ICSM: An order $N$ method for calculating electrostatic interactions added to TINKER

Katherine Baker$^a$, Andrij Baumketner$^{b,c}$, Yuchun Lin$^d$, Shaozhong Deng*,$^a$, Donald Jacobs$^b$, Wei Cai$^a$

$^a$Department of Mathematics and Statistics, University of North Carolina at Charlotte, Charlotte, NC 28223, United States
$^b$Department of Physics and Optical Science, University of North Carolina at Charlotte, Charlotte, NC 28223, United States
$^c$On leave from the Institute for Condensed Matter Physics, NAS of Ukraine, 1 Svientsistsky Str, Lviv, 79011
$^d$Department of Chemical and Biomolecular Engineering, University of California, Berkeley, CA 94720, United States

Abstract

We present an order $N$ method for calculating electrostatic interactions that has been integrated into the molecular dynamics portion of the TINKER Molecular Modeling package. This method, introduced in a previous paper [J. Chem. Phys. 131 (2009) 154103] and termed the Image-Charge Solvation Model (ICSM), is a hybrid electrostatic approach that combines the strengths of both explicit and implicit representations of the solvent. A multiple-image method is used to calculate reaction fields due to the implicit part while the Fast Multipole Method (FMM) is used to calculate the Coulomb interactions for all charges, including the explicit part. The integrated package is validated through test simulations of liquid water. The results are compared with those obtained by the Particle Mesh Ewald (PME) method that is built in the TINKER package. Timing performance of TINKER with the integrated ICSM is benchmarked on bulk water as a function of the size of the system. In particular, timing analysis results show that the ICSM outperforms the PME for sufficiently large systems with the break-even point at around 30,000 particles in the simulated system.

Key words: Method of images, reaction field, electrostatic interaction, TINKER

* Corresponding author. Tel.: +1 704 687 6657; Fax: +1 704 687 6415
Email address: shaodeng@uncc.edu (Shaozhong Deng).
1 Introduction

In Ref. [1], we introduced a new solvation method for calculating electrostatic interactions in computer simulations of biomolecules in aqueous solutions. This method, termed the image-charge solvation model (ICSM), is a hybrid solvation model that combines the strengths of both explicit and implicit solvent representations. In this model, solute molecules are placed in a central spherical cavity filled with explicit water, while the solvent outside the cavity is modeled as a dielectric continuum whose effect on the solute is treated through reaction field corrections. For improved computational efficiency, an accurate and efficient multiple-image charge method is used to compute these reaction field corrections [1–4], and at the same time the adaptive fast multipole method (FMM) [5–8] is employed to calculate the electrostatic force field inside the simulation box, including the direct Coulomb interactions between the explicit particles. When there are $N$ charges inside the cavity, the use of the FMM will result in a speedup from $O(N \log N)$ for the Particle Mesh Ewald (PME) to only $O(N)$ operations. The ICSM was originally tested systematically using an in-house program written specifically for use with bulk water [1], and later it was further tested through simulations of ions solvated in water using the modified in-house program [9]. For both cases, it has been found that, by using an optimal set of model parameters, the ICSM can faithfully reproduce many known properties of the simulated biological systems.

In order to apply the ICSM as a general tool to more complicated molecules such as proteins and viruses, we have integrated it into the TINKER Molecular Modeling package [10–16]. Generally speaking, the TINKER molecular modeling software is a complete and general package for molecular mechanics and dynamics, with some special features for biopolymers. It is a free, open-source software that was written and is maintained by Professor J. Ponder’s group at the Washington University. The package contains different molecular force-field parameter sets including AMBER, CHARMM 19, 22, and 27, among several others [10]. The TINKER package has been extensively tested on multiple platforms. Its programs are written in Fortran and can be easily modified. In our case, we have integrated the ICSM into the molecular dynamics portion of the TINKER package.

The paper is organized as follows. In Section 2, the ICSM and its theoretical background are briefly reviewed. Next in Section 3, details regarding how to install as well as how to run the integrated package are discussed. A brief description of the program structure is also presented there. Then in Section 4 the integrated package is tested through simulations of liquid water and the results are analyzed. Also included in this section are some timing analysis results. Finally, concluding remarks are presented in Section 5.

