

# Accelerating electrostatic pair methods on graphical processing units to study molecules in supercritical carbon dioxide†

John A. Baker and Jonathan. D. Hirst\*

Received 14th February 2014, Accepted 5th March 2014

DOI: 10.1039/c4fd00012a

Traditionally, electrostatic interactions are modelled using Ewald techniques, which provide a good approximation, but are poorly suited to GPU architectures. We use the GPU versions of the LAMMPS MD package to implement and assess the Wolf summation method. We compute transport and structural properties of pure carbon dioxide and mixtures of carbon dioxide with either methane or difluoromethane. The diffusion of pure carbon dioxide is indistinguishable when using the Wolf summation method instead of PPPM on GPUs. The optimum value of the potential damping parameter,  $\alpha$ , is 0.075. We observe a decrease in accuracy when the system polarity increases, yet the method is robust for mildly polar systems. We anticipate the method can be used for a number of techniques, and applied to a variety of systems. Substitution of PPPM can yield a two-fold decrease in the wall-clock time.

## 1. Introduction

GPU architectures and their use in molecular dynamics (MD) simulations have attracted much recent attention.<sup>1–4</sup> Maximizing computational throughput on graphical processing units (GPUs)/hybrid architectures is of great interest. Decreasing the wall-clock time for each MD time step allows longer timescales to be sampled, giving greater confidence in the convergence of ensemble averages. GPUs have been widely adopted in various computational disciplines, due to the highly parallel design, low power consumption (FLOPS/Watt) and commodity cost (£/FLOPs). The GPU architecture is optimised for processing massive amounts of parallel calculations, and so has many more cores than a CPU, but at the sacrifice of memory capabilities.

Electrostatic interactions can be divided into first-order effects, which comprise point charge interactions decaying reciprocally with respect to inter-molecular distance and higher-order interactions that decay more rapidly.

School of Chemistry, University of Nottingham, University Park, Nottingham, NG7 2RD, UK. E-mail: Jonathan.Hirst@nottingham.ac.uk; Fax: +44 1159 513562; Tel: +44 1159 513478

† Electronic Supplementary Information (ESI) available: See DOI: 10.1039/c4fd00012a



Electrostatic interactions can be modelled explicitly by various methods, including: cut-off truncation,<sup>5–7</sup> switched/shifted cut-off truncation,<sup>8</sup> Ewald<sup>9</sup> and its mesh derivatives, particle mesh Ewald (PME),<sup>10</sup> particle–particle particle mesh (PPPM)<sup>11</sup> and smoothed particle mesh Ewald (SPME).<sup>12</sup> The Ewald summation provides a good approximation for the electrostatic energy, as the algorithm accounts for periodicity of the simulation domain, but it scales as  $O(N^{3/2})$ . An approximation that is frequently adopted is PPPM, due to its superior scaling,  $O(N \log N)$ , over the Ewald summation. SPME,<sup>13</sup> PPPM<sup>14</sup> and multi-level summation<sup>15</sup> have been implemented on the GPU, yet the requirement for fast Fourier transforms (FFTs) intrinsic to Ewald methods, like SPME and PPPM, reduces the parallelism<sup>14</sup> and leads to poor scaling on GPUs. Pair-wise algorithms derived from cut-off techniques show superior scaling  $O(N)$  and greater effectiveness on GPUs.

Cut-off techniques can reproduce experimental quantities, for instance, the Madelung constant. Wolf *et al.*<sup>16</sup> reported that the electrostatic (Coulombic) potential for condensed systems was effectively short-ranged, and the energies are in agreement with Ewald methods when the cut-off sphere reached neutrality. This was observed whilst calculating the Madelung constant for rocksalt (NaCl), where the relationship between truncation distance and electrostatic energy was investigated. When the cut-off sphere was truncated at a distance to achieve charge neutrality, the electrostatic energy was significantly closer to the Madelung energy. The repeating lattice structure of NaCl is well-suited for treatment with the Wolf method, but the supercritical phase of polar solutes can be more difficult to model as the atomic partial charges within the cut-off sphere are dynamic. The resulting shifted Coulomb potential achieves charge neutrality by projecting each atom charge,  $q_j$ , from  $q_i$  onto the edge of the solvation sphere. Therefore, every  $j$ th atom in the solvation sphere of atom  $i$  has a charge of equal but opposite sign set at the cut-off ( $R_c$ ). This results in an artificially neutral solvation sphere for every  $i$ th atom, which effectively makes the system charge neutral.

$$E_{\text{Wolf}} = \frac{1}{2} \sum_{i=1}^N \sum_{\substack{j(i) \\ r_{ij} < R_c}} \left[ \left( \frac{q_i q_j \text{erfc}(\alpha r_{ij})}{r_{ij}} - \lim_{r_{ij} \rightarrow R_c} \left\{ \frac{q_i q_j \text{erfc}(\alpha R_c)}{R_c} \right\} \right) - \left( \frac{\text{erfc}(\alpha R_c)}{2R_c} + \frac{\alpha}{\sqrt{\pi}} \right) \sum_{i=1}^N q_i^2 \right]$$

where  $q_i$  and  $q_j$  are point charges,  $r_{ij}$  is the intermolecular distance between atoms  $i$  and  $j$ ,  $\alpha$  is the damping coefficient in  $\text{\AA}^{-1}$  and  $R_c$  is the cut-off distance. To obtain accurate electrostatic contributions, a damping function is applied. The electrostatic energy decay oscillates around a rate of  $1/R_c$ . Introducing the damping function quickly flattens the oscillations as the cut-off increases, effectively determining how fast the complementary error function falls from unity at  $r_{ij} = 0$ , to zero at the cut-off.<sup>16</sup> The damping function adopted is the complementary error function, as it has a close connection to the Ewald sum.<sup>17</sup> The coefficient of the error function,  $\alpha$ , denotes the rate at which convergence is achieved. A large  $\alpha$  value will converge the energy rapidly using a short cut-off, but the errors can be larger. A smaller  $\alpha$  leads to less contamination of the potential, but the sum will fluctuate more rapidly. Assigning a value of 0 to  $\alpha$ , results in the truncated shifted-force (SF) potential. This SF calculation is faster than the Wolf method, but the



selection of  $\alpha$  can enhance the accuracy of electrostatic forces and energies by optimizing agreement to Ewald methods.<sup>18</sup>

Simple cut-off based methods are unreliable for computing the forces, as the potential truncates abruptly at the cut-off, which causes forces to be undefined if the molecule lies at the cut-off boundary. The reliance upon Ewald methods instead of efficient cut-off methods has been discussed by Fennel and Gezelter,<sup>17</sup> who compared the accuracy of damped shifted-force (DSF) and shifted potential cut-off methods for water, argon in water, and NaCl. Benchmarking and validation of the DSF potential was applied to polyelectrolyte brushes on GPUs; speedups were achieved of a factor between 1.1 and 3.9, depending on the system and size of the cut-off with respect to PPPM.<sup>19</sup> Group based cut-offs, where the cut-off is dynamically allocated to ensure entire molecules are within the solvation shell, were investigated,<sup>17</sup> but the energies deviated significantly from those obtained using PME.

