Optimal Position of Solute for Simulations

MIHALY MEZEI

Department of Physiology and Biophysics, Mount Sinai School of Medicine, CUNY, New York, New York 10029

Received 28 May 1996; accepted 9 September 1996

ABSTRACT: It is shown that, by optimizing the placement of a solute in a solvent droplet or in a periodic simulation cell, the number of solvent molecules can be reduced without affecting the quality of the simulation. © 1997 by John Wiley & Sons, Inc. *J Comput Chem* **18**: 812–815, 1997

Keywords: computer simulation, solvent layer

Introduction

Simulations of solvated molecules are generally performed either under periodic boundary conditions or in a droplet of solvent containing the solute. While realistic simulation of solvated systems requires the presence of several solvent layers around the solute, each additional layer increases progressively the number of solvent molecules to be included in the model, resulting in a concomitant large increase in the computing time requirements for simulations. Once the required number of layers is completed, still additional solvent molecules are needed whose sole role is to keep the model from producing artifactual effects (for droplets) or to just fill the space (for systems with periodic boundary conditions). The purpose of this article is to show that, in most cases, the position of the solute within the simulation system affects the number of additional solvent molecules needed for a given level of treatment; thus, by its judicious choice, it is possible to reduce significantly the number of these additional solvent molecules by optimal positioning of the solute within the droplet or the periodic cell. Because such optimizations can be performed at negligible or small computational expense, the net result is a significant saving in both computation time and storage requirements without compromising the accuracy of the results.

For simulations without periodic boundary conditions it has been shown that surrounding a nonspherical solute with a solvent layer of uniform thickness will lead to artifactual stress due to surface tension.[1] Thus, the initial layer has to be completed into a sphere. The location of the center of this sphere, however, will affect the number of extra solvent molecules needed to fill in this sphere. This number will be minimal if the *sphere's center*

Correspondence to: M. Mezei

coincides with the center of the smallest sphere that still encloses the solute. Besides reducing the number of solvent molecules required to form a layer of given minimum thickness, use of the optimized center will also minimize the fluctuation in the layer thickness around the solute, resulting in a more well-balanced system. Note that, when the solvent is water, it is reasonable to exclude the hydrogens of the solute during its positioning because, independent of its polarity, the distance of a hydrogen-bonded water's oxygen from the nearest heavy atom on a solute (containing first-row atoms) is about 3 Å.

Periodic boundary conditions have been introduced to avoid the introduction of an explicit boundary. However, they also impose unwanted periodicity and symmetry onto the system. The effect of periodicity, fortunately, can be kept small by keeping the periodic images as far apart as possible. Usually the image–image distance is controlled by the edge(s) of the periodic cell, most frequently chosen as a cube or a rectangular box. It has been long recognized, however, that the cubic box for periodic boundary conditions is generally not optimal and other simulation cell shapes can result in larger image–image distances for a given volume (i.e., number of solvent molecules). For bulk liquid simulations the optimum is reached if one chooses the cell with the largest inscribed sphere for a given volume. For simulating a solution with an arbitrary shaped solute it is *the closest distance between two solute atoms on different solute images that is to be maximized*. Generally, this optimum is approximated by choosing a simulation box that follows the shape of the solute (e.g., a long rectangular cell for potential of mean force calculations or a hexagonal prism or simulating a DNA fragment). However, in most cases, the smallest image–image distance can also be increased by optimizing the orientation of the solute within this cell.

To see the effect of both the box shape and solute orientation on the smallest image–image distance, consider a rodlike molecule of length l in the two-dimensional periodic system shown in Figure 1. The repeating unit is a rectangle with edges X and Y along the x- and y-axes, respectively. The smallest distance, D^y, between rods translated along the y-axis is $D^y(\phi) = Y\cos(\phi)$ as seen from the triangle $A'A''C$. Clearly, this is maximal for $\phi_{\text{opt}}^y = 0$. The smallest distance between rods translated along the x-axis, D^x, is the smaller of the distances AB' and AB''. Straightforward calculation shows that D^x takes its largest value at:

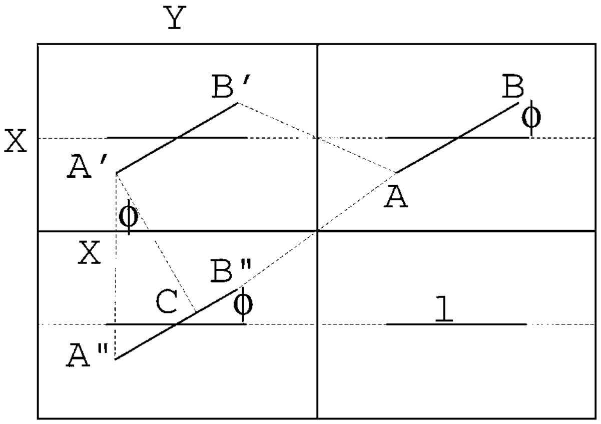

FIGURE 1. Two-dimensional periodic cells with a linear rod as solute.