2 Overview of the ICSM and its theoretical background

The ICSM was described in detail in Ref. [1]. Here we give only a brief overview of the most important features of the model for consistency. The model itself can be schematically
illustrated by Fig. 1. A regular truncated octahedron (TO) Λ is employed as the main simulation box. The central part of the box, marked as Region I, aims to accommodate the solute molecule under study, while the remaining part of the box (including Region II and the area in Region I not occupied by the solute molecule) is filled with solvent. The solute and the solvent molecules inside the main simulation box Λ are to be treated in atomic detail. Region III represents a buffer zone whose purpose is to eliminate or reduce the surface effects that would otherwise be induced by the sharp boundary between explicit and implicit solvents. The solvent molecules in Region III are just periodic images of the solvent particles in Region II, defined as in the usual periodic boundary condition with respect to the TO Λ for long range electrostatic calculation. And the solvent outside the spherical cavity Γ is modeled as a dielectric continuum whose effect on the solute is treated through reaction field corrections. Note that the solute particles in Region I are not periodically imaged and only the solvent particles in Region II could have periodic images in Region III, leading to a most important feature of the ICSM that a solute may be solvated without suffering any artificial electrostatic solute-solute interactions. For this reason, Region I is also called the productive region in the ICSM.

Fig. 1. A schematic illustration of the ICSM. Note that the productive region, Region I, can accommodate a solute molecule of maximum diameter of $d = L(\sqrt{3} - \sqrt{5}/2) - 2\tau$, where $L$ is the size of the cube from which the regular TO simulation box Λ is built and $\tau$ is the thickness of the buffer zone. Particles in Region I are not periodically imaged. Region II contains particles which may have periodic images in Region III. Region III, served as the so-called buffer zone, contains the nearest periodic images of the particles in Region II. The solvent outside the spherical cavity Γ is modeled as a dielectric continuum. Given a source charge $q$ located at position $r_s$ inside this spherical cavity Γ, the reaction field $\Phi_{RF}(r)$ at position $r$ inside the cavity Γ due to the polarization of the implicit solvent can be approximated by the potential created by the image charges, $q'_k, q'_i, i \geq 2$, located at positions $r_k, r_i, i \geq 2$, respectively.
In summary, in the ICSM with the method of images to calculate the aforementioned reaction field corrections, the electrostatic force field calculation involves the following three successive steps: (1) locate periodic images in Region III for those solvent particles in Region II of the main simulation box, (2) find image charges of reaction field for all charges inside the spherical cavity \( \Gamma \), including all real charges inside the main simulation box \( \Lambda \) and all periodic images in Region III, and (3) calculate the electrostatic force inside the main simulation box \( \Lambda \) exerted by all charges, including all real charges inside the main simulation box \( \Lambda \), all periodic images in Region III, and all image charges of reaction fields outside the spherical cavity \( \Gamma \).

Therefore, central to the ICSM is, for a charge inside a spherical cavity creating reaction field, how to find the image charges that are located outside the cavity. To this end, theoretically let us consider a spherical volume \( \Gamma \) of radius \( a \) and dielectric permittivity \( \epsilon_i \) embedded in an infinite solvent of dielectric permittivity \( \epsilon_o \). Assume that a single point charge \( q \) is located at position \( r_s \) inside the spherical cavity. Then it is well known that the total electrostatic potential \( \Phi(r) \) inside the cavity satisfies the following Poisson equation:

\[
\epsilon_i \Delta \Phi(r) = -q \delta(r - r_s),
\]

(1)

where \( \delta(\cdot) \) denotes the Dirac delta function. Moreover, the total potential can be written as \( \Phi(r) = \Phi_S(r) + \Phi_{RF}(r) \) where \( \Phi_S(r) = q/(4\pi \epsilon_i |r - r_s|) \) is the primary field that results from the source charge and \( \Phi_{RF}(r) \) is the reaction field, respectively. Now the problem is how to calculate the reaction field \( \Phi_{RF}(r) \) by the method of images.

