ADVANCED CBMC TECHNIQUES

THIJS J.H. VLUGT AND BEREND SMIT
Department of Chemical Engineering, Nieuwe Achtergracht 166,
1018 GB Amsterdam, The Netherlands
E-mail: {tampert, smit}@its.chem.uva.nl

Configurational Bias Monte Carlo (CBMC) simulations are often used to compute thermo-dynamic properties of flexible chain molecules. In particular for long chains, these calculations are still computationally expensive. In this article we will present several improved algorithms for CBMC and illustrate how this method can be implemented efficiently on a parallel computer.

1 Introduction

Configurational Bias Monte Carlo (CBMC) [1,2,3,4] is an advanced simulation technique for complex molecules. In CBMC, a chain is grown segment by segment using several trial positions for each segment. This introduces a bias, which can be removed exactly by modifying the acceptance/rejection rules. CBMC simulations are often used for the calculation of vapor-liquid equilibria of linear alkanes [5,6,7], branched alkanes [8,9,10], and for the simulation of adsorption of alkanes in porous structures [11,12,13,14,15,16,17]. These simulations are still computationally expensive. Therefore, we have developed some improved algorithms to speedup the calculations:

- The Dual Cut-Off CBMC algorithm (DC-CBMC) by Vlugt et al. [18]. This algorithm uses an additional bias in the selection of trial segments. It can be shown that hard-core (repulsive) interactions are more important than long-range interactions in the selection of trial segments. Therefore, one can use a second potential cut-off radius ($r_{\text{cut}}\star$) for the selection of trial segments. The second cut-off radius can be chosen quite small, for example for a Lennard-Jones (LJ) fluid one can use $r_{\text{cut}}\star = \sigma$. This saves a lot of CPU time, because short-range interactions can be calculated efficiently using a linked-cell list. The use of a second cut-off radius introduces an additional bias in the CBMC algorithm, which can be removed exactly in the acceptance/rejection rule.

- Parallel CBMC [19,20]. We have constructed a parallel CBMC algorithm based on the DC-CBMC algorithm by Vlugt et al. and the parallel CBMC algorithm by Esselink et al. [20]. In this algorithm, multiple chains are grown in each CBMC trial move to increase the acceptance probability. Important to note is that all simulations in the article of Esselink et al. were performed on a single workstation and not on a parallel computer, although the title of that article suggests otherwise.

- Generation of trial positions. In CBMC, trial positions have to be generated according to the Boltzmann factor of the internal energy. It turns out that in some cases special tricks are needed, especially for branched alkanes.

vlu002

To explain the improved algorithms, the basics of CBMC are summarized below. A more detailed discussion can be found in Ref. [4]. In the CBMC scheme it is convenient to split the total potential energy of a trial site into two parts. The first part is the internal, bonded, intra-molecular potential (u^{internal}) which is used for the generation of trial orientations. For alkanes, the internal potential often has the form [21,22]

$$u^{\text{internal}} = \sum u^{\text{stretch}}(l) + \sum u^{\text{bend}}(\theta) + \sum u^{\text{tors}}(\phi) \tag{1}$$

$$u^{\text{stretch}}(l) = \frac{1}{2}k_l[l - l_0]^2 \tag{2}$$

$$u^{\text{bend}}(\theta) = \frac{1}{2}k_\theta[\theta - \theta_0]^2 \tag{3}$$

$$u^{\text{tors}}(\phi) = \sum_{i=0}^{i=5} C_i \cos^i(\phi) \tag{4}$$

where $u^{\text{stretch}}(l)$ is the bond-stretching energy, $u^{\text{bend}}(\theta)$ is the bond-bending energy, and $u^{\text{tors}}(\phi)$ is the torsion energy. The second part of the potential, the external potential (u^{ext}), is used to bias the selection of a site from the set of trial sites.

To generate a new configuration, a randomly chosen molecule is regrown segment by segment. We will use the symbol l for the length of the chain, n for a new configuration and o for an old configuration. If the entire molecule is being regrown then f trial sites for the first bead are placed at random positions in the simulation box [20]. The Rosenbluth weight of this segment is

$$w_1(n) = \sum_{j=1}^{f} \exp\left[-\beta u_{1j}^{\text{ext}}\right] \tag{5}$$

where $\beta = 1/(k_B T)$, and one trial site (i) is selected with probability

$$P_{1i}^{\text{selecting}}(\mathbf{B_i}) = \frac{\exp\left[-\beta u_{1i}^{\text{ext}}\right]}{w_1(n)} \tag{6}$$

