JCTC Journal of Chemical Theory and Computation

A Graphics Processing Unit Implementation of Coulomb Interaction in Molecular Dynamics

Prateek K. Jha,[†] Rastko Sknepnek,[‡] Guillermo Iván Guerrero-García,[‡] and Monica Olvera de la Cruz*[,‡,†,§]

Department of Chemical and Biological Engineering, Department of Materials Science and Engineering, and Department of Chemistry, Northwestern University, Evanston Illinois 60201

Received June 29, 2010

Abstract: We report a GPU implementation in HOOMD Blue of long-range electrostatic interactions based on the orientation-averaged Ewald sum scheme, introduced by Yakub and Ronchi (*J. Chem. Phys.* **2003**, *119*, 11556). The performance of the method is compared to an optimized CPU version of the traditional Ewald sum available in LAMMPS, in the molecular dynamics of electrolytes. Our GPU implementation is significantly faster than the CPU implementation of the Ewald method for small to a sizable number of particles ($\sim 10^5$). Thermodynamic and structural properties of monovalent and divalent hydrated salts in the bulk are calculated for a wide range of ionic concentrations. An excellent agreement between the two methods was found at the level of electrostatic energy, heat capacity, radial distribution functions, and integrated charge of the electrolytes.

1. Introduction

The introduction of highly optimized, specialized hardware, that is, the graphics processing unit (GPU), has allowed for rendering high definition, nearly photorealistic 3D scenes in real time on a standard personal computer. Ever increasing market demand for fast and realistic graphics has driven a rapid development of inexpensive GPU devices, with a doubling of computational power every 12 months. A modern GPU is a highly parallel, multithreaded device with floating point speed close to 1 TFLOPS and a bandwidth in the 100 GB/s range. The GPU derives its superb computational power from its design, specialized in performing intensive computations on large sets of data in parallel. In recent years, the GPU hardware has become available to nongraphical applications through the advent of general-purpose programmability of the device. Problems that can take advantage of the high-throughput parallel computations can greatly benefit from the GPU architecture and easily reach a 100-fold increase in performance over equivalent implementation on a CPU.[1,2] A notable example is molecular dynamics (MD) with reports of GPU implementations achieving speed-ups in excess of 100 times compared to the standard MD codes. However, the high level of data parallelization comes at the expense of limited caching and flow control compared to the CPU.[1,3−5] Thus, in most cases, it is not possible to simply recompile existing CPU codes on the GPU, and it is often required to substantially redesign existing methods and to develop new algorithms.

In order to reduce finite-size effects, periodic boundary conditions are imposed in a typical MD simulation. That is, if a particle crosses the simulation box boundary, it immediately reappears from the opposite side. Equivalently, this can be seen as if the system has been replicated infinitely many times in each direction and each particle has infinitely many images. In principle, one has to include contributions from all the images of all the particles in order to compute the total energy of the system. In practice, this is seldom necessary, and it is sufficient to cut off interactions at a certain distance r_c and evaluate only the interaction between particles that are within r_c from each other. Formally, if we assume that a system containing N particles is homogeneous

* To whom correspondence should be addressed. E-mail: m-olvera@northwestern.edu.

[†] Department of Chemical and Biological Engineering.

[‡] Department of Materials Science and Engineering.

[§] Department of Chemistry.

10.1021/ct100365c
Published on Web 09/15/2010

and isotropic with density ρ, then the error introduced in the total energy by truncating the potential at r_c is[6]

$$U_{\text{error}} = \frac{N\rho}{2}\int_{r_c}^{\infty} u(r) 4\pi r^2\, \mathrm{d}r \tag{1}$$

where $u(r)$ is the true, nontruncated potential and we explicitly used the fact that the system is isotropic to write the integral in spherical coordinates. If $u(r) \propto r^{-\alpha}$ with $\alpha > 3$, the correction $U_{\text{tail}} \propto r_c^{3-\alpha}$ can be made arbitrarily small by increasing the cutoff distance r_c. However, if $u(r)$ falls off slower than r^{-3}, any such cutoff will result in a divergent correction to the total energy. Most intermolecular potentials fall off faster than r^{-3} and can be considered short-range. Practically, we can safely truncate them at a suitable cutoff distance, typically chosen to be less than half the diameter of the simulation box, an approximation commonly known as the nearest image convention. Important exceptions are Coulomb and dipolar interaction potentials that fall off with distance as r^{-1} and r^{-3}, respectively. These electrostatic potentials describe interaction between point charges and dipoles ubiquitous in nature, most notably in biological systems. It has been shown in the past that a truncation of long-range interactions can lead to artifacts like the formation of nonphysical structures in ionic liquids.[7,8]

A proper treatment of the electrostatic interaction is a necessary feature in a general-purpose molecular dynamics code. The Ewald summation method[9] (henceforth referred to as the ES method) and its derivatives are most commonly used, though several alternatives exist.[10,11] The trick behind the ES method is to separate the electrostatic energy into a short-range and a long-range contribution, with the long-range contribution computed efficiently in reciprocal space. The numerical effort needed to calculate the total electrostatic energy using ES method scales as $O(N^{3/2})$ with the system size.[6] The computational expense can be reduced to $O(N \log N)$ by interpolating charges to a lattice and using fast Fourier transform to compute the reciprocal space sum. This is the basis of the smoothed particle mesh Ewald (SPME) method,[12] used in several MD packages. However, an implementation of these methods on the GPU is a challenging task since the long-range contribution has to be treated carefully in order to harvest the full benefit of the massive data parallelization. A successful implementation of the SPME method on the GPU has been recently reported.[2] Also, an alternative algorithm based on the multipole expansion has been proposed.[13] Unfortunately, although very efficient, both schemes are complex, and full apprehension of these algorithms requires intimate knowledge of the GPU architecture.

In this paper, we take a different approach and present results of the GPU implementation of a treatment of the electrostatic interaction recently introduced by Yakub and Ronchi (henceforth referred to as the YR method).[14−16] This approximation is particularly suitable for isotropic ionic fluids. The expressions for the electrostatic energy and the interparticle force are remarkably simple and can be easily implemented into an existing MD code.