In this work we assess the applicability of the Wolf method to model electrostatic interactions for various systems in the supercritical region of carbon dioxide. Supercritical carbon dioxide (scCO<sub>2</sub>) is an attractive 'green solvent' used in many industrial processes, such as caffeine extraction,<sup>20</sup> polymer solvation<sup>21</sup> and synthesis,<sup>22,23</sup> enzyme catalysis<sup>24,25</sup> and stabilization<sup>26,27</sup> and transition metal catalysis.<sup>28,29</sup> Although carbon dioxide does not possess a dipole, it has a significant quadrupole<sup>30</sup> ( $13.4 \times 10^{-40}$  Cm<sup>2</sup>), which means that electrostatic interactions are an important component of interactions involving scCO<sub>2</sub>. Su and Maroncelli<sup>31</sup> observed that neglecting electrostatic interactions, *i.e.*, considering only Lennard-Jones interactions, led to a systematic 14% error in the solvation free energies of polymer-scCO<sub>2</sub> systems. This observation is attributed to quadrupole-dipole and quadrupole-quadrupole interactions,<sup>31</sup> which are inherently modelled in all point charge models. A nonpolar fluid should be well suited for treatment using the Wolf method, as quadrupolar interactions decay more rapidly than dipolar ones. This bodes well for the efficient electrostatic treatment of systems solvated by carbon dioxide by utilizing the Wolf method on GPU architectures. We also investigate the computational cost of the DSF method, with respect to the Wolf method.

Fluorous polymers are well known to dissolve in scCO<sub>2</sub>,<sup>32,33</sup> which has been partially explained by <sup>19</sup>F-NMR experiments, which suggested a number of specific interactions between carbon dioxide and the fluorous solute that increase the solubility.<sup>34</sup> Many biomolecular systems are stable in the presence of scCO<sub>2</sub>, but this is dependent on the species, the water content, and experimental conditions.<sup>24</sup> Protein stability can be observed when scCO<sub>2</sub> solvates the hydrophobic residues, and water solvates the polar/hydrophilic residues.<sup>26</sup> Water possesses a strong dipole moment, which will make it less amenable to treatment with the Wolf method. We aim to follow this work with an investigation for protein systems solvated by water and carbon dioxide.

To assess the applicability of the Wolf method to systems containing carbon dioxide, important physical quantities, such as PVT relationships and diffusion coefficients, can be calculated and compared with the values calculated using PPPM. The applicability of the Wolf method to study methane plus carbon dioxide gas hydrates has been reported, and the results show good agreement between lattice sum and reaction field methods.<sup>35</sup> Analysis of the convergence behaviour of the Wolf method by Angoshtari and Yavari<sup>36</sup> shows the method to be



robust, but convergence can be problematic if poor choices are made for the cut-off or  $\alpha$  values. In our study, we investigate the effect of increasing the polarity of the fluid by incorporating difluoromethane molecules, which possess a strong dipole. For low polarity systems the Wolf method should be well suited, but as the polarity increases the effect of long-range dipole interactions will become important and we assess the point where Ewald techniques will become necessary.

## 2. Method

LAMMPS is a multi-purpose MD code,<sup>37</sup> which is widely used in the fields of atomistic, coarse-grained and mesoscopic simulations. The code can process massive numbers of particles per simulation, which it achieves using an optimized spatial decomposition technique. The GPU optimized version, written in CUDA C, can run over multiple GPUs, either in conjunction with the CPU or entirely on the GPU. We have implemented the Wolf method into LAMMPS-CUDA the GPU-exclusive version of LAMMPS and LAMMPS-GPU the CPU/GPU implementation, as a new potential incorporating Lennard-Jones and electrostatic interactions. LAMMPS-CUDA was written exclusively for use with CUDA, but LAMMPS-GPU can be compiled and run on AMD GPUs and other applicable accelerators using OpenCL. The new pair style is implemented as a double potential function in LAMMPS as lj/charmm/coul/wolf for CPU, lj/charm/coul/wolf/cuda for LAMMPS-CUDA and lj/charm/coul/wolf/gpu for LAMMPS-GPU. The LAMMPS-CUDA version is solely GPU based, with the exception of file I/O and pre/post simulation setup. The GPU version of LAMMPS utilizes the GPU for the force and/or neighbour list generation, whilst all other operations and I/O are performed on the CPU. The GPU version allows for  $n$  CPUs to be used per GPU, whilst the LAMMPS-CUDA version only allows for one CPU per GPU. An important difference between packages is that the GPU version on LAMMPS uses the CPU to calculate FFTs for PPPM, whilst the LAMMPS-CUDA version performs the calculation on the GPU. We have also implemented the DSF implementations, lj/charm/coul/dsf/cuda and lj/charm/coul/dsf/gpu, for benchmarking purposes. We used the CHARMM<sup>38</sup> Lennard-Jones potential in the AB form with a switching function in conjunction with the Wolf method. A cut-off of  $2.5\sigma$  was used for the Lennard-Jones potential, and tapered to zero at  $2.65\sigma$ , where  $\sigma$  was calculated

**Table 1** EPM2 force field parameters for carbon dioxide<sup>39</sup> and Palmer and Anchell parameters for methane and difluoromethane.<sup>40</sup> ( $A^{1/12}$ ) and ( $B^{1/6}$ ) have units ((kcal mol)<sup>1/12</sup> Å) and ((kcal mol)<sup>1/6</sup> Å) respectively

| Atom site                           | $A^{1/12}$ | $B^{1/6}$ | $q$ ( $ e $ ) |
|-------------------------------------|------------|-----------|---------------|
| C (CO <sub>2</sub> )                | 2.448      | 2.173     | −0.3256       |
| O (CO <sub>2</sub> )                | 2.922      | 2.815     | +0.6512       |
| C (CH <sub>4</sub> )                | 3.200      | 3.200     | −0.4160       |
| H (CH <sub>4</sub> )                | 1.910      | 1.390     | +0.1040       |
| C (CH <sub>2</sub> F <sub>2</sub> ) | 2.900      | 3.590     | +0.0500       |
| H (CH <sub>2</sub> F <sub>2</sub> ) | 1.712      | 0.000     | +0.1550       |
| F (CH <sub>2</sub> F <sub>2</sub> ) | 2.650      | 2.237     | −0.1800       |



from  $A^{1/12}$  in Table 1. We used the rigid EPM2 atomistic force field<sup>39</sup> to represent carbon dioxide and parameters derived by Palmer and Anchell<sup>40</sup> to represent Lennard-Jones and charge interactions for methane and difluoromethane.

