



## Controllers for imposing continuum-to-molecular boundary conditions in arbitrary fluid flow geometries

Matthew K. Borg , Graham B. Macpherson & Jason M. Reese

To cite this article: Matthew K. Borg , Graham B. Macpherson & Jason M. Reese (2010) Controllers for imposing continuum-to-molecular boundary conditions in arbitrary fluid flow geometries, *Molecular Simulation*, 36:10, 745-757, DOI: [10.1080/08927021003752812](https://doi.org/10.1080/08927021003752812)

To link to this article: <https://doi.org/10.1080/08927021003752812>



© 2010 The Author(s). Published by Taylor & Francis.



Published online: 15 Oct 2010.



Submit your article to this journal [↗](#)



Article views: 443



Citing articles: 34 [View citing articles ↗](#)

## Controllers for imposing continuum-to-molecular boundary conditions in arbitrary fluid flow geometries

Matthew K. Borg<sup>a\*</sup>, Graham B. Macpherson<sup>b</sup> and Jason M. Reese<sup>a</sup>

<sup>a</sup>Department of Mechanical Engineering, University of Strathclyde, Glasgow G1 1XJ, UK; <sup>b</sup>OpenCFD Ltd, 9 Albert Road, Caversham, Reading, Berkshire RG4 7AN, UK

(Received 6 October 2009; final version received 5 March 2010)

We present a new parallelised controller for steering an arbitrary geometric region of a molecular dynamics (MD) simulation towards a desired thermodynamic and hydrodynamic state. We show that the controllers may be applied anywhere in the domain to set accurately an initial MD state, or solely at boundary regions to prescribe non-periodic boundary conditions (PBCs) in MD simulations. The mean molecular structure and velocity autocorrelation function remain unchanged (when sampled a few molecular diameters away from the constrained region) when compared with those distributions measured using PBCs. To demonstrate the capability of our new controllers, we apply them as non-PBCs in parallel to a complex MD mixing nano-channel and in a hybrid MD continuum simulation with a complex coupling region. The controller methodology is easily extendable to polyatomic MD fluids.

**Keywords:** molecular dynamics; hybrid MD continuum simulations; boundary conditions; nanofluidics; thermodynamic state; controllers

### 1. Introduction

The use of periodic boundary conditions (PBCs) is typical of molecular dynamics (MD) [1,2] simulations, as they provide a conservative, efficient, simple and robust implementation for studying systems at equilibrium. Despite these advantages, however, many engineering applications require a non-periodic alternative to control continuum properties at a boundary region. Some examples include MD systems with open boundaries, such as fluid flows requiring inlet and outlet boundary conditions, or non-equilibrium MD systems exhibiting gradients of density and temperature across boundaries of the domain. Additionally, an important class of examples is hybrid MD continuum methods [3–7] that require boundary conditions prescribed from a continuum solution at the coupling interface.

Various methods for controlling continuum properties have been proposed in the literature. Sun and Ebner [8] proposed a 1D ‘piston’ at a boundary that forces molecules into the bulk. At specified time intervals, the piston resets to its original position and inserts molecules within the dilute region so as to match the density. Methods for changing density in the global MD domain also exist, which require changing the volume of the domain and rescaling all molecular positions [2]. The application of a gravitational force to all molecules within the domain [9] is a widely used technique to drive Poiseuille-type flow configurations, but the channel centre velocity is typically an outcome of the simulation rather than an input.

Furthermore, the method still relies mostly on the application of PBCs, hence its usefulness still seems limited. O’Connell and Thompson [3] and Nie et al. [4] derived a constrained Lagrangian technique for imposing momentum in hybrid simulations of liquids, while Flekkøy et al. [5] and Delgado-Buscalioni and Coveney [6] instead proposed flux exchange schemes between continuum and MD formulations. Finally, concepts derived from control theory are being established as plausible methods for changing and converging the state of a fluid in MD simulations. To the author’s knowledge, the feedback control concept in MD was originally described by Li et al. [10], and further exploited by Kotsalis et al. [11,12] for controlling 1D density error gradients, next to non-periodic planar boundaries.

Several issues surround the models proposed in the literature. First, the use of *global* Cartesian coordinates for imposing *local* perturbations on molecules precludes the imposition of spatially varying properties in complex 3D regions. Typically, control is applied to 1D boundary regions in cuboid-geometry domains with target values that are spatially uniform. Second, there is a lack of generality and flexibility in the proposed models, such as the ability to regulate the frequency of control, the applicability of the models to both pure and multi-species fluids and the relative ease of applying many controllers within an MD simulation. Finally, it is sometimes unclear from the publications whether the algorithms have been designed and applied for parallel processing, an issue which is of

\*Corresponding author. Email: matthew.borg@strath.ac.uk

utmost importance in computationally intensive simulations. These limitations, therefore, impede MD and hybrid MD continuum simulations of more realistic engineering applications.

In this paper, we present a new set of controller algorithms for imposing density and velocity. These have several unique features. The mesh is kept static, no moving pistons are used nor is the simulation domain permitted to expand or contract. *Individual* cells on the mesh are utilised for *localised* control of continuum properties. Parallelisation on distributed processors is achieved, spatially varying flows are obtained and control is possible in complex 3D regions. Moreover, non-periodic boundary conditions (NPBCs) may be applied at generally complex boundaries of MD simulations, or at the arbitrary shaped interfaces of hybrid MD continuum simulations. In such a hybrid scheme, the continuum finite-volume cells are linked to the molecular cells via the controllers. In addition, a feedback loop algorithm is implemented; for density control, molecules are inserted or deleted depending on the error between the measured and target density within an arbitrary cell. For velocity control, an external force is incorporated in the equations of motion of molecules occupying the cell, which corresponds to the velocity error relating the measured to target velocity within the same cell. Finally, a flexible decoupled time scheme is proposed on a per-controller basis so that an arbitrary continuum property may be controlled at frequencies appropriate to the time variations of the target property dictated by the problem, and measured over longer timescales to increase its statistical accuracy when supplied in the feedback loop.

