

## A Velocity Extraction Method in Molecular Dynamic Simulation of Low Speed Nanoscale Flows

Wenfei Zhang

School of Civil Engineering and Mechanics, Yanshan University, Qinhuangdao, Hebei Province, P.R. China, 066004; E-mail: zwenfei@gmail.com.

Received: 31 May 2006; in revised form: 14 September 2006 / Accepted: 19 September 2006 /

Published: 28 September 2006

---

**Abstract:** A new algorithm to extract the velocity caused by the external forces in molecular dynamic simulation of nanoscale flow problems is proposed. The flow velocity, an important component in these type of problems, is usually obtained from the average value in the time space because the accumulation of the thermal velocity will approach zero when the time period is large, but this method is not always suitable, especially when the flow velocity is much smaller than the thermal velocity. Based on the idea of the linear accumulation of the flow velocity, in this study a new algorithm is derived to extract the flow velocity. This algorithm can be used to calculate nanoscale flow problem no matter whether the value of the flow velocity is big or small. Using this new algorithm, the 2-D liquid flow of argon in nanochannels was simulated. The numerical result demonstrates the effectiveness of the new algorithm.

**Keywords:** Velocity extraction; molecular dynamic simulation; low speed; nanoscale flow.

---

### Introduction

Nanoscale flow studies are important in physiology, medicine and for design of nanoscale devices, etc. Molecular dynamic simulation (MDS) is used to calculate nanoscale flow problems because the continuum model is no longer valid when the dimensions of flow systems approach the molecular size. Good reviews regarding the models and solution methods of nanoscale flow problems have been published [1, 2]. In many papers, i.e. by Travis *et al.* [3, 4], Thompson and Robbins [5], Heinbuch and Fischer [6], MDS methods were applied to resolve nanoscale flow problems, particularly for the study

of the boundary slip phenomena. 2-D and 3-D pressure driven channel flow problems were also investigated using the MDS method. In most papers the fluid was liquid argon and the Lennard-Jones potential was used to describe the intermolecular interactions. Usually the Couette or Poiseuille steady flow was considered to illustrate the effectiveness of the MDS method. Travis *et al.* [3, 4] found that if the pore width was less than five molecular diameters, the Navier-Stokes (NS) equations break down. Travis *et al.* also reported [3] that when the channel width equals 5.1 molecular diameters, the solution given by the molecular dynamics approach deviates significantly from the NS equations solution, whereas when the channel width equals 10.2 molecular diameters, the molecular dynamics' solution is similar to that of the Navier-Stokes equations. This means that when the channel width is very small the NS solution is meaningless and we should use molecular dynamics to simulate it. Zhang *et al.* [7] calculated a confined fluid with a nonequilibrium MDS method and discussed the phenomena of constant pore size and constant load. Thompson and Robbins [5] simulated boundary slips with different solid wall density and interaction strength parameters. They found that the boundary slip could be observed only for very weak fluid-wall interactions. Similar problems were also studied by Koplik *et al.* [8] and Koplik and Banavar [9]. Todd *et al.* [10] introduced an effective viscosity method to deal with the inhomogeneous viscosity distribution problem. Instead of solving the difficult inhomogeneous viscosity problem, the effective viscosity method adjusts the real viscosity to match the molecular dynamic simulation result. Moseler and Landman [11] and Eggers [12] have discussed nano jet problems.

Although the MDS method has been used to model some nanoscale flow problems, as mentioned above, there is a crucial problem associated with this approach. In the MDS method, the flow velocity of the molecules caused by the applied external force is mixed with the thermal velocity. If the flow velocity is much smaller than the thermal velocity, which is usually more than  $100 \text{ m}\cdot\text{s}^{-1}$ , it is difficult to extract the flow velocity from the much bigger thermal velocity. As most nanoscale flow problems relevant to real applications deal with low speed flows, the direct application of MDS method is not workable. Until now, in all studies the minimum flow velocity used was several meters per second.

Certainly, MDS is correct for nanoscale flow. The problem is the method used to calculate the flow velocity is not suitable. Thus, how to extract the flow velocity is an important part in the MDS of nanoscale flow. The traditional way to extract the flow velocity is using the idea of time average because the accumulation of the thermal velocity will approach to zero while the time period is large, but this method is not always appropriate, especially when the flow velocity is less than several meters per second. That is, numerically, the accumulation of the thermal velocity will never reach zero and the algorithm based on this idea is not valid for low speed problems, regardless of the elegance of the algorithm.

In this paper we propose a new algorithm to extract the velocity caused by the external forces. The idea of this algorithm is as follows: in each time step, we calculate an increment of the flow velocity, and the total flow velocity is the accumulation of the increments in all time steps. This idea is reasonable because at any given time step, we can get the external force increment from the linear development of the total force function. This algorithm can be used to calculate nanoscale flow problem even the value of the flow velocity is very small. Furthermore, when using MDS method in the surface problems [13-16], fracture mechanics [17-19], heat transfer problems, lubrication problems [20], material science [21] and friction problems [22], we also need to calculate the moving velocity

caused by the external force, or other external conditions, so the algorithm presented here is also suitable for these kinds of problems. To demonstrate the effectiveness of this new method, we simulated several examples of 2-D flow in slit nano-channels with different channel heights.

### Principle of the new method

Let us consider an isothermal pressure-driven steady state 2-D Poiseuille flow in a straight channel. Let  $x$  represent the direction of flow, while  $h$  represents the channel height in the  $y$  direction, that is the direction perpendicular to the flow. Each channel wall, i.e., the boundary perpendicular to the  $y$  direction, is modeled in terms of four rows of particles of solid wall material [5]. Initially, both the fluid particles and the solid wall particles are positioned in fcc lattices [23]. In the  $x$  direction we use the periodic boundary conditions [24].

