Molecular Dynamics Simulations using GRAPE/Wine Hardware

Thomas Huber and Alistair Rendell[1]
Supercomputer Facility
Australian National University
Canberra, ACT 0200
Australia

June 10, 1999

Aim

The aim of this work is to produce a basic molecular dynamics (MD) simulation package, which through the use of the purpose specific GRAPE/Wine hardware [1,2] for calculating long range pairwise interactions, is both fast and accurate. Target applications for the new code are large systems and long time frame simulations.

Our aim is *not* to produce "yet another MD package", but to interface well established MD packages to the GRAPE/Wine hardware. The advantage of this is that it allows the user to exploit all the pre– and post–processing functionality that is available in the established package, and also enables easy comparison between the new and original code/environment.

Description of Code

Our work is initially based upon use of the GROMOS package [3]. There are two main reasons for this:

- The functional forms for non-bonded interactions are simple, eg unlike many other MD codes it does not have special terms for hydrogen interactions. This greatly simplifies the use of GRAPE/Wine and also makes the code more computationally efficient.
- The atom charges used in the GROMOS force field are generally smaller compared with other force fields. It is hoped that this will lead to smaller errors in the special way cutoffs are implemented when using GRAPE.

In our current work we have developed a replacement code for the GROMOS executable *promd*. We term our new code CMOS, for Canberra MOlecular Simulation code. In comparison to *promd*, CMOS has much reduced functionality, but still contains all the essential elements required to enable the user to perform a variety of "state of the art" MD simulations. In summary CMOS:

[1]Email: Thomas.Huber@anu.edu.au or Alistair.Rendell@anu.edu.au

- Performs molecular dynamics simulations by leap frog integration [4] of Newton's equation of motions.
- Treats systems that are rectangular periodic or non-periodic.
- Includes two independent Berendsen thermostats [5] for solute and solvent that can be used to control the temperature of the system.
- Can monitor simulation variables (such as coordinates, energies and temperature) and store these in GROMOS format for further analysis.
- Enables bond lengths to be constrained during a simulation [6,7] and overall system motion to be removed if required.

Beyond these fairly standard MD features, CMOS includes a number of unique features:

- Non-bonded (van der Waals and Coulomb) interactions are calculated using the purpose built GRAPE/Wine hardware with both periodic and non-periodic boundary conditions.
- To avoid $O(N^2)$ scaling a cell method is implemented. This divides the total system (whose size is many times bigger than the chosen cutoff) into a number of independent sub-cells, reducing the computational cost to $O(N)$.
- Long range electrostatic interactions can be evaluated using either cutoffs or an Ewald summation.
- The code has been designed from the outset to vectorise and parallelise well. This is achieved by special pre-processing of the simulation data and a careful selection of MD algorithms, eg to constrain bond lengths the LINCS [6] and SETTLE [7] algorithms are used instead of SHAKE [8].

Benchmark Systems

CMOS has been extensively tested on two systems:

- The first (simpler) system consist of pure Lennard Jones (LJ) fluid at a density of $\rho^* = 0.95$ and a temperature of $T^* = 1.05$ (values are given in reduced units). An integration timestep of $t^* = 0.005$ is used, and for canonical simulations the temperature bath coupling constant is set to 0.1. Five different system sizes are used containing 125, 1000, 3375, 8000 and 27000 atoms.
- The second (more realistic) system is a protein obtained from Dr Suenaga at RIKEN. The protein contains 28 residues (323 atoms) and is simulated both in vacuo with no boundary conditions, and solvated in a cubic water box of 4059 water molecules with periodic boundary conditions. The system is at 300K, a timestep of 2fs is used, and all bond lengths are contrained to their equilibrium values.

Program Testing

As will be discussed in more detail later, the way in which cutoffs are implemented for GRAPE/Wine hardware makes direct comparison of GROMOS and CMOS difficult. Furthermore this is especially true when we consider long range electrostatic interactions, since the periodic system/Ewald summation approach used in CMOS is fundamentally

different to the reaction field treatment adopted by GROMOS. Despite these difficulties it is possible to make a number of tests to ensure that the new code is working correctly.

For non-periodic systems simulated using a cutoff that is larger than the system size the results obtain from GROMOS and CMOS should essentially be the same. This appears to be the case for both the LJ fluid and the small protein, eg after 2500 simulation steps on the protein the root mean square deviation between the velocities and coordinates obtained from CMOS and GROMOS is 1.6E-5 and 1.2E-3 respectively; this is roughly the same as the difference seen when running GROMOS on different computer platforms.

More generally for a system that is not coupled to a temperature bath (micro–canonical simulation) the total energy of the system should be conserved. In figure 1 we plot the kinetic, internal and total energy of the small 28 residue protein for 1000 simulation timesteps; while the kinetic and internal energy fluctuates as expected, the total energy remains almost constant. The total energy fluctations for all the test systems are quantified in table 1, where we give the root mean square deviations of the total energy E from the initial value normalised by the number of atoms in the system. This is defined as:

$$\sigma_E = \frac{1}{N_a}\sqrt{\frac{1}{n_t}\sum_{t}^{n_t}(E(t)-E_{init})^2}$$

where N_a is the number of atoms in the system, n_t the number of integration steps analysed, $E(t)$ the total energy at timestep t and E_{init} the initial total energy of the system. We note that this definition is slightly different to that used in assessing the performance of the PEACH software package [9, 10], where the result was normalised by the average total energy. The latter is in our opinion misleading in that the total energy is an arbitrary offset that could be chosen to be anything in order to obtain any desired precision.

