

# GPU-Based Massive Parallel Kawasaki Kinetics in the Dynamic Monte Carlo Simulations of Lipid Nanodomains

Mateusz Lis,<sup>†</sup> Lukasz Pintal,<sup>†</sup> Jerzy Swiatek,<sup>†</sup> and Lukasz Cwiklik<sup>‡,§,\*</sup>

<sup>†</sup>Institute of Informatics, Faculty of Computer Science and Management, Wrocław University of Technology, Wyb. Wyspiańskiego 27, 50370 Wrocław, Poland

<sup>‡</sup>J. Heyrovský Institute of Physical Chemistry, Academy of Sciences of the Czech Republic, v.v.i., Dolejškova 3, 18223 Prague 8, Czech Republic

<sup>§</sup>Institute of Organic Chemistry and Biochemistry, Academy of Sciences of the Czech Republic, Flemingovo nám. 2, 16610 Prague 6, Czech Republic

**ABSTRACT:** Multicomponent lipid membranes in the liquid phase exhibit dynamic lateral heterogeneities which play an important role in specific cell membrane functions. A GPU-based parallel algorithm for two-dimensional lattice Dynamic Monte Carlo simulations of nanodomain formation in binary lipid membranes was developed and tested. Speedups of up to 50-times over CPU-based calculations were achieved, and simulations employing lattices of up to  $1800 \times 1800$  sites were performed. The existence of large nonregular lipid domains of sizes up to 160 nm was demonstrated. This reveals the necessity to employ lattices of at least  $\sim 900 \times 900$  sites to study the lateral lipid organization in complex lipid membranes.

## 1. INTRODUCTION

An understanding of mechanisms and conditions leading to formation and stabilization of nanodomains in lipid bilayers in the liquid state is essential to fully address such biologically relevant issues as lipid raft formation or lipid–protein interactions in cell membranes.<sup>1,2</sup> Nanodomains can emerge in the liquid phase of lipid mixtures as a result of fluctuations and instabilities due to nonideal mixing of different lipids.<sup>3–5</sup> The nature of these small and dynamic complexes is still debated, but they cannot be generally characterized as microscopic phases,<sup>2,6</sup> unlike the domains formed due to either liquid/gel or liquid-disordered/liquid-ordered phase transitions, both relatively well characterized in lipid bilayers. From the experimental point of view, the nonscopic lipid domains (i.e., of sizes below 200 nm) are challenging to study because they are smaller than the optical resolution limit and are of a highly dynamic nature. Therefore, theoretical and computational methods are essential for a better understanding of lateral heterogeneities in liquid lipid bilayers.

Usually, computational studies of membrane structure and dynamics employ molecular dynamics (MD) simulations. Atomistic-scale MD calculations were recently used to study lateral heterogeneities in binary lipid membranes with a non-random lateral lipid arrangement.<sup>7</sup> However, due to a high computational cost, the atomistic-scale MD is limited to small lateral sizes of membranes ( $\sim 10 \times 10$  nm<sup>2</sup>) and short time scales ( $\sim 200$  ns). Larger lateral sizes and longer time scales were probed employing coarse-grained MD simulations; nevertheless they are still significantly limited ( $\sim 20 \times 20$  nm<sup>2</sup> and  $\sim 30$  ns, accordingly).<sup>8,9</sup> The time and size restrictions are crucial from the point of view of lipid nanodomains formation as linear sizes of lipid aggregates as well as the characteristic time scales of the process operative during their formation are typically beyond those accessible by the coarse-grained MD.<sup>2</sup>

The drawbacks of molecular-level simulations in application to lipid membranes can be overcome by employing lattice Monte Carlo (MC) simulations due to discretization of spatial variables and an approximate treatment of the free-energy of the system. In lattice MC, individual leaflets of a lipid membrane are typically assumed to be independent and modeled as a two-dimensional triangular lattice with lipid molecules occupying lattice sites, thus each lattice site corresponds to one lipid molecule (or one acyl chain).<sup>10–12</sup> In the case of multicomponent membranes, different lipid types are assigned to lattice sites. The free-energy of the system is a function of a lattice configuration due to lipid–lipid interactions, typically calculated between the adjacent lattice sites, as well as a function of internal degrees of freedom of individual lipids (for instance, the state of lipids: gel or liquid). The length scale is given by comparison of an average lipid–lipid distance on the lattice with that experimentally measured in lipid bilayers. During an MC simulation, the system evolves from an initial configuration via a series of steps. Such a model of lipid membrane is actually a variant of the two-dimensional Ising spin model.<sup>13</sup> If equilibrium properties of the system are of primary interest, then Metropolis-type algorithms are employed for the dynamics in a way that usually does not reflect the physical time evolution of the system. If the dynamics is of primary interest, then nonequilibrium Dynamic Monte Carlo (DMC) simulations can be employed. In the case of interfacial phenomena, DMC simulations typically involve the Kawasaki dynamics to mimic diffusion processes and the time-evolution of the system.<sup>13,14</sup> The length of individual time-steps in DMC of lipid membranes can be scaled by comparison with lateral diffusion coefficients of lipids that can experimentally estimated using fluorescence correlation spectroscopy. Both equilibrium

Received: July 10, 2012

Published: September 21, 2012



and dynamic MC simulations were successfully employed for investigating lateral heterogeneities in lipid bilayers.<sup>10,12,15–17</sup> In most of these studies, however, a gel–liquid domain coexistence was of a main interest with thermodynamically stable phases present. Lattice sizes employed were limited to  $400 \times 400$  sites which corresponds to approximately  $300 \times 300$  nm in phospholipid bilayers. Although for well separated thermodynamic phases such lattice sizes seem adequate, we will demonstrate that in the case of the fluctuation-driven nanodomain formation in the liquid membrane state the use of larger lattices is necessary.

The dynamic MC simulations of domains in lipid membranes in liquid state require long simulation times because of two reasons. First, the use of large lattices involves long equilibration times. Second, as the time evolution of the domains is important, long trajectories must be simulated to gain statistically relevant information about the domain dynamics. To allow for long simulation times it is necessary to employ fast and optimized simulation schemes. The preferred increase of computational efficiency can be achieved using parallel simulation algorithms. Additionally, a substantial performance gain can be achieved by using graphical processing units (GPUs) as their rapid development in recent years gave possibility to implement and perform parallelized simulations on these devices.<sup>18–20</sup>

In this paper, we present a newly developed massively parallelized algorithm for two-dimensional lattice Dynamic Monte Carlo simulations with the Kawasaki-type dynamics and its implementation on GPUs. We employ our algorithm for simulations of lipid domains in liquid–lipid membranes. We achieve a significant efficiency improvement over CPU-based simulations. This allows us to perform simulations employing relatively large lattices, up to  $1800 \times 1800$  sites. We demonstrate that the use of such extensive lattices is necessary to address phenomena involved in phospholipid nanodomains formation. The rest of this paper is organized as follows: Section 2 briefly discusses the GPU-based calculations, presents Kawasaki-type dynamics in MC simulations, and introduces our Massive Parallel Kawasaki Kinetics algorithm and its implementation. The results obtained with this algorithm for a selected model of phospholipid membrane including a comparison with CPU-based simulations are presented in Section 3. Summary and conclusions are given in Section 4.