If  $\mathbf{v}$  represents the total velocity vector of the fluid, it can be expressed by

$$\mathbf{v} = \mathbf{v}^T + \mathbf{v}^e \tag{1}$$

In a macroscopic problem, if there is no external force, the fluid will be in an equilibrium state, and the flow velocity  $\mathbf{v} = \mathbf{0}$ , but when we simulate a nanoscale problem with the MDS method, even if there is no external force and it is an equilibrium molecular dynamics problem, the calculated flow velocity will never be zero. Certainly, at this time, the nonzero flow velocity is due to the modeling and numerical errors. We represent this part of the velocity by  $\mathbf{v}^T$ . When external forces are exerted, a real flow velocity of the fluid will exist. We represent this part of velocity by  $\mathbf{v}^e$ . Theoretically,  $\mathbf{v}^T$  should be zero, but numerically it is not zero. In fact, in MDS  $\mathbf{v}^T$  is not only nonzero, but it is quite big and usually equals several hundred meters per second, so if the real flow velocity is small, it is difficult to know the exact value of  $\mathbf{v}^e$  according to equation (1).

In MDS methods [3-5], the flow velocity  $\mathbf{v}^e$  can be calculated by

$$\mathbf{v}^e = \lim_{N_t \rightarrow \infty} \frac{1}{N_t} \sum_{k=1}^{N_t} \mathbf{v}_k = \lim_{N_t \rightarrow \infty} \frac{1}{N_t} \sum_{k=1}^{N_t} (\mathbf{v}_k^T + \mathbf{v}_k^e), \tag{2}$$

where  $N_t$  is the number of time steps. The subscript  $k$  refers to the  $k^{\text{th}}$  time step. Equation (2) is based on the concept that if  $N_t$  approaches infinity

$$\mathbf{v}^T = \lim_{N_t \rightarrow \infty} \frac{1}{N_t} \sum_{k=1}^{N_t} \mathbf{v}_k^T = \mathbf{0}. \tag{3}$$

This is the traditional method used in most MDS algorithms. During the calculation we consider a small volume with a specific point inside and  $\mathbf{v}_k$  is the average velocity vector of all particles inside this volume at a particular moment [24]. Many studies verifying this method are available when the scalar velocity  $\mathbf{v}^e$  is not less than several meters per second, but when  $\mathbf{v}^e$  is much smaller than the scalar velocity  $\mathbf{v}^T$ , the above method does not work because, numerically,  $\mathbf{v}^T = \mathbf{0}$  is not true even when  $N_t$  is sufficiently big. Therefore we have to devise a new way to calculate the velocity  $\mathbf{v}^e$ .

If, at the  $k-1^{\text{th}}$  time step, we know the velocity of the  $i^{\text{th}}$  particle  $\mathbf{v}_{ik-1}$ , then, at the  $k^{\text{th}}$  time step, the velocity of the  $i^{\text{th}}$  particle  $\mathbf{v}_{ik}$  can be expressed by

$$\mathbf{v}_{ik} = \mathbf{v}_{ik-1}^T + \mathbf{v}_{ik-1}^e + \Delta \mathbf{v}_{ik}^T + \Delta \mathbf{v}_{ik}^e \tag{4}$$

Let  $\mathbf{F}_{ik}$  represent the total force exerted on the  $i^{\text{th}}$  particle at the end of the  $k-1^{\text{th}}$  time step;  $\mathbf{F}_{ik}^T$  and  $\mathbf{F}_{ik}^e$  represent the force corresponding the equilibrium MDS problem and the force caused by the

external loads, respectively. If there is no external load then  $\mathbf{F}_{ik}^e = 0$ , but  $\mathbf{F}_{ik}^T$  is still nonzero. We have  $\mathbf{F}_{ik} = \mathbf{F}_{ik}^T + \mathbf{F}_{ik}^e$ . In equation (4) the increment of velocity  $\Delta\mathbf{v}_{ik}^T + \Delta\mathbf{v}_{ik}^e$  depends on the force  $\mathbf{F}_{ik}$ . Here  $\Delta\mathbf{v}_{ik}^T$  is the velocity increment caused by  $\mathbf{F}_{ik}^T$  at the  $k^{\text{th}}$  time step;  $\Delta\mathbf{v}_{ik}^e$  is the velocity increment caused by  $\mathbf{F}_{ik}^e$  at the  $k^{\text{th}}$  time step. So the flow velocity of the  $i^{\text{th}}$  particle at the  $k^{\text{th}}$  time step  $\mathbf{v}_{ik}^e$  can be calculated by

$$\mathbf{v}_{ik}^e = \mathbf{v}_{ik-1}^e + \Delta\mathbf{v}_{ik}^e = \mathbf{v}_{ik-1}^e + \frac{\mathbf{F}_{ik}^e}{m_i} \Delta t, \tag{5}$$

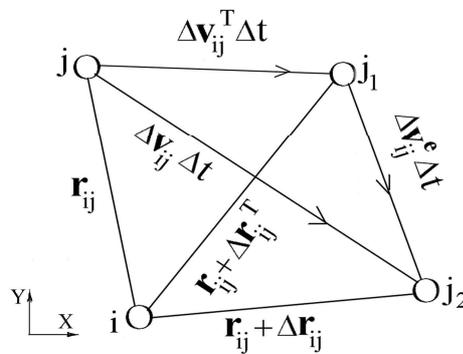
and we have the flow velocity

$$\mathbf{v}_{ik}^e = \sum_{k=1}^{N_t} \Delta\mathbf{v}_{ik}^e = \sum_{k=1}^{N_t} \frac{\mathbf{F}_{ik}^e}{m_i} \Delta t, \tag{6}$$

where the initial value of the flow velocity  $\mathbf{v}_{i0}^e = \mathbf{0}$  and  $m_i$  is the mass of the  $i^{\text{th}}$  particle.