2. Methodology

The total electrostatic energy of a system of N charges placed in a cubic box of length L with periodic boundary conditions is

$$E_{\text{el}} = \frac{1}{2}\frac{1}{4\pi\varepsilon_0\varepsilon_{\text{r}}}\sum_{\vec{n}}\sum_{i,j=1}^{N}{}' \frac{q_i q_j}{|\vec{r}_i - \vec{r}_j + \vec{n}L|} \tag{2}$$

where ε_0 is the vacuum permittivity, ε_{r} is the relative static permittivity, $\vec{r}_i$ ($\vec{r}_j$) is the position of charge q_i (q_j). $\vec{n} = (n_x, n_y, n_z)$, where n_x, n_y, and n_z are arbitrary integers, counts all periodic images. The prime in the second sum indicates that the $i = j$ term should be omitted for $\vec{n} = 0$, and the 1/2 prefactor accounts for double counting. The sum in eq 2 is only conditionally convergent and cannot be directly used in simulations. The idea behind the Ewald method is to separate eq 2 into short- and long-range parts, each expressed as a rapidly converging sum. The total electrostatic energy can be written as the sum of these two contributions plus a constant self-energy contribution[6]

$$E_{\text{el}} = E_{\text{short}} + E_{\text{long}} + E_{\text{self}} \tag{3}$$

The short-range contribution is calculated in the real space as

$$E_{\text{short}} = \frac{1}{2}\frac{1}{4\pi\varepsilon_0\varepsilon_{\text{r}}}\sum_{\substack{i,j=1\\ i\neq j}}^{N} \frac{q_i q_j}{|\vec{r}_i - \vec{r}_j|}\operatorname{erfc}(\sqrt{\alpha}|\vec{r}_i - \vec{r}_j|) \tag{4}$$

where $\operatorname{erfc}(x) = 1 - \operatorname{erf}(x)$ is the complementary error function and α is the Ewald parameter. The long-range sum (E_{long}) is evaluated in the reciprocal space as

$$E_{\text{long}} = \frac{1}{2L^3}\frac{1}{\varepsilon_0\varepsilon_{\text{r}}}\sum_{\vec{k}\neq 0}\frac{\exp(-k^2/4\alpha)}{k^2}|S(\vec{k})| \tag{5}$$

where $\vec{k} = (2\pi)/(L)\vec{n}$ are the reciprocal lattice vectors, and

$$S(\vec{k}) = \sum_i q_i\, e^{i\vec{k}\cdot\vec{r}_i} \tag{6}$$

is the charge structure factor. In addition, a self-energy term

$$E_{\text{self}} = -\frac{1}{4\pi\varepsilon_0\varepsilon_{\text{r}}}\sqrt{\frac{\alpha}{\pi}}\sum_{i=1}^{N} q_i^2 \tag{7}$$

arises, and it has to be added to the sum of short- and long-range terms. Note that the Ewald parameter α is related to the position of splitting between short- and long-range parts in the Ewald sum. In a simulation, α has to be carefully tuned to ensure the most optimal performance.

We briefly summarize the YR method. A detailed derivation is presented in the original paper.[14] In an ordered phase, the crystal lattice sets a natural direction for the simulation box. This is not the case in fluids where all directions are equivalent; that is, there is no preferred orientation of the simulation box. Thus, Yakub and Ronchi proposed to average eq 3 over all directions of the reciprocal lattice vector $\vec{k}$; that is, $E_{\text{el}} = \langle E_{\text{el}}\rangle$, where

$$\langle ... \rangle = \frac{1}{4\pi}\int_0^{\pi} \mathrm{d}\theta \sin\theta \int_0^{2\pi} \mathrm{d}\phi ... \tag{8}$$

is the average over the polar angle θ and the azimuthal angle ϕ. Note that the averaging is performed over all possible orientations of $\vec{k}$ while $\vec{r}_i$ is kept fixed; thus it is only necessary to average the E_{long} term since E_{short} and E_{self} terms have no θ or ϕ dependence. If we impose electroneutrality, $\sum_{i=1}^{N} q_i = 0$, the expression for the angularly averaged total electrostatic energy takes a surprisingly simple form[14]

$$E_{\mathrm{el}} = \frac{1}{2}\sum_{\substack{i,j\\ i\neq j}} \phi^{(C)}(r_{ij}) \tag{9}$$

with the pair potential

$$\phi^{(C)}(r_{ij}) = \frac{q_i q_j}{4\pi\varepsilon_0\varepsilon_{\mathrm{r}} r_{ij}}\left[1 + \frac{1}{2}\left(\frac{r_{ij}}{r_m}\right)^3\right] \tag{10}$$

where $r_{ij} = |\vec{r}_i - \vec{r}_j|$, and $r_m = (3/4\pi)^{1/3}L$ is the radius of a sphere of volume L^3. Note that unlike the adjustable parameter α in the Ewald method, r_m is fixed by size of the simulation box.[14] One counts only the interactions between particles at distances $0 \leq r_{ij} \leq r_m$. A drawback of eq 10 is that the pair potential $\phi^{(C)}(r_{ij})$ is nonzero at its minimum at $r_{ij} = r_m$, $\phi^{(C)}(r_m) = 3q_iq_j/8\pi\varepsilon_0\varepsilon_{\mathrm{r}}r_m$, that results in a jump in the cutoff scheme. It is therefore convenient to shift this potential by $-\phi^{(C)}(r_m)$ to bring the boundary values to zero. That is, a modified interionic potential is defined

$$\tilde{\phi}^{(C)}(r_{ij}) = \begin{cases} \dfrac{q_iq_j}{4\pi\varepsilon_0\varepsilon_{\mathrm{r}}r_{ij}}\left[1 + \dfrac{1}{2}\left(\dfrac{r_{ij}}{r_m}\right)\left(\left(\dfrac{r_{ij}}{r_m}\right)^2 - 3\right)\right] & r_{ij} < r_m \\ 0 & r_{ij} \geq r_m \end{cases} \tag{11}$$

such that $\tilde{\phi}^{(C)}(r_{i\,j}) \to 0$ as $r_{ij} \to r_m$. By using the electroneutrality condition, sum $\sum_{i=1}^{N} q_i = 0$, once again, the expression for the total electrostatic energy (eq 9) in terms of the modified interionic potential can be written as

$$E_{\mathrm{el}} = -\sum_{i=1}^{N}\frac{3q_i^2}{16\pi\varepsilon_0\varepsilon_{\mathrm{r}}r_m} + \frac{1}{2}\sum_{\substack{i,j\\ i\neq j}}\tilde{\phi}^{(C)}(r_{ij}) \tag{12}$$

However, the expression for the interparticle force $\vec{f}_{ij} = -\nabla\phi_{ij}^{(C)} = -\nabla\tilde{\phi}_{ij}^{(C)}$ is not affected. Equation 11 is the effective electrostatic pair potential $\tilde{\phi}^{(C)}$ associated with the Coulombic system and is the central result of the YR method. Thus, for an isotropic electroneutral system subject to periodic boundary conditions, long-range effects of the electrostatic interaction can be expressed in terms of a finite range potential.