As stated above simulations using the Ewald summation method can not be directly compared with GROMOS simulations, since GROMOS uses the reaction field method to treat long range interations. Despite this it appears that the two methods produce remarkably similar results. Starting from identical initial conditions in figure 2a-b we plot the energies obtained at the first 1000 simulation steps when using CMOS and GROMOS to simulate the small protein system in water. While the absolute energies differ (since we neglect a constant energy term in the Ewald summation) the overall trends are very similar, especially in the first 100 timesteps.

In summary it appears that the program is working for the systems considered here, but of course there is a need for continued quality assurance as the code is applied to a greater variety of problems.

Timing Analysis

A detailed timing analysis for CMOS running on the DEC Alpha 464 workstation and GRAPE/Wine hardware located at RIKEN (machine named LEDA) has been made. In our time analysis energy of the systems are evaluated at each time step, though we note that in an application simulation energy is generally only neccessary to monitor a simulation, and thus, needs only be calculated once every 50-100 simulation steps.

In table 2 we present timings for the small protein in a 5nm cubic box with 4059 water molecules (total system size 12500 atom). This calculation does not use an Ewald summation. The results show that approximately 95% of the total time is spent in the non-bonded part of the calculation, or that part of the calculation that is performed on the GRAPE hardware. The latter is then broken down into the time required to compute the van der Waals (vdW) and electrostatic, energies and forces. Not surprisingly these four terms take roughly the same time. For the vdW terms the calculation time is further decomposed into the different stages required when using GRAPE:

index sort	- time to sort atom types
init board	- initialisation of board, done each timestep at the moment
set coord+parm	- sending coordinates and parameters to board
calculate	- time to calculate E+F on GRAPE board
free board	- freeing the board after calculation (again each timestep)

This shows that while roughly 86% of the vdW time is spent performing calulations on GRAPE, about 7% of this communicating coordinate and paramater data to/from GRAPE.

Similar results for the LJ fluid with 125, 1000, 3375, 8000 and 27000 atoms are given in table 3. As expected the total time increases roughly quadratically with system size. Concentrating on the breakdown of the vdW time, we see the *calculate* time increasing quadratically, the *index sort* and *set coord/parm* times increasing linearly, while the *init board* time remains roughly constant. All this is as expected. What is slightly strange is the fact that the *free board* time increases, but since this is a relatively small fraction of the total time we have not investigated this further. As a result of this scaling pattern we find that while only 25% of the total vdW time is *calculate* time for the 125 atom system, this rises to 91% for the 27000 atom system. Thus GRAPE appears to be best suited to reasonable sized system, where the communication and setup overheads are less dominant.

The long term aim of our work is to perform MD simulations using multiple GRAPE/Wine boards. This is achieved by using a cell method in which the total system is divided into a series of spatially distinct rectangular cells. Each cell has dimensions that are greater than the cutoff radius, and as a consequence each atom only interacts with other atoms in the same cell or those in the surrounding 26 nearest-neighbour cells. Separate calls are made to the GRAPE hardware for each cell, but each call requires the coordinates of all

atoms in that cell and the surrounding 26 cells to be communicated to the GRAPE board. An advantage of this scheme is, however, that each cell can be processed on a different GRAPE board totally independently.

To assess the potential of multiple GRAPE boards as outlined above, we have divided the 27000 atom LJ simulation up into 64 and 125 separate cells. The timing details are shown in Table 4. These show that use of the cell method when we only have a single GRAPE board is not advantageous for this system, rather the total time is increase by a factor of 2-3. For this relatively small system this is probably not surprising. However, using the cell method we now have either 64 or 125 independent parallel tasks, so theoretically it should be possible to run these all in parallel using multiple GRAPE boards. This would provide substantial gains over the single GRAPE board implementation. Examination of the detailed vdW times in table 4 shows that while the *calculate* time is decreasing when using the cell method, the *set coord/parm* is increasing dramatically, and to a lesser extent the board initiating time is increasing. We note that presumably when running in a dedicated environment there would be no need to continually initiate or free the GRAPE board and this would reduce some of the overhead.

All the above results have only considered cutoff based simulations using just the GRAPE board. In table 5 we present results obtained for the solvated small protein using periodic boundary conditions and the Ewald summation method. The total system contains 12500 atoms, and calculations have been performed using 1, 27, 64 and 125 cells. In this case the non-bonded interaction is broken down into real and reciprocal space parts. The real space part performs evaluations of pairwise interactions using a cutoff, in an identical manner to that discussed above. The reciprocal space calculation is performed using Wine, but with a small component of the calculation done on the host computer.