Figure 1 illustrates the positions of particles  $i$  and  $j$  at moment  $t$  and  $t + \Delta t$ , corresponding to the  $k-1^{\text{th}}$  and the  $k^{\text{th}}$  time steps, respectively. The vector distance between these two particles at moment  $t$  is  $\mathbf{r}_{ij}$ . The vector velocity difference between these two particles is  $\Delta\mathbf{v}_{ij}$ . Because only the relative positions is involved, without lose of generality, we can assume that the position of particle  $i$  is not changed. At moment  $t + \Delta t$ , because of the velocity difference  $\Delta\mathbf{v}_{ij}$ , the particle  $j$  moves from point  $j$  to point  $j_2$ . The distance between these two particles is  $\mathbf{r}_{ij} + \Delta\mathbf{r}_{ij}$ , where  $\Delta\mathbf{r}_{ij} = \Delta\mathbf{v}_{ij}\Delta t$ . The distance change is  $\Delta\mathbf{r}_{ij} = \Delta\mathbf{r}_{ij}^T + \Delta\mathbf{r}_{ij}^e$ , where  $\Delta\mathbf{r}_{ij}^T = \Delta\mathbf{v}_{ij}^T\Delta t$ ;  $\Delta\mathbf{r}_{ij}^e = \Delta\mathbf{v}_{ij}^e\Delta t$ ;  $\Delta\mathbf{v}_{ij}^T$  and  $\Delta\mathbf{v}_{ij}^e$  are the velocity increments caused by the heat movement and the external forces, respectively.

**Figure 1.** Distance between particles  $i$  and  $j$  at moment  $t$  and  $t + \Delta t$ .



From Figure 1 we see that  $\Delta\mathbf{r}_{ij} = \Delta\mathbf{r}_{ij}^T + \Delta\mathbf{r}_{ij}^e = \Delta\mathbf{v}_{ij}^T\Delta t + \Delta\mathbf{v}_{ij}^e\Delta t$ . It means the distance between particles  $i$  and  $j$  is influenced by both  $\Delta\mathbf{v}_{ij}^T$  and  $\Delta\mathbf{v}_{ij}^e$ . Obviously, the distances between particles  $i$  and  $j$  are different for  $\Delta\mathbf{v}_{ij}^e = 0$  and  $\Delta\mathbf{v}_{ij}^e \neq 0$ . Also, the interaction forces are different for  $\Delta\mathbf{v}_{ij}^e = 0$  and  $\Delta\mathbf{v}_{ij}^e \neq 0$  because they are functions of  $\mathbf{r}_{ij} + \Delta\mathbf{r}_{ij}$ . At moment  $t + \Delta t$ , the interaction force is  $f(\mathbf{r}_{ij} + \Delta\mathbf{r}_{ij}) = f(\mathbf{r}_{ij} + \Delta\mathbf{r}_{ij}^T + \Delta\mathbf{r}_{ij}^e)$ . The form of  $f(\mathbf{r}_{ij} + \Delta\mathbf{r}_{ij})$  depends on the potential function. Thus, we think the interaction force as consisting of two parts. The first part corresponds to  $\Delta\mathbf{v}_{ij}^e = 0$ , that is  $f(\mathbf{r}_{ij} + \Delta\mathbf{r}_{ij}^T)$ , we represent it by  $\mathbf{F}_{ij}^T$ . The second part is the additional force increment because  $\Delta\mathbf{v}_{ij}^e \neq 0$ , that is  $f(\mathbf{r}_{ij} + \Delta\mathbf{r}_{ij}^T + \Delta\mathbf{r}_{ij}^e) - f(\mathbf{r}_{ij} + \Delta\mathbf{r}_{ij}^T)$ , we represent it by  $\mathbf{F}_{ij}^{eI}$ . If there is no external force, we have  $\Delta\mathbf{v}_{ij}^e = 0$  and then  $\mathbf{F}_{ij}^{eI} = \mathbf{0}$ . So  $\mathbf{F}_{ij}^{eI}$  is caused by external force. Now  $\mathbf{F}_{ik}^e$  consists of two parts,  $\mathbf{F}_{ik}^e = \mathbf{F}_{ik}^{eI} + \mathbf{F}_{ik}^{eII}$ , where  $\mathbf{F}_{ik}^{eI}$  represents the interaction force due to the external load caused additional distance change between different particles;  $\mathbf{F}_{ik}^{eII}$  represents the external load exerted directly on

particle  $i$ , usually  $m_i \mathbf{g}_i$ .  $\mathbf{F}_{ik}^{eII}$  depends on the form of the external load. If the potential function between the  $i^{\text{th}}$  particle and the  $j^{\text{th}}$  particle is  $u(r_{ij})$  we have the following interaction force function

$$\mathbf{F}_{ij}^T + \mathbf{F}_{ij}^{eI} = -\nabla u(r_{ij}) = -\frac{\partial u(r_{ij})}{\partial r_{ij}} \frac{\mathbf{r}_{ij}}{r_{ij}}. \quad (7)$$