For the other segments a of the molecule, k trial orientations $\mathbf{B_i}$ are generated according to the Boltzmann weight of the internal potential of that segment

$$P_{ai}^{\text{generating}}(\mathbf{B_i})\,d\mathbf{B} = \frac{\exp\left[-\beta u_{ai}^{\text{internal}}\right]d\mathbf{B}}{\int \exp\left[-\beta u_{a}^{\text{internal}}\right]d\mathbf{B}} \tag{7}$$

Out of these k trial orientations one (i) is chosen according to the Boltzmann weight of its external potential

$$P_{ai}^{\text{selecting}}(\mathbf{B_i}) = \frac{\exp\left[-\beta u_{ai}^{\text{ext}}\right]}{\sum_{j=1}^{k} \exp\left[-\beta u_{aj}^{\text{ext}}\right]}. \tag{8}$$

This procedure is repeated until the entire chain has been grown. The Rosenbluth weight $W(n)$ of the new configurations is defined as

$$W(n) = \frac{w_1(n) \prod_{a \neq 1} \left[\sum_{j=1}^{k} \exp\left[-\beta u_{aj}^{\mathrm{ext}}\right] \right]}{f \times k^{l-1}} \tag{9}$$

The old configuration is retraced in a similar way, except that for each segment only $k-1$ ($f-1$ for the first bead) trial orientations are generated with a probability according to Eq. (7) (randomly for the first bead). The k-th (f-th) trial orientation is the old orientation. The Rosenbluth weight $W(o)$ of the old configuration is defined as

$$W(o) = \frac{w_1(o) \prod_{a \neq 1} \left[\sum_{j=1}^{k} \exp\left[-\beta u_{aj}^{\mathrm{ext}}\right] \right]}{f \times k^{l-1}} \tag{10}$$

where $w_1(o)$ is the Rosenbluth weight of the first segment of the old configuration. To satisfy the detailed balance condition, the new configuration is accepted with a probability

$$\mathrm{acc}(o \to n) = \min\left(1, \frac{W(n)}{W(o)}\right) \tag{11}$$

The CBMC algorithm greatly improves the conformational sampling for molecules with articulated structure and increases the efficiency of chain insertions (required for the calculation of chemical potentials, grand canonical, and Gibbs ensemble simulations) by several orders of magnitude.

2 Dual cut-off CBMC

It is possible to split the external potential that is used for the selection of trial segments into two parts

$$u^{\mathrm{ext}} = \overline{u}^{\mathrm{ext}} + \delta u^{\mathrm{ext}} \tag{12}$$

where $\overline{u}^{\mathrm{ext}}$ is a potential that is less expensive to calculate than u^{ext}, and δu^{ext} the difference between the two potentials. A useful choice for $\overline{u}^{\mathrm{ext}}$ is the potential described by a shorter cut-off radius ($r_{cut}\star$)

$$u^{\mathrm{ext}}(r) = \overline{u}^{\mathrm{ext}}(r < r_{cut}\star) + \delta u^{\mathrm{ext}}(r_{cut}\star < r < r_{cut}) \tag{13}$$

where $\overline{u}^{\mathrm{ext}}(r < r_{cut}\star)$ consists only of interactions within a distance $r_{cut}\star$. Dual Cut-off Configurational Bias Monte Carlo (DC-CBMC) uses the potential $\overline{u}^{\mathrm{ext}}(r < r_{cut}\star)$ for generation of the chain and thus calculates the Rosenbluth weight faster than CBMC. However, this would lead to an incorrect distribution if we would use the conventional acceptance rule (Eq. (11)). The correct distribution is recovered by using

$$\mathrm{acc}(o \to n) = \min\left(1, \frac{\overline{W}(n)}{\overline{W}(o)} \exp\left[-\beta\left[\delta u^{\mathrm{ext}}(n) - \delta u^{\mathrm{ext}}(o)\right]\right]\right) \tag{14}$$

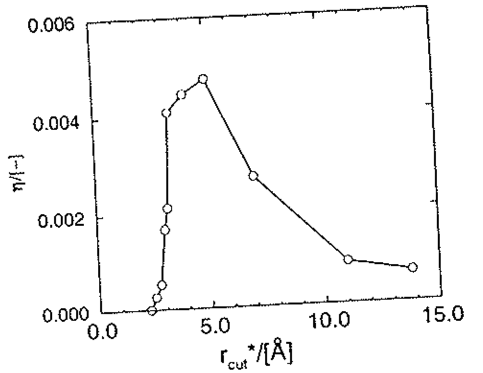

Figure 1. Efficiency (η) as a function of the second cut-off radius ($r_{\mathrm{cut}}\star$). Details about the simulation can be found in Appendix A.