## 2. MATERIALS AND METHODS

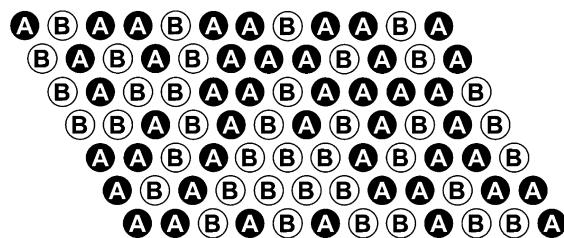
**2.1. GPU Architecture.** CPUs (Central Processing Units) of modern computers are designed to provide a vast amount of flexible computational resources to a single task. However, in the case of large data processing, the limits of sequential calculations are reached. If a spatial decomposition of the problem is possible, instead of sequential processing, then a parallel approach can be applied. In sequential computations, it is assumed that single computational tasks have to be executed one after another, whereas the parallel approach creates a possibility of simultaneous execution of computational tasks. The recent development of general purpose GPUs enabled the use of a massive parallelization, in the order of hundreds of processing units localized on a single GPU, whereas in the case of the CPU parallelization, the number of processing units is on the order of tens. The small computational power and flexibility of each GPU processing unit is compensated by a massive parallel execution.

The advantage of GPU processing over that achieved on CPUs is a field of intensive analyses. The computational speedup varies between  $1.5^{21}$  and 100 or even more,<sup>18</sup> and is a matter of

applied problem an optimizations. It is commonly agreed that the GPU processing provides a substantial boost for problems that involve simple single precision calculations on large data sets that do not require synchronization. Because of that, a rapid development of GPGPU software libraries in recent years is observed with different solutions (CUDA,<sup>22</sup> DirectCompute,<sup>23</sup> OpenCL<sup>24</sup>) introduced for various applications ranging from those realized on desktop workstations to computational clusters.

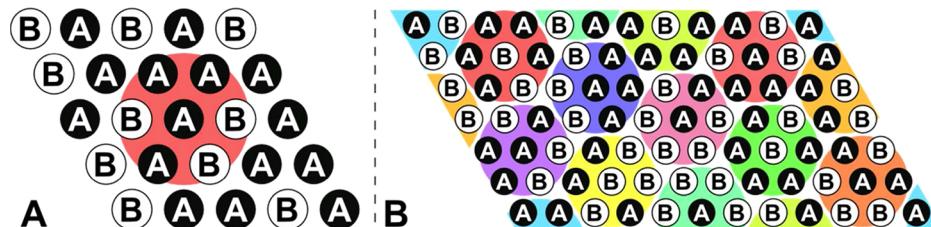
Much of the challenge in the design of efficient GPU applications lies in a proper decomposition of the problem to available computational resources. Every computational thread in GPU is capable of performing simple tasks on a specified subset of data. A group of threads is called a block, the size of which varies on different hardware devices (up to 1024 threads per block are present in modern architectures). The threads belonging to one block can share a fast access memory, called a *shared memory*, whereas the memory shared between the blocks is significantly slower. Thus, the data exchange between blocks should be avoided, as well as threads synchronization, as the latter effectively reduces the parallel processing to the sequential one. Another important issue is the data transfer between the computer and the device memory, which should be reduced to minimum as being time-consuming. In a perfect case, it would be possible to copy whole data to the device memory, then perform simulations and move the data back after computations. However, saving intermediate results usually requires the CPU access to the data; this is usually the case in simulations where a complete trajectory of the system is required to be exported during computations. Therefore, a successful implementation of a massive parallel algorithm needs to deal with three major design issues: data decomposition between blocks and threads, operations synchronization, and memory management. Each of these issues is addressed in the algorithm description given in the rest of this section.

**2.2. Kawasaki-Based Kinetics in Monte Carlo Simulations.** The two-dimensional lattice model of a lipid membrane is employed in which individual membrane leaflets are assumed to be independent and the membrane is described as a triangular lattice with lipid molecules occupying lattice sites; each lattice site corresponds to one lipid molecule or one acyl chain.<sup>10–12</sup> We consider multicomponent membranes, thus lattice sites may be occupied by different lipid types. The model of a membrane with two lipid components is schematically depicted in Figure 1. In Monte Carlo simulations, the system



**Figure 1.** The triangular lattice model of a lipid bilayer with lipids A and B randomly distributed on the lattice.

evolves from one microscopic state to another, based on an assumed probability distribution of the states under given conditions.<sup>13</sup> In the membrane model employed here, different microscopic states of the system correspond to different configurations of lipids on the lattice. Internal degrees of freedom



**Figure 2.** Minimal (seven-lipid) domain (A) and the whole lattice covered by minimal domains (B). Each domain is color-coded to depict helical boundary conditions.

of individual lipids (in particular, the state: liquid or gel) are not included in this model, as we focus on membranes in the liquid state.

In MC simulations in thermodynamic equilibrium, a conditional probability distribution is employed:  $q(x^*|x^{(n)})$ , where  $x^*$  is a candidate state, and  $x^{(n)}$  is a currently occupied state. The generated candidate states are iteratively accepted or rejected; usually based on the criterion introduced by Metropolis.<sup>25</sup> The algorithm proceeds as follows:

- 1 Generate an initial state  $x^{(0)}$
- 2 For  $i = 1$  to  $J$ :
  - a Generate a candidate state  $x^*$  according to the distribution  $q$ .
  - b Sample  $u \approx U[0,1]$
  - c If  $u < \min\{1, (p(x^*))/p(x^{(i-1)})\} (q(x^{(i-1})|x^*)/(q(x^*|x^{(i-1)}))\}$ ,  $x^{(i)} = x^*$ , else  $x^{(i)} = x^{(i-1)}$

where  $J$  is the number of iterations, i.e., attempts of transition from one microscopic state to another, and  $U[0,1]$  generates a uniformly distributed random number. To ensure a proper sampling of thermodynamic equilibrium, the transition distribution  $q(x^*|x^{(n)})$  has to satisfy two conditions: ergodicity and detailed balance.<sup>13</sup> The standard Metropolis algorithm has a naturally sequential form; therefore it is difficult to parallelize.<sup>26</sup> However, we demonstrate in this paper, for special cases, with an appropriate transition distribution, its parallelization is feasible.

