Packing Optimization for Automated Generation of Complex System's Initial Configurations for Molecular Dynamics and Docking

JOSÉ MARIO MARTÍNEZ,[1] LEANDRO MARTÍNEZ[2]
[1]*Department of Applied Mathematics, IMECC-UNICAMP, University of Campinas, CP 6065, 13081-970 Campinas SP, Brazil*
[2]*Institute of Chemistry, University of Campinas, Campinas SP, Brazil*

Received 27 May 2002; Accepted 17 September 2002

Abstract: Molecular Dynamics is a powerful methodology for the comprehension at molecular level of many chemical and biochemical systems. The theories and techniques developed for structural and thermodynamic analyses are well established, and many software packages are available. However, designing starting configurations for dynamics can be cumbersome. Easily generated regular lattices can be used when simple liquids or mixtures are studied. However, for complex mixtures, polymer solutions or solid adsorbed liquids (for example) this approach is inefficient, and it turns out to be very hard to obtain an adequate coordinate file. In this article, the problem of obtaining an adequate initial configuration is treated as a "packing" problem and solved by an optimization procedure. The initial configuration is chosen in such a way that the minimum distance between atoms of different molecules is greater than a fixed tolerance. The optimization uses a well-known algorithm for box-constrained minimization. Applications are given for biomolecule solvation, many-component mixtures, and interfaces. This approach can reduce the work of designing starting configurations from days or weeks to few minutes or hours, in an automated fashion. Packing optimization is also shown to be a powerful methodology for space search in docking of small ligands to proteins. This is demonstrated by docking of the thyroid hormone to its nuclear receptor.

 J Comput Chem 24: 819–825, 2003

Key words: large-scale optimization; box constraints; molecular dynamics; docking

Introduction

Molecular Dynamics (MD) is a powerful technique for the comprehension at molecular level of a great variety of chemical processes.[1] With the enhancement of computational resources, more and more complex systems are being studied, as large biochemical systems,[2] zeolite encapsulated liquids[3] and many-component solutions.[4] Simulations are accompanied by the development of adequate theories for their analyses and corresponding software for manipulation and visualization of coordinate and trajectory files.[5,6] Many packages for performing simulations are now commercially or freely available.[5,7,8]

The simulations need starting points that must have adequate energy requirements, given by experimental data. The density of the system must be provided, specifying the size of the simulation box. In general, the energy distribution of the molecules is adjusted to the desired temperature by scaling. Both simulation and temperature scaling involve the resolution of Newtonian equations of motion by means of a numerical method.[1,9] The kinetic energy of the atoms is scaled in such a way that the temperature fits the desired one at every step. This procedure is repeated at every integration step for a reasonable time period and takes the total energy of the system to the thermodinamic internal energy at the corresponding temperature.[1]

However, if the starting configuration has close atoms, the temperature scaling is disrupted by excessive potentials that accelerate molecules over the accepted velocities for almost any reasonable integration time step. In fact, the starting coordinate file must be reliable in the sense that it must not exhibit overlapping or close atoms, so that temperature scaling can be performed with reasonable time steps for a relatively fast energy equilibration of the system.

For simple liquids or solutions with small solutes, the initial configurations are usually regular lattices with random velocities

Correspondence to: J. M. Martínez; e-mail: martinez@ime.unicamp.br

Contract/grant sponsor: PRONEX-Optimization, FAPESP; contract/grant number: 2001-04597-4

Contract/grant sponsors: CNPq and FAEP-UNICAMP

attributed to the atoms. For polymers or biopolymers in solution, it is common to remove the molecules that are in close contact, assuming that the removed solvent molecules will correspond approximately to the volume of large solutes. More complex systems generally need manipulation or energy optimization to make them reliable, maintaining solvent and solute concentrations and densities. This is, in general, a very hard work that might take several days or even weeks to generate an initial point that guarantees safe pairwise potentials.

In this article, we suggest to represent the problem of finding the initial positions of the molecules as a "packing problem" (see ref. 17). The goal is to place known objects in a finite domain in such a way that the distance between any pair of points of different objects is larger that a threshold tolerance. In our case, objects are molecules and points are atoms. Following this idea, we define a mathematical model that gives rise to an optimization problem. The mathematical (optimization) problem consists in the minimization of a function of (generally many) variables, subject to bounds. This is the so-called box-constrained minimization problem in mathematical literature (see refs. 19–24, among others).

