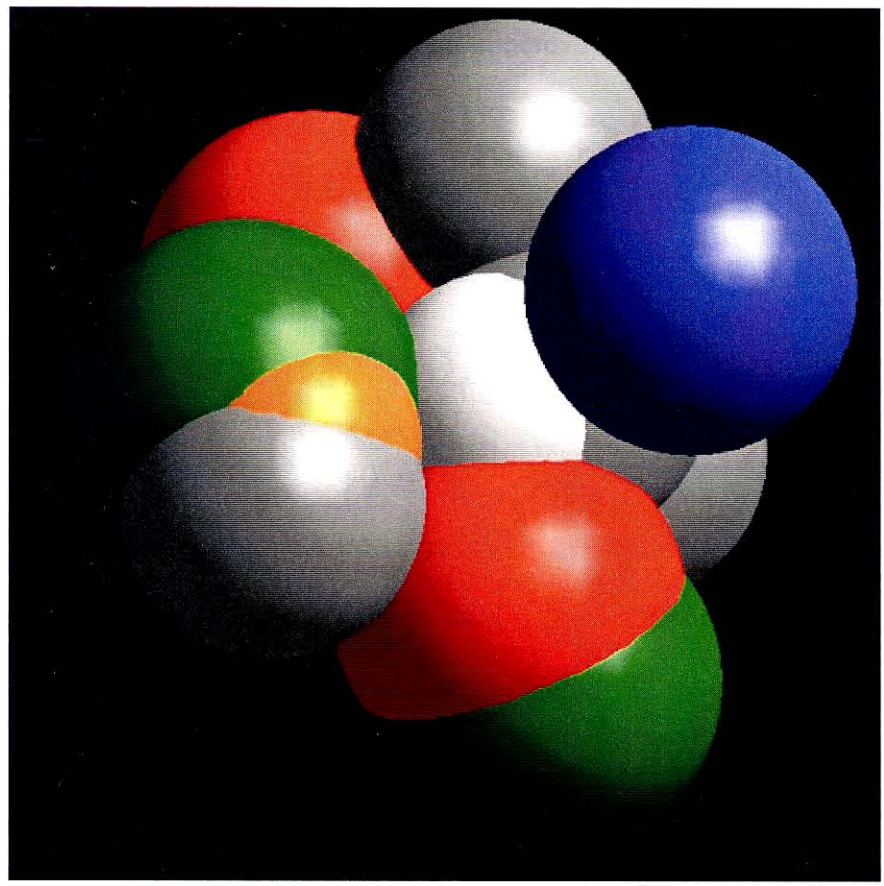

Figure 8. An example of the fused-sphere model: The central atom (white) is surrounded by a number of neighbors that define its exposed surface area and volume.

3.1 Volume Calculation

Numerical techniques were the first to be used for the calculation of molecular volumes, either by grid or Monte Carlo methods. Grid points or randomly generated points are identified as being in or out of the fused-sphere model. The fraction of the points inside determines the volume of the solute. Such methods besides not being able to calculate derivatives are very computationally inefficient and cannot provide the individual sub-volumes contributed by each sphere. Another method proposed was the "inclusion-exclusion" method by Kratky [59] where using the inclusion-exclusion formula of set theory we can calculate the union of N spheres as N summations over combination of intersections of the individual spheres:

$$V(1 \cap 2 \cap \ldots \cap N) = \sum_{i}^{N} V(i) - \sum_{i>j}^{N} V(i \cap j) + \sum_{i>j>k}^{N} V(i \cap j \cap k) - \ldots \qquad (48)$$

Obviously this algorithm can be very complex for complicated topologies of multiple intersections and for systems with large number of spheres. There have been attempts to simplify these expressions [60], [61] but the implementation of these methods is still particularly cumbersome and it is not possible to include all topologies. Also, the method does not provide the individual sub-volumes of each atom.

In order to calculate the individual sub-volumes we need to unambiguously define a way to partition the fused-sphere model to the individual contributions of each sphere. The simplest case of two overlapping spheres can give us the principle of the decomposition: the two spheres intersect each other and form a circle on the boundary, the circle of intersection (COI) (Figure 9). The COI defines a separating plane that cuts

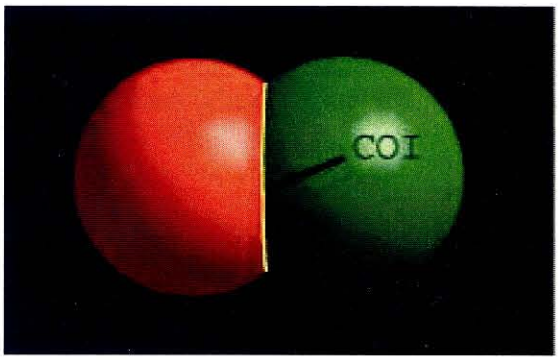

Figure 9. Two intersecting spheres and the circle of intersection (COI.)

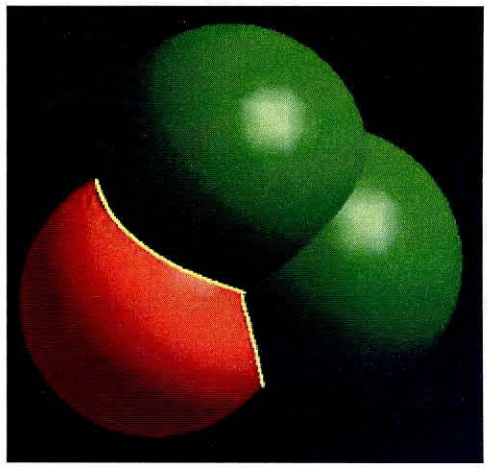

Figure 10. Three intersecting spheres. The COI's intersect with each other.

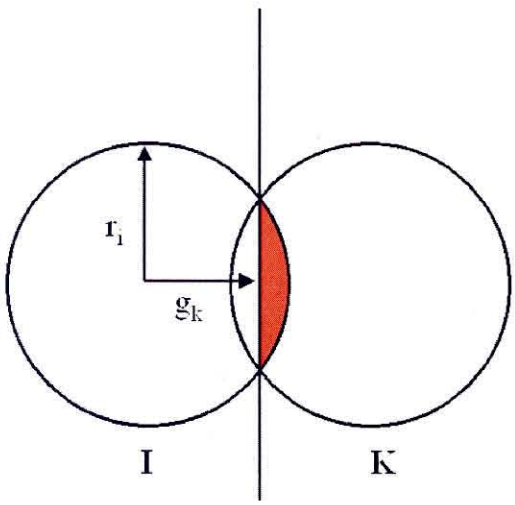

Figure 11. Two intersecting spheres, i and k, separated by the separating plane. The distance between the separating plane and the center of sphere i is g_k.

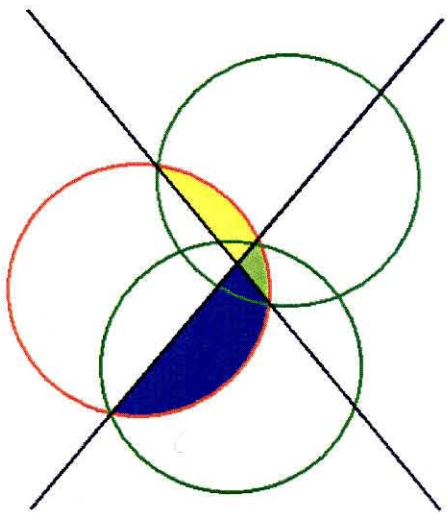

Figure 12. Three intersecting spheres and the corresponding separating planes for the central sphere (red.)

through the spheres (Figure 11). The intersection of the separating plane with the surface of each sphere is the COI. If there are more than two spheres, the separating planes might

intersect each other, forming a more complicated topology. The circles of intersection intersect each other (Figure 10), as the separating planes do also (Figure 12).

The intersecting planes that cut through the spheres separate the fused-sphere model into building blocks. Each building block is a complicated geometrical shape that is made of a sphere that has been cut by the corresponding intersecting planes from each neighbor. Figure 13 illustrates the decomposition of the fused-sphere model into the building blocks. The building blocks consist of planar faces and regions of the sphere surface that is left uncut.

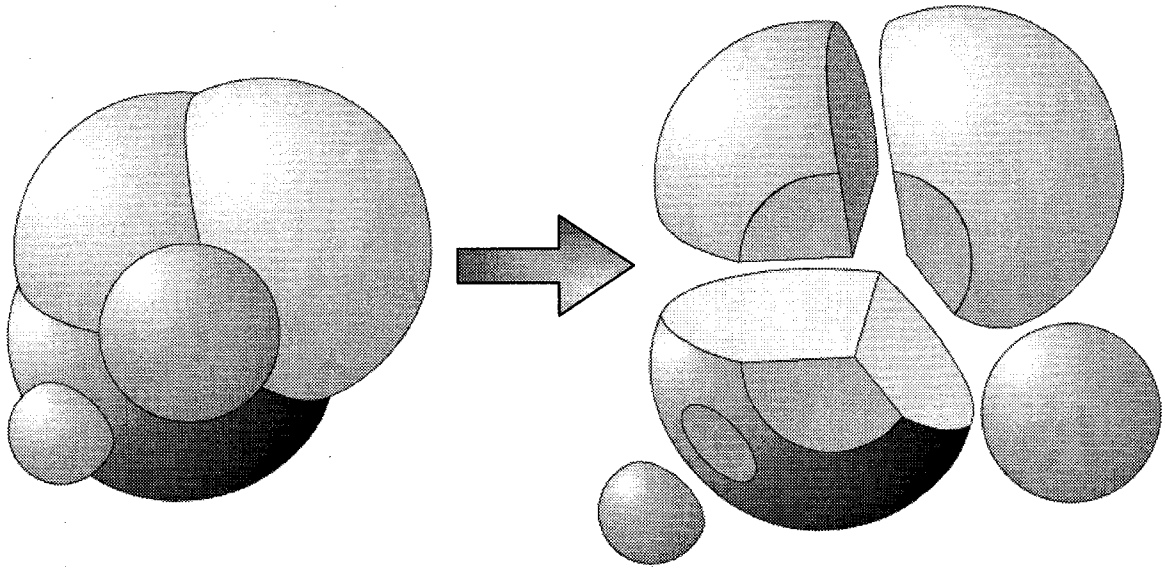

Figure 13. Decomposition of the fused-sphere model into the building blocks that correspond to each atom.

The decomposition procedure that we introduce here is very similar to the concept of the Voronoi diagrams [62] and the weighted Voronoi diagrams or power diagrams [63], which are fundamental structures in computational geometry and have found many applications in different fields in science. In short, given a set of points in space their

Voronoi diagram divides the space into regions according to the nearest-neighbor rule: each point is associated with the region of space closest to it. Thus, for every point inside a region, its distance to the generating point is less than (or equal to) its distance to any other point in the set. According to this definition, the regions are defined by separating planes, which are the bisector planes between two neighboring points. The weighted Voronoi diagram is a generalization of the Voronoi diagram; the separating plane is not the bisector plane, but it is parallel to it. If we assign spheres of different radii to every point in the set, the separating plane is defined as the plane for which every point in it has equally long tangent line segments to both of the spheres. If the spheres intersect, this plane is the separating plane of the two spheres. An example of a weighted Voronoi diagram is given in Figure 14.

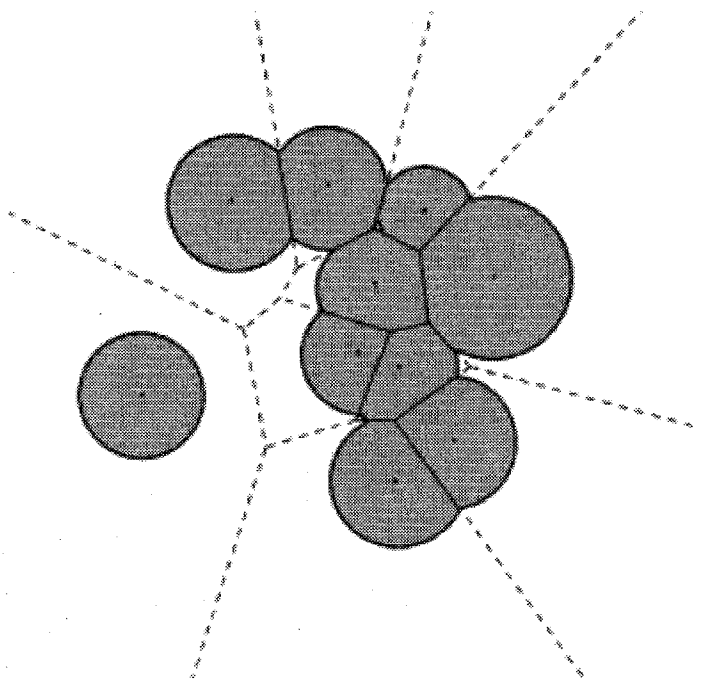

Figure 14. The weighted Voronoi diagram (or power diagram) for a set of spheres, in two dimensions. (Figure from reference [64].)

The difference between the weighted Voronoi diagram and the decomposition described here is that we do not take into account separating planes between spheres that do not intersect, as is shown in Figure 14. However, both methods produce exactly the same decomposition of the fused-sphere model. We prefer the method proposed here, which is essentially a simplified version of the weighted Voronoi diagram because it facilitates the calculation of each atom independently of the others, as it will be shown in the following. Instead, methods for the calculation of the Voronoi diagrams are global in character [65] which has disadvantages in the parallelizability and robustness of the implementation of the calculation. A formal study of the applications of advanced computational geometry constructs on the fused-sphere model is given in [66].

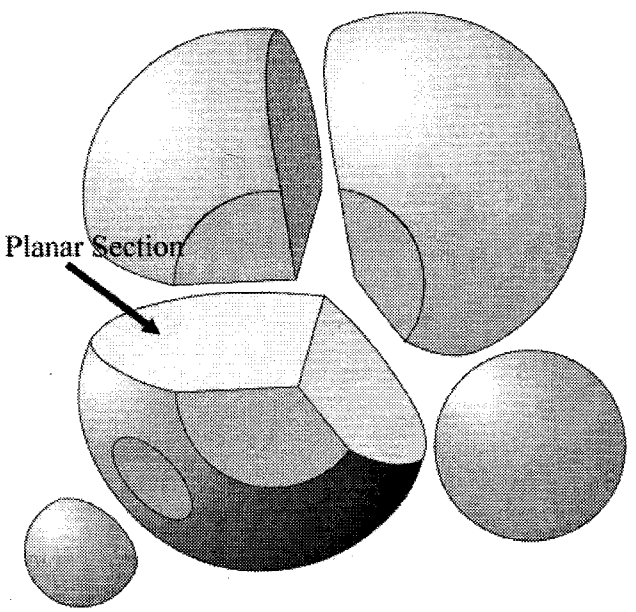

Figure 15. The building block and the planar sections formed by the neighbors.

In order to calculate the volume of the building blocks we need to continue the decomposition process hierarchically, until we are able to describe the objects from well-

defined geometrical shape. The building block is made from the atom's sphere cut by the intersecting planes between the atom and its neighbors. These cuts are planar sections on the surface of the block and they correspond to the neighbor that formed them (Figure 15).

If we connect each point on a planar section to the center of the sphere we form a solid that has the shape of a cone-pyramid with the planar section as its base. If we "carve" out all the cone-pyramids from the building block, we are left with a spherical sector. The spherical sector is the solid that results by connecting all points of the exposed surface of the atom (the spherical part of the building block) to the center of the sphere. The cone-pyramid decomposition was proposed in [67] and it is illustrated in Figure 16.

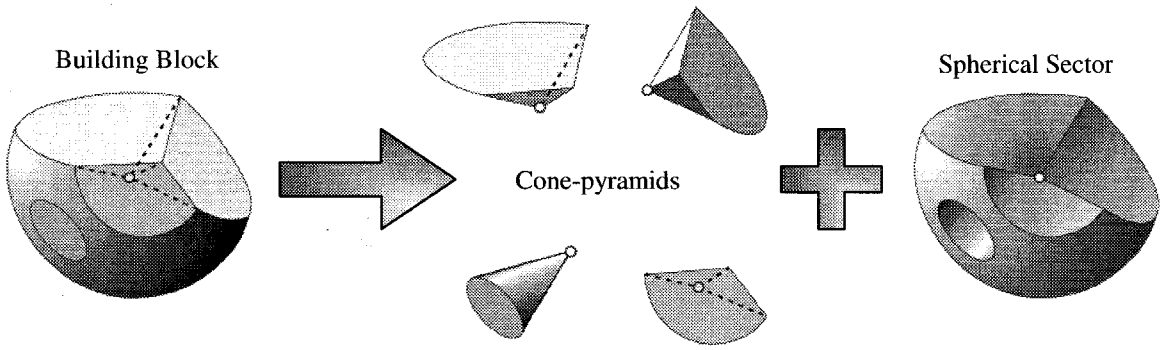

Figure 16. Decomposition of the building block into cone-pyramids and a spherical sector.

The advantage of this decomposition is that the volume of cone-pyramids and spherical sectors can be calculated analytically. If a cone-pyramid has a base of area A and distance from the tip d, then the volume is $V_{con-pyr} = \frac{1}{3}Ad$. Similarly, the volume of

the spherical sector of radius r is the sum of the volumes of infinitesimal pyramids of base area ds :

$$V_{sph-\sec} = \int dV = \int \frac{1}{3} r ds = \frac{1}{3} rS \tag{49}$$

Thus, the volume of the spherical sector is proportional to the exposed area of the atom. In general, for the building block of an atom i with radius r_i and exposed surface area $S_i^{\exp}$, formed by $j = \{1,...,M\}$ neighbors that are separated from i by planes of distance g_{ij} from the center of i, the volume V_i is given by

$$V_i = \frac{1}{3} r_i S_i^{\exp} + \sum_{j=1}^{M} \frac{1}{3} g_{ij} A_{ij} \tag{50}$$

where A_{ij} is the area of the planar section that is formed on atom i from neighbor j. Note that in equation (50) the distance g_{ij} of the separating plane of neighbor j from atom i can be negative if the neighbor "swallows" the atom; that is, the center of i is buried by j (see Figure 11). The algebraic sum subtracts correctly the overlaps that may appear between the cone-pyramids in the case of a swallower neighbor [68].

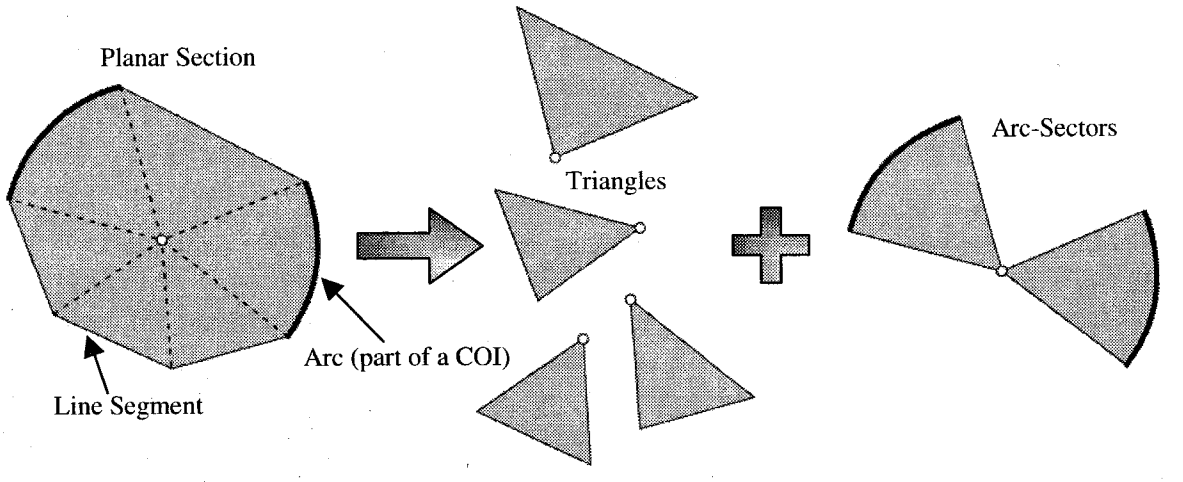

Figure 17. Decomposition of a planar section into triangles and arc-sectors.

The area of a planar section, A_{ij}, between atom i and neighbor j can be calculated in a similar fashion by decomposing it into triangles and arc-sectors (Figure 17). In general, the planar section is bounded by a set of arcs and line segments. The arcs are parts of the COI formed between i and neighbor j. The line segments correspond to intersections of the atom and neighbor with other neighbors k of the atom. They are formed by the intersections of the atom-neighbor plane with the other neighbors' intersection planes. For the decomposition, we need to pick a reference point on the surface of the $i-j$ intersecting plane that will define the triangles and arc-sectors. This point is defined to be the intersection between the line that connects the centers of i and j, and the intersection plane. Obviously, the connecting line is normal to the intersection plane. Thus, if we have M_{ij}^{arcs} arcs on the planar section and M_{ij}^{seg} line segments, then the planar section's area is

$$A_{ij} = \sum_{\lambda=1}^{M_{ij}^{arcs}} \frac{1}{2} a_{ij} S_{\lambda} + \sum_{\mu=1}^{M_{ij}^{seg}} \frac{1}{2} h_{\mu} t_{\mu} \tag{51}$$

where S_{λ} is the length of the λ^{th} arc on the planar section, a_{ij} is the radius of the $i-j$ COI, t_{μ} is the length of the μ^{th} line segment and h_{μ}, the distance between the reference point and the line segment.

Equations (50) and (51) allow us to calculate the volume of the building blocks and thus the volume of each atom accurately, as long as we can calculate the exposed area S_i^{exp} and all the other quantities, g_{ij}, a_{ij}, t_{μ}, h_{μ}. We will present a method to calculate exposed areas in section 3.2.

3.2 *Area Calculation*

The calculation of the surface area of the fused-sphere model has attracted considerable attention in the literature because of its importance in the description of the solvation energy. The description of solvation in terms of the solvent accessible surface area (SASA) -see equation (2)- made the calculation of the SASA and its gradient with respect to the atomic coordinates necessary. Numerical methods are characterized by the way of approximating the surface, and they are too slow to be used in molecular simulations (see [69] and references therein). Analytical methods were first proposed by Connolly [70] and Richmond [71], and they use the Gauss-Bonnet theorem of differential geometry [72].

The Gauss-Bonnet (GB) theorem is the most fundamental theorem in differential geometry and topology, and in its simplest form it asserts that the excess over π of the sum of the interior angles φ_1, φ_2, φ_3 of a geodesic triangle T is equal to the integral of the gaussian curvature K over T, or formally

$$\sum_{i=1}^{3}\varphi_i - \pi = \iint_T K(s)\,ds \tag{52}$$

The general form of the GB theorem (global GB theorem) is: Let R be a regular region of an oriented surface and let C_1, C_2,..., C_n be the closed, simple, piecewise regular curves which form the boundary ∂R of R. If each C_i is positively oriented and Ω_1, Ω_2, ..., Ω_m are all the external angles of the curves C_i, then

$$\sum_{i=1}^{n}\int_{C_i} k_g(l)\,dl + \iint_R K(s)\,ds + \sum_{i=1}^{m}\Omega_i = 2\pi\chi(R) \tag{53}$$

where l denotes the arc length of C_i, $k_g(l)$ is the geodesic curvature of the arc, $K(s)$ is the gaussian curvature of the surface and $\chi(R)$ the Euler-Poincaré characteristic of the surface R. The Euler-Poincaré characteristic is a topological constant of the surface and, in general, if a two-dimensional surface has q holes and h handles, then $\chi = 2-(2h+q)$ [74].

The global GB theorem can be applied in the case of an atom in the fused sphere model. We will attempt to calculate the area of an atom that is buried by the neighboring atoms (Figure 18).

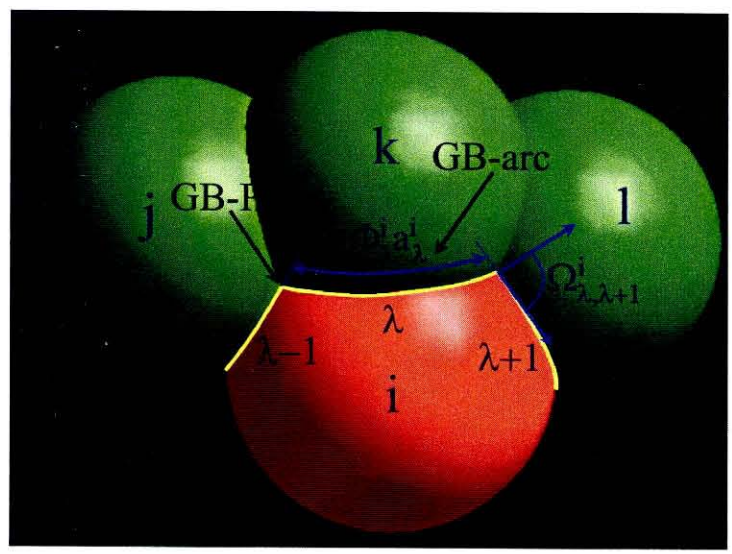

Figure 18. Application of the Gauss-Bonnet theorem on the surface of sphere i, intersected by neighbors j, k, l. (Figure adapted from reference [73].)

The intersecting neighbors form circles on the sphere of atom i, the COI's, which intersect each other. The buried surface of i is bounded by arcs which are pieces of the

COIs. We name these arcs the Gauss-Bonnet arcs (GB-arcs) and the closed oriented path that bounds the buried surface the Gauss-Bonnet path (GB-path). For the λ^{th} arc, the arc length is Φ_λ^i and the radius of the corresponding COI is a_λ^i and the polar angle of the COI is Θ_λ^i. The exterior angle between arc λ and $\lambda+1$ is Ω_λ^i. The gaussian curvature of the sphere of radius r_i is $K = 1/r_i^2$. In order to calculate the geodesic curvature of the arc, we apply the GB theorem (53) on a spherical cap of height $r_i - d_\lambda^i$ and radius a_λ^i on the base. Since for a spherical cap the area is $2\pi r_i\left(r_i - d_\lambda^i\right)$ and $\chi = 1$, we have

$$k_g^\lambda 2\pi a_\lambda^i + 2\pi r_i\left(r_i - d_\lambda^i\right)\frac{1}{r_i^2} = 2\pi$$

hence (54)

$$k_g^\lambda = \frac{d_\lambda^i}{r_i a_\lambda^i}$$

Then, since $\cos\Theta_\lambda^i = d_\lambda^i / r_i$, the geodesic curvature of the λ arc must be

$$k_g^\lambda = \frac{\cos\Theta_\lambda^i}{a_\lambda^i} \tag{55}$$

By applying the GB theorem (53) on the surface buried by the neighbors of atom i and using (55), we get the buried area:

$$\sum_{\lambda=1}^{P}\frac{\cos\Theta_\lambda^i}{a_\lambda^i}\Phi_\lambda^i a_\lambda^i + S_i^{Buried}\frac{1}{r_i^2} + \sum_{\lambda=1}^{P}\Omega_\lambda^i = 2\pi\chi$$

hence (56)

$$\mathrm{S_i^{Buried}} = r_i^2\left[2\pi\chi - \sum_{\lambda=1}^{P}\left(\Omega_\lambda^i + \Phi_\lambda^i\cos\Theta_\lambda^i\right)\right]$$

The Euler-Poincaré characteristic describes the topology of the buried surface. The buried surface is topologically equivalent (homomorphic) to a sphere with n holes, as many as the closed paths formed by the neighbors, hence $\chi = 2 - n$. In general, there can be more

than one disconnected piece of the surface of the atom that is buried. The exposed area of the atom is the total area $4\pi r_i^2$ minus the sum of the buried areas of each disconnected piece, given from equation (56):

$$S_i^{\exp} = r_i^2\left[2\pi(2-\chi)-\sum_{\lambda=1}^{P}\left(\Omega_\lambda^i+\Phi_\lambda^i\cos\Theta_\lambda^i\right)\right] \tag{57}$$

where now χ is the sum of the individual χ's for each disconnected piece of the buried surface and P is the total number of GB-arcs on the surface of the atom.

Using the GB theorem, the calculation of the exposed area is reduced to the calculation of the arcs of the COI's of the neighbors on the surface of the atom and the angles between them. In order to calculate these quantities, we need to parameterize the geometrical problem. Initial attempts used a Cartesian system that made the final formulas extremely cumbersome [71]. Instead, we will use a parameterization that is equivalent for each atom and was introduced in its basic form in [70] and formalized in [73]. We will use the center of the central atom i as the center of the coordinate system (Figure 19). If atom i of radius r_i is located at $\vec{x}_i$ and its neighbor k of radius r_k at $\vec{x}_k$, then the distance from the center of i to the COI that is formed by k is $g_k^i\hat{\mu}_k^i$, where

$$\hat{\mu}_k^i = \vec{x}_k - \vec{x}_i \big/ \left|\vec{x}_k - \vec{x}_i\right|$$
$$\text{and} \tag{58}$$
$$g_k^i = \frac{\left|\vec{x}_k - \vec{x}_i\right|^2 + r_i^2 - r_k^2}{2\left|\vec{x}_k - \vec{x}_i\right|}$$

and the radius of the COI, a_k^i, is

$$a_k^i = \sqrt{r_i^2 - \left(g_k^i\right)^2} \tag{59}$$

The polar angle Θ_k^i of the COI is

$$\cos\Theta_k^i = g_k^i / r_i \tag{60}$$

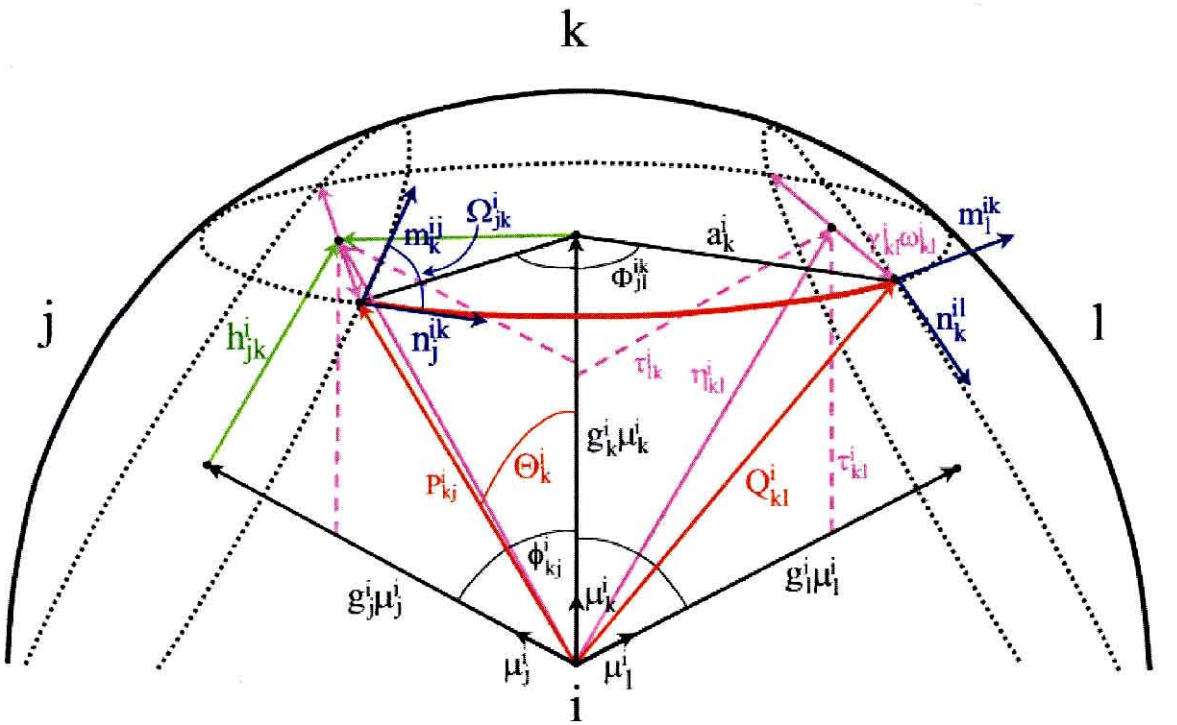

Figure 19. Parameterization of the Gauss-Bonnet arcs. In this example, the central atom i is intersected by three neighbors, j, k, l. The $i-j$ and $i-k$ COI's intersect each other, as the $i-k$ and $i-l$ do also. See text for explanation of the vector quantities. (Figure adapted from reference [73].)