For the membrane model with the lattice containing a binary mixture of lipids, every site contains a lipid of either type A or B. The number of lipids of each type is assumed to be constant throughout whole simulation; this assumption corresponds to the conserved order parameter type of the Ising model.<sup>13</sup> Because of that, we utilize Kawasaki kinetics as a transition distribution for lattices, which ensures a constant number of lipids of each type.<sup>14</sup> In the Kawasaki algorithm, a new state of the system is generated by a random selection of a pair of nearest neighboring lipids and an exchange of their positions on the lattice. The Kawasaki-type dynamics has an advantage over other methods of candidate states generation due to the fact that it mimics diffusion-like processes.

**2.3. Massive Parallel Kawasaki Kinetics.** The idea of introducing massive parallel computations to Kawasaki kinetics was previously considered by Shultz et al.<sup>27</sup> The concept of parallelization was based on a spatial lattice decomposition to disjoint domains. Simulations were performed independently for each domain and after a fixed number of iterations the domains were randomly translated. To prevent concurrent memory access, domains had to be separated by small buffers of lipids not taking part in simulations. This approach was reported to be up to 30 times faster than the classical Kawasaki method, but it introduces errors due to the independence of simulations in domains and existence of lipids that are not

involved in the simulation process. Those errors depend on the size of domains and the number of iterations before recalculation of the domains. Thus, a trade-off between the performance and accuracy is introduced in that simulation scheme.

Our Massive Parallel Kawasaki Kinetics (MPKK) is based on an extreme decomposition of triangular lattice to minimal domains that allow Kawasaki transition, namely the domains consisting of seven lipids, see Figure 2A. It can be shown that for certain sizes of the lattice, a set of such domains that completely covers whole lattice exists, see Figure 2B. Moreover, if a lattice can be covered in that way, there are exactly seven domain decompositions (coverages) for this lattice. Each of the possible decompositions can be generated using one of the first seven lipids on the lattice as a center point of a starting domain. Thus, a particular decomposition depends only on one parameter, namely the index of the center lipid of the generating domain. An important constraint for this decomposition method is that the lattice size has to be multiple of 7, because it is then possible to create a full coverage with domains of size 7. Moreover, in helical boundary conditions which are employed in the present study, it is necessary that the lattice length and width are not equal, namely the width has to be smaller by 2 lattice sites than the length (or vice versa), thus the possible sizes are, for instance,  $112 \times 110$ ,  $119 \times 115$ ,  $126 \times 124$ , and so forth.

In every step of MPKK a simulation, a random minimal domain decomposition is generated, and a Kawasaki step is performed on each domain simultaneously. Thus, a negative influence of the presence of domain boundaries and of the independence of individual domains is avoided while a high parallelization level can be preserved. Due to a large number of domains, a GPU-based parallelization can be efficiently utilized by performing the simulation of each domain on a different processing unit. This approach requires the domain recalculation in every step. Nevertheless, such a recalculation is only a matter of generating a random integer between 1 and 7.

The MPKK algorithm has the following form:

- 1 For  $i = 1$  to  $N$ :
  - a For  $j = 1$  to 7:
    - i  $k = \text{rand}\%7$   $k = \text{rand}\%7$
    - ii Generate a decomposition  $D(k)$  over  $k$ -th lipid
    - iii For each domain  $d$  in  $D(k)$ :
      - i Perform a Kawasaki step on the center lipid of  $d$

where  $N$  is the number of simulation steps to perform.

Each step of the algorithm requires seven iterations (domain recalculations) to provide compatibility with the traditional Kawasaki method. In lattice MC simulations, one MC step is

typically defined as consisting of the number of transition attempts equal to the number of lattice sites.<sup>13</sup> In MPKK, one MC step consists of seven times more transition attempts than the number of lattice sites.

A properly defined transition distribution in an MC algorithm must satisfy criteria of both ergodicity and detailed balance.<sup>13</sup> The condition of ergodicity requires that the transition distribution enables to reach any state of the system from any other state. A single iteration of MPKK performs a Kawasaki-type exchange of states. By performing a series of MPKK iterations for different domain decompositions, each state of the system can be reached. Thus, MPKK satisfies the ergodicity requirement. To satisfy the detailed balance, the proposal distribution needs to provide equal probability of achieving all possible candidate states. In a single iteration of MPKK, the probability of selecting any particular pair of lipids to exchange is equal to  $1/7 \times 1/6$  since every one of seven lipids becomes a basis of decomposition and for every lipid there are six neighbors to exchange with. Therefore, the proposal distribution is equal for every possible candidate state. Thus, MPKK satisfies the detailed balance requirement.

**2.4. Simulations of Lipid Membrane Employing the MPKK Algorithm.** In order to simulate lipid membranes in the liquid state with the MPKK algorithm, we employed a simplified model of the membrane assuming that each lipid is in its liquid state; therefore, no gel–liquid transitions occur (similar models have been employed previously by other authors.<sup>11,28</sup>) Membrane leaflets are assumed to be independent and described with a two-dimensional triangular lattice. In a binary mixture of lipids A and B, only one thermodynamic parameter is needed to describe the free-energy of interaction in the system, namely, the unlike nearest-neighbor free-energy:<sup>28</sup>

$$\omega_{AB} = g_{AB} - \frac{1}{2}(g_{AA} + g_{BB})$$

where  $g_{AA}$ ,  $g_{AB}$ ,  $g_{BB}$  are Gibbs free-energies of interactions between nearest-neighboring lipid molecules AA, AB, and BB, respectively. The value of  $\omega_{AB}$  is characteristic for a given pair of lipids A and B, and can be experimentally estimated. This model, although relatively simple, is capable of providing biologically relevant results regarding lateral organization of lipid membranes.<sup>16,29,30</sup> In a single Kawasaki-type exchange of two lipids the change of the free energy equals:  $\Delta G = \Delta n_{AB}\omega_{AB}$ , where  $\Delta n_{AB}$  is the change in the number of unlike neighbors. The acceptance probability for a new configuration equals  $\min\{1, \exp(-\Delta G/kT)\}$ , where  $k$  is Boltzmann constant and  $T$  is temperature; and corresponds to the acceptance probability given in the step 2c of the Metropolis algorithm. As the Kawasaki kinetics mimics physical lateral diffusion of lipids in the membrane, such model, after estimating the time scale of a single MC step using the experimental diffusion coefficients, can be employed to simulate the dynamics of lipids in the membrane.<sup>13,17</sup>

The membrane model employed here can be extended in order to include longer range interactions, such as electrostatic interactions between zwitterionic and charged phospholipids; a similar model was previously introduced by Huang and co-workers.<sup>11</sup> In the case of the parallelization strategy introduced here, incorporation of such long-range interactions would reduce the performance gain, however, it would be feasible. On the other hand, an extension of the model toward inclusion of additional internal degrees of freedom of lipid molecules would

be relatively straightforward and would not significantly limit the computational efficiency.