The parallelisation strategy used for the reciprocal space part of the computation is somewhat different to that used for the real space part. For the latter each cell corresponds to a different task and a separate call to the GRAPE hardware (which may or may not be done in parallel). For the recripocal space part parallelisation is over k vectors, and in this case there are 20 k vectors in each cartesian direction. Thus we have a maximum degree of parallelisation of 20^3. In table 5 the time given under "Wine coulomb recip. forces" is that required to call Wine as many times as there are cells (ie 1, 27, 64, 128 times) treating 20^3/(number-of-cells) k vectors in each call. Not surprisingly we see the total time increase, but as with the real space part of the calculation we now have many parallel tasks which can be performed in parallel.

Charged Groups in Cutoff Based Simulations

While long range electrostatic interactions can be evaluated using GRAPE and a cutoff radius, this cutoff is based on a strict r distance. This is in contrast to what is generally done in biomolecular simulations, where the r distance is somewhat "flexible". Specifically

biomolecules are usually composed of a series of residues, and while each atom may be partially charged overall each residue is usually neutral. This is also the case for water, eg in the SPC water model a partial charge of -0.82 is placed on the oxygen atom which is neutralised by a charge of +0.41 on each hydrogen. The cutoff actually used is based on residues, such that if any atom of the residue falls within the cutoff distance all atoms in that residue are assumed to lie within the cutoff. Effectively this means that interactions between charge neutral dipoles are evaluated rather than interactions between charges; as the former converges as r^{-3} while the latter converges as r^{-1} this offers a considerable advantage.

To illustrate this effect more clearly, we consider two (rigid) water molecules in a plane facing such a way that the H1-H1 and H2-H2 separations are equal and bigger than the O-O separation. A schematic picture of this is given in figure 3. The electrostatic energy of this system as a function of the O-O distance is shown in figure 4. Also shown in figure 4 are the electrostatic energies calculated using a cutoff distance of 1.0 nm for both an atom based cutoff and a charged group based cutoff (interactions between charge groups are calculated when the oxygen-oxygen distance is lower than the cutoff). The atom based cutoff scheme experiences huge (artificial) energy jumps as the distances between different atom pairs becomes bigger than the cutoff radius. This first happens when the H1-H2 distance is larger than the cutoff giving rise to an attractive force. This force further increases when the H1-H1 distance falls beyond the cutoff. The next interaction to be removed is the attractive H1/H2-O force, leaving only the replusive O-O interaction, which eventually disappears to zero as all atoms move beyond the cutoff. In contrast when the molecule based cutoff is used the interaction potential reproduces the exact result.

The above water/water example is not a pathological example; simulations with atom based cutoffs can give huge energy fluctuations or incontinuities in the energy surface. As a consequence control of the system's temperature is difficult, if not impossible. This problem is, of course, avoided if a full Ewald summation is used. Thus at the moment it appears that when using GRAPE/Wine for biomolecular simulation use of Ewald is compulsory - but this may not be a bad thing!

Summary

What has been done

- A complete GROMOS *promd* clone was written that includes the following features:
  - GROMOS energy and force calculation with periodic and non-periodic boundaries, using GRAPE/Wine hardware for non–bonded part.
  - Leap frog integrator.
  - Berendsen thermostats for solute and solvent.
  - Bond constraints using LINCS for solute, SETTLE for water (solvent).
  - Removal of overall translational and rotational system motion.
  - Highly vectorised with special pre-processing of simulation parameter.

  - Force calculations, integration and constraint satisfaction fully parallelised.
  - Cell method to calculate non-bonded interactions with GRAPE.
  - Long range electrostatics via Ewald method using WINE hardware.
  - Exclusion of hydrogens from van der Waals energy calculation.
  - Exclusion of uncharged atoms from Coulomb interaction calculation.
- The new code has been thoroughly tested on a variety of systems.
- A timing analysis has been made for the new code and various conclusions drawn concerning the code's likely parallel performance.

Things to do

The code is ready to run as is on MD-GRAPE2; as soon as the new chips become available we will be able to run CMOS and evaluate performance. It will be especially interesting to have multiple GRAPE and Wine chips, then we can perform real tests on the parallel performance of the code. This would be possible even with existing GRAPE/Wine boards.

Beyond the above there are a few items that are not dependent on MD-GRAPE2, and which we are looking into now:

- A SETTLE algorithm for methanol solvents (this is highly desirable as it decreases the number of atoms per unit volume).
- Multiple solute molecules (problems currently arise in the computation of the bonded interactions if one of the solute molecule crosses the box boundary)
- Output routines (ensuring that the coordinates, velocities etc are written in standard format).

References

[1] Ebisuzaki, T., Makino, J., Fukushige, T., Taiji, M., Sugimoto, D., Ito, T., and Okumura, S., GRAPE project: An overview, *Publ. Astron. Soc. Japan* **45**, 269–278 (1993).

[2] Narumi, T., Susukita, R., Ebisuzaki, T., McNiven, G., and Elmegreen, B., Molecular Dynamics Machine: Special-purpose computer for molecular dynamics simulations, *Mol. Sim.* **21**, 401–415 (1997).

[3] van Gunsteren, W. and Berendsen, H., *Groningen Molecular Simulation (GROMOS) Library Manual*, Biomos, Groningen, 1987.

[4] van Gunsteren, W. and Berendsen, H., A leap-frog algorithm for stochastic dynamics, *Mol. Simulation* **1**, 173–185 (1988).