It is worth mentioning that the cutoff radius $r_m = (3/4\pi)^{1/3}L \approx 0.62L$ is larger than $L/2$. This fact has to be accounted for when calculating the electrostatic potential

$$\tilde{\Phi}^{(C)}(\vec{r}_i) = \sum_{j\neq i}\tilde{\phi}^{(C)}(r_{ij}) \tag{13}$$

on a charge q_i located at $\vec{r}_i$. Namely, the electrostatic contribution of some charges has to be included twice, that is, as the original charges and as their "phantom" images.[14] These ions are contained in the shaded region in Figure 1 and are obtained by the overlap of a sphere of radius r_m centered on an ion and six spheres of the same radius centered on the images of the ion. To illustrate the calculation of the effective pair potential in the YR method, we show four particles in the xy plane ($z = 0$) with coordinates $P(0, 0, 0)$, $A(x_A, y_A, 0)$, $B(x_B, y_B, 0)$, and $C(x_C, y_C, 0)$. Particle A is in the nonoverlap region of the sphere centered at P, and thus its interaction with P needs to be counted once. That is, the effective pair potential between particles P and A is

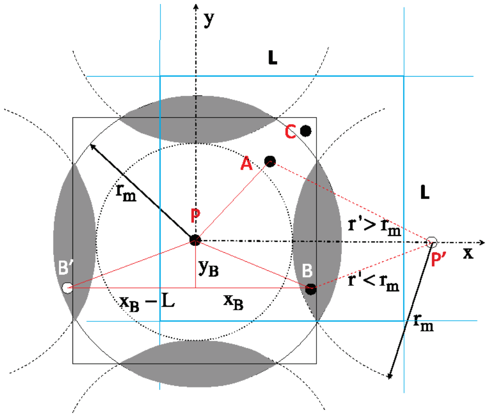

Figure 1. Main unit cell and spheres of radius r_m centered on an ion P and its nearest images P'. Shaded regions indicate the overlap of these spheres. The effective interaction of ion B with ion P is counted twice, both as the original ion and its "phantom" image B'. Ion A is counted only once since it is in a nonoverlap region, and the effective interaction between ions P and C is zero. The blue line indicates the boundaries of the cubic simulation box. See section 3 for description of the dotted circle.

$$\tilde{\phi}_{PA}^{(C)} = \tilde{\phi}^{(C)}\left(\sqrt{x_A^2 + y_A^2}\right) \tag{14}$$

Particle B is in the overlap region of the sphere centered at P, and thus its interaction with P needs to be counted twice, both with particle B and its "phantom" image B'. The effective pair potential between P and B is

$$\tilde{\phi}_{PB}^{(C)} = \tilde{\phi}^{(C)}\left(\sqrt{x_B^2 + y_B^2}\right) + \tilde{\phi}^{(C)}\left(\sqrt{(x_B - L)^2 + y_B^2}\right) \tag{15}$$

Since particle C is outside the sphere centered at P, it needs not be counted, and the effective pair potential between P and C, $\tilde{\phi}_{PC}$, is zero.

3. Implementation Details

A direct consequence of the cutoff radius $r_m = (3/4\pi)^{1/3}L$ being larger than $L/2$ is that each ion has more than N neighbors, rendering the use of a neighbor list impractical, from the point of view of both the memory required to store it and the overhead to update it. Instead, one simply loops over all ions and decides which ones contribute once, which twice, and which do not contribute at all to the sum in eq 13. This implies that when implementing the YR method into the HOOMD Blue package, it is not possible to use the sophisticated EvaluatorPair class template specifically designed for the ease of implementing additional short-range potentials. Instead, we implemented a specialization of the

```
$i \leftarrow blockIdx.x \times blockDim.x + threadIdx.x$
$\phi_i \leftarrow 0, \vec{f}_i \leftarrow 0$
for $j = 1, N_{ions}$ do
    if $i \neq j$ then
        $r_{ij} \leftarrow |\vec{r}_i - \vec{r}_j|$
        if $r_{ij} \leq r_m$ then
            Compute potential $\tilde{\phi}^{(C)}(r_{ij})$ using Eq. (11)
            Compute force $\vec{f}_{ij} = -\nabla\tilde{\phi}_{ij}^{(C)}(r_{ij})$
            $\phi_i \leftarrow \phi_i + \tilde{\phi}^{(C)}(r_{ij}), \vec{f}_i \leftarrow \vec{f}_i + \vec{f}_{ij}$
            if $r_{ij} > L - r_m$ then
                for $\vec{e} = (-1,-1,-1), (-1,-1,0), (-1,-1,1), \ldots, (1,1,1)$ do
                    $\tilde{\vec{r}}_{ij} \leftarrow \vec{r}_i - \vec{r}_j + L\vec{e}$
                    if $\tilde{\vec{r}}_{ij}$ is inside the shaded regions in Figure 1 then
                        Compute potential $\tilde{\phi}^{(C)}\left(\left|\tilde{\vec{r}}_{ij}\right|\right)$ using Eq. (11)
                        Compute force $\tilde{\vec{f}}_{ij} = -\nabla\tilde{\phi}_{ij}^{(C)}\left(\left|\tilde{\vec{r}}_{ij}\right|\right)$
                        $\phi_i \leftarrow \phi_i + \tilde{\phi}^{(C)}\left(\left|\tilde{\vec{r}}_{ij}\right|\right), \vec{f}_i \leftarrow \vec{f}_i + \tilde{\vec{f}}_{ij}$
                    end if
                end for
            end if
        end if
    end if
end for
```

Figure 2. Pseudocode for computing pair Coulomb interactions in the YR method. *blockIdx* and *threadIdx* are standard CUDA structures that contain information about the current execution block and thread, respectively.