The interaction force increment between  $t$  and  $t+\Delta t$  is

$$\begin{aligned} \Delta \mathbf{F}_{ij}^T + \Delta \mathbf{F}_{ij}^{eI} &= \frac{d(\mathbf{F}_{ij}^T + \mathbf{F}_{ij}^{eI})}{dt} \Delta t \\ &= -\frac{d}{dt} \left( \frac{\partial u(r_{ij})}{\partial r_{ij}} \frac{\mathbf{r}_{ij}}{r_{ij}} \right) \Delta t \\ &= -\frac{\partial^2 u(r_{ij})}{(\partial r_{ij})^2} \frac{dr_{ij}}{dt} \frac{\mathbf{r}_{ij}}{r_{ij}} \Delta t - \frac{\partial u(r_{ij})}{\partial r_{ij}} \frac{d}{dt} \left( \frac{\mathbf{r}_{ij}}{r_{ij}} \right) \Delta t \\ &= -\frac{\partial^2 u(r_{ij})}{(\partial r_{ij})^2} \Delta \mathbf{r}_{ij} \frac{\mathbf{r}_{ij}}{r_{ij}} - \frac{\partial u(r_{ij})}{\partial r_{ij}} \frac{1}{r_{ij}^2} \left( \frac{d\mathbf{r}_{ij}}{dt} r_{ij} - \frac{dr_{ij}}{dt} \mathbf{r}_{ij} \right) \Delta t \\ &= -\frac{\partial^2 u(r_{ij})}{(\partial r_{ij})^2} \Delta \mathbf{r}_{ij} \frac{\mathbf{r}_{ij}}{r_{ij}} - \frac{\partial u(r_{ij})}{\partial r_{ij}} \frac{1}{r_{ij}} (\Delta \mathbf{r}_{ij} - \Delta r_{ij} \frac{\mathbf{r}_{ij}}{r_{ij}}) \\ &= -\frac{\partial^2 u(r_{ij})}{(\partial r_{ij})^2} (\Delta \mathbf{r}_{ij}^e + \Delta \mathbf{r}_{ij}^T) \frac{\mathbf{r}_{ij}}{r_{ij}} - \frac{\partial u(r_{ij})}{\partial r_{ij}} \frac{1}{r_{ij}} [(\Delta \mathbf{r}_{ij}^e + \Delta \mathbf{r}_{ij}^T) - (\Delta r_{ij}^e + \Delta r_{ij}^T) \frac{\mathbf{r}_{ij}}{r_{ij}}], \end{aligned} \quad (8)$$

where  $\Delta r_{ij}^e$  and  $\Delta \mathbf{r}_{ij}^e$  are the change of the scalar and vector distance  $r_{ij}$  and  $\mathbf{r}_{ij}$  caused by the external conditions, respectively;  $\Delta r_{ij}^T$  and  $\Delta \mathbf{r}_{ij}^T$  are the change of the scalar and vector distance  $r_{ij}$  and  $\mathbf{r}_{ij}$  without external loads, respectively. From (8) we know that the increment of the interaction force between the  $i^{\text{th}}$  particle and the  $j^{\text{th}}$  particle caused by the external conditions is

$$\Delta \mathbf{F}_{ij}^{eI} = -\frac{\partial^2 u(r_{ij})}{(\partial r_{ij})^2} \Delta r_{ij}^e \frac{\mathbf{r}_{ij}}{r_{ij}} - \frac{\partial u(r_{ij})}{\partial r_{ij}} \frac{1}{r_{ij}} (\Delta \mathbf{r}_{ij}^e - \Delta r_{ij}^e \frac{\mathbf{r}_{ij}}{r_{ij}}). \quad (9)$$

Thus, at the  $k^{\text{th}}$  time step, the total interaction force of the  $i^{\text{th}}$  particle, caused by the external conditions, can be expressed by

$$\mathbf{F}_{ik}^{eI} = \sum_j \mathbf{F}_{ijk}^{eI} = \mathbf{F}_{ik-1}^{eI} + \sum_j \Delta \mathbf{F}_{ijk-1}^{eI} = \mathbf{F}_{ik-1}^{eI} - \sum_j \left[ \frac{\partial^2 u(r_{ij})}{(\partial r_{ij})^2} \Delta r_{ijk-1}^e \frac{\mathbf{r}_{ijk-1}}{r_{ijk-1}} + \frac{\partial u(r_{ij})}{\partial r_{ij}} \frac{1}{r_{ijk-1}} (\Delta \mathbf{r}_{ijk-1}^e - \Delta r_{ijk-1}^e \frac{\mathbf{r}_{ijk-1}}{r_{ijk-1}}) \right], \quad (10)$$

$\Delta r_{ijk-1}^e$  and  $\Delta \mathbf{r}_{ijk-1}^e$  are the change of the scalar distance  $r_{ij}$  and the vector distance  $\mathbf{r}_{ij}$  at the  $k-1^{\text{th}}$  time step caused by the external conditions.  $\Delta r_{ijk-1}^e$  and  $\Delta \mathbf{r}_{ijk-1}^e$  can be calculated according to the positions of particle  $i$  and  $j$ . At the beginning of simulation, fluid particles are allowed to move without applying the external force until a thermodynamic equilibrium state is reached. At this stage, it is an equilibrium problem. Usually, this process needs thousands of time steps or more. Then the external forces are applied and the non-equilibrium simulation starts.

According to the geometrical relation in Figure 1 we have the following scalar distance expressions:

$$(r_{ij} + \Delta r_{ij})^2 = r_{ij}^2 + (\Delta v_{ij} \Delta t)^2 + 2\Delta t \Delta \mathbf{v}_{ij} \cdot \mathbf{r}_{ij}, \quad (11)$$

and

$$(r_{ij} + \Delta r_{ij}^T)^2 = r_{ij}^2 + (\Delta v_{ij}^T \Delta t)^2 + 2\Delta t \Delta \mathbf{v}_{ij}^T \cdot \mathbf{r}_{ij}. \quad (12)$$

Subtracting (12) from (11) and considering  $\Delta r_{ij}^e = \Delta r_{ij} - \Delta r_{ij}^T$ , after some mathematical manipulations, we obtain:

$$\begin{aligned} \Delta r_{ij}^e (2r_{ij} + \Delta r_{ij} + \Delta r_{ij}^T) &= (r_{ij} + \Delta r_{ij})^2 - (r_{ij} + \Delta r_{ij}^T)^2 \\ &= (\Delta v_{ij} \Delta t)^2 - (\Delta v_{ij}^T \Delta t)^2 + 2\Delta t \Delta \mathbf{v}_{ij} \cdot \mathbf{r}_{ij} - 2\Delta t \Delta \mathbf{v}_{ij}^T \cdot \mathbf{r}_{ij} \\ &= (\Delta v_{ij}^2 - (\Delta v_{ij}^T)^2) \Delta t^2 + 2\Delta t \Delta \mathbf{v}_{ij}^e \cdot \mathbf{r}_{ij} \\ &= \Delta \mathbf{v}_{ij}^e \cdot (2\Delta \mathbf{v}_{ij} - \Delta \mathbf{v}_{ij}^e) \Delta t^2 + 2\Delta t \Delta \mathbf{v}_{ij}^e \cdot \mathbf{r}_{ij}. \end{aligned} \quad (13)$$