For solving the optimization problem, we use BOX-QUACAN (see ref. 22), a well-established method for which freely available software exists (see ref. 26). BOX-QUACAN is a local-minimization method, in the sense that, in theory, its limit points are only guaranteed to be critical points of the original problem. For this reason, we coded a multistart procedure for our problem, by means of which different initial approximations are given with the aim of finding a global minimizer, among the local ones.

The same procedure is proposed to handle the general problem of docking of small ligands to proteins. Usually, the docking problem involves evaluation of potential-energy functions and exhaustive search in the space for local minima. Because the evaluation of Lennard–Jones potentials is very time-consuming, the packing model is better for space searching, and requires only local energy minimization for determination of final configurations with effective ligand–protein interactions. This methodology does not consider potential-energy functions, and turned out to be quite efficient for finding candidate cavities in the protein structure for the docking of the rigid 3,3′,5′-triiodotreonine (T3) to its nuclear receptor. This article includes a comparison of this method against the usual technique of potential energy minimization.

This article is organized as follows. In the next section we present the mathematical model and we discuss some of its essential features. Then we describe properties and usage of BOX-QUACAN. Finally, we discuss the input parameters, and then the examples are presented and discussed.

The Packing Model

Let us call nmol the total number of molecules that we want to place in a region of the three-dimensional space defined by the bounds $\ell_k \le x_k \le u_k$, $k = 1, 2, 3$. For each $i = 1, \ldots,$ nmol, let $n\text{atom}(i)$ be the number of atoms of the ith molecule. Molecules can be grouped in different types (water, hormonium, and so on), but this is irrelevant for the model description. Each molecule is represented by the orthogonal coordinates of its atoms. We call barycenter of a molecule to the point whose coordinates are the arithmetic averages of the coordinates of the atoms. To facilitate the visualization, assume that the origin is the barycenter of all the molecules. For all $i = 1, \ldots,$ nmol, $j = 1, \ldots, n\text{atom}(i)$, let

$$A(i, j) = (a_1^{ij}, a_2^{ij}, a_3^{ij})$$

be the coordinates of the jth atom of the ith molecule.

Now, suppose that one rotates the ith molecule sequentially around the axes x_1, x_2, and x_3, being α_i, β_i, and γ_i the angles that define such rotations. Moreover, suppose that after these rotations, the whole molecule is displaced so that its barycenter, instead of the origin, becomes $C_i = (c_1^i, c_2^i, c_3^i)$. These movements transform the atom of coordinates $A(i, j)$ in a displaced atom of coordinates

$$P(i, j) = (p_1^{ij}, p_2^{ij}, p_3^{ij}).$$

Observe that $P(i, j)$ is always a function of $(C_i, \alpha_i, \beta_i, \gamma_i)$ but we do not make this dependence explicit to simplify the notation.

Our objective is to find angles α_i, β_i, γ_i and displacements C_i, $i = 1, \ldots,$ nmol, in such a way that, for all $j = 1, \ldots, n\text{atom}(i)$, $j' = 1, \ldots, n\text{atom}(i')$,

$$\|P(i, j) - P(i', j')\|^2 \ge \varepsilon, \text{ whenever } i \ne i' \tag{1}$$

and, for all $i = 1, \ldots,$ nmol, $j = 1, \ldots, n\text{atom}(i)$,

$$\ell_k \le p_k^{ij} \le u_k \text{ for } k = 1, 2, 3, \tag{2}$$

where $\varepsilon > 0$ is a user-specified tolerance. The symbol $\|\cdot\|$ stands for the usual Euclidian distance. In other words, the rotated and displaced molecules must remain in the specified box and the squared distance between any pair of atoms must not be less than ε.