Now, if the neighbor k is intersected by two other neighbors, j and l, the $i-k$ COI is intersected by the $i-j$ and $i-l$ COIs (located at $g^i_j\hat{\mu}^i_j$ and $g^i_l\hat{\mu}^i_l$ respectively) and the $i-k$ COI becomes a GB-arc. The orientation of the GB-Path is very important because depending on this orientation the calculation of the buried area, equation (57), will yield the area on one or the other side of the GB-path. The right orientation for the

GB-arcs in our problem is CCW, looking from above (outside the central atom). Then, the k GB-arc is bounded by two points, $\vec{P}_{kj}^{\,i}$ and $\vec{Q}_{kl}^{\,i}$, that are the intersection points between the $i-k$ and $i-j$ COIs and $i-k$ and $i-l$ COIs respectively. We call these points "GB-points." If the midpoint of the segment defined by the intersection of the $i-k$ and $i-j$ COIs is $\vec{\eta}_{kj}^{\,i}$, $2\gamma_{kj}^{i}$ is the total length of the segment and $\hat{\omega}_{ikj}$ the unit vector on the direction of the segment, then we have the following:

$$\begin{aligned}
\cos\phi_{kj}^{i} &= \hat{\mu}_{k}^{i}\cdot\hat{\mu}_{j}^{i} \\
\vec{\eta}_{kj}^{\,i} &= \tau_{kj}^{i}\hat{\mu}_{k}^{i}+\tau_{jk}^{i}\hat{\mu}_{j}^{i} \\
\tau_{kj}^{i} &= \frac{g_{k}^{i}-g_{j}^{i}\cos\phi_{kj}^{i}}{\sin^{2}\phi_{kj}^{i}} \\
\hat{\omega}_{ikj} &= \frac{\hat{\mu}_{k}^{i}\times\hat{\mu}_{j}^{i}}{\sin\phi_{kj}^{i}} \\
\gamma_{kj}^{i} &= \sqrt{r_{i}^{2}-g_{k}^{i}\tau_{kj}^{i}-g_{j}^{i}\tau_{jk}^{i}}
\end{aligned} \tag{61}$$

where ϕ_{kj}^{i} is the angle between the $i-k$ and $i-j$ COIs. We are now able to calculate the GB-points:

$$\begin{aligned}
\vec{P}_{kj}^{\,i} &= \vec{\eta}_{kj}^{\,i}+\gamma_{kj}^{i}\hat{\omega}_{ikj} \\
\vec{Q}_{kl}^{\,i} &= \vec{\eta}_{kl}^{\,i}-\gamma_{kl}^{i}\hat{\omega}_{ikl}
\end{aligned} \tag{62}$$

and then the tangent unit vectors that define the exterior angles Ω between consecutive GB-arcs,

$$\begin{aligned}
\hat{n}_{j}^{ik} &= \frac{\hat{\mu}_{k}^{i}\times\vec{P}_{kj}^{\,i}}{a_{k}^{i}} \\
\hat{m}_{l}^{ik} &= \frac{\hat{\mu}_{k}^{i}\times\vec{Q}_{kl}^{\,i}}{a_{k}^{i}} \\
\Omega_{jk}^{i} &= -\arccos\left(\hat{n}_{j}^{ik}\cdot\hat{m}_{k}^{ij}\right)
\end{aligned} \tag{63}$$

since the exterior angles are negatively oriented [72]. The angular arc length of the GB arc can be calculated from the inner product of the vectors $\vec{n}_j^{ik}$ and $\vec{m}_l^{ik}$. The arc length is either the arc-cosine of the inner product or the complimentary angle. In compact form, we have

$$\begin{aligned} \Phi_{jl}^{ik} &= \left(1 - S_{jl}^{ik}\right)\pi + S_{jl}^{ik} \arccos\left(\vec{n}_j^{ik} \cdot \vec{m}_l^{ik}\right) \\ &\text{where} \\ S_{jl}^{ik} &= \text{sign}\left(\hat{\mu}_k^i \times \left(\vec{n}_j^{ik} \times \vec{m}_l^{ik}\right)\right) \end{aligned} \tag{64}$$

S_{jl}^{ik} is the sign of the relative orientation of the vector $\hat{\mu}_k^i$ and the tangent vectors $\vec{n}_j^{ik}$ and $\vec{m}_l^{ik}$. Thus, the arc-length of the $i-k$ COI's GB-arc bounded by neighbors j and l is

$$S_{jl}^{ik} = a_k^i \Phi_{jl}^{ik} \tag{65}$$

This is the arc-length of each arc-sector in equation (51). Finally, for the planar section formed by neighbor k, since the reference point for the planar section is $g_k^i \hat{\mu}_k^i$, the base of a triangle in the decomposition corresponds to the line segment formed by the intersection of the COIs of the neighbors k and j that define it. Then, the height of the triangle is (see Figure 19)

$$h_{jk}^i = \frac{g_k^i - g_j^i \cos\phi_{kj}^i}{\sin\phi_{kj}^i} \tag{66}$$

and the length of the base, t_{jk}^i, can be determined by the positions of the vertices that define it. These vertices are the intersection points of three spheres, which might or might not include the central atom (see Figure 15). If the central atom is included, then the vertex is the corresponding GB-point; otherwise it can be calculated analytically by employing the same vector parameterization that we have used in this section.

We now have analytical expressions for the areas and volumes of each atom in the fused-sphere model. We can differentiate these expressions with respect to each neighbor's coordinates to get partial gradients of these quantities. Since moving the central atom in one direction is equivalent to moving all the neighbors in the opposite direction, we can compute the gradient with respect to the central atom by summing the partial gradients with respect to its neighbors' positions. In particular, for the exposed surface area of atom i, $S_i^{\exp}$, if the central atom has M_i neighbors, then

$$\frac{\partial S_i^{\exp}}{\partial \vec{x}_i} = -\sum_{k=1}^{M_i} \frac{\partial S_i^{\exp}}{\partial \vec{x}_k} \tag{67}$$

The derivation of the partial gradients is tedious but straightforward and has been presented in [73]. The total gradient is the sum of the partial gradient with respect to the central atom and the partial gradient of the neighbors with respect to the central atom:

$$\nabla_i S_i^{\exp} = \frac{\partial S_i^{\exp}}{\partial \vec{x}_i} + \sum_{j=1}^{M_i} \frac{\partial S_j^{\exp}}{\partial \vec{x}_i} \tag{68}$$

3.3 Topological Analysis

The analysis in sections 3.1 and 3.2 illustrates the formulas needed for the analytical calculation of areas and volumes with gradients for every atom in the fused-sphere model. This assumes that we know for every atom in the model which neighbors intersect it and in which order. In particular, the application of the Gauss-Bonnet theorem implies that we know which neighboring atoms create the GB-arcs on the surface of the central atom and also the ordering of the GB-arcs as they form the closed GB-paths. The

topologies that may arise in a molecular simulation can vary greatly and can be extremely complex (see Figure 8). It is imperative that we have an algorithm that can deal with all possible topologies and also be computationally very efficient. There are two problems that we have to solve: First, we need to identify which neighbors that intersect the central atom are truly contributing to the exposed area and volume of the atom. Second, we need to order the true neighbors as they form the GB-paths on the surface of the central atom.

3.3.1 Intersection of Half-Spaces (IHS)

In order to identify which neighbors intersect the central atom in an efficient way we divide the simulation space into cells (Figure 20). We assign each atom into the cell that contains it and then search which atoms intersect the central atom only for the cell it belongs to and the 26 neighboring cells.

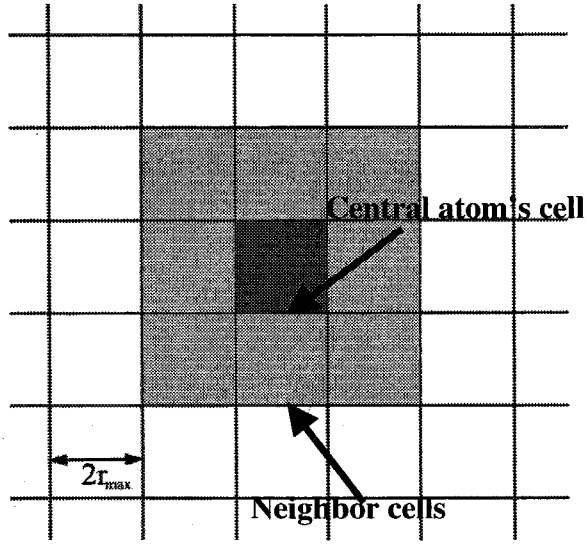

Figure 20. Partition of the simulation space into cells. For each atom we search the cell the atom belongs to (dark gray) and the 26 neighboring cells (light gray.)

If the sphere of radius $r_i + 2r_{\max}$ centered on the central atom of radius r_i and a neighboring cell do not intersect, no atom in that cell can intersect the central atom. Thus, only if the neighboring cell and the $r_i + 2r_{\max}$ sphere intersect we search for intersecting atoms in that cell. In order for the search to be complete when we restrict ourselves to searching only the nearest neighbor cells, the length of each cell has to be at least twice the maximum radius in the system, $r_{\max}$.

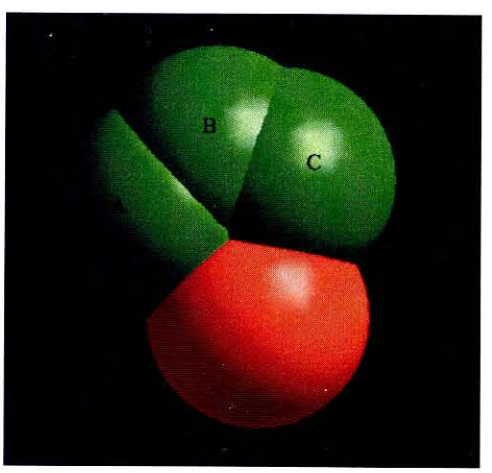

Figure 21. The central atom (red) is intersected by the neighbors A, B and C (green). Neighbor B is occluded by A and C.

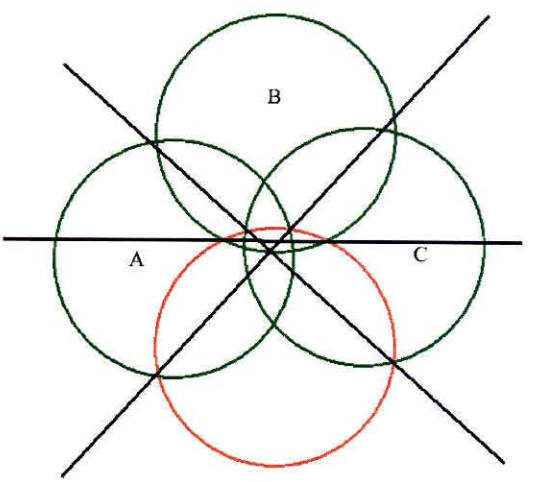

Figure 22. Same as Figure 21, also showing the intersecting planes of each neighbor.

However, finding the atoms that intersect the central atom is not enough. There can be many spheres that intersect the central sphere that do not really contribute to the exposed area and volume according to the decomposition described in section 3.1. This is because most of the intersecting spheres are actually occluded by the truly intersecting neighbors. In the example in Figure 21 and Figure 22 neighbor B is occluded by the other

two neighbors A and C and does not contribute to the area and volume calculation. In order to identify the truly intersecting neighbors we will transform the problem into a well-known problem in computational geometry, the problem of "the intersection of N half-spaces" [65], [73], [75].

Every neighboring atom that intersects the central atom defines an intersection plane that cuts through the two spheres, as shown in Figure 23. This plane divides the space into two half-spaces, H_1 and H_2 (Figure 24).

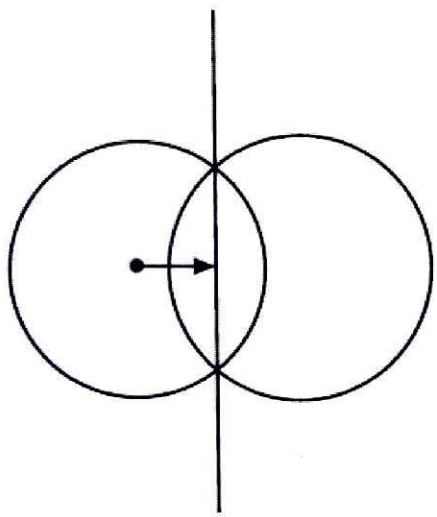

Figure 23. The intersecting plane between the central atom (left) and a neighbor.

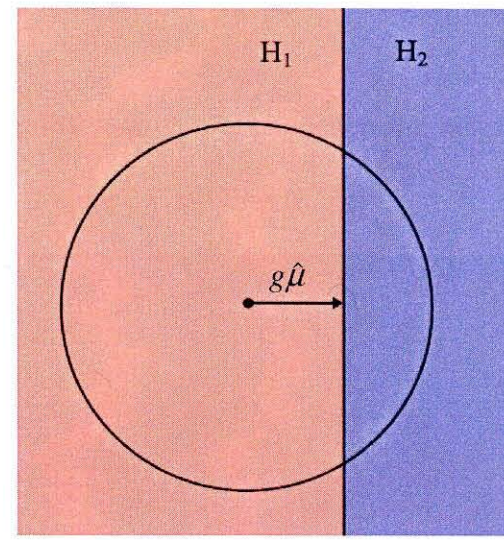

Figure 24. The two half-spaces H_1 and H_2 defined by the intersecting plane. The exposed area and excluded volume of the central atom is on H_1.

The half-spaces can be formally described as follows: A plane in space that is at distance l from the origin can be defined by a vector $\vec{p}$ that is normal to that plane and such that $|\vec{p}| = l$. All points $\vec{r}$ in space that belong to this plane obey the plane equation:

$$\vec{r} \cdot \vec{p} - |\vec{p}|^2 = 0 \tag{69}$$

The plane is the boundary between the two half-spaces it defines:

$$\begin{aligned} \vec{r}\cdot\vec{p}-\left|\vec{p}\right|^2 &< 0 \quad \text{for } \vec{r}\in H_1 \\ \vec{r}\cdot\vec{p}-\left|\vec{p}\right|^2 &> 0 \quad \text{for } \vec{r}\in H_2 \end{aligned} \tag{70}$$

According to the vector parameterization defined in section 3.2, the vector that defines the intersecting plane between atom i and neighbor k is $g_k^i\hat{\mu}_k^i$, where the unit vector $\hat{\mu}_k^i$ points towards the center of the neighbor. For the case shown in Figure 24 the solvent excluded volume and solvent accessible surface of the central atom are on the H_1 half-space that includes the origin, which is defined as

$$g_k^i\hat{\mu}_k^i\cdot\vec{r}-\left(g_k^i\right)^2\le 0 \tag{71}$$

where we also include the boundary. A slightly different case arises when the neighbor "swallows" the central atom, but not completely (Figure 25 and Figure 26).

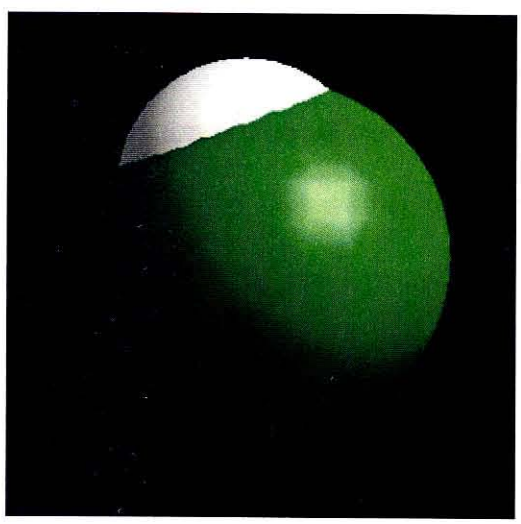

Figure 25. Example of a swallower: the neighbor (green) "swallows" the central atom (white) but not completely.

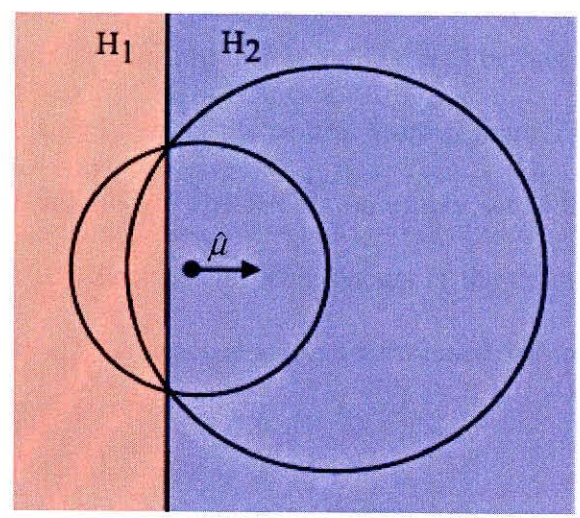

Figure 26. The half-spaces defined in the case of a swallower neighbor (right). The exposed area and excluded volume of the central atom (left) is on half-space H_1.

In such case, the intersecting plane is still defined by the vector $g_k^i \hat{\mu}_k^i$ although now g_k^i is negative. The half-space H_1 that includes the excluded volume and accessible area is the one that does not include the origin, thus

$$g_k^i \hat{\mu}_k^i \cdot \vec{r} - \left(g_k^i\right)^2 \geq 0$$
$$\text{or,} \tag{72}$$
$$-g_k^i \hat{\mu}_k^i \cdot \vec{r} + \left(g_k^i\right)^2 \leq 0$$

In general, there can be both non-swallower and swallower neighbors. The solution to the set of equations (71) and (72) is the intersection of half-spaces (IHS) and it is the region of space that satisfies all the linear constraints imposed by the intersecting planes of all the neighbors. Clearly, the IHS will be either an empty set or a convex set (infinite or finite) bounded by the planes that truly contribute to the exposed area and excluded volume. Thus, the faces of the convex polyhedron that bound the IHS belong to the intersection planes of the neighbors that truly intersect the central atom (Figure 27). By identifying the IHS we identify the true neighbors. The IHS is an empty set if the constraints posed by the intersecting planes are inconsistent. This occurs if the central atom is completely swallowed by its neighbors. The exposed area and excluded volume for such case is zero.

The problem of finding the feasible points for a set of N linear constraints has many applications in computational geometry and mathematical optimization and its efficient solution has attracted considerable attention. We will present here a simplified version of the algorithm presented in [75] that is suitable for our needs. But first we will

have to introduce the concepts of geometric duality and the convex hull (CH) of a set of points.

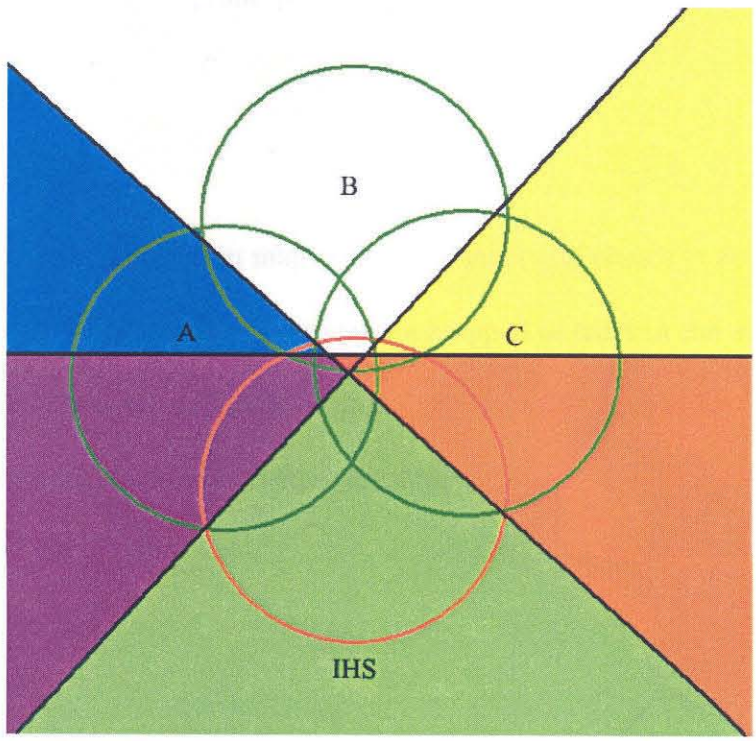

Figure 27. The IHS for the example of Figure 21. The half-spaces of neighbors A, B and C that include the central atom's exposed area and excluded volume are colored blue, red and yellow respectively. The overlap of the half-spaces is colored by the corresponding overlapping color, i.e., yellow+red=orange, blue+red=purple, blue+yellow=green. The common interior of the constraints is the IHS (green) and it is only due to the A and C half-spaces. Neighbor B is occluded.

3.3.2 Geometric Duality and the Convex Hull (CH)

As was shown in 3.3.1, a plane in space can be described by the vector $\vec{p}$ normal to it. The distance between the plane and the origin is $|\vec{p}|$. At the same time, any vector can define a point in space. In general, we define the geometric inversion in $\Re^d$ as a

point-to-point transformation of $\Re^d$ which maps a vector $\vec{p}$ applied to the origin to the vector $\vec{p}' = \vec{p}/|\vec{p}|^2$, applied to the origin [65]. Using geometric inversion we can define the dual of a plane $\vec{p}$ as the point $\vec{p}' = \vec{p}/|\vec{p}|^2$, and vice versa, the dual of a point $\vec{p}'$ as the plane $\vec{p} = \vec{p}'/|\vec{p}'|^2$.

The geometric dualization maps points to planes and planes to points. Thus, if it is applied to a convex polyhedron, its vertices are mapped to faces in the dual space and the faces are mapped to vertices. In the example in Figure 28 we see how a tetrahedron is mapped to another tetrahedron with equivalent topology.

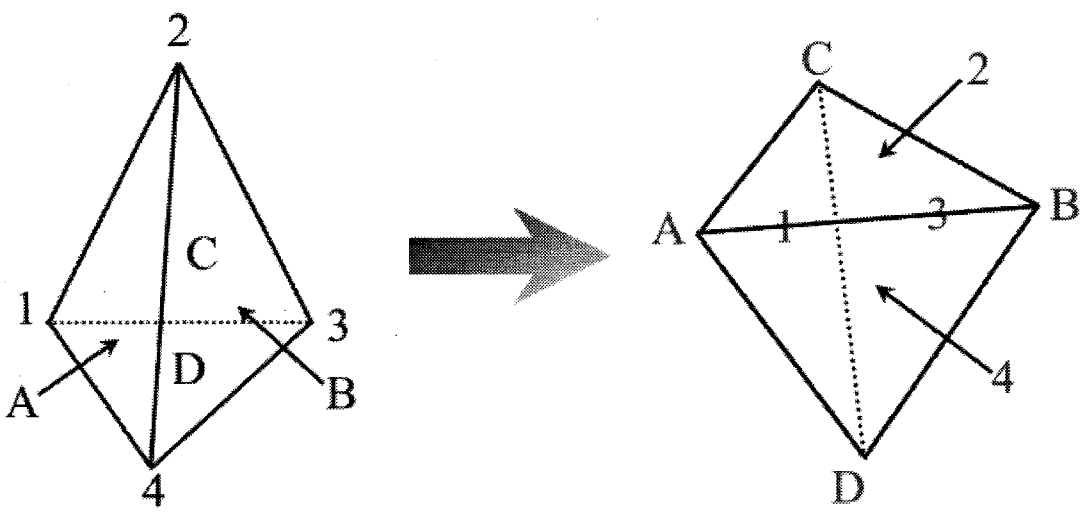

Figure 28. Geometric dualization of a tetrahedron. The vertices are mapped to faces and the faces to vertices. The topology (e.g., faces connected by a common edges) is preserved.

Vertex 1, where faces A, C and D meet, is mapped to face 1 in dual space, which is defined by the vertices A, C and D, the duals of the respective faces. Similarly, face A,

defined by vertices 1, 2 and 4 is mapped to its dual vertex A, where dual faces 1, 2 and 4 meet. Edge 1-2, which connects faces A and C is mapped to edge A-C, which connects the dual faces 1 and 2. The orientation of each face is preserved in the dual space. This topological equivalency between a convex polyhedron and its dual will prove to be crucial in our algorithm.

The convex hull (CH) of a set of points S in $\Re^d$ is the boundary of the smallest convex domain in $\Re^d$ containing S [65]. We can intuitively think of the convex hull of a set of points in space as the geometric figure that would arise if we were to tightly wrap the outside set of points with an elastic band. It is obviously a convex polyhedron and for any set of points there exists a convex hull. A two-dimensional example of a convex hull is shown in Figure 29. The CH is defined by the "most outwards" points of the set of points.

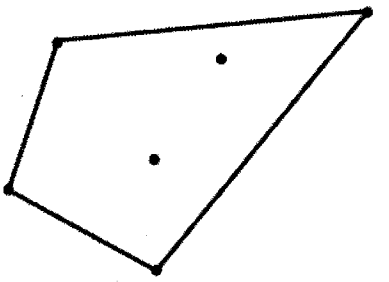

Figure 29. The convex hull of a set of points in two dimensions.

Now we have all the tools needed to solve the IHS problem. The crucial property of the IHS is that if we dualize the intersection planes, which define the IHS, to points, and construct the CH of that set of points, the dual of the CH is the IHS. In other words,

the dual of the IHS is the CH of the dual of the constraints. A proof of this theorem using projective geometry is given in [65]. This gives us the recipe for computing the IHS: Each intersecting plane k of the central atom i is a linear constraint described by the vector $g_k^i \hat{\mu}_k^i$. The dual of each intersecting plane is the point $\hat{\mu}_k^i / g_k^i$. Then we calculate the convex hull of all the dual points, including the origin (which is the center of the atom i). For each face of the CH, we find its dual point. Using the topological equivalence between the CH and its dual, as shown in Figure 28, we connect the duals of the faces of the CH. The resulting polyhedron is the IHS. This procedure is shown schematically in Figure 30 - Figure 37, for a two-dimensional example. Effectively this procedure is able to remove the occluded constraints because of the nature of the geometric inversion. This mapping brings points close to the center far away from it, and vice versa. The occluded constraints correspond to far-away points. By inverting them, we bring their duals close to the center and then the convex hull of the duals selects the ones farthest away from the center in the dual space (closest to the center in real space), thus removing the occluded constraints.

We include the center of the atoms in the set of points for which we construct the CH because it corresponds to a constraint at infinity. We call the center in the dual space the "zero point." If the IHS is open, like a convex cone instead of a convex polyhedron, as in the example in Figure 27, the center will be a vertex of the CH. After dualizing the CH, all faces that include the zero point are removed from the IHS polyhedron, thus creating the open IHS. These faces are called "zero" faces.

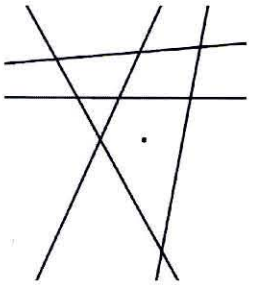

Figure 30. The linear constraints.

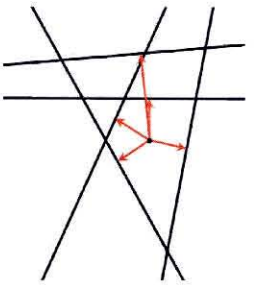

Figure 31. The normals to the constraint planes.

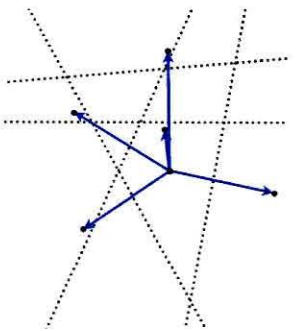

Figure 32. Dual points of the constraint planes.

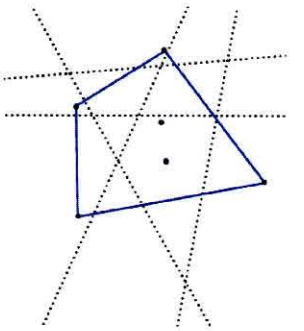

Figure 33. The convex hull of the dual points.

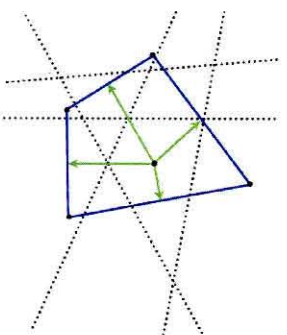

Figure 34. The normal vectors to the faces of the CH.

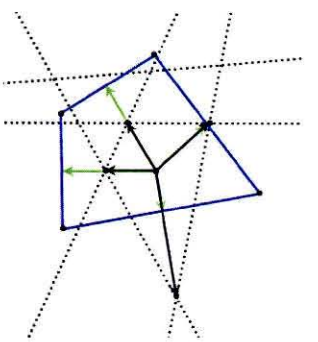

Figure 35. The dual points of the faces of the CH.

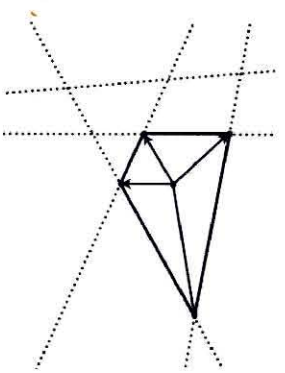

Figure 36. Connecting the dual points of the CH.

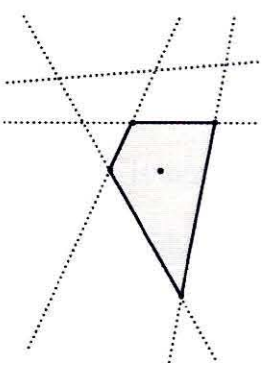

Figure 37. The IHS.

A key property for this procedure is the existence of the coordinate center inside the IHS. In order to dualize the intersection planes, we must have a point with respect to which we do the geometric inversion. This point was taken as the center of the atom, because it is usually inside the IHS. However, if at least one of the neighbors is a swallower, the center of the atom is no longer in the IHS, as shown in Figure 25 and Figure 26. Thus, before we can apply the aforementioned procedure, we need to find an interior point of the IHS. This is a non-trivial issue since we are seeking for a point inside a region for which we do not yet know its boundaries, and finding these boundaries is the actual problem we have to solve and we need the interior point for. We can find though a vertex of the IHS, without knowing anything else but the constraints, by using linear programming, as described in section 3.3.3.

An additional complication that arises when there are swallower neighbors is the existence of both a common interior (the IHS) as well as a common exterior, the region of space for which the constraints (71) and (72) are inverted (if the constraints are not inconsistent). As described above, the duals of the faces of the CH correspond to vertices of the common interior (IHS). In the presence of swallowers, however, the duals of certain faces of the CH correspond to vertices of the common exterior. These faces have to be identified and removed from the CH in order to build correctly the IHS. The procedure then is slightly modified as follows: The zero point is not included in the construction of the CH. After the CH is calculated, we identify which faces of it are visible from the zero point (the concept of visibility of a face from a point is explained in detail in section 3.3.4). The duals of those faces are vertices of the common exterior. We

name these faces "negative" faces because if we reformulate the problem using homogeneous coordinates in four dimensions, then these vertices are points of the hyperplane $x_4 = -1$, whereas the three-dimensional space we're working on is the hyperplane $x_4 = 1$. (See Figure 38 for a two-dimensional example and Figure 39 for the projection in three dimensions and [65] for details on homogeneous coordinates and the interpretation of the problem in higher dimensions). Negative faces actually correspond to constraints at infinity, much like the zero point corresponds to a constraint at infinity, thus creating an open IHS.

