Springer Catalogue Search

Available
DM 149,-
Recommended List Price

Put in Bag

Fields: Computational Mathematics and Scientific Computing; Computer Applications in Chemistry; Mathematical and Computational Methods

Written for: For computational scientists

Book category: Proceedings

Publication language: English

Krause, E., *RWTH Aachen, Germany*
Jäger, W., *University of Heidelberg, Germany*
(Eds.)

High Performance Computing in Science and Engineering '99

Transactions of the High Performance Computing Center, Stuttgart (HLRS) 1999

2000. VIII, 515 pp. 281 figs., 99 in color, 28 tabs.

3-540-66504-8
DM 149,-
Recommended List Price

The book contains reports about the most significant projects from science and engineering of the Federal High Performance Computing Center Stuttgart (HLRS). They were carefully selected in a peer-review process and are showcases of an innovative combination of state-of-the-art modeling, novel algorithms and the use of leading-edge parallel computer technology. The projects of HLRS are using supercomputer systems operated jointly by university and industry and therefore a special emphasis has been put on the industrial relevance of results and methods.

Keywords: High-Performance-Computing Scientific Computing, Numerical methods, Parallel computing

Last update: 08.03.2000
Your inquiries about Springer products are welcome at *International Sales and Marketing*

MD Simulation of a Phospholipid Bilayer

Ralf Schmelter, Alfons Geiger

Physikalische Chemie, Universität Dortmund

Abstract. We report first results of a molecular dynamics simulation of a fully hydrated dipalmitoyl–*sn*–glycero–phosphatidylcholin (DPPC) bilayer using the NγT-ensemble. Because of the large size of the simulated system, a parallel version of the simulation package MOSCITO [1] has been developed, which employs a dynamic loadbalancing algorithm to ensure uniform workload among the processors.

1 Introduction

Phospholid bilayers can be regarded as models of eukaryontic cell membranes and therefore have been heavily investigated by experimental and theoretical methods [2] [3] [4]. In recent years it became feasible to perform molecular dynamics (MD) and Monte-Carlo (MC) simulations of such systems, by which one hopes to gain unterstanding of the structural and dynamic properties of the membranes at the atomic level. While the first simulations suffered from short simulation times, too small system sizes or improper treatment of the long range electrostatic interactions, the more recent simulations had overcome these limitations, mainly because of the rapid growth in available computer power [5] [6].

Nevertheless there are some open questions regarding the simulation of a fully hydrated phospholipid bilayer and one of the most important and most discussed topics is the choice of the ensemble to be used [7] [12]. Many of the first simulations were carried out in an NVT ensemble. In this case one has to choose the box dimensions carefully, which means to select a proper area per lipid and the correct multilamellar repeat distance. Unfortunately there are many different experimental values for this area [8] (they range between 0.56 nm^2 and 0.72 nm^2) and the multilamellar repeat distance. Another problem is the validation of the forcefield, because there are only few experimental results which can be easily and unambiguously compared with the results of the simulation. One of the most important parameters to be compared to the experiment is the deuterium order parameter S_{CD} of the alkyl chains, which almost exclusively depends (at least in the simulation) on the area per lipid [9], so that after a proper choice of the area even a totally inadequate forcefield can give sensible results for the order parameters, while the other properties may be far from reality.

Consequently simulations using the NPT ensemble were carried out, in which the box dimensions are adjusted during the simulation to produce a given pressure. To take the anisotropy of the membrane system into account,

High Performance Computing
in Science and Engineering,
Eds.: E. Krause, W. Jäger,
Springer-Verlag 1999

the box was allowed to change its dimensions independently in each direction. But the question arose, whether an isotropic pressure is an adequate choice or whether different pressure values parallel und perpendicular to the bilayer normal have to be used. This led to the important question of the surface tension of a phospholipid bilayer [10]. While some argued, that a bilayer should have a vanishing surface tension [11], Feller and Pastor showed [13] that because of the periodic boundary conditions and the relatively small size of a simulation box these arguments are not valid for a MD–simulation. They gave an approximation of the surface tension to be applied, based on the statistical theory of bilayers. Chiu [14] on the other hand applied a surface tension which was taken from a lipid-monolayer–air-interface and which was therefore much higher than the surface tension proposed by Pastor and Feller. So in the moment there is no consensus on the value of the surface tension to be applied. Nevertheless the NγT ensemble is mostly used now for bilayer simulations. In this work γ is obtained from the approximation by Pastor and Feller.

Another crucial point in the simulation setup is the treatment of the electrostatic interactions. In the early days of bilayer simulation more or less elaborate cutoff-schemes were used, because the computer power was too limited to use more appropriate schemes like the Ewald summation. Some authors even argued, that Ewald summation would enhance the artifacts of the periodic boundaries [7]. But in the last years the computers have become faster and new methods for the efficient treatment of the long range electrostatic interactions, like the smooth particle mesh Ewald summation technique (SPME) [15] used in our simulation have emerged. Thus the computational advantages of a simple cutoff-scheme is not significant anymore (unfortunately, this is not really true for parallel simulations, because of the increased ammount of communication required by the more advanced schemes, which reduce the scalabilty of such algorithms).