The objectives (1) and (2) lead us to define the following merit function f:

$$\begin{aligned} f(C_1, \ldots, C_{\text{nmol}}, \alpha_1, \beta_1, \gamma_1, \ldots, \alpha_{\text{nmol}}, \beta_{\text{nmol}}, \gamma_{\text{nmol}}) \\ = \sum_{i=1}^{\text{nmol}-1} \sum_{i'=i+1}^{\text{nmol}} \sum_{j=1}^{n\text{atom}(i)} \sum_{j'=1}^{m\text{atom}(i')} \max\{0, \varepsilon - \|P(i, j) - P(i', j')\|^2\}^2 \\ + \sum_{i=1}^{\text{nmol}} \sum_{j=1}^{n\text{atom}(i)} \sum_{k=1}^{3} \max\{0, p_k^{ij} - u_k, \ell_k - p_k^{ij}\}^2. \end{aligned} \tag{3}$$

Note that $f(C_1, \ldots, C_{\text{nmol}}, \alpha_1, \beta_1, \gamma_1, \ldots, \alpha_{\text{nmol}}, \beta_{\text{nmol}}, \gamma_{\text{nmol}})$ is nonnegative for all angles and displacements. Moreover, f vanishes if, and only if, the objectives (1) and (2) are fulfilled. This means that, if we find displacements and angles where $f = 0$, the atoms of the resulting molecules are sufficiently separated. This leads us to define the following minimization problem:

$$\text{Minimize } f(C_1, \ldots, C_{\text{nmol}}, \alpha_1, \beta_1, \gamma_1, \ldots, \alpha_{\text{nmol}}, \beta_{\text{nmol}}, \gamma_{\text{nmol}}) \tag{4}$$

$$\text{subject to } \ell_k \le c_k^i \le u_k \ \forall \ i = 1, \ldots, \text{nmol}, k = 1, 2, 3.$$

The objective function f is continuous and differentiable, although their second derivatives are discontinuous. The number of variables is $6 \times$ nmol (three angles and a displacement per molecule). The analytical expression of f is cumbersome, because it involves consecutive rotations and its first derivatives are not very easy to code. However, optimization experience lead us to pay the cost of writing a code for computing derivatives, with the expectancy that algorithms that take advantage of first-order information are really profitable, especially when the number of variables is large (see ref. 20). Having a code that computes f and its gradient, we are prepared to solve (4) using BOX-QUACAN.

The Optimization Solver

BOX-QUACAN is a box-constraint solver introduced in ref. 22. It is an iterative method that, at each iteration, approximates the objective function by a quadratic and minimizes this quadratic model in the box determined by the natural constraints and an auxiliary box that represents the region where the quadratic approximation is reliable (trust region). If the objective function is sufficiently reduced at the (approximate) minimizer of the quadratic, the corresponding trial point is accepted as new iterate. Otherwise, the trust region is reduced. The type of problems for which BOX-QUACAN is designed is

$$\text{Minimize } f(x) \text{ subject to } x \in \Omega, \tag{5}$$

where $f : \mathcal{R}^n \to \mathcal{R}$ has continuous first derivatives and Ω is the n-dimensional box given by

$$\Omega = \{x \in \mathcal{R}^n | \ell_i \leq x_i \leq u_i\}.$$

In our case, $x = (C_1, \ldots, C_{\text{nmol}}, \alpha_1, \beta_1, \gamma_1, \ldots, \alpha_{\text{nmol}}, \beta_{\text{nmol}}, \gamma_{\text{nmol}})$, so $f(x) = f(C_1, \ldots, C_{\text{nmol}}, \alpha_1, \beta_1, \gamma_1, \ldots, \alpha_{\text{nmol}}, \beta_{\text{nmol}}, \gamma_{\text{nmol}})$.

Implementations of BOX-QUACAN for general box-constrained problems can be found in ref. 26. The main steps of the algorithm are given below:

Algorithm Box

Assume that the initial point $x^0 \in \Omega$ is given. Let k be the iteration counter and set $k \leftarrow 0$. Let $\Delta_{\min} > 0$ be the "minimum trust-region radius."

Step 1. If the projected gradient of f at x^k is small, according to some user-given tolerance, terminate the execution of the algorithm.

Step 2. Let B_k (an $n \times n$ symmetric matrix) be an approximation to the Hessian of f at x^k. Define $Q_k(d)$, the quadratic approximation of $f(x^k + d) - f(x_k)$, as

$$Q_k(d) = \frac{1}{2} d^T B_k d + \nabla f(x^k)^T d.$$

Let $\Delta \geq \Delta_{\min}$ be the current trust-region radius.