It has been found that such a reaction field can be approximated by the potential generated by a single image charge outside the sphere, namely,

\[
\Phi_{RF}(r) \approx \frac{q_k}{4\pi \epsilon_i |r - r_k|},
\]

(2)

where \( q_k = \gamma (a/r_s) q, r_k = (a/r_s)^2 r_s \), and \( \gamma = (\epsilon_i - \epsilon_o)/(\epsilon_i + \epsilon_o) \). Friedman [17] was the first to apply this image charge method in the context of biomolecular solvation problems. For better accuracy, however, the ICSM employs an image method using multiple image charges [2]. In short, the same reaction field can be approximated by the potential produced by a set of image charges located along a ray outside the spherical cavity that extends from the image location \( r_k \) to infinity, as shown in Fig. 1, namely,

\[
\Phi_{RF}(r) \approx \frac{q_k}{4\pi \epsilon_i |r - r_k|} + \sum_{m=2}^{N_i} \frac{q_m'}{4\pi \epsilon_i |r - r_m'|},
\]

(3)

where \( q_k' = q_k + q_1' = (1 + \omega_1 \epsilon_i/2 \epsilon_o) q_k \), and for \( m \geq 2 \),

\[
q_m' = \frac{\epsilon_i (\epsilon_i - \epsilon_o)}{2 \epsilon_o (\epsilon_i + \epsilon_o)} \frac{\omega_m r_m}{a} q, \quad r_m = r_k \left( \frac{2}{1 - s_m} \right)^{1+\epsilon_i/\epsilon_o}.
\]

(4)

Here \( \{s_m, \omega_m\}_{m=1}^{N_i} \) represent the points and the weights of the Gauss-Radau quadrature [18]. Note that when \( N_i = 1 \), we have \( s_1 = -1 \) and \( \omega_1 = 2 \). Therefore, the corresponding single
image charge at \( r_k \), termed the modified Friedman image, has a slightly different magnitude \( q_k' = (1 + \epsilon_i/\epsilon_o)q_k \) from that of the Friedman image charge \( q_k \) in (2).

Another important feature of the ICSM lies in the fact that it can be easily combined with the fast multipole method (FMM) [5,6,19,20], leading to an \( O(N) \) algorithm for calculating electrostatic interactions in biomolecular systems. Moreover, computational efficiency can be further improved by a special local expansion (LE) procedure. Simply speaking, we note that, at each time step, the most time-consuming part is Step (3), namely, calculation of the electrostatic forces exerted on all real charges inside the simulation box \( \Lambda \) by all sources (real charges in the simulation box \( \Lambda \), periodic images in Region III, and image charges of reaction fields outside the cavity \( \Gamma \)). In order to speed-up this force calculation, we first introduce a reference sphere \( S_r \) of radius \( \kappa R_c \) centered at the origin with \( \kappa > 1 \). Then the electrostatic force field within the simulation box \( \Lambda \) due to the charges inside this reference sphere is evaluated by an adaptive FMM [21], while that due to the source charges outside this reference sphere is calculated by the special local expansion procedure whose detail can be found in Ref. [1] and thus are omitted here.

In Ref. [1] the C++ software KIFMM developed by Ying et al. [22] using a kernel-independent adaptive FMM was used. In this work, however, we adopt the Fortran software FMM-Yukawa developed by Huang et al. [7,8] using the new version of the adaptive FMM that uses plane wave expansions to diagonalize the multipole-to-local translations. The program and its full description are available at http://www.fastmultipole.org/.

3 Integration of the ICSM into TINKER

3.1 Implementation of the ICSM in TINKER

As pointed out earlier, the ICSM has been integrated into the molecular dynamics portion of the TINKER package to provide a new way to calculate electrostatic interactions. As described in detail later, the changes needed to the original TINKER source files are minimal. As a matter of fact, the only major change required is to add in the TINKER source file echarge1.f a call to the ICSM subroutine imgmethod() and another subsequent call to another ICSM subroutine fmmle(). In this sense, the (modified) TINKER file echarge1.f could be regarded as the interface between TINKER and the ICSM. In simulations the call for the ICSM is controlled by the keyword imgmethod in the keyword parameter file. It should also be mentioned that a new subroutine echarge1h() is added into echarge1.f for calculating electrostatic interactions by the ICSM without using the FMM. And whether or not to use the FMM in the ICSM is determined by another keyword fmmle in the keyword parameter file.