2 Simulation Setup

The bilayer consists of 200 molecules of dipalmitoyl–*sn*–glycero–phosphatidylcholin (DPPC, see fig. 1) and 6125 water molecules, which corresponds to about 30 water molecules per lipid and a water weight fraction of c = 0.43, which is above the experimentally determined value of a fully hydrated multilamellar DPPC system in the L_α phase (where values of 0.36 and 0.40 are reported [7]). The AMBER forcefield [16] was used for the DPPC, except for the alkyl chains, where the all-atom OPLS [17] forcefield was employed. The missing parameters for the improper dihedral of the ester link were taken from CHARMM [18], which should cause no problems, because of the inflexibility of this region. The charges were calculated by the RESP procedure [19], fitting them to the electrostatic potential of dipropyl–*sn*–glycero–phosphatidylcholin. For the alkyl chains the charges from the OPLS forcefield

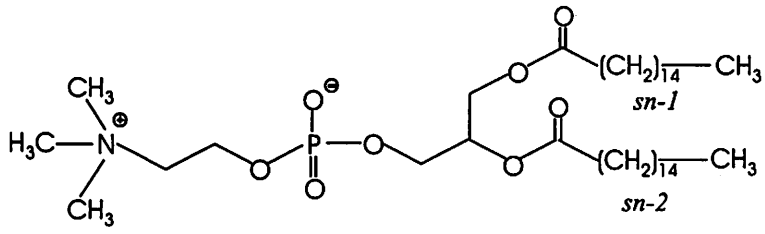


Fig. 1. The structure of the DPPC molecule

were used. The quantum chemical calculations of the electrostatic potential where performed on the HF-6-31G*-level with Gaussian 94 [20]. The water model was flexible SPC/E [21], which differes from the normal SPC/E model [22] in that is has an angle bending term. A simulation timestep of 1 fs was chosen. The pressure and temperature was controlled by a Berendsen thermostat [23] with a coupling constant of 2.0 ps for the pressure and 0.2 ps for the temperature during the equilibration procedure and 100 ps and 1 ps for the rest of the simulation. The three box dimensions were allowed to change independently. The simulation was carried out at a temperature of 50 ^{0}C and a pressure of 1 atm in the direction of the bilayer normal (the y-axis) and -11 atm in the direction of the x-, and z-axis, which corresponds to a surface tension of $9.12 \cdot 10^{-3} \mathrm{Nm}^{-1}$. The electrostatic interactions were calculated by the SMPE method with an interpolation order of 4 and a grid spacing of 0.1 nm. A cutoff distance of 0.9 nm was chosen for the real space part of the interactions.

The system itself was prepared as follows: A single DPPC molecule was energy minimized, while having a slight repulsive force applied between the N-atom and the terminal C-atoms of the alkyl chains to bring the molecule in a linear form. Then the bilayer was build by inserting the minimized DPPC molecules (randomly rotated around the y-axis) on a 10 × 10 × 2 grid in such a manner, that there were no unfavourable interactions between the molecules. This system was then simulated for 10 ps at a temperature of 10 K and an isotropic pressure of 1.0 atm. After this a water cap constisting of 6125 molecules (where the x- and z-dimensions were the same as the dimensions of the bilayer box) was added. This system was then equilibrated for 1.2 ns at the temperatur and pressure given above. After this equilibration a production run consisting of 600 ps was performed.

3 Computational Details

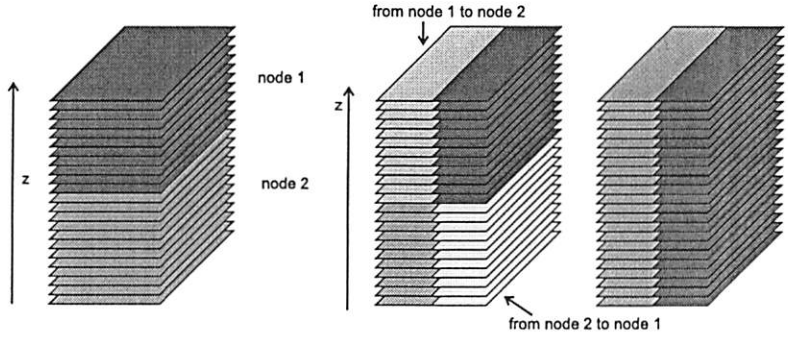
Because of the exceeding system size of 44375 interaction sites, the MOSCITO [1] simulation program was parallelized on the basis of the replicated data scheme. This can easily be incorporated into an existing simulation code.

Moreover the SPME method more or less requires a replicated data approach. Unfortunately this approach does not scale very well to a large number of processors, because the amount of communication per processor does not decrease with increasing number of processors (in fact it's slightly increasing because of the additional communication for the temperature and pressure scaling).

The replicated data approach works as follows: At the beginning of a timestep each node has the coordinates of all atoms , so each node could calculate all forces. This allows to partition the force calculation in any way that seems appropriate and this flexibility is the the main advantage of the replicated data scheme. After the forces are calculated the trajectory integration step and the SHAKE [24] algorithm need to be done. Accordingly we assign to each node a subset of molecules for which it has to know the forces and velocities. After the pair force calculations, each node sends a proper subset to any other node, so that at the end of this step each node knows all the forces on all atoms for which it has accomplish the integration and SHAKE step. After the integration is done, each node sends its subset of new coordinates to any other node, so that finally each node has the new coordinates of all atoms and the next timestep can be performed. As one can see, the amount of data that has to be sent or received by a node does not depend on the number of nodes but scales linearly with the number of atoms (to be more precise we have to say that it tends toward a constant as the number of nodes is increased). This means, that this approach is not feasible for a calculation on many nodes (see [27] for another approaches, which scales better, but is more complicated to implement). The maximum number of nodes one can use efficiently, strongly depends on the speed of the underlying network and the speed of the processors (the higher the network speed or the slower the processors, the higher the number of nodes that can be used).