Step 3. Use QUACAN to solve, approximately, the bound constrained quadratic subproblem

$$\text{Minimize } Q_k(d) \tag{6}$$

subject to

$$x^k + d \in \Omega, \ -\Delta \leq d_i \leq \Delta \ \forall \ i = 1, \ldots, n. \tag{7}$$

Step 4. If

$$f(x^k + d) \leq f(x^k) + 0.1 \ Q_k(d) \tag{8}$$

define $x^{k+1} = x^k + d$, and $k \leftarrow k + 1$ and go to Step 1. If (8) does not hold, reduce Δ (e.g., $\Delta \leftarrow \Delta/2$) and go to Step 3.

This algorithm is especially designed to handle large-scale problems. For this reason, no factorization of matrices are used at any stage. The domain of the problem, called Ω here is, as mentioned before, an n-dimensional box. It can be divided in disjoint faces of dimensions $0, 1, \ldots, n$ according to the variables that, at each point, are at the bounds. For example, vertices are faces of dimension 0, edges are faces of dimension 1 and the interior of the box is an n-dimensional face.

QUACAN is the quadratic solver used to deal with the subproblems of the box-constrained algorithm BOX. The subproblem consists in the minimization of a quadratic function subject to an auxiliary box, which is the intersection of the original box with the trust region. QUACAN visits the different faces of its domain using conjugate gradients on the interior of each face and "chopped gradients" as search directions to leave the faces. See refs. 25, 22, and 18 for a description of the current implementation of QUACAN. At each iteration of QUACAN, a matrix-vector product of the Hessian approximation and a vector is needed. Because Hessian approximations are difficult to compute, we use the "Truncated Newton" approach, so that each Hessian $\times$ vector product is replaced by an incremental quotient of gradients along the direction given by the vector. Namely, in (6)–(7) we use the approximation

$$B_k d \approx \frac{\nabla f(x_k + hd) - \nabla f(x_k)}{h}$$

where $h > 0$ is a small increment.

The general convergence results of BOX-QUACAN have been given in ref. 22. Because the objective function has continuous partial derivatives and the Hessian approximations are bounded, every limit point of a sequence generated by BOX is a critical point. This means that the algorithm approximates a point that satisfies first-order optimality conditions with an arbitrary precision. In general, these critical points are local minimizers. In other words, the algorithm eventually stops at Step 1 satisfying the optimality criterion.

Program Usage

Our computer code is called Packmol. For generating a coordinate file box it requires the cartesian coordinates of an isolated molecule of each type, the number of molecules of each type, the minimum desired distance between pairs of atoms, the box-type specification and the coordinates of the box. Cartesian coordinates of isolated molecules are simply obtained[6] and the number of

molecules of each type must be defined by the user. The minimum desired distance may be set to the approximate maximum hydrogen bond distance of 2 Å even if atoms with very large Van der Waals radius are present (see next section) or to any other value if necessary.

The box-type specification is a useful resource of the program that allows the molecules to be generated with specified positions and rotations. This resource is useful for creating boxes with interfacial molecules or any other ordered structure. For usual boxes, maximum and minimum cartesian coordinates of the atoms of each type should be set, specifying the box.

This code is also valuable for docking of small molecules in large structures as zeolites or proteins.[29] Docking is accomplished by establishing a fixed position for the macromolecule and defining the ligand box within the protein or zeolite cores.

Numerical Experiments

For illustrating our approach, three complex systems were generated: (a) a large box containing a 929-atoms protein solvated by a 7 mol L^{-1} solution of urea in water represented by 1063 water and 182 urea molecules. (b) A 600-molecules box with 10 different components represented by 60 molecules each. (c) An interface of 1000 water and 200 chloroform molecules with a 35-atoms hydrofobic molecule positioned at the interface. A fourth group of starting configurations [test problem (d)] with three different molecules in three different concentrations is presented. These are rather simple configurations; however, they were already used in molecular dynamics simulations.[27] The convergence of energies from the initial box to the thermalized system are shown here. The protein volume in test problem (a) was calculated using the package Tinker.[5] The minimum allowed distance was set to 2 Å in all test problems, and box sizes were set to fit the experimental densities.

The starting configurations found for the test problems (a), (b), and (c) are shown in Figure 1. Larger problems have greater initial objective function values and require more iterations, as expected. The solvated protein is an especially hard problem for the optimization algorithm because solvent molecules have limited mobility within the box due to the presence of the large protein molecule. However, providing starting points with homogeneous solvent density (a resource already implemented in Packmol) the solution was also found. For mixtures with interfaces or many components, as in test problems (b) and (c), the optimization is easier and convergence is achieved in a small number of iterations.