If the keyword imgmethod is present in the keyword parameter file, the ICSM subroutine imgmethod() is called. The variables unique to the ICSM are first calculated, including
the Gauss-Radau quadrature points and weights $\{s_m, \omega_m\}_{m=1}^{N_i}$. The atoms in each molecule inside the simulation box $\Lambda$ are then checked to see if they are inside Region II. If any part of the molecule is inside Region II, the positions of the periodic images of its atoms are calculated and stored in an image-position array, and the charges of these periodic images are stored in an image-charge array accordingly. Remember that only those periodic images that are indeed located inside the buffer zone are kept and contribute to the reaction field corrections. Next, once all periodic images are found and stored, the locations as well as the magnitudes of image charges are calculated using Eq. (4) and stored in corresponding arrays for all real and periodic image charges inside the spherical cavity $\Gamma$ if $N_i \geq 1$. Lastly, the electrostatic forces exerted on all real “target” charges inside the simulation box $\Lambda$ by all “source” charges, including real charges inside $\Lambda$, periodic images in Region III, and image charges of reaction fields outside the spherical cavity $\Gamma$, are calculated either pairwisely or by the FMM, depending on whether the keyword $\text{fmmle}$ is present in the keyword parameter file. Once this last step is done, the electrostatic forces are then added to the values in TINKER’s force arrays and control is returned to TINKER by the ICSM.

If the keyword $\text{fmmle}$ is present in the keyword parameter file, the ICSM subroutine $\text{fmmle()}$ is called to calculate the above electrostatic forces by the FMM together with the local expansion procedure described in Section 2. First, all “source” charges are split into two groups based on where they are located. If a charge is located inside the cut-off reference sphere $S_r$, it is put in the “inside” group; otherwise it is put in the “outside” group. Next, the electrostatic forces exerted on all “targets” inside the simulation box $\Lambda$ by those “source” charges from the “inside” group are evaluated using the FMM, while the forces exerted on the same “targets” by those “source” charges from the “outside” group are calculated using the local expansion (LE). The forces calculated by the FMM and the local expansion are then added together to finally obtain the total electrostatic forces.

3.2 Installation and overview of individual components

To install the integrated TINKER and ICSM package, one needs first to download the TINKER package from the website http://dasher.wustl.edu/TINKER/, and extract the package to the directory that it will be run from. Note that all source files of the TINKER package are located in the source subdirectory. Then, one needs to download the ICSM package from http://www.math.uncc.edu/~wcai/TINKER-icsm. After extraction of the zip file, the ICSM package contains several modified TINKER files and four top-level modules: ICSM, FMM, LE, and EXAMPLES. Next, one needs to copy all of the files and directories of the ICSM package into TINKER’s source subdirectory. Finally, one needs to modify the make file $\text{makefile}$, located in the source directory, to include the run path and computer specifications for the system the integrated package will be compiled on. For detailed information on the installation see the manual included with TINKER [10].

As mentioned, several TINKER files were edited. More specifically, the following TINKER files have been modified in order to integrate the ICSM into TINKER.
The subdirectory ICSM contains the following subroutines for the ICSM.

(a) imgmethod.f - the main call for the ICSM.
(b) fmmle.f - use the FMM and the local expansion to calculate the electrostatic force field in the TO box.
(c) kimchchg.f - initialize variables singular to the ICSM. Also calculate the Gauss-Radau quadrature points and weights \(\{s_m, \omega_m\}_{m=1}^{N_i}\).
(d) finding.f - find image charges for charges inside the spherical cavity \(\Gamma\) as shown in Fig. 1.
(e) imgchgs.i - define variables unique to the ICSM

The subdirectory LE contains the files needed for the special local expansion as pointed out in Section 2. On the other hand, the subdirectory FMM contains the FMM-Yukawa source files downloaded from the website http://www.fastmultipole.org/ but with a few files being modified and renamed. In particular, the new file ICSMFMMdriver.f serves as the interface between the ICSM and the FMM. The new file ICSMFMMadaplap.f, modified based on and renamed from fmmadaplap.f, now calculates both electrostatic potentials and forces. In addition, parm-alap.h is renamed as fmm.i for naming consistency.

The last subdirectory EXAMPLES contains an input file water_30A.xyz and a keyword parameter file water_30A.key needed for a test simulation of liquid water for a 30 Å simulation box, using the integrated TINKER and ICSM package with one image charge and a buffer layer of thickness \(\tau = 6\) Å.