**2.5. Image Analysis.** To facilitate the analysis process, reduce the disk space required for storing simulation trajectories, and to limit amount of data transferred between GPU and the main memory, snapshots of specific frames were saved in the form of previously scaled images. Each lattice site was represented as a white or black pixel on the image plane, depending on its type (lipid A or B). Then the supersampling interpolation method,<sup>31</sup> generating gray scale images where each pixel represents a group of several nearest neighboring lattice sites and its color corresponds to the ratio of the numbers of lipids A to lipids B in the group, was used to generate the image that represented to the considered trajectory snapshot. The resulting image was then sloped to preserve the original geometry of the triangular lattice and scaled down to a desired size. The above operations were performed on GPU, and thus the number of expensive memory copying operations was minimized. In this work, we generated and transferred images from GPU to main memory every  $10^5$  simulation steps. It is important to note that performing analytical operations on scaled images is significantly faster than on original data while preserving high quality results. Thus the efficiency gain due to the employment the image analysis instead of an analysis of original trajectories originates from the fact that image generation algorithms can be employed on GPU and, additionally, the amount of data to be analyzed and transferred is reduced. The trajectories generated during MC calculations and stored in the form of images were used for conducting two types of data analyses. First, the fraction of first neighbors (FFN) was estimated for the minor lipid component. Typically, the FFN analysis on a triangular lattice proceeds as follows: for every site in the lattice occupied by a lipid of the minor type, one of its nearest neighbors is randomly chosen and the similar neighbors counter is incremented by one if both sites are occupied by the lipids of the same type. Finally, the similar neighbors counter is divided by the number of lattice sites occupied by the minor lipids. In the case of the gray scale images resulting from the supersampling procedure, FFN is applied to pixels instead of lipids. As pixel colors have integer values that are proportional to the ratio of lipids of each type, we define a similarity measure as a color space distance between pixels, and introduce a threshold value for pixels comparison. The threshold must be chosen empirically, but can stay constant for all analyses performed on different trajectories. In the present work, the value of the color distance equal to 35 was chosen as the threshold.

The second method of the trajectory analysis employed here was the analysis of lipid clusters which evaluates the tendency of lipids to be grouped in domains. In particular, it provides a quantitative method of estimation whether lipids of the type A prefer to mix with lipids of the type B or not. A choice of an efficient cluster analysis method is essential as clusters need to be analyzed regularly during MC simulations to provide information about the dynamics of lipid aggregates. Thus, the algorithm with the complexity of  $O(n)$  is required. Here, the Hoshen-Kopelman algorithm,<sup>32</sup> which satisfies this requirement, was employed with a modification applied to the algorithm to account for the triangular lattice geometry. Namely, helical boundary conditions were used for finding the six nearest neighbors on the lattice in the stage of assigning labels to lattice sites.

**2.6. Devices Utilized for Performance Measurements.** MC simulations discussed in the next section were obtained

using the nVidia Tesla M2090 card with 512 cores and 6 GB of memory (GPU results), and Intel Xeon CPU E5640 2.67 GHz with 8 cores used and 2.8 GB of RAM per core (CPU results). The GPU implementation used the CUDA 4.1 library. All implemented algorithms are publicly available<sup>33</sup> (version r104 was used for the results reported here). The implementation of the MPKK algorithm utilizes the Mersenne-Twister random number generator.<sup>34</sup>

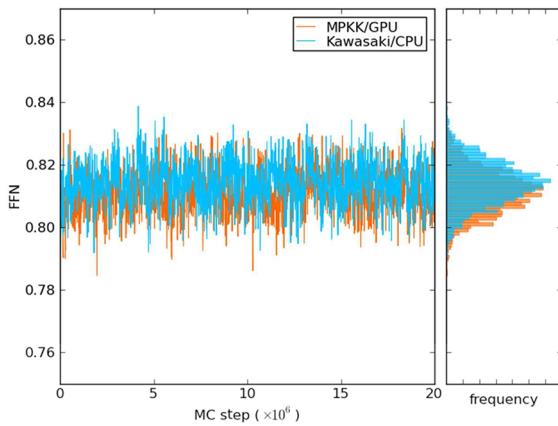
### 3. RESULTS AND DISCUSSION

#### 3.1. Comparison between MPKK and Kawasaki Algorithms.

The MPKK algorithm was tested for different sets of model parameters. We report here the results obtained for the interaction parameter  $\omega_{AB} = 350$  cal/mol at the temperature of 310 K. This particular choice of  $\omega_{AB}$  and the temperature corresponds to a case of strongly interacting lipids under physiologically relevant temperature<sup>16</sup> and is employed here in order to study the system where formation of lipid nanodomains is expected to be significant. The ratio of lipids was set to 2:1 in order to mimic a mixture with an abundance of one of the lipids. To enable the comparison between Kawasaki and MPKK algorithms, MC simulations employing both of these methods for small ( $112 \times 112$ ) lattices were performed and analyzed. The fractions of first neighbors (FFN, calculated for the lipids of the minor type) estimated during simulations with  $20 \times 10^6$  MC steps length and randomly generated initial patterns of lipids are presented in Figure 3. Histograms of FFN,