In Figure 1a the solvated protein is shown. The distribution of water and urea molecules around the protein is uniform due to the randomness of the initial point. The protein, represented with a space-filling model, had its barycenter fixed at the center of the box. In Figure 1b the interface between water and chloroform is presented, and the 3,5,3′-triiodo-L-thyronine (T3) molecule is emphasized. The ability of the program for specifying molecular or box coordinates is well represented in this picture.

Figures 1c, d, and e show three different representations of the 10-component mixture. Figure 1c shows the distribution of water and sodium ions, as spacefilling models, and Figure 1d shows ball-and-stick representations of the benzoic acid and terbutanol molecules. These pictures well represent the distribution of these species in the box. On the other hand, Figure 1e shows all the species as spacefilling models with one color per specie type. This picture well represents the compactness of the structure, which is not evident with stick representations.

Finally, the three-component mixtures of the last example are composed by water, urea, and the hydrofobic iodine containing molecule T3. This is an interesting system for testing the energy convergence in thermalization from an initial box because the Lennard–Jones cutoff radius for the iodine atom is 4.3 Å,[31] which is much larger than the minimum allowed distance of 2 Å. The proximity of two atoms leads to the disruption of the system whenever the potential energy given by the Lennard–Jones interaction is too large. Because the cutoff radius is the crucial parameter for the repulsion at short distances, the smooth convergence of the energy in a system containing iodine reveals that the minimum allowed distance is restrictive enough for building valuable starting configurations. Figure 2 shows the energy convergence for three boxes containing one T3 molecule and different concentrations of urea in water, giving a total of 479 molecules. The time step used for thermalization was 1 fs, and the numerical integration was performed using the well known algorithm Shake.[9] The energy of these systems converge to the equilibrium internal energy of the solution after only 5 ps of thermalization, which is quite acceptable. Because the iodine atom possess one of the largest Van der Waals radius, the distance of 2 Å seems to be good enough for any system.

A comparison of an MD run of an unprepared configuration and a configuration designed by our method is in order. A simple system was chosen, similar to the mixture of test problem (d), but without urea molecules. The mixture was composed of 478 water molecules and one T3 molecule. The first test was to build a water box and soak the T3 into it with the xyzedit tool of the package Tinker, designed for this purpose. This program sets the molecules within the box with random positions for their barycenters. The resulting configuration had 758 atoms of different molecules with a distance less than 2.0 Å, including 262 with distances less than 1.0 Å. The simulation, with the dynamic program of the same package, with a time step of 0.001 fs and a temperature-coupling bath at 298 K at every step started with a temperature of 138,231 K that went to 9,948,740 K in nine dynamic steps, when the system totally disrupted. This means that even thermalization was not possible for this system due to very repulsive interactions. Second, a random starting configuration was generated and its OPLSAA potential energy was optimized with two procedures: the L-BFGS[10,11] implemented in the program *minimize* of the package Tinker and the BOX-QUACAN algorithm. The system potential energy was 9.2×10^{12} kcal mol^{-1} before optimization. The L-BFGS optimization failed (and was interrupted) due to ill-definition of the objective function at trial points. (Potentials tend to infinity when the distance between atoms tend to zero.) Simulations starting from the final configuration obtained by L-BFGS and using the code dynamic disrupted at the first integration step.

The BOX-QUACAN full-potential optimization produced a configuration (very likely, close to a local minimizer) with 1.2×10^7 kcal mol^{-1} of potential energy. Simulation from the best configuration obtained also disrupted at the first step even with very short time steps, as 0.1 fs. These results with two different

