POSSIM

POlarizable Simulations with Second order Interaction Model

Version 2.0

Software for geometry optimizations and Monte Carlo simulations of single molecules, gas-phase complexes, liquids and solutions

George A. Kaminski

Worcester Polytechnic Institute

Worcester, MA

email: gkaminski@wpi.edu

Contents

Introduction

POSSIM is a software suite designed for the following tasks:

- geometry optimizations – single molecules, gas-phase complexes or solvated systems (periodic boundary conditions can be applied);
- Monte Carlo simulations – single molecules, gas-phase complexes or solvated systems (again, periodic boundary conditions can be applied);
- Monte Carlo simulations with statistical perturbation theory (ΔG calculations).
- Geometry optimizations can be performed with the Fuzzy-Border continuum solvent model [1a] (in version 2.0 – for non-polarizable simulations only).

Each of the above tasks is performed with use of the internal coordinates (Z-matrices) for intramolecular atomic positions.

POSSIM can use an OPLS-like fixed-charges force field or a polarizable force field with the polarization represented by inducible point dipoles (POSSIM). There is also an option to use the second-order approximation to the polarization energy [1b], and this is, in fact, the main intended focus of the program, but the full-scale polarization can be employed as well.

At this point, POSSIM is implemented for UNIX and Linux binary files are included. The code should be portable enough to be compiled and run on non-UNIX platforms, but the author has not tested it this way.

Sample Jobs

Sample jobs are included in the distribution, and examples are present for all the POSSIM modules. The input files for these jobs can serve as templates for desired simulations. Each sample job directory has a README file with a brief description of the performed simulation and instructions on how to run the job.

Citations

If you publish a work in which POSSIM was used, please include one of the following references:

Kaminski, G. A.; Ponomarev, S. Y.; Lin. A. B. “Polarizable Simulations with Second-Order Interaction Model – Force Field and Software for Fast Polarizable Calculations: Parameters for Small Model Systems and Free Energy Calculations”, *J. Chem. Theory Comput.*, *5*, 2935-2943, **2009**.

Ponomarev, S. Y.; Kaminski, G. A. “Polarizable Simulations with Second-Order Interaction Model (POSSIM) Force Field: Developing Parameters for Alanine Peptides and Protein Backbone”, *J. Chem. Theory Comput.*, *7*, 1415-1427, **2011**.

If the Fuzzy-Border continuum solvent model is employed, please use the following reference instead:

Sharma, I.; Kaminski, G. A. “Calculating pKa Values for Substituted Phenols and Hydration Energies for Other Compounds with the First-Order Fuzzy-Border Continuum Solvation Model”, *J. Comput. Chem.*, *33*, 2388-2399, **2012**.

Also, please be advised that we use the L-BFGS-B optimizer in geometry optimizations, and we are reproducing here the following references as required by the authors:

[1] R. H. Byrd, P. Lu, J. Nocedal and C. Zhu, ``A limited memory algorithm for bound constrained optimization'', SIAM J. Scientific Computing 16 (1995), no. 5, pp. 1190--1208.

[2] C. Zhu, R.H. Byrd, P. Lu, J. Nocedal, ``L-BFGS-B: FORTRAN Subroutines for Large Scale Bound Constrained Optimization" Tech. Report, NAM-11, EECS Department, Northwestern University, 1994.

Website: http://users.eecs.northwestern.edu/~nocedal/lbfgsb.html

Likewise, the following is included to satisfy requirements set forth by the authors of the random number generator used in Monte Carlo modules of POSSIM:

A C-program for MT19937, with initialization improved 2002/1/26. Coded by Takuji Nishimura and Makoto Matsumoto.

Before using, initialize the state by using init_genrand(seed) or init_by_array(init_key, key_length). Copyright (C) 1997 - 2002, Makoto Matsumoto and Takuji Nishimura, All rights reserved. Copyright (C) 2005, Mutsuo Saito, All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The names of its contributors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Any feedback is very welcome. http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/emt.html email: m-mat@math.sci.hiroshima-u.ac.jp

---

FORTRAN77 translation by Tsuyoshi TADA. (2005/12/19)

Force Fields

OPLS-AA – type fixed charges force field.

The OPLS-AA is a fixed-charges force field developed and tested in Prof. W. L. Jorgensen group at Yale University [2]. The total molecular system energy E_{tot} is evaluated as a sum of the following components – the non-bonded energy E_{nn}, bond stretching and angle bending terms E_{bond} and E_{angle}, and the torsional energy $E_{torsion}$. The non-bonded part is computed as a sum of the Coulomb and Lennard-Jones contributions for pairwise intra- and intermolecular interactions:

$$E_{nb} = \sum_{i}^{onA} \sum_{j}^{onB} [q_i q_j e^2 / r_{ij} + 4\varepsilon_{ij}(\sigma_{ij}^{12} / r_{ij}^{12} - \sigma_{ij}^{6} / r_{ij}^{6})] f_{ij} \quad (1)$$

Geometric combining rules for the Lennard-Jones coefficients are employed: $\sigma_{ij}=(\sigma_{ii}\,\sigma_{jj})^{1/2}$ and $\varepsilon_{ij}=(\varepsilon_{ii}\,\varepsilon_{jj})^{1/2}$. The summation runs over all the pairs of atoms $i < j$ on molecules A and B or A and A for the intramolecular interactions. Moreover, in the latter case, the coefficient f_{ij} is equal to 0.0 for any *i-j* pairs connected by a valence bond (1-2 pairs) or a valence bond angle (1-3 pairs). f_{ij} = 0.5 for 1,4-interactions (atoms separated by exactly 3 bonds) and f_{ij} = 1.0 for all the other cases.

The bond stretching and angle bending energies are obtained in accordance with Equations 2 and 3.

$$E_{bond} = \sum_{bonds} K_r (r - r_{eq})^2 \quad (2)$$

$$E_{angle} = \sum_{angles} K_\Theta (\Theta - \Theta_{eq})^2 \quad (3)$$

Here the subscripts *eq* are used to denote the equilibrium values of the bond length *r* and angle Θ.

Finally, the torsional term is computed as follows:

$$E_{torsion} = \sum_i \frac{V_1^i}{2}[1 + \cos(f_i)] + \frac{V_2^i}{2}[1 - \cos(2f_i)] + \frac{V_3^i}{2}[1 + \cos(3f_i)] + \frac{V_4^i}{2}[1 - \cos(4f_i)], \quad (4)$$

with the summation performed over all the dihedral angles *i* with values f_i. .

Polarizable Force Field

The only difference here is that an additional E_{pol} tern is added to the total energy (of course, the values of the permanent charges and Lennard-Jones parameters can be very different, but the general formalism for them remains the same). It should be noted that the scaling factor for 1,4- interactions is still applied as before for Lennard-Jones and charge-charge interactions, but not for the charge-dipole or dipole-dipole interactions. Moreover, 1,2- and 1,3- pairs are excluded from charge-dipole but not from dipole-dipole interactions.

Electrostatic polarization, when take into account in the form of the induced point dipole model, leads to addition of the following component to the total energy expression for a molecular system:

$$E_{pol} = -\frac{1}{2}\sum_i \boldsymbol{\mu}_i \mathbf{E}_i^0 \tag{5}$$

Here $\boldsymbol{\mu}_i$ represents the induced dipole moment on the *i*th polarizable site and $\mathbf{E}_i^0$ stands for the electrostatic field produced by permanent charges only in the absence of the induced dipoles. The induced dipole moment depends on the total electrostatic field (produced by both the permanent charges and other dipoles) as shown in Equation 6.

$$\boldsymbol{\mu}_i = \boldsymbol{\alpha}_i \mathbf{E}_i^{tot} \tag{6}$$

where $\boldsymbol{\alpha}_i$ is the polarizability of the *i*th site. The total field $\mathbf{E}_i^{tot}$ is computed as follows:

$$\mathbf{E}_i^{tot} = \mathbf{E}_i^0 + \sum_{j \neq i} \mathbf{T}_{ij} \boldsymbol{\mu}_j \tag{7}$$

where

$$\mathbf{T}_{ij} = \frac{1}{R_{ij}^3}\left(\frac{3\mathbf{R}_{ij}\mathbf{R}_{ij}}{R_{ij}^2} - \mathbf{I}\right) \tag{8}$$

is the dipole-dipole interaction tensor, and **I** is the unit tensor. Thus,

$$\boldsymbol{\mu}_i = \boldsymbol{\alpha}_i \mathbf{E}_i^0 + \boldsymbol{\alpha}_i \sum_{j \neq i} \mathbf{T}_{ij} \boldsymbol{\mu}_j \tag{9}$$

or, with $\mathbf{A} = \boldsymbol{\alpha}^{-1}(\mathbf{I} - \boldsymbol{\alpha}\mathbf{T})$,

$$\mathbf{A}\boldsymbol{\mu} = \mathbf{E}^0 \tag{10}$$

and one has to solve a system of linear equations in order to determine the values of induced dipole moments $\boldsymbol{\mu}_i$ to be substituted into the Equation 1.