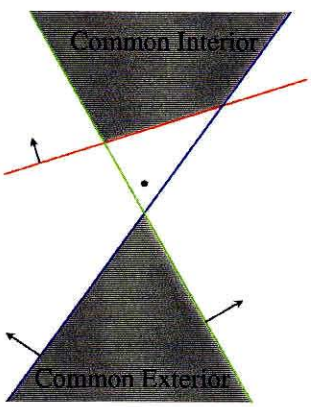

Figure 38. Constraints and their half-spaces, in 2D.

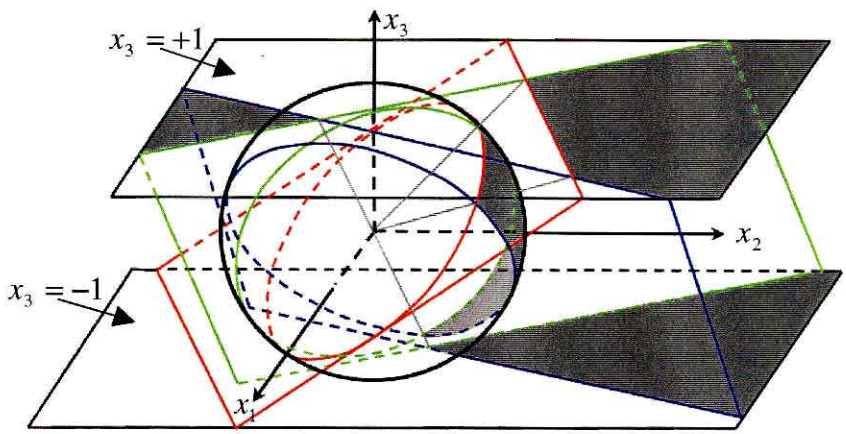

Figure 39. Projection of the 2D problem in 3D. The constraint lines on the hyperplane $x_3 = +1$ become planes that pass through the 3D origin. The problem is mapped on a 3D unit sphere. (Figure adapted from [65].)

3.3.3 Linear Programming

Linear programming (LP) is a fundamental optimization problem that has found applications in various fields, from computer science to economics to business

administration, and it has attracted considerable attention in the literature. In its general form it is formulated as follows [76]: For N independent variables $x_1, x_2, ..., x_N$ we seek a vector in the N dimensional space that maximizes the linear function $f(x_1, x_2, ..., x_N) = \sum_{i=1}^{N} c_i x_i$ (called the objective function) subject to M linear constraints, $\sum_{j=1}^{N} a_j^i x_j \leq a_{N+1}^i$, where $i = 1, ..., M$ (some of the inequalities could also be equalities). Any vector that satisfies all the constraints is called a feasible vector. The feasible vector that optimizes the objective function is called the optimal feasible vector. If the independent variables are restricted to be positive, the LP problem is said to be in its normal form.

An optimal feasible vector can fail to exist for two possible reasons: the constraints are incompatible or there is a direction in the N - dimensional space for which one or more of the variables can be taken to infinity while still satisfying all the constraints, giving an unbounded value for the objective function. The linear constraints effectively reduce the search space into a convex polyhedron, which could be open. If there is an optimal feasible vector, since the objective function is linear, it will have to be a vertex of that polyhedron, which is the point at which some N of the constraints meet. Thus, we can apply LP on the three-dimensional linear constraints (71) and (72), the half-space inequalities, with an arbitrary objective function, to find a vertex of the IHS. However, a vertex of the IHS is inappropriate to use as the coordinate center for dualization, because the planes that meet at that vertex will be dualized to infinity. We must find a truly interior point of the IHS instead.

In order to find a point in the interior of the IHS we have to "shrink" the constraint planes by a positive constant ε, towards the half-spaces of interest. Then, if we apply LP for the shrinked constraints the optimal feasible vector will be a vertex of the shrinked constrained polyhedron and thus an interior point of the IHS (Figure 40).

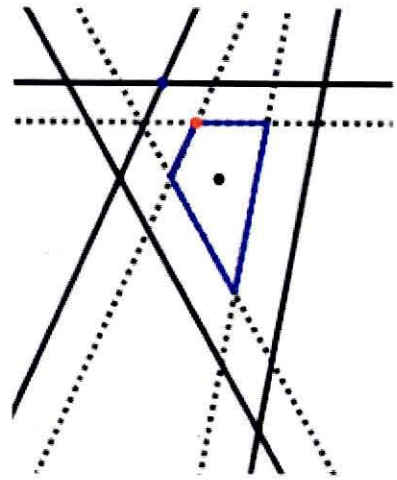

Figure 40. The vertex of the IHS of the shrinked constraints (dotted lines) is an interior point of the original IHS.

The constraint planes are described by equations (71) and (72), which have the form

$$\begin{aligned} \vec{p}\cdot\vec{r} - |\vec{p}|^2 &\le 0 \\ \vec{p}\cdot\vec{r} - |\vec{p}|^2 &\ge 0 \end{aligned} \tag{73}$$

respectively, where $\vec{p} = g\hat{\mu}$ is the vector that defines each plane. The constraints can be rewritten as

$$\begin{aligned} \hat{p}\cdot\vec{r} - |\vec{p}| &\le 0 \\ \hat{p}\cdot\vec{r} - |\vec{p}| &\ge 0 \end{aligned} \tag{74}$$

The first equation corresponds to the half-space that includes the origin and the second that excludes the origin, which is the case of a swallower neighbor as shown in Figure 26. We want to translate the constraints towards the half-spaces of interest by a constant positive value ε, so the constraint inequalities become

$$\begin{aligned} \hat{p}\cdot\vec{r} - \left(|\vec{p}| - \varepsilon\right) \le 0 \\ \hat{p}\cdot\vec{r} - \left(|\vec{p}| + \varepsilon\right) \ge 0 \end{aligned} \tag{75}$$

as is shown in Figure 41 and Figure 42.

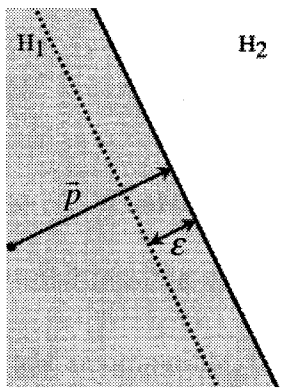

Figure 41. Translating the constraint plane $\vec{p}$ by ε, towards the half-space H_1, for the case of a non-swallower neighbor.

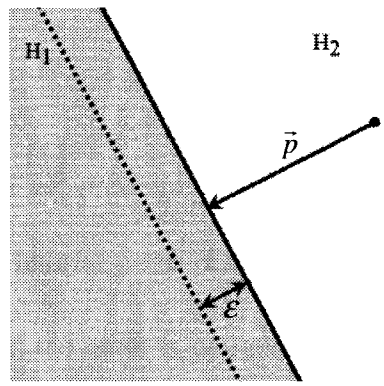

Figure 42. Translating the constraint plane $\vec{p}$ by ε, towards the half-space H_1, for the case of a swallower neighbor.

The value of ε describes by how much the constraints are shifted. We want an optimal value for ε to ensure we find a point "way in" the interior of the IHS. We can achieve this by setting it as an independent variable and optimizing its value using LP. So, instead of solving the three-dimensional LP problem of the M constraints (71) and (72), where

M is the total number of neighbors intersecting the central atom, we solve the LP problem in the four-dimensional space $(\vec{r},\varepsilon)$ that consists of the following constraints:

$$\begin{aligned} g_k^i \hat{\mu}_k^i \cdot \vec{r} + g_k^i \varepsilon - \left(g_k^i\right)^2 \le 0, \quad \text{if } g_k^i > 0 \\ -g_k^i \hat{\mu}_k^i \cdot \vec{r} - g_k^i \varepsilon + \left(g_k^i\right)^2 \le 0, \quad \text{if } g_k^i < 0 \\ -\varepsilon \le 0 \end{aligned} \tag{76}$$

which result from equation (75) by plugging $\vec{p} = g\hat{\mu}$. We want to optimize the value of ε to ensure we have an optimal interior point, so the objective function to be maximized is

$$f(\vec{r},\varepsilon) = \varepsilon \tag{77}$$

If there is a solution to the four-dimensional LP problem of equations (76) and (77), the optimal feasible vector $(\vec{r}_{opt},\varepsilon_{opt})$ gives the interior point of the IHS, $\vec{r}_{opt}$. If the constraints are incompatible, there is no solution and the atom is completely swallowed by its neighbors, so the exposed area and excluded volume are zero. The case of an unbounded solution has to be considered with care, since it can still provide us with an interior point as long as we can identify the direction in the N dimensional space that gives an unbounded value for the objective function. The standard algorithms used to solve LP problems [76], [77], assume the problem in its normal form and cannot deal with unbounded solutions. Instead, we implemented the algorithm introduced by Seidel [78], which can deal effectively with these issues.

In Seidel's algorithm, the optimum vertex is determined by a recursive procedure. Initially, a constraint is picked in random, and a guess for the optimal vertex is made

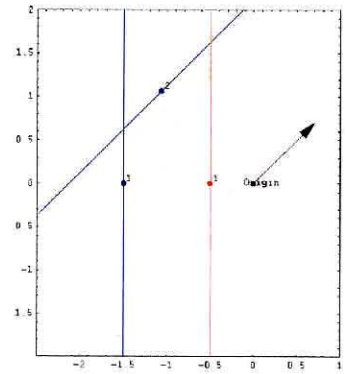

Figure 43. The constraints and the objective function.

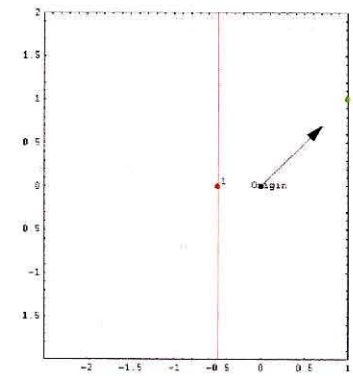

Figure 44. Choosing a random point on the direction of the objective function.

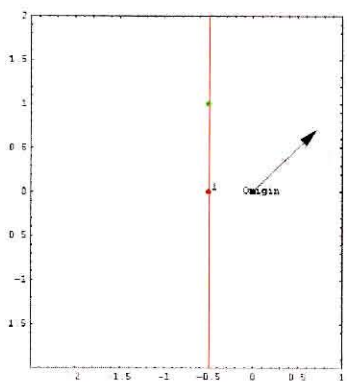

Figure 45. Projecting the point to a randomly picked constraint.

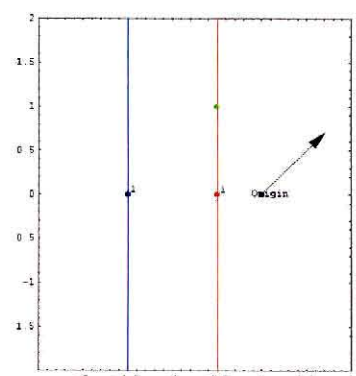

Figure 46. Adding another constraint. The point satisfies this constraint.

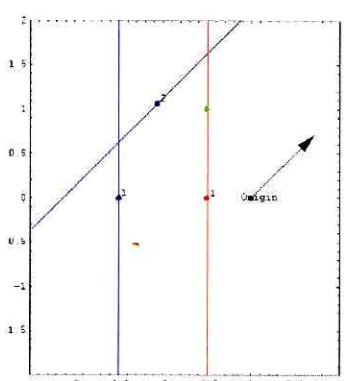

Figure 47. Adding the last constraint. Solving the problem in one dimension (on the constraint red-1.)

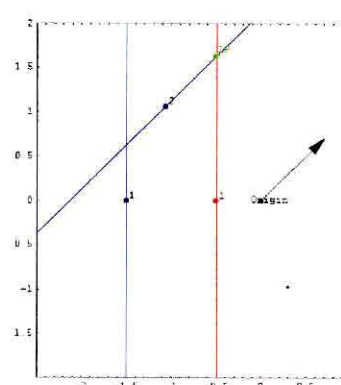

Figure 48. Optimizing the point with respect to the last constraint in one dimension and "lifting" the solution on the two-dimensional space.

along the direction of the objective function. Then we keep adding the rest of the constraints and check if the trial vertex satisfies them. If there is a constraint that is not satisfied by the trial vertex, we project the vertex and all the other constraints on the hyperplane of that constraint and recursively solve the problem of $M-1$ constraints in $N-1$ dimensions. The recursion will keep going into lower dimensions until we reach a one-dimensional problem whose solution is trivial. That solution is then "lifted" onto the higher dimensions. Additional constraints are added dynamically in the algorithm that bound the optimal vertex in case of an unbounded solution. The pseudocode for this algorithm is described in detail in [78]. A two-dimensional example of this procedure is shown in Figure 43-Figure 48. There, the lines in red are the boundaries of half-spaces that do not include the origin and the lines in blue bound the half-spaces that include the origin.

3.3.4 Construction of the Convex Hull

As was described in 3.3.2, after we dualize the intersection planes to points, we calculate the convex hull of the duals in order to eliminate the redundant constraints. If there are no swallowers, the dualization is with respect to the center of the atom; otherwise we apply LP to find a point in the interior of the IHS and use this point as the center. The calculation of the convex hull is done by means of a randomized incremental algorithm that has optimal expected performance [79].

We start the calculation by choosing four points in random to form a tetrahedron. This is the starting point for the construction of the CH. At every consecutive step we

will incrementally add and remove faces to this polyhedron until we arrive at the CH. To do that, we pick a point in random from the remaining set of points and check all the faces of the current polyhedron to determine which faces are "visible" from that point. The faces of the polyhedron are orientated in a CCW fashion, looking from the outside. This orientation defines a vector normal to the face pointing outwards. A face is visible from a point if the point is on the half-space that the normal vector is pointing towards. Mathematically, if the vertices of the face are the vectors $\vec{a}$, $\vec{b}$ and $\vec{c}$, and the point is $\vec{d}$, then the sign of the determinant D determines if the face is visible, where

$$D = \begin{vmatrix} a_x - d_x & a_y - d_y & a_z = d_z \\ b_x - d_x & b_y - d_y & b_z - d_z \\ c_x - d_x & c_y - d_y & c_z - d_z \end{vmatrix} \tag{78}$$

If $D < 0$ then the face is visible, if $D > 0$ the face is not visible and if $D = 0$ then the face and the point are coplanar. This is because the signed volume of the tetrahedron is $D/6$ and the sign has to do with the orientation of the triplet $\vec{a} - \vec{d}$, $\vec{b} - \vec{d}$, $\vec{c} - \vec{d}$, centered at the vertex $\vec{d}$. After we determine the visible faces of the current polyhedron for the picked point, we delete the visible faces that share an edge, along with the common edge. If visible faces share a common vertex it is also deleted. We then create new edges from the point to the undeleted vertices of the deleted faces and form new faces, making sure we preserve the CCW orientation. The polyhedron that arises with the new faces includes the deleted vertices in its interior. The procedure is continued by picking another point in random from the remaining set and adding and deleting faces to the polyhedron after the visibility checks. At the end of each addition the polyhedron created is the convex hull of the subset of points that have been utilized up to that step.

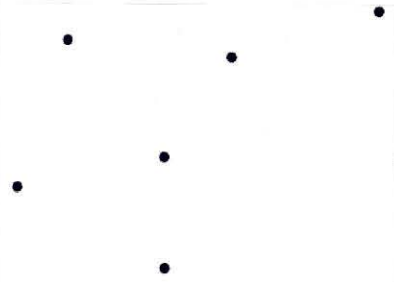

Figure 49. Set of points.

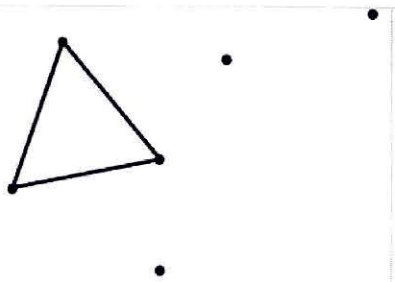

Figure 50. Initial simplex.

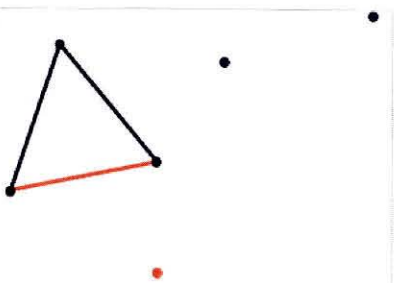

Figure 51. Visibility check.

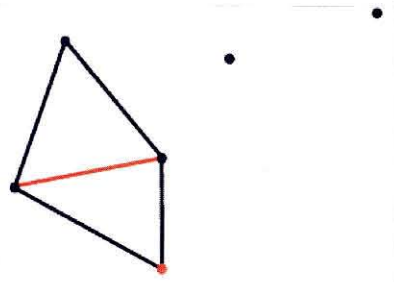

Figure 52. Add new faces.

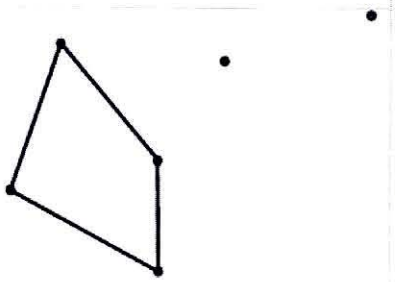

Figure 53. Remove visible faces.

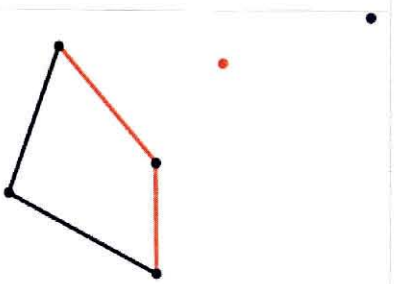

Figure 54. Visibility check.

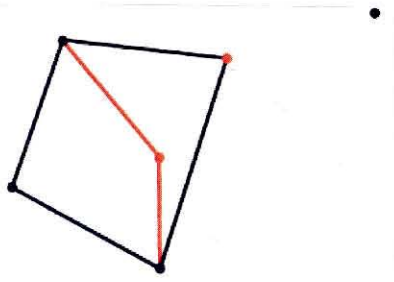

Figure 55. Add new faces.

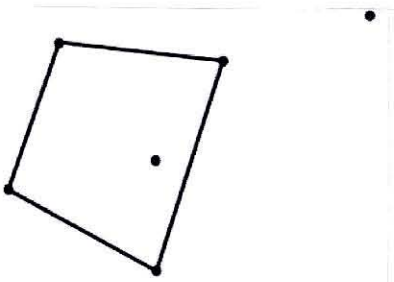

Figure 56. Remove visible faces.

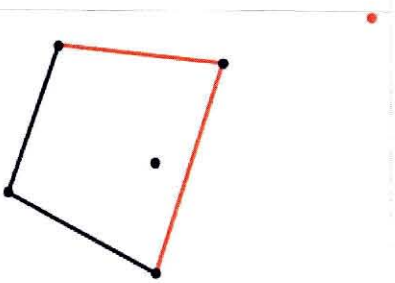

Figure 57. Visibility check.

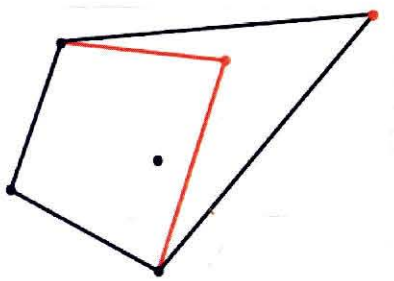

Figure 58. Add new faces.

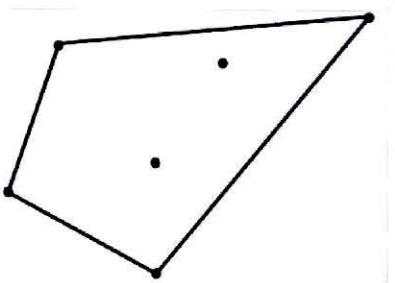

Figure 59. Convex hull.

When there are no more points remaining, the algorithm guarantees the final polyhedron is the CH of the initial set of points. This procedure is shown schematically in Figure 49-Figure 59 for a two-dimensional case. The visible faces to be removed at every step are marked with red.

It is clear from the above that the actual speed of the algorithm depends a lot on the input set of points and the number of points eliminated at every step. The worst case is if all points are on the surface of a sphere, then all points are included in the CH and we must examine every point. Also, due to the random nature of the algorithm, the performance will depend on order of the points picked. The further they are at the beginning of the process, the more points are excluded and thus fewer points have to be examined. This implies that if we somehow bias the selection process towards points further away, we should increase the computational speed. This is the idea behind the QuickHull variation of the randomized incremental algorithm, which was proposed in [80] and was utilized in this implementation.

The only change on the incremental algorithm because of QuickHull is that, at the beginning, we loop over all the points and for each point we find the first face that is visible to it. For each face we create an "outside" list of points that it is visible to, and the list is sorted according to the distance of the point from the face. This way we partition the set of points to the outside lists of the faces. Then, instead of selecting in random a point for the incremental algorithm, we select the point with the largest distance from its face. At each successive step, after some faces are deleted and others created, the points

in the outside lists of the deleted faces are repartitioned to the outside lists of the new faces. In addition, the initial tetrahedron is created by four points that have maximum coordinates. These additions to the QuickHull algorithm make the CH construction much more efficient.

3.3.5 Determining the GB-paths

The procedure described in sections 3.3.1-3.3.4 allows us to determine which neighbors are truly intersecting the central atom, or equivalently, which neighbors form GB-arcs on the surface of the central atom. The next task is to identify the exact topology of the true neighbors on the surface of the atom. If the COIs of these neighbors are intersecting each other then they intersect at the GB-points (see Figure 18 and vectors $\vec{P}_{kj}^{i}$, $\vec{Q}_{kl}^{i}$ in Figure 19). Each true neighbor contributes at least one GB-arc (except in the trivial case of an isolated neighbor, as in Figure 9) and two neighbors that intersect each other may contribute at most two GB-points. Each GB-arc is a section of the COI of a particular neighbor, bounded by two GB-points at the beginning and the end. The orientation of the GB-arcs is CCW, looking from top, according to the convention we set in section 3.2. The goal then is to identify which neighbors' COIs intersect each other thus forming a GB-point, group the GB-points into subsets that belong to each GB-path and then order the GB-points of each group as they form the GB-path in a CCW fashion. When the GB-paths have been determined we can use equation (57) to calculate the exposed area and equation (50) for the excluded volume. The key property that will allow us to solve this problem is the fact that the edges of the IHS pierce the surface of the atom

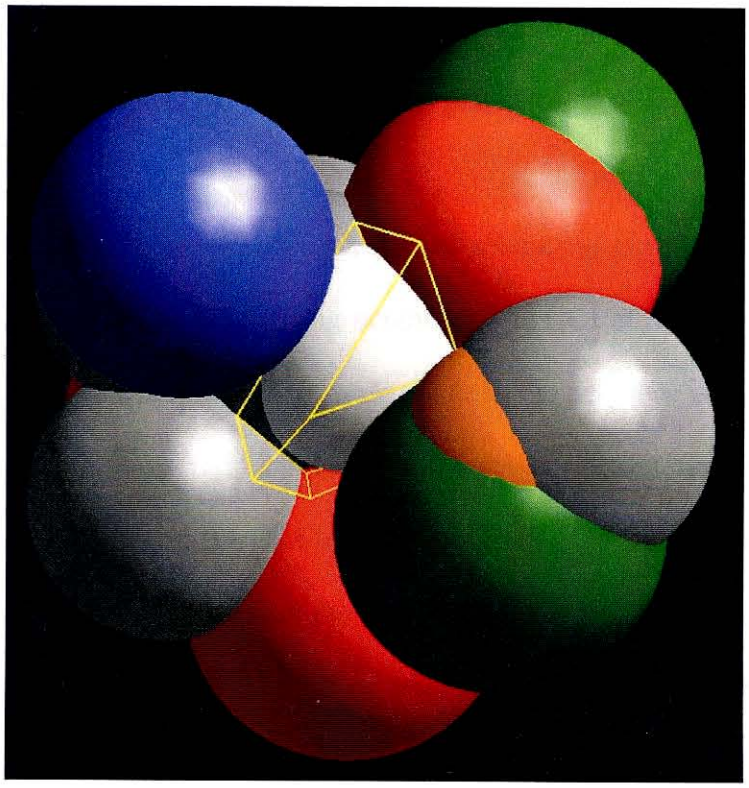

Figure 60. The IHS polyhedron formed by the neighbors of the central atom (white) for the example in Figure 8.

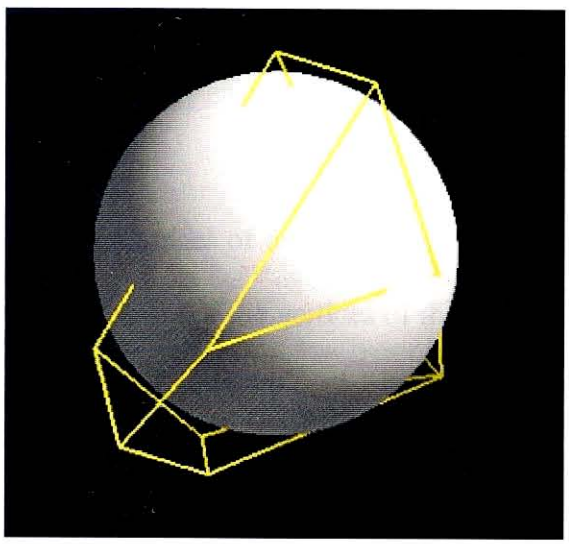

Figure 61. The IHS polyhedron of Figure 60 as it cuts through the central atom.

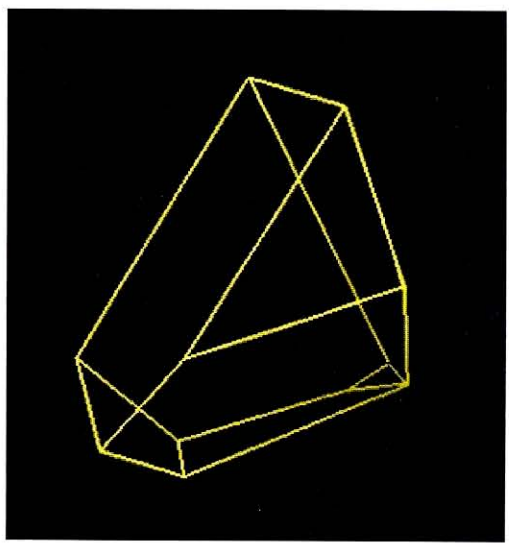

Figure 62. The IHS polyhedron of Figure 60.

at exactly the GB-points. This is clearly shown in Figure 60, Figure 61 and Figure 62, where the IHS polyhedron for the central atom in the example of Figure 8 was calculated.

Each face of the IHS is a polygon-shaped section of the planar boundary of the half-space imposed by the corresponding neighbor. The COI of that neighbor lies on the plane of the face. If the whole COI lies inside the face, the neighbor is isolated (Figure 9). If the whole COI lies outside the face (but on the boundary plane) then the neighbor does not contribute any GB-arcs. If the COI lies partially inside and outside the face, then each point on the COI that is on an edge of the IHS is a GB point (see Figure 19). An edge of the IHS is common to two of its faces. If an edge pierces the central atom, then the COIs of the corresponding atoms intersect at the GB points. In the case that all the vertices of the IHS are completely buried inside the atom, no edges intersect the surface of the atom and thus there are no GB-points. In such case, the atom has no exposed surface because it is buried under all its neighbors. However, the volume is not zero and it is the volume of the IHS.

The edges of the IHS pierce the central atom zero, one or two times. This depends on the location of the vertices of the IHS that define the edges, with respect to the surface of the central atom. If both vertices are buried inside the atom then the edge does not intersect the surface of the central atom. If both vertices are exposed the edge can intersect in two points or none at all. This will have to be determined explicitly by the following sphere-line intersection test: If a line goes through a point $\vec{r}_0$ and is parallel to the direction $\hat{u}$ then every point on the line obeys the line equation $\vec{r} = \vec{r}_0 + \lambda\hat{u}$, for any

real value of λ. The unit vector $\hat{u}$ is determined by the positions of the two vertices. All points on a sphere of radius R, centered at point $\vec{r}_C$ obey the equation $|\vec{r} - \vec{r}_C| = R$. The system of the two equations has a solution if the quantity $\Delta/4 = (\vec{a} \cdot \hat{u})^2 - \left(|\vec{a}|^2 - R^2\right)$ is positive, where $\vec{a} = \vec{r}_0 - \vec{r}_C$. Finally, if one vertex of the edge is buried and the other exposed then the edge pierces the atom at exactly one point. (See Figure 63-Figure 66.)

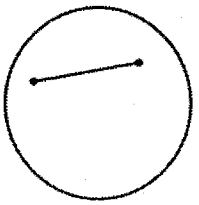

Figure 63. Buried-buried edge.

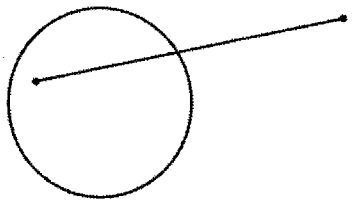

Figure 64. Buried-exposed edge.

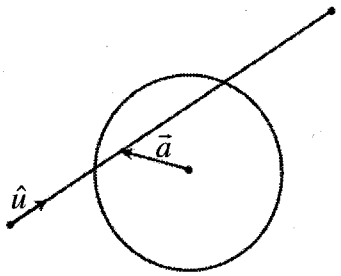

Figure 65. Exposed-exposed intersecting edge.

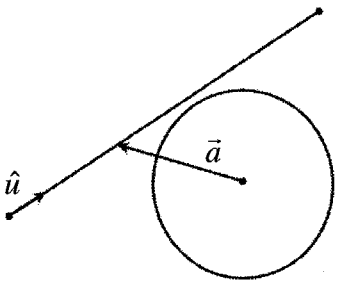

Figure 66. Exposed-exposed non-intersecting edge.

The vertices of the IHS can give us even more information. By the very nature of the IHS, a vertex of it is the point in space where three boundary planes meet. If a vertex is buried then the three neighbors that correspond to each plane are connected on the surface of the atom. The three edges that emerge from that vertex will necessarily pierce the central atom once each, creating three GB-points (see Figure 67 and Figure 68). If the vertex is exposed then the neighbors are not connected and the edges pierce twice,

creating six GB-points (see Figure 69 and Figure 70). We call these edges "double piercers."

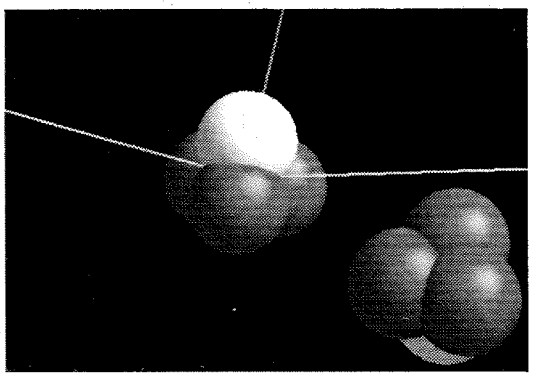

Figure 67. A buried IHS vertex corresponds to three connected neighbors and three GB-points.

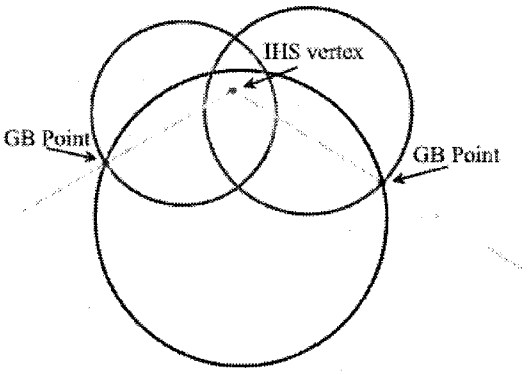

Figure 68. Two-dimensional representation of a buried IHS vertex.