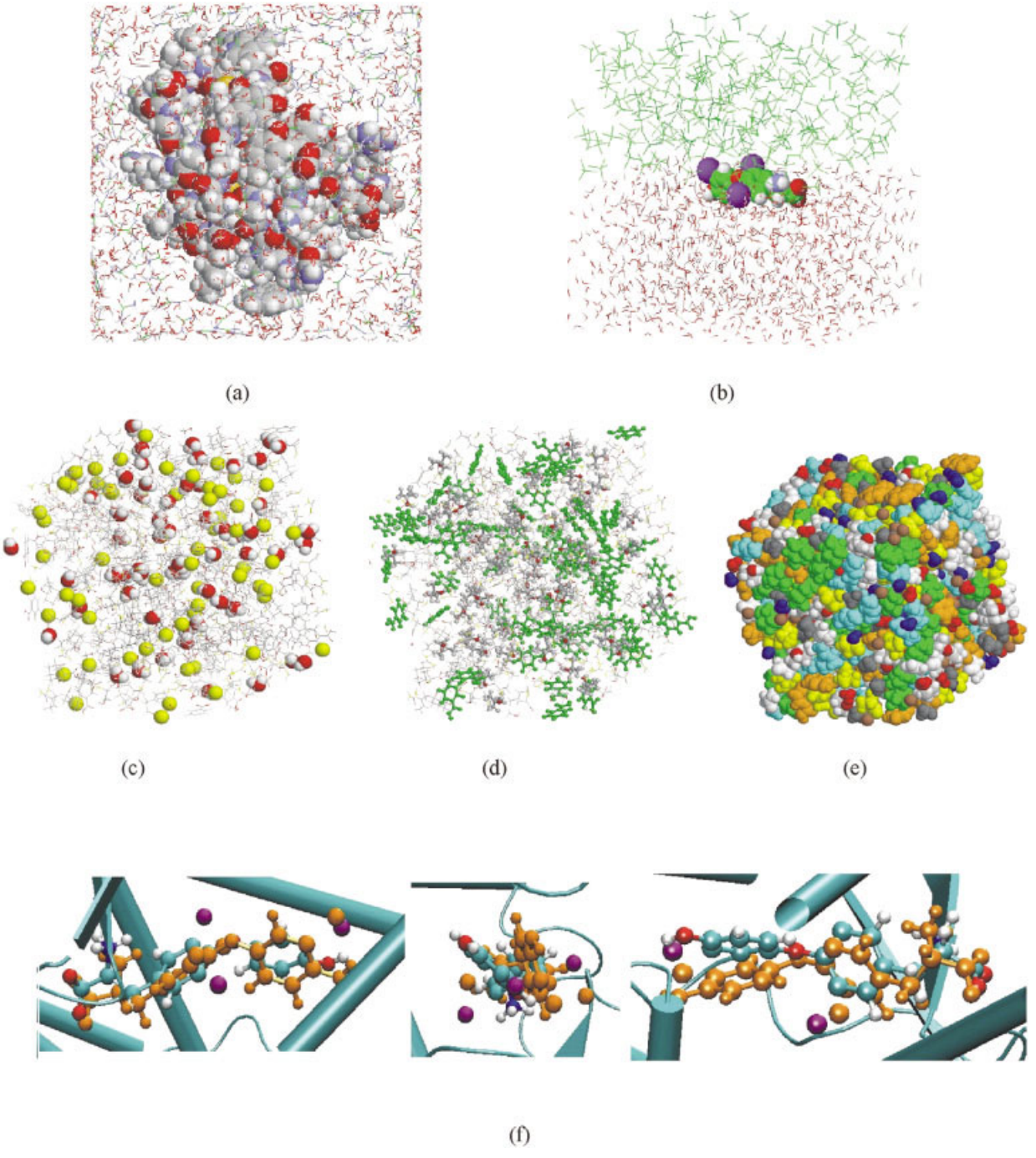

Figure 1. Pictorial representations for test problems (a), (b), and (c) and for the docking problem. Solvated protein: (a) Interface; (b) ten-component mixture; (c–e) comparison between the solutions of the docking with packmol (orange) and the docking with full OPLSAA potential evaluation (f).

implementations of two different and well established optimization methods show that the optimization of full potentials, although it might work in some cases, is not a good strategy for the general problem of generating starting configurations. This is due to the fact that full potential energy functions are not defined over all the space and their complexity turns the optimization of potentials a very hard task. We also tried to use simulated annealing to find an initial configuration for MD but, as expected, the process finished with disruption because, essentially, simulated annealing is an MD technique in this case.