Unphysical growth of the induced dipoles at close distances to each other and to the permanent electrostatic charges can be avoided by introducing a cutoff procedure for small interactomic distances R_{ij}.[3]. In this program, the "perceived" of "effective" distance is modified if it is under a cutoff R^{cut}.

$$R_{ij}^{cut} = R_i^{cut} + R_j^{cut} \tag{11}$$

In this case,

$$R_{ij} = (1 - x^2 + x^3)R_{ij}^{cut}, \text{ where } x = R_{ij} / R_{ij}^{cut} \tag{12}$$

This way the effective distance used in calculating the polarization energy can never approach zero.

The system in Equation 10 can be solved with direct matrix inversion, with elimination method, or iteratively, when the left-hand side of the Equation 9 is calculated by substituting an initial guess for $\boldsymbol{\mu}$ into the right-hand side, and then the cycle is repeated until the desired level of self-consistency is achieved. The latter technique or the extended Lagrangian method are normally used in application to molecular systems, as they are by far less demanding in terms of the computational resources [4].

But even with the iterative solving method employed, the procedure is still CPU-time and memory consuming if the system in hand is large enough – for example, a condensed-phase one or a large biomolecule. Indeed, if one has to explicitly simulate, for example, 3000 polarizable sites, the matrix **A** in Equation 10 will have dimensions of 9,000 x 9,000 = 81,000,000 elements. And the CPU-time needed for the iterations will be quite significant. To avoid the problem, only a relatively small part of a larger system is usually treated as possessing explicit polarizable atomic sites. The rest of the atoms are treated differently – as having no polarizability at all, as a dielectric continuum medium, etc.[2] Moreover, most of liquid-phase molecular simulations with explicitly included atomic polarizabilities are performed with molecular dynamics rather than Monte Carlo technique. This is due to the fact that, in spite of its general computational simplicity, Monte Carlo with explicit polarization requires to solve the Equation 6 every time when even one molecule in the system is moved, and the number of configurations in an average Monte Carlo computation is by orders of magnitude greater than in a molecular dynamics run. Thus, employing Monte Carlo becomes much less practical for polarizable systems, even though it might be otherwise preferable.

In order to decrease the computational resources necessary to utilize the dipole polarization model, we employed an approximation described below. Equation 13 shows the iterative procedure which is, in fact, usually employed in solving Equation 10.

$$\boldsymbol{\mu}_i^{I} = \alpha_i \mathbf{E}_i^0 \tag{13a}$$

$$\boldsymbol{\mu}_i^{II} = \alpha_i \mathbf{E}_i^0 + \alpha_i \sum_{j \neq i} \mathbf{T}_{ij} \boldsymbol{\mu}_j^{I} = \alpha_i \mathbf{E}_i^0 + \alpha_i \sum_{j \neq i} \mathbf{T}_{ij} \alpha_j \mathbf{E}_j^0 \tag{13b}$$

$$\boldsymbol{\mu}_i^{III} = \alpha_i \mathbf{E}_i^0 + \alpha_i \sum_{j \neq i} \mathbf{T}_{ij} \boldsymbol{\mu}_j^{II} = \alpha_i \mathbf{E}_i^0 + \alpha_i \sum_{j \neq i} \mathbf{T}_{ij} \alpha_j \mathbf{E}_j^0 + \alpha_i \sum_{j \neq i} \mathbf{T}_{ij} \alpha_j \sum_{k \neq j} \mathbf{T}_{jk} \alpha_k \mathbf{E}_k^0 \tag{13c}$$

Doing the substitution infinitely many times produces the exact solution, but the procedure is normally stopped as soon as the changes in $\boldsymbol{\mu}$ between two iterations become sufficiently small.

Let us now consider the first- and the second-order approximations (Equations 13a and 13b, respectively). The energy is still computed according to the Equation 5. The first-order approximation has the physical meaning of using inducible dipoles, with magnitudes determined in assumption that they cannot interact with each other at all. This allows some non-additivity and thus many-body interactions to be included into the calculations. Other researchers have employed this approximation and found that it has a good level of computational efficiency, but allows only a limited improvement of accuracy compared to pairwise-additive non-polarizable force fields [5].

We use a different, higher level of the theory in this software. Our main objective is to utilize the more accurate second-order approximation from the Equation 13b, which, while retaining a greater part of the dipole-dipole interactions than the first-order approximation in Equation 8a, also provides the benefits of reduced computational cost compared to the full-scale polarization model. Indeed, if Equation 13b is used instead of the Equation 6, the time needed to find the dipole moments vector $\boldsymbol{\mu}$ from scratch is equal to the time needed for just one iteration in the full-scale point dipole method. And we will show below that the time needed for the iterations is the most time-consuming part of polarizable calculations. Thus, this approximation reduces computational cost dramatically.

It should be noted that the second-order approximation in Equation 13b does not have a direct physical meaning. It can be viewed as introducing a set of induced dipoles with magnitudes calculated in the assumption that each of them perceives all the other dipoles as if those other dipoles were induced by the electrostatic field of the permanent charges only.

Smoothing of Long-Range Charge-Charge Interactions

Interactions for distances greater than pre-set values are ignored. The charge-charge interactions can be quadraticaly switched to zero over the last 0.5 Å before the cutoff distance R_{max} (this is the recommended option). If the distance R between two charges i and j is $R_{max} - 0.5$ Å $< R < R_{max}$, then the energy of interaction between these two charges is:

$$E_{ij} = \frac{q_i \cdot q_j \cdot e^2}{R} \cdot \frac{{R_{max}}^2 - R^2}{{R_{max}}^2 - (R_{max} - 0.5\text{Å})^2} \tag{14}$$

This technique allows to avoid unnecessary noise which would otherwise emerge because of charges moving in and out of the cutoff sphere of radius Rmax.

Disregarding Interactions beyond R_{max}

It should be noted that all atoms are combined into groups for the purpose of using the cutoff distances R_{max}. There is a designated center atom in each of the groups. Interactions between an atom in one group and an atom in another group are included only if the distance between the center atoms of these groups is below R_{max}. There is also an option for every single atom of a group to serve as the central atom. In this case, if any of the atoms is within the R_{max} distance of the central atom (or atoms) of another group, interactions between all the atoms in the two groups are considered. There are separate R_{max} distances for solute-solute interactions (RmaxCxx), solvent-solvent interactions (RmaxCss), solute-solvent interactions (RmaxCxs), and intramolecular interactions (RmaxCi). Moreover, dipole-dipole interactions are not included for distances larger than RmaxD.

Long-Range Correction for Disregarded Lennard-Jones Interactions

To compensate for disregarding non-electrostatic Lennard-Jones interactions beyond the R_{max} distance, the following corrections can be (and usually is) added for the solute-solvent and solvent-solvent interactions:

$$U_{LRC} = \frac{1}{2} 4\pi\rho \int_{R_{max}}^{\infty} V_{LJ}(r) \cdot r^2 \cdot dr\,, \tag{15}$$

where r is the density of the particles and $V_{LJ}(r)$ is the Lennard-Jones potential.

Fuzzy-Border Continuum Solvent Model

This model is described in detail in Reference 1a. Briefly, it can be introduced as follows.