We utilized the LAMMPS software to simulate  $scCO_2$  at several densities, comparing the results obtained by using PPPM or the Wolf method. We selected a tolerance setting of 0.0001 for the calculations involving PPPM that enables the root mean square error to be within a factor of 10 000 of the reference force, which is calculated analytically for short-range interactions<sup>41</sup> and in  $k$ -space.<sup>42</sup> The optimum value of  $\alpha$  was investigated for binary mixtures of carbon dioxide and methane or difluoromethane. The criterion for selecting  $\alpha$  is the level of agreement between the Coulombic energies computed by PPPM and Wolf. To consider the effects of neglecting only the periodic long-range Coulombic effects, we treat the cut-off as being half the length of the simulation box. We also investigate the level of approximation that arises from use of a quarter box cut-off in conjunction with the Wolf method.

## 2.1 Benchmarking GPU electrostatics

Two boxes of 10 000 and 50 000 carbon dioxide molecules were constructed with a molar density of  $0.01 \text{ mol cm}^{-3}$  and minimized for 1000 iterations using the conjugate gradient method, to an energy tolerance of  $1 \times 10^{-6} \text{ kcal mol}^{-1}$ . Both systems were heated from 0 K to 308.2 K using the Nosé–Hoover thermostat<sup>43</sup> with a damping time of 500 fs for 2 ns, followed by 5 ns of equilibration at 308.2 K. We evaluated the performance of the Wolf method on two GPU architectures, Tesla (Tesla C1060) and Kepler (Tesla K10). The K10, released in 2012, is designed for high throughput calculations performed in single precision. The number of giga floating-point operations per second (GFLOPs) is 4577 in single precision, compared to 933 for the C1060. This improvement was achieved in part by increasing the number of cores in the streaming multiprocessor from 8 (Tesla), 32 (Fermi) to 192 (Kepler). Memory bandwidth on the GPU to global memory is  $103 \text{ GB s}^{-1}$  for the C1060, but  $320 \text{ GB s}^{-1}$  for Kepler K10. We include an eight core Intel Xeon (E5-2609) CPU for benchmarking purposes that has an approximate 77 GFLOPs with  $34 \text{ GB s}^{-1}$  bandwidth to RAM. Calculations were performed using single-precision on two separate nodes, both comprise an eight core Xeon CPU and either a Kepler K10 or two Tesla C1060. Multiple GPU tests were only carried out using the C1060, which yielded an almost two-fold increase in performance, which corresponds well to the linear dependency. We use the FFTW 3.3.1 library in single-precision for all  $k$ -space calculations of PPPM. We considered two modes for calculations using LAMMPS-GPU, one where all force and neighbour calculations are performed on the GPU, and the other where the force and neighbour calculations are dynamically assigned between CPU and GPU. We used the CUDA 5.0 Tokito (GPU driver v. 304.54), which has produced a noticeable improvement in performance over CUDA 4.0. A new feature, CUDA dynamic parallelism, allows kernels running on the GPU to spawn more grids and to continue to generate work depending on the calculation.<sup>44</sup> This feature has not been incorporated in our study.

## 2.2 Pure carbon dioxide

The 10 000 carbon dioxide molecule system was used to generate densities (box lengths) of 0.001 (255.1 Å), 0.002 (202.5 Å), 0.004 (160.72 Å), 0.005 (149.2 Å),



0.01 (118.42 Å) and 0.02 (94.0 Å) mol cm<sup>-3</sup>, all within the supercritical region at 308.2 K. Each box was minimized for 1000 iterations using the conjugate gradient method, with an energy tolerance of  $1 \times 10^{-6}$  kcal mol<sup>-1</sup>. The system was heated in the NVT ensemble using the Nosè–Hoover thermostat<sup>43</sup> with cubic periodic boundary conditions from 0 K to 308.2 K for 1 ns using a 1 fs time step. This was followed by 1 ns of equilibration. Integration was performed using the time-reversible velocity Verlet algorithm.<sup>45</sup> The procedure was repeated nine times to generate different equilibrated configurations for the purposes of error analysis, and the resulting standard deviation is noted in the error bars. The mean square displacement (MSD) of the centre of mass of carbon dioxide was obtained by calculating the gradient of the linear portion of the relationship between MSD and lag time over 10 ns production dynamics in the NVT ensemble. The pressures were obtained every 50 fs, and averaged over 10 ns to calculate the PVT relationship.

### 2.3 Carbon dioxide and solute

To study the effects of polar and non-polar solutes, we considered different quantities of methane and difluoromethane molecules. Five boxes of 10 000 carbon dioxide molecules were constructed with mole fractions  $\chi_{(\text{solute})} = 0.0001, 0.001, 0.01, 0.09$  and  $0.5$ , where the solute was either methane or difluoromethane. These systems correspond to box lengths (Å) of 95.01, 95.99, 97.49, 99.27 and 125.31 for CH<sub>4</sub>/CO<sub>2</sub> and 81.64, 81.78, 81.83, 84.44 and 105.16 for CH<sub>2</sub>F<sub>2</sub>/CO<sub>2</sub>. Each system was minimized and heated to 308.2 K using the same procedure as above. The pressure was equilibrated to 80 atmospheres (corresponding to a molar density of  $\sim 0.02$  mol cm<sup>-3</sup>), using the Berendsen barostat<sup>46</sup> with a 1 ps damping time for 20 ns with a 1 fs time step. The procedure was repeated nine times to generate different equilibrated configurations for the purposes of error analysis, and the resulting standard deviation is noted in the error bars. The MSDs of carbon dioxide and solutes were obtained from the linear portion of the relationship over a production run of 10 ns. We compare the errors in Coulombic energy with respect to  $\alpha$  for  $\chi_{(\text{solute})} = 0.09$  with PPPM and Wolf, which we performed by decomposing the energy into group contributions. We investigate the relationship between system polarity and diffusion coefficients, and the total Coulombic energy.

