Computer Physics Communications 128 (2000) 550–557

Computer Physics Communications

www.elsevier.nl/locate/cpc

The GROMOS96 benchmarks for molecular simulation

Alexandre M.J.J. Bonvin [b,1], Alan E. Mark [c,2], Wilfred F. van Gunsteren [a,*]

[a] *Laboratory of Physical Chemistry, Swiss Federal Institute of Technology Zürich, ETH Zentrum, CH-8092 Zürich, Switzerland*
[b] *Bijvoet Center for Biomolecular Research, NMR Spectroscopy, Utrecht University, Padualaan 8, 3584 CH Utrecht, The Netherlands*
[c] *Laboratory of Biophysical Chemistry, University of Groningen, Nijenborgh 4, 9747 AG Groningen, The Netherlands*

Received 22 November 1999

Abstract

A set of biomolecular systems is presented, which can be used to benchmark the performance of simulation programs and computers. It is applied, using the GROMOS96 biomolecular simulation software, to a variety of computers. The dependence of computing time on a number of model and computational parameters is investigated. An extended pair list technique to select non-bonded interaction pairs and long-range interactions is shown to increase the efficiency by a factor 1.5 to 3 when compared to standard procedures. The benchmark results can be used to estimate the computer time required for simulation studies, and to evaluate the efficiency of various computers regarding molecular simulations. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Molecular dynamics; Non-bonded interactions; Cut-offs; Extended pair list

1. Introduction

Simulation of the behaviour of biomolecular systems on a computer is a steadily expanding area of theoretical and computational biochemistry. In molecular dynamics (MD) simulation, Newton's equations of motion for thousands of atoms are integrated forward in time using small, i.e. femtosecond size, time steps during which the forces on the atoms may be assumed to be nearly constant. Today, a simulation study of a small protein containing about 100 amino acid residues in water involves about 10^6 of such time steps covering a simulation period of 1 nsec. Although this seems sufficiently long to sample local, fast relaxing properties, it is too short to study slow, global processes such as protein folding and unfolding. In order to extend a simulation period as long as possible for a molecular system of a given size within a given time span, it is of paramount importance to use a very fast computer and a very efficient simulation program.

The time needed to complete a simulation project depends on a number of factors:

(1) Size of the molecular system, i.e. the number of atoms or particles or interaction sites.
(2) The length of the simulation that is needed to sufficiently sample the relevant motions of the system.
(3) The complexity of the potential energy function used, from which the forces on the atoms are derived.
(4) The spatial range of the forces between atoms.
(5) The settings of various interaction or computational parameters, such as the non-bonded interaction cut-off or the frequency of updating specific interactions.

[*] Corresponding author. E-mail: wfvgn@igc.phys.chem.ethz.ch.
[1] E-mail: abonvin@nmr.chem.uu.nl.
[2] E-mail: mark@chem.rug.nl.

0010-4655/00/$ – see front matter © 2000 Elsevier Science B.V. All rights reserved.
PII: S0010-4655(99)00540-8

(6) The handling of spatial boundary conditions, e.g., periodic versus vacuum.
(7) The spatial homogeneity of the system, i.e. whether the atom density is a smooth function of position or not.
(8) The characteristics of the computer used, e.g., its CPU, clock rate, memory handling, cache size.
(9) The structure of the program and algorithms involved.
(10) The programming language, compiler and compiler options used to produce an executable program.

In order to give an impression of the computing effort required to simulate a biomolecular system a benchmark has been formulated and carried out on a variety of computers using the Groningen Molecular Simulation GROMOS96 software [1,2]. A number of the factors mentioned above, which determine the time required for a simulation, have been investigated:

- size of the system (1),
- range of the forces (4),
- frequency of non-bonded force update (5),
- vacuum versus periodic boundaries (6),
- computer architecture (8),
- use of an extended pair list technique to select non-bonded interaction pairs (9).

The results of the GROMOS96 benchmark can be used to estimate the CPU time required for a typical biomolecular simulation project and to evaluate the efficiency or performance-to-price ratio for various computers regarding simulation of biomolecular systems.