PotentialPair template with a custom EvaluatorPairCoulomb class designed to avoid costly use of the HOOMD's neighbor list system.

The CUDA kernel for computing the pair Coulomb interaction in the YR approximation is described in Figure 2. Each thread handles one ion i of the main cell, and one loops over all ions j different from i. If the interionic distance $r_{ij} \leq L - r_m$, that is, if both ions are inside the dotted circle in Figure 1, their contribution to the Coulomb energy is counted once. On the other hand, if $L - r_m < r_{ij} \leq r_m$, one needs to include the contribution of the image ion $\tilde{\vec{r}} = \vec{r}_i + L\vec{e}$ as well, if $\tilde{\vec{r}}_{ij} = \tilde{\vec{r}}_i - \vec{r}_j$ is inside one of the shaded regions in Figure 1. $\vec{e}$ is one of the vectors $(-1, -1, -1)$, $(-1, -1, 0)$, $(-1, -1, 1)$, ..., $(1, 1, 1)$, excluding $(0, 0, 0)$. Finally, if $r_{ij} > r_m$, the ion pair (i, j) is ignored. Since the order in which the contributions from different ions are added to the force and potential sums is irrelevant, a fully coalesced memory read is trivially achievable.

In order to compare the performance and accuracy of the YR method against the ES method in electrolyte systems, we performed MD simulations of hydrated monovalent and divalent electrolytes, with valence $z_+ = -z_- = 1$ and $z_+ = -z_- = 2$, respectively. We use the restricted primitive model (RPM), where an ion is modeled as a hard sphere with a point charge embedded in its center immersed in a continuum dielectric medium. The excluded volume of ions is modeled by the repulsive part of the shifted Lennard-Jones (LJ) potential[17,18]

$$U_{\mathrm{LJ}}(r) = \begin{cases} 4\varepsilon_{\mathrm{LJ}}\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right] + \varepsilon_{\mathrm{LJ}} & r < 2^{1/6}\sigma \\ 0 & r \geq 2^{1/6}\sigma \end{cases} \tag{16}$$

where σ is the diameter of bulk hydrated ions, taken as 6.6 Å and 8.25 Å for monovalent and divalent ions respectively.[19,20] $\varepsilon_{\mathrm{LJ}} = 1k_{\mathrm{B}}T$ is the LJ interaction strength, where k_{B} is the Boltzmann constant and T is the temperature. RPM has been quite successful in the prediction of thermodynamic properties of bulk ionic solutions, and in describing several interesting phenomena associated with charged colloidal systems—such as charge inversion and charge reversal.[21,22] In charge inversion, co-ions and counterions switch their roles near an electrified surface, and in charge reversal, the native surface charge of a colloid is overcompensated by counterions. These effects are due to the ion-size correlations, treated in a coarse-grained description by associating an excluded volume to the hydrated ions.

MD simulations were performed in an NVT ensemble at a reduced temperature $T^* = k_{\mathrm{B}}T/\varepsilon = 1$ with a time step of 0.005τ, where $\varepsilon = 1k_{\mathrm{B}}T$ and $\tau = (m\sigma^2/\varepsilon)^{1/2}$ are the reduced LJ units[23] of energy and time, respectively, and m is the mass of ions, set to unity. The relative static permittivity of the solvent is $\varepsilon_{\mathrm{r}} = 78.4$, corresponding to an aqueous solution. The average electrostatic energy per ion is defined as

$$E^* = \frac{\langle E_{\mathrm{el}}\rangle}{Nk_{\mathrm{B}}T} \tag{17}$$

where E_{el} is defined in eq 12 and $\langle ... \rangle$ stands for the time average. The heat capacity per ion is defined as

$$C^* = \frac{C_v}{Nk_{\mathrm{B}}} = \frac{\langle E_{\mathrm{el}}^2\rangle - \langle E_{\mathrm{el}}\rangle^2}{N(k_{\mathrm{B}}T)^2} \tag{18}$$

where C_v is the heat capacity in real units. These averages and corresponding standard deviations were calculated from the snapshots collected every 100 time steps, which is well beyond the sample correlation time determined from the associated autocorrelation function.[6] A total of 100 000 to 1 million MD time steps were performed, where the longer runs correspond to the more dilute systems. The time averages are calculated from the second half of each run, well beyond the equilibration time.

The YR method is implemented in the development version of the HOOMD Blue package,[24] revision 3109. HOOMD Blue currently supports only single precision arithmetic. Simulations were performed on NVIDIA GTX 295 and GTX 480 GPUs installed in a custom built workstation with an Intel Core i7 920 CPU, 12 GB of RAM, running the Fedora 12 Linux operating system, CUDA 2.2, and NVIDIA Linux driver version 195.36.24. In all runs, only one of the two GPUs on the GTX 295 card was used while the other was kept idle. No monitors were attached to either GTX 295 or GTX 480 cards. On GTX 295, maximum performance is achieved with 64 CUDA threads per block, while on GTX 480, the most optimal thread per block count was 160. Simulations with the ES method were performed in LAMMPS[25,26] on 32 CPU cores, that is, on four IBM iDataplex blades with two quad-core 2.4 GHz Intel Xeon E5520 processors, 48 GB of memory, and interconnected through a DDR InfiniBand network.