[5] Berendsen, H., Postma, J., van Gunsteren, W., DiNola, A., and Haak, J., Molecular dynamics with coupling to an external bath, *J. Chem. Phys.* **81**, 3684–3690 (1984).

[6] Hess, B., Bekker, H., Berendsen, H., and Fraaije, J., LINCS: A linear constraint solver for molecular simulations, *J. Comput. Chem.* **18**, 1463–1472 (1997).

[7] Miyamoto, S. and Kollman, P., SETTLE: An analytic version of SHAKE and RATTLE algorithm for ridgid water models, *J. Comp. Chem.* **13**, 952–962 (1992).

[8] Ryckaert, J.-P., Ciccotti, G., and Berendsen, H., Numerical Integration of the cartesian equation of motion of a system with constraints: molecular dynamics of n-alkanes, *J. Comput. Phys.* **23**, 327–341 (1977).

[9] Komeiji, Y., Yokoyama, H., Uebayasi, M., Taiji, M., Fukushige, T., Sugimoto, D., Takata, R., Shimizu, S., and Sugimoto, D., A high performance system for molecular dynamics simulation of biomolecules using a special-purpose computer, in *Biocomputing*, edited by Hunter, L. and Klein, T., volume 1, pp. 472–496, World Scientific, Singapore, New Jersy, London, Hong Kong, 1996.

[10] Komeiji, Y., Uebayasi, M., Takata, R., Shimizu, A., Itsukashi, K., and Taiji, M., Fast and accurate molecular dynamics simulation of a protein using a special-purpose computer, *J. Comp. Chem.* **18**, 1546–1563 (1997).

Table 1: normalised standard deviation of total energy for 100 steps micro–canonical simulation

	number of atoms	σ_E
Lennard–Jones fluid	125	8.55694e-05
	1000	1.72172e-04
	3375	1.51958e-04
	8000	1.47527e-04
	27000	1.40664e-04
protein system	323	3.24444e-03
protein system (with constraints)	323	5.23974e-03

Table 2: Wall clock time (ms) for 1 simulation step of a 12500 atom solvated protein benchmark (-*- indicates times smaller than timer resolution)

total	3918.0
non-bonded	3733.184
bonds	33.332
angles	-*-
improper dihedral angles	-*-
dihedral angles	-*-
non-bonded 1-3	36.665
non-bonded 1-4	3.333
integration	89.996
non-bonded total	3733.184
GRAPE vdW energy tot	964.961
GRAPE vdW forces tot	963.295
GRAPE coulomb energy tot	896.631
GRAPE coulomb forces tot	901.631
vdW total	1928.256
GRAPE vdW E+F: index sort	99.996
GRAPE vdW E+F: init board	18.333
GRAPE vdW E+F: set coord & param	141.661
GRAPE vdW E+F: calculate	1651.601
GRAPE vdW E+F: free board	11.666

Table 3: Wall clock time (ms) for 1 simulation step of Lennard–Jones fluid benchmarks. (-*- indicates times smaller than timer resolution)

	number of LJ particles				
	125	1000	3375	8000	27000
total	68.0	173.3	580.1	1864.8	13883.0
non-bonded	65.831	167.3	562.977	1823.927	13747.783
integration	-*-	2.5	10.500	26.499	94.996
non-bonded total	65.831	167.3	562.977	1823.927	13747.783
GRAPE vdW energy tot	18.666	43.665	145.994	466.481	3478.194
GRAPE vdW forces tot	17.999	45.832	145.828	468.481	3476.528
GRAPE coulomb energy tot	14.499	38.998	133.828	440.149	3381.531
GRAPE coulomb forces tot	14.666	38.165	134.828	443.982	3391.531
vdW total	33.665	89.497	291.822	934.962	6955.311
GRAPE vdW E+F: index sort	0.167	6.500	23.166	58.331	229.991
GRAPE vdW E+F: init board	17.833	18.333	20.499	18.833	19.999
GRAPE vdW E+F: set coord & param	9.500	17.666	40.165	90.330	283.322
GRAPE vdW E+F: calculate	8.666	44.832	203.325	754.303	6383.078
GRAPE vdW E+F: free board	0.333	1.167	3.167	8.666	29.999

Table 4: Wall clock time (ms) for 1 simulation step of Lennard–Jones fluid benchmark with 27000 atoms using cell method to sub–divide system.

number cells	1	4x4x4	5x5x5
number of GRAPE calls	1	64	125
total	13883.0	29720	30550
non-bonded	13747.783	29565.484	30415.450
integration	94.996	99.996	99.996
non-bonded total	13747.783		
GRAPE vdW energy tot	3478.194		
GRAPE vdW forces tot	3476.528		
GRAPE coulomb energy tot	3381.531		
GRAPE coulomb forces tot	3391.531		
vdW total	6955.311		
GRAPE vdW E+F: index sort	229.991	849.966	783.302
GRAPE vdW E+F: init board	19.999	1183.286	2299.908
GRAPE vdW E+F: set coord & param	283.322	8199.672	8682.986
GRAPE vdW E+F: calculate	6383.078	4299.828	3266.536
GRAPE vdW E+F: free board	29.999	33.332	49.998