DC-CBMC regrowth only requires the calculation of the full potential for the final configuration, and not for all of the trial orientations. Note that the division of the external energy can be done in any consistent fashion, and is typically used in conventional CBMC to account for LJ tail corrections and for Ewald corrections in charged systems.

In Figure 1, we have plotted for a typical simulation the efficiency (number of accepted trial moves divided by the amount of CPU time) as a function of the second cut-off radius $r_{\mathrm{cut}}\star$. The short-range part of the potential can be calculated efficiently using a cell list, see Appendix A. For $r_{\mathrm{cut}}\star = 0$, completely unbiased chains are grown, resulting in a very low efficiency. When $r_{\mathrm{cut}}\star = r_{\mathrm{cut}} = 13.8$Å, we obtain the original CBMC algorithm. This is not very efficient because for the selection of a trial segment, a lot of long-range energy terms have to be calculated. The optimum value of $r_{\mathrm{cut}}\star$ is around 5.0Å, which is slightly larger than the LJ size parameter (here: $\sigma = 3.6$Å). As in the DC-CBMC scheme the long-range energy has to be calculated only for the selected configuration and not for every trial configuration, DC-CBMC is much more efficient. For a more in-depth discussion the reader is referred to Ref. [18].

3 Parallel CBMC

In general, there are several criteria to design a successful parallel algorithm:

- There should be a good load-balance, every processor should be doing roughly the same amount of work.
- The amount of communication between processors should be minimized.

There are several reasons why the conventional CBMC algorithm cannot be parallelized efficiently:

sequential process by nature.

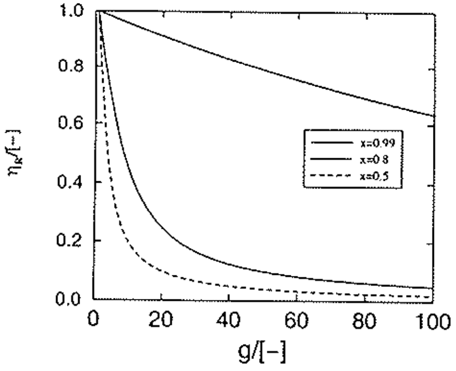

Figure 2. Relative efficiency (η_R) as a function of x and g for a hard-sphere chain according to Eq. (15).

- Because only the short-range part of the potential energy is needed to select trial configurations, the growth of a chain is computationally cheap and therefore it is not possible to parallelize this task efficiently on a large number of processors.

To avoid these problems, Esselink et al. [20] have developed a CBMC algorithm in which multiple (independent) chains are grown instead of only one chain in the conventional CBMC algorithm. Out of these chains, one is selected according to its Rosenbluth weight and the other chains are thrown away. The effect of the use of multiple chain growth is that it is more likely that one chain is grown in a favorable position, resulting in a larger fraction of accepted trial-moves. This will be most effective when the fraction of accepted trial-moves is low. When the fraction of accepted trial-moves is high, however, the use of a larger number of trial chains (g) will have a small effect only. This can be demonstrated very easily for the growth of hard-sphere chains. When the probability of generating a chain with an overlap is equal to x and the number of chains that is grown in parallel is equal to g, the relative efficiency η_R (fraction of accepted trial moves per grown chain divided by the fraction of accepted trial-moves for $g = 1$) will be equal to

$$\eta_R(g, x) = \frac{1 - x^g}{(1 - x) \star g} \tag{15}$$

This function is plotted in Figure 2. The relative efficiency will always decrease with increasing g. This decrease is less dramatic when the probability of generating a chain with an overlap is high. This means that CBMC algorithms with a moderately high acceptance rate like recoil growth [23] will have a poor parallel performance.

In order to solve this problem, it is possible to combine the Dual Cut-off scheme with the parallel CBMC scheme of Esselink et al. The total external energy u^{ext} can be split into two parts (short-range and long-range energy) according to Eq. (12). For the growth of a chain, $\overline{u}^{\mathrm{ext}}$ is used only. Repeatedly, the following steps are executed:

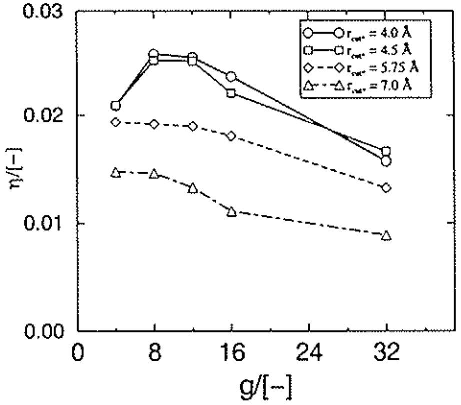

Figure 3. Efficiency (η) as a function of the number of chains (g) and the second cut-off radius ($r_{\mathrm{cut}\star}$). The number of processors was equal to 4. See Appendix A for details about the simulation.