### 3.3 Execution and modification of the keyword parameter file

After successful compilation of the package, the command line to run TINKER with the integrated ICSM module is the same as that to start TINKER; see Section 4.1 for a specific example. In order to invoke the ICSM to calculate electrostatic interactions, however, the following two keyword lines need be added to the keyword parameter file:

```
imgmethod  <buffer size tau>  <number of images N_i>
fmmle      <LE order p>      <factor kappa>
```

For both keywords `imgmethod` and `fmmle`, the meaning of their two values are self-explanatory. Indeed, `<buffer size tau>` specifies the thickness \(\tau\) of the buffer zone in Angstrom as shown in Fig. 1, `<number of images N_i>` specifies the number of images \(N_i\) used to calculate image charges as given by Eq. (3), `<LE order p>` specifies the local expansion order
as defined in Eq. (13) of Ref. [1], and finally <factor kappa> specifies the parameter \( \kappa \) used to define the reference sphere for the local expansion as shown in Fig. 5 of Ref. [1], respectively.

For instance, the following two keyword lines in the keyword parameter file

\[
\text{imgmethod} \quad 6.0 \quad 1 \\
\text{fmmle} \quad 10 \quad 2.0
\]

would make TINKER call the ICSM module to calculate the electrostatic force field in the main simulation cell with these model parameters: \( \tau = 6 \) Å, \( N_i = 1 \), \( p = 10 \), and \( \kappa = 2.0 \), respectively.

4 Test simulations and discussions

In order to validate the ICSM integrated into the TINKER molecular modeling package, we apply TINKER 6.0 with the integrated ICSM to liquid water simulation. We run the same benchmarks on the physical properties of bulk water as in our previous paper [1], including the density, the structural oxygen-oxygen radial distribution function, the self-diffusion coefficient, and the dielectric constant. We compare the results to the PME calculations included in the TINKER package and to the published results obtained by the original in-house ICSM software [1]. That is, unless otherwise specified, for each test, we run three different kinds of simulations: one by TINKER with the built-in PME using neighbor lists, one by TINKER with the integrated ICSM using one image charge \( (N_i = 1) \), and one by the original in-house ICSM program used in Ref. [1] also with one image charge. And for convenience, the corresponding results are marked as “TINKER - PME”, “TINKER - ICSM”, and “ICSM”, respectively.

4.1 Simulation details

We use the TIP3P [23] all-atom model to characterize properties of bulk water. The implementation details of the in-house ICSM program can be found in Ref. [1]. In all three approaches employed, positions and velocities of particles are calculated using the velocity Verlet algorithm, coupled with the Nosé-Hoover thermostat. For each test, three different simulation boxes with \( L = 30, 45, \) and \( 60 \) Å are considered, the number of atoms contained in these boxes being 1224, 4224, and 10275, respectively. The corresponding initial coordinates are the identical input files used in Ref. [1], which were generated from an equilibrium 200 ps simulation in NPT ensemble, performed by the GROMACS program package. Since it was found in Ref. [1] that the thickness of the buffer zone had to be at least 6 Å in order to yield the uniform density of the simulated bulk water inside the TO \( \Lambda \), we use a six Angstrom buffer zone for all tests reported in this work, namely, \( \tau = 6 \) Å. In addition, each dynamics
simulation was 1.1 ns long with the first 0.1 ns used for equilibration time, while the integration time step was set to 2 fs. The simulations were performed under constant temperature conditions at $T = 300$ K. The trajectories were recorded at every 0.2 ps for subsequent analysis whose details can be found in Ref. [1] too. In all simulations, for the FMM-Yukawa software, the maximum number of particles in a leaf box of the adaptive oct-tree structure was set to 80 ($n_{box}=80$). The number of terms in the multipole and local expansions and that in the plane wave expansion were both set to 9 for three-digit accuracy ($n_{terms}=9$, $n_{lambds}=9$).