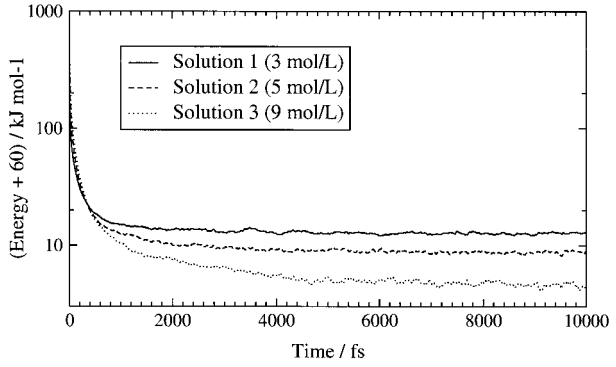

Figure 2. Energy convergence profiles for the solutions of test problem (d). Note energy convergence soon after 5 ps of thermalization in all cases.

Other alternative strategies, as building regular lattices, removing manually overlapping molecules and editing coordinate files are sometimes successful, but demand a lot of human time and are increasingly difficult or even impossible for large or complex systems. The same trial system was also generated using packmol in less than 3 min and the simulation ran smothly with a integration time step of 1.0 fs. Thermalization was achieved in less than 5 ps as in the slightly more complex, urea containing, systems of test problem (d).

The docking example is as follows: the 3989 atom TR-βLBD nuclear receptor protein[30] was fixed with its barycenter at the origin and no rotation. The structure is in its holo form, i.e., it has the ligand (T3 molecule, 35 atoms) placed at its actual binding site. The ligand was removed and one T3 molecule was randomly placed within a box that included the protein core. Two hundred cycles of 100 iterations of BOX-QUACAN were taken with random initial coordinates for the ligand molecule. The minimum distance allowed was set to 1.6 Å. This small distance was chosen aiming the definition of an effective space-searching algorithm for further energy minimization. Twelve solutions were found for the docking problem, that were further optimized with the full OPLSAA potentials[12,13,14] for the T3 and protein. Only intermolecular interactions were computed and the molecules were kept rigid. The total packing time was 38.15 min and the OPLSAA potential energy minimization of the packing solutions took 43.33 min, which gives a total docking time of 81.48 min. The average OPLSAA energy of these solutions was -10547 kcal mol^{-1}. The average time for finding each of these solutions was 6.79 min.

The full OPLSAA potential energy minimization from random starting configurations was also taken for comparison against our packing method. This procedure is similar to the most popular docking procedures in the sense that full Lennard–Jones and charge-to-charge interactions are optimized.[15,16] BOX-QUACAN was used as the optimization algorithm. It must be mentioned that grid-based methodologies[16] exist that turn the function evaluation faster (although not exact). Of course, the same grid techniques can be applied to our packing function based on the distance. We will show that the packing technique combined with local full potential optimization produces solutions of the same value and with similar frequency as other methods and, on the other hand, is much faster. As in the packing procedure, 200 cycles of OPLSAA potential energy optimization were performed. Only intermolecular interactions were evaluated and the protein and the hormone were kept rigid. The overall optimization time was 357.70 min. Because the packing procedure gave 12 solutions, the 12 best solutions of this procedure were taken for comparison. The average OPLSAA potential energy for these solutions was -10869 kcal mol^{-1}, which is less than the average energy of the potential optimized packing solutions by 3%. However, taking the 13 best solutions, we get an average energy of -10376 kcal mol^{-1}, which is higher than the average optimized potential packing solutions. The comparison between the energy of the solutions shows, then, that the quality of the solutions, taken as their OPLSAA energy, is similar. The comparison of the best solution of each methodology, as shown in Figure 2(f) confirms this conclusion. Both structures, as all the best solutions of each method, are placed in the active site of the protein. The average computer time for each of the 12 solutions for the full OPLSAA optimization was, however, 29.67 min. Because there are no meaningful differences regarding the quality of the solutions and the computational time necessary for finding a solution with the packing procedure is 4.4 times less, we conclude that packing optimization is indeed a good strategy for space searching.

Two characteristics of the objective function proposed here make the packing strategy more successful than full potential optimization: first, its evaluation is much faster because the function is much simpler than the Lennard–Jones and charge-to-charge interactions. Second, the function proposed is continuous, has continuous first derivatives, and is defined over all the space. This makes the optimization with almost any procedure faster and more effective than Lennard–Jones-based minimizations. The implementation of this strategy, with BOX-QUACAN or other optimization method, is simple and deserves to be considered in the development of large-scale docking packages.