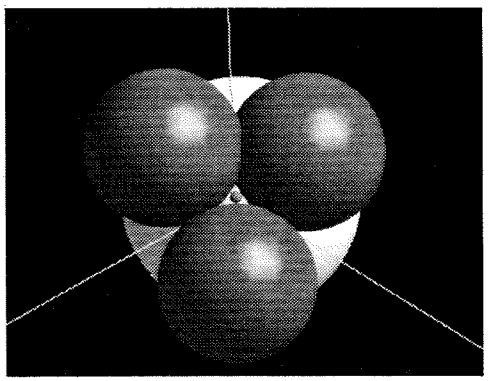

Figure 69. An exposed IHS vertex corresponds to three disjoint neighbors and three GB-points.

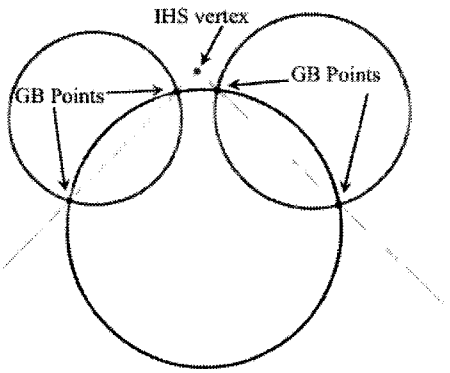

Figure 70. Two-dimensional representation of an exposed IHS vertex.

In the case of an open IHS, certain vertices are at infinity. These vertices are the duals of the zero faces of the CH in the case that there are no swallowers. If there are swallowers there are no zero faces since we do not include the center, but there are negative faces, as explained in section 3.3.2. The duals of those faces are taken to be at infinity, in order to create the true IHS, and clearly, they are exposed vertices (outside the

central atom). The edges that have such vertices will fall either in the exposed-exposed or exposed-buried category, depending on the properties of the other vertex. The analysis for these cases is the same as before, with the exception that the direction vector $\hat{u}$ of the edge (for the intersection check) is taken towards the open face of the IHS, which in the case of a negative face is opposite to the dual of the face.

Using the above ideas, we can generate the list of GB-points that are on the surface of the central atom. After the IHS is constructed, we loop over each edge and analyze its vertices to recognize how many GB-points it creates. Each edge is connecting two neighbors. The GB-points are then assigned to one of the two neighbors, according to the CCW orientation that we have chosen. For example, in Figure 19, the GB-point $\vec{P}_{kj}^{i}$ would be assigned to neighbor j and the GB-point $\vec{Q}_{kl}^{i}$ to neighbor k. This way we create a list of connectivities between the true neighbors on the surface of the atom. The

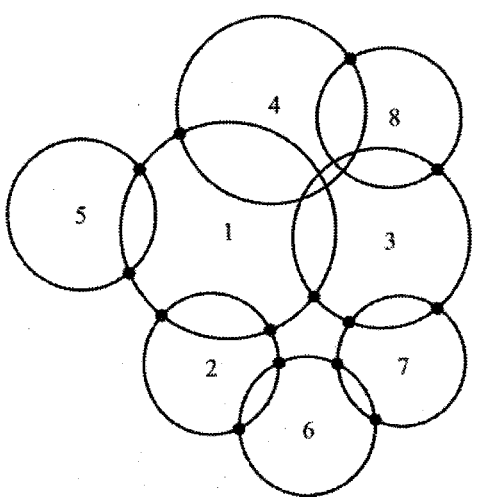

Figure 71. Example of topology of neighbors on the central atom.

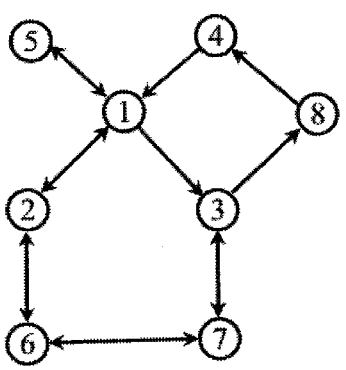

Figure 72. The connectivity graph for the example of Figure 71. The oriented edges correspond to GB-points.

1: 2 3 5
2: 6 1
3: 7 8
4: 1
5: 1
6: 2 7
7: 3 6
8: 4

Figure 73. The connectivity table for the example of Figure 71.

connectivity list is an oriented graph, where each oriented edge is a GB-point and each node is a neighbor. In the example of Figure 71 the COIs of the neighbors on the surface of an atom are shown unfolded onto a plane. The resulting connectivity graph and connectivity table of the GB-points is shown in Figure 72 and Figure 73.

It is possible that the connectivity graph can have more than one connected component. Each connected component corresponds to a disconnected piece of the buried surface of the atom, for which the Euler-Poincaré characteristic is $2-n$ where n is the number of GB-paths on that piece, as was explained in section 3.2. In order to identify the connected components of the connectivity graph, we use the Depth-First-Search (DFS) algorithm [81], a recursive graph-searching algorithm. After finding the connected components, we divide the graph to the respective sub-graphs. We then need to sort the vertices of each sub-graph, thus ordering the GB-points as they form the GB-paths. The GB-paths correspond to cycles of the connectivity graph. In the example of Figure 71 there are two GB-paths. However, the related connectivity graph of Figure 72 has more than two cycles. The reason for this inconsistency is that as we traverse the graph, there can be more than one option to select the next node, as shown in Figure 74.

In this example, we start traversing the connectivity graph from node 5, so the next node can only be node 1. However, from that node we have three possibilities for the next step, nodes 2, 3 and 5, as is shown in Figure 72 and Figure 74, because there are 3 GB-points on the COI of neighbor 1. To pick the right one, we use the parameterization shown in Figure 19. The tangent vector on the COI $\hat{n}_j^{ik}$, at the GB-point $\vec{P}_{kj}^{\,i}$ from

neighbors $j-k$, points to the direction of the GB-arc for that GB-point. The next GB-point has to be the end point of that GB-arc and thus the first point we encounter as we traverse the COI in a CCW fashion. Hence, the next GB-point has to be the "left-most" point relative to the direction $\hat{n}_j^{ik}$, which is easily determined by the dot product of the direction vector to the vector that connects the current GB-point to the candidate next GB-points, as is shown in Figure 74.

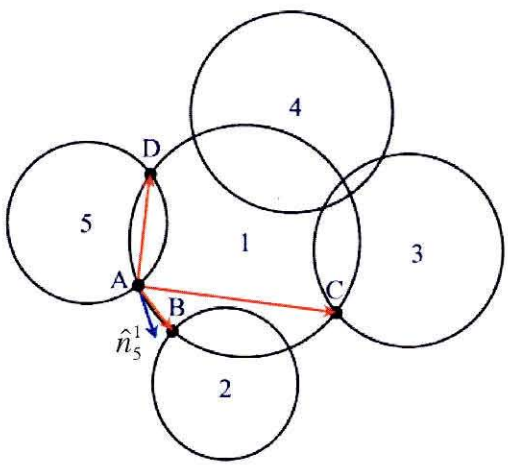

Figure 74. Traversing the connectivity graph of Figure 71: starting from the GB-point A on neighbor 5, the next GB-point has to be on neighbor 1. Out of the three possibilities B, C, D, the GB-point B is the correct choice.

The determination of the cycles (the GB-paths), for each connected component of the connectivity graph, proceeds as follows: we pick a node in random, and pick an edge for that node in random. This node is the head of the current GB-path. The next node is chosen according to the "left-most" criterion described above. When the next node to be chosen is the head, we form a cycle, which is the GB-path. If there are oriented edges in the graph that have not been traversed, we pick one in random and continue this

procedure until all edges have been traversed. When we have finished this analysis for all connected components, we have determined all the GB-paths for that atom. (See Figure 75 and Figure 76 for the GB-paths and cycles of the example of Figure 71.)

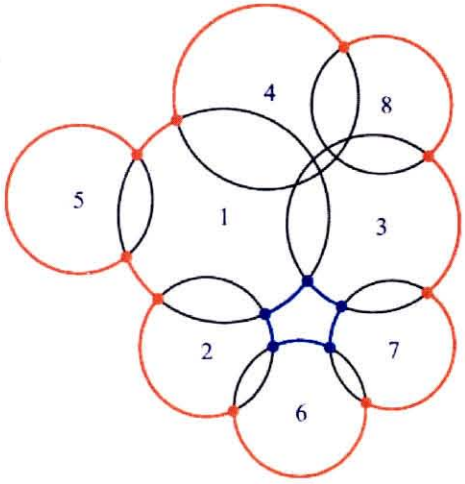

Figure 75. The GB-paths for the example of Figure 71.

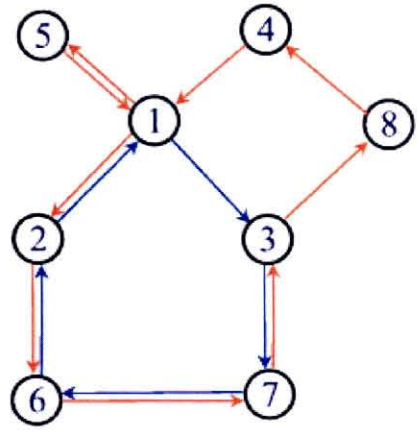

Figure 76. The selected cycles (GB-paths) of the connectivity graph for the example of Figure 71. The two GB-paths are: 5-1-2-6-7-3-8-4-1-5 and 1-3-7-6-2-1.

With the topology of the true neighbors on the surface of the central atom determined, we are free to proceed with the area and calculation as shown in section 3.3. Also, the knowledge of the IHS and the GB-paths allow us to understand better how the planar sections of the building blocks are formed, as shown in Figure 77. The planar sections are bounded by arc-segments and line segments, as was explained in Figure 17. The arc-segments are the GB-arcs and the line-segments are parts of the edges of the IHS. The vertices of the line-segments are either buried vertices of the IHS or GB-points. The identification of these vertices is necessary for the calculation of the volumes using equations (50) and (51). In particular, the knowledge of which neighbors intersect to

create the buried IHS vertices of the GB-points is needed for the calculation of the length of the line-segments, t_μ, in equation (51). The calculation of the IHS and its analysis, as explained above, provides this topological information.

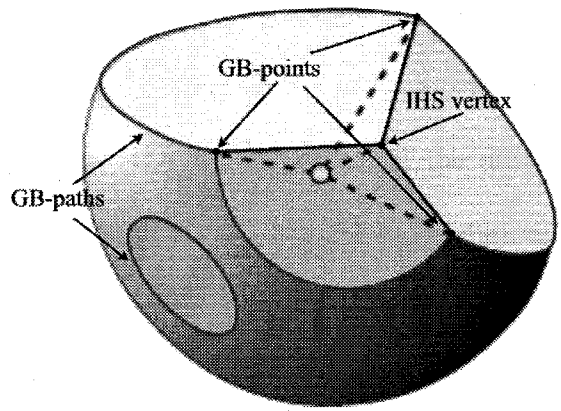

Figure 77. Relation of the planar sections of the building block of Figure 15 with the IHS and the GB-paths.

3.4 Implementation of the Geometric Algorithms

3.4.1 Robustness

The geometric algorithms presented in sections 3.1-3.3 were implemented in a computer program in the C programming language [82]. It is important for the implementation to be robust and efficient in order for it to be used in large molecular systems of complicated topology. The finite precision arithmetic used in computers can cause round-off errors that may lead to erroneous results, produce numerical infinities or even cause the program to crash. This is because of the dualization procedure described in section 3.3.2. If the distance of a plane from the center, g, is too small (the plane is

very close to the center), then the dual point of that plane will be very far away from the center in the dual space, and vice versa. Also, if four atoms come very close to intersecting at the same point in space, some GB-arcs can be very close to zero length and this can cause very small faces of the CH and numerical instabilities for the gradient calculation (see [83] for an analysis of problematic atom topologies). The geometric predicate used to calculate the CH determines the visibility of a face from a point and it can give wrong answers due to round-off errors in the aforementioned cases. If the construction of the CH fails, then the result of the calculation can produce unphysical results, like negative surface area or volume, or infinite gradients. In molecular dynamics where the atoms are propagated in space by infinitesimal distances at every step, it is inevitable that we will come across such cases in the course of a long simulation for a large system. It is crucial that we can deal with these problems effectively.

These numerical precision problems were already known in the computational geometry community. The calculation of geometric constructs (convex hulls, Voronoi diagrams, Delaunay triangulations, etc.) faces similar issues due to finite precision. The solutions proposed fall into two categories: perturbation schemes and arbitrary precision arithmetic techniques. In perturbation schemes, the results are checked for unphysical conditions and if errors are detected the spheres in the fused-sphere model are perturbed about their centers by a very small distance that overcomes the degeneracies and precision problems [84], [85], [86]. Arbitrary precision techniques attempt to create geometric predicates that produce results of arbitrary precision on finite precision machines. These techniques are conceptually better but computationally inefficient. New

adaptive methods, however, promise efficient calculations for the geometric predicates [87], [88]. In this implementation we used an adaptive arbitrary precision software library for the calculation of the geometric predicates, which is in the public domain [88].

Finally, another topology that can produce numerical instability is coplanar (or almost coplanar) molecules, like benzene. The problems are similar in nature as described above but the solution is simpler: if the centers of the atoms are almost coplanar, instead of calculating the CH in three dimensions we project the center of every atom on a plane and solve the two-dimensional CH problem.

3.4.2 Scaling and Performance

For a molecular system of N atoms we have to perform N calculations for the area and volume of each atom. Each calculation in turn involves the calculation of the IHS and its analysis. The time for the calculation of the IHS depends on the number of neighbor atoms, M, that intersect each atom. The calculation of M half-spaces scales as $O(M \log M)$ [65], thus the calculation of the whole system takes time $O(NM \log M)$. It is important then to determine if and how the number of neighbors M depends on the total size of the system, N [89].

Let us define $r_{\max}$ and $r_{\min}$ the maximum and minimum radius of any sphere in the system and $\kappa = r_{\max}/r_{\min}$ their ratio. Also, if $d_{\min}$ is the minimum distance between any two atoms (physically corresponding to the minimum valence bond length), then we

define the ratio $\lambda = d_{\min}/r_{\max}$. Let M be the neighbors of any atom i, or radius r_i. Then, all M neighbors' spheres will be completely engulfed in a sphere centered at i with radius $r_i + 2r_{\max}$. For each neighbor we define a sphere of radius $\lambda r_{\min}/2$. Then these spheres do not intersect each other because their centers are at a distance smaller than $d_{\min}$:

$$2\left(\frac{\lambda}{2} r_{\min}\right) = \lambda r_{\min} = \frac{d_{\min}}{r_{\max}} r_{\min} = \frac{d_{\min}}{\kappa} < d_{\min} \tag{79}$$

since $\kappa > 1$. Hence, the sum of their volumes has to be less than the volume of the engulfing sphere:

$$M \frac{4}{3}\pi\left(\frac{\lambda}{2} r_{\min}\right)^3 \le \frac{4}{3}\pi(r_i + 2r_{\max})^3$$

$$M \le \frac{(r_i + 2r_{\max})^3}{\left(\frac{\lambda}{2} r_{\min}\right)^3} \le \frac{(3r_{\max})^3}{\left(\frac{\lambda}{2} r_{\min}\right)^3} \tag{80}$$

Thus:

$$M \le \left(6\frac{\kappa}{\lambda}\right)^3$$

So, the number of neighbors M is bounded by a constant that depends on the radii and distances of the spheres of the fused-sphere model. In practice, for the parameters we used, we noticed that the maximum number of intersecting neighbors never exceeded 150, whereas the average value was around 80. Hence, the calculation of the areas and volumes scales linearly with the size of the system, $O(N)$ (See Figure 78 for scaling on an Intel Pentium Xeon 866 MHz).

The algorithms presented were implemented in three different platforms: Linux (Intel), Irix (Silicon Graphics) and AIX (IBM) operating systems. The results are extremely accurate as compared to numerical calculations of the volumes of test molecules and very fast. On an Intel Pentium III 866 MHz the average time spent for the area and volume calculation of one atom is around 0.8ms (Figure 78). Areas of proteins of typical size, 2000-5000 atoms can be calculated analytically, with gradients, in just a couple of seconds. To our knowledge, this is the fastest implementation of the analytical calculation of areas and volumes per atom for a fused-sphere model to date.

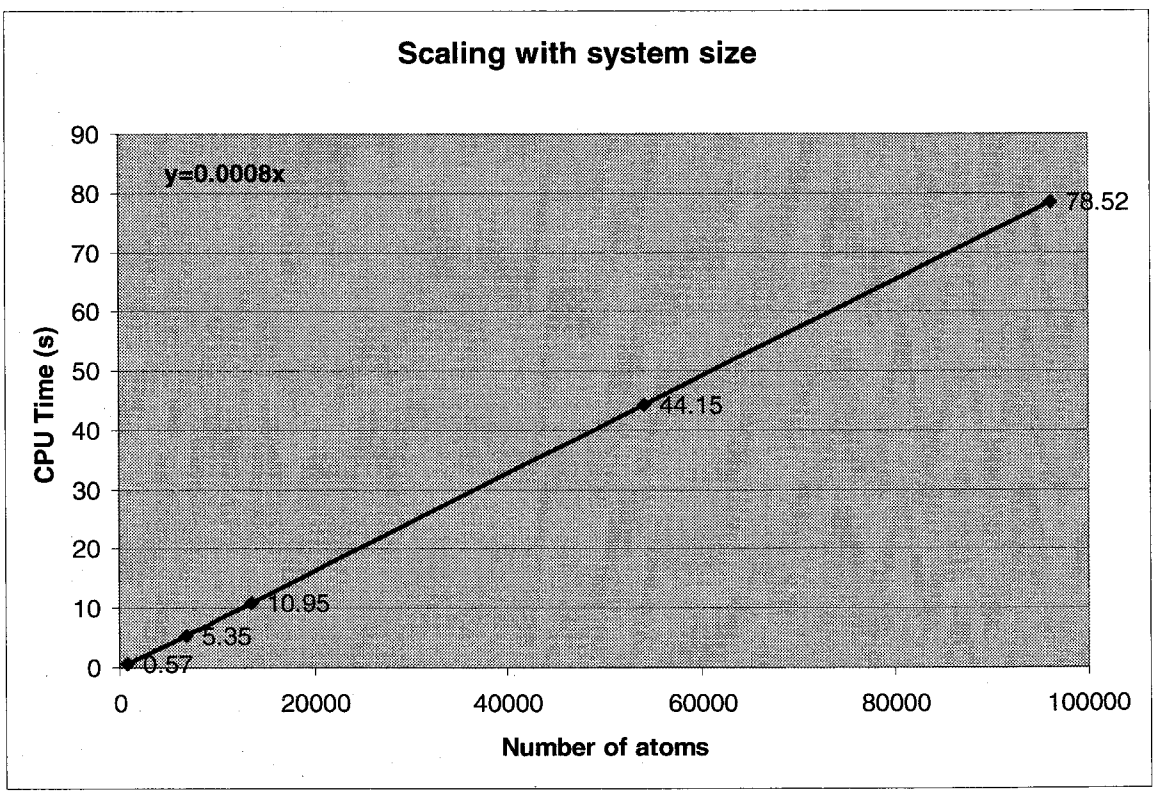

Figure 78. Linear scaling of the area/volume calculation with respect to the number of atoms in the system.

4 The AVGB-SAS Solvation Model

The algorithms in chapter 3 allow us to calculate accurately and efficiently the solvent accessible surface area A_i and solvent excluded volume V_i, with their gradients, for every atom i in a molecular system, assuming a fused-sphere model for the system where each atom is represented by a sphere of radius $r_i + r_p$, where r_i is the van der Waals radius of atom i and r_p the probe radius of the solvent, as it "rolls" around the solute. The volumes of each atom are necessary for the calculation of polar solvation effects according to the continuum dielectric theory, in the Generalized Born approximation, as was described in chapter 2. In particular, under certain approximations that were illustrated in sections 2.3 and 2.4, the Born radii are calculated by equations (37) and (41) for which the solvent excluded volumes V_i of each atom i are needed. Since these volumes are calculated analytically, we name this version of Generalized Born the "Analytical Volume Generalized Born" (AVGB) method. At the same time, as was explained in section 1.3.4, short range solvation effects are linearly dependent on the solvent accessible surface (SAS) areas of each atom in the system. The volume calculations have as a prerequisite the exposed area, so the short-range term is readily available in this calculation. The full solvation model includes both short-range and long-range (polar) effects and we call this the AVGB-SAS solvation model. In the following we will investigate the performance of the model as far as physical predictions and computational efficiency.

4.1 Validation of the AVGB Model

The AVGB model is an approximation to the solutions of the PB equation (7). In order to assess the quality of the AVGB model, we must compare our predictions to the results of numerical solutions of the PB equation. The best-known implementations are the DelPhi program [22], UHBD [23] and PBF [24]. We will also compare our results to the SGB version [55] of the GB approximation since it is the only GB implementation that does not depend on fitting parameters, as AVGB does neither. The comparison will be performed over different sets of molecules: small organic molecules, aminoacids, large proteins and dimers.

4.1.1 Small Molecules

The 376-molecule set for the comparisons is from reference [43], and each molecule has at most 40 atoms. The complete list is given in the Appendix. In order for the comparisons to be unbiased, we must use the same parameters for each test molecule in all methods. The van der Waals radii are taken from the DREIDING forcefield parameter set [90]. The charges are calculated from electrostatic potential (ESP) fitting [91] of the quantum-mechanical wave functions of each atom in the molecule, which in turn were estimated by Hartree-Fock ab-initio electronic structure calculations using the JAGUAR program [92] and the 6-31G** basis set. The probe radius of water was taken to be 1.4Å, the hard-sphere radius of water in the liquid state [12]. The dielectric constant of water is 78.2, and for the interior dielectric constant we chose a value of 1.

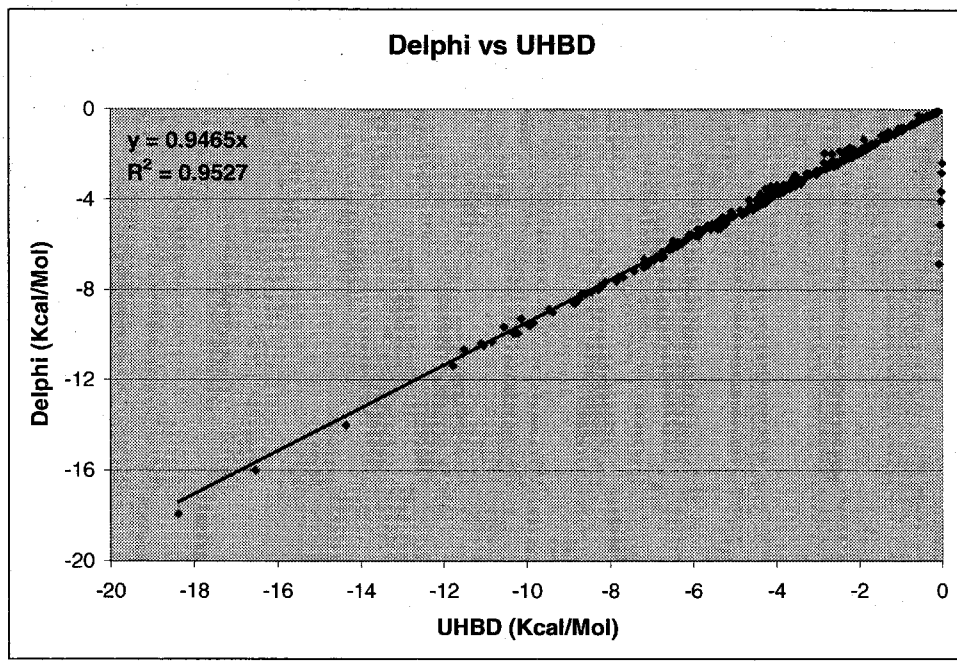

Figure 79. Comparison of the polar solvation energies between Delphi and UHBD for the molecule set of Table 6. The RMS difference is 0.62 Kcal/Mol.

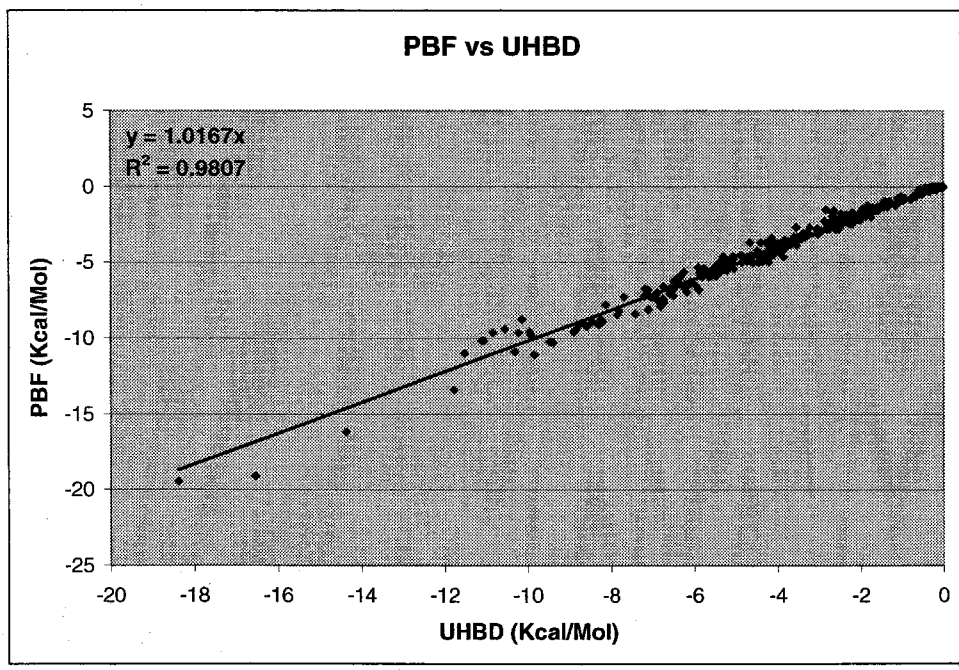

Figure 80. Comparison of the polar solvation energies between PBF and UHBD for the molecule set of Table 6. The RMS difference is 0.41 Kcal/Mol.

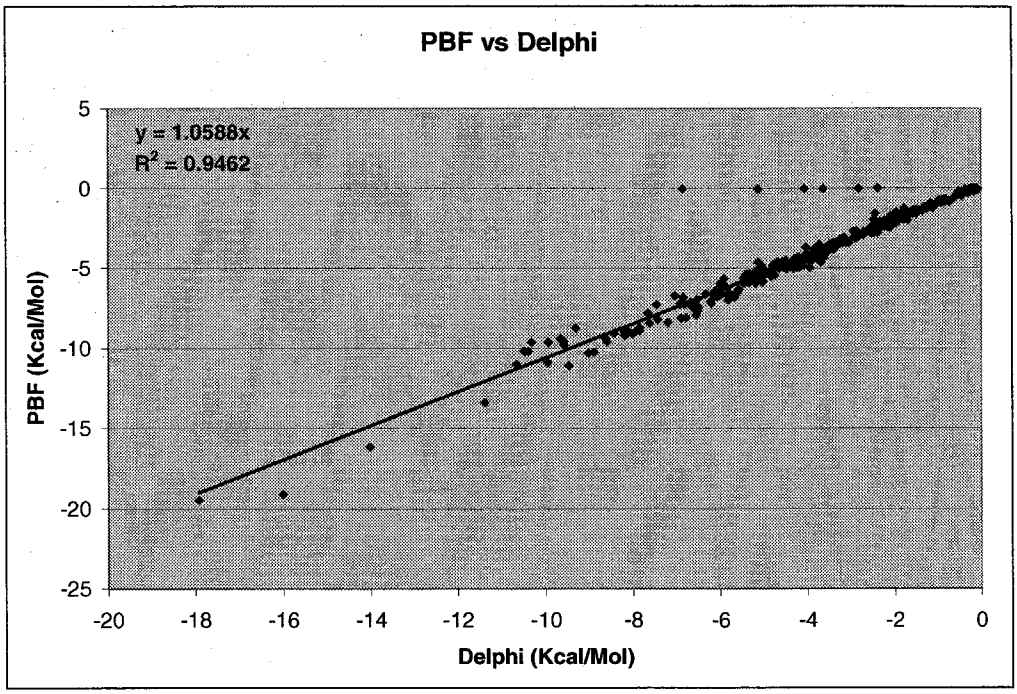

Figure 81. Comparison of the polar solvation energies between PBF and Delphi for the molecule set of Table 6. The RMS difference is 0.73 Kcal/Mol.

In Figure 79, Figure 80 and Figure 81 we compare the different methods that calculate numerically the PB equation, UHBD, Delphi and PBF. All methods correlate strongly with each other, as is shown by the linear regression coefficient R^2. The linear regression fit for every comparison is very close to the ideal $y = x$ line. However, looking more closely we see that UHBD behaves better overall, as is shown by the RMS differences between the results of the different methods. Delphi produces erroneous results for a few molecules and PBF does not correlate as strongly as the other two methods with each other. Thus, for the rest of the comparisons we will focus on UHBD as the method of choice for comparing GB to PB results.

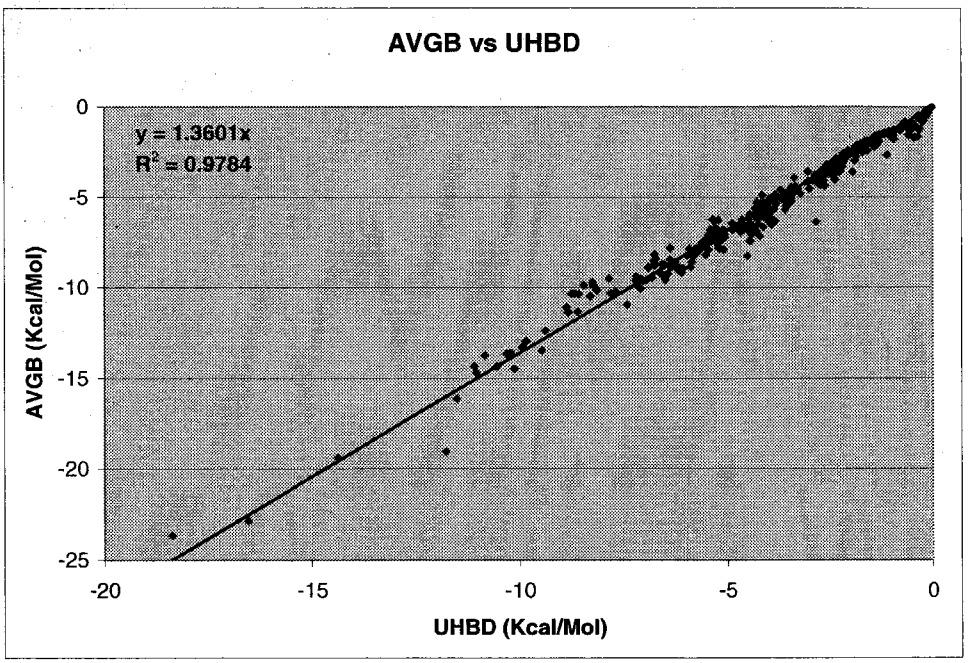

Figure 82. Comparison of the polar solvation energies between AVGB and UHBD for the molecule set of Table 6. The RMS difference is 1.79 Kcal/Mol.