The simple picture given above is only true, if one uses a cutoff-based treatment of the electrostatic interactions. The use of more appropriate methods, like the SPME technique or the Fast Multipole Method (FMM) [26] complicates this picture significantly, because additional communication during the force caculation is required. As we use the SPME method in our program, we will shortly discuss this method from a more communication centred point of view (for a more detailed discussion see [25]). The SPME method, like the normal Ewald summation, splits the electrostatic interactions in a real space part (which can be calculated with a normal cutoff-scheme) and a sum in the reciprocal space. For the calculation of the reciprocal space part the charges are 'interpolated' on a grid by cardinal B-splines of a given order. Then a 3-D FFT is performed, the grid points are modified and a back 3-D FFT transformation is added. After these steps the forces on the atoms due to the reciprocal part can be interpolated from the values of the grid. The first part of this scheme, the filling of the grid, is relativly easy to parallelize (in fact it requires no communication at all if one accepts a few redundant calculations, aw we have done in our case). We assigned to each node incorporated

in the calculation of the reciprocal part of the SPME (nodes N_{rec}) a region of the grid with a z-coordinate in a given range (see fig. 2(a)). Because of the



(a) Initial partitioning (b) Communication

Fig. 2. The parallelization of the recprocal part of the SMPE

use of cardinal B-splines for the grid assignment, one has to calculate only the B-spline coefficients of the atoms which are in the partition of the node or in its nearest neighbourhood. After the end of this step, every node N_{rec} has the values of all grid points in its slice. The next step is the 3-D Fast Fourier Transformation, which is split into two parts. The first part consists of a 2-D FFT in the the x-y-plane. This can be done locally on each node, because only grid points calculated by the node in the first step are involved. But for the second part, a FFT along the z-axis, parts of the grid must be exchanged between the nodes (see fig. 2(b)). This step requires additional und huge communication between the nodes N_{rec} and this is the limiting factor in our simulation. The next step involves a modification of the grid values and can be done without communication at all. After this a back 3-D FFT is performed analogous to the first 3-D FFT (and with an equal ammount of communication) and in the last step the forces are calculated from the grid coefficients, which requires only little communication between some nodes to exchange the grid coefficients in the border region. As one can see, it is desirable to do the reciprocal part of the SPME on as few nodes as possible, because of the additional communication required. In fact it can be advantageous to do it only on one node and therefore have no additional communication, even when this calculation takes longer than the other nodes need for the calculation of the other forces (see below).

A problem that arises when performing parallel calculations is that of load-balancing. Because the time required for a timestep is limited by the slowest node (which can be the slowest, because it has a weaker CPU or

because it has to do more work than the other nodes). Therefore it is necessary to partition the work in a manner, that ensures that all nodes need the same time for their calculations. In the case of the MD-Simulation it is slightly more complicated, because there are two times to consider. The first time is the time needed to calculate the forces. This time has to be equal among all nodes, because each node has to wait for all other nodes in order to get the forces needed to do the integration and SHAKE step. The second time is the time needed to accomplish these steps, which are normally much shorter than the time needed for the force calculation, so that an unbalance during these steps is not that critical und is not considered here further. On the other hand it is not known a priori, what runtime characteristics a given partition will have. In our case this is mainly due to the fact, that the number density in the simulation box varies to some degree and therefore the number of nearest neighbours varies and with it the number of forces to calculate. Because of this we employed a dynamic loadbalancing scheme, which works as follows: First, all nodes are assigned to calculate the force acting on an equal number of atoms (because we use Newton's third law, some forces for a particular atom may in fact be computed on a different node than that node, which was assigned to the atom). One node additionally has to calculate the reciprocal SPME term. Now the time t_{real} required for the calculation of the real space part of the forces is taken on each node, as is the time t_{rec} to calculate the reciprocal part of the SPME. Next the partitioning is adjusted accoring to the deviation of $t_{real} + t_{rec}$ of each node from the mean value (if the node needed more time than the average it will get fewer forces to calculate and vice versa). Note that only the partition for the calculation of the real space part (which affects only t_{real}) is modified during this step. The partitioning of the reciprocal SPME part is treated seperately, because of the two additional communication steps needed. This requires, that all the nodes N_{rec} have to reach these two communication steps at the same time, because otherwise the other nodes are idle, until the last node is ready. But this goal is hardly ever reached when the particle distribution along the z-axis is not uniform. Fortunately a rather uniform particle distribution along the z-axis is given in our simulation, so this load balancing step works analogous to the balancing of the real space force calculation.