4. Results and Discussion

The YR method requires $O(N^2)$ computations to evaluate the total electrostatic energy, as opposed to $O(N^{3/2})$ computations in the ES method. However, due to its simplicity, computation of the electrostatic interaction in the YR method requires a relatively small number of simple arithmetic

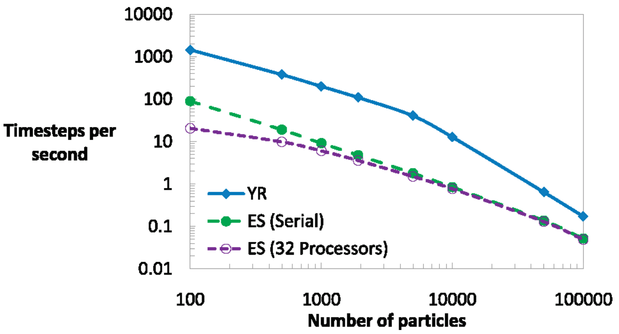

Figure 3. Time steps per second against the number of particles for the YR method on an NVIDIA GTX 480 GPU, and the serial and parallel executions of the ES method on an Intel Xeon computer cluster, for 0.1 M concentration of monovalent salt. The time steps per second for the ES method are defined per unit processor.

operations compared to significantly more complex calculations needed for the same evaluation in the ES method. Therefore, the YR method is significantly faster than the ES method even for 10^5 particles, as shown in Figure 3. The performance gain is even higher when compared to the parallel execution of the ES method, since a significant amount of time is spent in communication between processors for this range of simulation sizes. Note that while the YR method is free of adjustable parameters, the performance of the ES method is sensitive to changes in the real-space cutoff and the reciprocal space precision. We use a real space cutoff of approximately one-fifth of the simulation box size and the reciprocal space calculations were performed with a precision of 10^{-5}. Simulations on the NVIDIA GTX 480 were approximately twice as faster as that on the NVIDIA GTX 295.

Next, we evaluate the thermodynamic predictions of the YR and ES methods for a range of concentrations of monovalent and divalent electrolytes. The number of ions in the simulation box can be chosen arbitrarily as long as the electroneutrality condition is preserved. In this study, we use 1912 ions, a number chosen to balance a full utilization of the GPU with reasonably short execution times. The calculated values of electrostatic energy and heat capacity from the ES and YR methods are shown in Tables 1 and 2 for monovalent and divalent electrolytes, respectively. We observed an excellent agreement between the two methods for a wide range of concentrations—even for divalent ions, where the electrostatic correlations are stronger. Note that there is an appreciable difference in the heat capacities obtained by the ES and the YR methods for dilute systems, in particular for the monovalent case. We attribute this to the inability of the MD to successfully equilibrate a dilute system of charges,[27,28] and we believe this is not a drawback of the YR method.

To evaluate the concordance of the YR and ES methods in reproducing the structural details of the electrical double layer, we calculate the radial distribution functions, g_{++} (r^*) and g_{+-} (r^*), of the like- and unlike-charged ions, respectively, where $r^* = r/\sigma$ is the reduced distance from the center of the reference ion. These quantities are averaged from snapshots taken every 10 time steps. Radial distribution functions calculated by the two methods show excellent agreement, as is clear from the curves being virtually indistinguishable in Figures 4 and 5. Further, in both methods, $g_{++}(r^*)$ and g_{+-} (r^*) approach one far from the central ion and at the border of the simulation box, as shown in the insets of Figures 4 and 5. This condition is necessary to ascertain that the system is free of finite size effects, and it is often not met in truncation schemes for handling electrostatic interactions, even for significantly large simulation box sizes.[29] For the monovalent ions at 0.01 M (Figure 4a), the contact values show an attraction and a repulsion between unlike- and like-charged ions, respectively, as is expected of bare Coulomb interactions. Interestingly, for the 1 M concentration (Figure 4b), the excluded volume of hydrated monovalent ions leads to a slight attraction between like-charged ions. For the divalent case at 0.005 M (Figure 5a), we observe repulsion and attraction of like- and unlike-charged ions, respectively, that increased in comparison to the monovalent instance at 0.01 M (Figure 4a). In addition, at a 0.5 M concentration (Figure 5b) of divalent ions, we observe a region of charge inversion between $r^* \approx 1.7$ and $r^* \approx 2.6$.

A more stringent test is the calculation of the integrated charge of ions,[30]

Table 1. Average Electrostatic Energy per Ion, $E^* = \langle E_{\mathrm{el}} \rangle / Nk_{\mathrm{B}}T$ and Heat Capacity per Ion, $C^* = C_v/Nk_{\mathrm{B}}$ Calculated by the ES and YR Methods for 1:1 Bulk Hydrated Electrolyte at Different Salt Concentrations (ρ)[a]

ρ [M]	E^*_{ES}	E^*_{YR}	C^*_{ES}	C^*_{YR}
1.0	−0.4470(4)	−0.4472(3)	0.095(1)	0.093(1)
0.75	−0.4144(3)	−0.4145(4)	0.091(1)	0.094(2)
0.5	−0.3722(3)	−0.3721(3)	0.095(1)	0.092(1)
0.25	−0.3066(3)	−0.3067(3)	0.085(1)	0.084(1)
0.1	−0.2316(2)	−0.2319(3)	0.074(1)	0.077(1)
0.075	−0.2105(1)	−0.2106(3)	0.069(1)	0.072(1)
0.05	−0.1828(2)	−0.1828(2)	0.062(1)	0.065(1)
0.025	−0.1415(2)	−0.1419(2)	0.055(1)	0.055(1)
0.01	−0.09829(5)	−0.09842(3)	0.0424(2)	0.0428(2)
0.0075	−0.08712(6)	−0.08713(4)	0.0366(2)	0.0375(1)
0.005	−0.07329(4)	−0.07331(2)	0.0314(2)	0.0321(1)
0.0025	−0.05403(7)	−0.05415(3)	0.0212(1)	0.0239(2)
0.001	−0.03530(2)	−0.03529(1)	0.0170(1)	0.0150(1)

[a] Uncertainties in the last digit are indicated in parentheses.

Table 2. Average Electrostatic Energy per Ion, $E^* = \langle E_{\mathrm{el}} \rangle / Nk_{\mathrm{B}}T$ and Heat Capacity per Ion, $C^* = C_v/Nk_{\mathrm{B}}$ Calculated by the ES and YR Methods for 2:2 Bulk Hydrated Electrolyte at Different Salt Concentrations (ρ)[a]