2. Methods

The bulk of the computational time required by a simulation time step is used for calculating the non-bonded interactions, that is, for finding the nearest neighbor atoms and subsequently evaluating the van der Waals and electrostatic interaction terms for the obtained atoms pairs. Since the non-bonded interaction between atoms decreases with the distance between them, only interactions between atoms closer to each other than a certain cut-off distance R_{cp} are generally taken into account in simulations. The GROMOS96 force field makes use of the concept of *charge groups* to reduce the errors due to the application of a cut-off radius R_{cp}. The atoms that belong to a charge group are chosen such that their partial atomic charges add up to zero, except for fully charged groups like the side-chains of Arg or Asp where the partial atomic charges may add up to $+e$ or $-e$. When the (partial) atomic charges of a group of atoms add up to exactly zero, the leading term of the electrostatic interaction between two such groups is of dipolar ($1/r^3$) character and the error due to the application of a cut-off radius is reduced compared to the $1/r$ monopole contributions. Therefore in the GROMOS96 non-bonded interaction routines the cut-off radius is used to select nearest-neighbor charge groups. The simplest way to find the neighboring charge groups of a charge group, that is, the charge groups that lie within R_{cp}, is to scan all possible charge group pairs in the system. For a system consisting of N_{cg} charge groups, the number of pairs amounts to $\frac{1}{2}N_{\mathrm{cg}}^2$, which makes the computer time required for finding the neighbors in this way proportional to N_{cg}^2. Once the neighbors have been found, the time required for calculating the non-bonded interaction is proportional to N_{cg}. We note that the non-bonded interaction within a charge group may need to be calculated, when the charge group contains more than a few atoms.

In order to evaluate the non-bonded interaction with sufficient accuracy, a long cut-off radius R_{cl} has to be used; for biomolecular systems a value of at least 1.4 nm seems necessary [3]. But such a range is very expensive if pair interactions are evaluated at every MD step; the number of neighbor atoms within 1.4 nm will exceed 900. Therefore in GROMOS the non-bonded interaction can be evaluated using a *triple-range method*. The electrostatic interactions beyond the long-range cut-off R_{cl} – typically 1.4 nm – can be approximated by a Poisson–Boltzmann generalized *reaction field* term [4]. The non-bonded interactions are evaluated at every simulation step using the charge group pair list that is generated with a short-range cut-off radius R_{cp}. The longer-range non-bonded interactions, that is, those between charge groups at a distance longer than R_{cp} and smaller than R_{cl}, are evaluated less frequently, viz. only at every nth simulation step when also the pair list is updated. They are kept unchanged between these updates. In this way the long-range non-bonded forces can be approximately taken into account, without increasing the computing effort significantly, at the expense of neglecting the fluctuation of the forces beyond R_{cp} during n simulation steps.

The efficiency of the pair list generation on serial computers can be improved by using grid-search techniques [5–9], since the computational effort to select nearest-neighbor charge-groups using a spatial grid is proportional to N_{cg}. The improvement over the scanning of all charge group pair is only significant, however, for systems in which the non-bonded interaction cut-off radius R_{cl} is an order of magnitude shorter than the size of the simulated molecular system. For simulation of proteins in solution with $R_{\mathrm{cl}} \approx 1.4$ nm and periodic box lengths of about 5.0 nm, containing typically 10^4 atoms, the computational gain is very modest. Another method to speed-up the non-bonded pair list generation consists of keeping a second, "*extended pair-list*" generated using an extended cut-off radius $R_{\mathrm{cx}} > R_{\mathrm{cl}} \geqslant R_{\mathrm{cp}}$ and searching through this extended pair list to update the standard pair list instead of scanning all $\frac{1}{2}N_{\mathrm{cg}}^2$ charge group pairs in the system. Such an approach becomes especially effective for large systems. It is necessary to choose the extended cut-off R_{cx} larger than the long-range cut-off R_{cl} to ensure that the long-range forces are correctly evaluated. If the extended pair list is automatically updated once an atom has moved by more than the skin defined as $R_{\mathrm{cx}} - R_{\mathrm{cl}}$, the resulting trajectory is identical to that generated using a triple-range cut-off. In this case the update frequency will depend on the choice of R_{cx}, on the fastest moving particles in the system and on the temperature T. At $T = 300$ K, with an extended cut-off R_{cx} exceeding by 0.4 nm the long range cut-off R_{cl}, update frequencies around once per 100 MD time steps are obtained. A drawback of the extended pair list technique is the storage requirement, since, depending on the value of the various cut-offs, up to ten times the number of pairs in the standard non-bonded pair list has to be kept in memory. The idea of using a skin around the sphere with radius R_{cp} has been described previously [10,11], although in a slightly different way: the extended pair list generated with a cut-off R_{cx} is scanned at every MD step and the non-bonded interactions are only calculated for those atoms which lie within R_{cp}. In our implementation, the extended pair list is used to generate the normal non-bonded pair list with cut-off R_{cp} and to calculate the long-range forces within the long range cut-off R_{cl}, which is done every nth MD step (typically 5 to 10) and allows further reduction of the computing time.