One method of quantifying the transport properties of a system is the MSD, and thus the diffusion coefficient using the Einstein relationship.

$$D = \lim_{t \rightarrow t_{\text{lag}}} \frac{1}{6t} \left\langle [r_i(t) - r_i(0)]^2 \right\rangle$$

where  $D$  is the macroscopic diffusion coefficient,  $r_i(t)$  is the position of the centre of mass at time  $t$ ,  $r_i(0)$  is the initial position of the centre of mass,  $t_{\text{lag}}$  is the lag time, and  $\langle [r_i(t) - r_i(0)]^2 \rangle$  is the ensemble averaged MSD. The diffusion coefficient is calculated from the slope of the MSD against lag time,<sup>47</sup> which is a measure of the atomic displacements through time.

The pressure of a system can be calculated using the virial equation (below), and the PVT relationship has an influence on the diffusion coefficients.<sup>48</sup> The long-range part of PPPM has a different contribution to the pressure.<sup>49</sup>

$$PV = Nk_{\text{B}}T + \frac{1}{x} \left\langle \sum_{i=1}^N r_i \cdot F_i \right\rangle$$



where  $P$  is the pressure of the system,  $V$  is the volume,  $N$  is the number of particles,  $x$  is the dimensionality,  $K_B$  is the Boltzmann constant and the term in brackets is the total intermolecular force multiplied by the interaction distances.

### 3. Results

#### 3.1 Benchmarking

To compare the efficiency of the Wolf method, we measured the computational throughput of the electrostatic routines using the 10 000 and 50 000 molecule systems. The timings of the implementations are calculated with respect to the number of neighbours in the cut-off sphere. Fig. 1(a) and 1(b) compare the throughput for different architectures, and show the acceleration gained when substituting PPPM for Wolf using LAMMPS-CUDA. The computational throughput is almost twice as high with the Wolf treatment than with PPPM. The Wolf method is slower than PPPM on the CPU, which is unexpected, but the CPU is well suited to handle FFTs that feature in PPPM. This is due to faster memory accesses between cache and compute cores, and the less parallel nature of the algorithm. The DSF method is marginally slower than PPPM in LAMMPS-CUDA, but is marginally faster than the Wolf method for the LAMMPS-GPU

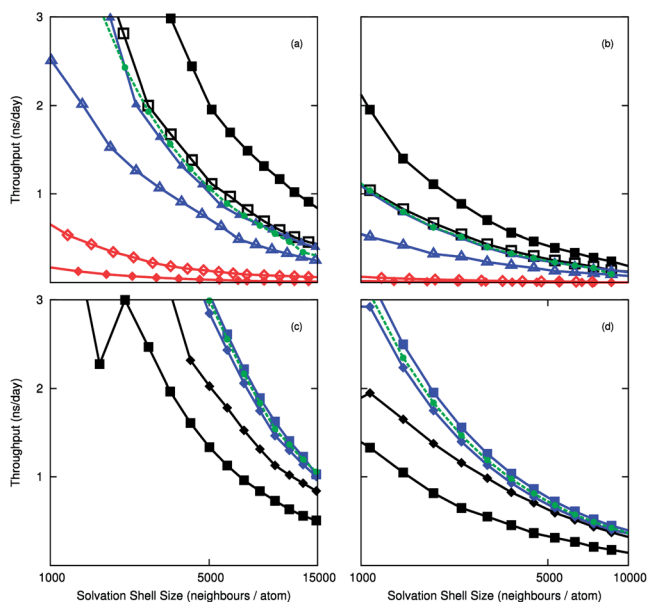


Fig. 1 Computational throughput as a function of number of neighbours for (a,c) 10 000 CO<sub>2</sub> molecules (B-D) 50 000 CO<sub>2</sub> molecules. Fig. (a) and (B) were calculated using the LAMMPS-CUDA package, and (c) and (d) were calculated using the LAMMPS-GPU package. In (a) and (B), three electrostatic treatments were considered, Wolf (filled shapes), PPPM (unfilled shapes) and DSF (green circles) for different architectures. (Triangles = Tesla C1060, squares = Kepler K10, diamonds = eight core Intel Xeon). In figures (c) and (d) the force/neighbour implementation is shown where the GPU is used exclusively (diamonds) and dynamically assigned between CPU/GPU (squares). PPPM is shown in black, the Wolf method is shown in blue, and the DSF force/neigh exclusively GPU is shown in green.





implementation. The difference between pair styles using LAMMPS-GPU is much less pronounced than LAMMPS-CUDA, where the acceleration gained by using the Wolf or DSF method is approximately 1.5 times faster. The difference between dynamic partitioning and exclusive GPU computation is small for the cut-off methods, and the dynamic mode is marginally quicker. For PPPM the opposite is observed, which could be due to the CPU already being used for the  $k$ -space calculations. The Kepler K10 GPU shows the greatest acceleration, which bodes well for future GPU releases.

### 3.2 Optimum alpha and cut-off parameters

We consider first the accuracy of the electrostatic energy and its convergence as a function of  $\alpha$  and cut-off distance. We then turn to the properties computed from the simulation, considering structural, dynamic and thermodynamic aspects. The simulations have all been executed in the supercritical region of carbon dioxide, and therefore each molecule has a large number of neighbours in the solvation sphere. We compare the effects of selecting  $\alpha$  values between zero and 0.3 in intervals of 0.025. We decompose the energy interactions into pairwise group contributions to the total Coulombic energy for CH<sub>4</sub> in carbon dioxide, and CH<sub>2</sub>F<sub>2</sub> in carbon dioxide. The relationship between  $\alpha$  and the accuracy of the Wolf summation method is shown in Fig. 2.

We observe for half box cut-offs the best agreement for Wolf is at low values of  $\alpha$ , where for both the polar and non-polar systems the optimum value for  $\alpha$  is