ρ [M]	E^*_{ES}	E^*_{YR}	C^*_{ES}	C^*_{YR}
1.0	−2.056(3)	−2.056(4)	0.27(1)	0.28(1)
0.75	−1.931(2)	−1.931(3)	0.28(1)	0.28(1)
0.5	−1.773(2)	−1.773(2)	0.28(1)	0.30(1)
0.25	−1.533(2)	−1.533(2)	0.31(1)	0.31(1)
0.1	−1.254(3)	−1.254(2)	0.32(1)	0.31(1)
0.075	−1.173(2)	−1.173(3)	0.31(1)	0.31(2)
0.05	−1.063(2)	−1.063(3)	0.31(2)	0.31(1)
0.025	−0.888(3)	−0.887(2)	0.30(1)	0.29(1)
0.01	−0.676(2)	−0.675(3)	0.29(2)	0.26(2)
0.0075	−0.616(2)	−0.617(2)	0.24(1)	0.25(1)
0.005	−0.535(3)	−0.535(3)	0.27(2)	0.25(2)
0.0025	−0.413(2)	−0.414(2)	0.20(1)	0.21(1)
0.001	−0.283(1)	−0.285(1)	0.159(3)	0.164(9)

[a] Uncertainties in the last digit are indicated in parentheses.

$$P_i(r) = z_i + \int_0^r [\sum_{j=+,-} z_j \rho_j g_{ij}(r')] 4\pi r'^2 dr' \qquad (19)$$

where ρ_j is the bulk density of ion species j in the simulation box. $P_i(r)$ corresponds to the net charge inside a sphere of radius r centered at an ion of species i and hence measures the neutralization of such an ion by the surrounding ionic cloud. At the surface of an ion of species i ($r = 0$), the integrated charge is equal to its valence z_i, whereas sufficiently far from the ion ($r \rightarrow \infty$), $P_i(r)$ approaches zero due to the electroneutrality condition. $P_+(r)$ is identical to $-P_-(r)$ for electrolytes symmetric in valence and size.

The integrated charge of a positive ion, $P_+(r^*)$, for the monovalent and divalent electrolytes are displayed in Figures 6 and 7, respectively. As expected from the radial distribution functions, the concordance of the YR and ES methods is very good for both monovalent and divalent salts, especially near the ionic surface. The fluctuations in the integrated charge near the border of the simulation box are displayed in the insets of Figures 6 and 7. As a check of the global electroneutrality condition, we require that the integrated charge $P_+(r^*)$ approaches zero near the boundary, which is indeed met by the two methods disregarding minor statistical fluctuations. For the monovalent electrolyte at 0.01 M concentration (Figure 6a), the profile of $P_+(r^*)$ shows a monotonic neutralization of the ionic charge. In contrast, for 1 M concentration (Figure 6b), a nonmonotonic neutralization is observed. In fact, there is a region near the ionic surface where the sign of the integrated charge is opposite the sign of the central ion, indicating charge reversal. This behavior is caused by the large excluded volume associated with the hydrated monovalent ions. For the 0.005 M concentration of divalent electrolyte (Figure 7a), a monotonic ionic neutralization behavior akin to that of the 0.01 M monovalent case (Figure 6a) is observed. However, for the 0.5 M concentration of the divalent electrolyte (Figure 7b), the magnitude of maximum charge reversal near the ionic surface increased compared to the 1 M monovalent instance (Figure 6b), and several oscillations in the integrated charge are observed.

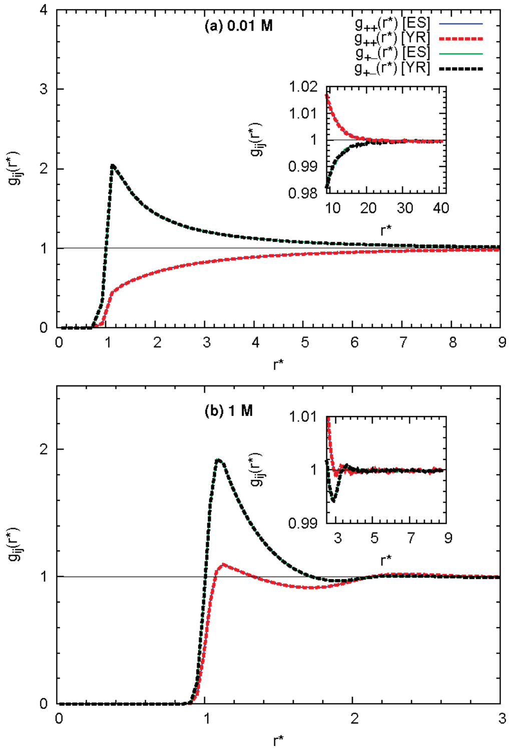

Figure 4. Pair distribution functions, g_{++} (r^*) and g_{+-} (r^*), for 1:1 electrolyte at different concentrations. Bold and dashed lines indicate the ES and YR methods, respectively. Notice that the profiles obtained by the ES and YR methods are virtually indistinguishable. Behavior near the box boundary is shown in the insets.

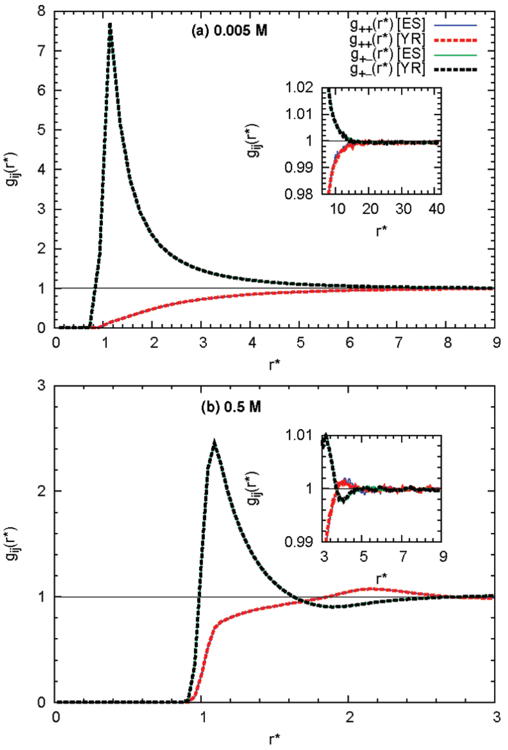

Figure 5. Pair distribution functions, g_{++} (r^*) and g_{+-} (r^*), for 2:2 electrolyte at different concentrations. Bold and dashed lines indicate the ES and YR methods, respectively. Notice that the profiles obtained by the ES and YR methods are virtually indistinguishable. Behavior near the box boundary is shown in the insets.