1. Among Q processors, g new (n) chains of length l are divided and grown using the standard CBMC algorithm. Note that to avoid load imbalance, $\mathrm{mod}\,(g, Q) = 0$ must hold. For the selection of trial segments, $\overline{u}^{\mathrm{ext}}$ is used only. This results in a Rosenbluth factor $\overline{W}$ for each chain, see Eq. (9).
2. Out of the g chains, one chain (i) is selected according to
$$p_i = \frac{\overline{W}_i}{\sum_{j=1}^{j=g} \overline{W}_j} \tag{16}$$
3. For the selected chain, δu^{ext} is calculated, resulting in
$$Z(n) = \frac{\exp\left[-\beta \delta u^{\mathrm{ext}}\right] \sum_{j=1}^{j=g} \overline{W}_j}{g} \tag{17}$$
4. For the old (o) configuration, a similar procedure is used to calculate $Z(o)$. Note that the first chain of the g chains is actually the old chain that is retraced using standard CBMC retracing. The remaining $g - 1$ chains are new chains.
5. This trial-move is accepted with a probability
$$\mathrm{acc}\,(o \to n) = \min\left(1, \frac{Z(n)}{Z(o)}\right) \tag{18}$$

In Ref. [19] it is shown that this scheme obeys detailed balance. An important quantity is the amount of CPU time for the growth of one chain divided by the total amount of CPU time that is spent in the calculation of δu^{ext}. When this ratio is low, one can grow multiple chains at almost no computational cost (because most time is spent on calculating δu^{ext}). This means that for increasing g the efficiency of the algorithm will increase instead of decrease (Figure 2). This is for example the

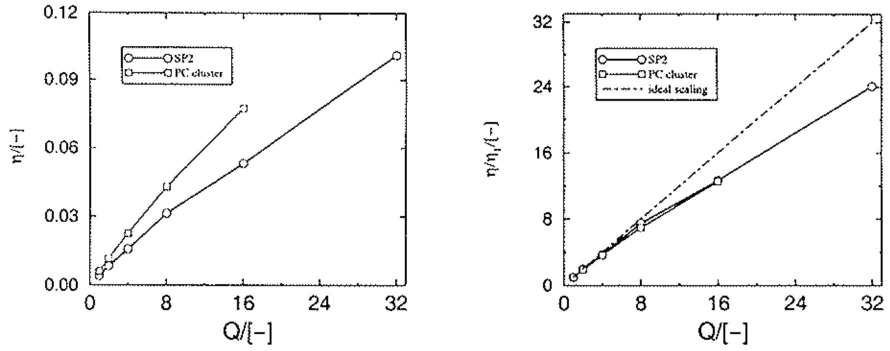

Figure 4. Left: Efficiency (η) as a function of the number of processors (Q) for $r_{\text{cut}}\star = 4.5\mathring{A}$ and $g = 32$. Simulations were performed on a IBM SP2 and a PC cluster. See Appendix B for details about these machines. Right: Same, but the efficiency is divided by the efficiency of a simulation on one processor.

case with the use of a small value for $r_{\text{cut}}\star$, see Figure 3. In this case, the efficiency increases for increasing g. This means that growing multiple chains can increase the overall performance of the MC algorithm, even on a single workstation.

To investigate the scaling of the algorithm with the number of processors, we have performed a simulation with $g = 32$ on a PC cluster and a IBM SP2, see Figure 4. See Appendix B for details on these machines. On both machines, the algorithm scales ideal for $Q \leq 8$. For $Q = 16$ and $Q = 32$, there is a small deviation of ideal scaling due to communication overhead.

4 Generation of trial segments

In this section we will discuss how to generate trial positions according to the Boltzmann factor of the internal potential.

Let us consider the growing of a structure $\mathbf{x} - \mathbf{y} - (\mathbf{b_1}, \mathbf{b_2})$. In this structure there are three bonds, $\mathbf{x} - \mathbf{y}$, $\mathbf{y} - \mathbf{b_1}$ and $\mathbf{y} - \mathbf{b_2}$ (with bond-stretching potentials according to Eq. (2)) and three bond-angles $\mathbf{x} - \mathbf{y} - \mathbf{b_1}$, $\mathbf{x} - \mathbf{y} - \mathbf{b_2}$ and $\mathbf{b_1} - \mathbf{y} - \mathbf{b_2}$ (with bond-bending potentials according to Eq. (3)). This structure corresponds to a united atom model of isobutane, which is the simplest branched alkane [17].