The assignment of the number of nodes to the calculation of the reciprocal force requires a special treatment. This is due to the fact, that an increase in this number can give an increase in the time each node needs for the overall calculation. This can be illustrated by the following example: Let's assume that we are working on three nodes. Node 1 is doing only the reciprocal force calculations, while the others are doing the rest. Node 1 needs 5.0 seconds for its calculations and node 2 and 3 need 4.5 seconds. Now it seems a good idea to split the work of the reciprocal force calculation among node 1 and node 2. But this means, that node 1 and 2 have to do two additional communication steps. If we assume, that the additional communication takes 1 second per node (this value is reasonable on computers like an SP2), after load balancing

we get the following result: Node 1 and 2 need 5.33 seconds (2.5 seconds for the SPME, 1 second for the additional communication and 1.833 seconds for the calculations of the other forces) and node 3 needs 5.33 seconds for the calculation of the other forces. As one can see, the overall time increased. If the third node is also assigned to do a part of the SPME calculation the communication time per node in the reciprocal part will stay approximately constant. This assumption may seem questionable, because the ammount of data each node must communicate is slightly decreasing by about 10 percent, but on the other hand the number of communications per node doubles, and therefore the time lost due to the latency time needed to start a communication. Another factor is that the chance of load unbalace increases, because now three nodes instead of two have to reach the communication steps at the same time. With a reasonable value for the communication time of 1 seconds every node will need 5.67 seconds (1.67 seconds for the SPME, 1.0 seconds for the communication + 3 seconds for the calculation of the other forces). As one can see, the overall time increased instead of decrased as the SPME calculation was spread across more nodes and this would even be the case if the communication time per node would be decreased by the above mentioned 10 percent, which would result in an overall time of 5.57 seconds. This example shows that one has to use specific heuristics for the assignment of nodes to the reciprocal force calculation in addition to the normal load balancing mechanism, because these effects can not be captured by a simple scheme, especially when the needed communication times can not be predicted a priori.

Because of the not optimal scaling of the replicated data algorithm itself and the additional problems of the parallelization of the SPME method, we normally used only 10 nodes in our calculations. Table 1 and fig. 3 shows the speedups depending on the number of nodes used. As one can see it makes not much sense to use more than 10 nodes, especially if one has only a limited amount of computer time available and every derivation from the optimal speedup means a loss of valuable computer time.

4 Results

This simulation run was finished recently. Since then we are evaluating the trajectory (see fig. 4 for a snapshot of the simulation box). First results are given below:

The system equilibrated to an area per lipid of 0.56 nm^2 which is at the lower end of the experimentally determined values. But as Nagle [8] argued, the most likely experimental value of the area per lipid is around 0.62 nm^2, so the simulation value is significantly too low (about 10 percent). Another indication that this area is too small arises from the deuterium order parameters -S_{DC}, which strongly depend on the area per lipid [9] and proved to be too high in our simulation (see fig. 5). This implies further, that the mul-

nodes	speedup	time per timestep
1	1	43.2
2	1.95	22.1
3	2.9	14.9
4	3.8	11.4
6	5.6	7.7
8	7.3	5.9
10	8.9	4.8
12	9.8	4.4
14	10.6	4.1
16	11.2	3.9