On the left hand side of (13) both  $\Delta r_{ij}^e$  and  $r_{ij}$  are scalar lengths. Usually  $\Delta t$  is small and both  $\Delta r_{ij}$  and  $\Delta r_{ij}^T$  are much smaller than  $r_{ij}$ . We should notice here the smaller  $\Delta r_{ij}$  and  $\Delta r_{ij}^T$  are ensured by the size of  $\Delta t$ , so we do not need to assume the flow velocity is small. Considering that all the variables in (10) correspond to the  $k-1$ <sup>th</sup> time step, we add a subscript  $k-1$  to represent the value at this time step and then equation (13) can be simplified to

$$\Delta r_{ijk-1}^e = (\Delta \mathbf{v}_{ijk-1}^e \cdot (2\Delta \mathbf{v}_{ijk-1} - \Delta \mathbf{v}_{ijk-1}^e) \Delta t^2 + 2\Delta t \Delta \mathbf{v}_{ijk-1}^e \cdot \mathbf{r}_{ijk-1}) / (2r_{ijk-1}). \quad (14)$$

The procedure of the algorithm is: at the  $k$ <sup>th</sup> time step, for any particle  $i$ , we calculate  $\Delta r_{ijk-1}^e$  by formulae (14) and  $\Delta \mathbf{r}_{ijk-1}^e = \Delta \mathbf{v}_{ijk-1}^e \Delta t$ , then the force exerted on this particle at this moment  $\mathbf{F}_{ijk}^e$  and  $\mathbf{F}_{ik}^e$  by (10). Finally, the flow velocity increment  $\Delta \mathbf{v}_{ik}^e$  and the flow velocity  $\mathbf{v}_{ik}^e$  can be calculated by (5) and (6), respectively. When using (14) to calculate  $\Delta r_{ijk-1}^e$  the term  $\Delta \mathbf{v}_{ijk-1}^e$  is the velocity increment corresponding to the  $k-1$ <sup>th</sup> time step and it is already known.

The considered Poiseuille flow is a steady state problem. The system temperature should be fixed throughout the flow process. To avoid increases of the system temperature during the simulation due to external forces or other unphysical effects, a constant temperature constraint is added. If the external forces are included in the system, it becomes a non-equilibrium molecular dynamics problem. Reference [25] is a good review regarding nonequilibrium MDS in flow problems. In this case the movement equation of particle  $i$  is

$$m_i \dot{\mathbf{v}}_i = \mathbf{F}_i + m_i \mathbf{g}_i - \alpha (\mathbf{v}_i - \mathbf{v}_i^e). \quad (15)$$

This is so called Gaussian thermostat method for nonequilibrium flow problems [25, 26]. For equilibrium flow problems, both  $\mathbf{g}_i$  and  $\mathbf{v}_i^e$  are zero vectors in (15). Because the flow velocity should not affect the temperature of the system, in the last term of (15), we deduct it from the total velocity. Here  $\dot{\mathbf{v}}_i$  is the derivative vector of the velocity of the  $i$ <sup>th</sup> particle with respect to the time variable.  $m_i \mathbf{g}_i$  is the external force vector  $\mathbf{F}_i^{eII}$ .  $\mathbf{F}_i$  is the total interaction force exerted on the  $i$ <sup>th</sup> particle. It is composed of two parts,  $\mathbf{F}_i = \mathbf{F}_i^T + \mathbf{F}_i^{eI}$ .  $\mathbf{F}_i^T$  is the interaction force without external loads.  $\mathbf{F}_i^{eI}$  is the additional interaction force caused by the external force. We do not need to calculate  $\mathbf{F}_i$  separately. This means that during the simulation, we do not need to use the sum of  $\mathbf{F}_i^T$  and  $\mathbf{F}_i^{eI}$  to get  $\mathbf{F}_i$ . We can calculate  $\mathbf{F}_i$  according to  $\mathbf{F}_i = \sum_{j \neq i} (\mathbf{F}_{ij}^T + \mathbf{F}_{ij}^{eI}) = -\sum_{j \neq i} \nabla u(r_{ij})$  and the position of particles directly.  $\alpha$  is a

thermostat multiplier to keep the kinetic temperature constant in the system. It can be obtained from the constant temperature condition [25, 26]. This is one of the usually used methods in thermostat molecular dynamics. When calculate the system temperature, for each particle, the flow velocity should be subtracted from the total velocity.

### Simulations of 2-D Poiseuille flow

In this Poiseuille flow we assume that the external force exerted on fluid particle  $i$  is  $m_i \mathbf{g}_i$ , that is  $\mathbf{F}_{ik}^{\text{elI}} = m_i \mathbf{g}_i$ . Here  $\mathbf{g}_i$  is the equivalent constant gravity acceleration vector. There are  $N$  particles within a length  $L$  along the flow direction and a height  $h$  of the flow channel. The initial velocity is given by the original temperature. In these simulations the fluid material is liquid argon and the Lennard-Jones potential is used. The cutoff radius is taken as  $r_c = 2.5\sigma$ . The interaction between the liquid and the solid wall is considered by the interaction strength parameter of the wall particles to the liquid particles. As mentioned before, in our simulations, the solid wall particles are fixed. Certainly this fixed boundary particle assumption will influence the accuracy of the flow velocity a little but it is not affect the examination of the effectiveness of our new velocity extraction method. During the simulation we fix the positions of these four rows of particles on each side.