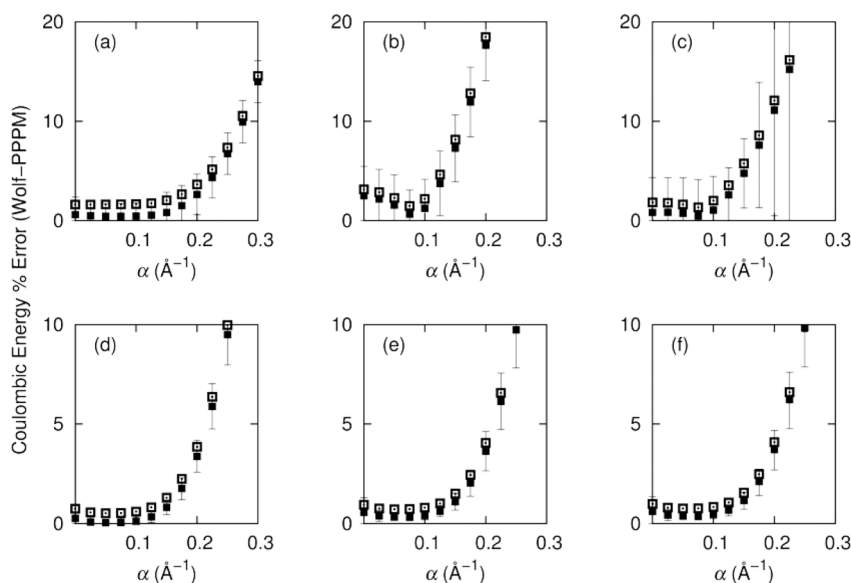


Fig. 2 Percentage error in Coulombic energy between PPPM and Wolf with respect to  $\alpha$  for CH<sub>4</sub>/CO<sub>2</sub> and CH<sub>2</sub>F<sub>2</sub>/CO<sub>2</sub> binary mixtures for  $\chi(\text{solute}) = 0.1$  at 308.2 K, 80 atmospheres. The results are decomposed into (a) (CH<sub>2</sub>F<sub>2</sub>) CO<sub>2</sub>–CO<sub>2</sub> pair errors (b) (CH<sub>2</sub>F<sub>2</sub>) CH<sub>2</sub>F<sub>2</sub>–CH<sub>2</sub>F<sub>2</sub> pair errors, (c) (CH<sub>2</sub>F<sub>2</sub>) CO<sub>2</sub>–CH<sub>2</sub>F<sub>2</sub> pair errors, (d) (CH<sub>4</sub>) CO<sub>2</sub>–CO<sub>2</sub> pair errors, (e) (CH<sub>4</sub>) CH<sub>4</sub>–CH<sub>4</sub> pair errors and (f) (CH<sub>4</sub>) CO<sub>2</sub>–CH<sub>4</sub> pair errors. The black squares and error bars indicate the average error and uncertainty for a half box cut-off, whilst the empty squares show the average error for a quarter box cut-off.





0.075. For non-polar systems increasing  $\alpha$  beyond 0.2 results in an  $\sim 2\text{--}3\%$  error (with respect to the PPPM Coulombic energy), whilst for the polar system the average error is  $\sim 20\%$ . The non-polar  $\text{CH}_4/\text{CO}_2$  system resulted in the lowest average errors of 0.05% for  $\text{CO}_2\text{--CO}_2$  interactions, 0.39% for  $\text{CO}_2\text{--CH}_4$  and 0.34% for  $\text{CH}_4\text{--CH}_4$ . We observe a similar trend for the polar system of  $\text{CO}_2/\text{CH}_2\text{F}_2$  with increased average errors of 0.44% for  $\text{CO}_2\text{--CO}_2$ , 0.47% for  $\text{CO}_2\text{--CH}_2\text{F}_2$  and 0.67% for  $\text{CH}_2\text{F}_2\text{--CH}_2\text{F}_2$ . At  $\alpha = 0.075$  the greatest variance in the non-polar system is  $\sigma^2 = 0.2$  for  $\text{CH}_4\text{--CH}_4$ , and for the polar system the maximum variance is  $\sigma^2 = 2.94$  for  $\text{CH}_2\text{F}_2\text{--CH}_2\text{F}_2$ .

A quarter box cut-off gives greater errors in the Coulombic energies, but follows the same trends as the half box cut-off. For the non-polar system the optimum value for  $\alpha$  is 0.05, which is seen for all pair interactions of methane and carbon dioxide.

The average error increases to 0.5% for  $\text{CO}_2\text{--CO}_2$  and to  $\sim 0.75\%$  for energies involving interactions with methane. The errors vary more when used with the polar system over a wide range of  $\alpha$ , but at  $\alpha = 0.075$  the error for all interactions is below 2%. The error increases sharply for the polar system with varying  $\alpha$ . For interactions between  $\text{CH}_2\text{F}_2\text{--CH}_2\text{F}_2$  the variance is higher than for methane, which can be attributed to the high polarity of both substituents.

### 3.3 Pure carbon dioxide

Diffusion coefficients of  $\text{scCO}_2$  calculated from MD simulations using the Ewald sum<sup>50</sup> have been previously compared with experimental results.<sup>51</sup> Our simulations show that the Wolf method gives diffusion coefficients comparable to that of PPPM (Fig. 3b). Calculations were concurrently run on the CPU using the Wolf method, and the energies were within three decimal places. Simulations were also performed at 323.2 K, which show good agreement between PPPM and the Wolf method.<sup>†</sup> Low density boxes using the Wolf method show the best agreement, but all the densities investigated are within the bounds of error of PPPM. Pressures obtained over a range of densities coincide well with an equation of state,<sup>52</sup> for both the Wolf and PPPM implementations (Fig. 3a). Both electrostatic methods

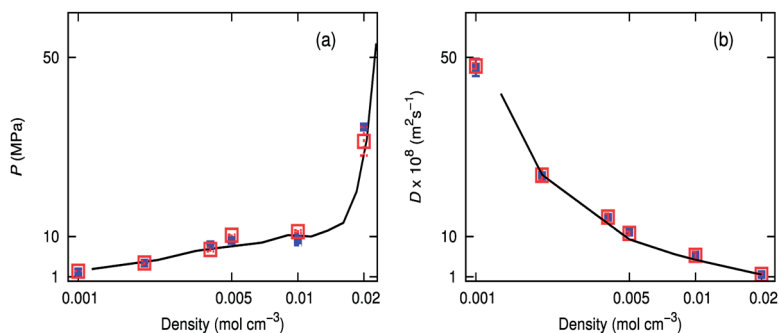


Fig. 3 (a) Diffusion coefficients for pure carbon dioxide obtained for PPPM (empty red squares) and Wolf (filled blue squares), compared with experimental (black line) at 308.2 K. (b) PVT relationship of pure carbon dioxide simulated using PPPM (empty red squares) or the Wolf method (filled blue squares) compared with experimental (black line) at 308.2 K.



capture the PVT relationship properties at high and low densities, for both temperatures studied.

### 3.4 Binary mixture of carbon dioxide with difluoromethane or methane

The Coulombic energy for the polar systems is compared for different mole fractions of difluoromethane using the Wolf method and PPPM. As the system becomes more polar, the error of the Wolf method with respect to the PPPM value becomes greater. Fig. 4 shows the total PPPM Coulombic energy, and results for  $\alpha = 0.2$  and  $\alpha = 0.075$ . For the system containing 100 difluoromethane molecules, the error is within 0.2% and 1.8% for  $\alpha = 0.075$  and  $\alpha = 0.2$  respectively. As the composition tends towards a 1 : 1 ratio, the errors increase, which indicates, as anticipated, that the Wolf approach is not suitable when the system becomes highly polar. The errors for  $\alpha = 0.075$  are reasonable (0.9%) but for  $\alpha = 0.2$  the errors are 3%. We observe better agreement for methane. For the 1 : 1 binary mixture, the error in  $\alpha = 0.2$  is 1.8% and  $\alpha = 0.075$  is 0.15%. The total Coulombic energy for difluoromethane is more negative than for methane, which can be explained by NPT dynamics resulting in a lower volume box and therefore closer contacts.