3. Results

The six systems used for benchmarking are defined in Table 1. These are representative of the range of systems typically studied by MD simulations. The first two benchmarks are for a short cyclic peptide, cyclosporin A (11 residues), in vacuum taking all interactions into account ($R_{\mathrm{cp}} = \infty$) (I), and in water in a truncated octahedron under periodic boundary conditions (II). The next two are for a protein of 295 residues, thrombin, in vacuum (III) and in water in a truncated octahedron under periodic boundary conditions (IV). The last two benchmarks correspond to medium- (V) and large-sized (VI) pure water systems in a cubic box under periodic boundary conditions. For benchmarks II, III, V and VI a non-bonded cut-off $R_{\mathrm{cp}} = R_{\mathrm{cl}}$ of 0.8 nm was used. For benchmark IV, the long-range non-bonded interactions were calculated up to 1.4 nm (R_{cl}). In all benchmarks including explicit solvent, a reaction field contribution [4] was added to the atomic forces. The extended cut-off R_{cx} was chosen such that $R_{\mathrm{cx}} = R_{\mathrm{cl}} + 0.4$ nm, i.e. 1.2 nm for benchmarks II, III, V and VI and 1.8 nm for benchmark IV.

Results on a variety of computers for the standard and extended pair list versions are listed in Tables 3 and 4, respectively. The operating systems, compiler flags and spec fp95 of the various systems are listed in Table 2. A comparison of Tables 3 and 4 reveals that the use of an extended pair list allows a significant reduction of the computing time, especially for large systems. The speed-up is however very much dependent on the choice of parameters for the non-bonded interactions. For benchmark VI, the large water box system, a speed-up of a factor 2 to 3 depending on the type of computer is achieved using the extended pair list. For the thrombin in water benchmark (IV), however, because of the use of a long range cut-off, speed-ups of only a factor 1.5 to 2 are achieved. Because of the long range cut-off R_{cl} of 1.4 nm, an extended cut-off R_{cx} of 1.8 nm had to be used in benchmark IV, resulting in a much larger extended pair list ($\sim$10 times the normal pair list covering $R_{\mathrm{cp}} = 0.8$ nm) than when a single cut-off is used

Table 1
GROMOS96 benchmark systems

Benchmark	Molecules	N_{sm}	N_{sa}	N_{solvm}	N_{solva}	N_{a}	N_{cg}	Boundary condition	N_{MD}
I	cyclosporin A	1	90	0	0	90	40	vacuo	1000
II	cyclosporin A in water	1	90	764	2298	2388	804	octa	100
III	thrombin	1	3078	0	0	3078	1285	vacuo	100
IV	thrombin in water	1	3078	5427	16281	19359	6712	octa	10
V	H_2O (medium)	0	0	1728	5184	5184	1728	cubic	100
VI	H_2O (large)	0	0	13824	41472	41472	13824	cubic	10