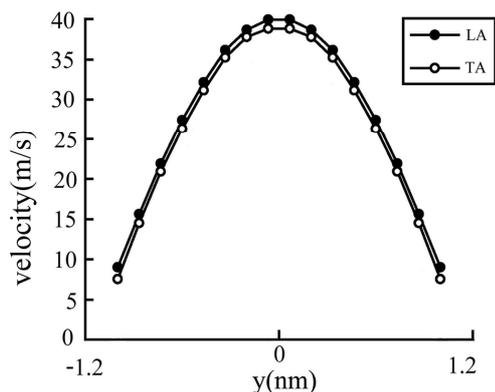
The values of the parameters used in the calculations are: the size of particles  $\sigma_f = \sigma_w = 3.4 \times 10^{-10}$  m; the mass of the particles  $m_f = m_w = 6.69 \times 10^{-26}$  kg; the temperature  $T = 84$  K; the interaction strength parameter between argon molecules is  $\varepsilon_f = 1.657 \times 10^6$  kg·m<sup>-1</sup>·s<sup>-1</sup>. In the above, the subscript  $f$  represents fluid and the subscript  $w$  represents solid wall. In our simulations, the size of each time step is chosen as  $\Delta t = 10^{-16}$  s. In many papers  $\Delta t = 10^{-14}$  s is recommended, but to ensure our linearized algorithm is stable  $\Delta t = 10^{-16}$  s is necessary. In principle, we can use different interaction strength parameters  $\varepsilon_w$  to represent different solid wall materials [24, 27]. We used  $\varepsilon_w = 3\varepsilon_f$  in our simulations. The interaction strength parameter for the interaction between a wall particle and a fluid particle  $\varepsilon_{wf}$  can be evaluated [24, 27] by  $\varepsilon_{wf} = (\varepsilon_f \varepsilon_w)^{1/2}$ . In all the simulations the density number is 0.82, corresponding to the liquid density of argon at  $T=84$  K.

We calculated two different cases with different channel heights. For the first case,  $L=13.61$  nm;  $h=2.31$  nm;  $N=243$ . For the second case,  $L=13.61$  nm;  $h= 4.36$  nm;  $N=459$ . The velocity profiles calculated in our simulations for these two cases are plotted in Figures 2-5.

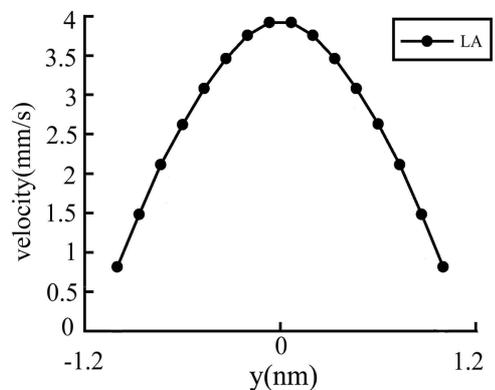
Because in this new method the total flow velocity  $\mathbf{v}^e$  is the linear accumulation of velocity increment of each time step we call it the LA method. The traditional method is based on the idea of time average, we call this the TA method. Figure 2 shows the simulation of case 1. In this simulation the equivalent acceleration in the flow direction is  $10^{13}$  m·s<sup>-2</sup> and the simulated maximum velocity which is calculated in the  $x$  direction by the method of this paper (the LA method) is  $39.9$  m·s<sup>-1</sup>. The MDS result where the velocity is calculated by the traditional method (TA method) is also shown in the Figure. Because the velocity is big both the TA and LA methods can be used to calculate the flow velocity. Figure 3 is another example of case 1, where the pressure difference is smaller. The equivalent acceleration in the flow direction is  $10^9$  m·s<sup>-2</sup> and the maximum velocity in the  $x$  direction is  $3.92 \times 10^{-3}$  m·s<sup>-1</sup>. Because the flow velocity is much smaller than the thermal velocity, the traditional method (TA method) is not available, but, as shown in the Figure, the LA method proposed in this paper is still effective. Figure 4 shows the simulation of case 2. In this simulation the equivalent

acceleration in the flow direction is  $10^{12} \text{ m}\cdot\text{s}^{-2}$  and the maximum velocity in the  $x$  direction is  $13.92 \text{ m}\cdot\text{s}^{-1}$ . Figure 5 is another example of case 2, where the equivalent acceleration in the flow direction is  $10^8 \text{ m}\cdot\text{s}^{-2}$  and the maximum velocity in the  $x$  direction is  $1.37 \times 10^{-3} \text{ m}\cdot\text{s}^{-1}$ . In Figure 3 and Figure 5 we can only get results for the LA method because in these simulation situations or for such small flow velocities the TA method is not available. In fact, when we simulated the same problems with the TA method under the same conditions as in Figure 3 and Figure 5, the maximal flow velocity is always great than  $2 \text{ m}\cdot\text{s}^{-1}$ . Certainly this is not the correct solution of the problems. Up to now, we have not found any algorithm which can be used to solve such a low velocity nanoscale flow problem. That is why the LA method is important for low flow velocity problems. However, we can compare the results of TA and LA methods for high flow velocity cases, as shown in Figure 2 and Figure 4. Because, in principle, the LA method can be used for both high and low flow velocity problems the conclusions obtained from the high flow velocity comparisons can be extended to low flow velocity problems.

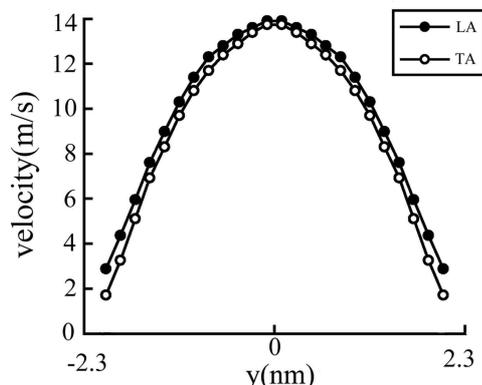
**Figure 2.** Poiseuille flow simulations of case 1 (high flow velocity simulations). The vertical coordinate is the flow velocity and the horizontal coordinate is the height of the channel.  $\text{---}\bullet\text{---}$  is the result of this paper (LA ).  $\text{---}\circ\text{---}$  is the result of the traditional (TA) method.