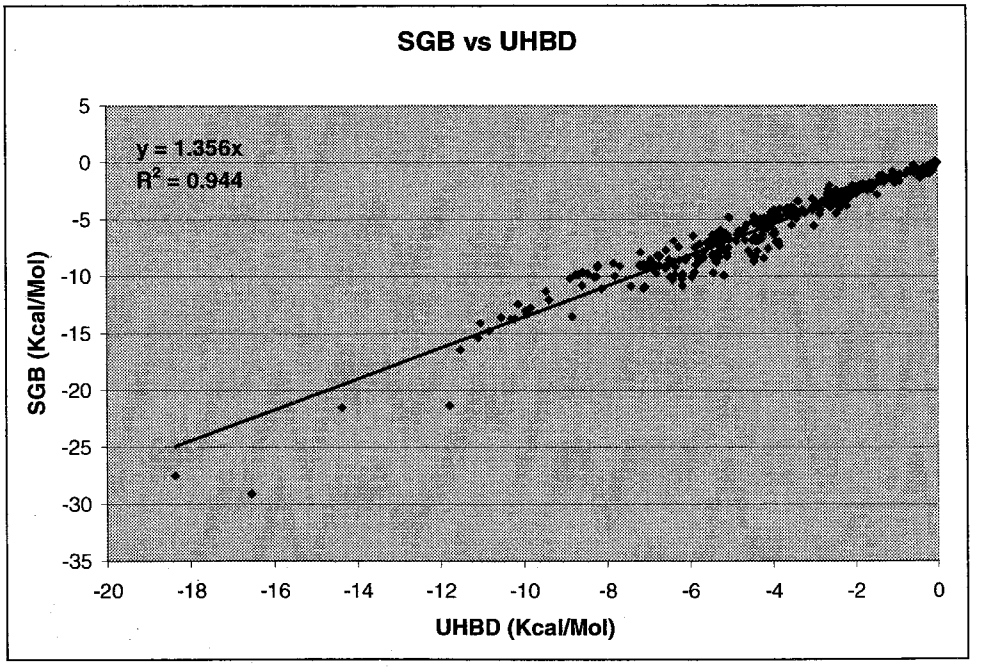

Figure 83. Comparison of the polar solvation energies between SGB and UHBD for the molecule set of Table 6. The RMS difference is 1.93 Kcal/Mol.

In Figure 82 we compare the results between AVGB and UHBD and in Figure 83 between SGB and UHBD. In both cases we see that the GB results are very well correlated to the PB numerical solutions, as shown by the linear regression coefficient, with AVGB correlating slightly better to the PB results than SGB does ($R^2 = 0.98$ for AVGB and $R^2 = 0.94$ for SGB). However, in both methods we observe a systematic deviation from the PB results, as is shown by the regression fits to the line. The slope a is different than unity and almost the same for both methods, 1.36, which means that both GB methods overestimate the polar solvation energy as the system gets more solvated. We observed such systematic deviation in Figure 6, where we used PB-derived Born radii and the interpolation formula (19). At the same time, the Coulombic approximation introduced in the Born radius calculation in section 2.3, along with the pairwise approximation, equation (41), may lead to additional errors that result in the systematic deviation from the correct PB results. The fact that both GB methods have almost the same systematic error hints to the fact that the deviation is probably not due to the volume or area calculation or the approximations used for the calculation of equation (37). It is the interpolation formula (19) and the coulombic approximation that leads to the formula for the calculation of the Born radii, equation (37), that are inducing the systematic error in the calculation of the polar solvation energies.

The fact that the error is systematic allows us to calibrate our parameters such that the predictions of AVGB match exactly the PB solutions. Of course, the success of such approach will depend on the number of parameters that need calibration, and the applicability of the calibrated results to molecular systems outside the set used for

calibration. Since the deviation is systematic and linear in nature, we can take advantage of the dependence on the interior dielectric constant ε_{in} of equation (18) to calibrate the AVGB results. In particular if U_{PB} is the PB polar solvation energy and U_{GB} the GB polar solvation energy, then, according to Figure 82 it is $U_{GB} = aU_{PB}$. We can rewrite equation (18) as

$$U_{GB} = \left(\frac{1}{\varepsilon_{in}} - \frac{1}{\varepsilon_{out}}\right) S \tag{81}$$

where $S = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{q_i q_j}{f_{ij}}$. We seek for a value $\varepsilon_{in}^{'}$ of the interior dielectric constant such that $U_{GB}^{'} = U_{PB}$. Then, we must have

$$\left(\frac{1}{\varepsilon_{in}^{'}} - \frac{1}{\varepsilon_{out}}\right) = \frac{1}{a}\left(\frac{1}{\varepsilon_{in}} - \frac{1}{\varepsilon_{out}}\right)$$

or, (82)

$$\frac{1}{\varepsilon_{in}^{'}} = \frac{1}{\varepsilon_{out}} + \frac{1}{a}\left(\frac{1}{\varepsilon_{in}} - \frac{1}{\varepsilon_{out}}\right)$$

Using equation (82) we can predict the value $\varepsilon_{in}^{'}$ that would give results that are very close to the PB solutions. In our case, using $a = 1.36$, $\varepsilon_{out} = 78.2$ and $\varepsilon_{in} = 1.0$, we get $\varepsilon_{in}^{'} \approx 1.3$. In Figure 84 we compare the results of AVGB with $\varepsilon_{in} = 1.3$ to UHBD with $\varepsilon_{in} = 1.0$. The correlation coefficient $R^2 = 0.98$ obviously is the same as in Figure 82, but the slope of the linear fit is now much closer to unity, $a = 1.04$. The RMS difference between the two methods is 0.46 Kcal/Mol. The agreement of the AVGB results to the numerical solutions of the PB equation is excellent and the differences are not larger than

the ones between the different implementations of numerical solutions of the PB equation, as shown in Figure 79, Figure 80 and Figure 81.

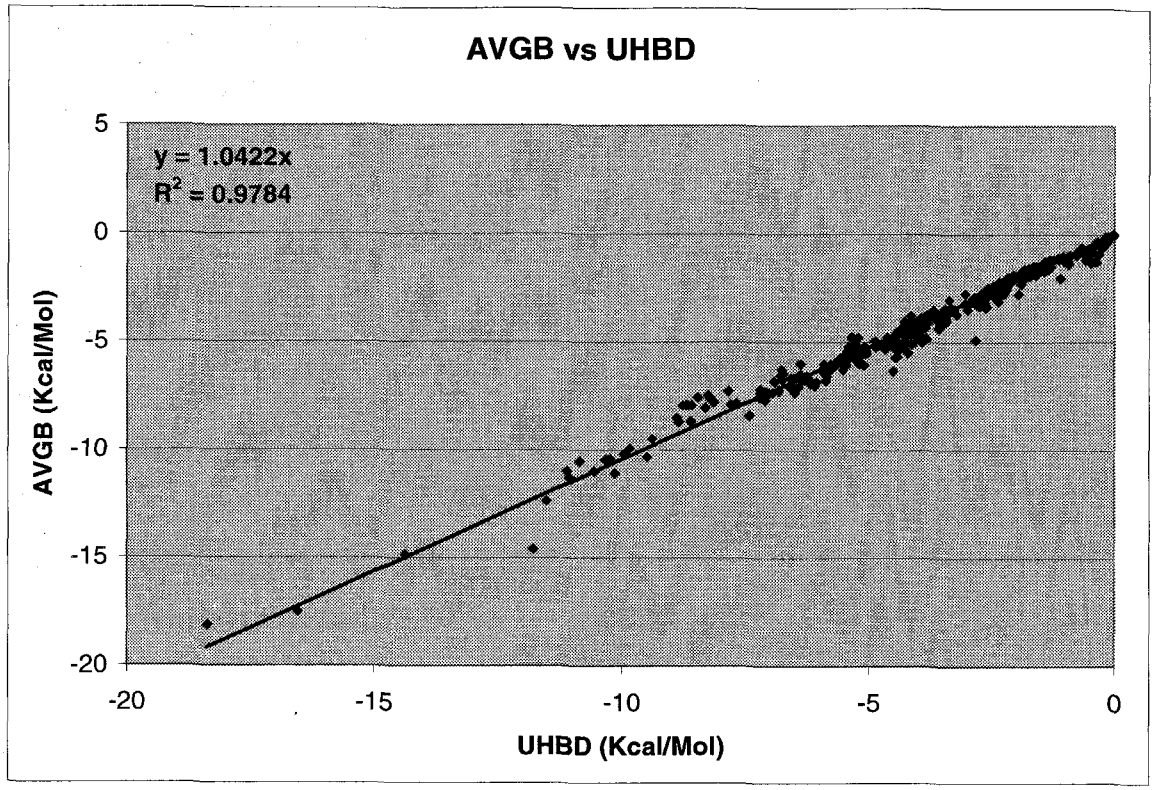

Figure 84. Comparison of AVGB with $\varepsilon_{in} = 1.3$ to UHBD with $\varepsilon_{in} = 1.0$ for the molecule list of Table 6. The RMS difference is 0.46 Kcal/Mol.

4.1.2 Large Molecules

The success of AVGB with small molecules is an encouraging step, but in order for the method to be applicable to realistic systems, it must be equally successful in predicting the polar solvation energies of larger molecular systems. Also, the calibration of the interior dielectric constant that was described in section 4.1.1 should not have to be

redone for the large systems. We compared the polar solvation energies for 11 proteins, shown in Table 1.The radii used were from the DREIDING forcefield [90], charges from the CHARMM22 forcefield [93] and the protein structures from the PDB protein databank. The solvent dielectric constant was taken $\varepsilon_{out} = 78.2$ for water and we tested AVGB with $\varepsilon_{in} = 1.0$ and $\varepsilon_{in} = 1.3$, and compared to UHBD with $\varepsilon_{in} = 1.0$. The solvent probe radius was 1.4Å.

Table 1. AVGB and UHBD polar solvation energies for 11 proteins.

Protein (nickname)	**Size (Atoms)**	**AVGB $\varepsilon_{in} = 1$ (Kcal/Mol)**	**AVGB $\varepsilon_{in} = 1.3$ (Kcal/Mol)**	**UHBD (Kcal/Mol)**
L-Arabinose (ara)	4671	-4478.08	-3458.34	-3236.64
Carbonic Anhydrase II (cah)	4032	-3613.93	-2790.97	-2557.5
Carboxypeptidase A (car)	4791	-3985.97	-3078.29	-2665.98
Cytochrome P-450cam (cyt)	6444	-7922.67	-6118.53	-6354.81
Intestinal FABP (fab)	2112	-2091.36	-1615.12	-1549.72
Neuraminidase (nad)	5978	-5329.57	-4115.93	-3686.73
Penicillopepsin (pep)	4550	-7003.04	-5408.32	-6061.3
ε-Thrombin (ret)	4766	-5407.50	-4176.11	-4106.13
Ribonuclease T1(rib)	1462	-2191.02	-1692.08	-1790.5
Thermolysin (tmn)	4700	-5150.37	-3977.54	-3835.41
Trypsin (trp)	3231	-2929.91	-2262.71	-2086.56

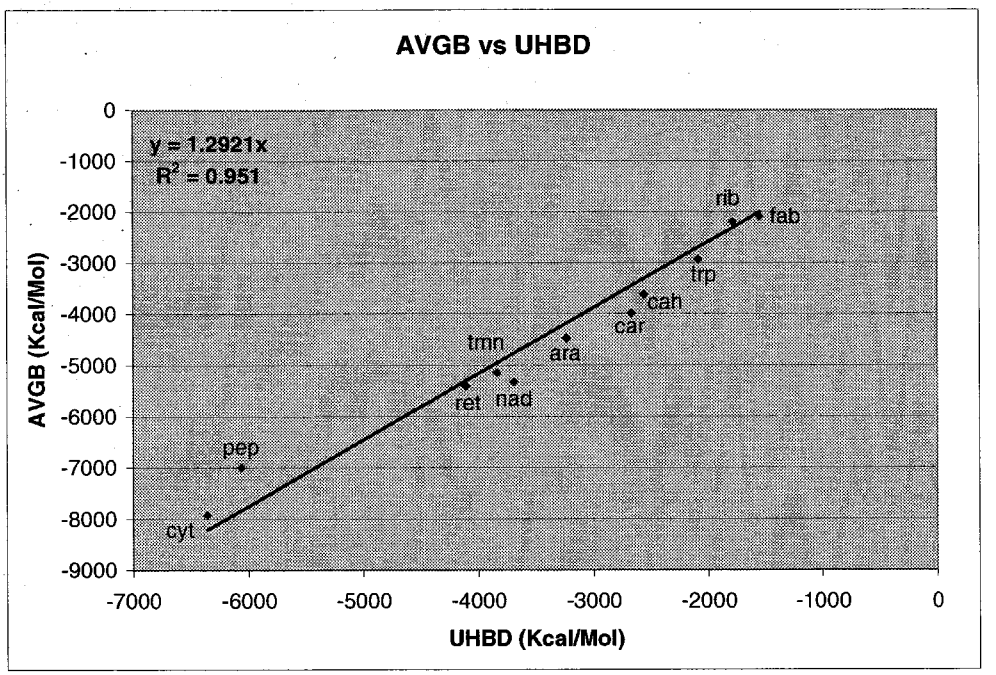

Figure 85. Comparison of AVGB and UHBD with $\varepsilon_{in} = 1.0$, for the proteins of Table 1. The RMS difference is 1169 Kcal/Mol.

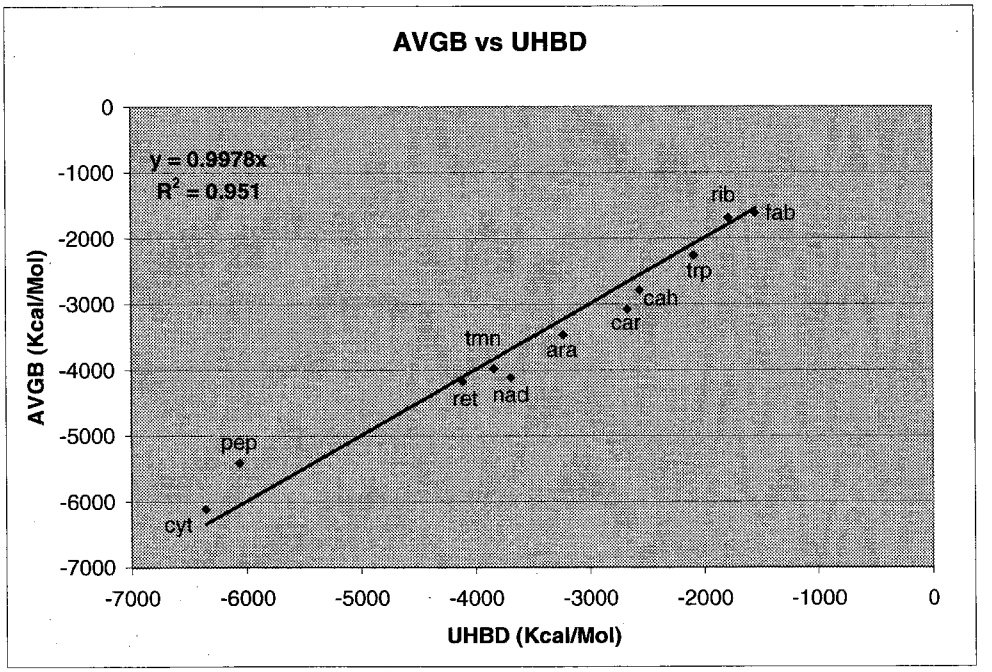

Figure 86. Comparison of AVGB with $\varepsilon_{in} = 1.3$ and UHBD with $\varepsilon_{in} = 1.0$, for the proteins of Table 1. The RMS difference is 303 Kcal/Mol.

In Figure 85 we see how AVGB compares to the numerical PB solution for the 11 proteins when we use the same value for the interior dielectric, $\varepsilon_{in} = 1.0$. We see that the two methods correlate very well to each other, $R^2 = 0.95$, but as with the small molecules, there is a systematic error since the slope of the linear fit is 1.29. In Figure 86 we compare AVGB with the calibrated value of the interior dielectric, $\varepsilon_{in} = 1.3$. The AVGB predictions are very close to the PB results as shown by the linear fit slope 0.998. The RMS difference is 303Kcal/Mol, which is about 10% of the polar solvation energy of these systems. The differences between AVGB and UHBD are on the same order as between UHBD and Delphi, which are different implementations of the numerical solution of the PB equation.

The fact that the calibrated value of the interior dielectric that we obtained from the small molecule set works so well with these large systems implies that we can use $\varepsilon_{in} = 1.3$ for any molecular system in order to get the polar solvation energy of that system, as predicted by the PB equation with $\varepsilon_{in} = 1.0$. Obviously, if we want the polar solvation energy for $\varepsilon_{in} \neq 1.0$ we need to redo the calibration procedure explained in section 4.1.1. From this analysis it is clear that the calibrated AVGB results are guaranteed to predict the polar solvation energy for molecules of any size and any solvent and solute dielectric constants.

4.1.3 Intermolecular Polar Solvation Energies

In sections 4.1.1 and 4.1.2 we proved that AVGB can successfully reproduce the polar solvation energies for small and large molecules. For the method to be applicable to any system, it must be able to describe accurately complexes and multi-molecular systems. This problem is more difficult than a single molecule calculation because of the complexity of the geometry of the solute-solvent boundary and the screened intermolecular interactions that have to be accounted for. To test the behavior of AVGB in such cases, we examined the polar solvation energy of a THF dimer in different orientations, as a function of the distance between the two molecules. Qualitatively, we expect that the energy of the dimer at infinite distance should be equal to the sum of the individual molecules' energies. At very close distances, the polar solvation energy should increase as the distance decreases; otherwise solvation would favor the collapse of the dimer. At intermediate distances we expect to have a minimum for which the configuration is optimally favorable.

In Figure 88 we see the polar solvation energy for the THF dimer of Figure 87, where the polar oxygen atoms face each other. The solvation energy correctly reproduces the infinity and zero distance limits, and it is a smooth function of the intermolecular distance. Similarly, in Figure 90 the polar solvation energy of the THF dimmer of Figure 89 obeys similar behavior. In this case, the polar atoms are away from each other. AVGB is able to correctly reproduce the intermolecular polar solvation energy. We note that for the same test cases, all other methods (UHBD, Delphi, PBF and SGB) did not predict the right energies at the infinity limits and the polar solvation energy was not a smooth

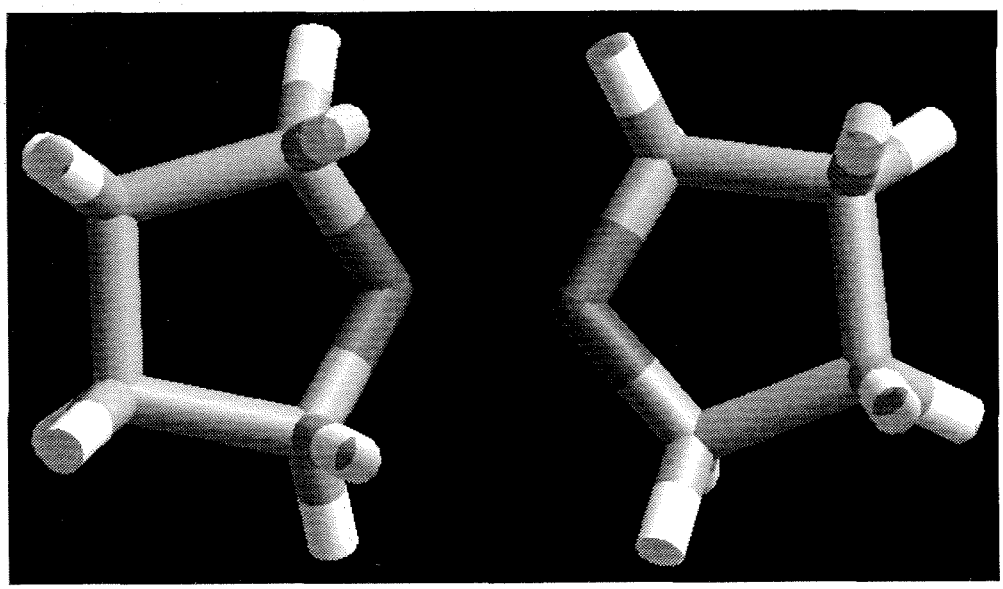

Figure 87. THF dimer with the polar parts facing each other.

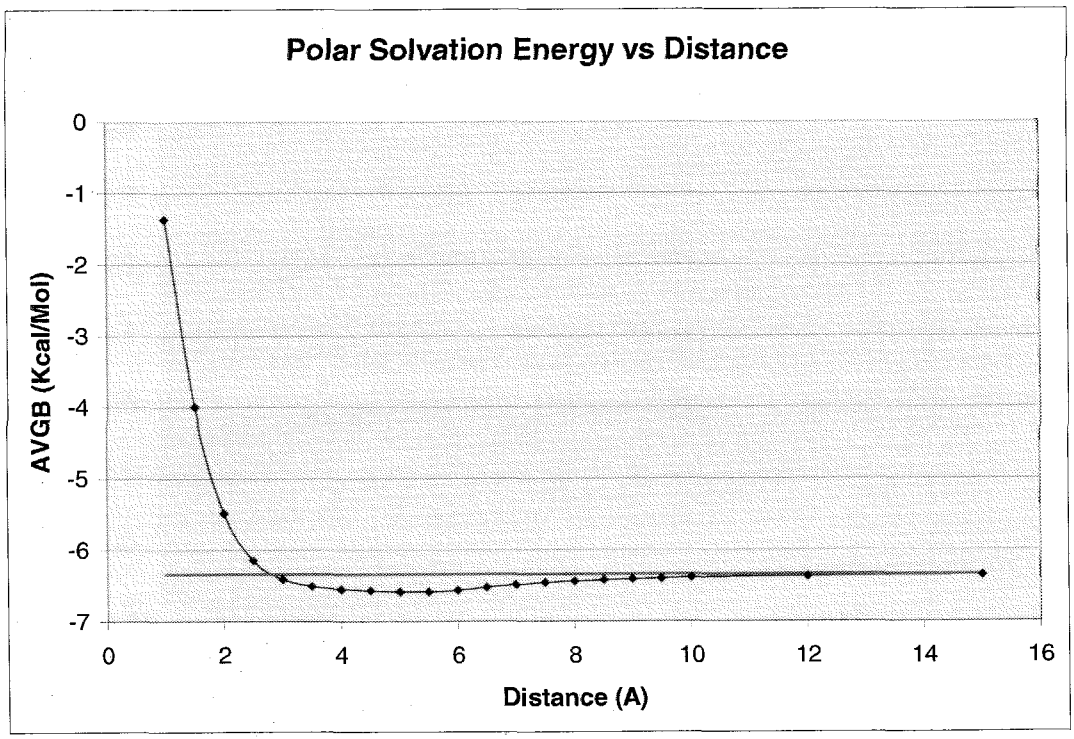

Figure 88. Polar solvation energy for the system of Figure 87 from AVGB as a function of the distance between the two THF molecules. The red line shows the energy when the molecules are infinitely separated from each other.

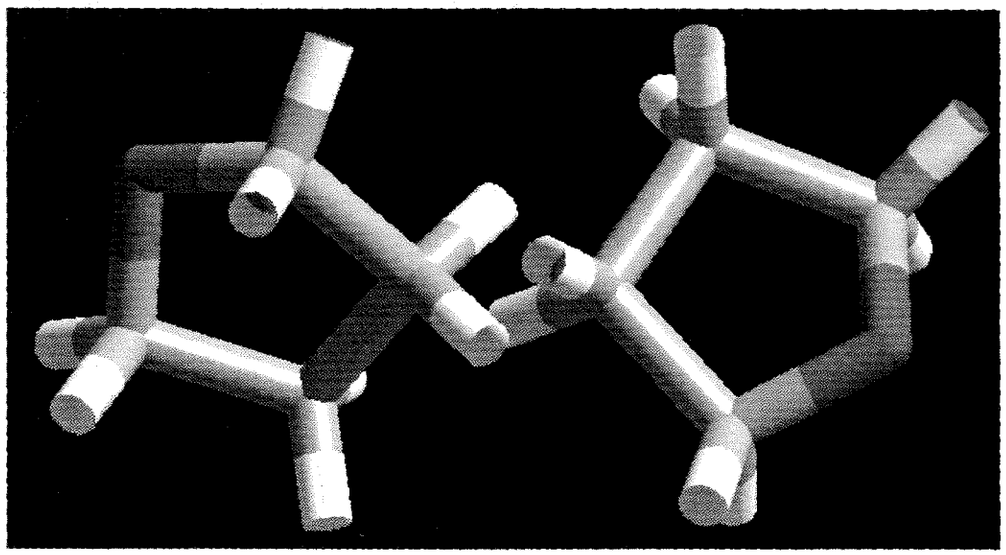

Figure 89. THF dimer with the polar parts away from each other.

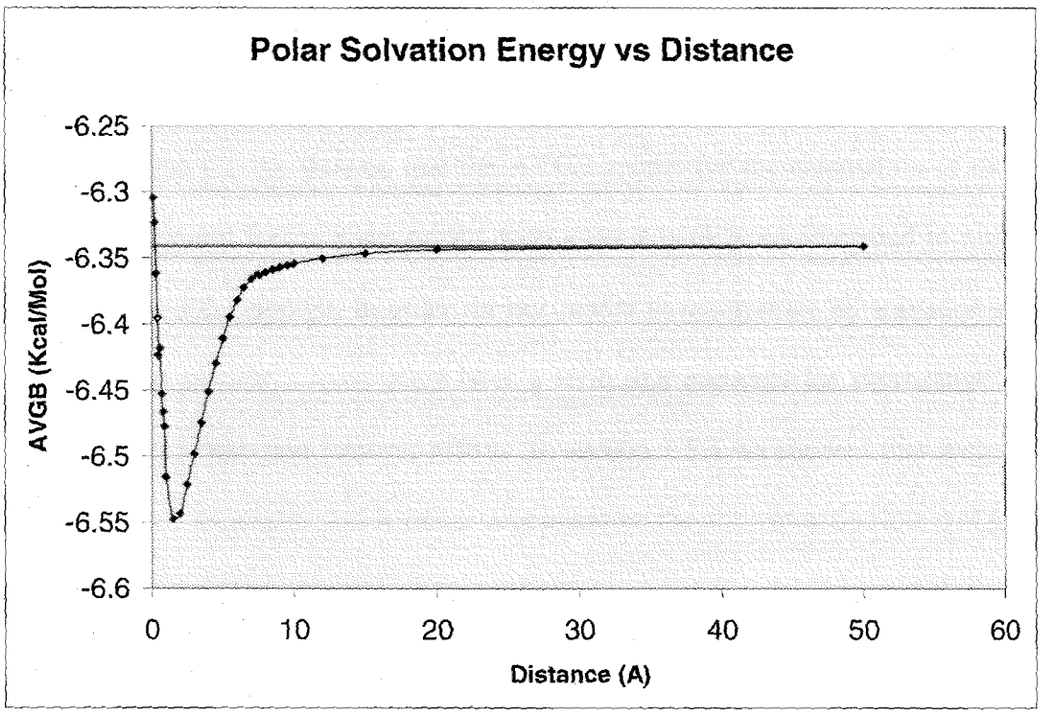

Figure 90. Polar solvation energy for the system of Figure 89 from AVGB as a function of the distance between the two THF molecules. The red line shows the energy when the molecules are infinitely separated from each other.

function of the distance. It is not clear why these methods fail to calculate the polar solvation energy correctly for the intermolecular problem, although they can calculate the energies correctly for individual molecules. Since the main difference between AVGB and all other methods is the analytical calculation of the geometry of the molecules, we believe that the numerical calculation of the boundaries of the molecules in multi-molecular systems used is wrong for such cases in these methods. At the same time, it is possible that since dimers with large intermolecular distances were examined, a much higher grid resolution would be needed to solve accurately the PB equation, but that would make the calculation practically infeasible.

4.2 The Short-Range Term

In section 4.1 we showed that the AVGB model for the calculation of the polar solvation energy and forces gives results with great accuracy, as compared to numerical solutions of the PB equation. In order for our model to incorporate all solvation effects, as explained in section 1.2, we must have a term that accounts for short-range effects such as van der Waals and entropic effects. In section 1.3.4 we showed that such effects can be described by a term that is linear dependent on the solvent accessible surface area A_i of every atom i in an N - atom system:

$$\Delta G_{vdW} + \Delta G_{cav} = \sum_{i}^{N} \sigma_i A_i \tag{83}$$

The parameters σ_i have units of surface tension, Energy/Area, and in general could be different for every atom in the system. However, in order to reduce the number of parameters in the model and to have a method that can be applicable to any system, there

have to be some rules on which surface tension would correspond to each atom in the system. Such rules have been previously developed for surface tension models (section 1.3.1) and their complexity can vary a lot. The surface tension can be a constant for all atoms, or depend on the atom element only, or depend on the element, the hybridization and connectivity with its valence-bond neighbors.

In order to determine the surface tension parameters for water, according to the AVGB-SAS model, we subtract the polar energies as predicted from AVGB from the experimental solvation energies of a large set of molecules, to get the predicted short-range contribution to the solvation energies. We then fit the surface tensions for each atom so that the differences in the energy predicted from equation (83) compared to the short-range term, for each molecule, are minimal. Clearly, the surface tensions that we get from this procedure depend on the molecule set and the parameters used in the AVGB calculation. It is important then to have experimental solvation free energies for a large set of molecules with diverse chemical groups. We use a set of 376 organic compounds for which the experimental solvation energies in water are given in [43] and the references therein.

The parameters for AVGB are the atomic radii, the probe radius of the solvent, the charges of the atoms, and the three-dimensional structure of each molecule. For the atomic radii we used the values from reference [94], which were optimized to reproduce water solvation energies of a small number of organic compounds by means of Hartree-Fock and Poisson-Bolzman calculations. The probe radius of water was taken to be 1.4Å,

according to [12], and the charges were derived from quantum-mechanical calculations described in section 4.1.1. The three-dimensional structures of the molecules were derived from energy minimization using the DREIDING forcefield [90]. Finally, the solute dielectric constant was 78.2 and for the interior we used the calibrated value of 1.3, according to section 4.1.1.

Initially we attempted to reproduce the experimental water solvation energies for our molecule list by using solvation types that depend only on the element of each atom. With the above-mentioned parameters the linear optimization procedure gave the surface tensions shown in Table 2 for each element represented in the molecule list.

Table 2. Surface tensions for water in (Kcal/Mol Å^2) per element, for the AVGB-SAS solvation model.

Element	**Surface Tension (Kcal/MolÅ^2)**
H	0.0007
C	0.0107
O	0.0094
N	-0.0448
F	0.0149
S	-0.0076
Cl	0.0003
Br	-0.0009
I	0.0021
P	0.5047

The comparison between the experimental and the predicted water solvation energies for the 376 small organic compounds is shown in Figure 91. The name, experimental and predicted solvation energies are shown in the Appendix.