N_{sm} = number of solute molecules, N_{sa} = number of solute atoms, N_{solvm} = number of solvent molecules, N_{solva} = number of solvent atoms, N_{a} = total number of atoms, N_{cg} = total number of charge groups, N_{MD} = number of MD time steps. The boundary conditions are: vacuo = vacuum boundary condition, octa = periodic truncated octahedron and cubic = periodic cubic box. The integration time step was 0.002 fs. The non-bonded pair list was updated every 5 steps. The non-bonded cut-off was infinite ($R_{\mathrm{cp}} = R_{\mathrm{cl}} = \infty$) for benchmark I, $R_{\mathrm{cp}} = R_{\mathrm{cl}} = 0.8$ nm for benchmarks II, III, V and VI. A twin-range cut-off of $R_{\mathrm{cp}} = 0.8$ and $R_{\mathrm{cl}} = 1.4$ nm was used for benchmark IV. All simulations were performed at constant particle number, volume and temperature (N, V, T).

like in benchmark VI ($R_{\mathrm{ex}} = 1.2$ nm resulting in an extended pair list about 4.5 times longer than the normal pair list). In addition, more time is spent evaluating the long-range forces.

To assess the effect of the non-bonded cut-off on the computing time, benchmarks IV and V were run for increasing R_{cp} values (Fig. 1). The long-range cut-off R_{cl} was kept constant (1.4 nm) in benchmark IV. When no long-range contribution is evaluated, the computing time increases almost linearly with the number of pairs in the non-bonded pair list. A cut-off increase of a factor two results in a time increase of a factor 6 and an increase of the size of the non-bonded pair list by a factor 8. When a long-range contribution is calculated as in benchmark IV the computing time increase as function of the non-bonded cut-off is less pronounced (factor 2 only), the increase of the non-bonded pair list being compensated by a smaller number of long-range force evaluations. The update frequency of the non-bonded pair list and the long range forces also affects the computing time (Fig. 2). Only minor speed-ups are obtained for update frequencies below once per 5 MD time steps while heating effects start appearing which affect the precision of the simulations. This is especially true for simulations in water using long-range forces where water reorientation becomes a source of heating with a once per 10 steps or lower update frequency. Updating more than once per 5 steps, the computing time starts increasing significantly, especially when long-range

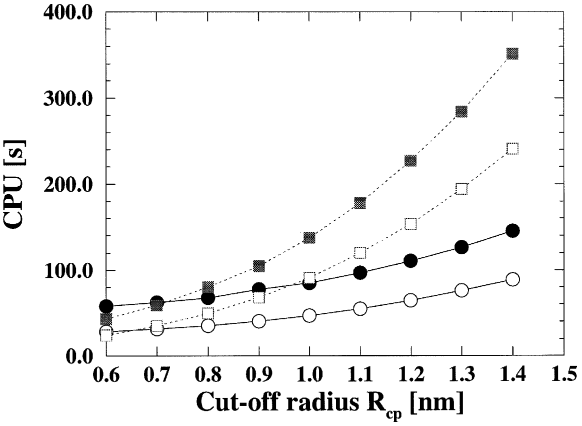

Fig. 1. CPU in seconds on a SGI impact R10000, 195 MHz, as a function of the non-bonded cut-off radius R_{cp}. For benchmarks IV (circles), the long-range cut-off radius R_{cl} was kept constant at $R_{\mathrm{cl}} = 1.4$ nm, while for benchmark V (squares), it was taken equal to R_{cp} ($R_{\mathrm{cl}} = R_{\mathrm{cp}}$). See also Table 1 for the values of the other parameters. Results using the standard and extended pair-list versions (see text) are indicated by filled and open symbols, respectively.