For the proposed TIP3P liquid water tests, a typical command line for running TINKER with the integrated ICSM would be:

```
Dynamic water_30A.xyz 550000 2.0 0.2 2 300
```

where Dynamic is the TINKER executable, water_30A.xyz represents the input file for the 30 Å simulation box, 550,000 is the number of total time steps, 2.0 is the time step size in femtoseconds, 0.2 indicates the trajectory is recorded after every 0.2 picoseconds, 2 indicates to use the NVT ensemble, and 300 is the temperature, respectively. And a corresponding typical keyword parameter file water_30A.key could be:

```
parameters none
octahedron
randomseed 123456789
thermostat Nose-Hoover
vdwtype LENNARD-JONES
vdw-cutoff 10.0
radiusrule geometric
radiustype SIGMA
radiussize DIAMETER
epsilonrule geometric
dielectric 80.0
a-axis 30.0000
b-axis 30.0000
c-axis 30.0000
integrate verlet
rattle water
tau-temperature 0.1

# ICSM Parameters
imgmethod 6.0 1
fmmle 10 2.0

# Water Parameters
atom 1 O "O Water (TIP3P)" 8 15.9994 2
atom 2 H "H Water (TIP3P)" 1 1.008 1
```
4.2 Results and discussions

To validate TINKER with the integrated ICSM, as in Ref. [1], first we examine the local particle density across the simulation box as a measure of the homogeneity of the simulated bulk water. More specifically, we compute the relative density of oxygen atoms along the diagonal of the TO simulation box, and the results are plotted in Fig. 2. As can be seen, for all three box sizes, the relative densities obtained by the three different approaches are in good agreement with each other, all showing a uniform density distribution with some statistical variations which are quantitatively summarized in Table 1.

Table 1
Standard deviations of relative densities along the diagonal of TO simulation boxes.

<table>
<thead>
<tr>
<th></th>
<th>TINKER - ICSM</th>
<th>TINKER - PME</th>
<th>ICSM</th>
</tr>
</thead>
<tbody>
<tr>
<td>30 Å</td>
<td>0.004</td>
<td>0.004</td>
<td>0.002</td>
</tr>
<tr>
<td>45 Å</td>
<td>0.004</td>
<td>0.003</td>
<td>0.003</td>
</tr>
<tr>
<td>60 Å</td>
<td>0.003</td>
<td>0.004</td>
<td>0.004</td>
</tr>
</tbody>
</table>

Next, we examine the structure of the simulated bulk water. Again, we evaluate the structure using \( g_{oo}(r) \), the oxygen-oxygen radial distribution function (RDF), over the entire simulation box, and the results are plotted in Figs. 3-5. Recall that the most important features of \( g_{oo}(r) \) are the locations as well as the magnitudes of the first three density peaks and the first two density minima. As can be seen, for all three box sizes, the RDFs obtained by the three different approaches are in excellent agreement with each other, and a closer look at \( g_{oo}(r) \) using a higher resolution, shown in the insets of Figs. 3-5, reveals some noticeable but yet slight difference only for the first two maxima and the first minimum. For example, the first density maxima all lie at 2.76 Å for all box sizes for all three models, while their values are shown in Table 2. On the other hand, the first density minima are all located around 3.56 Å and their values are given in Table 3.

Then, we examine the dynamical properties of the simulated water. More precisely, we once again choose to evaluate the self-diffusion coefficient \( D \), a transport coefficient characterizing how quickly equilibrium is established in particle density following a small perturbation. The self-diffusion coefficients obtained by the three different approaches are recorded in Table 4. As it shows, the self-diffusion coefficient obtained by TINKER with the integrated ICSM
is consistent, within statistical errors, with both the values obtained by the PME included in TINKER and the original findings from Ref. [1] which were obtained using the original in-house ICSM software. Also note that the standard deviation decreases as the box size
Fig. 3. Computed oxygen-oxygen RDFs for a 30 Å TO simulation box. The inset shows a closeup of the first two density minima.

Fig. 4. Same as Fig. 3 but for a 45 Å simulation box.

Last, to validate TINKER with the integrated ICSM, we examine the dielectric properties of the simulated bulk water by computing its dielectric constant \( \varepsilon \). In particular, Formula (A3) of Ref. [1] is used to calculate the dielectric constant \( \varepsilon(R) \) as a function of the sample radius \( R \) which can vary between 0 and \( (\sqrt{3}/4)L \) for a TO simulation box with size \( L \); the readers are referred to Ref. [1] for the meaning of this radius \( R \) and the detailed derivation of this formula. As stated in Ref. [1], since \( \varepsilon \) is a macroscopic property of a liquid material, it is physically meaningful only for a relatively large volume of the material objects. Once
Fig. 5. Same as Fig. 3 but for a 60 Å simulation box.

Table 4
Computed self-diffusion coefficients $D$ with standard deviations (unit: $10^{-9}$ m$^2$s$^{-1}$).