Table 1. Dependence of the speedup on the number of nodes

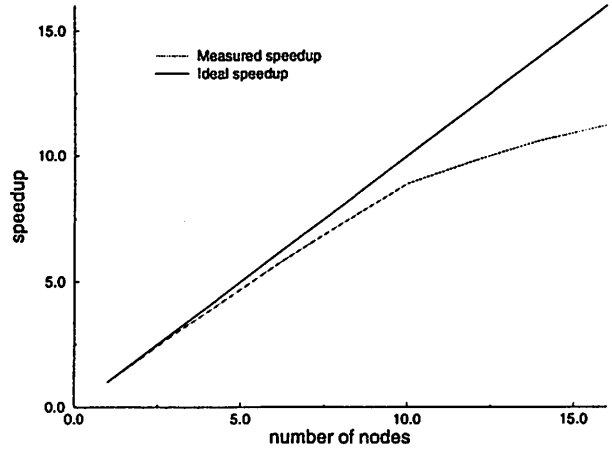


Fig. 3. Dependence of the speedup on the number of nodes

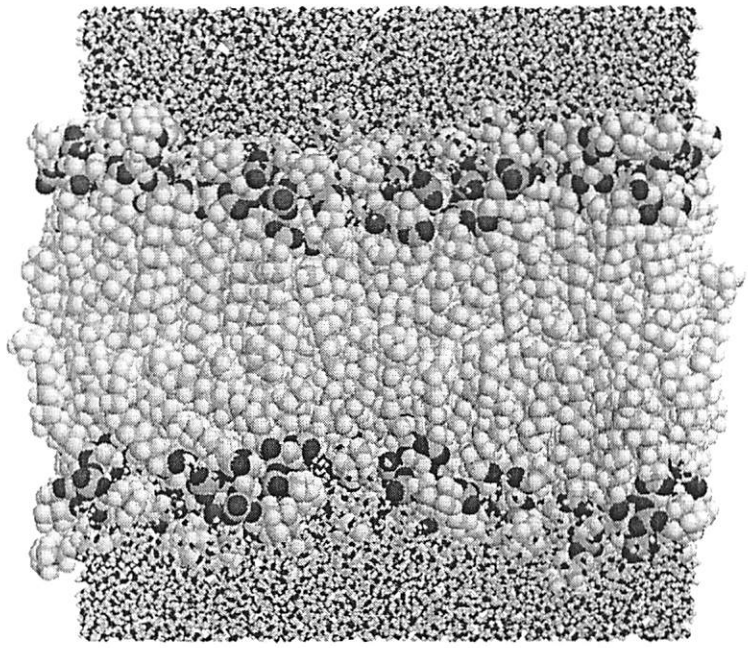


Fig. 4. Snapshot of the simulation box

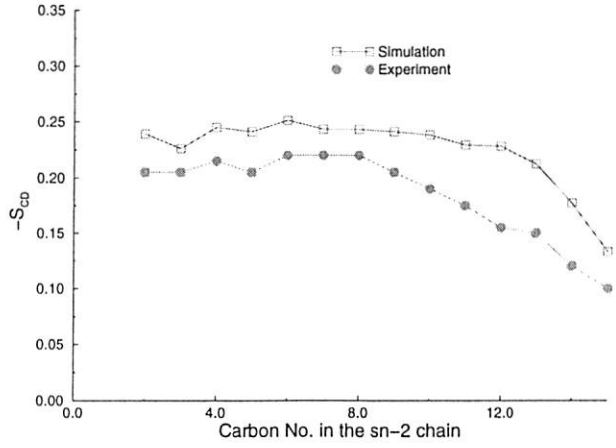


Fig. 5. Experimental and simulated S_{CD} parameters for the sn-2 chain

tilamellar repeat distance is too large (7.6 nm) and clearly shows, that the applied forcefield has some deficits. But the basic properties of a phospholipid membrane should nevertheless be captured by this simulation.

To ensure that the reduce area per lipid is not the result of a decoupling of the temperature of the DPPC molecules from the temperature of the water molecules, the temperatures of both molecule types were determined (see fig. 6). As one can see, the mean temperature of the two molecule types

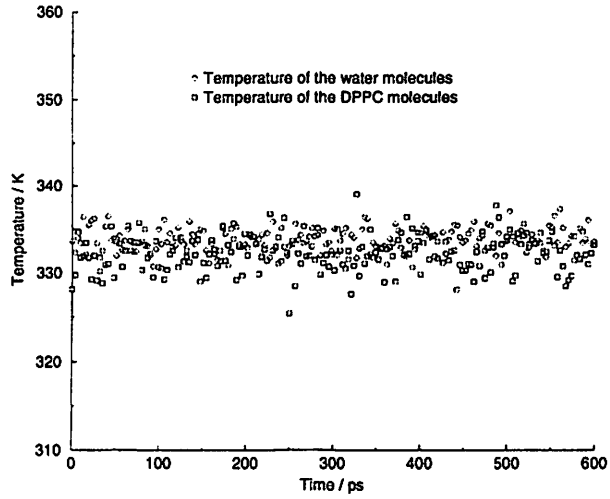


Fig. 6. Temperature of the water and the DPPC molecules throughout the simlation

are slightly different (the average temperature of the DPPC is about 1.5 K smaller than the temperature of the water molecules). But this difference is not large enough to explain the deminished value of the area per lipid as an effect of the lower temperature of the DPPC molecules (this would require a temperature difference of 10 K or more).

Fig. 7 shows the mean square displacement (MSD) of the center of mass (COM) of the DPPC molecules in y direction (this is parallel to the bilayer normal) and in the xz plane, which is perpendicular to the bilayer normal. Apart from the first 50 ps the curves are rather linear (see the dashed lines which are calculated by a linear regression in the range between 100 ps and 350 ps). The calculated diffusion constants are $D_y = 4.45 \times 10^{-7} cm^2 s^{-1}$ in the y direction and $D_{xz} = 2.67 \times 10^{-7} cm^2 s^{-1}$ in the xz plane. It may seem surprising that the lateral diffusion coefficient D_{xz} is smaller than the transversal diffusion coefficient D_y, because experiments suggest the opposite behaviour. But one has to remember that the simulation time is too short to

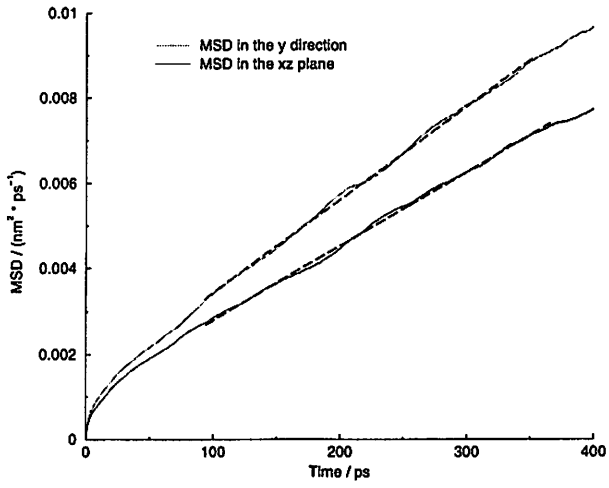


Fig. 7. Mean square displacement of the center of mass of the DPPC molecules in y direction and the xz plane

observe a real transversal diffusion and the value of D_y is only determind by undulation like motions, so that D_y will be significantly lower given a much longer simulation time. The value of D_{xz} is in excellent aggrement with a value recently obtained by Essman [5] from a 10 ns simulation.