In this paper, our new control method is described in Section 2, and test cases demonstrated in Section 3. Conclusions are made in Section 4.

## 2. Method

We consider a domain of arbitrary geometry, defined by an unstructured polyhedral mesh, typically used in finite-volume computational fluid dynamics (CFD). The MD fluid occupies the domain geometry and consists of point masses (which we will refer to as ‘molecules’) that interact through a pairwise potential  $U(r_{ij})$ , where  $r_{ij} = |\mathbf{r}_{ij}|$  and  $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$  is the separation vector between a pair of molecules ( $i, j$ ). The standard shifted Lennard-Jones (LJ) 12-6 potential [2] is used:

$$U(r_{ij}) = \begin{cases} 4\varepsilon \left( \left( \frac{r_{ij}}{\sigma} \right)^{-12} - \left( \frac{r_{ij}}{\sigma} \right)^{-6} \right) - U_{\text{LJ}}(r_{\text{cut}}), & r_{ij} \leq r_{\text{cut}}, \\ 0, & r_{ij} > r_{\text{cut}}, \end{cases} \quad (1)$$

where  $\sigma$  is the characteristic length scale,  $\varepsilon$  is the characteristic potential energy,  $r_{\text{cut}}$  is the cut-off radius and

$U_{\text{LJ}}(r_{\text{cut}})$  is the potential energy value at which the LJ potential gets truncated.

The positions  $\mathbf{r}_i$  and velocities  $\mathbf{v}_i$  of molecules evolve according to standard Newtonian dynamics

$$m_i \mathbf{a}_i = \mathbf{f}_i, \quad (2)$$

where  $\mathbf{a}_i = \ddot{\mathbf{r}}_i$  and  $m_i$  are the molecule’s acceleration and mass, respectively. The Verlet Leapfrog algorithm [2] is used to integrate numerically the equations of motion, by a numerical MD time step  $\Delta t_m$ : update first the mid-step velocity  $\mathbf{v}_i(t + \Delta t_m/2) = \mathbf{v}_i(t) + 1/2 \mathbf{a}_i(t) \Delta t_m$  and advance the molecules to their new positions  $\mathbf{r}_i(t + \Delta t_m) = \mathbf{r}_i(t) + \mathbf{v}_i(t + \Delta t_m/2) \Delta t_m$ . Here, we use an efficient tracking scheme [13] on molecules as they move within the mesh. Molecules get transferred across processors as they collide with interprocessor boundaries, and also a molecule’s cell occupancy is updated as it moves from cell to cell of the mesh. Next, we compute the intermolecular force on all the molecules:

$$\mathbf{f}_i = \sum_{j=1(\neq i)}^{N_{\text{mol}}} f(r_{ij}) \frac{\mathbf{r}_{ij}}{|\mathbf{r}_{ij}|}, \quad (3)$$

where  $f(r_{ij}) \mathbf{r}_{ij}/|\mathbf{r}_{ij}| = -\nabla U(r_{ij})$  is the pair force potential and  $N_{\text{mol}}$  is the number of molecules located within the sphere of radius  $r_{\text{cut}}$ , centred at  $\mathbf{r}_i$ . This step makes use of the arbitrary interacting cells algorithm scheme [14], which is a generalisation of the linked cells algorithm [1] for computing intermolecular pair forces in complex meshes. The acceleration is given by  $\mathbf{a}_i(t + \Delta t_m) = \mathbf{f}_i/m_i$ . Finally, we update the second-step velocity  $\mathbf{v}_i(t + \Delta t_m) = \mathbf{v}_i(t + \Delta t_m/2) + (1/2) \mathbf{a}_i(t + \Delta t_m) \Delta t_m$ . The MD code which we use, including the algorithms briefly mentioned above, is implemented in OpenFOAM v. 1.6 (the open source CFD toolbox; available online: [www.openfoam.org](http://www.openfoam.org)), which is open source and available to be downloaded freely from OpenFOAM.

### 2.1 Controller methodology

Our general controller methodology is used to converge continuum properties within individual cells of the mesh (see Figure 1).

The methods for imposing non-equilibrium MD proposed in the literature generally involve applying additional perturbations (e.g. in the form of external forces  $\mathbf{f}_i^{\text{ext}}$ ) to molecules by using global coordinates. An example is a code that loops over all molecules in the domain and applies control if the molecule’s position is within, say,  $x_{\text{min}} \leq x_i \leq x_{\text{max}}$  (e.g. [4]).

