Kraków 8.11.2001

Dr Witold Dzwinel
AGH Institute of Computer Science,
Al.Mickiewicza 30,
30-059 Krakow, Poland
tel +48 12 6173520
fax +48 12 6179406

The Editor of Concurrency Practice and Experience
Professor Dr G.Fox

Dear Professor Fox,

Thank you very much for good news about acceptance of our paper (C519) for publication in your journal. Below please find the rebuttal to the referee remarks. We thank (with prof D.A.Yuen) the reviewers for their work and valuable remarks which help us to improve our manuscript.

Best regards

Professor Dr Witold Dzwinel

Rebuttal

Referee 1 (answers to the comments numbered 1-3)

Comm. No.	Answers
1	The description of parallel implementation (page 7) was improved according to the referee criticism.
2	Unfortunately, we did not get the copy with editorial corrections made by the referee. We made our best.
3	The order of figures is correct now. The sentence corresponding to the Fig.4b is reordered. The rhombic dodecahedron can be easily constructed by using the checker board 3-D mesh presented in Fig.4b (see the reference [34])
Comments in the text of review	1) The text on page 7 is used as a comment to the scheme of parallel realization shown in Fig.2. O.K. some redundancy and confusing statements were removed. 2) The acronyme for mutual nearest neighborhood was corrected according to the referee comment. 3) As the referee suggests, the definition of ccNUMA was moved before this acronyme was used.

Referee 2 (answers to detailed comments numbered from 1 to 28)

Comm. No.	**Answers**
1 - 2	O.K. Corrected.
3 – 4	This is an error coming from MS Word .doc translation to .pdf format. The rotational velocities were replaced with question marks. Sorry for this fault.
5	The temperature definition was moved to the more suitable place (after Eq.(8)).
6	The forces Fr and Ft (Eqs.4,5) have non-central component coming from the non-zero A(r) component of tensor T from Eq.(7).
7	The respective sentence was replaced with “The kinetic theory for FPM has been developed for deriving transport coefficients by assuming that conservative forces are absent.”. The iterative procedure was used for simulations where conservative forces were large comparing to the dissipative and Brownian components and in the cases where it was necessary. I mean, in the cases where quantitative results were needed.
8	The assumption that Atylda(r)=0 comes from some freedom in selecting the functions A,B,C, (tylda) (see [17]). The problem in selecting weight functions is less trivial. I identify the weight functions with some sort of “form factor” of the fluid particle, which describes mesoscopic properties of the cluster, droplet, of fluid molecules and (like interactions between atoms in a simple fluid) defines macroscopic parameters of the fluid. In fact, what we need from weight functions (both in DPD and FPM) is that they should be normalized and >0 in [0, rcut] interval. In “continuum limit” the qualitative hydrodynamic behavior of particle fluid does not depend on shape of the weight functions, however, the partial pressure and the transport coefficients (Eqs.13-15) depend on their first two momenta (integrals). In this stage of study, the linear character of weight functions (same as in DPD) let us discern better the differences between FPM and DPD coming from non-central character of FPM forces and rotational components introduced in FPM. However, analogously as it is in smoothed particle hydrodynamics (SPH) macroscopic model, we think that the weight functions should be rather bell-shaped than linear. In our opinion this problem of selecting weight functions must be studied soon to make particle fluid not only a qualitative model but the valuable quantitative simulation tool.
9	I cannot see any “dot”. ???
10	I cannot understand the question about the computational cost in the “continnum limit”. The computational cost increases proportionally with the number of particles simulated and the volume of the system. In the case of rescaling, i.e., increasing the size of “liquid droplet” with increasing spatio-temporal scale, the problem with interpretation and matching remains still opened. In our opinion, on the path from nano-mesoscopic scale to macroscale the interpretation of fluid particle should transform from “lump of fluid” in mesoscale to “approximation point” (as it is in SPH) in macroscale. In macroscopic systems, the identification and definition of fluid particle seems to be very artificial.