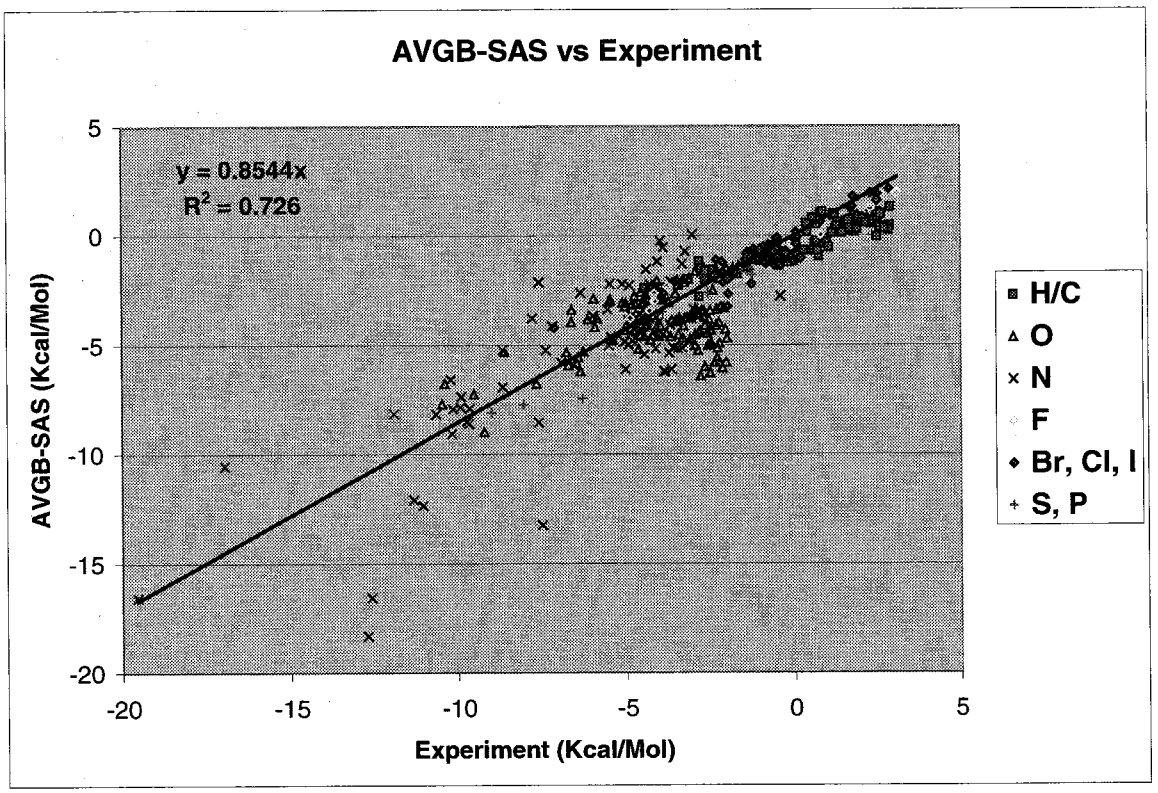

Figure 91. Comparison between the experimental and predicted water solvation energies for the AVGB-SAS solvation model, using solvation types by element. The different chemical groups are shown.

The correlation between experimental and predicted values is quite good, $R^2 = 0.73$. The mean unsigned error is 1.23Kcal/Mol and the RMS deviation is 1.61Kcal/Mol. The slope of the linear regression is 0.854. These results show that the use of equation (83) for the short-range term is a valid approximation because the predicted energies correlate quite well. However, the use of only element-dependent solvation

types is probably an oversimplification. The local environment of each atom, as described by its hybridization and valence bond connectivity, certainly plays a role in the short-range solvation energy. Thus we will attempt to redo the short-range term parameterization for a more complicated set of solvation types.

The solvation types we used are from [95] and they are shown, along with the corresponding surface tensions in Table 3. The surface tensions were derived in a similar fashion as before. The starting point for the linear optimization procedure was the surface tensions per element that we already derived. Then, the linear optimization was done separately for each set of molecules belonging to the same chemical group. Each time we examined a new chemical group, all the previously determined surface tensions remained constant and only the new solvation types that are introduced by the new group were optimized.

Table 3. Solvation type definitions and surface tension values.

Description	Solvation Type	Surface Tension
H bonded to sp3 C	H_MET	0.005
H bonded to sp2 C	H_BZN	-0.003
H bonded to sp1 C	H_ACE	0.038
H bonded to sp3 C bonded to OH group	H_ALC	0.041
H bonded to sp3 N in primary amine	H_N31	-0.019
H bonded to sp2 N bonded to single C	H_N21	0.135
H bonded to sp2 N in secondary amine	H_N32	-0.150
H bonded to sp2 N bonded to two C	H_N22	0.394
H bonded to S	H_S	0.055
sp3 C in a diol	C_DOL	0.085
Any C bonded to sp1 N	C_N1	0.048

Any C bonded to sp2 N	C_N2	0.002
sp3 C bonded to sp3 N	C_N3	0.020
Any other C	C_	0.010
sp3 O bonded to two C	O_32	-0.010
Nitrous O	O_2N	0.024
Carbonyl O in peptide bond	O_PEP	-0.133
Carboxylate O	O_2CM	0.028
Primary alcohol O or any other O	O_31	-0.075
sp3 N in tertiary amine	N_33	-1.785
sp2 N bonded to sp2 C and H	N_22	-2.165
Nitrous N	N_O2	-0.246
sp2 N in peptide bond	N_PEP	-0.084
sp2 primary N bonded to sp2 C bonded to three sp2 N	N_NNH	2.646
Any other N	N_	-0.035
F bonded to sp3 C bonded to three F	F_MET	0.019
Any other F	F_	0.015
Any Br	Br_	-0.001
Any Cl	Cl_	0.000
Any I	I_	-0.001
Any P	P_	2.179
Any S	S_	-0.022

The comparison between the experimental and predicted solvation energies using the surface tensions of Table 3 is shown in Figure 92. The average unsigned error is 0.72Kcal/Mol and the RMS deviation 0.98Kcal/Mol. The slope of the linear regression fit is 0.95 and the correlation coefficient is 0.9. Overall, the quality of the predictions of AVGB-SAS with the solvation types of Table 3 is very good. It is clear that the additional solvation types improve dramatically the predictions of the AVGB-SAS model. This fact could prompt us to continue refining our results by adding more solvation types. However, this would make the model more dependent on free parameters, which is not desired. At the same time, it is not clear by how much the

derived surface tensions are applicable to molecules outside the set we used for optimization. We certainly expect that these results would be at least qualitatively right, but predicting the experimental values for all possible molecules is a rather futile goal. In any case, the method presented in this section can be used with any solvation type set.

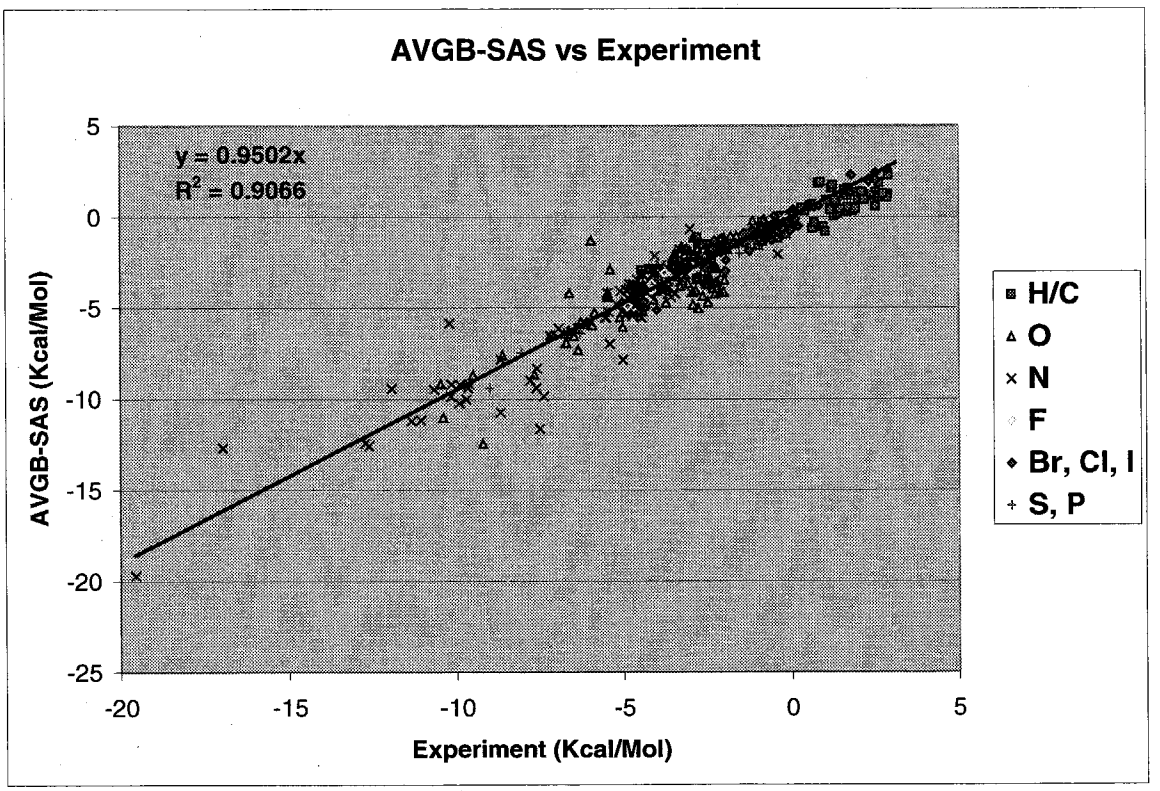

Figure 92. Comparison between the experimental and predicted water solvation energies for the AVGB-SAS solvation model, using the solvation types of Table 3. The different chemical groups are shown.

The surface tensions we derived are strictly applicable for solvation in water. In order to use the AVGB-SAS model for other solvents one must repeat the optimization procedure presented here. The only limitation is the need for experimental solvation energy data for a sufficiently large and diverse molecule set, for the solvent of interest.

After the atomic radii, the solvent probe radius and the atomic charges are determined, the polar energies can be calculated with AVGB with the respective interior/exterior dielectric constants for the new solvent. Then the procedure for calculating the surface tensions is identical to the one we used for water. The quality of the predictions, as compared to the experimental data, will determine if new solvation types will need to be defined.

4.3 Implementation of the AVGB-SAS Solvation Model

The AVGB-SAS solvation model was implemented and ported into two different molecular simulation packages, MPSim [96] and Dock [97]. MPSim is a parallel molecular dynamics program that uses the POSIX-threads multiprocessing standard [98] for parallelization. It is capable of performing molecular dynamics and energy minimization for very large systems under various thermodynamic ensembles. It can do cartesian, torsional or rigid-body dynamics and uses the cell-multipole method for fast calculation of the non-bond contributions to the atomic energies and forces [99]. Dock is a molecular database searching program that ranks ligands by their ability to bind in a specified site of a receptor. For every ligand, it generates a number of conformations into the target site of the receptor that are scored and ranked by their binding energy. The purpose of Dock is to identify the ligands that bind best to a specific receptor as part of rational drug design. The reason for developing the AVGB-SAS model was to be able to incorporate solvation effects in an accurate and efficient way in such methods. The accuracy of the AVGB-SAS model was established in sections 4.1 and 4.2. Here we will examine the performance of the model in these methods.

4.3.1 Parallel Molecular Dynamics

The AVGB-SAS model was implemented in MPSim [96]. The solvation energy and the solvation force were added into the total energy and force of every atom. MPSim is ported in three different platforms/Operating Systems: SGI/Irix, IBM/AIX, Intel/Linux. The AVGB-SAS method was implemented in all three platforms and the timings for a 3401 atom protein (PDB code: 1mcp) for each platform are shown in Figure 93. The tests were performed on an SGI Origin R10000 195MHz, IBM Power3-II 375MHz and Intel Pentium III 866MHz respectively.

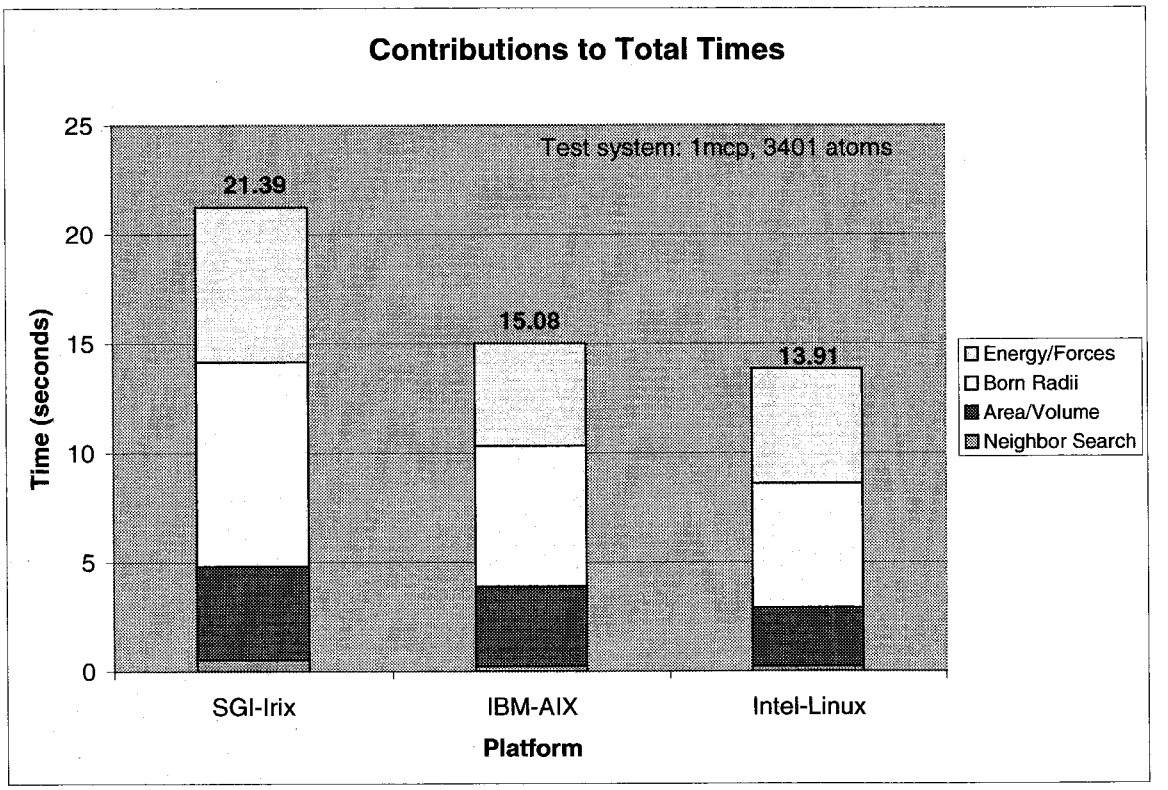

Figure 93. Total times for the AVGB-SAS model for a system of 3401 atoms, for different platforms. The contributions of the different parts of the calculation are shown.

Each energy/force calculation includes the following steps: neighbor search for each atom, identification of the true neighbors and the area and volume calculation, Born radii calculation using equations (37) and (41) and polar solvation energy calculation from equation (18). The short-range term is easily calculated after the areas for each atom have been determined. It is clear that the bulk of the CPU time spent for the calculation is on the Born radii and secondarily on the polar solvation energy/force calculation. This is because of the $O(N^2)$ nature of these calculations where N is the number of atoms in the molecular system. The area and volume calculations scale as $O(N)$, as was shown in section 3.4.2. The overall scaling of the AVGB-SAS method is shown in Figure 94.

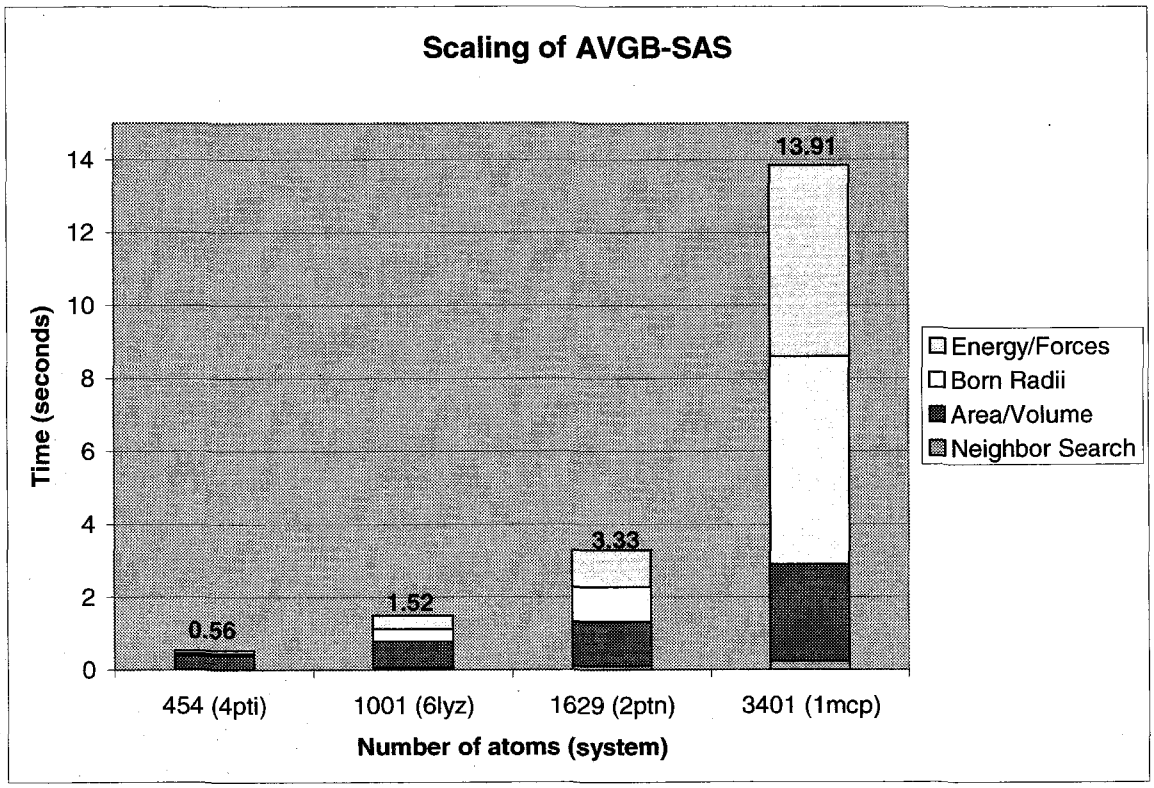

Figure 94. Scaling of the AVGB-SAS method as a function of the size (number of atoms) of the molecular system. The calculations were performed on an Intel Pentium III 866MHz.

The CPU time required for the solvation energy calculation of the same test system, using numerical solutions to the PB equation, depends on the software and the grid resolution used. In any case, the minimum time spent was about a minute for Delphi and about 8 minutes for UHBD which is obviously very inefficient for use in molecular dynamics where a typical run involves at least calculations for a 1000 steps. The CPU time comparison of AVGB to SGB is shown in Figure 95. AVGB is 3-5 times faster than SGB, depending on the platform. This is a significant improvement over other versions of the Generalized Born method.

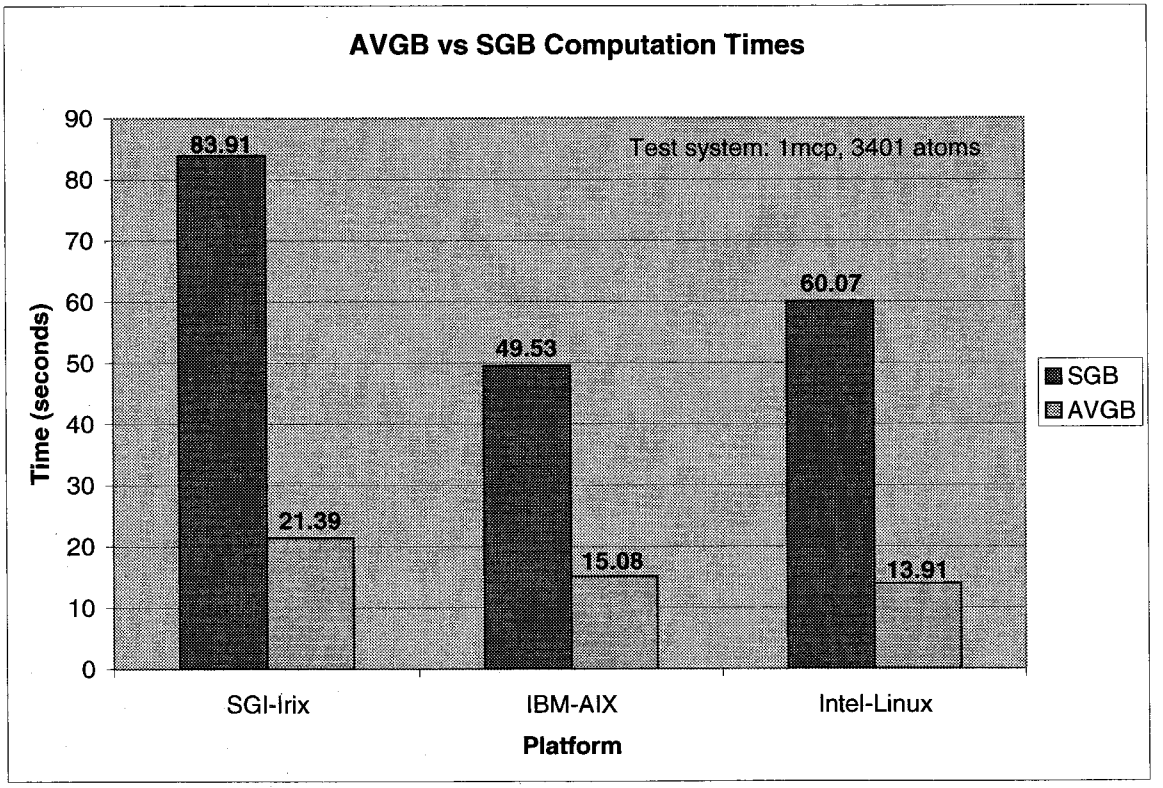

Figure 95. Comparison of CPU time spent for calculating the solvation energy of 1mcp between SGB and AVGB, for three different platforms.

We can boost the performance of AVGB-SAS even more by using parallel computers. The AVGB-SAS model is straightforwardly parallelizable due to the nature of the calculation. All the atoms in the system are uniformly distributed to all available processors. The solvent accessible surface and solvent excluded volume is calculated for every atom, independently of all other atoms. After the area and volume calculations are done, the Born radii and polar solvation energies can again be calculated independently for each atom using equations (41) and (18). The fact that the area, volume, Born radii and polar solvation energy for each atom are calculated independently of all other atoms means that the information necessary to be passed between different processors is minimal and this makes the algorithm naturally parallel.

The scaling of the CPU time spent as a function of the number of processors used is shown in Figure 96. This test was performed on a 4-processor SMP (Symmetric Multi Processor) shared memory machine, for the protein1mcp (3401 atoms). The scaling is very good which means that the overhead due to processor communication is indeed minimal and the parallel implementation is very efficient. For 4 processors we get a boost of about 3.6 (or 360%) in the computation time. The time spent for the calculation of the same system using SGB (which is not parallelizable due to the global area calculation of the surface of the solute) is about 4 times more than AVGB for one processor and 14 for 4 processors. The excellent parallel performance of AVGB and the simplicity of the parallel implementation open the way for the simulation, including solvation effects, of very large systems with the use of massively parallel computers. In contrast, this is very difficult to do for methods that calculate numerically solutions of the PB equation.

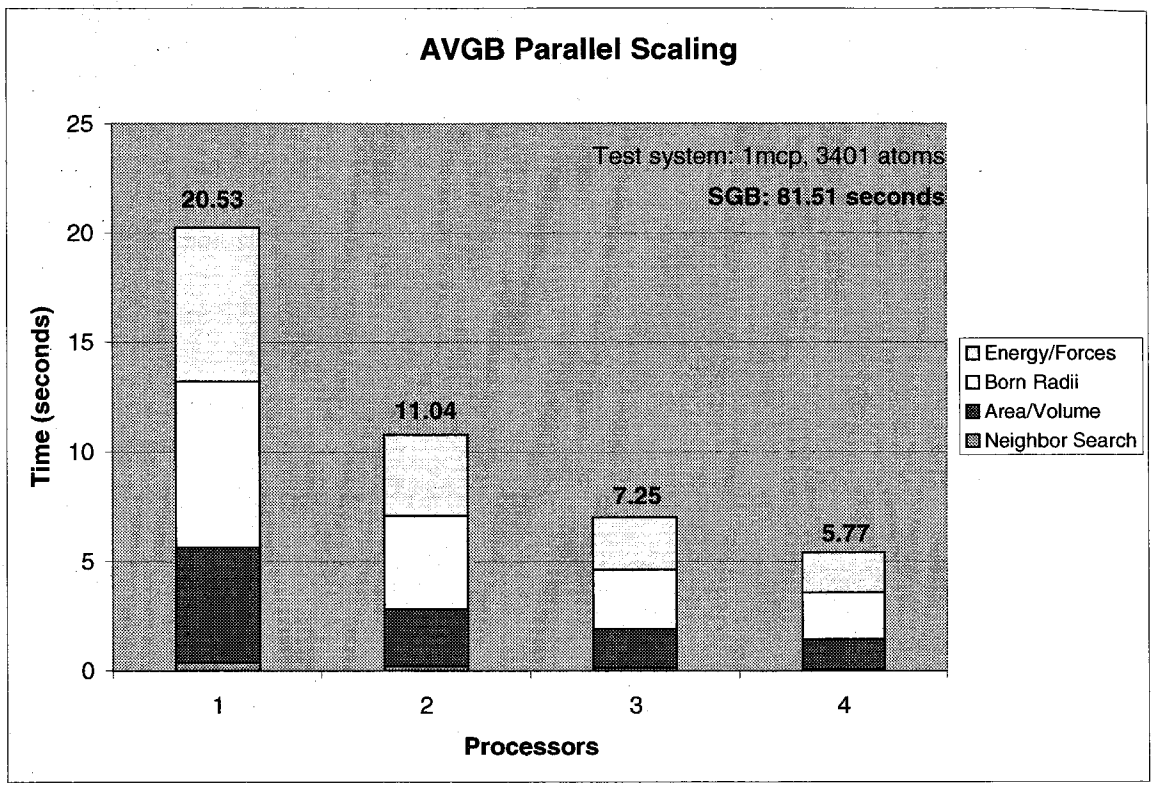

Figure 96. Parallel scaling of AVGB-SAS on a shared memory symmetric 4-processor Intel Pentium III Xeon 550MHz, for a 3401 atom protein (1mcp.)

From the breakdown of the total CPU time to the individual contributions of the different parts of the AVGB-SAS calculation, Figure 93, it is clear that the most expensive part of the calculation is the computation of the Born radii. In molecular dynamics, the calculation is done in successive steps that usually correspond to a time of 1fs. After every step, the structure of the molecules changes very little. Only after a rather large number of steps (usually of the order of 100 or more) does the structure change noticeably. The exposed areas, volumes, Born radii and thus the polar solvation energy and short-range term are dependent exclusively on the structure of the solute. We expect then that the Born radii will not change by any significant amount in a few steps and

accordingly the solvation energy. It is probably a good approximation then to update the Born radii only every few steps, thus saving a lot of computation time in the meantime. To test this we run dynamics for a small protein for 200 steps with different Born radii update frequencies: 1, 2, 5, 10, 20 and 50 steps. The solvation energy for each run is shown in Figure 97.

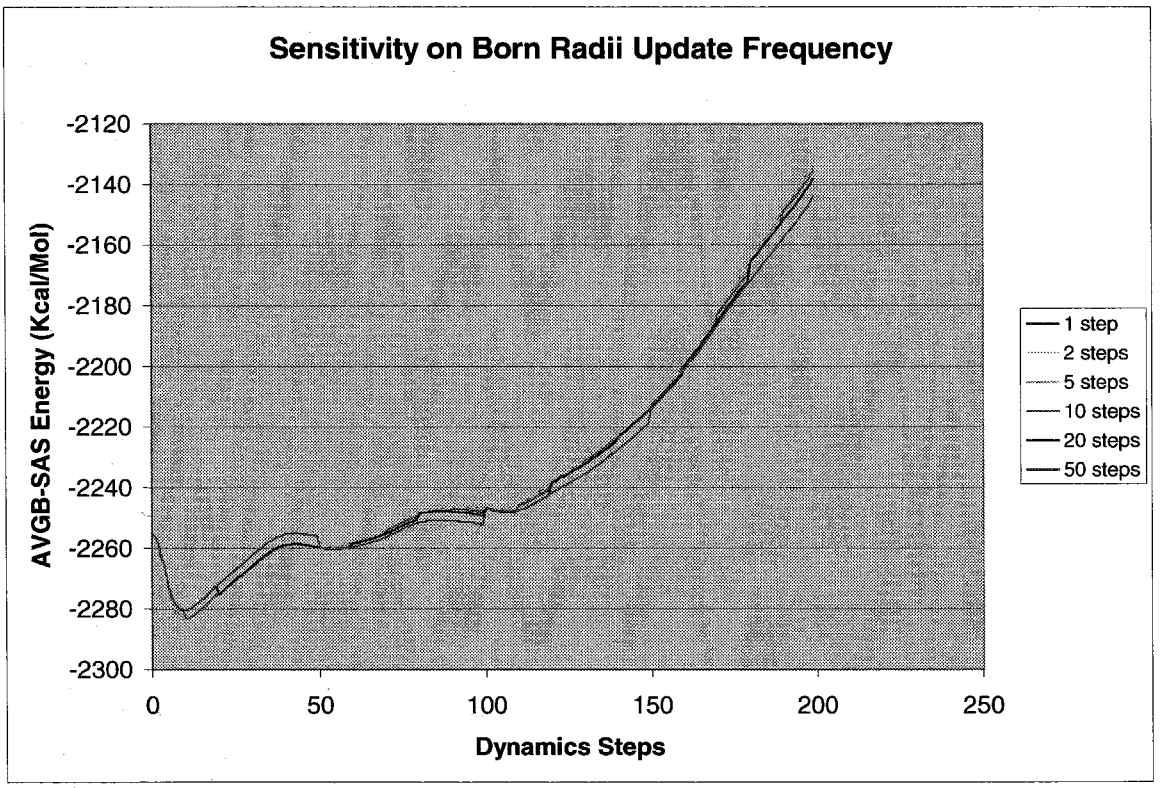

Figure 97. Sensitivity of the AVGB-SAS energy with the update frequency of the Born radii. The test was performed on a small protein (4pti, 454 atoms) for 200 steps.

We notice that the differences in the solvation energy are very small, just a few Kcal/Mol, and the trajectory of the dynamics is stable during the course of the run, even when updating the Born radii every 50 steps. By periodically updating the Born radii we

gain significant speedup in the molecular dynamics, which are necessary for long time simulations.

4.3.2 Molecular Docking

In molecular docking the goal is to rank a large number of ligands by their ability to bind in a specified site of a receptor as part of rational drug design. The particular conformations of each ligand in the binding site are scored according to the binding energy. It is important to include solvation effects in such calculations so that the selected ligands bind best with the receptor in the natural aqueous environment. The binding energy in solvent is calculated by using the thermodynamic cycle of Figure 98.

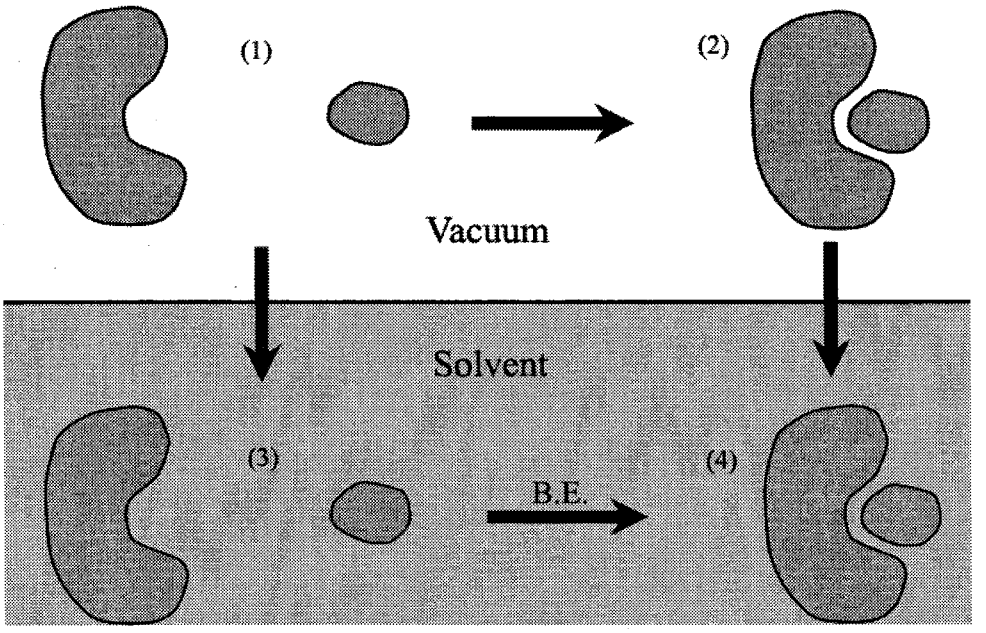

Figure 98. Thermodynamic cycle for the calculation of the binding energy of a receptor-ligand complex in solution.