$$D^x(\phi_{\text{opt}}^x) = \left(l^2 + X^2 - 2lX\cos(\phi_{\text{opt}}^x)\right)^{1/2} \quad (1)$$

where:

$$\phi_{\text{opt}}^x = \sin^{-1}(Y/l) \quad (2)$$

Thus, for $X \geq l + Y$ the optimal orientation, ϕ_{opt}, is at $\phi_{\text{opt}} = \phi_{\text{opt}}^y$ (and for $Y \geq l + X$ is at $\phi_{\text{opt}} = \pi$). For intermediate cases, however, the optimal angle is obtained by simultaneous consideration of D^x and D^y. For $D^x(\phi_{\text{opt}}^x) \leq D^y(\phi_{\text{opt}}^x)$, the optimal orientation is clearly ϕ_{opt}^x. However, when the opposite is true, the optimum is reached at a smaller angle, obtained from the condition:

$$D^y(\phi_{\text{otp}}) = Y\cos\phi_{\text{opt}} = \sqrt{l^2 + X^2 - 2lX\cos(\phi_{\text{opt}})} = D^x(\phi_{\text{opt}}) \quad (3)$$

giving:

$$\cos(\phi_{\text{opt}}) = \frac{-lX + \sqrt{l^2X^2 + l^2 + X^2}}{Y^2} \quad (4)$$

The above discussion not only shows the potential importance of optimizing the orientation of the solute but it also demonstrates the importance of the choice of relative magnitudes of the simulation cell parameters. In cases where the optimal orientation of the solute does not depend on all of the cell parameters, the remaining ones can be reduced without compromising the optimum, thereby reducing the volume of the simulation cell without reducing the smallest image–image approach (e.g., for $Y > l + X, Y$ can be reduced to

$l + X$ without affecting the smallest image–image distance).

Joint consideration of the optimal orientation and cell parameters can be formulated as the problem of minimizing the volume of the simulation cell while keeping the nearest approach in the optimal orientation constant. Here the cell parameterization can include not only the values of the cell dimension but the shape of the cell. The choice of the face-centered cubic or truncated octahedral cell for bulk liquid simulation is made in this spirit. While numerical realization of such optimization can be daunting in general, our simple model can be examined from this point of view. In particular, it can be shown that for a given smallest image–image approach, D, the optimal cell has $X = l + D$ and $Y = D$. Obviously, for more complex solutes there is unlikely to be an analytical solution to this optimization problem, but this example still provides some justification of the aforementioned prescription which required that the shape of the simulation cell follow the shape of the molecule as closely as possible.

Method

The problem of finding the smallest sphere enclosing a given set of points is an old one. Several fast algorithms exist (for a comparison and list of references see ref. 2), and it has been shown that a solution can be obtained in $O(n)$ time.[3] Our realization is a generalization of the century-old method referred to in ref. 2 as the Chrystal–Peirce algorithm[4,5] to three dimensions. For a system of $O(10^3)$ atoms it obtains the result practically instantaneously.

For the optimal placement of a solute in the periodic cell we start with cell parameters that are chosen to be the smallest possible still enclosing the solute in its initial orientation, (preferably chosen to align the principal axes of the molecule with the coordinate axes) incremented by half of the smallest image–image distance acceptable.

Next, we optimize the solute orientation in this cell to yield the largest value for the smallest image–image distance. The smallest image–image distance for a given orientation is obtained by considering all pairs of atoms $\{i, j\}$ of the solute and, for every pair, one has to consider all images. This optimization has to find the Euler angles ϕ, ψ, ϑ such that the rotation matrix, $\mathbf{R}(\phi, \psi, \vartheta)$, maximizes:

$$\min_{\{i,\, j,\, k\}} \left[\mathbf{R}(\phi, \psi, \vartheta)\mathbf{r}_i - \mathbf{R}(\phi, \psi, \vartheta)\mathbf{r}_j - \mathbf{C}_k\right]^2 \quad (5)$$

where $\mathbf{r}_i$ are the coordinates of the solute atoms and $\mathbf{C}_k$ are the centers of the image cells surrounding the simulation cell. This optimization is, in general, more exacting computationally than finding the center of the smallest enclosing sphere. In our implementation the simplex method from *Numerical Recipes*[6] was used. The symmetry of the placement of the image cells offer some nonnegligible simplification, however.