Experimental results from O'Hern and Martin<sup>51</sup> indicate pure carbon dioxide has a diffusion coefficient of  $5 (\times 10^8 \text{ m}^2 \text{ s}^{-1})$  at 308.2 K, 80 atmospheres. The non-polar binary mixture of  $\text{CH}_4/\text{CO}_2$  has a diffusion coefficient of  $\sim 5 (\times 10^8 \text{ m}^2 \text{ s}^{-1})$  for the inclusion of one methane molecule in 10 000 molecules of carbon dioxide

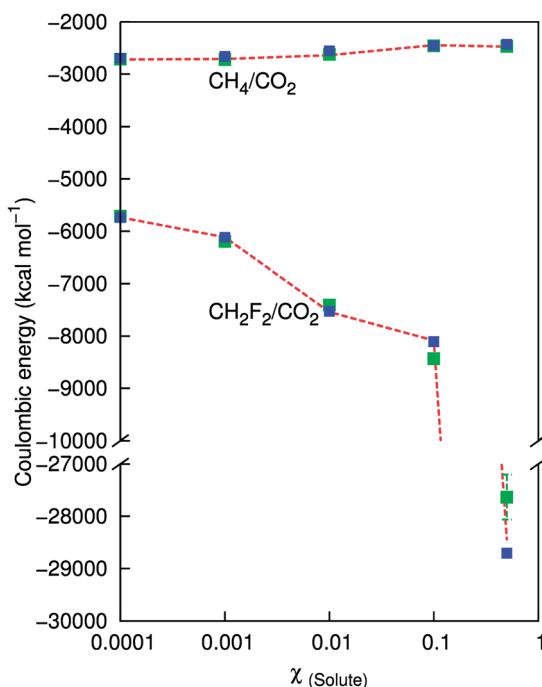


Fig. 4 Total Coulombic energy for methane and difluoromethane in carbon dioxide at 308.2 K, 80 atmospheres. The dashed red line indicates the total PPPM energy, whilst the green squares indicate  $\alpha = 0.2$  and the blue squares indicate  $\alpha = 0.075$ .



(Fig. 5). As the number of methane molecules increases, the overall diffusion coefficients remain relatively constant. Diffusion coefficients for CH<sub>4</sub> and carbon dioxide are similar, with an average increase of  $\sim 0.6 \times 10^8 \text{ m}^2 \text{ s}^{-1}$  between carbon dioxide and CH<sub>4</sub>. As  $\chi_{(\text{solute})}$  increases, the accuracy of the solute diffusion coefficients increase, due to better averaging from more solute molecules. With the exception of  $\chi_{(\text{solute})} = 0.0001$  where averaging is poor, the agreement between Wolf and PPPM is good. We observe a reduction in diffusion coefficients for difluoromethane as the polarity increases. The values decay from  $4.5 \times 10^8 \text{ m}^2 \text{ s}^{-1}$ , to  $1.3 \times 10^8 \text{ m}^2 \text{ s}^{-1}$ , which can be attributed to favourable interactions between solute and solvent. Agreement between Wolf and PPPM is good, although there is an observable increase in the error in the diffusion coefficients for difluoromethane when the fraction of solute reaches 10%. This indicates the difluoromethane interacts strongly with carbon dioxide, thus limiting diffusion.

To characterise the interactions between difluoromethane and carbon dioxide, we calculate the radial distribution (RDF) between the centre of mass for CH<sub>4</sub>/CO<sub>2</sub> and CH<sub>2</sub>F<sub>2</sub>/CO<sub>2</sub> (Fig. 6) and the associated residence times and coordination numbers are shown in Table 2.

The RDF shows a larger density of carbon dioxide molecules in the difluoromethane mixture in the first and second solvation shell compared to methane. The same trend was noticed in the RDF by Do *et al.*,<sup>53</sup> where the first solvation shell has a higher density for difluoromethane than for methane. The number of carbon dioxide molecules present in the first and second solvation shells is  $\sim 30\%$  higher for difluoromethane, and carbon dioxide resides about four times longer compared to methane. This indicates that carbon dioxide has a higher affinity for difluoromethane.

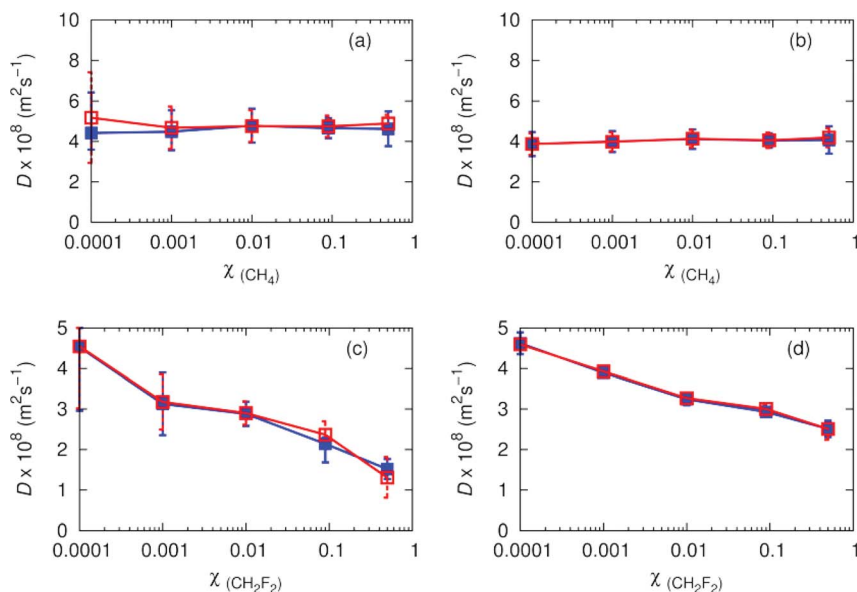


Fig. 5 Diffusion coefficients for (a) CH<sub>4</sub> in CO<sub>2</sub>/CH<sub>4</sub> (b) CO<sub>2</sub> in CO<sub>2</sub>/CH<sub>4</sub> (c) CH<sub>2</sub>F<sub>2</sub> in CO<sub>2</sub>/CH<sub>2</sub>F<sub>2</sub> and (d) CO<sub>2</sub> in CO<sub>2</sub>/CH<sub>2</sub>F<sub>2</sub> at 308.2 K, 80 atmospheres. Filled blue squares and error bars indicate diffusion coefficients obtained using PPPM, and empty red squares and error bars indicate diffusion coefficients obtained using the Wolf summation method.