Let us set as U^L, U^R, U^C the internal energy and ΔG^L, ΔG^R, ΔG^C the solvation energy of the ligand, receptor and ligand-receptor complex respectively. State 1 in Figure 98 corresponds to the ligand and receptor in vacuum, separated by an infinite distance. State 2 is the complex in vacuum. State 3 is the ligand and receptor in solvent, separated by an infinite distance, and state 4 is the complex in the solvent. The energy required to get from state 1 to state 2 is

$$\Delta W_{1\to 2} = U^C - U^L - U^R \tag{84}$$

The energy required to get from 2 to 4 is the solvation energy of the complex, $\Delta W_{2\to 4} = \Delta G^C$, and the energy required to get from 1 to 3 is the sum of the individual solvation energies of the ligand and the receptor, $\Delta W_{1\to 3} = \Delta G^L + \Delta G^R$. The binding energy of the ligand-receptor complex in water, $B.E.$, is then

$$\begin{aligned} B.E. &= \Delta W_{3\to 4} \\ &= \Delta W_{1\to 2} + \Delta W_{2\to 4} - \Delta W_{1\to 3} \\ &= \left(U^C + \Delta G^C\right) - \left(U^L + \Delta G^L\right) - \left(U^R + \Delta G^R\right) \end{aligned} \tag{85}$$

So, for every ligand conformation generated, to calculate the binding energy we must calculate the solvation energy of the ligand, the receptor and the complex, along with the internal energies of each system.

In order to save CPU time for the calculation, it is very common in docking simulations to assume the receptor to be fixed in space and only the ligand to be flexible as we search around the receptor's binding site to find an optimal configuration. This is a reasonable approximation because of the difference in the sizes of the two systems. A typical receptor has at least a few thousand atoms, where a ligand has not more than a

hundred or so. This way, only the energy of the ligand and the interaction energy between ligand and receptor need to be calculated. The intramolecular interactions in the receptor can be neglected since they are a constant for all possible ligand conformations.

We can use this fact to save additional CPU time in the calculation of the solvation energies. In particular, for most of the fixed atoms in the complex we need not calculate the areas and volumes since their intersecting neighbors are also fixed and thus the area and volume that they bury do not change. The fixed (receptor) atoms that need to be recalculated are the neighboring atoms of the movable (ligand) atoms. Also, an additional timesaving approximation that can be done is at the Born radius calculation: the Born radius of an atom i is dependent on the volumes of all other atoms in the system. The contribution of each atom to the Born radius of atom i falls approximately as the inverse of the distance between the two atoms (see equation (42)). Thus, for atoms that are fixed and far away from the binding site (where the ligand is) the Born radius should not change by much because of the presence of the ligand. For these atoms we use the initial Born radius value, without the ligand's contribution (receptor only). The Born radii are updated only for receptor atoms close to the ligand. With these additional timesaving schemes, we have been able to reduce the solvation binding energy of a 4000-atom complex from 14s down to 2.5s, which is a significant improvement.

5 *Applications of the AVGB-SAS Solvation Model*

In chapter 4 we examined the performance of the AVGB-SAS model as far as the quality of its predictions and the CPU time efficiency. The AVGB model has many advantages with respect to all previous methods for calculating the electrostatic solvation effects. In particular, the quality of the results is as good as for numerical solutions of the PB equation, but AVGB is orders of magnitude faster than various implementations of the PB model. At the same time the method is capable of calculating forces, analytically, which is necessary for the model to be useful in molecular dynamics. Unlike most other flavors of the Generalized Born theory, AVGB does not depend on any fitting parameters. In fact, we believe that the reason that we achieve such good results is our ability to calculate accurately the solute geometry, through the volume and area calculations described in chapter 3. Comparing AVGB to SGB, the only other GB method that does not depend on any ad-hoc parameterizations, AVGB is four times faster and the results are better. Also, the way the area and volume calculations are done allow for distributed computation of the solvation effects and as is shown in Figure 96 the overhead of the parallel calculation is minimal, thus achieving very good scaling and excellent CPU time performance. Finally, in the process of calculating the polar effects we have to calculate the solvent accessible areas for each atom, which is needed for the inclusion of short-range effects in the solvation energy. In the AVGB-SAS model, unlike other solvation methods, the long-range and short-range terms of the solvation energy and their derivatives are analytically calculated at the same time, which results in the excellent performance of the AVGB-SAS model.

At the same time, we should be aware of the limitations of the AVGB-SAS model. As any implicit solvation model it is unable to explicitly take into account charge transfer and hydrogen bonding effects between solute and solvent. Also, although salt effects are incorporated into the model in an average way, it is only in an approximation and the results are expected to be valid only for low salt concentrations. Finally, pH effects are not included at all in this model. Overall, the quality of the AVGB-SAS predictions can be only as good as the dielectric model is an applicable approximation for the system studied, which is often the case.

Keeping the aforementioned successes and limitations in mind, we implemented and incorporated the AVGB-SAS solvation model in MPSim [96], a parallel molecular dynamics program, and Dock [97], a ligand database searching program. In sections 5.1 and 5.2 we will apply the AVGB-SAS model in the simulation of the dynamics of nucleic acids and in virtual ligand screening (VLS).

5.1 B-DNA Molecular Dynamics

Nucleic acids play a fundamental role in biological functions, and their accurate simulation is of primary importance. Because of DNA's importance and complexity, there is a strong focus in the scientific community to simulate accurately its dynamics (see [100] for a review of molecular dynamics simulations of nucleic acids). The natural environment of DNA is water and DNA in solution exhibits complex dynamical behavior such as bending and stretching.

DNA has traditionally been a difficult system to simulate. It is a highly charged molecular system and long-range interactions have to be calculated accurately in order to maintain a stable double-helix structure. Also, it is essential that we include the effects of the water solvent in the simulation if we want to accurately simulate the behavior of DNA in its natural environment. Usually the water effects were incorporated by explicitly including water molecules in the simulation, with great expense in CPU time. In this section we show that by including solvation effects using the AVGB-SAS model, in a molecular dynamics simulation of DNA, we are able to achieve a stable double-helix structure and observe some dynamical effects.

For the simulations we started from the canonical form of B-DNA [101] (Figure 99) of the dodecamer $d(CGCATATATGCG)_2$ and used the molecular dynamics software package MPSim [96] with the AMBER forcefield [102]. We performed 80 ps of dynamics in vacuum and in solvent, using AVGB-SAS for the calculation of energies and forces from the solvent to the DNA. We run the simulation with two different boundary conditions: free tips and fixed tips. The latter was done so we could prevent the double helix from bending, and allow us to compare the final structures to the canonical B-DNA structure.

The final structures for the unconstraint simulations are shown in Figure 100 in vacuum and Figure 101 in solvent. We notice that when solvent is included, the double helix is bended, which is a phenomenon that has been observed experimentally [103]. In contrast, the vacuum simulation does not clearly show such effect.

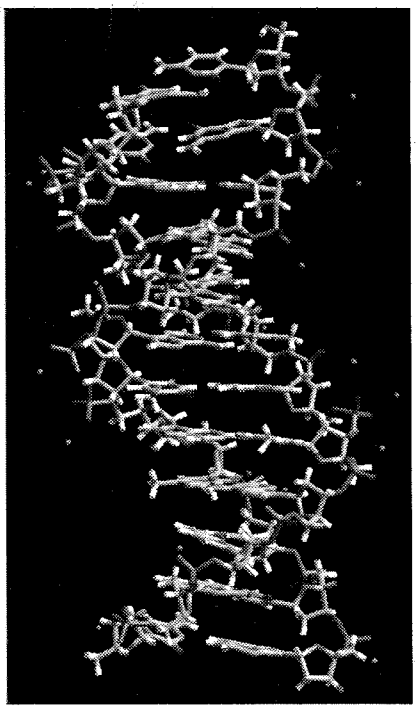

Figure 99. The initial structure: canonical B-DNA.

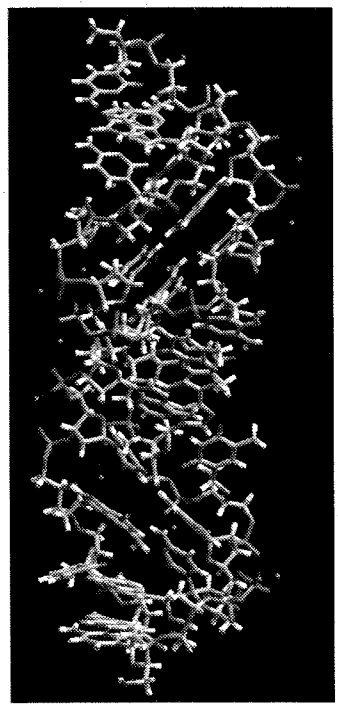

Figure 100. B-DNA after 80ps in vacuum with free tips.

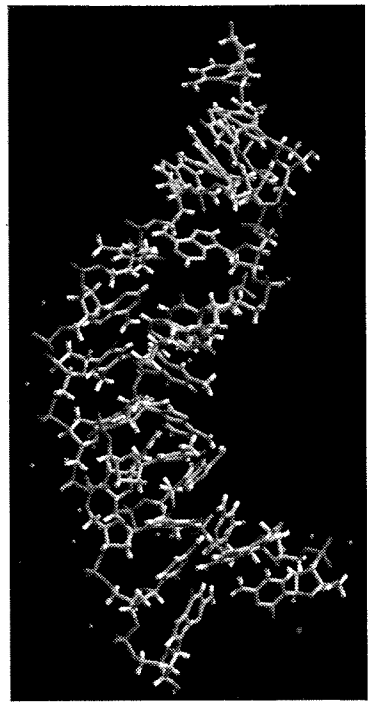

Figure 101. B-DNA after 80ps in implicit solvent with free tips.

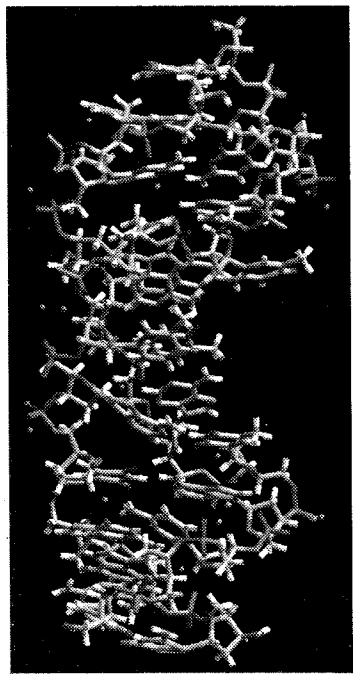

Figure 102. B-DNA after 80ps in vacuum with fixed tips.

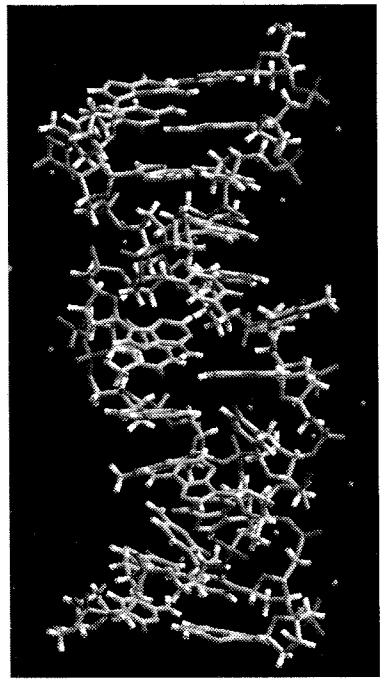

Figure 103. B-DNA after 80ps in implicit solvent with fixed tips.

In the simulations where the tips of the double helices are fixed, the final structures are shown in Figure 102 in vacuum and Figure 103 in solvent. Because of this constraint we expect that a good quality simulation should preserve the structure of the double helix as close as possible to the canonical B-DNA form. Qualitatively we can verify that the solvent simulation preserves the structure, whereas the vacuum one does not do as well. We can quantify this by examining the root mean square deviation (RMSD) of the non-hydrogen atoms at each step in the simulation versus the canonical B-DNA.

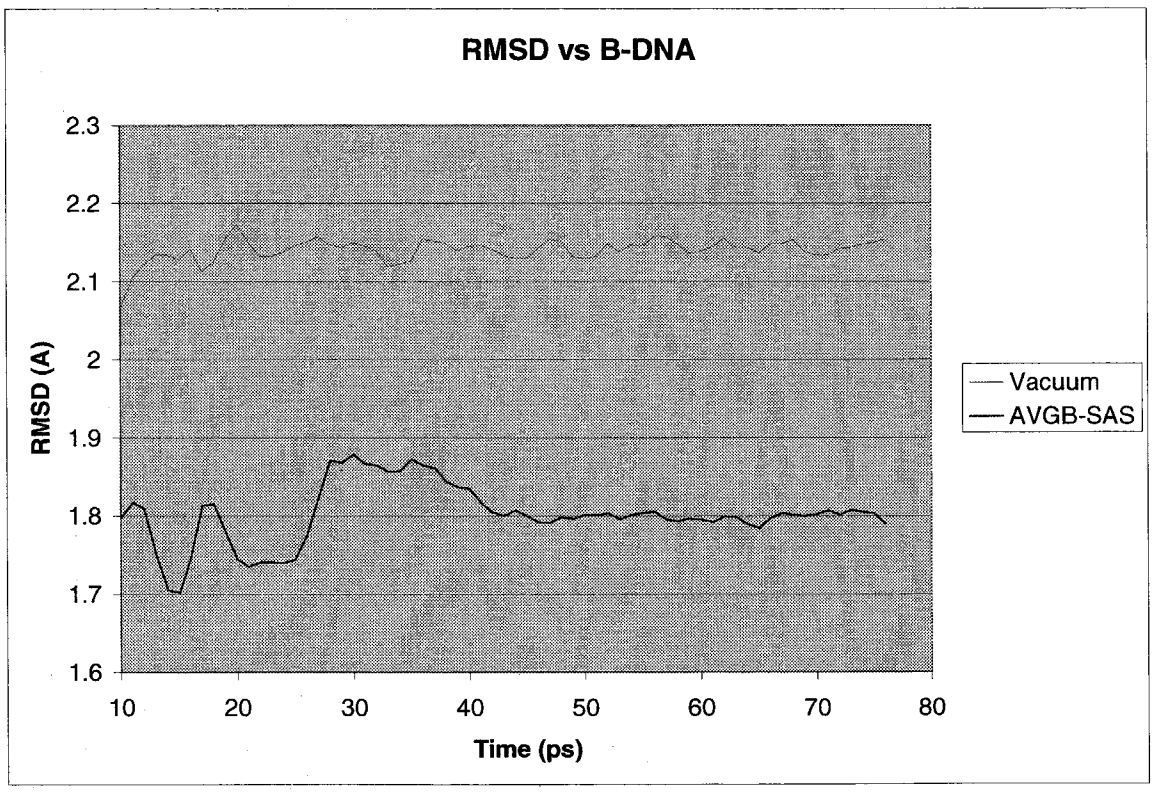

Figure 104. RMSD of non-hydrogen atoms between simulation snapshots and the canonical B-DNA, for vacuum and solvent simulations using AVGB-SAS.

The plots in Figure 104 show the RMSD of non-hydrogen atoms for every snapshot of the simulation (taken every 1 ps) versus the canonical B-DNA. The plots are 3-point averaged. We clearly see that the simulation in solvent produces structures that are closer to the canonical B-DNA than the vacuum simulation. This is a good indication that the solvent is indeed crucial in order to get a stable DNA structure and that the AVGB-SAS is a good method to simulate the solvent effects.

We would like to emphasize that these results are by no means a complete analysis of the dynamic and static behavior of DNA in solvent. This is merely a proof of concept for the AVGB-SAS model. We give an example of the application of the AVGB-SAS model in simulations and show that the results have physical consequences that have been found experimentally. A complete study of the behavior of DNA should involve much longer simulations and a more elaborate analysis of the structures and energetics during the dynamics.

5.2 Virtual Ligand Screening (VLS)

Simulations can play a vital role in rational drug design. The recent discovery of the human genome [1] has made possible the identification of the sequence of proteins that mediate important cell functions. Drugs are small molecules (ligands) that bind to a specific site of a target protein (receptor) such that it will either stop or aid a function of that protein. Molecular simulations can assist in the discovery of such drugs by searching through large databases of potential ligands and identifying the best candidates according to their binding properties. This way, experiments can be done on a much smaller set of

ligands in order to identify the right compound, saving significant time and resources in drug discovery.

If a protein's crystal structure is not known, we can use homology modeling [104] and structure refinement techniques to predict the structure of the target protein. The target binding site is determined either by experimentally identified co-crystal structures or by computational techniques that examine the surface of the fused-sphere model of the protein [64]. For each ligand examined, a large number of conformations are generated and the binding energy of the ligand-receptor complex is calculated in order to rank the ligands, as was explained in section 4.3.2.

In this work, we examine 11 proteins and a set of 37 co-crystals, listed in Table 4, and a screening library consisting of 10037 ligands against each protein. These structures were used in a comparative study of several algorithms for flexible ligand docking [105] which allow us to directly compare our results to other methods.

Table 4. List of proteins and co-crystal complexes examined.

Target protein (nickname)	**Complex PDB names**
Intestinal FABP (fab)	1icm
	1icn
	2ifb
Neuraminidase (nad)	1nsc
	1nsd
	1nnb
Penicillopepsin (pep)	1apt

	1apu
ε-Thrombin (ret)	1etr
	1ets
	1ett
Ribonuclease T_1 (rib)	1gsp
	1rhl
	1rls
L-Arabinose binding protein (ara)	1abe
	1abf
	5abp
Carbonic Anhydrase II (cah)	1cil
	1okl
	1cnx
Carboxypeptidase A (car)	1cbx
	3cpa
	6cpa
Cytochrome $P\text{-}450_{cam}$ (cyt)	1phf
	1phg
	2cpp
Thermolysin (tmn)	3tmn
	5tln
	6tmn
Trypsin (trp)	3ptb
	1tng
	1tni
	1tnj
	1tnk
	1tnl
	1tpp
	1pph

Since the best binders for each protein are already known (the 37 co-crystals) the goal in this test is to search the 10037 ligand database and identify the co-crystal ligands for each protein, as shown in Table 4, as the best binders. Because of the complexity of the problem we will consider as "successful" binders the top 2% of the ligands, as they are ranked according to their binding energy with each protein. Then, the success of the method will depend on how many of the co-crystal ligands show on the top 2% of the ranked 10037 ligand database.

We expect that the water solvent effects are very important for the right ranking of the ligands and it is important that we can include these effects accurately in the calculations. In order to assess the quality of the AVGB-SAS results in virtual ligand screening applications, we will perform the database search with and without solvation and then compare the number of the co-crystal ligands that are in the top 2% for every protein with each method. The database searching protocol we used utilizes both Dock [97] and MPSim [96] and it consists of the following steps:

- Defining the docking region.

The docking regions for each target protein was defined based on the superposition of the co-crystal structures available for that target.

- Protein grid calculation.

Dock uses an energy grid for the protein contribution to the interaction energy. This grid is calculated once per target.

- Level 0.

Docked conformations for each ligand were generated using flexible docking and energy scoring with minimization in Dock. The best 50 conformations for each ligand were saved and used in the subsequent steps.

- Filter.

A combined criterion of exposed surface area and binding energy score from Dock is applied to each of the 50 conformations per ligand from level 0. The best 5 conformations are carried to the next step.

- Level 1.

Minimization in vacuum for 25 steps with fixed protein coordinates using MPSim and the Dreiding forcefield [90] is performed for each of the 5 conformations per ligand that passed the filter. The best conformation per ligand is carried to the next step.

- Ranking.

The protein-ligand affinities are calculated based on the energy of the best conformation from level 1, corrected by the energy of the free ligand in the initial (undocked) conformation. The ligand list is then sorted by binding affinities and the ranking of the corresponding co-crystal ligands is recorded. This step is done twice, once with solvation using AVGB-SAS and once in vacuum. The binding energy when solvation is included is given by equation (85).

The results from the protocol, for the solvation and vacuum screening of the 10037 ligands, along with the predictions from Dock [97], FlexX [107] and ICM [108], as reported in [105] are shown in Table 5.

Table 5. Comparison of VLS results from searching the 10037 ligand database of [105], using the protocol described, with and without solvation, along with the results from Dock, FlexX and ICM. The shaded entries identify the co-crystal ligands that rank in the top 2%.

Co-crystal	Vacuum	Solvent	DOCK	FlexX	ICM
fab					
1icm	2.85	1.67	10.1	68.8	2.6
1icn	5.21	1.16	3.4	74.4	0.5
2ifb	1.21	0.12	7.1	99.3	0.5
nad					
1nnb	0.26	0.02	6.7	1.4	0.43
1nsc	0.29	0.05	19	7.2	5.01
1nsd	0.16	0.13	9.4	3	0.44
pep					
1apt	0.12	0.07	0.3	41	1.1
1apu	34.74	1.53	3.8	80	12.8
ret					
1etr	4.93	0.28	5.6	0.5	1.6
1ets	1.29	0.03	0.4	0.3	6.7
1ett	13.35	3.02	0.9	9	0.06
rib					
1gsp	0.62	0.25	4.4	10.7	0.6
1rhl	0.64	0.66	5.6	10.2	0.4
1rls	0.56	0.19	68.9	7.4	12.3
ara					
1abe	6.88	0.75	0.5	0.2	0.01
1abf	6.15	0.15	1.2	0.2	0.03
5abp	6.85	0.23	0.2	0.4	0.02
cah					
1cil	50.30	18.53	35	3.7	71.7
1okl	72.31	30.52	18.5	19.3	30.9

1cnx	87.85	33.92	0.3	14.5	90.6
car					
1cbx	0.18	0.10	36.8	2.7	0.2
3cpa	22.13	6.23	14.2	1.4	0.3
6cpa	0.04	0.03	8.5	2.7	3.4
1phf	14.44	7.39	26.9	13.2	3.7
1phg	29.72	21.51	60.6	1.4	12.7
2cpp	13.99	7.11	7.3	12.6	11.7
tmn					
3tmn	76.95	5.59	34.3	3.5	2.8
5tln	36.62	49.58	66	0.6	28.5
6tmn	60.81	48.49	15	8.5	77.6
trp					
3ptb	7.21	3.42	20.3	14	0.1
1tng	20.83	31.00	28.7	30.8	0.3
1tni	43.95	65.80	43.6	38.5	0.6
1tnj	44.84	67.75	45.1	47.9	1.1
1tnk	44.24	64.00	58.1	37.1	14.6
1tnl	18.71	21.78	82.9	39.1	51.6
1tpp	2.34	0.41	4	0.3	0.4
1pph	1.27	0.10	37	0.6	0.3

The entries in Table 5 correspond to the percentage of ligands in the 10037-compound database that score better than the corresponding co-crystal ligand for the respective protein. Shaded entries identify the co-crystals for which the respective docking method ranked it in the top 2%. Such rankings are considered successful because the co-crystal ligand is by definition a top binder to its receptor. Clearly, a good VLS method should identify most of the co-crystal ligands in the top 2%, if not all of them. The protocol described above, without solvation, identifies only 12 out of the 37 co-

crystal ligands as top binders. In contrast, when solvation is included, using AVGB-SAS, 20 of the co-crystal ligands are in the top 2%. The best other VLS methods in the literature, Dock, FlexX and ICM score 7, 11 and 20 co-crystal ligands in the top 2% respectively. These results clearly demonstrate the importance of the solvation effects in VLS and the success of the AVGB-SAS model for describing such effects.

Of course, solvation is only a part of the whole VLS procedure, although an important one. For a problem of such complexity, there are many more issues that have to be resolved, such as a better scoring function that does not depend just on the binding energy, inclusion of entropic effects that have been neglected in this study, more efficient and accurate ligand conformation search, refinement of the forcefield parameters used and CPU time efficiency. The results we have achieved here are very promising but more work has to be done in order for VLS methods to be unambiguously accepted. Due to the importance of these methods in the biosciences and the pharmaceutical industry, it is inevitable that the obstacles will eventually be overcome.

6 Bibliography

[1] Venter J. C. *et al.*, *Science*, **291**, 1304 (2001)

[2] Allen M. P., Tildesley D. J., *Computer Simulation of Liquids*, Oxford Science Publications (1987)

[3] Marcus Y., *The properties of Solvents*, Wiley, Chichester (1998)

[4] Orozco M., Luque F. J., *Chem. Rev.*, **100**, 4187 (2000)

[5] Ben-Naim A., *J. Phys. Chem.*, **82**, 792 (1978)

[6] Cramer C. J., Truhlar D. G., *Chem. Rev.*, **99**, 2161 (1999)

[7] Jovin T. M., Soumpasis D. M., McIntosh L. P., *Annu. Rev. Phys. Chem.*, **38**, 521 (1987)

[8] Cohen-Tannoudhi C., Diu B., Laloe F., *Quantum Mechanics*, Wiley Interscience Publications (1977)

[9] Eisenberg D., McLachlan A. D., *Nature*, **319**, 199 (1986)

[10] Wesson L., Eisenberg D., *Protein Science*, **1**, 227 (1992)

[11] Lee B., Richards F. M., *J. Mol. Biol.*, **55**, 379 (1971)

[12] Gogonea V., Baleanu C., Osawa E., *J. Molec. Struct. (THEOCHEM)*, **432**, 177 (1998)

[13] Vila J., Williams R. L., Vasquez M., Scheraga H. A., *PROT. Struct. Funct. Gen.*, **10**, 199 (1991)

[14] Gilson M. K., Honig B., *J. Compu. Aid. Mol. Des.*, **5**, 5 (1991)

[15] Stouten P. F. W., Frommel C., Nakamura H., Sander C., *Mol. Sim.*, **10**, 97 (1993)

[16] Nakamura H., *Q. Rev. Biophys.*, **29**, 1 (1996)

[17] Warshel A., Aqvist J., *Ann. Rev. Biophys. Chem.*, **20**, 267 (1991)

[18] Honig B., Sharp K., Yang A., *J. Phys. Chem.*, **97**, 1101 (1993)

[19] McQuarrie D. A., *Statistical Mechanics*, Harper & Row, NY (1976)

[20] Tanford C., Kirkwood J.G., *J. Am. Chem. Soc.*, **79**, 5333 (1957)

[21] Jayaram B., *J. Phys. Chem.*, **98**, 5773 (1994)

[22] Gilson M. K., Honig B., *Proteins*, **4**, 7 (1988)

[23] Davis M. E., McCammon J. A., *J. Comput. Chem.*, **10**, 386 (1989)

[24] Cortis C. M., Friesner R. A., *J. Comput. Chem.*, **18**, 1591 (1997)

[25] Dillet V., Rinaldi D., Rivail J.L., *J. Phys. Chem.*, **98**, 5034 (1996)

[26] Davis M. E., *J. Chem. Phys.*, **100**, 5149 (1994)

[27] Bliznyuk A. A., Gready J. E., *J. Phys. Chem.*, **99**, 14506 (1995)

[28] Mehler E. L., Guarnieri F., *Biophys. J.*, **75**, 3 (1999)

[29] Hassan S. A., Guarnieri F., Mehler E. L., *J. Phys. Chem. B*, **104**, 6478 (2000)

[30] Lorentz H. A., *Theory of Electrons*, Dover, NY (1952)

[31] Ehrenson S., *J. Comp. Chem.*, **10**, 77 (1989)

[32] Srinivasan J., Trevathan M. W., Beroza P., Case D. A., *Theor. Chem. Acc.*, **101**, 426 (1999)

[33] Schaefer M., Bartels C., Karplus M., *Theor. Chem. Acc.*, **101**, 194 (1999)

[34] Edinger S. R., Cortis C., Shenkin P. S., Friesner R. A., *J. Phys. Chem. B*, **101**, 1190 (1997)

[35] Zhang L. Y., Gallichio E., Friesner R. A., Levy R. M., *J. Comp. Chem.*, **22**, 591 (2001)

[36] Russel S. T., Warshel A., *J. Mol. Biol.*, **185**, 389 (1985)

[37] Klamt A., Schuurmann G., *J. Chem. Soc. Perk. Trans.*, **2**, 799 (1993)

[38] Born M., *Z. Phys.*, **1**, 45 (1920)

[39] Noyes R. M., *J. Am. Chem. Soc.*, **84**, 513 (1962)

[40] Bucher M., Porter T. L., *J. Phys. Chem.*, **90**, 3406 (1986)

[41] Babu C. S., Lim C., *J. Chem. Phys.*, **114**, 889 (2001)

[42] Still W. C., Tempczyk A., Hawley R. C., Hendrickson T., *J. Am. Chem. Soc.*, **112**, 6127 (1990)

[43] Wang J., Wang W., Huo S., Lee M., Kollman P., *J. Phys. Chem. B.*, **105**, 5055 (2001)

[44] Jayaram B., Liu Y., Beveridge D. L., *J. Chem. Phys.*, **109**, 1465 (1998)

[45] Schaefer M., Froemmel C., *J. Mol. Biol.*, **216**, 1045 (1990)

[46] Jackson J. D., *Classical Electrodynamics*, Wiley, NY (1975)

[47] Bashford D., Case D. A., *Annu. Rev. Phys. Chem.*, **51**, 129 (2000)

[48] Scarci M., Apostolakis J., Caflisch A., *J. Phys. Chem. A*, **101**, 8098 (1997)

[49] Qiu D., Shenkin P. S., Hollinger F. P., Still W. C., *J. Phys. Chem. A*, **101**, 3005 (1997)

[50] Dominy B. N., Brooks III C. L., *J. Phys. Chem. B*, **103**, 3765 (1999)

[51] Hawkins G. D., Cramer C. J., Truhlar D. G., *Chem. Phys. Let.*, **246**, 122 (1995)

[52] Hawkins G. D., Cramer C. J., Truhlar D. G., *J. Phys. Chem.*, **100**, 19824 (1996)

[53] Tsui V., Case D. A., *J. Am. Chem. Soc.*, **122**, 2489 (2000)

[54] Onufriev A., Bashford D., Case D. A., *J. Phys. Chem. B*, **104**, 3712 (2000)

[55] Ghosh A., Rapp C. S., Friesner R. A., *J. Phys. Chem. B*, **102**, 10983 (1998)

[56] Richards F. M., *J. Mol.Biol.*, **82**, 1 (1974)

[57] Schaefer M., Karplus M., *J. Phys. Chem.*, **100**, 1578 (1996)

[58] Schaefer M., Bartels C., Karplus M., *J. Mol. Biol.*, **284**, 835 (1998)

[59] Kratky K. W., *J. Statist. Phys.*, **25**, 619 (1981)

[60] Gibson K. D., Scheraga H. A., *Mol. Phys.*, **62**, 1247 (1987)

[61] Petitjean M., *J. Comp. Chem.*, **15**, 507 (1994)

[62] Aurenhammer F., *ACM Comp. Surv.*, **23**, 345 (1991)

[63] Aurenhammer F., *SIAM J. Comp.*, **16**, 78, (1987)

[64] Liang J., Edelsbrunner H., Fu P., Sudhakar S. V., Subramaniam S., *Proteins*, **33**, 1 (1998)

[65] Preparata F. P., Shamos M. I., *Computational Geometry*, Springer-Verlag, NY (1985)

[66] Edelsbrunner H., *Discrete Comput. Geom.*, **13**, 415 (1994)

[67] Dodd L. R., Theodorou D. N., *Mol. Phys.*, **72**, 1313 (1991)

[68] Irisa M., *Comp. Phys. Comm.*, **98**, 317 (1996)

[69] Eisenhaber F., Lijnzaad P., Argos P., Sander C., Scharf M., *J. Comp. Chem.*, **16**, 273 (1995)

[70] Connolly M. L., *J. Appl. Cryst.*, **16**, 548 (1983)

[71] Richmond T. J., *J. Mol. Biol.*, **178**, 63 (1984)

[72] DoCarmo M. D., *Differential Geometry of Curves and Surfaces*, Prentice-Hall, NJ (1976)

[73] Frazkiewicz R., Braun W., *J. Comp. Chem.*, **19**, 319 (1998)

[74] Nakahara M., *Geometry, Topology and Physics*, Institute of Physics Publishing, Bristol (1995)

[75] Preparata F. P., Muller D. E., *Theor. Comput. Sci.*, **8**, 45 (1979)

[76] Dantzig G. B., *Linear Programming and Extensions*, Princeton University Press, Princeton, NJ (1963)

[77] Press H. W., Teukolsky S. A., Vetterling W. T., Flannery B. P., *Numerical Recipes in C*, Cambridge University Press, Cambridge (1988)

[78] Seidel R., *Discrete Comput. Geom.*, **6**, 423 (1991) (Note: instead of $g_k = g_{d+2} = -1$ at page 433, it should be $g_k = -1$ and $g_{d+2} = +1$. Also, at the same page, instead of $(g - (1/a_k)a)$ it should be $(g + (1/a_k)a)$.)