The Electrostatic Component of the Solvation Energy

The solvation energy is calculated as:

$$\Delta G(solv) = \Delta G(el) + \Delta G(np), \tag{16}$$

where the $\Delta G(el)$ and $\Delta G(np)$ terms stand for the electrostatic and the non-polar parts of the solvation energy, respectively. The electrostatic part of the energy was calculated by using an approximation to the Poisson-Boltzmann formalism. Briefly, let us consider the solute-solvent surface, as shown on Figure 1.

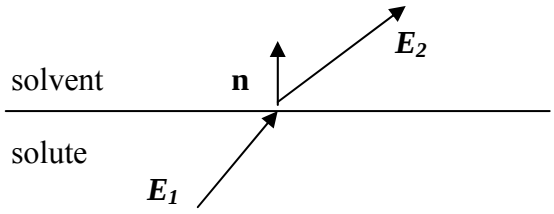

Figure 1. Electrostatic field at the solute-solvent interface.

Because of the continuity of the normal component of the electric displacement:

$$\varepsilon_1 \boldsymbol{E}_1 \cdot \boldsymbol{n} = \varepsilon_2 \boldsymbol{E}_2 \cdot \boldsymbol{n}, \tag{17}$$

where ε_1 and ε_2 are the dielectric constants inside and outside of the solute, respectively. If $\varepsilon_1 = 1$, then:

$$\boldsymbol{E}_1 \cdot \boldsymbol{n} = \varepsilon \boldsymbol{E}_2 \cdot \boldsymbol{n}, \tag{18}$$

According to the Gauss law,

$$4\pi\sigma = (\boldsymbol{E}_2 - \boldsymbol{E}_1) \cdot \boldsymbol{n}, \tag{19}$$

where σ is the surface charge density at the interface. Combining Equations 18 and 19,

$$\sigma = -\frac{1}{4\pi}(1 - 1/\varepsilon)\boldsymbol{E}_1 \cdot \boldsymbol{n} \tag{20}$$

Since the electrostatic field $\boldsymbol{E}_1$ itself depends on the surface charge density distribution, Equation 20 describes a self-consistent problem, just like in the general electrostatic polarization case. The electrostatic part of the solvation energy is:

$$\Delta G(el) = \frac{1}{2}\int_S \sigma\phi^0 d^2 r \tag{21}$$

Here ϕ^0 represents the electrostatic potential created by the charges of the solute only (not by the polarized solvent) and the integration is carried out over the solute-solvent interface.

When the equation is solved numerically, the surface is represented by a discrete set of points *i*. In this case, Equations 20 and 21 become:

$$q_i = -\frac{1}{4\pi}(1-1/\varepsilon)\boldsymbol{E}_{1,i}\cdot\boldsymbol{n_i} \tag{22}$$

$$\Delta G(el) = \frac{1}{2}\sum_i q_i\phi_i^0 \tag{23}$$

The electrostatic field $\boldsymbol{E}_{1,i}$ is calculated as:

$$\boldsymbol{E}_{1,i} = \sum_j \frac{q_j \boldsymbol{R}_{ij}}{R_{ij}^3} + \sum_{k\neq i}\frac{q_k \boldsymbol{R}_{ik}}{R_{ik}^3} \tag{24}$$

The first sum is taken over the solute charges (this expression can be easily extended to include higher-order multipoles), while the second one goes over the other solute-solvent interface points. $\boldsymbol{R}_{ij}$ stands for the vector from point *j* to point *i*.

In the fast polarization approximation which is a part of the Fuzzy-Border (FB) model, the self-consistency iterations are truncated. Equations 10 and 12 together form a self-consistent problem which can be solved iteratively. The first step consists of replacing $\boldsymbol{E}_{1,i}$ with $\boldsymbol{E}_{1,i}^{\boldsymbol{0}}$, the field created by the solute only and not by the polarized solute-solvent interface. If we stop at this stage, we obtain our first-order approximation. It still contains many-body interactions, since the electrostatic field is a vector quantity and it contains contributions from all the solute charges. But the problem is now analytical, not self-consistent, and the convergence is no longer an issue. If we now recompute the electrostatic field $\boldsymbol{E_{2,i}}$ taking into the account the field created by the interface charges using Equation 12, obtain the new charges with the Equation 10, but do not perform the next iteration, we obtain the second-order model, which is still safe from the electrostatic charges convergence problems since there are only two iterations and magnitudes of the charges cannot increase beyond the value they achieve at the first of the second iteration.

Choosing the Numerical Grid to Represent the Solute-Solvent Interface

We use a fixed cubic three-dimensional equally-spaced grid to minimize the noise resulting from grid rebuilding after moving a solute atom or a group of atoms. The interface between solute and solvent is assumed to consist of points with distances from $R-\Delta$ to $R+\Delta$ from the solute atom, as shown on Figure 2.

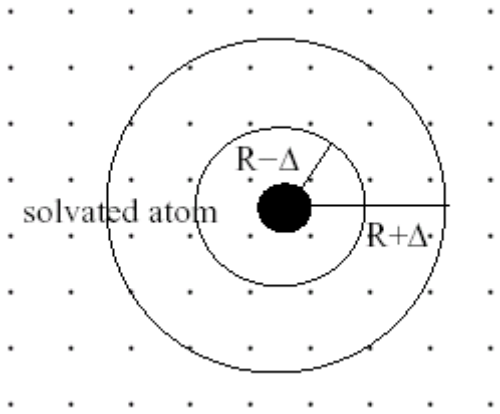

Figure 2. Schematic depiction of a solvated atom and the solvent grid.

A point is defined as solvent-accessible if the following two conditions were satisfied: (i) there is no solvent atom *i* which would be closer to the grid point than $R_i - \Delta_i$ and (ii) there is at least one point within distance no more than R_{slv} from the grid point in question, for which no solute atom *i* would be closer that $R_i - \Delta + R_{slv\,i}$, where R_{slv} stands for the effective radius of a solvent molecule. The following equation gives the weight associated with each point:

$$w_j = a_{0,i}\left[\left(\frac{R_{ij} - R_i}{\Delta_i}\right)^4 - 2\left(\frac{R_{ij} - R_i}{\Delta_i}\right)^2 + 1\right] \tag{25}$$

Here R_{ij} is the distance between the solvent grid point and the corresponding solute atom, $a_{0,i}$ is a parameter which depends on the solute atomtype, and the whole weight is maximum at the nominal solvation radius R_i and decreases to zero at distances $R_i - \Delta_i$ and $R_i + \Delta_i$.

We can now write down the overall FB continuum solvation formalism. Once the solvation surface grid points *j* are defined as described above, the zeroth-order electrostatic field at those points is found as:

$$\boldsymbol{E}_{2,j}{}^{0} = \sum_i \frac{q_i \boldsymbol{R}_{ij}}{R_{ij}^3} \tag{26}$$

The summation goes over all the solute points. The first-order FB charge on the grid point *j* is then:

$$q_j{}^{I} = -A_{scale}\frac{1}{4}\pi(\varepsilon - 1)w_j \boldsymbol{E}_{1,j}{}^{0} \cdot \boldsymbol{n}_j \tag{27}$$

A_{scale} is a scaling factor and an adjustable parameter of the theory. The first-order electrostatic part of the solvation energy can then be calculated as:

$$\Delta G(el)^I = \frac{1}{2}\sum_j q_j{}^{I}\phi_j^0 \tag{28}$$

If the second-order approximation is to be produced, the first-order electrostatic field is found as described in Equation 29:

$$\boldsymbol{E}_{1,j}{}^{I} = \boldsymbol{E}_{1,j}{}^{0} + \sum_{k \neq j} \frac{q_k \boldsymbol{R}_{kj}}{R_{kj}^3} \tag{19}$$

with the additional summation done over the solute-solvent interface points *k*. Equation 27 in then modified to include the first-order and not the zeroth-order field:

$$q_j^{II} = -A_{scale}\frac{1}{4}\pi(\varepsilon - 1)w_j \boldsymbol{E}_{1,j}^{I} \cdot \boldsymbol{n}_j, \tag{30}$$

and the resulting second-order energy is:

$$\Delta G(el)^{II} = \frac{1}{2}\sum_j q_j^{II}\phi_j^0 \tag{31}$$

Finally, the electrostatic part of the energy, regardless of whether it is calculated with the first- or second-order model, is multiplied by 332.0657418 in order to obtain the final result in kcal/mol.