Instead, we consider applying external perturbations locally using control cells, which is facilitated by the cell occupancy data structure that gets updated every time step: an arbitrary cell on the mesh stores an inexpensive link to

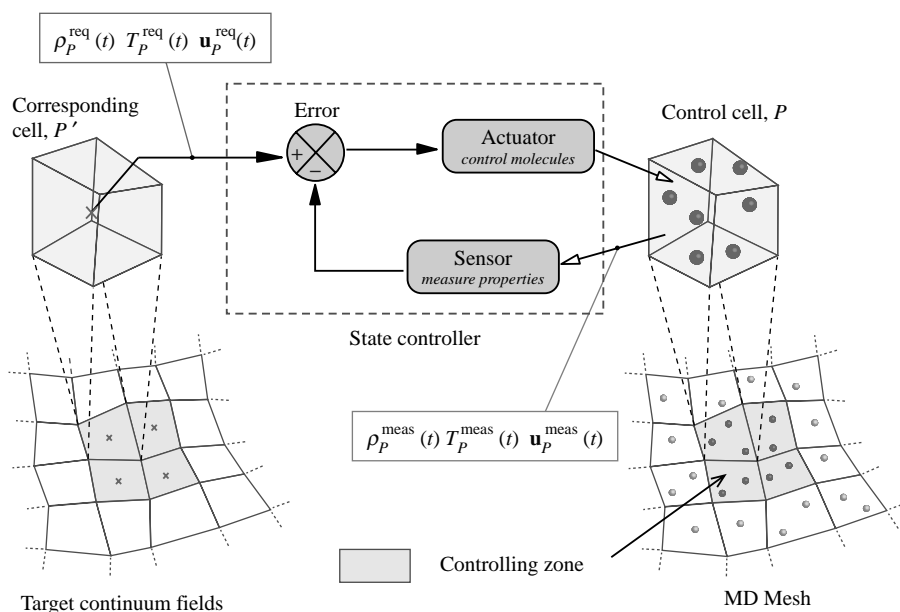


Figure 1. Schematic showing the operation of a generic controller using a cell-based approach.

those molecules that are currently residing within. This method introduces numerous benefits. First, control is implemented by looping over molecules within the cell, rather than over molecules within the system. Second, the controlling zone (a group of control cells) can be any arbitrary 3D shape, size and location within the simulation domain. For example, the control zone could be selected as the entire domain in order to converge to an initial MD state, and thereafter solely as the boundary regions in order to implement NPBCs. Third, spatially varying properties may be imposed within MD simulations using this local approach by specifying different target properties per control cell. Finally, the controller code is parallelised, since the MD mesh geometry is partitioned onto separate processors based on the cells that comprise the mesh (software package and libraries for sequential and parallel graph partitioning, static mapping and sparse matrix block ordering and sequential mesh and hypergraph partitioning; available online: [www.labri.fr/perso/pelegrin/scotch/](http://www.labri.fr/perso/pelegrin/scotch/)).

Fields of target continuum properties are prescribed to the state controller (see Figure 1). Each entry within a field corresponds to an MD cell in the controlling zone: an arbitrary control cell  $P$  ‘knows’ the required cell-centred density,  $\rho_P^{\text{req}}$ , velocity  $\mathbf{u}_P^{\text{req}}$  and temperature  $T_P^{\text{req}}$  at any time  $t$  of the simulation. Throughout the simulation, the fields may either be uniform, or be altered by some form of external function (temporally and/or spatially) or be directly modified by the continuum solution in an overlapping region if it is a hybrid simulation. In the latter, the overlaying coupling region is identical to the MD controlling zone: an arbitrary finite-volume CFD cell  $P'$

corresponds to the MD control cell  $P$  in shape, size and global coordinate.

### 2.1.1 Control functions

A simple closed-loop (negative feedback) control system is implemented. The three functions of the control system are defined as follows, see Figure 1:

- (1) *Sensor* – measures the macroscopic property from a zone cell  $P$ , using a bin-averaging technique.
- (2) *Error* – computes the difference between the target property in  $P'$  and the measured property in its corresponding cell  $P$ , and converts it into a quantity that is best suited for the actuation function. For example, the change in density has to be converted into a number of missing/extra molecules required within the cell.
- (3) *Actuator* – imposes the necessary control operations on molecular variables occupying  $P$ , based on the error signal.

### 2.1.2 Time scheme

We devise a flexible decoupled time scheme for measurement and control of an arbitrary continuum property within its controller architecture. The scheme is decoupled because control and measurement of the macroscopic property may occur at independent frequencies and over different timescales.

Timescales relating to *measurement* include the sampling time,  $\Delta t_s$ , which defines the time period between cell-averaged samples, and the averaging time,  $\Delta t_{av}$ , which

defines the time over which samples are accumulated and time averaged. The relationship between them is the number of samples:

$$S = \frac{\Delta t_{\text{av}}}{\Delta t_s}. \quad (4)$$

$\Delta t_{\text{av}}$  is dependent on the cell size and property being controlled [15], so that an accurate mean value is computed.

*Control* is applied at well-defined time intervals,  $\Delta t_{\text{cont}}$ . Between controlling steps, no perturbations are imposed within the control cells, so as to allow local equilibration of the fluid after being exposed to the controller's actions. The frequency of control, and hence the strength of coupling the measured property with the target, is therefore dependent on the value chosen for  $\Delta t_{\text{cont}}$ .

Control and measurement processes are effectively *linked*, by a common time interval,  $\Delta t = n_{\text{av}}\Delta t_{\text{av}} = n_{\text{cont}}\Delta t_{\text{cont}}$ , where  $n_{\text{av}}$  and  $n_{\text{cont}}$  are the number of averages and number of control steps, respectively, that are performed within the time interval. Practically,  $n_{\text{av}} = 1$  so that  $\Delta t = \Delta t_{\text{av}}$  is the maximum possible averaging time interval to reduce the statistical error of the measured property. Therefore, a property measurement carried out during the previous time interval  $\Delta t$ , is used to compute the necessary controlling actions to take place over the next  $\Delta t$ .

A key benefit of our proposed time scheme is that modification to  $\Delta t_{\text{cont}}$  is permitted in order to change the rate of control, without restricting  $\Delta t_{\text{av}}$  and hence affecting the accuracy of any measurement. We highlight this as an essential requirement in the controller methodology; poor sampling due to a small  $\Delta t_{\text{av}}$  will feed a noisy error signal to the actuator that may result in an unstable diverging state.