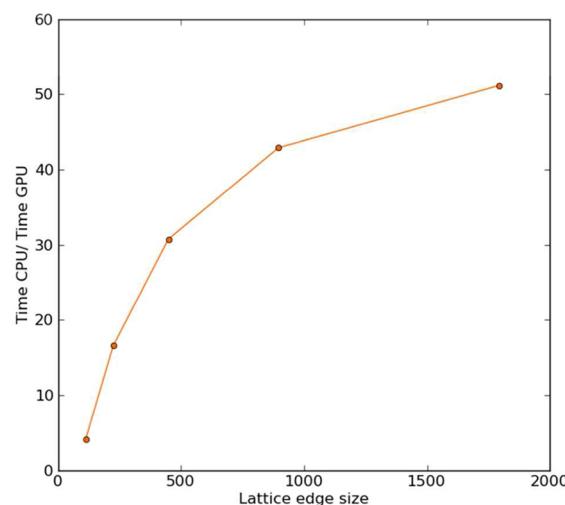
FFN after reaching the equilibrium demonstrate that the MPKK algorithm on GPU is equivalent to the Kawasaki algorithm realized on CPU. The observed minor deviations between both methods, in particular, a slightly lower average FFN in MPKK, result from two factors: the lattice decomposition introduced in MPKK and different pseudo random number generators employed in CPU and GPU simulations. While the choice of a random number generator should not matter, the domain decomposition may, in principle, influence the results. The domain decomposition employed in the MPKK algorithm is relevant in the length scales comparable with the size of individual domains, i.e., up to several lipids; and in short time scales, comparable with the time needed for a randomization of the domain choice. Note that only seven different realizations of the domain decomposition on the lattice are possible, therefore the randomization is achieved in approximately seven simulation steps which is negligible in comparison with the number of steps performed in each simulation. In the MPKK simulation, the lattice sites in a given step evolve neglecting the influence of the sites from the outside of their own spatial domain. Thus, in the system where the ordering of lipids occurs due to lipid–lipid interactions, the spatial correlations are expected to be somewhat reduced. This rationalizes a small value of the difference between FFN calculated employing both methods. These results demonstrate that in the time and length scales relevant for the properties of lipid membranes MPKK and Kawasaki algorithms are equivalent.

In Figure 4 the computational performance gain, defined as the ratio between the simulation time employing the MPKK



**Figure 3.** Fraction of first neighbors (FFN) calculated for the minor lipid component as a function of MC steps employing MKPP algorithm on GPU and Kawasaki algorithm on CPU for the lattice of  $112 \times 110$  sites. Histograms of FFN values are depicted in the right panel.

which represent the spread of FFN values calculated employing both algorithms, are presented in the right panel of the figure. In both algorithms, FMM increases during the initial  $2 \times 10^6$  MC steps, then its value stabilizes, although relatively strong fluctuations are present due to the small size of the lattice. The average FFN calculated between  $10 \times 10^6$  and  $20 \times 10^6$  MC step equals to  $0.812 (\pm 0.01)$  and  $0.814 (\pm 0.01)$  for, accordingly, MPKK on GPU and Kawasaki on CPU. These values are practically equal. Note that for a random mixture of lipids with the 2:1 lipid ratio, FFN calculated for the minor component equals  $1/3$ ; thus in both algorithms, a coarsening of lipids and formation of lipid domains occurs. The agreement between the kinetics of the increase of FFN in the initial simulation phase, the extent of FFN fluctuations, as well as the average value of

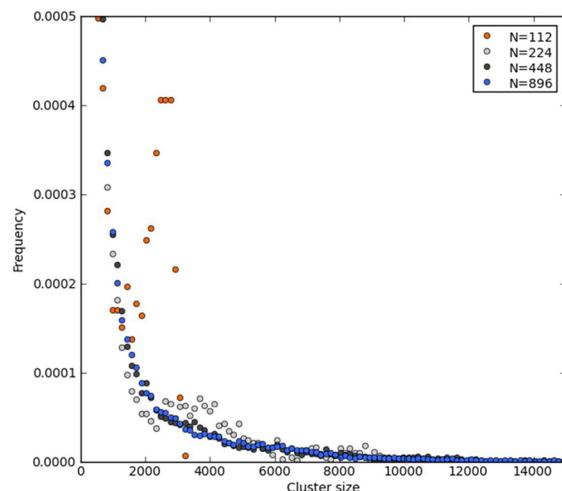


**Figure 4.** Performance gain (the ratio between the simulation time employing Kawasaki algorithm realized on CPU and the simulation time with the MPKK algorithm on GPU) as a function of lattice size.

algorithm realized on GPU and the simulation time with the Kawasaki algorithm on CPU for the same lattices, is presented for different lattice sizes. The performance gain increases with the increasing size of the lattice, for the  $100 \times 100$  lattice the gain is 5-fold while it is over 50-fold for the lattice of  $1800 \times 1800$  sites. As the lattice sizes in the range from  $500 \times 500$  to  $1000 \times 1000$  are of particular interest from the point of view of lipid nanodomains formation, the MPKK algorithm realized on GPU provides a valuable alternative for CPU-based MC calculations in computational studies of such phenomena occurring in lipid membranes. Note that in the simulations realized on

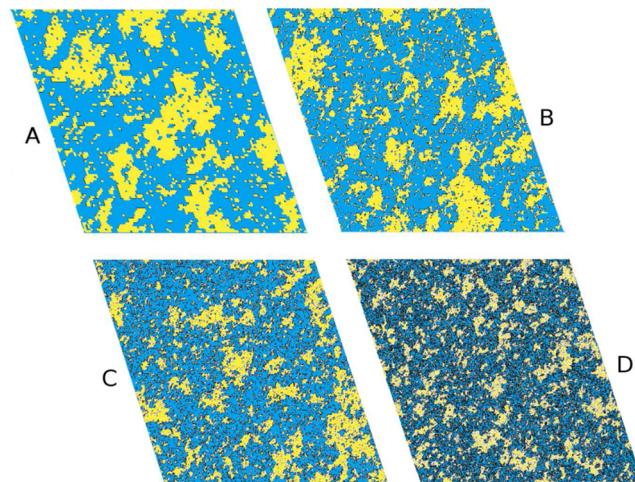
GPU, we obtained an additional performance gain by employing the image analysis methods instead of the trajectory-based approach. The lattice sizes are, in principle, not limited to  $1800 \times 1800$  sites. In the present implementation, we store lattices in the GPU memory as tables of char variables, in order to allow for future simulations employing many-state lattice sites. The amount of memory available on the GPU device used here (6 GB) would allow us for simulations of lattices with up to  $30\,000 \times 30\,000$  sites. But the limiting factor is the computational time required for an equilibration. For instance, our tests show that it is currently unfeasible to simulate lattices of  $9000 \times 9000$  sites.