Fig. 8 shows the mean square displacement of the COM of the water molecules in x, y and z direction and in the xz plane. As expected the mean square displacements in x and z direction are practically identical. The diffusion in y direction on the other hand is significantly lower especially for times above 30 ps, which results in a small curvature in the mean square displacement curve. This can easily explained by the fact, that the diffusion in y direction is hindered by the bilayer (see also the dependence of the overall diffusion coefficient on the position discussed below). The diffusion in the xz plane shows no peculiarities, as the MSD in x and z direction already suggest. To examine the diffusional behaviour further, the diffusion coefficient of the water molecules were calculated as a function of the position along the y-coordinate. The results are given in fig. 9. In this figure we also show the particle density of the water molecules as a function of the y coodinat. As one can see the diffusion constants drop from a value of about 3.35 cm^2s^{-1} in the bulk water between the bilayer to very low values for water molecules in the interior of the bilayer. It has to be mentioned that the values of the diffusion coefficients in regions with low water content could not be determined with high accuracy, because of the small ammount of water molecules found in this region. Another point to mention is the fact that no water molecules are found in the region of the alkyl chains. The low diffusion coefficients in the

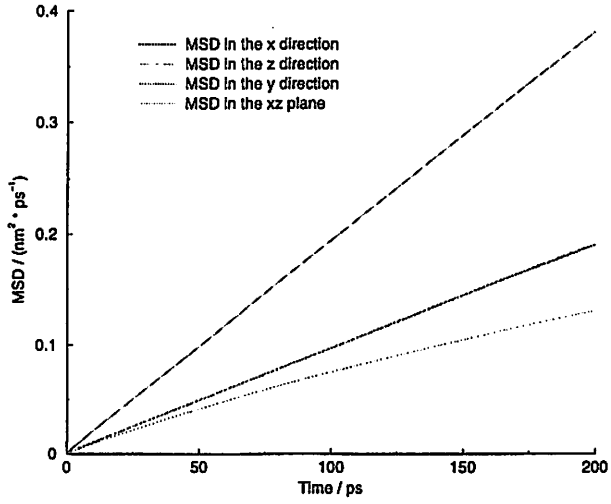


Fig. 8. Mean square displacement of the center of mass of the water molecules in x, y and z direction and the xz plane

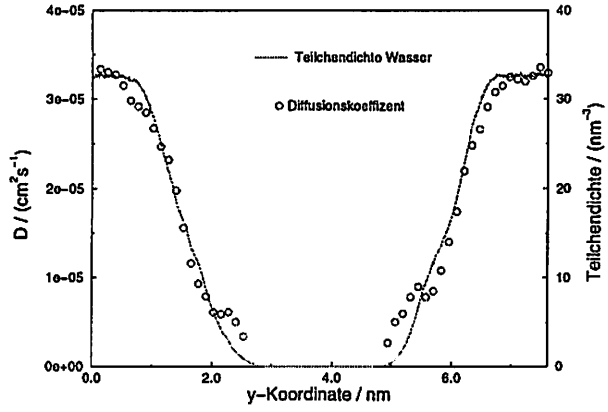


Fig. 9. Diffusion coefficient and particle density of water as a function of its y-coordinate

bilayer can be explained by the strong electrostatic interactions of the water molecules with the choline and phosphate groups. The plateau region visible at y coordinates of about 2 nm and 5.5 nm respectivetly can be explained by a 'trapping' of a single water molecule between two ester groups, but this has to be examined further to give a definite explanation. Another phenomenon that has to be examined further is the strong correlation of the water density and the diffusion coefficient.

As a last result some probability distributions of intramolecular distances between specific atoms are presented. In fig. 10 the probability density of the distance between the nitrogen atom in the choline group and the phosphorous atom in the phosphate group is shown. It can be seen, that there are

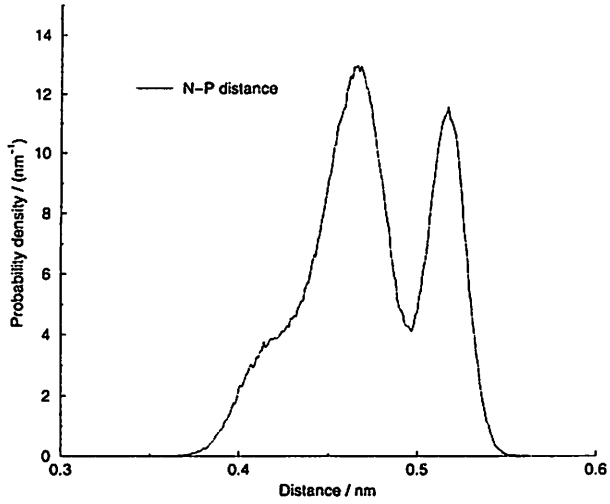


Fig. 10. Probability density of the intramolecular N–P distance

two peaks in the distribution which correspond to 2 different configurations of the DPPC headgroup. The second peak at about 0.52 nm corresponds to a stretched conformation of this region. Because of the positive net charge of the choline group and the negative net charge of the phosphate group this conformation may seem energeticly unfavorable in contrast to the conformation which corresponds to the first peak at 0.47 nm, where the charged centers are closer to one another. But one has to take into account, that the conformation of the second peak allows a greater contact with the surrounding water molecules, so that both configurations can be observed in comparable ammounts. It will be interesting to determnine the lifetime of the two configurations and the mechanism of the transformation between them. Such investigations are underway.