The Non-Polar Part of the Solvation Energy

The non-polar part of the solvation energy was calculated as a sum of two terms, one with a positive and one with a negative contribution:

$$\Delta G(np) = \sum_j w_j A_{np} - \sum_i \sum_j w_j \frac{A_i^{LJ}}{R_{ij}^6}, \tag{32}$$

The first term contains a sum taken over all the grid points. This is essentially the overall solvent-accessible surface area (SASA) contribution which is commonly employed in continuum solvation models. The second term is calculated with a double summation going over all the grid points *j* and all the solute atoms *i*. It approximates the attraction part of the Lennard-Jones energy for interactions between the solute and solvent atoms.

Once the electrostatic and non-polar terms of the solvation energy are found, the overall solvation energy can be calculated according to Equation 16.

Monte Carlo Simulations

Monte Carlo simulations are carried out with the Metropolis sampling. The ensemble is NPT, while NVT run can be performed by setting the frequency of volume moves to a large value. At each step, an attempt is made to either randomly change the volume or to randomly move one molecule. The following value is calculated:

$$C = [E(new) - E(old)]/RT \tag{33}$$

for the molecule moves, where E(old) is the energy of the last accepted configuration (or the initial energy at the beginning of the simulation), E(new) is the energy of the new configuration, R is the universal gas constant, and T is the temperature in K. For the attempted volume moves,

$$C = [E(new) - E(old)]/RT - NMOL\ \log(V(new)/V(old) \tag{34}$$

where NMOL is the total number of molecules, and V(new) and V(old) are the new and the old values of the volume of the system, respectively. Then exp(-C) is compared to a random number between 0 and 1. If exp(-C) is greater or equal than this number, the move is accepted. Otherwise it is rejected.

The Metropolis algorithm allows to calculate properties as a simple average over the accepted Monte Carlo steps.

Geometry Optimizations

Geometry optimizations are carried out in internal coordinates with the L-BFGS-B optimizer references in the Citations section. There is also a version with a direct-search optimizer that can be used in final geometry optimization steps carried out with the Fuzzy-Border solvent.

ΔG Calculations with Monte Carlo and Statistical Perturbation Theory

Free energy differences can be computed with the statistical perturbation theory. When a system is changed from form *i* to form *j*, the ΔG can be calculated as:

$$\Delta G = G_j - G_i = -kT \ln \langle \exp(-(E_j - E_i)/kT) \rangle_i \qquad (35)$$

Here the brackets stand for averaging over the initial system configuration space (with Monte Carlo sampling). For example, ethane can be mutated to methanol by making two hydrogen atoms disappear and adjusting the atomtypes of the remaining atoms accordingly. Normally, the change between two such systems is too large. Therefore, the whole process is usually broken into a number of steps. A parameter λ is used, with λ = 0 being the initial system and λ = 1 – the final one. At each step, the reference system *i* is used for the sampling, while the differences are calculated for two different j_1 and j_2. For example, λ = 0.050, 0.000, and 0.100 for *i*, j_1 and j_2, respectively. This procedure is known as double-wide sampling [6]. More information about this type of simulations can be found in references 6 and 7.

Implementation

POSSIM is implemented in a form of several modules designed for separate tasks. Linux executables are included with the distribution. Here is the list of the separate programs included in the POSSIM suite and possible UNIX commands for compiling:

<u>Geometry optimizations, no Fuzzy-Border (FB) continuum solvent:</u>
Directory: OPT_ZMAT
Name of the executable: possim_opt
Can be compiled by the following command:

f77 -O3 blas.f enscr.f lbfgsb.f linpack.f optimize.f possim_opt.f read_cutoff.f set_write.f solvpol.f timer.f tors.f valgrad.f zmatrix.f -o possim_opt -ffast-math

Geometry optimizations with FB (for non-polarizable solutes):
Directory: FB
Name of the executable: possim_opt_fb
Can be compiled by the following command

f77 -O3 blas.f enscr.f lbfgsb.f linpack.f optimize.f possim_opt_fb.f read_cutoff.f set_write.f solvpol.f timer.f tors.f valgrad.f zmatrix.f -o possim_opt_fb -ffast-math

Geometry optimizations with FB (for non-polarizable solutes), simple search optimizer (for final steps in the optimizations, if needed):
Directory: FB_GEO
Name of the executable: possim_opt_fb_ds
Can be compiled by the following command

f77 -O3 enscr.f possim_opt_fb_ds.f read_cutoff.f set_write.f solvpol.f tors.f valgrad.f zmatrix.f -o possim_opt_fb_ds -ffast-math

Monte Carlo simulations:
Directory: MC_ZMAT
Name of the executable: possim_mc
Can be compiled by the following command:

f77 -O3 delem.f delem_back.f enscr.f formc.f main.f ran.f read_cutoff.f read_first.f set_read.f set_write.f solvpol.f solvpoln.f stat.f tors.f zmatrix.f zrot.f -o possim_mc -ffast-math

Monte Carlo simulations with free energy perturbations (ΔG calculations):
Directory: MC_DG_ZMAT
Name of the executable: possim_dg_mc
Can be compiled by the following command:

f77 -O3 delem.f delem_back.f enscr.f formc.f main.f ran.f read_cutoff.f read_first.f readdg.f set_read.f set_write.f solvpol.f solvpoln.f stat.f tors.f zmatrix.f zrot.f -o possim_dg_mc -ffast-math

Input and Output Files

Geometry Optimizations: Input

Examples of input files are given below. Here and throughout this manual, text from the actual files is presented in bold, explanations are given in regular font. .

File input.inp
This file contains some energy-related input.

SOME ENERGY-RELATED INPUT
polarization (YES/NO)? polarization order (0-3)? convergence criterion?
Icalpol iNpol TolPol
YES 2 0.001

YES – include polarizable calculations. NO – do not calculate polarization energy, even if some atoms are listed as polarizable in other input files. iNpol – polarization order (normal for the POSSIM force field is 2, 0 – no polarization, 1 – only first order, Equation 13a. If this variable is set to 3, full polarizable calculations are carried out until the level of conversion indicated by the TolPol variable is achieved. POSSIM force field is NOT parameterized for the value of 3). TolPol – criterion for convergence in case of inpol = 3, units are e·Å, average change of a dipole component is compared with this parameter.

1,4-scaling factor, cutoffs for dipole-dipole and other interactions
f14 RmaxD RmaxC
0.5 7.0 7.0

The 1,4-scaling factor is 0.5 for both the POSSIM and OPLS-AA force fields.

distance to start checking for unphysical proximity (Rcheck)
5.5

Below this distance, it is checked if Equation 12 should be invoked to scale the interatomic distance.

geometry optimization convergence criteria, total gradient and projection
Grad_Tol1 Grad_Tol2
10000000.0 0.00001

These values are typical for the L-BFGS-B optimizations without the FB continuum solvent.

box size
XL YL ZL
1000.0 1000.0 1000.0
include LJ correction and smoothing of the electrostatics at RmaxC (YES/NO)?
iljcor ifeath
NO NO

Indicating whether Equation 14 should be used for the smoothing of the electrostatics and if the correction for neglected Lennard-Jones energy beyond the cutoff distance should be included.

include cluster potential (YES/NO)? cluster radius and force constant?
icluster Rcluster clapot
YES 12.0 0.5
if YES, molecule and atom at the center of the cluster
ncluster1 ncluster2

1 1

A returning harmonic cluster potential can be applied to molecules that are farther away from the atom ncluster2 in molecule ncluster1 than the distance Rcluster. The harmonic constant is equal to clapot.

File solu.inp
This file contains solute information.

this file contains solute coordinates and some other data
Z-matrix input file is also required
number of solutes:
1
for each solute (this line must be present for each solute):
atomic types and coordinates:
C01 157 1.088213 1.367530 -0.199419
C02 135 0.297596 0.429501 0.715005
……

For each solute atom, a name (up to four character), the atom type and Cartesian coordinates should be given. The list of atoms is terminated by a blank line.

bond stretches: atoms
1 2
1 3
……

angle bends: atoms
2 1 3
4 3 1
……

torsions: atoms and types:
4 3 1 2 1
4 3 1 5 2
……

polarizable atoms:
1
2
……

for each solute:
…….