[79] O'Rourke J., *Computational Geometry in C*, Cambridge University Press, New York (1993)

[80] Barber C. B., Dobkin D. P., Huhdanpaa H., *ACM Trans. Math. Soft.*, **22**, 469 (1996)

[81] Kozen D. C., *The Design and Analysis of Algorithms*, Springer-Verlag (1992)

[82] Kernighan B. W., Ritchie D. M., *The C Programming Language*, Prentice-Hall, NJ (1988)

[83] Wawak R. J., Gibson K. D., Scheraga H. A., *J. Math. Chem.*, **15**, 207 (1994)

[84] Emiris I., Canny J., *Proceedings 8th Annual Computational Geometry, Berlin, Germany*, 74 (1992)

[85] Eisenhaber F., Argos P., *J. Comp. Chem.*, **14**, 1272 (1993)

[86] Halperin D., Shelton C. R., *Comp. Geom.*, **10**, 273 (1998)

[87] Clarkson K. L., *Proceedings 33rd Annual IEEE Symposium on Foundations of Computer Science*, 387 (1992)

[88] Shewchuk J. R., *Discrete Comput. Geom.*, **18**, 305 (1997) (Note: public domain software implementation is available at http://www.cs.cmu.edu/~quake/robust.html)

[89] Halperin D., Overmars M. H., *Comp. Geom.*, **11**, 83 (1998)

[90] Mayo S. L., Olafson B. D., Goddard III W. A., *J. Phys. Chem.*, **94**, 8897 (1990)

[91] Cox S., Williams D., *J. Comput. Chem.*, **2**, 304 (1981)

[92] Ringnalda M., Langlois J. M, Greeley R., Mainz D., Wright J., Pollard W. T., Cao Y., Won Y., Miller G., Goddard III G. A., Friesner R. A., *Schrodinger Inc.* (1994)

[93] MacKerell, A.D. *et al.*, *J. Phys. Chem. B*, **102**, 3586 (1998)

[94] Tannor J. D., Marten B., Murphy R., Friesner R. A., Sitkoff D., Nicholls A., Ringnalda M., Goddard III G. A., Honig B., *J. Am. Chem. Soc.*, **116**, 11875 (1994)

[95] Blanco M., California Institute of Technology, Private Communication, Unpublished Data (2001)

[96] Lim K. T., Brunett S., Iotov M., McClurg R. B., Vaidehi N., Dasgupta S., Taylor S., Goddard III W. A., *J. Comp. Chem.*, **18**, 501 (1997)

[97] Ewing T. A., Makino S., Skillman A. G., Kuntz I. D., *J. Comp. Aid. Mol. Des.*, **15**, 411 (2001)

[98] Nichols B., Buttlar D., Farrell J. P., *Pthreads Programming*, O'Reilly & Associates (1996)

[99] Ding H. Q., Karasawa N., Goddard III W. A., *J. Chem. Phys.*, **97**, 4309 (1992)

[100] Cheatham III T. E., Kollman P. A., *Annu. Rev. Phys. Chem.*, **51**, 435 (2000)

[101] Arnott S., Hukins D. W. L., *Biochem. Biophys. Res. Communs.*, **47**, 1504 (1972)

[102] Cornell W. D., Cieplak P., Bayly C. I., Gould I. R., Merz K. M., Ferguson D. M., Spellmeyer D. C., Fox T., Caldwell J. W., Kollman P. A., *J. Am. Chem. Soc.*, **117**, 5179 (1995)

[103] Ulyanov N. B., James T. L., *Methods Enzymol.*, **261**, 90 (1995)

[104] Baker D., Sali A., *Science*, **294**, 93 (2001)

[105] Bursulaya B. D., Totrov M., Abagyan R., Brooks III C. L., *J. Med. Chem.*, (submitted)

[106] Vaidehi N., Floriano W. B., Goddard III W. A., California Institute of Technology, Unpublished Method (2000)

[107] Rarey M., Kramer B., Lengauer T., Klebe G., *J. Mol. Biol.*, **261**, 470 (1996)

[108] Totrov M., Abagyan R., *Proteins*, **Suppl. 1**, 215 (1997)

Appendix

Small Molecule List

The small organic molecules used for the validation of the AVGB model and the calculation of the surface tension parameters are listed in Table 6. The experimental solvation energies in water were given in [43] and the references therein. The list of molecules is selected so that it encompasses almost all possible organic groups. Each molecule is classified to a group category according to the dominating atomic group:

1 – Hydrocarbons

2 – Oxygenated

3 – Nitrogenated

4 – Fluorinated

5 – Halogenated (excluding F)

6 – Phosphated or Sulphated

Table 6 shows each compound, its group and experimental solvation energy in water and the predicted solvation energy from AVGB-SAS with expanded solvation types or element types, as described in section 4.2.

Table 6. List of small organic compounds with the experimental solvation energies in water from reference [43] and the predicted solvation energies from the AVGB-SAS model, in Kcal/Mol.

No.	Compound Name	Group	Experiment	Expanded types	Element types
1	methane	1	1.999	0.954	0.683
2	ethane	1	1.830	1.318	0.907
3	propane	1	1.954	1.332	0.815
4	butane	1	2.078	1.371	0.752

5	2-methylpropane	1	2.317	1.140	0.554
6	pentane	1	2.331	1.726	1.006
7	2-methylbutane	1	2.381	1.302	0.643
8	2,2-dimethylpropane	1	2.498	0.578	-0.055
9	hexane	1	2.484	1.816	0.994
10	2-methylpentane	1	2.522	1.599	0.850
11	3-methylpentane	1	2.508	1.774	1.048
12	2,2-dimethylbutane	1	2.594	1.080	0.389
13	heptane	1	2.618	1.866	0.971
14	2,2-dimethylpentane	1	2.876	1.245	0.458
15	octane	1	2.890	2.253	1.284
16	2,2,4-trimethylpentane	1	2.849	1.069	0.261
17	2,2,5-trimethylhexane	1	2.720	1.256	0.333
18	cyclopropane	1	0.748	-0.548	-0.963
19	cyclopentane	1	1.199	1.581	0.899
20	cyclohexane	1	1.228	1.729	0.973
21	methylcyclopentane	1	1.595	1.287	0.524
22	cycloheptane	1	0.795	1.837	1.017
23	methylcyclohexane	1	1.705	1.537	0.714
24	cyclooctane	1	0.855	1.911	1.057
25	1,2-dimethylcyclohexane	1	1.581	1.475	0.614
26	ethene	1	1.271	0.026	0.244
27	propene	1	1.268	0.345	0.308
28	1-butene	1	1.378	0.173	0.229
29	2-methylpropene	1	1.163	0.375	0.115
30	1-pentene	1	1.662	0.288	0.209
31	2-pentene	1	1.335	0.807	0.607
32	2-methyl-2-butene	1	1.309	0.873	0.411
33	3-methyl-1-butene	1	1.827	0.324	0.158
34	1-hexene	1	1.677	0.514	0.332
35	2-methyl-1-pentene	1	1.469	0.414	0.121
36	4-methyl-1-pentene	1	1.908	0.435	0.277
37	2-heptene	1	1.662	1.137	0.695
38	1-octene	1	2.169	0.926	0.569
39	cyclopentene	1	0.559	0.610	0.809
40	cyclohexene	1	0.368	0.481	0.542
41	1-methylcyclohexene	1	0.669	0.682	0.492

42	butadiene	1	0.614	-0.663	-0.347
43	1,4-pentadiene	1	0.941	-0.548	-0.163
44	2-methylbutadiene	1	0.681	-0.334	-0.258
45	1,5-hexadiene	1	1.008	-0.812	-0.346
46	2,3-dimethylbutadiene	1	0.396	0.293	0.136
47	1,3,5-cycloheptatriene	1	-0.989	-1.594	-1.151
48	ethyne	1	-0.014	-0.044	-1.187
49	propyne	1	-0.306	0.041	-0.735
50	1-butyne	1	-0.162	0.038	-0.877
51	1-pentyne	1	0.014	0.129	-0.895
52	1-hexyne	1	0.287	0.451	-0.680
53	1-heptyne	1	0.600	0.536	-0.700
54	1-octyne	1	0.709	0.643	-0.699
55	1-nonyne	1	1.051	0.893	-0.554
56	1-butene-3-yne	1	0.041	-0.437	-0.808
57	benzene	1	-0.865	-1.355	-1.011
58	methylbenzene	1	-0.886	-1.164	-1.074
59	ethylbenzene	1	-0.795	-1.108	-0.925
60	1,2-dimethylbenzene	1	-0.900	-0.769	-0.873
61	1,3-dimethylbenzene	1	-0.836	-0.952	-1.116
62	1,4-dimethylbenzene	1	-0.805	-0.739	-0.903
63	propylbenzene	1	-0.533	-0.938	-0.898
64	(1-methylethyl)benzene	1	-0.301	-0.998	-1.033
65	1,2,4-trimethylbenzene	1	-0.860	-0.317	-0.676
66	butylbenzene	1	-0.396	-1.202	-1.256
67	(1-methylpropyl)benzene	1	-0.449	-0.847	-0.978
68	(1,1-dimethylethyl)benzene	1	-0.437	-1.158	-1.357
69	(1,1-dimethylpropyl)benzene	1	-0.177	-0.950	-1.212
70	biphenyl	1	-2.642	-2.527	-2.019
71	1,1-methylenebisbenzene	1	-2.814	-3.386	-2.806
72	fluorene	1	-3.442	-2.818	-2.285
73	naphthalene	1	-2.391	-2.325	-1.891
74	1-methylnaphthalene	1	-2.367	-1.850	-1.656
75	1-ethylnaphthalene	1	-2.393	-1.852	-1.579
76	1,3-dimethylnaphthalene	1	-2.472	-1.518	-1.584
77	1,4-dimethylnaphthalene	1	-2.816	-1.134	-1.181
78	2,3-dimethylnaphthalene	1	-2.780	-1.490	-1.500

79	2,6-dimethylnaphthalene	1	-2.627	-1.545	-1.625
80	acenaphthene	1	-3.146	-2.560	-2.005
81	anthracene	1	-4.228	-2.891	-2.372
82	phenanthrene	1	-3.948	-2.889	-2.387
83	pyrene	1	-4.462	-2.925	-2.399
84	methanol	2	-5.111	-5.502	-3.077
85	ethanol	2	-5.011	-5.250	-3.071
86	1-propanol	2	-4.822	-5.201	-3.171
87	2-propanol	2	-4.753	-4.405	-2.760
88	1-butanol	2	-4.712	-4.729	-3.140
89	2-methyl-1-propanol	2	-4.521	-5.221	-3.079
90	2-butanol	2	-4.574	-3.721	-2.571
91	2-methyl-2-propanol	2	-4.512	-4.542	-3.083
92	1-pentanol	2	-4.471	-4.389	-2.899
93	3-methyl-1-butanol	2	-4.419	-5.199	-3.272
94	2-pentanol	2	-4.390	-4.878	-2.873
95	3-pentanol	2	-4.352	-4.748	-2.889
96	2-methyl-2-butanol	2	-4.428	-3.727	-2.703
97	1-hexanol	2	-4.361	-4.623	-3.219
98	2,3-dimethyl-2-butanol	2	-3.912	-3.625	-2.907
99	3-hexanol	2	-4.072	-3.070	-2.047
100	4-methyl-2-pentanol	2	-3.736	-4.708	-3.042
101	2-methyl-3-pentanol	2	-3.884	-3.558	-2.533
102	2-methyl-2-pentanol	2	-3.927	-3.638	-2.771
103	1-heptanol	2	-4.242	-2.841	-2.273
104	4-heptanol	2	-4.003	-4.020	-2.828
105	1-octanol	2	-4.091	-3.229	-2.410
106	2-propene-1-ol	2	-5.030	-6.056	-3.242
107	cyclopentanol	2	-5.491	-4.403	-3.008
108	cyclohexanol	2	-5.472	-4.284	-2.915
109	cycloheptanol	2	-5.482	-4.159	-2.905
110	phenol	2	-6.611	-6.271	-3.947
111	2-methylphenol	2	-5.871	-5.278	-3.772
112	4-methylphenol	2	-6.131	-5.902	-3.831
113	4-(1,1-dimethylethyl)phenol	2	-5.921	-5.967	-4.180
114	methyl	2	-1.894	-1.822	-1.754
115	ethyl	2	-1.634	-1.051	-1.374

116	1-methoxypropane	2	-1.662	-1.477	-1.760
117	2-methoxypropane	2	-2.004	-1.171	-1.469
118	1-ethoxypropane	2	-1.813	-1.056	-1.482
119	2-methoxy-2-methylpropane	2	-2.209	-1.584	-1.965
120	propylether	2	-1.154	-0.235	-0.955
121	diisopropylether	2	-0.533	-0.372	-0.993
122	butylether	2	-0.831	-0.174	-0.976
123	tetrahydrofuran	2	-3.468	-1.983	-2.043
124	tetrahydro-2H-pyran	2	-3.122	-1.753	-1.976
125	2-methyltetrahydrofuran	2	-3.301	-1.721	-1.998
126	2,5-dimethyltetrahydrofuran	2	-2.919	-1.137	-1.625
127	methoxybenzene	2	-2.398	-2.806	-2.472
128	1,2-dimethoxybenzene	2	-3.798	-3.389	-4.451
129	1,2,3-trimethoxybenzene	2	-5.396	-2.893	-4.775
130	methylamine	3	-4.560	-4.093	-4.016
131	ethylamine	3	-4.500	-3.847	-3.664
132	propylamine	3	-4.388	-4.202	-4.092
133	butylamine	3	-4.292	-3.764	-3.760
134	pentylamine	3	-4.094	-3.620	-3.722
135	hexylamine	3	-4.029	-3.748	-3.946
136	dimethylamine	3	-4.283	-3.973	-2.231
137	diethylamine	3	-4.065	-2.159	-1.169
138	dipropylamine	3	-3.657	-2.870	-2.067
139	dibutylamine	3	-3.325	-1.804	-1.284
140	aziridine	3	-5.412	-6.978	-4.764
141	azetidine	3	-5.553	-5.553	-3.373
142	pyrrolidine	3	-5.479	-4.178	-2.206
143	piperidine	3	-5.107	-4.081	-2.199
144	hexahydro-2H-azepine	3	-4.906	-4.013	-2.273
145	octahydroazocine	3	-4.402	-3.215	-1.542
146	trimethylamine	3	-3.236	-3.508	-0.717
147	triethylamine	3	-3.021	-0.688	0.043
148	1-methylpyrrolidine	3	-3.977	-3.382	-0.243
149	1-methylpiperidine	3	-3.891	-2.685	-0.528
150	pyridine	3	-4.696	-5.341	-4.656
151	2-methylpyridine	3	-4.629	-4.426	-4.168
152	3-methylpyridine	3	-4.770	-4.649	-4.274

153	4-methylpyridine	3	-4.932	-5.409	-5.001
154	2-ethylpyridine	3	-4.330	-4.149	-3.792
155	3-ethylpyridine	3	-4.600	-4.787	-4.340
156	4-ethylpyridine	3	-4.734	-5.445	-4.939
157	2,3-dimethylpyridine	3	-4.822	-3.713	-3.650
158	2,4-dimethylpyridine	3	-4.861	-4.459	-4.478
159	2,5-dimethylpyridine	3	-4.715	-3.711	-3.763
160	2,6-dimethylpyridine	3	-4.598	-3.437	-3.602
161	3,4-dimethylpyridine	3	-5.216	-4.781	-4.614
162	3,5-dimethylpyridine	3	-4.839	-3.934	-3.870
163	4-(t-butyl)pyridine	3	-4.462	-5.534	-5.437
164	2,6-bis(t-butyl)pyridine	3	-0.406	-2.057	-2.764
165	2-propanone	2	-3.970	-2.911	-4.134
166	2-butanone	2	-3.635	-2.776	-3.864
167	2-pentanone	2	-3.525	-2.701	-3.935
168	3-pentanone	2	-3.411	-2.884	-3.766
169	3-methyl-2-butanone	2	-3.239	-2.616	-3.764
170	2-hexanone	2	-3.287	-2.920	-4.131
171	4-methyl-2-pentanone	2	-3.060	-2.581	-3.781
172	2-heptanone	2	-3.038	-2.190	-3.613
173	4-heptanone	2	-2.923	-2.157	-3.321
174	2,4-dimethyl-3-pentanone	2	-2.737	-2.292	-3.389
175	2-octanone	2	-2.880	-2.039	-3.560
176	2-nonanone	2	-2.486	-1.842	-3.408
177	5-nonanone	2	-2.670	-2.388	-3.730
178	2-undecanone	2	-2.162	-1.452	-3.249
179	2-adamantanone	2	-4.457	-3.128	-4.598
180	acetophenone	2	-4.581	-3.881	-4.536
181	acetaldehyde	2	-3.501	-2.247	-3.922
182	propanal	2	-3.439	-2.295	-3.833
183	butanal	2	-3.174	-2.085	-3.750
184	pentanal	2	-3.029	-1.867	-3.639
185	hexanal	2	-2.809	-1.740	-3.615
186	heptanal	2	-2.670	-1.470	-3.427
187	octanal	2	-2.288	-1.317	-3.364
188	nonanal	2	-2.076	-1.183	-3.302
189	2-butenal	2	-4.223	-3.563	-4.559

190	2-hexenal	2	-3.678	-3.566	-4.610
191	2-octenal	2	-3.439	-3.276	-4.521
192	2,4-hexadienal	2	-4.631	-4.276	-5.194
193	benzaldehyde	2	-4.022	-4.069	-4.592
194	acetic	2	-6.700	-6.917	-5.939
195	propionic	2	-6.470	-6.569	-5.826
196	butanoic	2	-6.351	-7.314	-6.209
197	formic	2	-2.780	-5.012	-6.409
198	formic	2	-2.644	-4.351	-5.988
199	acetic	2	-3.313	-3.810	-4.671
200	formic	2	-2.482	-4.696	-6.258
201	formic	2	-2.018	-4.127	-5.788
202	acetic	2	-3.093	-3.632	-4.723
203	propionic	2	-2.931	-4.124	-5.018
204	formic	2	-2.221	-3.937	-5.787
205	acetic	2	-2.854	-4.049	-5.244
206	acetic	2	-2.644	-2.990	-4.100
207	propionic	2	-2.794	-3.317	-4.366
208	butanoic	2	-2.830	-3.447	-4.420
209	formic	2	-2.126	-4.157	-6.045
210	acetic	2	-2.548	-3.680	-5.001
211	formic	2	-2.357	-3.926	-5.775
212	propionic	2	-2.453	-3.259	-4.341
213	propionic	2	-2.221	-2.835	-3.970
214	butanoic	2	-2.494	-3.108	-4.292
215	pentanoic	2	-2.572	-3.531	-4.445
216	acetic	2	-2.453	-3.500	-4.902
217	acetic	2	-2.209	-3.574	-4.918
218	butanoic	2	-2.276	-4.216	-5.556
219	pentanoic	2	-2.522	-2.539	-3.811
220	hexanoic	2	-2.486	-3.713	-4.955
221	acetic	2	-2.262	-3.624	-5.138
222	propionic	2	-1.990	-3.466	-4.682
223	heptanoic	2	-2.302	-2.783	-4.100
224	octanoic	2	-2.037	-3.052	-4.273
225	benzoic	2	-4.280	-4.335	-4.753
226	acetamide	3	-9.704	-10.031	-8.500

227	fluoromethane	4	-0.220	-0.164	-0.399
228	chloromethane	5	-0.557	-0.898	-1.085
229	chloroethane	5	-0.628	-0.543	-0.855
230	1-chloropropane	5	-0.272	-0.632	-1.047
231	2-chloropropane	5	-0.246	-0.297	-0.704
232	1-chlorobutane	5	-0.136	-0.399	-0.893
233	1-chloropentane	5	-0.069	-0.273	-0.893
234	2-chloropentane	5	0.069	-0.256	-0.836
235	3-chloropentane	5	0.041	0.381	-0.202
236	chloroethene	5	-0.592	-0.547	-0.377
237	3-chloro-1-propene	5	-0.573	-1.300	-1.038
238	chlorobenzene	5	-1.120	-1.134	-0.846
239	bromomethane	5	-0.819	-1.113	-1.278
240	bromoethane	5	-0.695	-0.841	-1.130
241	1-bromopropane	5	-0.559	-0.978	-1.370
242	2-bromopropane	5	-0.478	-0.615	-0.996
243	1-bromobutane	5	-0.408	-0.853	-1.349
244	1-bromo-2-methylpropane	5	-0.026	-0.759	-1.202
245	1-bromo-2-methylbutane	5	0.205	-0.517	-1.033
246	bromobenzene	5	-1.459	-1.407	-1.108
247	1-bromo-4-methylbenzene	5	-1.390	-1.340	-1.295
248	1-bromo-2-ethylbenzene	5	-1.187	-1.175	-1.083
249	1-bromo-2-(1-methylethyl)benzene	5	-0.846	-1.081	-1.087
250	iodomethane	5	-0.886	-0.717	-0.516
251	iodoethane	5	-0.724	-0.472	-0.421
252	1-iodopropane	5	-0.585	-0.684	-0.738
253	2-iodopropane	5	-0.463	-0.291	-0.359
254	1-iodobutane	5	-0.258	-0.753	-0.909
255	methanethiol	6	-1.240	-1.359	-1.777
256	ethanethiol	6	-1.295	-0.901	-1.552
257	benzenethiol	6	-2.548	-2.934	-2.618
258	dimethylsulfide	6	-1.541	-2.028	-1.664
259	diethylsulfide	6	-1.431	-1.011	-1.121
260	methylthiobenzene	6	-2.728	-2.602	-1.875
261	acetonitrile	3	-3.884	-3.886	-6.213
262	propanenitrile	3	-3.843	-4.081	-6.230
263	butanenitrile	3	-3.642	-3.890	-6.124

264	nitroethane	3	-3.707	-3.932	-5.354
265	1-nitropropane	3	-3.339	-3.583	-5.136
266	2-nitropropane	3	-3.136	-3.172	-4.879
267	nitrobenzene	3	-4.115	-4.492	-5.159
268	1-methyl-2-nitrobenzene	3	-3.585	-4.397	-5.109
269	1-methyl-3-nitrobenzene	3	-3.449	-4.228	-5.149
270	1,2-ethanediol	2	-7.650	-8.683	-6.772
271	1,2,3-propanetriol	2	-9.210	-12.465	-8.982
272	dimethoxymethane	2	-2.931	-4.822	-4.523
273	1,2-dimethoxyethane	2	-4.832	-3.931	-3.770
274	1,1-diethoxyethane	2	-3.272	-1.628	-2.122
275	1,2-diethoxyethane	2	-3.530	-2.210	-2.645
276	1,3-dioxolane	2	-4.094	-4.560	-4.213
277	1,4-dioxane	2	-5.052	-4.683	-4.336
278	1,2-ethanediamine	3	-7.591	-9.420	-8.549
279	piperazine	3	-7.371	-9.852	-5.248
280	1-methylpiperazine	3	-7.770	-8.977	-3.778
281	1,4-dimethylpiperazine	3	-7.571	-8.348	-2.127
282	2-methylpyrazine	3	-5.515	-5.538	-5.014
283	2-ethylpyrazine	3	-5.453	-5.224	-4.629
284	2-(2-methylpropyl)pyrazine	3	-5.042	-5.218	-4.852
285	2-ethyl-3-methoxypyrazine	3	-4.392	-4.295	-3.888
286	2-(2-methylpropyl)-3-methoxypyrazine	3	-3.681	-4.252	-4.079
287	2-methoxyethanol	2	-6.762	-6.427	-5.333
288	2-ethoxyethanol	2	-6.602	-4.204	-3.389
289	2-propoxyethanol	2	-6.411	-6.131	-5.561
290	2-butoxyethanol	2	-6.263	-5.775	-5.309
291	2-methoxyethylamine	3	-6.542	-6.287	-5.914
292	3-methoxy-1-propylamine	3	-6.922	-6.133	-5.806
293	morpholine	3	-7.170	-6.525	-4.123
294	4-methylmorpholine	3	-6.332	-6.103	-2.615
295	1,1-difluoroethane	4	-0.110	-0.182	-0.494
296	dichloromethane	5	-1.404	-1.386	-1.490
297	1,1-dichloroethane	5	-0.846	-1.033	-1.252
298	1,1-dichlorobutane	5	-0.695	-0.685	-1.071
299	dibromomethane	5	-2.107	-1.352	-1.668
300	chlorofluoromethane	5	-0.774	-0.941	-1.088

301	1,2-dichloroethane	5	-1.732	-1.496	-1.710
302	1,2-dichloropropane	5	-1.254	-1.888	-2.203
303	1,2-dibromoethane	5	-2.099	-1.978	-2.147
304	1,2-dibromopropane	5	-1.935	-2.391	-2.670
305	1-bromo-2-chloroethane	5	-1.949	-1.745	-1.937
306	1,3-dichloropropane	5	-1.894	-1.658	-1.947
307	1,3-dibromopropane	5	-1.963	-3.001	-3.264
308	1,2-dichloroethene	5	-1.173	-0.753	-0.630
309	1,2-dichloroethene	5	-0.764	-0.749	-0.625
310	1,2-dichlorobenzene	5	-1.364	-0.987	-0.746
311	1,3-dichlorobenzene	5	-0.982	-0.917	-0.687
312	1,4-dichlorobenzene	5	-1.008	-0.885	-0.654
313	1,4-dibromobenzene	5	-2.302	-1.359	-1.104
314	trifluoromethane	4	0.805	0.710	0.031
315	trichloromethane	5	-1.065	-0.960	-0.986
316	1,1,1-trichloroethane	5	-0.246	-0.682	-0.814
317	tribromomethane	5	-2.126	-1.227	-1.206
318	chlorodifluoromethane	5	-0.497	0.006	-0.097
319	1,1,2-trichloroethane	5	-1.949	-1.855	-1.985
320	trichloroethene	5	-0.437	-0.377	-0.289
321	tetrafluoromethane	4	3.112	2.681	1.926
322	tetrachloromethane	5	0.096	-0.115	-0.064
323	chlorotrifluoromethane	5	2.522	2.409	1.890
324	dichlorodifluoromethane	5	1.691	1.275	1.260
325	bromotrifluoromethane	5	1.787	2.285	1.793
326	1,1,2,2-tetrachloroethane	5	-2.357	-1.780	-1.842
327	2-chloro-1,1,1-trifluoroethane	5	0.055	0.210	-0.384
328	tetrafluoroethene	4	1.376	2.367	2.283
329	tetrachloroethene	5	0.055	0.113	0.169
330	pentachloroethane	5	-1.364	-1.467	-1.466
331	hexachloroethane	5	-1.404	-0.892	-0.830
332	1,1,2,2-tetrachloro-1,2-difluoroethan	5	0.817	0.598	0.613
333	1,1,2-trichloro-1,2,2-trifluoroethane	5	1.772	1.331	1.317
334	1,1-dichlorotetrafluoroethane	5	2.508	2.022	1.596
335	1,2-dichloro-1,1,2,2-tetrafluoroethane	5	2.317	1.957	1.912
336	chloropentafluoroethane	5	2.864	2.599	2.123
337	1,1,2,3,3,3-hexafluoro-1-propene	4	2.310	1.755	1.239

338	4-bromophenol	5	-7.130	-6.504	-4.202
339	3-nitrophenol	3	-9.628	-9.187	-7.906
340	4-nitrophenol	3	-10.648	-9.476	-8.173
341	3-hydroxybenzaldehyde	2	-9.508	-8.691	-7.255
342	4-hydroxybenzaldehyde	2	-10.469	-9.187	-7.720
343	3-hydroxybenzonitrile	3	-9.659	-9.403	-8.599
344	4-hydroxybenzonitrile	3	-10.168	-9.870	-9.072
345	2-chloropyridine	5	-4.392	-4.663	-4.190
346	3-chloropyridine	5	-4.013	-5.107	-4.519
347	1,1-thiobis(2-chloroethane)	6	-3.917	-3.549	-3.465
348	2,2,2-trifluoroethanol	4	-4.304	-4.570	-3.421
349	1,1,1-trifluoropropan-2-ol	4	-4.156	-4.022	-3.006
350	2,2,3,3-tetrafluoropropan-1-ol	4	-4.875	-4.909	-3.776
351	2,2,3,3,3-pentafluoropropan-1-ol	4	-4.151	-3.716	-2.623
352	1,1,1,3,3,3-hexafluoropropan-2-ol	4	-3.767	-3.088	-2.314
353	N-methylacetamide	3	-10.201	-5.851	-6.583
354	N,N-dimethylacetamide	3	-8.658	-10.757	-5.225
355	guanine	3	-19.547	-19.693	-16.561
356	adenine	3	-12.602	-12.562	-16.544
357	thymine	3	-7.485	-11.634	-13.252
358	cytosine	3	-12.721	-12.432	-18.310
359	imidazole	3	-9.914	-10.237	-7.841
360	1-methylimidazole	3	-8.665	-7.830	-6.927
361	4-methylimidazole	3	-9.881	-9.237	-7.362
362	p-benzoquinone	2	-5.964	-1.311	-2.880
363	pyrrole	3	-4.927	-5.391	-3.896
364	1-methylpyrrole	3	-2.983	-4.149	-3.343
365	3-methylindole	3	-5.978	-5.792	-3.676
366	propylguanidine	3	-11.047	-11.158	-12.360
367	methylguanidine	3	-11.324	-11.212	-12.102
368	p-hydroquinone	2	-10.387	-11.039	-6.772
369	p-methoxyphenol	2	-8.606	-7.602	-5.296
370	o-nitrophenol	3	-5.023	-7.855	-6.111
371	m-nitrophenol	3	-10.144	-9.210	-7.929
372	p-nitrophenol	3	-11.916	-9.442	-8.139
373	propylphosphate	6	-6.286	-6.425	-7.466
374	ethylphosphate	6	-8.042	-7.463	-7.729

375	methylphosphate	6	-9.007	-9.409	-8.101
376	alanine_dipeptide	3	-16.958	-12.694	-10.551