Assume that we have already inserted the first two segments $\mathbf{x}, \mathbf{y}$ using the conventional growing schemes [4]. We now have to generate the position of a trial set $\mathbf{B} = (\mathbf{b_1}, \mathbf{b_2})$ where $\mathbf{b_1}$ and $\mathbf{b_2}$ are the trial positions of the two atoms that are connected to the branched atom ($\mathbf{y}$). In the CBMC scheme the probability of this set is proportional to its Boltzmann weight [4],

$$p(\mathbf{B})d\mathbf{B} \propto \exp\left[-\beta\left[u_{\text{bend}}(\mathbf{B}) + u_{\text{stretch}}(\mathbf{B})\right]\right]d\mathbf{B} \tag{19}$$

in which u_{bend} is the total bond-bending energy:

$$u_{\text{bend}}(\mathbf{B}) = u_{\text{bend}}(\mathbf{x}, \mathbf{y}, \mathbf{b_1}) + u_{\text{bend}}(\mathbf{x}, \mathbf{y}, \mathbf{b_2}) + u_{\text{bend}}(\mathbf{b_1}, \mathbf{y}, \mathbf{b_2}). \tag{20}$$

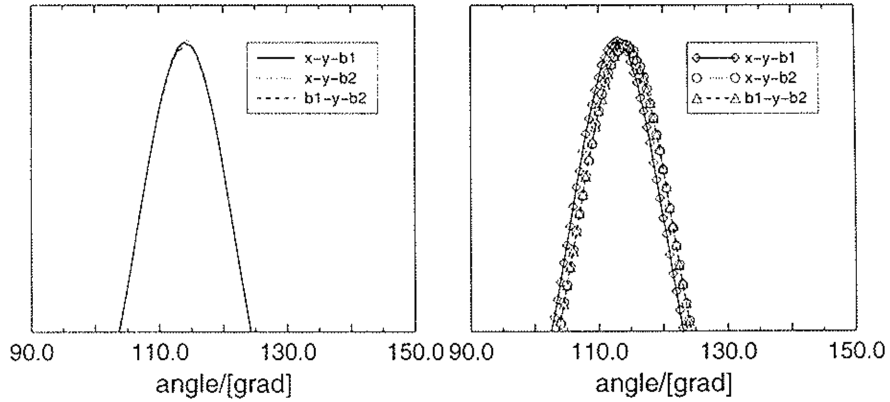

Figure 5. Part of the bond angle distributions of isobutane at T=1000 K ($k_\theta/k_B = 62500K$); (a) results of our algorithm and (b) the incorrect algorithm when the two beads are not inserted simultaneously ($\mathbf{b_1}$ always inserted before $\mathbf{b_2}$). Because of symmetry reasons, all angle distributions of isobutane should be equal. Note that the differences are small but significant.

and u_{stretch} is the total bond-stretching energy

$$u_{\text{stretch}}(\mathbf{B}) = u_{\text{stretch}}(\mathbf{y}, \mathbf{b_1}) + u_{\text{stretch}}(\mathbf{y}, \mathbf{b_2}) \tag{21}$$

It is possible to write $d\mathbf{B}$ in spherical coordinates

$$d\mathbf{B} \propto l^2 \sin(\theta)\, dl d\theta d\phi \tag{22}$$

It is easy to see that due the used harmonic bonded potentials (Eq. (2) and Eq. (3)), the bond-lengths $l_{\mathbf{b}_1}$ and $l_{\mathbf{b}_2}$ are independent of $\theta_{\mathbf{b}_1}$, $\theta_{\mathbf{b}_2}$, $\phi_{\mathbf{b}_1}$ and $\phi_{\mathbf{b}_2}$. This means that we can generate the bond-lengths independent of the other spherical coordinates. However, due to the presence of a bond-bending potential involving $\theta_{\mathbf{b}_1,\mathbf{y},\mathbf{b}_2}$, we cannot generate $\mathbf{b_1}$ and $\mathbf{b_2}$ independent from each other [17]. This means that CBMC schemes that do not insert all segments at a branch simultaneously generate incorrect distributions due to the presence of dependent bond-bending potentials [8,9,10,18], see Figure 5. A possible way to overcome this problem is to generate random vectors on a sphere simultaneously for all beads connected to a branch and to accept or to reject this configuration using the conventional acceptance/rejection rule [4,24,25]. A problem is that when the distribution $p(\mathbf{B})$ is small (for example at low temperatures, a large value of the bond-bending constants or many beads that are connected to a branch) this scheme becomes computationally expensive. A possible way overcome this problem is to generate $\mathbf{b_1}, \mathbf{b_2}$ using a separate MC simulation. In such a simulation, there are two possible trial moves to change either $\phi_{\mathbf{b}_i}$ or $\theta_{\mathbf{b}_i}$ (i is chosen at random either 1 or 2) with a random displacement. The acceptance/rejection rules of these trial-moves are respectively