Each list (bond stretches, etc.) is terminated by a blank line. All the stretches, etc. should be listed explicitly, but only the torsions require to have a type given after the atom numbers, as type for the bond stretches and angle bends are found automatically.

File slv.inp

This file contains information about solvent molecules (both solutes and explicit solvent can be included in geometry optimizations and any other POSSIM calculations). The format of this file in optimizations is the same as for solu.inp.

File zmat.inp

Z-matrix for the system. This file must be present for all simulation.

Z-MATRIX FILE
COORDINATES FROM ZMAT (YES/NO)?
YES

If NO, then the coordinates are taken from solu.inp and slv.inp, and zmat.inp is used only for designating the degrees of freedom to be used in the calculations. If YES, then the actual values of the coordinates in solu.inp and slv.inp are disregarded (and can be set to any values, such as 0.0 0.0 0.0).

INTERNAL COORDINATES IN Z-MATRIX FORMAT:
C1
C2 C1 1.53008003
O3 C1 1.41287797 C2 109.220069
H4 O3 0.948009613 C1 108.933803 C2 70.8296896
H5 C1 1.09117167 C2 110.95293 O3 121.205182
H6 C1 1.09144909 C2 110.301692 O3 239.858286
…..

The Z-matrix format is essentially the same as used by the Jaguar software [8], but all the values of the variables should be given right in the lines, with no additional section with the list of values allowed.

The names of the atoms given here may differ from the names in the solu.inp and slv.inp files, but the referencing atom labels should match. For example, in the above file, the atom named H6 has a distance of 1.0914909Å from atom C1, the angle H6–C1–C2 has a value of 110.301692º, and the dihedral H6–C1–C2–O3 is at 239.858286º.

If there is a # after a parameter value, this degree of freedom is kept fixed during the simulation. For example,

H4 O3 0.948009613# C1 108.933803 C2 70.8296896

in the above file would mean that the H4–O3 distance is kept constant at 0.948009613Å.

There should be no break between the solute and solvent molecules in zmat.inp file for optimizations, the complete list is terminated by a blank line.

File param.inp

List of atomtype parameters for non-bonded interactions.

parameter file. atomtypes. number of atomtypes:

type #	symbolic type	charge	sigma	epsilon	alpha	rcutp	rcutq	
135	CT	-0.18	3.500	0.066	0.5069	0.80	0.80	aliphatic C
140	HC	0.06	2.500	0.030	9999.99	0.80	0.80	aliphatic H

The list is terminated by a blank line. The two-character symbolic atomtypes are used in determining the bond-stretching and angle-bending types. Type numbers are designed to correspond to the OPLS-AA numbering, but they do not have to. They do not have to be in order. Charges are in electrons, sigma and epsilon as used in Equation 1, units are Å and kcal/mol, respectively. Please note that the optimized distance between two isolated particles is not σ but $\sigma \cdot 2^{1/6}$. Alpha is inverse atomic polarizability, $Å^{-3}$. Rcutp and rcutc in the current implementation have to be the same and correspond to the R^{cut} parameters in Equation 11. Text beyond the rcutq values is treated as comments.

File strbnd

Bond stretching and angle bending parameters.

stretch-bend parameters file

bond stretches:

#	symb. type	bond length	strength constant
1	CT-CT	1.529	268.0
2	CT-OH	1.41	320.0

angle bends:

#	symb. type	angle	strength constant
1	CT-CT-OH	109.5	50.0
2	CT-OH-HO	108.5	55.0

The lists are terminated with blank lines. The file is self-explanatory, the values of the parameters are currently the same as in the OPLS-AA.

File tors.par

Torsional parameters.

torsional parameters

#	V1	V2	V3	V4	
1	-0.356	-0.174	0.492	0.0	CT-CT-OH-HO
2	0.0	0.0	0.350	0.0	HC-CT-OH-HO

The list is terminated by a blank line. Only the numbers for each type are needed, the symbolic types listed are comments.

File cutoffs.inp
Indicates what groups of atoms interact which each other and how are the cutoffs set.

This file contains information on cutoff groups and atoms.
Include all intermolecular interactions (YES/NO)?
NO

If YES, all intermolecular interactions are included.

If not, give the total number of cutoff groups:
2
for each group (this line must be present for each group):
REPEAT 2 TIMES

Each group used for cutoff determination can be repeated several times. For example, in this file, the center of the group is atom number 1, atoms 1 – 9 are included into the group, and this order is valid for two molecules. (REPEAT 2 TIMES). The REPEAT statement can include several molecules spanning both solutes and solvents. Solute molecules always precede the solvents. It is not necessary to use the REPEAT keyword, one can simply enter the cutoff information for each molecule/group explicitly.

central atom (molecule and atom numbers):
1 1
all the atoms (assumed from the same molecule as the central one)
1
2
3
4
5
6
7
8
9

In each simulation, the system is divided into cutoff groups (whole molecules for small molecular systems or molecular fragments such as protein residues for larger ones). Two groups are considered interacting with each other of the distance between the designated "central" atoms of the groups is below the cutoff.

Additional Input for Fuzzy-Border Calculations

File fuzzy.inp
This file contains input parameters for the FB model.

PARANETERS FOR FUZZY-BORDER CONSTINUUM SOLVATION
ARE FUZZY-BORDER CALCULATIONS DESIRED (YES/NO)?
YES

IF YES, THEN:
GRID SPACING SOLVENT RADIUS SOLVENT EPSILON
fuzspace fuzrad fuzeps
0.40 1.4 80.4

Distance between the grid points, radius of solvent molecules, solvent dielectric constant.

FB ORDER
1

This is the normal value for which FB has been parameterized.

REMOVE INACCESSIBLE INTERNAL CAVITIES (YES/NO)?
NO

If YES, surfaces of the inaccessible internal cavities will not be included as solute-solvent interface. This option should not be turned on if such cavities are not present.

SCALING FACTOR FOR SURFACE-SURFACE CHARGE INTERACTIONS
fuzselfscale
1.0

This is the normal FB value.

NUMBER OF DIVISIONS PER DIMENSION FOR SMOOTHING (NORMALLY 1)
nfuzd
1

This is the normal FB value.

SHIFT OF THE COORDINATE SYSTEM
fuzdx fuzdy fuzdz
0.0 0.0 0.0

This is the normal FB value.

IF FB ORDER ABOVE 1ST REQUIRED, NUMBER OF ADDITIONAL ITERATIONS (NORMALLY 1)
ifuziter
1

This is the normal FB value.

ADDITIONAL FB CHARGE SCALING FACTOR
fuzqscale
0.07069

This is the normal FB value, do not change it.

FUZZY-BORDER PARAMETERS FOR SPECIFIC ATOMTYPES:

TYPE	R	DELTA	a0	ALJ	ANP
135	1.964	0.25	0.040	241.0	3.200
136	1.964	0.25	0.040	241.0	3.200
140	1.403	0.25	0.040	0.000	3.200

This list is terminated with a blank line. The parameters are the ones used in the FB model as described above. Values of all ANP should be the same. This is also so for the values of DELTA.

File input.inp for the direct search FB version (possim_fb_ds).

There are additional lines at the bottom of this file:

FOR GEOEMTRY OPTIMIZATION DIRECT SEARCH
RANGES OF ATTEMPTED MOVES FOR DISTANCES (A) AND ANGLES (RAD)

xgeod1	xgeod2
0.01	0.01

These two parameters set steps for the direct search. The values given above are reasonable and can be used in actual calculations.

Values of parameters in FORTRAN files:
If the program has to be recompiled with different values, please keep in mind that the same parameter can be defined in more than one FORTRAN file (and more than one routine in each file), and all these values for the same parameter have to be kept the same.

parameter (maxatm=3000)	maximum number if atoms
parameter (maxsol=10)	maximum number of solutes
parameter (maxslv=2000)	maximum number of solvents
parameter (maxmol=2010)	maximum number of molecules (maxsol + maxslv)
parameter (maxbnd=6000)	maximum number of bond stretches
parameter (maxang=6000)	maximum number of angle bends
parameter (maxtor=10000)	maximum number of torsions
parameter (maxpol=1400)	maximum number of polarizable atoms
parameter (maxpar=6000)	maximum number of atomtypes
parameter (maxcut=1000)	maximum number of cutoff groups
parameter (maxatcut=52)	maximum number of atoms in a cutoff group

Additional parameters for Fuzzy-Border:

parameter (maxfp=30000)	maximum number of solvation grid points
parameter (maxfuzside=100)	maximum number of grid points along one dimension
parameter (maxfuzperatm=5000)	maximum number of grid points around one atom
parameter (amaxover1=9.0d+0)	should not be changed
parameter (amaxover2=9.0d+0)	should not be changed

Please note that actual values of specific parameters may differ in different implementations.