In fig. 11 distance distributions with a different qualitative behaviour are shown. Here the distances of phosphorous atom to the three carbon atoms

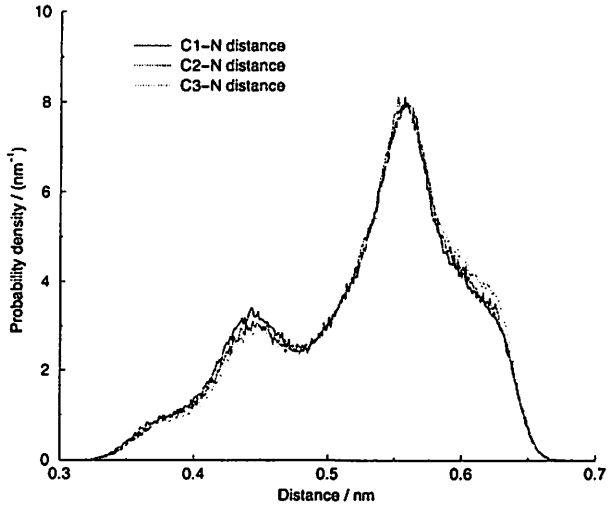


Fig. 11. Probability density of the intramolecular distance of the N-atom to the three carbon atoms of tetramethylammonium group

in the choline group are considered. As can be seen, the curves for the three different carbon atoms are virtually the same, which is a hint that the rotation of this group is not hindered very much. Once again there are two main peaks in this distribution, but this time the area under the first peak is less than half the area under the second peak. This is an indication for a conformation where one of the three carbon atoms is directed to the phosphorous atom, while the other two are showing in the opposite direction.

The last distance probability distributions, shown in fig. 12, give an impression of the overall shape of the DPPC moleules. The distributions of the distance between the nitrogen atom in the headgroup and the terminal carbon atoms of the two alkyl chains show that the DPPC molecule normally exist in a stretched conformation. But a considerable probability for a much shorter N-C distance can be observed. These shorter distanances correspond to configurations where the alkyl chains are no longer side by side, but where one (or even both) chains are orientated more perpendicular to the bilayer normal. This can be verified by the distribution of the distance between the two terminal carbon atoms, which is also shown in fig. 12. While the peak at about 0.5 nm suggests that the chains are close to each other most of the time, there are also conformations where this is no longer the case. This shows that there is a considerable ammount of disorder in the alkyl chains,

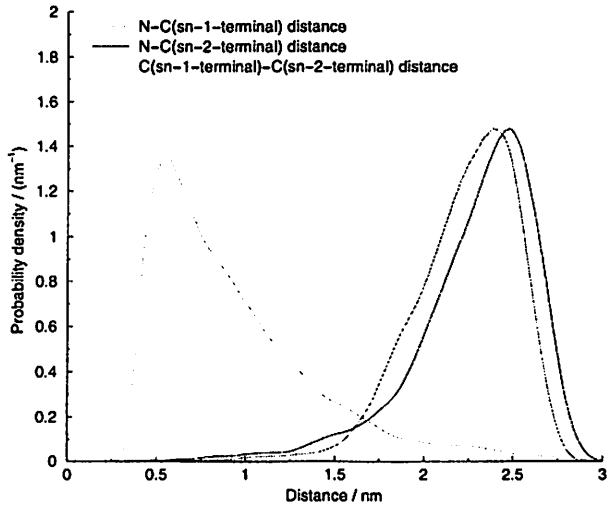


Fig. 12. Probability density of the intramolecular distance of the N-atom to the terminal carbon atoms of the sn-1 and sn-2 alkyl chains

which would for example not be the case, if the bilayer was in the gel phase, which is the stable form of the membrane at temperatures lower than about 40 ^{0} C.

References

1. Paschek, D.: Performing Moleculat Dynamics Simulations with MOSCITO. University of Dortmund, 1998
2. Pastor, R.: Molecular dynamics and Monte Carlo simulations of lipid bilayers. Curr. Opin. Struct. Biol. **4** (1996) 486–492
3. Tieleman, D. P., Marrink, S. J., Berendsen, H. J. C.: A computer perspective of membranes: molecular dynamics studies of lipid bilayer systems. Biochim. Biophys. Acta. **133** (1997) 235–270
4. Nagle, J. F. R., Zhang, R. , Tristam-Nagle, S., Sun, W., Petrahce, H., Suter, R. M.: X-ray structure determination of fully hydrated L_α phase DPPC bilayers. Biophys J. **70** (1996) 1419–1431
5. Essmann, U., Berkowitz, M. L.: Dynamic Proprties of Phospholipid Bilayers from Computer Simulation. Biophys. J. **76** (1999) 2081–2089
6. Husslein, T., Newns, D. M., Pattnaik, P. C., Zhong, Q. Moore, P. B., Klein, M. L.: Constant pressure and temperature molecular-dynamics simulation of the hydrated diphytanolpshophatidylcholine lipid bilayer. J. Chem. Phys. **109** (1998) 2826–2832
7. Tieleman, D. P., Berendsen, H. J. C.: Molecular dynamics simulation of a fully hydrated dipalmitoylphosphatidylcholine bilayer with different macroscopic boundary conditions and parameters. J. Chem. Phys. **105** (1996) 4871–4880)