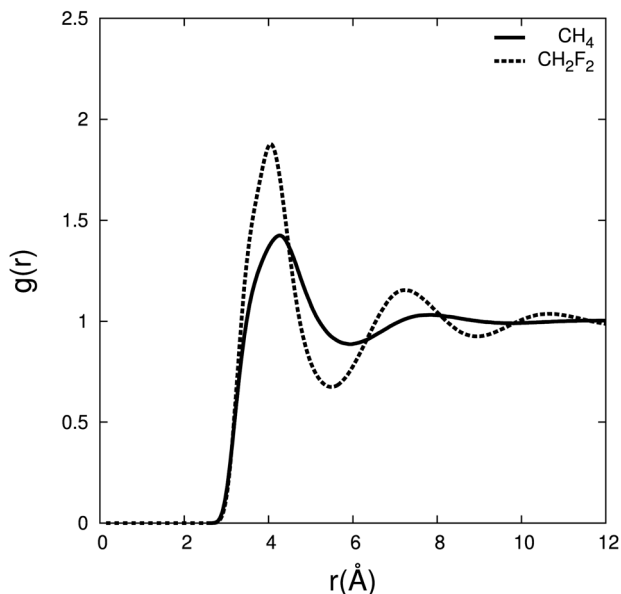


Fig. 6 Radial distribution function between the centre of mass of carbon dioxide and CH<sub>4</sub> (solid line), and CH<sub>2</sub>F<sub>2</sub> (dashed line) for  $\chi_{\text{(solute)}} = 0.1$  at 308.2 K, 80 atmospheres.

Table 2 Residence times and coordination for the centre of mass of carbon dioxide in CH<sub>4</sub> and CH<sub>2</sub>F<sub>2</sub> for  $\chi_{\text{(solute)}} = 0.1$  at 308.2 K, 80 atmospheres

| Shell                             | Distance limit (Å) | Residence time – Wolf (ps) | Residence time – PPPM (ps) | Coordination number – Wolf | Coordination number – PPPM |
|-----------------------------------|--------------------|----------------------------|----------------------------|----------------------------|----------------------------|
| <b>Methane</b>                    |                    |                            |                            |                            |                            |
| 1 <sup>st</sup>                   | 0.00–5.95          | 1.2                        | 1.2                        | 8.6                        | 8.6                        |
| 1 <sup>st</sup> + 2 <sup>nd</sup> | 0.00–9.60          | 2.6                        | 2.5                        | 37.2                       | 37.1                       |
| 2 <sup>nd</sup>                   | 5.95–9.60          | 1.3                        | 1.3                        | 28.6                       | 28.5                       |
| <b>Difluoromethane</b>            |                    |                            |                            |                            |                            |
| 1 <sup>st</sup>                   | 0.00–5.65          | 4.6                        | 4.6                        | 12.1                       | 12.1                       |
| 1 <sup>st</sup> + 2 <sup>nd</sup> | 0.00–9.24          | 10.5                       | 10.3                       | 55.6                       | 55.6                       |
| 2 <sup>nd</sup>                   | 5.65–9.24          | 5.9                        | 5.7                        | 45.5                       | 45.5                       |

## 4. Conclusions and discussion

The Wolf method shows good agreement with PPPM when modelling the electrostatic interactions of scCO<sub>2</sub> on GPUs for non-polar systems, whilst being approximately twice as fast. The choice of  $\alpha$  is important, and may need to be investigated on a case by case basis to enable satisfactory agreement. For modelling carbon dioxide in the supercritical region it is advisable to use a half-box cut-off and a low value for  $\alpha$ . In this investigation, all values of  $\alpha$  less than 0.15 produced errors less than 2% for non-polar interactions, whilst polar interactions require  $\alpha$  to be less than 0.1. Upon increasing the polarity of the system, the



potential begins to degrade. Errors of the Wolf method with respect to PPPM are approximately 0.2% when considering a 100 : 1 mixture of carbon dioxide and difluoromethane with  $\alpha = 0.075$ . We observe a strong affinity of carbon dioxide to difluoromethane compared to methane, which can be seen by a decline in diffusion coefficients with increasing mole fraction of solute. Carbon dioxide resides about four times longer in the solvation sphere of difluoromethane compared to methane.

We can conclude that the significance of using the Wolf method on GPUs allows simulations to reach timescales twice as long as those run with PPPM, without significant loss in accuracy for a carefully chosen value of  $\alpha$  for non-polar and mildly polar systems. We aim to follow up the investigation with further analysis of solvent–solute interactions and the study of fluorinated polymers, which have high solubilities in  $\text{scCO}_2$ . We will be investigating the free energy changes of fluorinated polymers, with an aim of further understanding the high affinity of fluorinated polymers for carbon dioxide. Many free energy methods require long timescales in order to reach convergence; utilizing the Wolf method for this purpose will help achieve this goal.

## Acknowledgements

This work was supported by the EPSRC, grants EP/I006559/1 and EP/K000128/1. We are grateful for access to the University of Nottingham High Performance Computing facility and MidPlus High Performance Computing facility. We appreciate useful discussions with Professor Paul Kelly and his co-workers (Imperial College) and Dr Chris Skylaris (University of Southampton). We thank Drs Richard Wheatley and Hainam Do for their critical reading of the manuscript.

## References

- 1 J. E. Stone, J. C. Phillips, P. L. Freddolino, D. J. Hardy, L. G. Trabuco and K. Schulten, *J. Comput. Chem.*, 2007, **28**, 2618.
- 2 I. Buch, M. J. Harvey, T. Giorgino, D. P. Anderson and G. De Fabritiis, *J. Chem. Inf. Model.*, 2010, **50**, 397.
- 3 J. A. Baker and J. D. Hirst, *Mol. Inf.*, 2011, **30**, 498.
- 4 M. J. Harvey and G. De Fabritiis, *Wiley Interdiscip. Rev.: Comput. Mol. Sci.*, 2012, **2**, 734.
- 5 M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids*, Oxford University Press, 1989.
- 6 W. F. van Gunsteren and H. J. C. Berendsen, *Angew. Chem., Int. Ed. Engl.*, 1990, **29**, 992.
- 7 M. Bergdorf, C. Peter and P. H. Hünenberger, *J. Chem. Phys.*, 2003, **119**, 9129.
- 8 B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan and M. Karplus, *J. Comput. Chem.*, 1983, **4**, 187.
- 9 P. P. Ewald, *Ann. Phys.*, 1921, **369**, 253–287.
- 10 R. Hockney and J. Eastwood, *Computer Simulation Using Particles*, CRC Press, 1981.
- 11 T. Darden, D. York and L. Pedersen, *J. Chem. Phys.*, 1993, **98**, 10089.
- 12 U. Essmann, L. Perera, M. L. Berkowitz, T. Darden, H. Lee and L. G. Pedersen, *J. Chem. Phys.*, 1995, **103**, 8577.