<table>
<thead>
<tr>
<th></th>
<th>TINKER - ICSM</th>
<th>TINKER - PME</th>
<th>ICSM</th>
</tr>
</thead>
<tbody>
<tr>
<td>30 Å</td>
<td>$6.15(\pm0.12)$</td>
<td>$6.16(\pm0.12)$</td>
<td>$6.28(\pm0.03)$</td>
</tr>
<tr>
<td>45 Å</td>
<td>$6.16(\pm0.04)$</td>
<td>$6.24(\pm0.07)$</td>
<td>$6.19(\pm0.02)$</td>
</tr>
<tr>
<td>60 Å</td>
<td>$6.03(\pm0.03)$</td>
<td>$6.06(\pm0.05)$</td>
<td>$6.02(\pm0.01)$</td>
</tr>
</tbody>
</table>

a crossover from a finite-size to macroscopic dimensions is achieved, the dielectric constant should be largely independent of the geometry of the investigated object. In the case of $\varepsilon(R)$, a plateau should be expected in its graph starting at some radius. Figure 6 shows the computed dielectric constant $\varepsilon(R)$ obtained, respectively, by TINKER with the integrated ICSM and the original in-house ICSM software. Again, good agreement between these two simulation results can be observed.

4.3 Timing analysis

Finally, to test the performance of the ICSM integrated into TINKER, we compare the timing required to run TINKER with the integrated ICSM using the FMM against the timing required to run TINKER with the built-in PME. To this end, in addition to the three simulation boxes used above, larger simulation boxes with $L = 80$ Å, 100 Å, 120 Å, and 140 Å are also considered. For each case, the corresponding simulation is run for 1000 time steps and the time required to run, calculated using the TIME function built into the UNIX, is plotted in Fig. 7 as a function of the number of atoms in the system. As can be seen, for relatively small systems, TINKER with the built-in PME runs faster than TINKER with the integrated ICSM, while for sufficiently large systems the latter clearly outperforms the
Fig. 6. Dielectric constant $\varepsilon(R)$ as a function of the spherical sample radius $R$ computed using Formula (A3) of Ref. [1] for three different TO simulation boxes with $L = 30, 45,$ and $60$ Å, respectively. For sufficiently large $R$, $\varepsilon(R)$ is seen to level off, and the plateau value in $\varepsilon(R)$ is identified as the static dielectric constant of the material. Note that, as in Ref. [1], this static dielectric constant is seen to grow from $\sim 65$ at $L = 30$ Å to $\sim 80$ at $L = 60$ Å.

former. The break-even point is at around 30,000 atoms.

Fig. 7. Results of timing tests using TINKER with the integrated ICSM and TINKER with the built-in PME, respectively. Test runs were carried out for 1000 time steps and the time used was calculated using the TIME function built into the UNIX. TINKER with the integrated ICSM starts to outperform TINKER with the built-in PME at approximately 30,000 atoms.
5 Conclusions

In all tests using TINKER with the integrated ICSM module, the physical properties of the simulated water were found to be consistent with those reported in Ref. [1] which were obtained by the original in-house ICSM code. In that work, it was found that the ICSM could not compete with the PME in speed, although the theoretical scaling ($O(N)$ versus $O(N \log N)$) would suggest that the ICSM should be faster for very large systems. However, that comparison was made for relatively small systems and across multiple simulation packages. In this work, we showed that within one package, the ICSM outperforms the PME for large systems, as implied by the asymptotic complexity of these two methods. With the new implementation of the FMM, and with respect to the PME calculations available in TINKER, the crossover point has been substantially reduced. In particular, for the implementation presented here, the ICSM module in the TINKER will run faster for systems containing a large number of atoms of about 30,000 charges, or a simulation box of 80 Å across. We note that boxes of this size have become common in computational studies of biological molecules [24]. Moreover, continuing development of the FMM may further improve the ICSM’s performance in the future.

Currently we are testing the integrated TINKER package on a system containing an alanine dipeptide as this peptide has been widely studied in the literature and its properties are readily available for comparison purposes. We are also testing it for systems containing charged solvents. Any significant findings will be reported in future publications.

Acknowledgments

The support of the National Institutes of Health through Grant No. 1R01GM083600-04 is gratefully acknowledged.

References