8. Nagle, J. F.: Area/lipid of bilayers from NMR Biophys. J. **64** (1993) 1476–1481
9. Feller, S. E., Venable, R. M., Pastor, R. W.: Computer Simulations of a DPPC Phospholipid Bilayer: Structural Changes as a Function of Molecular Surface Area. Langmuir **13** (1997) 6555–6561
10. Roux, B.: Commentary: Surface Tension of Biomembranes. Biophys. J. **73** (1996) 1346–1347
11. Jähning, F.: What is the Surface Tension of a Lipid Bilayer Membrane? Biophys. J. **73** (1996) 1348–1349
12. Jakobsson, E., Subramaniam, S., Scott, H. L.: Strategic Issues in Molecular Dynamics Simulations of Membranes. Merz, K., Roux, B. (editors): Biological Membranes, Birkhäuser Boston 1996
13. Feller, S. E., Pastor, R. W.: On simulating lipid bilayers with an applied surface tension: periodic boundary conditions and undulations. Biophys. J. **71** (1996) 1350–1355
14. Chiu, S. -W., Clark, M., Balaji, V., Subramaniam, S., Scott, H. L., Jakobsson, E. : Incorporation of Surface Tension into Molecular Dynamics Simulation of an Interface: A Fluid Phase Lipid Bilayer Membrane. Biophys. J. **69** (1995) 1230–1245
15. Essmann, U., Perera, L., Berkowitz, M. L., Darden, T., Lee, H., Pdersen, L. G.: A smooth particle mesh Ewald method. J. Chem. Phys. **103** (1995) 8577–8593
16. Cornell, W. D., Cieplaj, P., Bayly, C. I., Gould, I. R., Merz, K. M., Ferguson, D. M., Spellmeyer, D. C., Fox, T., Caldwell, J. W., Kollman, P. A.: A Second Generation Force Field for the Simulation of Proreins, Nucleic Acids, and Organic Molecules. J. Am. Chem. Soc **117** (1995) 5179–5197
17. Kaminski, G., Duffy, E. M., Matsui, T., Jorgensen, W. L.: Free Energies of Hydration and Pure Liquid Properties of Hydrocarbons from the OPLS All-Atom Model. J. Phys. Chem. **98** (1994) 13077–13082
18. Schlenkrich, M., Brickmann, J., MacKerrel, A. D., Karplus, M.: An Empirical Potential Energy Function for Phospholipids: Criteria for Parameter Optimization and Applications. K. Merz, Roux, B. (editors): Biological Membranes Birkhäuser Boston 1996
19. Bayly. C. I., Cieplyk, P., Cornell, W. D., Kollman, P. A.: A well-behaved electrostatic potential based method using charge restraints for deriving atomic charges: The RESP model. J. Phys. Chem **97** (1993) 10269–10280
20. Frisch, M. J., Trucks, G. W., Schlegela, H. B., Gill, P. M. W., Johnson, B. G., Robb, M. A., Cheeseman, J. R., Keith, T., Petersson, G. A., Montgomery, J. A., Raghavachari, K., Al-Laham, M. A., Zakrzewski, V. G., Ortiz, J. V., Foresman, J. B., Cioslowski, J., Stefanov, B. B., Nanayakkara, A., Challacombe, Peng, M. C. Y., Ayala, P. Y., Chen, W., Wong, M. W., Andres, J. L., Replogle, E. S., Gomperts, R., Martin, R. L., Fox, D. J., Binkley, J. S., Defrees, D. J., Baker, J., Stewart, J. P., Head-Gordon, M., Gonzalez, C., Pople, J. A.: Gaussian 94, Revision C.3 Gaussian, Inc., Pittsburgh PA, 1995.
21. Alejandre. J., Tildesley, D. J., Chapela, G. A.: Molecular dynamics simulation of the orthobaric densities and surface tension of water. J. Chem. Phys. **102** (1995) 4574–4583
22. Berendsen, H. J. C., Grigera, J. R., Straatsma, T. P.: The missing term in effective pair potentials. J. Phys. Chem. **91** (1987) 6269–6271
23. Berendsen, H. J. C., Postma, J. P. M., van Gunsteren, W. F., DiNiola, A., Haak, J. R.: Molecular dynamics with coupling to an external bath. J. Chem. Phys. **81** (1984) 3684–3690

24. Ryckeart, J. -P., Cicotti, G., Berendsen, H. J. C.: Numerical Integration of the Cartesian Equations of Motion of a System with Constraints: Molecular Dynamics of n-Alkanes. J. Comp. Phys. **23** (1976) 327–341
25. Crowley, M.F., Darden, T.A.; Cheatham, T.E., III, Deerfield, D.W., II Adventures in improving the scaling and accuracy of a parallel molecular dynamics program. J. Supercomp. **11** (1997) 255–278
26. Greengard, L., Rokhlin, V.: A FastAlgorithm for Particle Simulations. J. Comp. Phys **73** (1987) 325–348
27. Lim, K. -T., Brunett, S., Iotov, M., McClurg, R. B., Vaidehi, N., Dasgupta, S., Taylor, S, Goddard III, W. A. Molecular Dynamics for Very Large Systems on Massively Parralel Computers: The MPSim Program. J. Com. Chem. **18** (1997) 501-521