The most challenging part of a decoupled time scheme is the requirement that the controlling models distribute alterations on molecular variables over a series of  $n_{\text{cont}}$  control time steps, as illustrated in Figure 2.

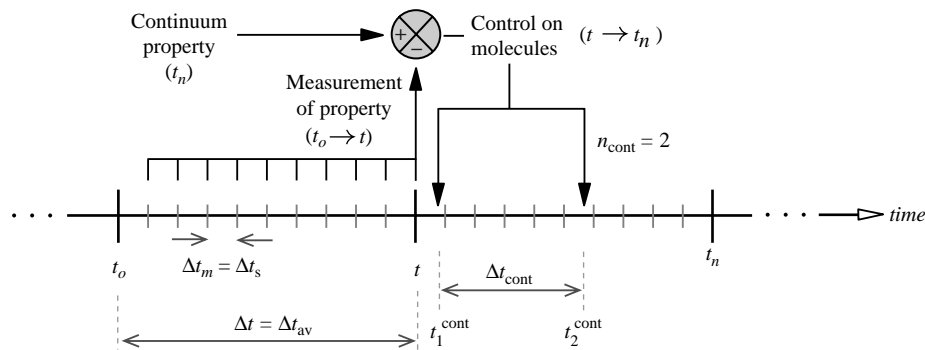


Figure 2. Schematic showing the timeline for an MD simulation with control. Processes of measurement and control are decoupled, but linked by a common time interval  $\Delta t$ .

## 2.2 Density control

Density is controlled within an arbitrary control cell  $P$ , using the following procedure:

- (1) At a time  $t$ , compute the density error in  $P$  and translate this into ‘molecules’ using the following formula:

$$\Delta N_P(t \rightarrow t_n) = \text{NINT}([\rho_P^{\text{req}}(t_n) - \rho_P^{\text{meas}}(t)]V_P), \quad (5)$$

where  $\rho_P^{\text{meas}}(t)$  is the measured cell density during the previous time period ( $t_o \rightarrow t$ ),  $\rho_P^{\text{req}}(t_n)$  is the required cell density at a later time ( $t_n = t + \Delta t$ ) and  $\Delta N_P(t \rightarrow t_n)$  are the total number of molecules to control from cell  $P$  during the next  $\Delta t$ . The  $\text{NINT}(x)$  function is required to apply the nearest integer, since only whole molecules may be introduced ( $\Delta N_P > 0$ ) or removed ( $\Delta N_P < 0$ ). If  $\rho_P^{\text{req}}(t_n)$  is not known at a time  $t_n$ , it may be extrapolated from old-time quantities:  $\rho_P^{\text{req}}(t_n) = 2\rho_P^{\text{req}}(t) - \rho_P^{\text{req}}(t_o)$ .

- (2)  $\Delta N_P$  is divided equally across the subsequent  $n_{\text{cont}}$  control steps:

$$\Delta N_P(t_j^{\text{cont}}) = \begin{cases} \text{ceil}(\Delta N_P/n_{\text{cont}}), & \text{if } \Delta N_P > 0, \\ \text{floor}(\Delta N_P/n_{\text{cont}}), & \text{if } \Delta N_P < 0, \end{cases} \quad (6)$$

$t \leq t_j^{\text{cont}} < t_n$ ,  $\{j = 1, 2, \dots, n_{\text{cont}}\}$ , followed by a residual correction to  $\Delta N_P(t_j^{\text{cont}})$  at each control step, so that the prescribed density is matched accurately during ( $t \rightarrow t_n$ ).

- (3) At an arbitrary control step,  $t_j^{\text{cont}}$ , two types of mass residuals are corrected. The number of molecules that *fail* to be inserted or deleted in a previous control step,  $\delta N_P^{\text{fail}}(t_{j-1}^{\text{cont}})$ , are added to  $\Delta N_P(t_j^{\text{cont}})$ . Then, the cumulative error resulting from the ceiling/flooring functions in Equation (6) is checked:

If  $|\Delta N_P(t_j^{\text{cont}}) + \Delta N_P^{\text{success}}| > |\Delta N_P|$ ,  
 Then modify  $\Delta N_P(t_j^{\text{cont}})$  to:

$$\Delta N_P(t_j^{\text{cont}}) = \Delta N_P(t \rightarrow t_n) - \Delta N_P^{\text{success}}, \quad (7)$$

where  $\Delta N_P^{\text{success}}$  is the cumulative number of molecules controlled within previous control steps. Note that the terms  $\Delta N_P^{\text{fail}}$  and  $\Delta N_P^{\text{success}}$  are positive or negative depending on whether  $\Delta N_P > 0$  or  $\Delta N_P < 0$ , respectively. Furthermore,  $\Delta N_P^{\text{success}}$  is reset to zero after the whole time interval ( $t \rightarrow t_n$ ), while  $\Delta N_P^{\text{fail}}$  is set to zero after every control step.

- (4) Insertion and deletion processes are performed immediately after the intermolecular force calculation step of the Leapfrog algorithm, and are described separately below.

### 2.2.1 Inserting molecules

If  $\Delta N_P > 0$ , the actuation function inserts these whole molecules sequentially into the cell  $P$ . During this process, existing molecules that currently reside within the domain remain fixed in space and time. The procedure for inserting one molecule,  $i$ , is described.