**3.2. Domain Formation.** From the point of view of physiologically relevant lipid membranes, the most interesting property which can be estimated employing lattice MC simulations is a distribution of lipid domain sizes. To examine lipid domains, we performed MC simulations employing lattices of different sizes (from  $112 \times 110$  to  $896 \times 894$ ) with the random pattern of lipids taken as the initial condition. Simulations of  $20 \times 10^6$  MC steps were performed and the last  $5 \times 10^6$  MC steps were taken for the data averaging. The average cluster size calculated upon equilibration for all considered lattice sizes equals to  $13(1 \pm)$ ; thus, the clustering of lipids occurs, and no difference between the lattices regarding the average lipid domain size. However, as will be demonstrated, distributions of cluster sizes are strongly altered. Figure 5 depicts equilibrated



**Figure 5.** Distributions of cluster sizes calculated for equilibrated systems of different lattice sizes. Simulations were performed for  $20 \times 10^6$  MC steps and the last  $5 \times 10^6$  MC steps were taken for averaging.

domain distributions calculated for lattices of different sizes. Representative snapshots from the final stages of each trajectory are depicted in Figure 6. In each system, the small domains of less than 1000 lipids prevail in number. For the lattice with  $112 \times 110$  sites, a peak located at about 2200 lipids is present. Similar but less pronounced maximum, located at about 3500 lipids, occurs for  $224 \times 222$  sites. For the lattice with  $448 \times 446$  sites, small fluctuations of the distribution are present close to 3500 lipids. The distribution for the largest lattice, with  $896 \times 896$  sites, does not exhibit any noticeable maximum and it relatively smoothly decays with the increasing cluster size. The presence of the maxima in the case of small lattices stems from finite size effects as for the lattices with both  $448 \times 446$  and  $896 \times 894$  sites no maxima corresponding to such clusters are

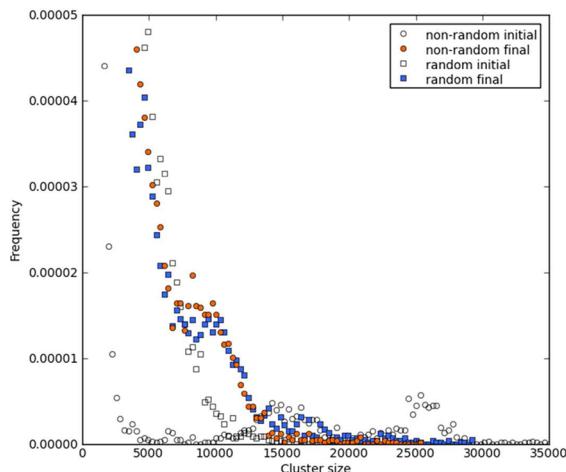


**Figure 6.** Representative snapshots of simulation lattices taken at final stages of MC simulation for lattices of different sizes:  $112 \times 110$  (A),  $224 \times 222$  (B),  $448 \times 446$  (C), and  $896 \times 894$  (D) sites employing the image compression algorithm (see Section 2). The snapshots are rescaled to obtain images of the same size. Lipids of the prevalent type, are shown in blue (for presentation purposes the original black color was exchanged with blue), the minor ones—in yellow (originally in white). The lattice points depicted in the shades of gray represent the lattice sites where mixed-type neighbors were present (for details, see Section 2).

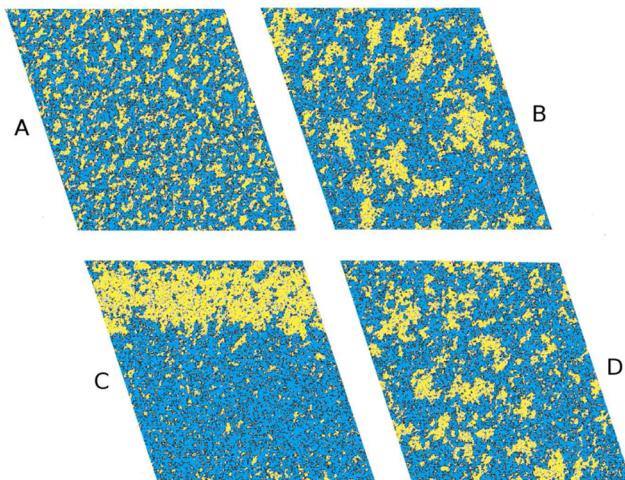
present under the considered conditions. This demonstrates that small lattices, of less than  $400 \times 400$  lipids, are unable to properly reproduce the clustering of lipids. Our results led to an estimate that lattices of at least  $450 \times 450$  lipids are required to properly describe nanodomains formation in the case of strongly interacting membrane components in the liquid lipid phase. It can be further anticipated that in more complicated systems, such as mixed lipid–protein membranes or membranes undergoing liquid–solid phase transitions accompanying raft formation, at least  $900 \times 900$  sites lattices would be required to avoid finite-size effects.

The snapshots depicted in Figure 6 reveal that the lipid domains of strongly nonregular shapes are typically formed in equilibrium. This indicates that the average cluster size is not a suitable quantity to fully characterize such lipid complexes; this rationalizes the observed differences between cluster distributions calculated for the different lattices while the average cluster size for each system was effectively equal. Moreover, this demonstrates the necessity to employ lattice MC methods for studying formation of lipid nanodomains, as these techniques are able to explicitly describe lateral heterogeneities in contrast to, for instance, standard mean-field approaches.<sup>13</sup>

To characterize the long-time scale behavior in the case of large lattices, distributions of cluster sizes in simulations started with varying initial conditions were calculated. More specifically, the lattice of  $490 \times 488$  sites and the 2:1 lipid ratio was initially populated with either a random pattern of lipids or a continuous stripe made of the minor lipid; then extensive simulations, consisting of  $110 \times 10^6$  MC steps, were performed. Note, that for the  $500 \times 500$  lattice, each MC step consists of  $2.5 \times 10^5$  exchanges of lipid pairs. Distributions of cluster sizes calculated in the initial and final stage of simulations are shown in Figure 7. The corresponding snapshots of the simulation lattices are shown in Figure 8. In the case of the random initial configuration, the initial distribution smoothly decays up to the



**Figure 7.** Distributions of cluster sizes calculated at initial (between 0 and  $1 \times 10^6$  MC step) and final (between  $110 \times 10^6$  and  $120 \times 10^6$  MC step) stages of MC simulations initiated with a random pattern of lipids and a nonrandom (the stripe made of the minor lipid) configuration for lattices of  $490 \times 488$  sites.