- 13 M. J. Harvey and G. De Fabritiis, *J. Chem. Theory Comput.*, 2009, **5**, 2371.
- 14 W. M. Brown, A. Kohlmeier, S. J. Plimpton and A. N. Tharrington, *Comput. Phys. Commun.*, 2012, **183**, 449.
- 15 D. J. Hardy, J. E. Stone and K. Schulten, *Parallel Comput.*, 2009, **35**, 164.
- 16 D. Wolf, P. Koblinski, S. R. Phillpot and J. Eggebrecht, *J. Chem. Phys.*, 1999, **110**, 8254.
- 17 C. J. Fennell and J. D. Gezelter, *J. Chem. Phys.*, 2006, **124**, 234104.
- 18 J. S. Hansen, T. B. Schröder and J. C. Dyre, *J. Phys. Chem. B*, 2012, **116**, 5738.
- 19 T. D. Nguyen, J.-M. Y. Carrillo, A. V. Dobrynin and W. M. Brown, *J. Chem. Theory Comput.*, 2013, **9**, 73.
- 20 H. Peker, M. P. Srinivasan, J. M. Smith and B. J. McCoy, *AIChE J.*, 1992, **38**, 761.
- 21 Z. Li and C. K. Hall, *Langmuir*, 2005, **21**, 7579.
- 22 J. Jennings, M. Beija, A. P. Richez, S. D. Cooper, P. E. Mignot, K. J. Thurecht, K. S. Jack and S. M. Howdle, *J. Am. Chem. Soc.*, 2012, **134**, 4772.
- 23 H. Lee, E. Terry, M. Zong, N. Arrowsmith, S. Perrier, K. J. Thurecht and S. M. Howdle, *J. Am. Chem. Soc.*, 2008, **130**, 12242.
- 24 Z. Wimmer and M. Zarevúcka, *Int. J. Mol. Sci.*, 2010, **11**, 233.
- 25 A. Gießauf and T. Gamse, *J. Mol. Catal. B: Enzym.*, 2000, **9**, 57.
- 26 R. L. Silveira, J. Martínez, M. S. Skaf and L. Martínez, *J. Phys. Chem. B*, 2012, **116**, 5671.
- 27 H.-L. Liu, W.-C. Hsieh and H.-S. Liu, *Biotechnol. Prog.*, 2004, **20**, 930.
- 28 W. Leitner, *Acc. Chem. Res.*, 2002, **35**, 746.
- 29 D. J. Heldebrant and P. G. Jessop, *J. Am. Chem. Soc.*, 2003, **125**, 5600.
- 30 D. M. D'Alessandro, B. Smit and J. R. Long, *Angew. Chem., Int. Ed.*, 2010, **49**, 6058.
- 31 Z. Su and M. Maroncelli, *J. Chem. Phys.*, 2006, **124**, 164506.
- 32 J. M. Desimone, Z. Guan and C. S. Elsbernd, *Science*, 1992, **257**, 945.
- 33 S. P. Nalawade, F. Picchioni and L. P. B. M. Janssen, *Prog. Polym. Sci.*, 2006, **31**, 19.
- 34 A. Dardin, J. M. DeSimone and E. T. Samulski, *J. Phys. Chem. B*, 1998, **102**, 1775.
- 35 A. Sadeghifar, M. Dadvar, S. Karimi and A. F. Ghobadi, *J. Mol. Graphics Modell.*, 2012, **38**, 455.
- 36 A. Angoshtari and A. Yavari, *Phys. Lett. A*, 2011, **375**, 1281.
- 37 S. Plimpton, *J. Comput. Phys.*, 1995, **117**, 1.
- 38 A. D. MacKerell, D. Bashford, R. L. Dunbrack, J. D. Evanseck, M. J. Field, S. Fischer, J. Gao, H. Guo, S. Ha, D. Joseph-McCarthy, L. Kuchnir, K. Kuczera, F. T. K. Lau, C. Mattos, S. Michnick, T. Ngo, D. T. Nguyen, B. Prodhom, W. E. Reiher, B. Roux, M. Schlenkrich, J. C. Smith, R. Stote, J. Straub, M. Watanabe, J. Wiórkiewicz-Kuczera, D. Yin and M. Karplus, *J. Phys. Chem. B*, 1998, **102**, 3586.
- 39 J. G. Harris and K. H. Yung, *J. Phys. Chem.*, 1995, **99**, 12021.
- 40 B. J. Palmer and J. L. Anchell, *J. Phys. Chem.*, 1995, **99**, 12239.
- 41 J. Kolafa and J. W. Perram, *Mol. Simul.*, 1992, **9**, 351.
- 42 M. Deserno and C. Holm, *J. Chem. Phys.*, 1998, **109**, 7694.
- 43 W. Hoover, *Phys. Rev. A: At., Mol., Opt. Phys.*, 1985, **31**, 1695.
- 44 NVIDIA, *CUDA Dynamic Parallelism Programming Guide*, 2012.
- 45 W. C. Swope, *J. Chem. Phys.*, 1982, **76**, 637.



- 46 H. J. C. Berendsen, J. P. M. Postma, W. F. van Gunsteren, A. DiNola and J. R. Haak, *J. Chem. Phys.*, 1984, **81**, 3684.
- 47 X. Michalet, *Phys. Rev. E: Stat., Nonlinear, Soft Matter Phys.*, 2010, **82**, 041914.
- 48 Y. Iwai, H. Higashi, H. Uchida and Y. Arai, *Fluid Phase Equilib.*, 1997, **127**, 251.
- 49 S. Nosé and M. L. Klein, *Mol. Phys.*, 1983, **50**, 1055.
- 50 H. Higashi and K. Tamura, *Mol. Simul.*, 2010, **36**, 772.
- 51 H. A. O'Hern and J. J. Martin, *Ind. Eng. Chem.*, 1955, **47**, 2081.
- 52 R. Span and W. Wagner, *J. Phys. Chem. Ref. Data*, 1996, **25**, 1509.
- 53 H. Do, R. J. Wheatley and J. D. Hirst, *J. Phys. Chem. B*, 2010, **114**, 3879.