Geometry Optimizations: Output

Several files are produced as a result of executing geometry optimizations.

Files solu.out, slv.out and zmat.out contain the final versions of the files with the solute, solvent and Z-matrix data Please note that both the Z-matrix and the coordinate files contain the correct final coordinates, regardless of which ones were used for the input.

File plt lists the final Cartesian coordinates for all atoms. This file can be read by other software (such as XChemEdit [9] on the Figure 3 below) to visualize results of the simulations.

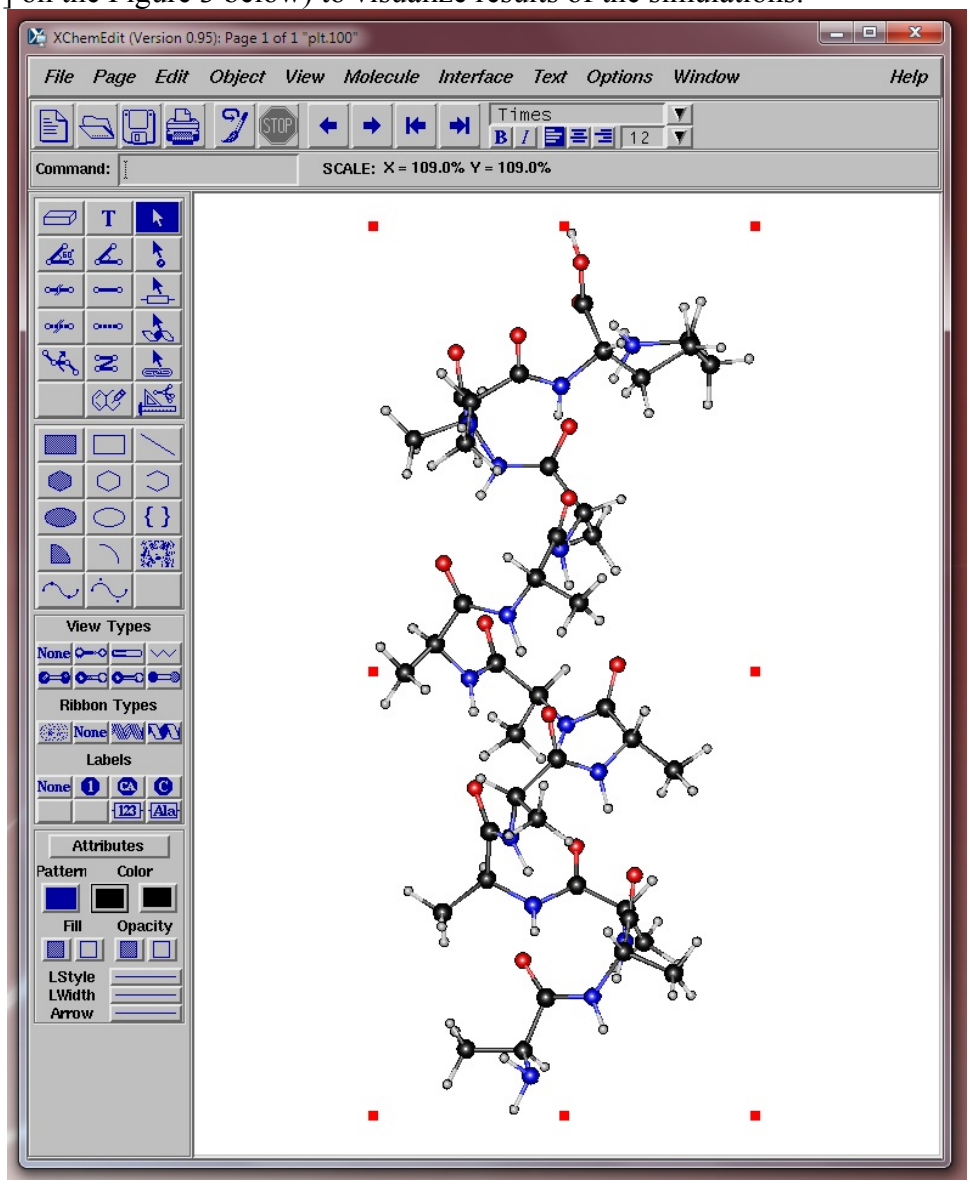

Figure 3. A small alpha-helix system resulting from a POSSIM run displayed by XChemEdit, plt file read.

General output data is sent to the standard output. A more detailed version is saved in file output.out. The content of this file is generally self-explanatory. The following notations are used for components of energy:

ESTR – bond stretching energy
ENBND – angle bending energy
ENTOR – torsional energy
ENNA14 – 1,4- non-bonded interactions
ENNA – all the other intramolecular non-bonded interactions
ENXX – solute-solute non-bondedn intermolecular interactions
ENSX – solute-solvent non-bondedn intermolecular interactions
ENSS – solvent-solvent non-bondedn intermolecular interactions
ENINTER = ENSS + ENXX + ENSX
ENLJCOR – Lennar-Jones correction energy
ECLU – the cluster retention energy
EPOL – polarization energy
ENERGY or ENEW – total energy.

Additionally, outputs of simulations with the Fuzzy-Border continuum solvent contain the following energy components:

ENFUZ – continuum solvation energy
ENFUZEL – the electrostatic component of the continuum solvation energy
ENFUZNP – the nonpolar component of the continuum solvation energy (without the Lennard-Jones part)
ENFUZLJ – the Lennard-Jones component of the continuum solvation energy

Monte Carlo: Input

Much of the input used in Monte Carlo simulations has the same format as the geometry optimization input files. There are however some differences, as described below.

File cutoffs.inp

The only difference from the optimization case is that the central atom could now be given in 1 0 format. In this case (atom number set to zero), every single atom in the group is treated as a potential central atom, and the cutoff is based on the shortest distance between atoms of the group and the central atom (or atoms) of another group.

File mc.inp
This file contains various data needed for Monte Carlo simulations

THIS FILE CONTAINS DATA FOR MONTE CARLO SIMULATIONS
T (C) P (ATM)
25.0 1.00
NUMBER OF MC STEPS AND FREQUENCY OF VOLUME AND SOLUTE MOVES
nstep nfreqvol nfreqsol
001000 600 20

nstep is the total number of Monte Carlo configurations. Volume moves are attempted every nfreqvol configurations and solute moves – every nfreqsol configurations. Volume moves take precedence.

RANGES OF ATTEMPTED MOVES FOR SOLUTES, SOLVENTS AND VOLUME
delu delau dels delas vdel
0.10 10.0 0.10 10.0 150.0

Ranges of attempted moves. delu and dels are ranges for translational motion in Å, delau and delas are ranges of rotational motion in degrees. Units for vdel are $Å^3$. This ranges should be adjusted in the course of the simulations to yield acceptance ratios of 40-50%.

MAXIMUM NUMBER OF INTERNAL DEGREES OF FREEDOM CHANGED AT EACH STEP
maxvar
15

The maximum number of internal coordinates that can be changes in one attempted move. 15 is the normal value and, generally speaking, should not be changed.

RANGE OF ADDITIONAL MOTION IN CARTESIAN COORDINATES
amovvar
0.00

In Monte Carlo simulations, additional motion in Cartesian coordinates can be attempted for atoms designated in solu.inp and slv.inp files. The can be useful, for example, when a ring or a long polymer (such as a protein) is present. The variable amovvar indicates this additional movement range in Å.

FREQUENCY OF COORDINATE OUTPUT
nfreqout
999999

This option can be used for debugging, but normally the value should be set to a large number to avoid creating large unneeded output files.