These results show that packing optimization is a very fast and effective procedure for space searching, because cavity searching does not need potential-energy evaluations. The development of this methodology for docking, including flexibility for the ligand molecule and evaluation of solutions by local energy minimization, are future steps in the development of Packmol.

Conclusions

Packing optimization is a useful tool for building molecular dynamics starting configurations, which is almost independent of system's complexity, and, as so, is able to avoid cumbersome manipulation of structures. The original motivation for this approach was to build rather simple boxes, as the one presented in Example 4. For these boxes, Packmol finds adequate starting configurations in few minutes. Therefore, we encourage its use even for very simple systems, because it provides molecules that are randomly distributed in the box, making the process of thermalization faster than when one uses regular lattices. The objective function proposed in this article is much cheaper to compute than Lennard–Jones potentials and provides adequate configurations as well. For docking, our approach enhances space searching so that only local energy optimization for final conformational energy

comparison is required. Our code is already being successfully used in the Molecular Dynamics research group of Prof. Munir S. Skaf both for simple mixtures[27,28] and more complex ones, as water encapsulated zeolites.[29]

Acknowledgments

We are indebted to two anonymous referees for useful comments that helped us to improve this article.

References

1. Frenkel, D.; Smit, B. Understanding Molecular Simulations, From Algorithms to Applications; Academic Press: San Diego, 1996.
2. Duan, Y.; Kollman, P. A. Science 1998, 282, 740.
3. Arya, G.; Chang, H. C.; Maginn, E. J. J Chem Phys 2001, 115, 8112.
4. Vishnyakov, A.; Neimark, A. V. J Phys Chem B 2001, 105, 7830.
5. Ponder, J. W. TINKER Software Tools for Molecular Design, Version 3.8, Oct. 2000.
6. Schaftenaar, G.; Noordik, J. H. J Comput-Aided Mol Design 2000, 14, 123.
7. Brooks, B. R., et al. J Comput Chem 1983, 4, 187.
8. Pearlman, D. A., et al. Comp Phys Commun 1995, 91, 1.
9. Ryckaert, J. P., et al. J Comput Phys 1977, 23, 327.
10. Liu, D. C.; Nocedal, J. Math Prog 1989, 45, 503.
11. Nocedal, J. Math Comp 1980, 35, 773.
12. Jorgensen, W. L.; Maxwell, D. S.; Tirado–Rives, J. J Am Chem Soc 1996, 117, 11225.
13. Maxell, D. S.; Tirado–Rives, J.; Jorgensen, W. L. J Comput Chem 1995, 16, 984.
14. Jorgensen, W. L.; McDonald, N. A. THEOCHEM-J Mol Struct 1998, 424, 145.
15. Vieth, M., et al. J Comput Chem 1998, 19, 1623.
16. Morris, G. M., et al. J Comput Chem 1998, 19, 1639.
17. Aste, T.; Wearie, D. The Pursuit of Perfect Packing; Institute of Physics Publishing: London, 2000.
18. Bielschowsky, R. H., et al. Invest Oper 1998, 7, 67.
19. Conn, A. R.; Gould, N. I. M.; Toint, Ph. L. Trust-Region Methods; SIAM-MPS: Philadelphia, 2000.
20. Dennis, J. E.; Schnabel, R. B. Numerical Methods for Unconstrained Optimization and Nonlinear Equations; Prentice-Hall: Englewood Cliffs, NJ, 1983.
21. Facchinei, F.; Júdice, J. J.; Soares, J. SIAM J Opt 1998, 8, 158.
22. Friedlander, A.; Martínez, J. M.; Santos, S. A. Appl Math Opt 1994, 30, 235.
23. Krejić, N., et al. Comput Opt Appl 2000, 16, 247.
24. Lin, C. J.; Moré, J. J. SIAM J Opt 1999, 9, 1100.
25. Friedlander, A.; Martínez, J. M. SIAM J Opt 1994, 4, 177.
26. www.ime.unicamp.br/~martinez/software.htm.
27. Martínez, L.; Skaf, M. S. Unpublished results.
28. Sonoda, M. T.; Skaf, M. S. Unpublished results.
29. Martins, L. R.; Skaf, M. S. Unpublished results.
30. Wagner, R. L., et al. Nature 1995, 378, 690.
31. Blaney, J. M., et al. J Am Chem Soc 1982, 23, 6424.