5. Conclusion

We have implemented an efficient method for long-range electrostatic interactions in the molecular dynamics on graphics processing units (GPU) based on the scheme originally proposed by Yakub and Ronchi.[14] The method is implemented in the MD package HOOMD Blue.[24] In order to test the accuracy of this method applied to the electrolyte systems, thermodynamic and structural properties of bulk hydrated monovalent and divalent salts were calculated. An excellent agreement was found with respect to the conventional Ewald summation method, available in LAMMPS. The current implementation of the YR method is particularly suited for moderate to high concentrations of charges. Its limited applicability to dilute systems is not a flaw of the method but, we believe, is an artifact of MD simulations related to their inability to reach thermodynamic equilibrium in a reasonable time. Additionally, the GPU implementation

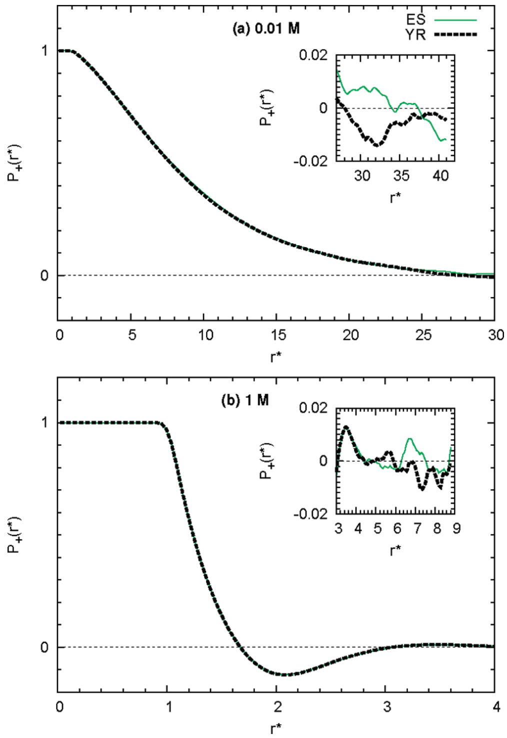

Figure 6. Integrated charge, $P_+(r^*)$, of the 1:1 electrolyte at different concentrations. Bold and dashed lines indicate the ES and YR methods, respectively. Notice that the profiles obtained by the ES and YR methods are virtually indistinguishable. Fluctuations of the integrated charge near the boundary of the box are shown in the insets.

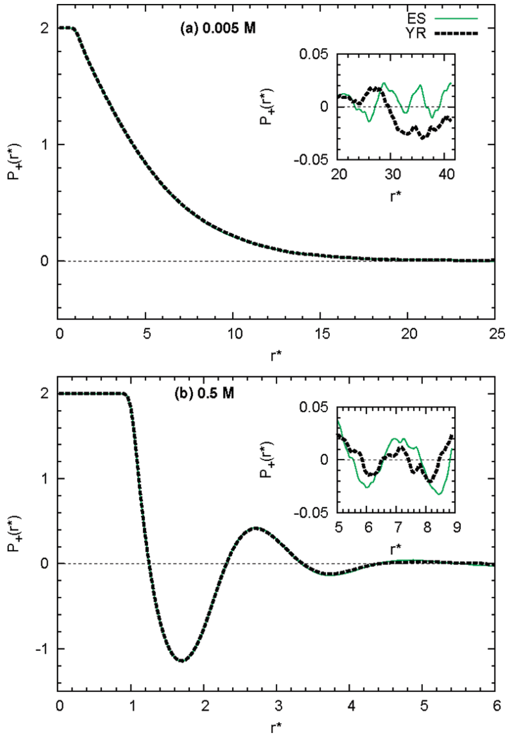

Figure 7. Integrated charge, $P_+(r^*)$, of the 2:2 electrolyte at different concentrations. Bold and dashed lines indicate the ES and YR methods, respectively. Notice that the profiles obtained by the ES and YR methods are virtually indistinguishable. Fluctuations of the integrated charge near the boundary of the box are shown in the insets.

is significantly faster than the fully optimized Ewald summation method for the simulation sizes commonly used in simulations of electrolytes (10^2 to 10^5).

We would like to mention that there is another class of finite or short-range methods for electrostatic interactions such as the Wolf method and its variations[31−33] that can potentially also benefit from the GPU's high FLOPS count. However, in such schemes, the cutoff and the damping constant must be calibrated for each particular system, whereas the YR method is free of adjustable parameters. The present implementation can be easily extended to study more complicated systems including charged spherocylinders,[34,35] nanoparticles and colloids,[33,36−40] asymmetric ionic liquids,[41−44] and polyelectrolyte solutions and networks,[45−48] with the incorporation of the corresponding short-range interactions that are already available in the GPU codes. Efforts in these directions are currently underway.

Acknowledgment. G.I.G.-G. thanks Eugene Yakub for his invaluable help in clarifying certain numerical and theoretical aspects of the YR method. The authors thank Graziano Vernizzi for stimulating discussions and William Kung for suggestions on the manuscript. We thank anonymous reviewers for useful suggestions. Molecular dynamics simulations were in part performed on Quest cluster at Northwestern University. A part of GPU simulations presented in this work were executed on GTX 480 GPUs provided by NVIDIA through their professor partnership program. P.K.J. and M.O.d.l.C. are supported by NSF under Award number DMR-0907781. R.S. is supported by the NSEC program on integrated nanopatterning (EEC-0647560) at Northwestern University. G.I.G.-G. is supported by the MRSEC program of the NSF (DMR-0520513) at the Materials Research Center at Northwestern University.

References

(1) Anderson, J. A.; Lorenz, C. D.; Travesset, A. *J. Comput. Phys.* **2008**, *227*, 5342–5359.

(2) Harvey, M. J.; De Fabritiis, G. *J. Chem. Theory Comput.* **2009**, *5*, 2371–2377.

(3) Stone, J. E.; Phillips, J. C.; Freddolino, P. L.; Hardy, D. J.; Trabuco, L. G.; Schulten, K. *J. Comput. Chem.* **2007**, *28*, 2618–2640.

(4) Owens, J. D.; Houston, M.; Luebke, D.; Green, S.; Stone, J. E.; Phillips, J. C. *Proc. IEEE* **2008**, *96*, 879–899.

(5) Friedrichs, M. S.; Eastman, P.; Vaidyanathan, V.; Houston, M.; Legrand, S.; Beberg, A. L.; Ensign, D. L.; Bruns, C. M.; Pande, V. S. *J. Comput. Chem.* **2009**, *30*, 864–872.