**Figure 8.** Representative snapshots of simulation lattices taken at initial, i.e., at  $10^4$  (A) and  $1 \times 10^6$  (C) MC step; and final, i.e., at  $120 \times 10^6$  step (B and D) stages of the MC simulations initiated with either a random pattern of lipids (A, B) or the stripe made of the minor lipid (C, D) at lattices of  $490 \times 488$  sites.

cluster size of approximately 15 000 lipids. Note that, for instance, lipid aggregates of 10 000 lipids correspond to domains with the diameter of approximately 90 nm (assuming the circular shape of a domain and taking the area per lipid equal to  $0.65 \text{ nm}^2$  which is the typical value for phosphatidylcholine lipids in hydrated lipid bilayers<sup>35</sup>). During simulation, the growth of larger lipid domains at the expense of smaller aggregates is evident (compare Figure 8A,B), and the distribution calculated in the final stage of the simulation exhibits a maximum at the cluster size of approximately 10 000 lipids. Moreover, few smaller maxima are present, at 17 000 and 23 000, and the distribution reaches zero only at lipid sizes of approximately 30 000. Thus, the largest domains found in the system have diameter of approximately 195 lipids, which corresponds to about 160 nm. The presence of clusters of such size demonstrates that for model parameters employed here the lattices smaller than  $400 \times 400$  sites, i.e., with the linear size being only two times larger than the size of clusters, would exhibit strong finite size

effects; in agreement with the conclusion drawn beforehand based on the values of FFN.

In the case of the simulation with the continuous stripe of minor lipids used as an initial configuration, the initial distribution of cluster sizes exhibits peaks at approximately 15 000, 26 000, and 60 000 lipids (the latter not shown in Figure 7) which originate from the melting of the initial lipid stripe (see Figure 8 C). The melting process generates large clusters ( $>10\,000$  lipids), and numerous small clusters as evident from both the nonzero values of the distribution below cluster size of 5000 lipids and from the snapshot in Figure 8 C. During the simulation, most of large clusters melt and the number of small clusters increases (see Figure 8D). The final distribution exhibits the maximum at approximately 10 000 lipids and resembles the distribution obtained from the simulation started with the random initial pattern of lipids (compare Figure 8B,D). It demonstrates, that independently on the initial conditions, similar lipid aggregates are formed. Large domains of lipids, up to 30 000, are present in the equilibrated systems with the preference toward aggregates of  $\sim 10\,000$  lipids. Large clusters are accompanied by numerous smaller aggregates.

The melting of the stripe of lipids observed in the simulation initiated with the nonrandom lattice configuration demonstrates that no macroscopic phase separation occurs in the considered lipid mixture system under the conditions employed. The initial stripe can be considered as a “macroscopic” phase, as the interface between the domains consisting of lipids A and B is relatively small with respect to the bulk of the domains. However, the distribution of lipid domain sizes observed in the equilibrated system, with the presence of relatively large aggregates, indicates that the system is close to the demixing phase transition. This is in accord with other studies where liquid lipid membranes under physiological conditions were shown to be close to the demixing point.<sup>36</sup>

#### 4. CONCLUSIONS

We have developed a massive parallel algorithm for Kawasaki-type kinetics in two-dimensional lattice Dynamic Monte Carlo simulations, and implemented it on GPU in order to study the structure and dynamics of domains formed in two-component liquid lipid membranes. The algorithm is based on a fine domain decomposition with individual domains consisting exclusively of the sites forming the nearest neighborhood of a given site. Up to 50-fold speedup was achieved with respect to the simulations performed on CPU. In practice, on the presently available GPU hardware, the algorithm allows for performing MC simulations on lattices consisting of up to approximately  $1800 \times 1800$  sites. In most of the recent works concerning lipid membranes, lattices of up to  $500 \times 500$  sites were employed and CPU-based simulations were used. We have demonstrated that under physiologically relevant conditions in mixtures of strongly interacting lipids, large domains, consisting of up to 30 000 lipids, are present in equilibrium; and relatively large lattices should be used to properly describe formation of nanodomains in such systems. The shapes of the domains formed are strongly nonregular; therefore, they cannot be characterized by the average cluster size. It can be expected that in more complex assemblies with stronger interactions, such as in mixed lipid–protein membranes or membranes undergoing liquid–solid phase transitions that accompany raft formation, the lattices of about  $900 \times 900$  would be required to avoid finite-size effects and obtain statistically relevant data. The present MPKK

algorithm realized on GPUs is well suited for performing DMC simulations of such computationally demanding systems.

## AUTHOR INFORMATION

### Corresponding Author

\*E-mail: lukasz.cwiklik@jh-inst.cas.cz.

### Notes

The authors declare no competing financial interest.

## ACKNOWLEDGMENTS

We thank the Wroclaw Centre for Networking and Supercomputing (<http://www.wcss.wroc.pl>), Grant No. 137; and acknowledge the support from the Czech Science Foundation, Grant No. P208/12/G016. M.L. thanks Bartlomiej Piekarski from the Wroclaw University of Technology for technical insight into GPU computing.