Table 2
Description of the computers on which the GROMOS96 benchmarks were run

Computer	Operating system	SPEC fp95[a]	Compiler version and flags	
CRAY YMP/C916	UNICOS 9.0.2.2	not available	cf77, 6.3	-Wf” -dp” -Oscalar3 -Ovector0 -Oinline2 -Wf” -dp -o aggress -em” -Zv (special flags for vector non-bonded routines)
DEC alpha 600, 300 MHz	Digital UNIX T4.0D-2	12.2	f77	-O5 -r8 -u -tune host
DEC 21164, 600 MHz	Redhat Linux 5.1, 2.034	27.0	f77	-O5 -r8 -u -tune ev56 -non_shared (compiled under Digital UNIX T4.0D-2)
DEC 21264, AS8400, 575 MHz	OSF V4.0	47.1	f77	-O5 -r8 -u -tune host
IBM 6000/397, 160 MHz	AIX 4.2	26.6	xlf	-O3 -u
Pentium Pro, 200 MHz	Redhat Linux 5.1, 2.034	6.8	egcs 1.0.3	-O3
Pentium II, 450 MHz	Solaris 2.6	13.3	PGI f77 3.0	-Munroll -O3 -tp p6 -r8 -Mnoframe -Mdclchk -byteswapio
HP 9000-735, 99 MHz	HP-UX B.10.20	3.4	f77	+O3 -R8
SGI O2, R5000, 180 MHz	IRIX 6.3	5.4	f77, 7.1	-n32 -O3 -r8 -OPT:roundoff=3: IEEE_arithmetic=3: fast_sqrt=OFF
SGI O2, R10000, 175 MHz		8.8		
SGI PowerChallenge R10000, 194 MHz	IRIX64 6.2	12.4		
SGI Impact, R10000, 195 MHz		13.8		
SGI Origin200, 180 MHz	IRIX64 6.4	15.6	f77, 7.1	-mips3 -n32 -O3 -r8 -r5000 -OPT:roundoff=3: IEEE_arithmetic=3: fast_sqrt=OFF
SGI Octane, 195 MHz		17.0		
Sun Ultra-1, 200 MHz	Solaris 2.6	9.4	f77, 4.2	-r8 -native -O4 -libmil -dalign -xlibmopt -depend -unroll=4
Sun Ultra-10, 333 MHz		12.9		
Sun Ultra-30, 300 MHz		14.9		

[a] SPECfp95 obtained from the Standard Performance Evaluation Corporation (http://www.spec.org).

Table 3
Computer CPU time [s] required for MD simulations

Computer	Benchmark system					
	I	II	III	IV	V	VI
CRAY YMP/C916 (using special vector version)	8.6	24.6	28.2	29.8	50.3	46.9
DEC alpha 600, 300 MHz	8.8	39.6	38.1	84.4	89.0	151.4
DEC 21164, 600 MHz, Linux	3.9	20.9	16.2	46.7	47.0	81.7
DEC 21264, AS8400, 575 MHz	2.4	11.4	9.9	24.3	24.8	39.0
IBM 6000/397, 160 MHz	8.7	51.8	38.6	114.8	104.3	173.6
Pentium Pro, 200 MHz, Linux	11.5	63.7	46.6	122.7	151.7	222.2
Pentium II, 450 MHz, Solaris 2.6	4.2	22.5	18.7	42.5	48.2	73.7
HP 9000-735, 99 MHz	18.0	96.9	75.7	218.4	210.5	374.2
SGI O2, R5000, 180 MHz	14.6	69.3	60.3	132.5	150.1	252.8
SGI O2, R10000, 175 MHz	7.5	39.9	32.8	78.1	87.6	149.5
SGI PowerChallenge R10000, 194 MHz	7.8	43.8	33.4	77.4	97.6	156.8
SGI Impact, R10000, 195 MHz	6.7	35.8	28.4	69.4	78.7	130.8
SGI Origin200, 180 MHz	6.9	36.0	29.9	71.2	79.3	135.5
SGI Octane, 195 MHz	6.4	33.3	28.1	66.0	73.4	125.8
Sun Ultra-1, 200 MHz	11.6	49.3	48.1	104.9	105.6	166.6
Sun Ultra-10, 333 MHz	7.4	33.8	30.7	68.9	74.0	128.8
Sun Ultra-30, 300 MHz	8.0	34.3	34.5	72.6	73.9	126.7

All benchmarks were run with the standard GROMOS96 version [1,2]. See Table 1 for benchmarks definitions and Table 2 for a description of the operating system, compiler options used and specfp95 values for the various computers. The CPU timing will depend on the machine configuration, the compiler options used, the operating system and the structure of the GROMOS code.