- (1) Search for an insertion site,  $\mathbf{r}_i$ , that ensures non-overlapping molecules. We employ the USHER algorithm [16], which performs a steepest-descent iterative search in the potential energy landscape. A site to insert a molecule is accepted if its potential energy is equal to the average potential energy per molecule within the cell, i.e.

$$U_P^{\text{req}} = \langle U_i \rangle_P = \frac{\sum_{k=1}^S \sum_{i=1}^{N_P(t_k)} \frac{1}{2} \sum_{j=1(\neq i)}^{N_{\text{mol}}} U(r_{ij})}{\sum_{k=1}^S N_P(t_k)}, \quad (8)$$

where  $N_P(t_k)$  is the number of molecules residing in  $P$  at time  $t_k$ .

- (2) Create a molecule at  $\mathbf{r}_i$ , and update the acceleration and potential energy of the surrounding molecules  $j$ , within interaction range:

$$\mathbf{a}_j^n = \mathbf{a}_j^o - \frac{f(r_{ij})\mathbf{r}_{ij}}{m_j|\mathbf{r}_{ij}|}, \quad (9)$$

$$U_j^n = U_j^o + \frac{U(r_{ij})}{2}. \quad (10)$$

- (3) The initial velocity,  $\mathbf{v}_i$ , of the newly inserted molecule is sampled randomly from a Maxwell–Boltzmann distribution at the required temperature,  $T_P^{\text{req}}$ , and mean velocity,  $\mathbf{u}_P^{\text{req}}$ .

### 2.2.2 Deleting molecules

If  $\Delta N_P < 0$ , molecules may be deleted sequentially using the reverse of the insertion process. Deletion is more computationally efficient and also more successful than insertion. Existing molecules within the cell are held fixed in time and space as candidate molecules are deleted. The process for deleting one molecule is described:

- (1) Initially, select a candidate molecule  $i$  from a cell  $P$ , using a criterion that maintains the potential energy of the cell the same. We use a scheme that loops over all molecules within cell  $P$  and chooses the molecule with its potential energy closest to  $U_P^{\text{req}}$ .
- (2) Update the accelerations and potential energies of surrounding molecules to account for the molecule being deleted. These equations are similar to Equations (10) and (9) but with the opposite signs in front of the second terms on the RHS.
- (3) Delete molecule  $i$  from cell  $P$ .

### 2.2.3 Multi-species control

Our density controller is designed to operate on both single- and multi-species fluid systems. For multi-species fluids, the set-up consists of applying a density controller per species in a common controlling zone, and providing the controllers with the target partial densities within the control cells.

### 2.2.4 Parallelisation

The density controller is parallelised by allowing processors to communicate during the control steps in order to check for conflicts immediately after the insertion site, and/or candidate molecules for deletion have been identified on each processor. A conflict occurs if any two or more designated molecules are within  $r_{\text{cut}}$ . The conflict is resolved by arbitrarily assigning a priority to each processor. The processor with the highest priority proceeds with the insertion/deletion of the molecule, while processors of lower priority that detect a conflict terminate their tries and reattempt them in the next try.

## 2.3 Velocity control

The convective velocity is controlled within  $P$  by the following procedure:

- (1) At time  $t$ , we compute the velocity error that is to be imposed per control step:

$$\begin{aligned} \Delta \mathbf{u}_P(t_j^{\text{cont}}) &= \frac{\Delta \mathbf{u}_P(t \rightarrow t_n)}{n_{\text{cont}}} \\ &= \frac{\lambda}{n_{\text{cont}}} (\mathbf{u}_P^{\text{req}}(t_n) - \mathbf{u}_P^{\text{meas}}(t)), \end{aligned} \quad (11)$$

where  $\mathbf{u}_P^{\text{meas}}(t)$  is the measured cell velocity at current time  $t$ ,  $\mathbf{u}_P^{\text{req}}(t_n)$  is the required cell velocity at a later time ( $t_n = t + \Delta t$ ) and  $\lambda/n_{\text{cont}} = K_p$  is the proportional gain of the controller. The latter is a modifiable parameter (typically  $\lambda \approx 1$  and  $n_{\text{cont}} \approx 25$ ) that defines the rate at which the velocities of molecules occupying cell  $P$  are accelerated towards  $\mathbf{u}_P^{\text{req}}(t_n)$ . Similarly, if  $\mathbf{u}_P^{\text{req}}(t_n)$  is unknown at a time  $t_n$ , it too may be extrapolated from known quantities:  $\mathbf{u}_P^{\text{req}}(t_n) = 2\mathbf{u}_P^{\text{req}}(t) - \mathbf{u}_P^{\text{req}}(t_o)$ .

- (2) At each control step, we add an external force to the equations of motion of *all* molecules  $i$  occupying cell  $P$ , after the intermolecular force calculation step:

$$\mathbf{f}_i^{\text{ext}} = \frac{\Delta \mathbf{u}_P(t_j^{\text{cont}})}{\Delta t_m} m_i. \quad (12)$$

## 2.4 Temperature control

For temperature control, we implement the popular thermostats [2] using a cell-based approach. In this paper, we adopt the velocity scaling thermostat [2], which scales molecular velocities based on the root of the ratio between target and measured temperatures.

## 2.5 NPBCs in arbitrary geometries

Our implementation of NPBCs comprises control of density, velocity and temperature at a boundary region in addition to mass, momentum and energy flow at the same boundary. The latter we now describe briefly. Consider a region located close to an arbitrary non-periodic boundary

$b$  of an MD computational domain. The boundary region is divided into two: an initial layer of thickness  $r_{\text{cut}}$  adjacent to the MD terminating boundary acts as a *buffer* region for imposing flow of molecular properties; an adjacent *control* region is where the controllers are applied. The thickness of the control region is selected to be  $\sim 10\sigma$ , comprising two layers of cells of  $\sim 5\sigma$  each (see Figure 3).