$$\text{acc}(o \rightarrow n) = \exp\left[-\beta\left[u(n) - u(o)\right]\right] \tag{23}$$

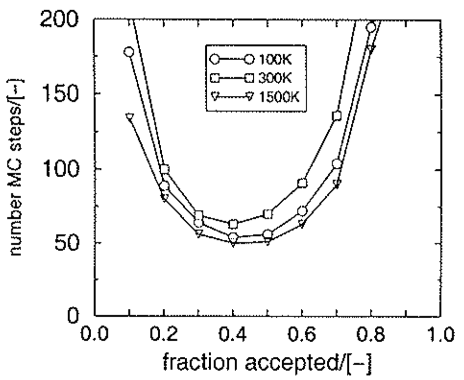

Figure 6. Number of MC steps to obtain independent angles for a single isobutane molecule as a function of the fraction of accepted trial moves for various temperatures ($k_\theta / k_B = 62500K$). In order to obtain the highest efficiency, one should adjust all maximum angle rotations to an acceptance probability of 0.4.

and

$$\operatorname{acc}(o \to n) = \frac{\sin(\theta(n))}{\sin(\theta(o))} \exp\left[-\beta\left[u(n) - u(o)\right]\right] \tag{24}$$

It is possible to start such a MC simulation from a previous conformation. In that case, one has to make sure that the final configuration is independent of the starting configuration. In Figure 6, we have plotted the number of required MC steps to obtain an independent configuration as a function of the fraction of accepted trial moves. It turns out that an acceptance rate of 40% is most efficient. In the case that more complex potentials like torsion potentials or potentials in which bond-stretching and bond-bending cannot be separated are present, it may even become impossible to generate trial segments directly according to the bonded intra-molecular potential. A way to correct for this is to generate trial segments with a distribution that is close to the correct distribution and correct for this difference afterwards (see also Eq. (14)). One would like to avoid the situation in which the (computationally more expensive) external energy has to be calculated for a trial configuration with a very high internal energy, because such configurations are usually not accepted. A possible way to avoid this problem is to use CBMC on the generation of a trial segment as well. In such a scheme, many (computationally inexpensive) configurations are created at random and out of these random configurations, several configurations are selected according to their Boltzmann weight of their internal bonded potential. Only for these selected configurations, the external energy has to be calculated. This will eventually lead to an acceptance rule with two different Rosenbluth factors: one for internal and one for external interactions.

Concluding remarks

We have discussed several improvements of CBMC and demonstrated how to implement CBMC efficiently on a parallel computer. It turns out that the generation of a branched molecule is slightly more subtle than a linear molecule.

Acknowledgments

Financial support provided from NWO-CW (Nederlandse Organisatie voor Wetenschappelijk Onderzoek, Chemische Wetenschappen) is acknowledged. A large part of the computer resources were generously provided by SARA (Stichting Academisch Rekencentrum Amsterdam) and EPCC (Edinburgh Parallel Computing Centre).

Appendix A: Details of the model

To test the efficiency of our algorithms, simulations were performed for the regrowth of one n-hexane (C_6) molecule in the zeolite Silicalite. The zeolite was modeled as a rigid crystal. The number of units cells was chosen $2 \times 2 \times 4$, resulting in a system size of $40.044 Å \times 39.798 Å \times 53.532 Å$. For hexane, a united atom model was used. All alkane-alkane and alkane-zeolite intermolecular interactions are described with a LJ potential with a cut-off radius of 13.8Å. In the Dual Cut-Off scheme, all particles including the zeolite particles were placed on a 3D grid with a grid-size at least equal to the second cut-off radius. This grid has to be recalculated only after an accepted trial-move. Intra-molecular interactions include a bond-stretching potential for two bonded atoms, a bond-bending potential for three successive atoms, a torsion potential for four successive atoms and a LJ potential potential for atoms that are separated by more than three bonds. The number of trial orientations was chosen to be 15 for the first bead and 10 for the remaining beads. In addition to regrow trial moves, a very small amount of particle displacement and particle rotation trial moves was used. Further details of the forcefield and the system can be found in Refs. [14,17,18].