Table 5: Wall clock time (ms) for 1 simulation step of PSV protein benchmark (12500 atoms) using Ewald summation and multiple-cell method to sub–divide system. energies with 5x5x5 subdivision deviates due to a too small cutoff for chosen $\alpha = 1.0$

number cells	1	3x3x3	4x4x4	5x5x5
number of GRAPE calls	1	27	64	125
total	4993	16236	20146	27691
non-bonded	4946	16194	20103	27646
integration	47	42	43	45
non-bonded total	4946	16194	20103	27646
GRAPE vdW energy tot	258	1088	1430	2078
GRAPE vdW forces tot	257	1043	1378	2082
GRAPE coulomb direct energy	1430	3113	2777	3208
GRAPE coulomb direct forces	1435	3070	2752	3107
total coulomb recip. forces	1567	7781	11678	17053
Wine coulomb recip. forces	1560	7763	11648	17024
HOST coulomb recip. forces	2	8	15	17
vdW total	515	2131	2808	4160
GRAPE vdW E+F: index sort	8	43	30	32
GRAPE vdW E+F: init board	20	462	1090	2062
GRAPE vdW E+F: set coord&param	33	958	1157	1500
GRAPE vdW E+F: calculate	448	637	462	485
GRAPE vdW E+F: free board	0	3	18	33

Figure 1: Energy trajectories of protein system in vacuo during a microcanonical simulation