For mass and momentum flow, we consider a mass flux model that is similar to the density controller but inserts/deletes molecules close to the boundary based on a prescribed continuum mass flux  $\dot{m}_f = (\rho \mathbf{u})_f \cdot \mathbf{A}_f$ , where  $f$  is an arbitrary boundary face,  $\mathbf{A}_f = A_f \hat{\mathbf{n}}_f$  is the normal face-area vector and  $(\rho \mathbf{u})_f$  are the face-interpolated values for density and velocity. Potential energy flow is imposed by appending to the equations of motion of those real molecules located within the buffer region, an external force that accounts for the missing molecules beyond the boundary at every time step, and an update of the missing potential energy and virial. An arbitrary molecule  $i$  located at a normal distance  $r_{bi} = \mathbf{r}_{fi} \cdot \hat{\mathbf{n}}_f$  from the closest boundary face  $f$  is given an additional force  $\mathbf{f}_i^{\text{ext}} = -f_{bi}(r_{bi}) \hat{\mathbf{n}}_f$ , where  $\mathbf{r}_{fi} = \mathbf{r}_f - \mathbf{r}_i$ . Additionally, a potential energy update of  $U_i^n = U_i^o + U_{bi}(r_{bi})$  and a virial update  $w_i^n = w_i^o + w_{bi}(r_{bi})$  are performed, where  $U_i = 1/2 \sum_{j(\neq i)}^{N_{\text{mol}}} U(r_{ij})$  and  $w_i = 1/2 \sum_{j(\neq i)}^{N_{\text{mol}}} f(r_{ij}) r_{ij}$ . The missing virial contribution is required for measuring pressure variations correctly at the non-periodic boundary. The distributions are shown in Figure 4, and are sampled from an equilibrium MD simulation at state point  $\rho = 0.6(\sigma^{-3})$ ,  $T = 2.4(\epsilon/k_b)$ . A ‘Maxwell Demon’ technique [17] is also applied to molecules in the terminating buffer region. Molecules are collided at random with a momentum and heat bath using a

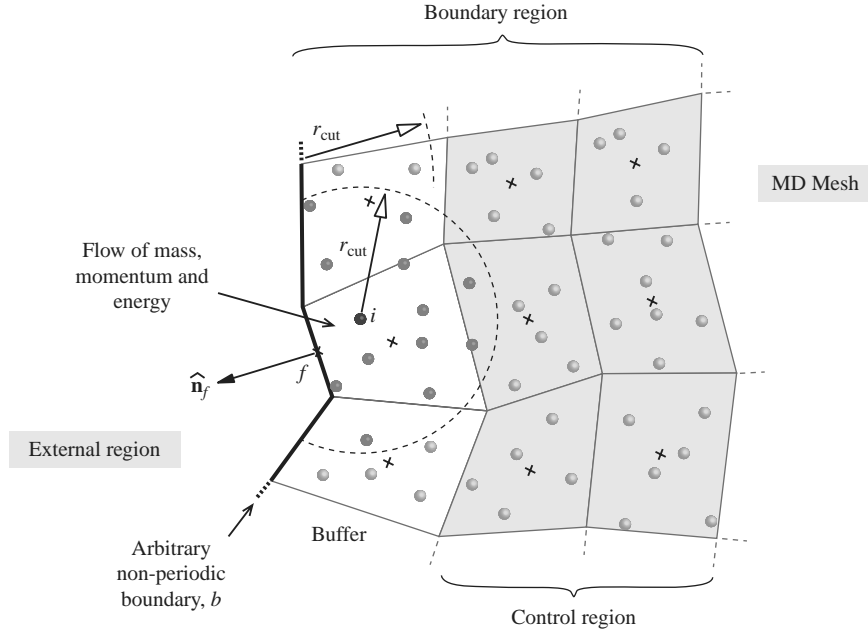


Figure 3. Schematic showing the implementation of NPBCs using the controllers and other flux models.

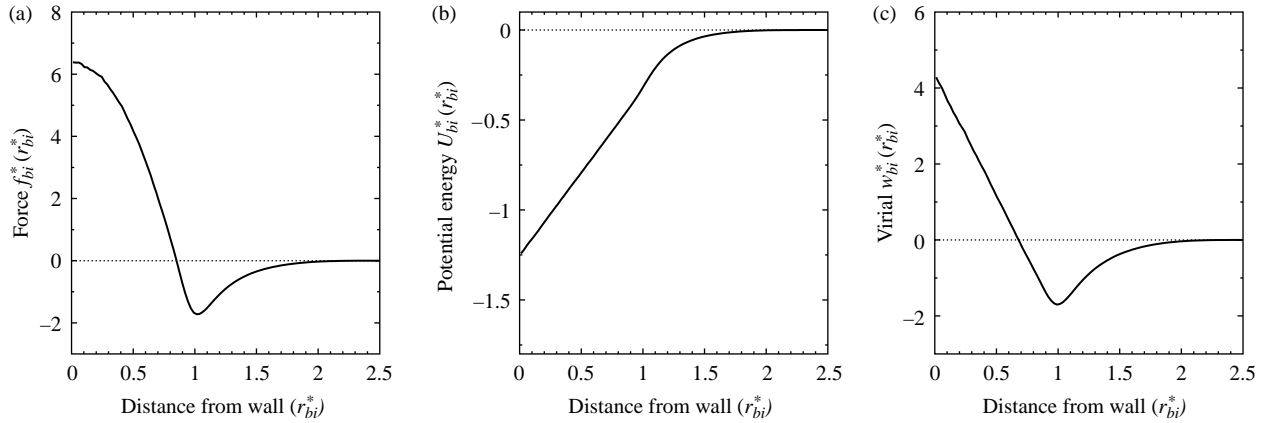


Figure 4. Distributions of mean (a) force  $f_{bi}^*(r_{bi}^*)$ , (b) potential energy  $U_{bi}^*(r_{bi}^*)$  and (c) virial  $w_{bi}^*(r_{bi}^*)$  due to missing molecules beyond the boundary,  $b$  at fluid density  $\rho^* = 0.6$ , and temperature  $T^* = 2.4$ .  $r_{bi}^*$  is the normal distance between an arbitrary molecule  $i$  and the boundary wall  $b$ .