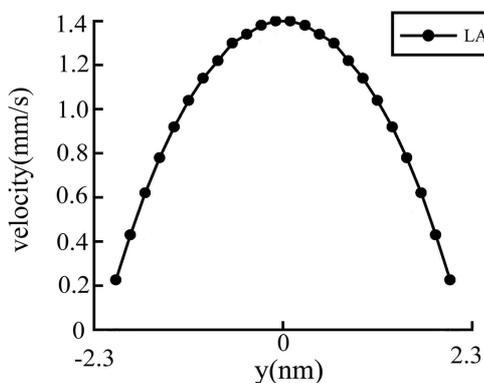
**Figure 3.** Poiseuille flow simulations of case 1 (low flow velocity simulations).  $\text{---}\bullet\text{---}$  is the result of this paper (LA ).



**Figure 4.** Poiseuille flow simulations of case 2 (high flow velocity simulations).  $\bullet$ — is the result of this paper (LA ).  $\circ$ — is the result of the traditional (TA) method.



**Figure 5.** Poiseuille flow simulations of case 2 (low flow velocity simulations).  $\bullet$ — is the result of this paper (LA ).



These numerical examples demonstrate that the velocity extraction method proposed in this paper is a suitable approach for both low and high speed flow problems. When the flow velocity is big the result of LA method is the same as the result of TA method. When the flow velocity is small TA method is not available we have to use LA method to extract  $v^e$ .

### Convergence and Efficiency analysis

In this new algorithm, we only used the linear terms of the development of the force function with respect to  $\Delta r_{ij}^e$  to approximate the force increment caused by the external conditions. What is the influence of the truncate errors of this approximation? Is it convergent and what are the conditions of convergence? These are the questions which should be answered for a reliable algorithm. We do a series of simulations with different size of time step from  $\Delta t = 10^{-14}$  s to  $\Delta t = 10^{-18}$  s. We find that when  $\Delta t = 10^{-14}$  s the simulation of our new velocity extraction method is not good. This means for the linear approximation the size of time step should be smaller. If the time step is big the small  $\Delta r_{ij}$  assumption will be broken. When  $\Delta t = 10^{-18}$  s the result is almost the same with the result of  $\Delta t = 10^{-17}$  s. So we have reason to believe that when  $\Delta t = 10^{-17}$  s the simulation is already convergent. We show the simulation results with time step size from  $\Delta t = 10^{-15}$  s to  $\Delta t = 10^{-17}$  s in Figure 6. From this figure we see that when  $\Delta t = 10^{-15}$  s the simulation result is far from convergence and when  $\Delta t = 10^{-16}$  s we obtain

a quite satisfactory result. If we let  $\Delta t = 10^{-17}$  s the result is only improved a little, so for practical applications, the recommended time step size is  $\Delta t = 10^{-16}$  s.

**Figure 6.** Comparison of simulation results with different  $\Delta t$  (LA Method). Low flow velocity simulations of case 2. The vertical coordinate is the flow velocity and the horizontal coordinate is the height of the channel.

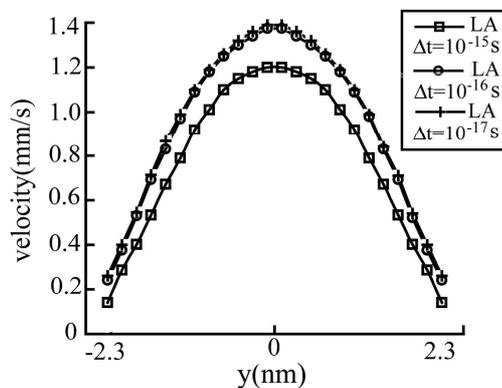
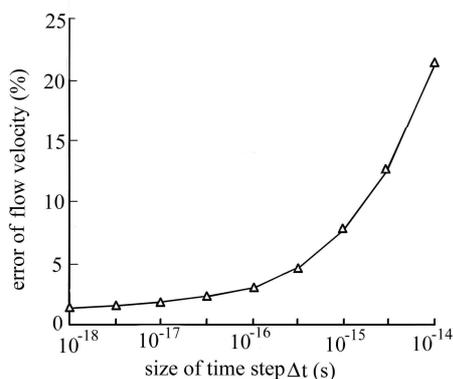


Figure 7 is the analysis of the error of the maximal flow velocity when the size of the time step is changed (LA Method). The simulation conditions are the same as the simulations in Figure 4. The vertical coordinate is the error of the maximal flow velocity  $|(v_{\max \text{ LA}} - v_{\max \text{ TA}})/v_{\max \text{ TA}}| 100\%$  and the horizontal coordinate is size of the time step  $\Delta t$  (s), where  $v_{\max \text{ LA}}$  is the maximal velocity of this paper and  $v_{\max \text{ TA}}$  is the maximal velocity of TA method.

**Figure 7.** Illustration of the relationship between the error of the maximal flow velocity and the size of the time step (LA Method).



Although the purpose of this new method is to extract the low flow velocity caused by the external conditions, which the conventional MDS method can not do, the efficiency of an algorithm is always an important part to be considered. In our new method we use  $\Delta t = 10^{-16}$  s. For the similar problems, in conventional MDS methods,  $\Delta t = 10^{-14}$  s is usually used. Because in this new method we do not need extra iterations to calculate the flow velocity, even 100 times smaller time step size is used, the total number of time steps is still less than that of the conventional MDS method. The simulation software was executed on a Pentium 2.8GHz PC. We compared the CPU time and the number of time steps as follows. In Figure 2, we used 89800 time steps (51 minutes CPU time) for LA method and  $3.2 \times 10^5$  time steps (108 minutes CPU time) for the TA method; in Figure 4, we used 288400 time steps (565

minutes CPU time) for the LA method and  $7.8 \times 10^5$  time steps (1022 minutes CPU time) for the TA method. The efficiency of the new method is evident.

## Conclusions

Using this method we can separate very small  $\mathbf{v}^e$  from the total velocity. It is very useful because in many cases we are more interested in the real movement of the medium. The new algorithm is also efficient compared to the traditional way. Because we do not need so many extra iterations as we usually do the new algorithm can reach a stable solution faster. This is important because usually the MDS use so much computation time that the number of particles is limited.