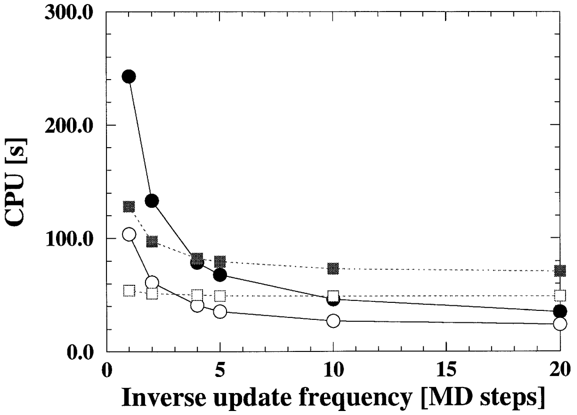

Fig. 2. CPU in seconds on a SGI impact R10000, 195 MHz, as a function of the number of MD time steps between non-bonded pair list updates. See also Table 1 for the values of the other parameters and the caption of Fig. 1 for definition of the symbols.

forces need to be evaluated (filled circles in Fig. 2). The use of an extended pair list almost completely removes the dependence of the required CPU time on the update frequency when a single cut-off is used (open squares in Fig. 2). The effect is less pronounced when long-range forces need to be evaluated (open circles in Fig. 2).

4. Discussion

Benchmarks for the GROMOS96 molecular dynamics program have been presented for various molecular systems on a variety of computers, which gives a reference for comparison with other MD programs and planning of future simulations. The dependence of the computing time on the treatment of the non-

Table 4
Computer CPU time [s] required for MD simulations using an extended cut-off for the non-bonded pair list generation

Computer	Benchmark system				
	II	III	IV	V	VI
DEC alpha 600, 300 MHz	29.2	26.4	47.3	61.5	50.8
DEC 21164, 600 MHz, Linux	16.2	13.3	25.4	33.6	28.5
DEC 21264, AS8400, 575 MHz	8.6	7.7	16.8	17.5	18.9
IBM 6000/397, 160 MHz	39.6	28.5	62.7	76.4	63.6
Pentium Pro, 200 MHz, Linux	54.8	39.8	79.2	111.0	93.3
Pentium II, 450 MHz, Solaris 2.6	18.6	14.6	33.7	37.2	39.5
HP 9000-735, 99 MHz	69.7	59.8	110.3	125.4	107.5
SGI O2, R5000, 180 MHz	49.3	46.1	73.2	99.7	85.4
SGI O2, R10000, 175 MHz	28.2	24.6	42.1	55.8	49.5
SGI PowerChallenge R10000, 194 MHz	25.8	24.4	41.1	52.2	43.4
SGI Impact, R10000, 195 MHz	25.8	21.3	35.2	48.3	40.6
SGI Origin200, 180 MHz	25.1	22.7	37.1	52.3	43.0
SGI Octane, 195 MHz	23.2	21.0	34.4	48.3	39.7
Sun Ultra-1, 200 MHz	52.6	41.4	69.4	94.1	78.3
Sun Ultra-10, 333 MHz	29.0	26.6	53.0	54.6	59.0
Sun Ultra-30, 300 MHz	32.0	28.6	47.3	60.5	48.7

The benchmarks were run with the same parameters as for the standard version. The extended pair list cut-off exceeded by 0.4 nm the standard cut-off values given in Table 1. The extended pair list was automatically updated when an atom had moved by more than 0.4 nm resulting in typical update frequencies around once per 100 MD steps. Benchmarks IV and VI were run for 100 MD steps and the resulting CPU times were divided by ten to allow comparison with the numbers in Table 3.