fixed probability  $p_{\text{col}}(t) = 1 - \exp(-\Delta t_m/\tau_T)$ , where  $\tau_T = 0.1$  is the time constant. When an arbitrary molecule collides with the fictitious bath, its velocity is resampled from a Maxwell–Boltzmann distribution at the target temperature  $T_P$  and mean velocity  $\mathbf{u}_P$ . The boundary is modelled using a stochastic thermal wall [18] that reflects molecules back into the domain, with their velocities resampled from a Maxwell–Boltzmann distribution at the colliding boundary-face temperature  $T_f$  and mean velocity  $\mathbf{u}_f$ .

### 3. Testing and results

In order to test these models, we investigate flows of liquid argon molecules, which have a characteristic length scale of  $\sigma = 0.34$  nm, a characteristic energy of  $\varepsilon = 120k_b = 1.65678 \times 10^{-21}$  J, where  $k_b$  is the Boltzmann constant, and mass  $m = 6.69 \times 10^{-26}$  kg. A cut-off radius of  $r_{\text{cut}} = 2.5\sigma = 0.85$  nm is used. We present our results in reduced units: time,  $t^* = t\sqrt{\varepsilon/m\sigma^2}$ ; density,  $\rho^* = \rho\sigma^3$ ; temperature,  $T^* = T(k_b/\varepsilon)$  and velocity,  $\mathbf{u}^* = \mathbf{u}\sqrt{m/\varepsilon}$ .

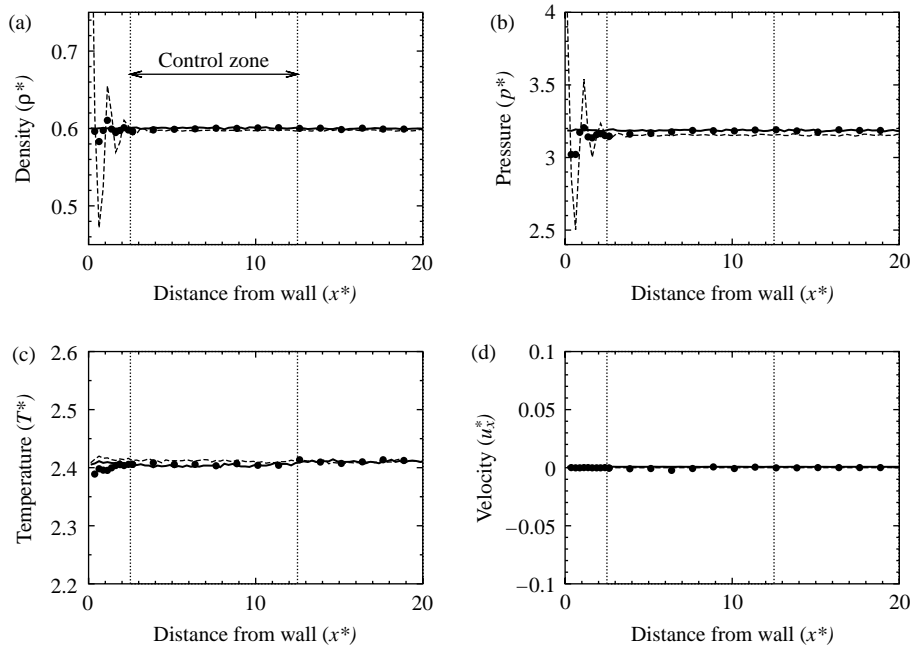


Figure 5. Results for the equilibrium case showing profiles for (a) density, (b) pressure, (c) temperature and (d) velocity; next to an MD boundary using PBCs (—), specular reflection (---) and our NPBCs (•). Large density and pressure oscillations that occur next to the specular wall case are minimised with the incorporation of the mean force and stochastic boundary models.



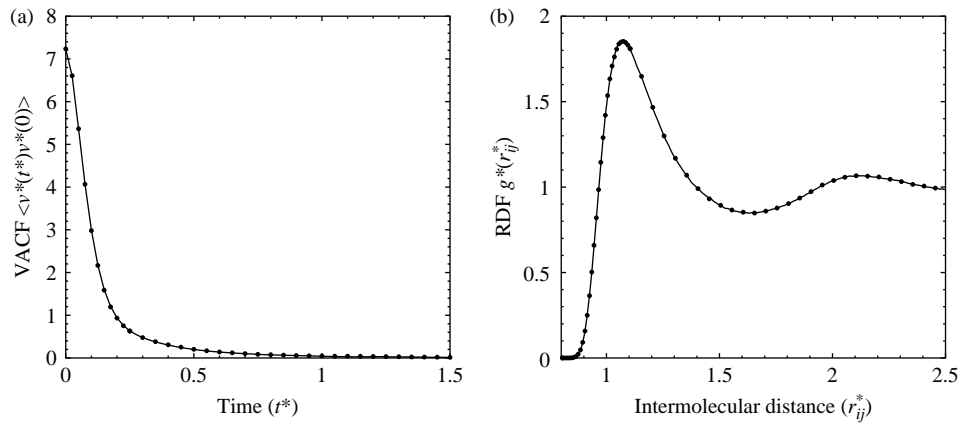


Figure 6. Comparisons between the fully periodic (—) and the non-periodic (•) boundary conditions cases for an equilibrium MD simulation: (a) the velocity autocorrelation function and (b) the radial distribution function.