THE FOLLOWING LINES CONTAIN DATA FOR MOLECULES, ONE LINE PER MOLECULE
MOLECULE #, INTERNAL FLEXIBILITY (YES/NO), CENTRAL ATOM FOR ROTATION
REPEAT 2 1 YES 1

This last section indicates whether each of the molecules has internal flexibility (YES/NO) and what is the central atom in the molecule for molecular rotation. The following are legitimate examples of this section:

REPEAT 216 1 YES 1

216 identical molecules, pure liquid

1 YES 2
REPEAT 215 2 NO 1

One flexible solute molecule, central atom is the atom #2, plus 215 rigid solvent molecules, central atom in each is #2, the first molecule in the series is molecule #2.

File input.inp
This file for Monte Carlo simulations contains a section related to WKC:

SOME ENERGY-RELATED INPUT
polarization (YES/NO)? polarization order (0-3)? convergence criterion?
Icalpol iNpol TolPol
YES 2 0.0001
1,4-scaling factor, cutoffs for dipole-dipole and other interactions
f14 RmaxD RmaxCXX RmaxCSS RmaxXXS RmaxCI
0.5 127.0 127.0 120.0 120.0 120.0
distance to start checking for unphysical proximity (Rcheck)
5.5
box size
XL YL ZL
100.0 100.0 100.0
use WKC (YES/NO)? If YES, WKC parameter
YES 100.0
include LJ correction and smoothing of the electrostatics at RmaxC (YES/NO)?
iljcor ifeath
YES NO
include cluster potential (YES/NO)? cluster radius and force constant?
icluster Rcluster clapot
YES 12.0 0.5
if YES, molecule and atom at the center of the cluster
ncluster1 ncluster2
1 1

WKC technique is used to make solvent molecules located close to the solute move faster than those located far away (preferential sampling). This way solvation effects are sampled better and convergence is faster. The WKC parameter is used to set the frequency of attempted moves to the solvent molecules at $1/(R^2 + WKC)$, where R is the distance from the origin (the center of the cell). WKC should increase

as the number of solvent molecules increasing to avoid gradual volume expansion. Examples of appropriate WKC parameters – 150.0 for 216 water molecules, 250.0 for 267 chloroform molecules.

IMPORTANT! RmaxD cannot be greater than any of the RmaxC values.

ifeath – flag for using the quadratic smoothing of electrostatic interactions over the last 0.5 Å before the R_{max} distance. **THIS OPTION WORKS ONLY FOR MONTE CARLO, NOT FOR OPTIMIZATIONS.**

File solu.inp

Please note that the first three atoms of each molecule are used for the molecular position and orientation, even if the coordinates are otherwise taken from zmat.inp. Therefore, the coordinates of the first three atoms have to be meaningful (for example, they cannot be all equal to 0.0).

When Monte Carlo modules are used, this file contains an additional section for each molecule:

atoms with additionally variable coordinates:
1
2
3

This section is located between the list of torsions and the list of polarizable atoms. The list of these additionally movable atoms is terminated by a blank line and can be empty. Atoms listed in this section have additional attempted movements with the maximum distance indicated by the value of variable amovvar in file mc.inp.

File slv.inp

This file has two new features when Monte Carlo simulations are run (compared to slv.inp for geometry optimizations).

First, the section for listing **atoms with additionally variable coordinates** is added for each molecule, just like in the case of file solu.inp.

Second, a REPEAT command is permitted here (but not in solu.inp!). For example:

…
for each solvent (this line must be present for each solvent):
REPEAT 216 DISTANCES 5.0 5.0 5.0
atomic types and coordinates:
C01 157 -0.428555 -0.200014 -3.338255
C02 135 -1.899447 0.021327 -2.979007
…

In the above case, the molecule will be replicated into 216 identical ones. These replicas will be arranged in a cube with distances of 5.0Å between the positions of the nearest neighbors, in each direction (along the X, Y and Z axes). However, the distances can be set to different values in each dimension. If the total number of molecules is not a cube of a natural number, some positions in the cube will be empty. All the molecules generated this way have the same orientation (no additional rotation is applied). This option is convenient in the initial setup of liquid-state simulations, but should be avoided in continued runs as such a configuration will almost invariable have a high energy and will require a number of Monte Carlo configurations for equilibration to be achieved.

File stat.inp

This file contains input data for building radial distribution functions and internal coordinate averages.

THIS FILE CONTAINS INPUT FOR COORDINATE AVERAGING AND RDFS
INTERNAL COORDINATES FOR AVERAGING:
FOR EACH COORDINATE (THIS LINE MUST BE PRESENT FOR EACH COORDINATE):
MOLECULES, FROM TO, ATOM, COORDINATE TYPE (1-DISTANCE, 2-BOND ANGLE, 3-DIHEDRAL)
1 6 4 3

In this example, distribution of the dihedral angle (as defined in zmat.inp) for atom number 4 in molecules 1 – 6 is recorded and averaged.

RANGE, FROM TO (THERE ARE ALWAYS 360 POINTS)
0.0 360.0

The distribution is collected in the range from 0.0º to 360.0º.

More coordinates for averaging can be added here, each starting with FOR EACH COORDINATE... The list is terminated with a blank line.

RADIAL DISTRIBUTION FUNCTIONS:
Rmin and deltaR for RDF'S (THERE ARE ALWAYS 401 POINTS)
1.2 0.022
FOR EACH RDF (THIS LINE MUST BE PRESENT FOR EACH RDF):
1ST ATOM: MOLECULES, FROM TO, ATOM
1 6 2
2ND ATOM: MOLECULES, FROM TO, ATOM
1 6 2

This radial distribution function is collected for molecules 1 through 6, distances between atoms number 2 and number 2 (or another molecule) are considered.

INCLUDE ATOMS IN THE SAME MOLECULE (YES/NO)?
NO

Atoms in the same molecule are not included in the example (otherwise there would be a large peak at zero distance). However, in other cases, it could be desirable to find distributions of distances between atoms in the same molecule.

More RDF's for averaging can be added here, each starting with FOR EACH RDF... The list is terminated with a blank line.

File zmat.inp

The format for this file to be used in Monte Carlo simulations is somewhat different from that for geometry optimizations. The main difference is that each molecule is listed separately (or in a separate group, with the REPEAT command).

Even if the coordinates are taken from zmat.inp, the positions and orientations for each molecule are obtained from the solu.in and slv.inp files, since Z-matrices provide internal coordinates only. Just like in geometry optimizations, degrees of freedom listed in Z-matrices can be kept fixed during Monte Carlo simulations with a # placed after the value of the specific degree of freedom that should stay constant.

Inputs for molecules (or groups of molecules) in zmat.inp are separated by blank lines.

Z-MATRIX FILE
NUMBER OF MOLECULES:
216
FOREACH MOLECULE (THIS LINE MUST BE PRESENT FOR EACH MOLECULE):
COORDINATES FROM ZMAT (YES/NO)? XYZ(1)& ORIENTATION ARE ALWAYS FROM solu/slv.inp
NO
RANGES FOR BOND LENGTHS AND ANGLES ADJUSTED AUTOMATICALLY. ADDITIONAL FACTOR:
0.85

Ranges for attempted bond lengths and bond angles changes are calculated automatically based on the force constants and temperature. The additional factor can scale these attempted ranges. We found that the value of 0.85 helps to make the acceptance ratio closer to 40%, but other values (between ca. 0.5 – 1.0) can be used as well. The ranges for the attempted moves in the dihedral angles (in degrees) are given after each atom. In this specific case, they all are equal to 5.0º.

Z-MATRIX AND RANGES OF ATTEMPTED MOVES IN DIHEDRALS
C01
O02 C01 0.0
H03 O02 0.0 C01 0.0
H04 C01 0.0 O02 0.0 H03 0.0 5.0
H05 C01 0.0 O02 0.0 H04 0.0 5.0
H06 C01 0.0 O02 0.0 H04 0.0 5.0

FOREACH MOLECULE (THIS LINE MUST BE PRESENT FOR EACH MOLECULE):
REPEAT 215

COORDINATES FROM ZMAT (YES/NO)? XYZ(1)& ORIENTATION ARE ALWAYS FROM solu/slv.inp
NO
RANGES FOR BOND LENGTHS AND ANGLES ADJUSTED AUTOMATICALLY. ADDITIONAL FACTOR:
0.85
Z-MATRIX AND RANGES OF ATTEMPTED MOVES IN DIHEDRALS
C01
O02 C01 0.0
H03 O02 0.0 C01 0.0
H04 C01 0.0 O02 0.0 H03 0.0 5.0
H05 C01 0.0 O02 0.0 H04 0.0 5.0
H06 C01 0.0 O02 0.0 H04 0.0 5.0

IMPORTANT! The amovvar variable denotes the change in atomic Cartesian coordinates IN ADDITION to the motion defined in the Z-matrix. Therefore, amovvar should be set to zero in most cases.