Appendix B: Details of the parallel computers used

We have used the following parallel computers to test the efficiency of our algorithm:

- IBM RS/6000 SP (isis.sp.sara.nl) with 72 nodes (160 MHz) and 512 MB distributed memory per node. MPI-F was used as a communication library.
- Cluster (42 nodes) of PC's (bombay.chem.uva.nl) with Intel PentiumII 350 MHz processors and 128 MB per node running Redhat Linux 5.2 [26]. LAM 6.1 [27] was used as a communication library, the switches [−O − c2c − nger] were used for fast communication. The PC's were connected using a 100 Mbit Ethernet network and a 3COM switch. More details about this cluster can be found on the web [28].

References

1. J.I. Siepmann and D. Frenkel. Configurational-bias Monte Carlo: A new sampling scheme for flexible chains. *Mol. Phys.*, 75:59–70, 1992.
2. D. Frenkel, G.C.A.M. Mooij, and B. Smit. Novel scheme to study structural and thermal properties of continuously deformable molecules. *J. Phys.: Condens. Matter*, 4:3053–3076, 1992.
3. J.J. de Pablo, M. Laso, and U.W. Suter. Estimation of the chemical potential of chain molecules by simulation. *J. Chem. Phys.*, 96:6157–6162, 1992.
4. D. Frenkel and B. Smit. *Understanding Molecular Simulations: from Algorithms to Applications*. Academic Press, San Diego, 1996.
5. J.I. Siepmann, S. Karaborni, and B. Smit. Simulating the critical properties of complex fluids. *Nature*, 365:330–332, 1993.
6. B. Smit, S. Karaborni, and J.I. Siepmann. Computer simulations of vapour-liquid phase equilibria of n-alkanes. *J. Chem. Phys.*, 102:2126–2140, 1995. Erratum: J. Chem. Phys. **109**, 352 (1998).
7. M.G. Martin and J.I. Siepmann. Transferable potentials for phase equilibria (TraPPe): I. united-atom description of n-alkanes. *J. Phys. Chem. B*, 102:2569–2577, 1998.
8. J.I. Siepmann, M.G. Martin, C.J. Mundy, and M.L. Klein. Intermolecular potentials for branched alkanes and the vapour liquid equilibria of n-heptane, 2-methylhexane, and 3-ethylpentane. *Mol. Phys.*, 90:687–693, 1997.
9. N.D. Zhuravlev and J.I. Siepmann. Exploration of the vapour-liquid phase equilibria and critical points of triacontane isomers. *Fluid Phase Equilibria*, 134:55–61, 1997.
10. S.T. Cui, P.T. Cummings, and H.D. Cochran. Configurational bias Gibbs ensemble simulation of vapour-liquid equilibria of linear and short-branched alkanes. *Fluid Phase Equilibria*, 141:45–61, 1997.
11. B. Smit and T.L.M. Maesen. Commensurate "freezing" of alkanes in the channels of a zeolite. *Nature*, 374:42–44, 1995.
12. E.J. Maginn, A.T. Bell, and D.N. Theodorou. Sorption thermodynamics, siting and conformation of long n-alkanes in silicalite as predicted by configurational-bias Monte Carlo integration. *J. Phys. Chem.*, 99:2057–2079, 1995.
13. S. Bandyopadhyay and S. Yashonath. Conformational analysis of n-butane in zeolite NaCaA: temperature and concentration dependence. *J. Phys. Chem. B*, 101:5675–5683, 1997.
14. T.J.H. Vlugt, W. Zhu, F. Kapteijn, J.A. Moulijn, B. Smit, and R. Krishna. Adsorption of linear and branched alkanes in the zeolite silicalite-1. *J. Am. Chem. Soc.*, 120:5599–5600, 1998.
15. R. Krishna, B. Smit, and T.J.H. Vlugt. Sorption-induced diffusion-selective separation of hydrocarbon isomers using silicalite. *J. Phys. Chem. A*, 102:7727–7730, 1998.
16. T. Maris, T.J.H. Vlugt, and B. Smit. Simulation of alkane adsorption in the aluminophosphate molecular sieve $AlPO_4-5$. *J. Phys. Chem. B*, 102:7183–7189, 1998.
17. T.J.H. Vlugt, R. Krishna, and B. Smit. Molecular simulations of adsorption isotherms for linear and branched alkanes and their mixtures in silicalite. *J. Phys. Chem. B*, 103:1102–1118, 1999.