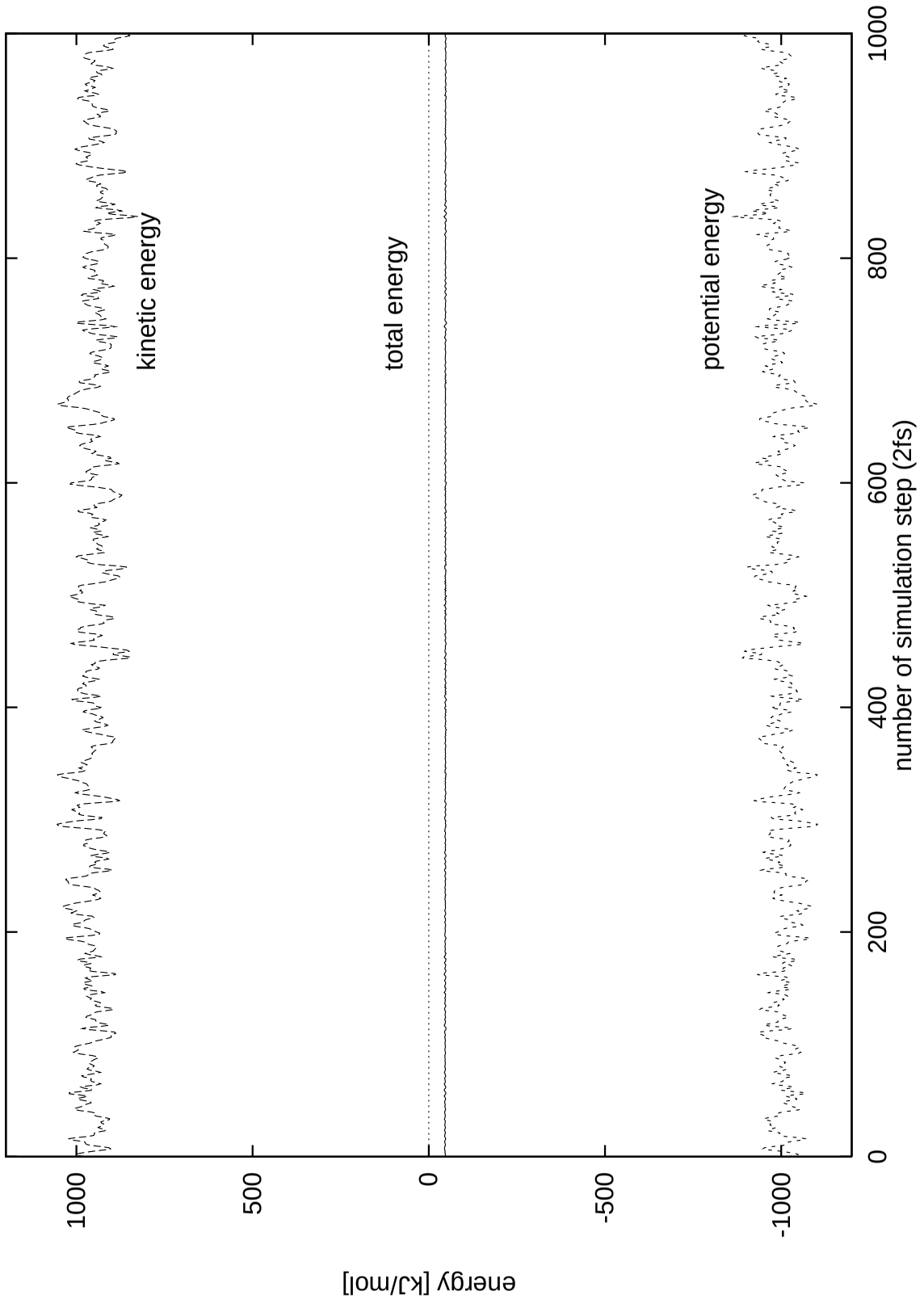


Figure 2a: PSV protein in water box: Comparison GROMOS96 vs Cmos