In this paper only the steady Poiseuille flow is simulated. Because, in this method, we do not use the time average algorithm the velocity computation is totally coincidence with the real transient state so the dynamic effective can be exhibited exactly. Hopefully this method will be suitable to simulate transient phenomena. It will be our later work. This method is also valid in other MDS problems where  $\mathbf{v}^e$  is important. In fact, in many MDS problems we want to know the small change of velocity caused by the exerted external conditions. The method will provide us a good way to solve this kind of problems.

## Acknowledgements

This work was supported by grants from Yanshan University, China.

## References

1. Polubotko, A.M.; Hatta, A.; Suzuki, Y.; Ciofalo, M.; Collins, M.W.; Hennessy, T.R. Modelling nanoscale fluid dynamics and transport in physiological flows. *Med. Eng. Phys.* **1996**, *18*, 437-451.
2. Giordano, N.; Cheng, J-T. Microfluid mechanics: progress and opportunities. *J. Phys. Condens. Mat.* **2001**, *13*, 271-295.
3. Travis, K.P.; Todd, B.D.; Denis, J.E. Departure from Navier-Stokes hydrodynamics in confined liquids. *Phys. Rev.* **1997**, *55*, 4288-4295.
4. Travis, K. P.; Gubbins, K. E. Poiseuille flow of Lennard-Jones fluids in narrow slit. *J. Chem. Phys.* **2000**, *112*, 1984-1994.
5. Thompson, P.A.; Robbins, M.O. Shear flow near solids: Epitaxial order and flow boundary conditions. *Phys. Rev. A* **1990**, *41*, 6830-6839.
6. Heinbuch, U.; Fischer, J. Liquid flow in pores: slip, no-slip or multiplayer sticking. *Phys. Rev. A* **1989**, *40*, 1144-1146.
7. Zhang, L.; Balasundaram, R.; Gehrke, S.H.; Jiang, S. Nonequilibrium molecular dynamics simulations of confined fluids in contact with the bulk. *J. Chem. Phys.* **2001**, *114*, 6869-6877.
8. Koplik, J.; Banavar, J.R.; Willemsen, J.F. Molecular Dynamics of Poiseuille flow and moving contact line. *Phys. Rev. Lett.* **1988**, *60*, 1282-1285.
9. Koplik, J.; Banavar, J.R. Corner flow in the sliding problem. *Phys. Fluids* **1995**, *7*, 3118-3125.

10. Todd, B.D.; Evans, D.J.; Daivis, P.J. Pressure tensor for inhomogeneous fluids. *Phys. Rev. E* **1995**, *52*, 1627-1638.
11. Moseler, M.; Landman, U. Formation, Stability and Breakup of Nanojets. *Science* **2000**, *289*, 1165-1169.
12. Eggers, J. Dynamics of liquid nanojets. *Phys. Rev. Lett.* **2002**, *89*, 084502.
13. Yang, J.X.; Koplik, J.; Banavar, J. R. Molecular Dynamics of Drop Spreading on a Solid Surface. *Phys. Rev. Lett.* **1991**, *67*, 3539-3542.
14. Klien, J.; Perahia, D.; Warburg, S. Forces between polymer-bearing surfaces undergoing shear. *Nature* **1991**, *352*, 143-145.
15. Fang, T.H.; Weng, C.I.; Chang, J.G. Molecular dynamics simulation of nano-lithography process using atomic force microscopy. *Surf. Sci.* **2002**, *501*, 138-147.
16. Fukui, K.; Sumpter, B.G.; Runge, K.; Kung, C.Y.; Barnes, M.D.; Noid, D.W. Collision dynamics and surface wetting of nano-scale polymer particles on substrates. *Chem. Phys.* **1999**, *244*, 339-349.
17. Landman, U.; Luedtke, W.D.; Burnham, N.A.; Colton, R.J. Microscopic Mechanisms and Dynamics of Adhesion, Microindentation and Fracture. *Science* **1990**, *248*, 454-461.
18. Kohlhoff, S.; Gumbsch, P.; Fischmeister, H.F. Crack propagation in b.c.c. crystals studied with a combined finite-element and atomistic model. *Philos. Mag. A* **1991**, *64*, 851-878.
19. Inoue, H.; Akahoshi, Y.; Harada, S. Molecular dynamics simulation on fracture mechanisms of nanoscale polycrystal under static and cyclic loading. *Int. J. Fatigue* **1997**, *19*, 266-266.
20. Sham, T.L.; Tichy, J. Scheme for hybrid molecular dynamics/finite element analysis of thin film lubrication. *Wear* **1997**, *207*, 100-106.
21. Wang, Y.U.; Jin, Y.M.; Cuitino, A.M.; Khachaturyan, A.G. Nanoscale phase field microelasticity theory of dislocations: model and 3D simulations. *Acta Mater.* **2001**, *49*, 1847-1857.
22. Matthey, T.; Hansen, J.P. Molecular dynamics simulation of sliding friction in a dense granular material. *Model. Simul. Mater. Sci. Eng.* **1998**, *6*, 701-707.
23. Somers, S. A.; Davis, H. T. Microscopic dynamics of fluids confined between smooth and atomically structured solid surfaces. *J. Chem. Phys.* **1992**, *96*, 5389-5407.
24. Rapaport, D.C. *The art of molecular dynamics simulation*; Cambridge University Press: Cambridge, **1995**; p. 16.
25. Todd, B.D. Computer simulation of simple and complex atomistic fluids by nonequilibrium molecular dynamics techniques. *Comput. Phys. Commun.* **2001**, *142*, 14-21.
26. Evans, D.J.; Morris, G.P. *Statistical mechanics of nonequilibrium liquids*; Academic Press: New York, London, **1990**.
27. Maitland G. C.; Rigby M.; Smith E. B.; Wakeham W. A. *Intermolecular forces*; Oxford University Press: New York, **1981**; p. 137.