18. T.J.H. Vlugt, M.G. Martin, B. Smit, J.I. Siepmann, and R. Krishna. Improving the efficiency of the CBMC algorithm. *Mol. Phys.*, 94:727–733, 1998.
19. T.J.H. Vlugt. Efficiency of parallel CBMC simulations. *Mol. Sim.*, 1999 (Submitted).
20. K. Esselink, L.D.J.C. Loyens, and B. Smit. Parallel Monte Carlo simulations. *Phys. Rev. E*, 51:1560–1568, 1995.
21. P. Van der Ploeg and H.J.C. Berendsen. Molecular dynamics simulation of a bilayer membrane. *J. Chem. Phys.*, 76:3271–3276, 1982.
22. W.L. Jorgensen, J.D. Madura, and C.J. Swenson. Optimized intermolecular potential function for liquid hydrocarbons. *J. Am. Chem. Soc.*, 106:6638–6646, 1984.
23. S. Consta, N.B. Wilding, D. Frenkel, and Z. Alexandrowicz. Recoil growth: an efficient simulation method for multi-polymer systems. *J. Chem. Phys.*, in press, 1999.
24. M.P. Allen and D.J. Tildesley. *Computer Simulation of Liquids*. Clarendon Press, Oxford, 1987.
25. M. Dijkstra. Confined thin films of linear and branched alkanes. *J. Chem. Phys.*, 107:3277–3288, 1997.
26. http://www.redhat.com
27. http://www.lsc.nd.edu/lam
28. http://molsim.chem.uva.nl/cluster

Published by

World Scientific Publishing Co. Pte. Ltd.
P O Box 128, Farrer Road, Singapore 912805
USA office: Suite 1B, 1060 Main Street, River Edge, NJ 07661
UK office: 57 Shelton Street, Covent Garden, London WC2H 9HE

British Library Cataloguing-in-Publication Data
A catalogue record for this book is available from the British Library.

MOLECULAR DYNAMICS ON PARALLEL COMPUTERS

ISBN 981-02-4232-8

Printed in Singapore by Regal Press (S) Pte. Ltd.

PREFACE

Molecular dynamics simulation is a well-established technique for modeling complex many-particle systems in diverse areas of physics and chemistry. The computational requirements of simulations of large systems, especially when taking into account long-range interactions, are enormous. They make the use of high-performance parallel computers indispensable and have led to the development of a broad range of advanced algorithms for these machines. There is, however, still a long way to go in order to simulate realistic systems over time scales of typical physical, chemical, and biological processes. Therefore, this workshop brought together researchers from different areas of science, who develop and apply the methodology of molecular dynamics, in order to discuss recent advances and to exchange experiences and ideas. Besides the presentation of the physical background and results of the manifold simulations, special emphasis was put on the discussion of the relevant algorithms and their adaptation to massively parallel computers. The following areas were covered:

- Algorithms and Programs
- Polymers
- Biochemistry
- Solid State Physics
- Granular Materials
- Astrophysics
- Lattice Gauge Theory

The program was compiled by Norbert Attig, Rüdiger Esser, Peter Grassberger, Johannes Grotendorst, Manfred Kremer, Marius Lewerenz, and Klaus Schilling.

This was the first conference organized by the John von Neumann Institute for Computing (NIC) which was founded in 1998 by Research Centre Jülich and the German Electron Synchrotron (DESY). Similar to its predecessor, the Höchstleistungsrechenzentrum (HLRZ), NIC aims to promote high-performance scientific computing by improving the methods, tools and applications in its own Research Groups and by giving the research community access to state-of-the-art high-performance computers provided by NIC through its Central Institute for Applied Mathematics in Jülich and the Center for Parallel Computing, DESY Zeuthen.

Molecular dynamics simulation is a typical example of the interdisciplinary research that has been carried out in the framework of HLRZ and is the subject of projects within NIC as well. A considerable portion of the research presented at the workshop originates from the NIC Research Groups and from projects which have used the supercomputing facilities provided by NIC.

WORKSHOP ON

Molecular Dynamics On Parallel Computers

John von Neumann Institute for Computing (NIC)
Research Centre Jülich, Germany
8–10 February 1999

edited by

Rüdiger Esser
Peter Grassberger
Johannes Grotendorst
NIC, Research Center Jülich, Germany

Marius Lewerenz
Université Pierre et Marie Curie, Paris, France

World Scientific
Singapore • New Jersey • London • Hong Kong