The orientational optimization can be significantly speeded up if only atoms on the surface of the solute are considered. For this purpose various definitions can be considered for a surface atom. From the mathematical point of view this is a ''risky'' operation, because for a given definition of surface it is likely that a set of points can be constructed for which the optimum would be missed. Fortunately, atoms on a molecule are distributed in a rather uniform pattern and such reduction is quite ''safe.'' Using the definition that an inside atom is bonded to at least three more atoms, no optimum was found missed.

Once the optimal orientation is selected, the cell parameters are examined to determine if any of them can be reduced without affecting the optimum. This can be done by finding out the smallest value of each that still does not reduce the value of the smallest image–image distance.

An additional complication is the nonlinearity of the minimization problem. As a result only a local minimum is generally found. This necessitates the repetition of the optimization starting from several different starting orientations. Note that for elongated solutes each random orientation has to be checked if the solute is within the simulation cell assumed.

Further optimization of the cell parameters is still possible, as formulated at the end of the Introduction. However, carrying it out rigorously would be very costly because each function evaluation would require a new optimization of the orientation. Also, as the cell shape has been already chosen with the solute shape in mind, it is not likely to yield much further improvement. Thus, for the purpose of comparison of the various local minima found by optimizations using different starting orientations, we found it useful just to find a scaling factor that scales down the cell parameters after optimization until the closest image–image distance is reduced to the originally targeted value.

Results and Discussion

These optimization techniques were applied to the moderate-sized proteins, bovine pancreatic trypsin inhibitor (BPTI) and calmodulin. Compared to the radius of the sphere centered at the center of mass, used by standard modeling programs such as CHARMM[7] or Insight[8] for immersing a molecule into a solution, the optimal sphere radius was found to be smaller by about 1 Å. Optimization of the orientation showed larger gains although at the expense of significantly longer computations (5–10 minutes on an SGI Challenge using R4400 CPU).

For several of the optimizations, it was possible to decrease one of the cell edges without affecting the nearest approach (in one case by 5 Å, equivalent to the elimination of ca. 500 waters). The importance of trying several different starting orientations should also be emphasized. In our experience, it seems that, as a minimum, five different starting orientations should be tried. Besides generating random rotations, starting from an orientation that rotates the original *x*-axis to match the diagonal of the cell appeared to be useful.

Starting form the crystal structure orientation, optimization increased the smallest image–image distance by 1.5 Å and 6.5 Å for the BPTI and calmodulin, respectively. If in each orientation the cells are scaled down until the smallest image–image distance is reduced to 18 Å the cell volumes of 97,350 Å^3 and 261,379 Å^3 in the initial orientation are reduced to respectively volumes of 87,406 Å^3 and 187,250 Å^3.

Such reductions of volume are equivalent to eliminating 331 and 2464 waters, respectively. To put these savings into the context of computational expenses, note that the storage requirements increase linearly with the number of atoms in the system and, more significantly, computation time increases (depending on the cutoff scheme employed) up to quadratically with the number of atoms. Such gains with minimal investments are especially important because it is now becoming increasingly recognized that many properties of macromolecular solutions require simulations on the nanosecond time scale.

The optimization algorithms describe din this study have been incorporated into a Fortran-77 program called Simulaid that is available from the author (e-mail: mezei@msvax.mssm.edu). For the optimal orientation determination, the program automatically generates several input orientations and performs the optimization steps described without further manual intervention.

Acknowledgment

Useful and enjoyable discussions with Dr. Frank Guarnieri helped to shape this project into its final form. Dr. Nimrod Megiddo is thanked for interesting correspondence and information on References 2 to 5.

References

1. M. Mezei, *Information Q.*, CCP5, Daresbury Lab, **37**, 38 (1993).
2. D. W. Hearn and J. Vijay, *Operat. Res.*, **30**, 777 (1982).
3. N. Megiddo, *SIAM J. Comput.*, **12**, 759 (1983).
4. J. J. Sylvester, *Philosophical Magazine*, **10** (fourth series), 203 (1860).
5. G. Chrystal, *Proc. Edinburgh Math. Soc.*, **3**, 30 (1885).
6. W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, (1989) *Numerical Recipes*, Cambridge University Press, Cambridge.
7. B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus, *J. Comput. Chem.*, **4**, 187 (1983).
8. InsightII, software program form Molecular Simulation, Inc. (formerly Biosym Technologies, San Diego).