Additionally, for calculations of changes in Gibbs free energy (possim_dg_mc):

File dg.inp
This file contains information needed for free energy perturbations.

information for deltaG calculations
perform deltaG calculations (YES/NO)?
YES
atoms with changing atomtypes (max=100)
molecule, atom, initial type, final type
1 1 135 157
1 2 935 100
1 6 140 100
1 7 140 100
1 8 140 100
1 9 100 154
1 10 100 955

reaction coordinate values (initial and two perturbed):
RC0 RC1 RC2
0.350 0.300 0.400

This file is needed to obtain free energy differences. The changes from the initial to the final system are made by changing atomtypes. For example, disappearing of an atom is done by switching its type to a dummy atom one. In the example above, seven atoms are changing (the list is terminated by a blank line). $\lambda = 0$ corresponds to atom 1 in molecule 1 being type 137 and atom 2 in molecule 1 being type 935, etc. $\lambda = 1$ corresponds to the first one having the atomtype 157 and the second atom becoming type 100 (the usual designation for a dummy atom type, though it still has to be defined explicitly in the

param.inp file). RC0 sets the reference system to $\lambda = 0.350$, the perturbed systems to $\lambda = 0.300$ and $\lambda = 0.400$. The output values of the ΔG will correspond to the $0.350 \rightarrow 0.300$ and $0.350 \rightarrow 0.400$ transitions.

Monte Carlo: Output

Much of the Monte Carlo output is similar to that for geometry optimizations. There are some obvious changes in the output.out file, but the energy components are still named the same way. Some changes in other output files are outlined below.

File input.out
This file is essentially the same as input.inp, except that the final box sizes XL, YL and ZL are listed. This is needed for restarting/continuing Monte Carlo runs (as described below).

File stat.out
This file contains resulting distributions for internal coordinates and radial distribution functions requested in stat.inp. The coordinate distribution is given in fractions (adding up to 1.0) and the RDF's are given in the conventional form, g vs. R.

Additionally, for ΔG calculations (possim_dg_mc):

File dg.out.

This file contains output related to the ΔG calculations.

RC0, RC1, RC2:
0.35 0.3 0.4
BETA AND XSTEP:
1.68783568 20000.

$1/(k_B \cdot T)$ and the number of Monte Carlo configurations.

RAW DATA FOR DELTAG1 AND DELTAG2 (BEFORE TAKING LN):
0.447387664 3.69943468

According to Equation 35, $\Delta G = G_j - G_i = -kT \ln < \exp(-(E_j - E_i)/kT) >_i$. Given here are $< \exp(-(E_j - E_i)/kT) >_i$ for RC0→RC1 and RC0→RC2.

FINAL DELTAG1 AND DELTAG2 ARE (AFTER TAKING LN):
0.476545087 -0.77506361

ΔG values for RC0→RC1 and RC0→RC2 calculated with Equation 35.

IMPORTANT! Currently, neither the quadratic feathering nor the zero atom number in cutoffs is implemented in the geometry optimization routines. Therefore, optimizations should be ran with very large RmaxC to assure that the whole molecule is included and there are effectively no cutoffs employed. However, the RmaxD values can (and should) still be used.

Monte Carlo: Running a Sequence of Jobs

Monte Carlo simulations normally carried out in a series of steps. For example, simulations of pure liquid methanol can be executed with 216 molecules in periodic boundary conditions with 1 x 10^6 configurations of equilibration and 4 x 10^6 configurations of averaging. These simulations would be normally carried out as a series of smaller steps, such as 2 x 10^5 configurations each. During the equilibration stage, ranges of solute and solvent intra- and inter-molecular movements and the range from the volume moves are adjusted to achieve a ca. 40%. Acceptance ratio target.

In order to run five smaller steps in a sequence, the following script can be used:

```
./possim_mc  >!  oa

cp  output.out  outa
cp  input.out  ina
cp  input.out  input.inp
cp  zmat.out  zmata
cp  zmat.out  zmat.inp
cp  solu.out  solua
cp  solu.out  solu.inp
cp  slv.out  slva
cp  slv.out  slv.inp
cp  plt plta
cp  stat.out  stata

cp dg.out dga (if ΔG calculations are carried out, thus the executable would be possim_dg_mc)

foreach  i  (b  c  d  e)
./possim_mc  >!  o$i

cp  output.out  out$i
cp  input.out  in$i
cp  input.out  input.inp
cp  zmat.out  zmat$i
cp  zmat.out  zmat.inp
cp  solu.out  solu$i
cp  solu.out  solu.inp
cp  slv.out  slv$i
cp  slv.out  slv.inp
```

cp plt plt$i
cp stat.out stat$i

cp dg.out dg$i (if ΔG calculations are carried out, thus the executable would be possim_dg_mc)

end
exit

The script can be executed by the following command:

csh <filename for the script>

By copying the files, two goals are achieved. First, the resulting outputs, Z-matrices, final coordinates and statistics are saved and can later be analyzed. Second, and very importantly, each step (except for the first one) is started from the coordinates and volume obtained at the end of the previous one, and thus the sequence of the runs is equivalent to just one long run. Please note that the initial energy of step i should be exactly the same as the final energy of step i-1. The files to be copied from .out to .inp are: solu.out, slv.out, input.out, zmat.out (the last one is needed if the internal coordinates are read from zmat.inp).

In the above script, the steps a and (b c d e) could be combined in one cycle. However, the script is written in such a way that it permits easy modifications for using different conditions at the first step, which is often the initial part of an equilibration.

Literature cited

(1) (a) Sharma, I.; Kaminski, G. A. *J. Comput. Chem.*, *33*, 2388-2399, **2012**; (b) Kaminski, G. A.; Friesner, R. A.; Zhou, R. H. *J. Comput. Chem.*, *24*, 267-276, **2003**.
(2) Jorgensen, W.L.; Maxwell, D.S.; Tirado-Rives, J. *J. Am. Chem. Soc., 118*, 11225-11236, **1996**.
(3) (a) Prevost, M.; van Belle, D.; Lippens, G.; Wodak, S. *Mol. Phys.*, **1990**, *71*, 587-603; (b) Jorgensen, W.L. *BOSS, Version 3.6*; Yale University: New Haven, CT, 1995; (c) Stern, H.A.; Kaminski, G.A.; Banks, J.L.; Zhou, R.H.; Berne, B.J.; Friesner, R.A. *J. Phys. Chem. B*, **1999**, *103*, 4730-4737.
(4) Halgren, T.A.; Damm, W. *Curr. Opin. Struc. Biol.,* **2001**, *11*, 236-242.
(5) (a) Roux, B. *Chem. Phys. Lett.* **1993**, *212*, 231-240; (b) Straatsma, T.P.; McCammon, J.A. *Mol. Simul., 5*, 181-192, **1990**. (c) Straatsma, T.P.; McCammon, *Chem. Phys. Lett., 167*, 252-254, **1990**. (d) Straatsma, T.P.; McCammon, *Chem. Phys. Lett., 177*, 433-440, **1991**.
(6) Jorgensen, W. L.; Ravimohan, C. *J. Chem. Phys., 83*, 3050-3054, **1985**.
(7) Zwanzig, R. W. *J. Chem. Phys., 22*, 1420-1426, **1954**.
(8) (a) *Jaguar v3.5*, Schrödinger, Inc. Portland, OR, 1998; (b) *Jaguar v4.2*, Schrödinger, Inc. Portland, OR, 2000; (c) *Jaguar, v7.6*, Schrödinger, LLC, New York, NY, 2009.
(9) XChemEdit, version 0.91, Dongchul Lim and William L. Jorgensen, Department of Chemistry, Yale University, New Haven, CT, 1999