bonded interaction, in particular of the non-bonded cut-off and on the update frequency of the non-bonded pair list has been discussed and a new implementation for the generation of the non-bonded pair list based on an extended pair list has been described. The idea of an extended pair list originates from the work of Fincham and Ralston [10], who defined a skin around the spheres corresponding to the non-bonded cut-off standardly used in MD simulations. It has been adapted here to allow its use with triple-range cut-offs for updating the standard non-bonded pair-list and long-range forces. The approach allows for a significant speed-up of MD simulations for large biomolecular systems. We should add here that the use of triple-range cut-off radii does not introduce any new artefacts compared to the standard use of cut-off radii for the calculation of non-bonded interactions, since it only affects the way the non-bonded pair list is generated, but neither its content nor the force calculation itself. Further, the speed-up achievable with an extended pair list is not very sensitive to the composition of the system (e.g., pure solvent, protein, ionic solutions) but rather to the choice of non-bonded interaction parameters such as the value of the twin-range cut-off radii and the non-bonded pair list update frequency.

Generally, MD simulation programs for molecular systems can be made faster using a variety of techniques and tricks, which can be classified as follows:

(1) vectorisation for vector processors such as Cray, Fujitsu and NEC computers,
(2) parallelisation for shared-memory systems or distributed-memory parallel computers such as the SGI Origin2000 and IBM SP,

(3) simplification of the algorithm and code by a simplification of the physical model or a reduction of the functionality, e.g., no Coulomb forces, no periodic boundary conditions, cut-off radius,

(4) use of more efficient algorithms to carry out specific tasks, such as selecting non-bonded neighbours, imposing constraints, calculating square roots.

Here, we have only considered a technique of the last category, the use of an extended pair list to select non-bonded neighbours. It enhances the computational efficiency by 50 to 300%: for a medium-sized protein ($\approx$300 residues) in water 50 to 100 psec of simulation per day can be produced on a fast workstation or personal computer. Other improvements of the type of category 4 are possible. To expect at least another order of magnitude increase in speed, leading to a production of 1 nsec of simulation per day for a 300-residue protein in aqueous solution on current hardware is reasonable. Using the shared memory parallel version of GROMOS96 on a limited (up till 8) number of processors, a speed-up of up to a factor 6 can be obtained. This means that by combining a variety of techniques MD simulations of a microsecond for a medium-sized protein in water will in the near future be in reach using readily available multiprocessor machines.

Acknowledgments

The authors thank Roland Bürgi, Wolfgang Damm and Xavier Daura for help in carrying out the benchmarks.

References

[1] W.F. van Gunsteren, S.R. Billeter, A.A. Eising, P.H. Hünenberger, P. Krüger, A.E. Mark, W.R.P. Scott, J.G. Tironi, Biomolecular Simulation, The GROMOS96 Manual and User Guide (Vdf Hochschulverlag AG an der ETH, Zürich, 1996) pp. 1–1042.

[2] W.R.P. Scott, P.H. Hünenberger, I.G. Tironi, A.E. Mark, S.R. Billeter, J. Fennen, A.E. Torda, T. Huber, P. Krüger, W.F. van Gunsteren, J. Phys. Chem. A 103 (1999) 3596.

[3] X. Daura, A.E. Mark, W.F. van Gunsteren, J. Comput. Chem. 19 (1998) 535.

[4] I.G. Tironi, R. Sperb, P.E. Smith, W.F. van Gunsteren, J. Chem. Phys. 102 (1995) 5451.

[5] B. Quentrec, C. Brot, J. Comput. Phys. 13 (1973) 430.

[6] R.W. Hockney, S.P. Goel, J.W. Eastwood, J. Comput. Phys. 14 (1974) 148.

[7] R.W. Hockney, J.W. Eastwood, Computer Simulation Using Particles (McGraw-Hill, New York, 1981).

[8] W.F. van Gunsteren, H.J.C. Berendsen, F. Colonna, D. Perahia, J. Hollenberg, D. Lellouch, J. Comput. Chem. 5 (1984) 272.

[9] M.P. Allen, D.J. Tildesley, Computer Simulation of Liquids (Clarendon, Oxford, 1987).

[10] D. Fincham, B.J. Ralston, Comput. Phys. Commun. 23 (1981) 134.

[11] S.M. Thompson, CCP5 Quarterly 8 (1983) 20.