## REFERENCES

- (1) Simons, K.; Ikonen, E. Functional rafts in cell membranes. *Nature* **1997**, *387*, 569–572.
- (2) Elson, E. L.; Fried, E.; Dolbow, J. E.; Genin, G. M. Phase separation in biological membranes: Integration of theory and experiment. *Annu. Rev. Biophys.* **2010**, *39*, 207–26.
- (3) Melchior, D. L.; Carruthers, A. How bilayer lipids affect membrane protein activity. *Trends Biochem. Sci.* **1986**, *11*, 331–335.
- (4) Wu, S. H.; McConnell, H. M. Phase separations in phospholipid membranes. *Biochemistry* **1975**, *14*, 847–854.
- (5) Fan, J.; Sammalkorpi, M.; Haataja, M. Formation and regulation of lipid microdomains in cell membranes: Theory, modeling, and speculation. *FEBS Lett.* **2010**, *584*, 1678–84.
- (6) Veatch, S. L.; Keller, S. L. Seeing spots: Complex phase behavior in simple membranes. *Biochim. Biophys. Acta* **2005**, *1746*, 172–85.
- (7) Pyrkova, D. V.; Tarasova, N. K.; Pyrkov, T. V.; Krylov, N. A.; Efremov, R. G. Atomic-scale lateral heterogeneity and dynamics of two-component lipid bilayers composed of saturated and unsaturated phosphatidylcholines. *Soft Matter* **2011**, *7*, 2569.
- (8) Faller, R.; Marrink, S.-J. Simulation of domain formation in DLPC-DSPC mixed bilayers. *Langmuir* **2004**, *20*, 7686–93.
- (9) Rosetti, C.; Pastorino, C. Comparison of ternary bilayer mixtures with asymmetric or symmetric unsaturated phosphatidylcholine lipids by coarse grained molecular dynamics simulations. *J. Phys. Chem. B* **2012**, *116*, 3525–37.
- (10) Mouritsen, O. G.; Boothroyd, A.; Harris, R.; Jan, N.; Lookman, T.; MacDonald, L.; Pink, D. a.; Zuckermann, M. J. Computer simulation of the main gel–fluid phase transition of lipid bilayers. *J. Chem. Phys.* **1983**, *79*, 2027.
- (11) Huang, J.; Swanson, J. E.; Dibble, a. R.; Hinderliter, a. K.; Feigenson, G. W. Nonideal mixing of phosphatidylserine and phosphatidylcholine in the fluid lamellar phase. *Biophys. J.* **1993**, *64*, 413–25.
- (12) Sugar, I. P.; Thompson, T. E.; Biltonen, R. L. Monte Carlo simulation of two-component bilayers: DMPC/DSPC mixtures. *Biophys. J.* **1999**, *76*, 2099–110.
- (13) Newman, M. E. J.; Barkema, G. T. *Monte Carlo Methods in Statistical Physics*; Oxford University Press: New York, 2001.
- (14) Kawasaki, K. Diffusion constants near the critical point for time-dependent ising models. I. *Phys. Rev.* **1965**, *224*–230.
- (15) Hac, A. E.; Seeger, H. M.; Fidorra, M.; Heimburg, T. Diffusion in two-component lipid membranes—a fluorescence correlation spectroscopy and monte carlo simulation study. *Biophys. J.* **2005**, *88*, 317–33.
- (16) Almeida, P. F. F. Thermodynamics of lipid interactions in complex bilayers. *Biochim. Biophys. Acta: Biomembr.* **2009**, *1788*, 72–85.
- (17) Ehrig, J.; Petrov, E. P.; Schwille, P. Near-critical fluctuations and cytoskeleton-assisted phase separation lead to subdiffusion in cell membranes. *Biophys. J.* **2011**, *100*, 80–89.
- (18) Preis, T.; Virnau, P.; Paul, W.; Schneider, J. J. GPU accelerated Monte Carlo simulation of the 2D and 3D Ising model. *J. Comput. Phys.* **2009**, *228*, 4468–4477.
- (19) Levy, T.; Cohen, G.; Rabani, E. Simulating lattice spin models on graphics processing units. *J. Chem. Theory Comput.* **2010**, *6*, 3293–3301.
- (20) Liu, L.; Liu, X.; Gong, J.; Jiang, H.; Li, H. Accelerating all-atom normal mode analysis with graphics processing unit. *J. Chem. Theory Comput.* **2011**, *7*, 1595–1603.
- (21) Lee, V. W.; Kim, C.; Chhugani, J.; Deisher, M.; Kim, D.; Nguyen, A. D.; Satish, N.; Smelyanskiy, M.; Chennupaty, S.; Hammarlund, P.; Singhal, R.; Dubey, P. Debunking the 100X GPU vs. CPU myth: An evaluation of throughput computing on CPU and GPU. *SIGARCH Comput. Archit. News* **2010**, *38*, 451–460.
- (22) NVIDIA CUDA C Programming Guide v. 4.2 [http://developer.download.nvidia.com/compute/DevZone/docs/html/C/doc/CUDA\\_C\\_Programming\\_Guide.pdf](http://developer.download.nvidia.com/compute/DevZone/docs/html/C/doc/CUDA_C_Programming_Guide.pdf) (accessed Sept 7, 2012).
- (23) DirectCompute Support on NVIDIA’s CUDA Architecture GPUs <http://developer.nvidia.com/directcompute> (accessed Sept 7, 2012).
- (24) Khronos OpenCL Working Group, The OpenCL 1.0 Specification <http://www.khronos.org/opencl/> (accessed Sept 7, 2012).
- (25) Metropolis, N.; Rosenbluth, A. W.; Rosenbluth, M. N.; Teller, A. H.; Teller, E. Equations of state calculations by fast computing machines. *J. Chem. Phys.* **1953**, *21*, 1087–1092.
- (26) Lee, A.; Yau, C.; Giles, M. B.; Doucet, A.; Holmes, C. C. On the utility of graphics cards to perform massively parallel simulation of advanced monte carlo methods. *J. Comput. Graph. Stat.* **2010**, *19*, 769–789.
- (27) Schulz, H.; Odor, G.; Kelling, J.; Heinig, K.-H.; Liedke, B.; Schmeißer, N. Computing the KPZ equation using GPU acceleration. In *3rd International Workshop “Innovation in Information Technologies—Theory and Practice*; Dresden, Germany, 2010; pp 46–48.
- (28) Almeida, P. F. F.; Pokorny, A.; Hinderliter, A. Thermodynamics of membrane domains. *Biochim. Biophys. Acta* **2005**, *1720*, 1–13.
- (29) Hinderliter, a.; Almeida, P. F.; Creutz, C. E.; Biltonen, R. L. Domain formation in a fluid mixed lipid bilayer modulated through binding of the C2 protein motif. *Biochemistry* **2001**, *40*, 4181–91.
- (30) Frazier, M. L.; Wright, J. R.; Pokorny, A.; Almeida, P. F. F. Investigation of domain formation in sphingomyelin/cholesterol/POPC mixtures by fluorescence resonance energy transfer and Monte Carlo simulations. *Biophys. J.* **2007**, *92*, 2422–33.
- (31) Goss, M.; Wu, K. *Supersampling Methods for Computer Graphics Hardware Antialiasing*. Technical Report 121R1; Hewlett-Packard Laboratories: Palo Alto, California, USA, 1999.
- (32) Hoshen, J.; Kopelman, R. Percolation and cluster distribution. I. Cluster multiple labeling technique and critical concentration algorithm. *Phys. Rev. B* **1976**, *14*, 3438–3445.
- (33) pymd2mc. <http://code.google.com/p/pymd2mc/>
- (34) Matsumoto, M.; Nishimura, T. Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans. Model. Comput. Simul.* **1998**, *8*, 3–30.
- (35) Gurtovenko, A. A.; Vattulainen, I. Effect of NaCl and KCl on phosphatidylcholine and phosphatidylethanolamine lipid membranes: insight from atomic-scale simulations for understanding salt-induced effects in the plasma membrane. *J. Phys. Chem. B* **2008**, *112*, 1953–1962.
- (36) Longo, G. S.; Schick, M.; Szleifer, I. Stability and liquid-liquid phase separation in mixed saturated lipid bilayers. *Biophys. J.* **2009**, *96*, 3977–86.