(6) Frenkel, D.; Smit, B. *Understanding Molecular Simulation, Second ed.: From Algorithms to Applications* (Computational Science Series, Vol 1), 2nd ed.; Academic Press: New York, 2001.

(7) Adams, D. J. *Chem. Phys. Lett.* **1979**, *62*, 329–332.

(8) Brush, S. G.; Sahlin, H. L.; Teller, E. *J. Chem. Phys.* **1966**, *45*, 2102.

(9) de Leeuw, S. W.; Perram, J. W.; Smith, E. R. *Philos. Trans. R. Soc. London, Ser. A* **1980**, *373*, 27–56.

(10) Koehl, P. *Curr. Opin. Str. Bio.* **2006**, *16*, 142–151.

(11) Karttunen, M.; Rottler, J.; Vattulainen, I.; Sagui, C. In *Computational Modeling of Membrane Bilayers*; Feller, S. E., Ed.; Academic Press: New York, 2008; Vol. 60, pp 49−89.

(12) Essmann, U.; Perera, L.; Berkowitz, M. L.; Darden, T.; Lee, H.; Pederson, L. G. *J. Chem. Phys.* **1995**, *103*, 8577–8593.

(13) Hardy, D. J.; Stone, J. E.; Schulten, K. *Parallel Comput.* **2009**, *35*, 164–177.

(14) Yakub, E.; Ronchi, C. *J. Chem. Phys.* **2003**, *119*, 11556–11560.

(15) Yakub, E.; Ronchi, C. *J. Low Temp. Phys.* **2005**, *139*, 633–643.

(16) Yakub, E. *J. Phys. A—Math. Gen.* **2006**, *39*, 4643.

(17) Messina, R.; González-Tovar, E.; Lozada-Cassou, M.; Holm, C. *Europhys. Lett.* **2002**, *60*, 383.

(18) Jiménez-Ángeles, F.; Messina, R.; Holm, C.; Lozada-Cassou, N. *J. Chem. Phys.* **2003**, *119*, 4842–4856.

(19) Nightingale, E. R. *J. Phys. Chem.* **1959**, *63*, 1381–1387.

(20) Israelachvili, J. *Intermolecular and Surface Forces*, 2nd ed.; Academic Press: London, 1992.

(21) Quesada-Pérez, M.; González-Tovar, E.; Martín-Molina, A.; Lozada-Cassou, M.; Hidalgo-Álvarez, R. *ChemPhysChem* **2003**, *4*, 234–248.

(22) Messina, R. *J. Phys.: Condens. Matter* **2009**, *21*, 113102.

(23) Rapaport, D. C. *The Art of Molecular Dynamics Simulation*, 2nd ed.; Cambridge University Press: Cambridge, U. K., 2004.

(24) HOOMD Blue. http://codeblue.umich.edu/hoomd-blue/ (August 20, 2010).

(25) Plimpton, S. *J. Comput. Phys.* **1995**, *117*, 1–19.

(26) LAMMPS Molecular Dynamics Simulator. http://lammps.sandia.gov/ (August 20, 2010).

(27) Oran, E. S.; Boris, J. P. *J. Phys. IV France* **1995**, *5*, 609.

(28) Zhang, Z.; Duan, Z. *Chem. Phys.* **2004**, *297*, 221–233.

(29) Linse, P.; Andersen, H. C. *J. Chem. Phys.* **1986**, *85*, 3027–3041.

(30) Guerrero-García, G. I.; González-Tovar, E.; Chávez-Páez, M. *Phys. Rev. E* **2009**, *80*, 021501.

(31) Wolf, D.; Keblinski, P.; Phillpot, S. R.; Eggebrecht, J. *J. Chem. Phys.* **1999**, *110*, 8254–8282.

(32) Fennell, C. J.; Gezelter, J. D. *J. Chem. Phys.* **2006**, *124*, 234104.

(33) Avendaño, C.; Gil-Villegas, A. *Mol. Phys.* **2006**, *104*, 1475–1486.

(34) Avendaño, C.; Gil-Villegas, A.; González-Tovar, E. *J. Chem. Phys.* **2008**, *128*, 044506.

(35) Avendaño, C.; Gil-Villegas, A.; González-Tovar, E. *Chem. Phys. Lett.* **2009**, *470*, 67–71.

(36) Linse, P.; Lobaskin, V. *Phys. Rev. Lett.* **1999**, *83*, 4208–4211.

(37) Guerrero-García, G. I.; González-Tovar, E.; Lozada-Cassou, M.; Guevara-Rodrguez, F. D. *J. Chem. Phys.* **2005**, *123*, 034703.

(38) Guerrero-García, G. I.; González-Tovar, E.; de la Cruz, M. O. *Soft Matter* **2010**, *6*, 2056–2065.

(39) Guerrero-García, G. I.; González-Tovar, E.; Chávez-Páez, M.; Lozada-Cassou, *J. Chem. Phys.* **2010**, *132*, 054903.

(40) dos Santos, A. P.; Diehl, A.; Levin, Y. *J. Chem. Phys.* **2010**, *132*, 104105.

(41) Yan, Q. L.; de Pablo, J. J. *Phys. Rev. Lett.* **2001**, *86*, 2054–2057.

(42) Yan, Q. L.; de Pablo, J. J. *Phys. Rev. Lett.* **2002**, *88*, 095504.

(43) Hynninen, A. P.; Dijkstra, M.; Panagiotopoulos, A. Z. *J. Chem. Phys.* **2005**, *123*, 084903.

(44) Panagiotopoulos, A. Z. *J. Phys.: Condens. Matter* **2005**, *17*, S3205–S3213.

(45) Liao, Q.; Dobrynin, A. V.; Rubinstein, M. *Macromolecules* **2003**, *36*, 3399–3410.

(46) Yin, D.-W.; de la Cruz, M. O.; de Pablo, J. J. *J. Chem. Phys.* **2009**, *131*, 194907.

(47) Narambuena, C.; Leiva, E.; Chávez-Páez, M.; E. Pérez, E. *Polymer* **2010**, *51*, 3293–3302.

(48) Hsiao, P.-Y.; Luijten, E. *Phys. Rev. Lett.* **2006**, *97*, 148301.

CT100365C