11	O.K. Corrected.
12	O.K. Corrected
13	It depends on how many processors you plan to used and which geometry of computational box is considered. Each of these two decompositions has as many advantages as disadvantages. The partitioning into a mesh of subboxes results in more complicated boundary conditions between domains and increasing of communication and making it more complex (load balancing is difficult). Decomposition into a mesh of subboxes can be more efficient for rectangular computational boxes, very large number of particles and large number of SLOW processors involved in computations (Beazley and Lomdahl MD simulations on CM-5 machine). In the corrected text we show that for L>>1 (where L is the length of computational box elongated in z direction) dividing the box along z direction gives lower overhead than for sub-boxes partition.
14	The results presented are obtained for the best arrangements of tables. We have added the following paragraph in the manuscript: “Because the particles that are the physical neighbors should also be closer one another in the computer memory, to avoid frequent cache misses the particles are renumbered every some period of time. In result the particles residing in the same cell have consecutive numbers. However, the gap between particle numbers still exists for the particles from different cells. This is due to the sequential numbering of particles in domains. Let us assume that the particles are numbered first along x, then y and finally z directions. By increasing 4 times the sizes of computational box in *x,y* plane, the gap between particle numbers from the neighboring cells in z-direction increases also 4 times. Thus, the respective **r**, **v** and w coordinates of two interacting particles from these cells can be very distant in memory generating cache misses”. Cache sizes are 4MB and 8MB for IBM SP and SGI/Origin, respectively.
15	Number of different arrays (variables) describing particles in MD and FPM are as follows **1.** for MD: 3 tables for positions of particles and 3 tables for forces. Sometimes velocity is stored, but, for example, in Boston MD code for large-scale MD, velocity is computed from particles positions. **6 arrays in total.** 2. for FPM: same as for MD plus 3 tables for angular and 3 for translational velocities, 3 tables for momenta, and 6 additional tables replicated arrays for velocities needed for integrating Newtonian equations of motion (see Eqs.(21-24)) (3 for angular and 3 for translational velocities). **24 arrays in total.** The number of variables describing FPM particle is much greater. Moreover, random number generator is invoked 4 times for Brownian component computation for each pair of interacting particles.
16	He is right. The memory is a problem especially for FPM. See page 8 “On the other hand, predictor-corrector numerical schemes are both very time and memory consuming, which for high memory load for FPM will result in additional overheads.”
17	O.K. Corrected.

18	Stability is crucial not the order of the scheme. We used stable scheme, which is of higher order than Verlet’s.
19	O.K.Corrected
20	O.K. Corrected
21	We replaced “transition rules” with ‘nearest image convention schemes [33]”. We give also the reference to these schemes.
22	No comment.
23	O.K. Corrected
24 25	Should be R14000/500 processor Now is O.K.
26	We have rearranged the sentence from the first paragraph on page 13 which is now “Making the computational box wider in *x*,*y* plane and proportionally shorter in z direction (in **pfpm1** the number of cells remains the same as in **pfpm0**) ….”. Also the whole paragraph has been rewritten according to the referee recommendation. Due to MPI interface used, there is no evidence of two particles from distant memories interacting. All interacting particles are located in local memories.
27	The communication between processors residing on a single node is faster than between processors from two nodes. Same is for, so called, frames on IBM SP. The frames consisting of 2-4 nodes are connected by Vulcan switch same as nodes in frames, however, communication between nodes from different frames and nodes inside a frame are organized in different ways. The switch between frames is more busy than switches inside frames, causing additional overheads for messages coming from different frames. The respective sentence has been rearranged: “This may come from communication delay between processors belonging to different IBM SP *frames*, which involve switches between the frames. The network is shared between other users. The machine remains very busy. Thus communication between processors from different frames (supernodes) may be much slower than in a single node or inside the frame.”
28	On the both machines we used MPI interface. According the criticism of the reviewer and new results we obtained from parallel clustering we rethink our point and completely change this and the following paragraphs.

Referee 3 (answers to detailed comments numbered from 1 to 5)

1	Captions will be put below the figures in editorial process
2	Tables are now reordered in ascending order.
3	I cannot agree that there are too many graphs on a single plot. The graphs are correlated and should be studied together.
4	The ordering of figures is corrected.
5	I cannot understand. All the references are cited in the text.