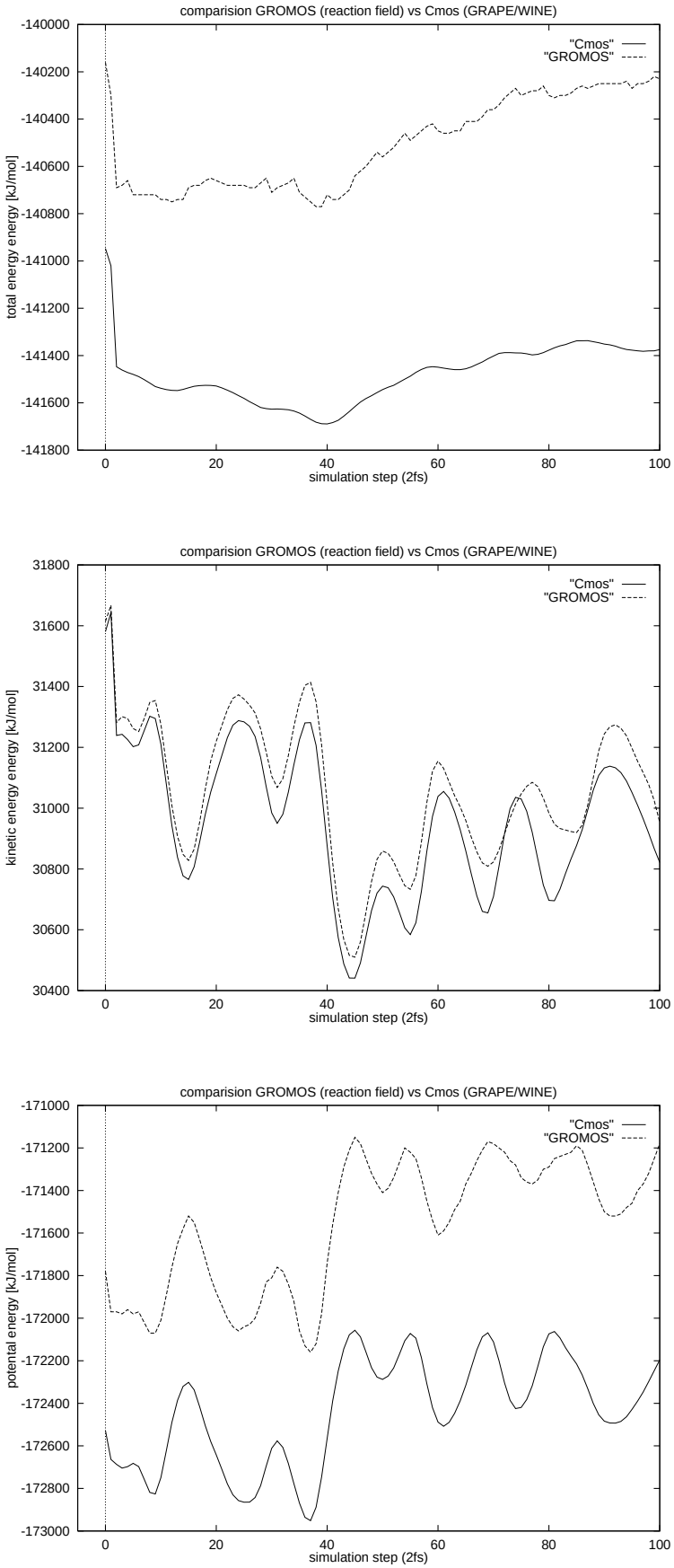
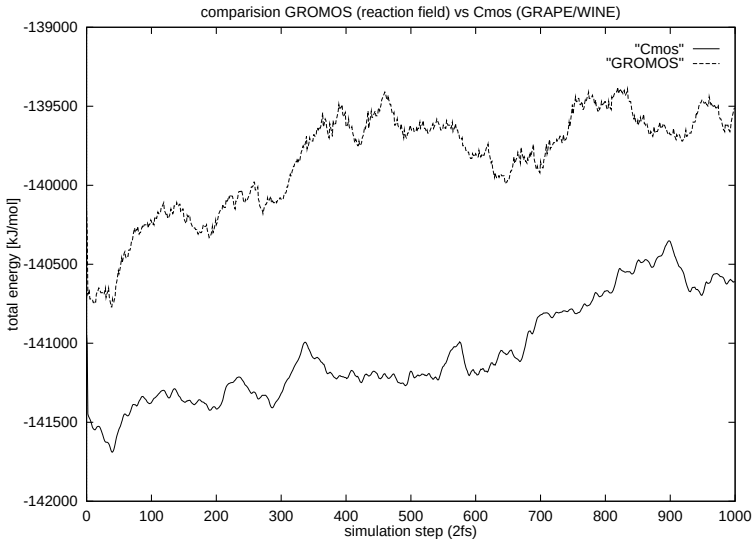
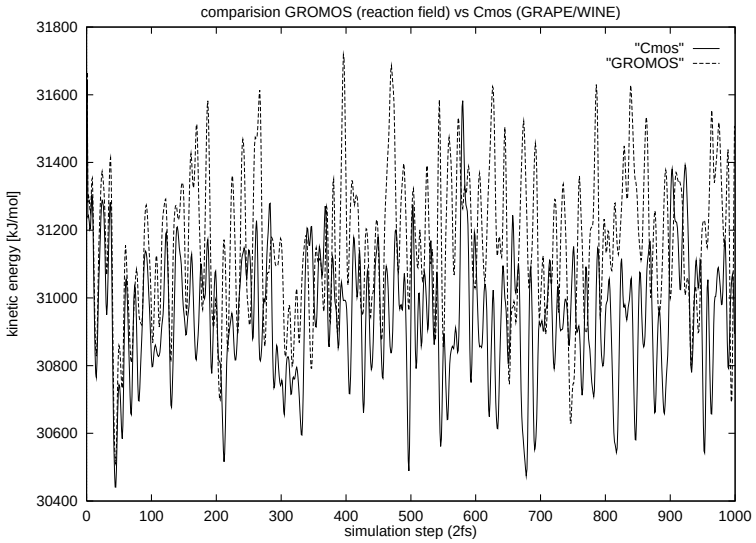


Figure 2b: as figure 2a only a simulation 10 times longer





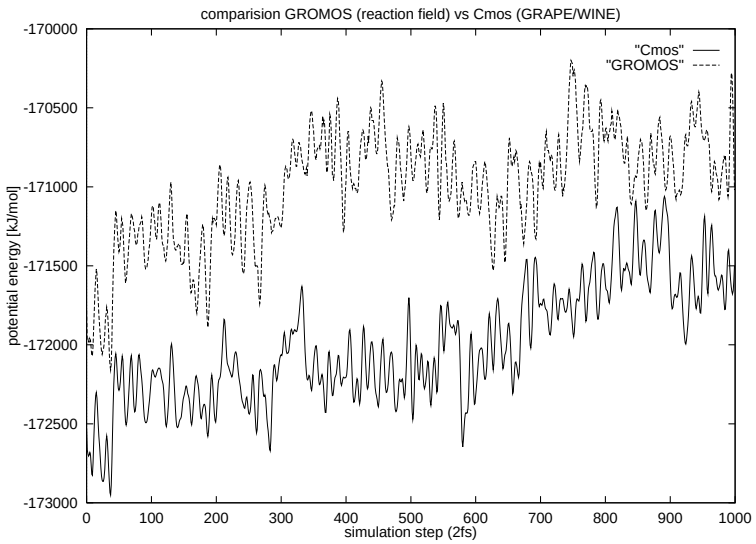


Figure 3: SPC water model to illustrate problem with atom based cutoffs

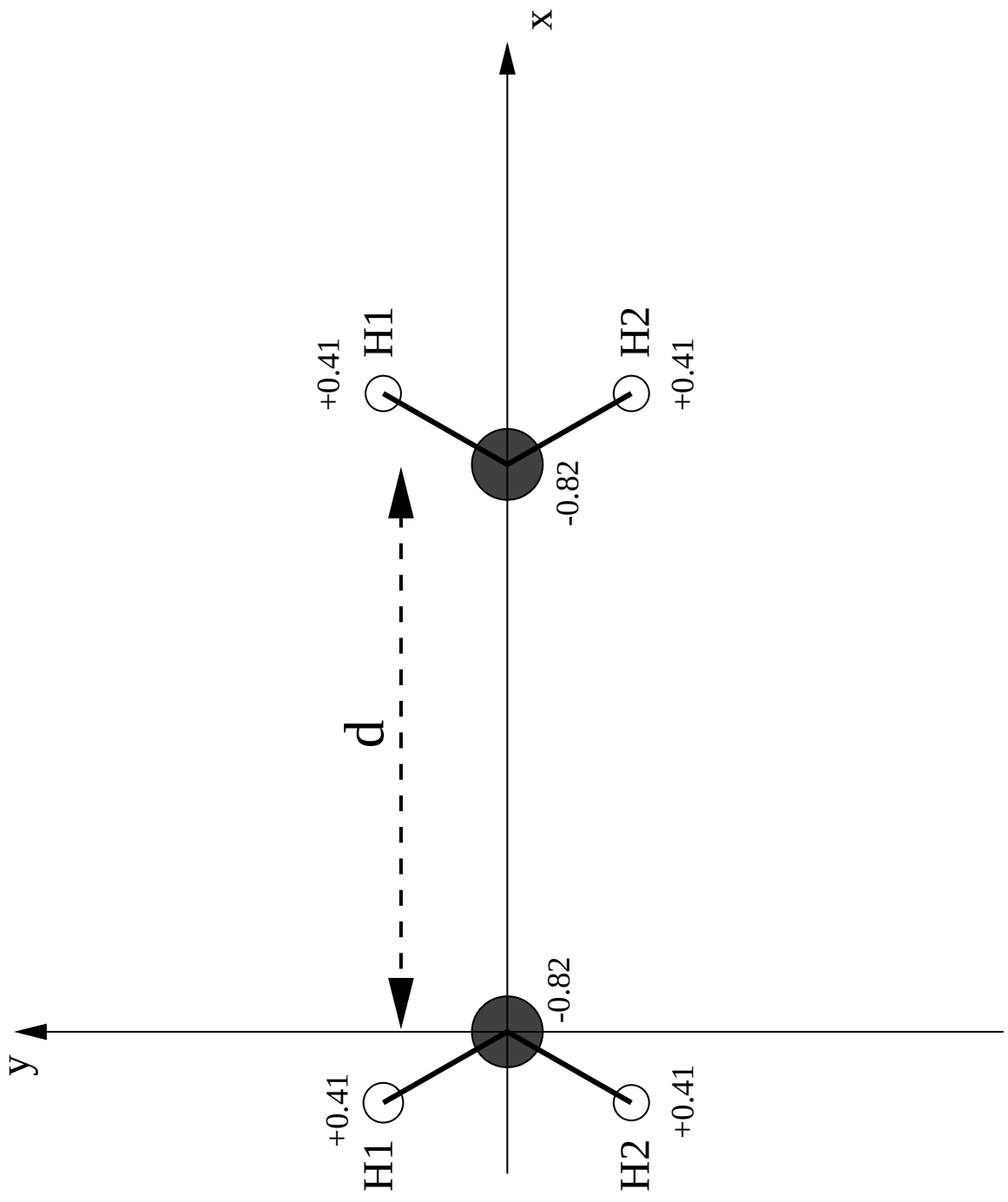
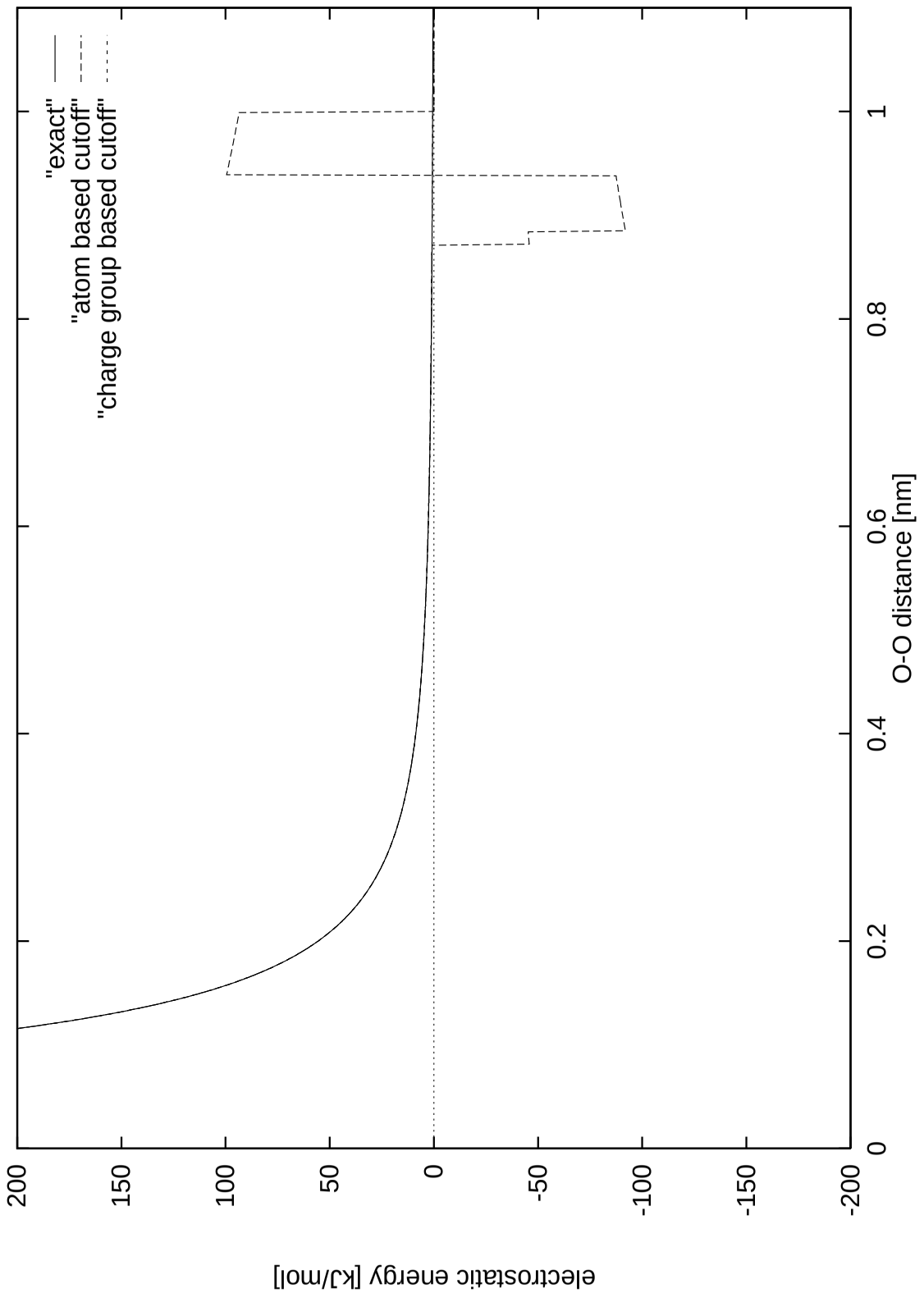



Figure 4: Electrostatic energy of SPC water model