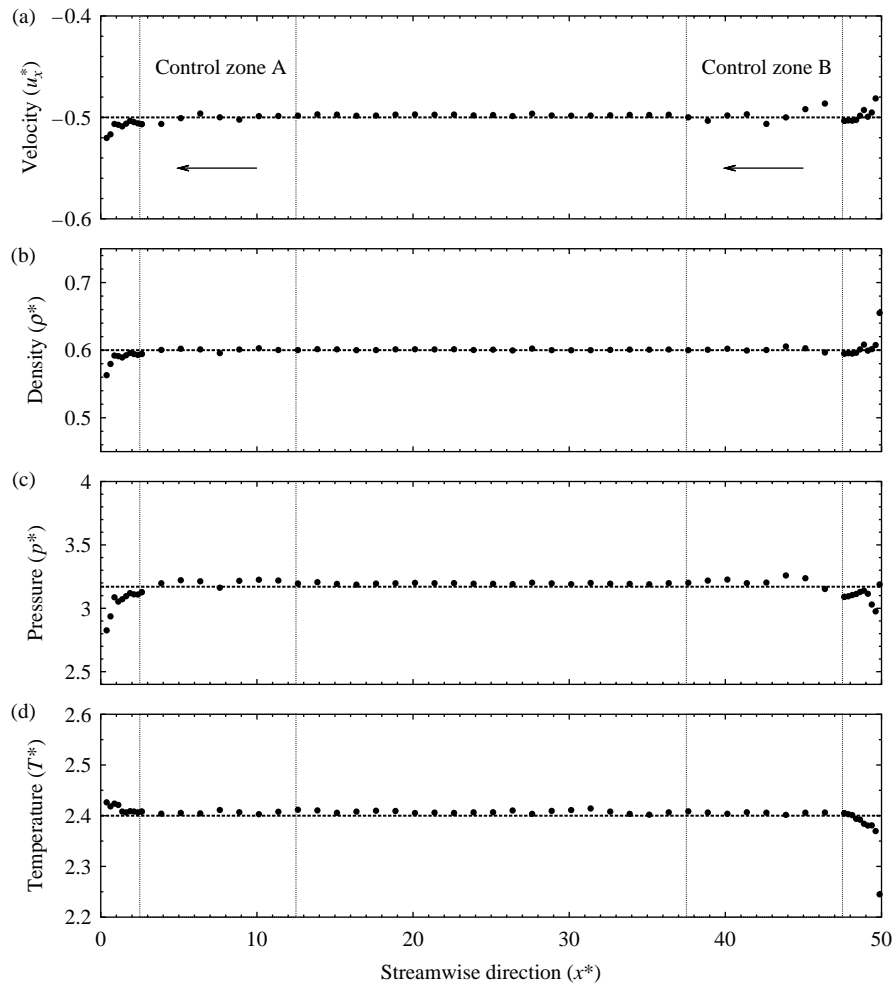


Figure 7. Results for the uniform-flow case showing profiles for (a) velocity, (b) density, (c) pressure and (d) temperature; next to an MD boundary using PBCs (---) and our NPBCs (•). The controllers and flux model defined within the NPBC description produce similar profiles as observed in the fully periodic simulations.

### 3.1 Verification studies

#### 3.1.1 Initial state for MD simulations

For the simulations that follow, and generally in most of our MD simulations, we use our controllers to set an accurate initial MD state. First, we use a pre-processing utility [19] for generating initial lattice structures of molecules in the domain mesh, based on input properties such as density, temperature, bulk velocity and lattice structure. Second, we run an MD simulation with the controllers applied in all cells of the mesh in order to converge the local fluid properties to the global targets. Finally, the controllers are switched off and the MD system is equilibrated.

#### 3.1.2 Test of NPBCs: equilibrium and uniform flow MD simulations

To simulate equilibrium and 1D uniform flow, we replace PBCs in one direction of a cubic domain by our NPBCs. We simulate a domain of dimensions  $(x, y, z) = (50\sigma, 20\sigma, 20\sigma)$ , set at a state point  $(T^* = 2.4, \rho^* = 0.6)$ , with two independent NPBCs applied at  $x = 0$  and  $x = 50\sigma$  that control this state point continually. In the equilibrium case, velocity control is set to zero,  $\mathbf{u}^* = (0, 0, 0)$ . We then compare density profiles of properties in the  $x$ -direction for three separate cases: fully periodic, semi-periodic (only reflective walls are applied) and our NPBCs. The results in Figure 5 show that conformity is achieved in the control region between the fully periodic case and the NPBCs that employ the controllers. Also, the density oscillations that occur due to the finite-size effects of the specular wall boundary are largely rectified. Furthermore, we see that the external perturbations imposed within the boundary region do not influence the liquid structure or its dynamics. Figure 6 shows a comparison of the radial distribution and velocity autocorrelation functions from a region located in the central part of the computational domain, for both fully periodic and non-periodic cases.

In the uniform-flow case, a mean velocity of  $\mathbf{u}^* = (-0.5, 0, 0)$  is applied at both boundaries of the previous equilibrium case, and steady state is allowed to be reached. We see in Figure 7 that our NPBCs effectively control the state at the boundary even if the imposed flow rate is large.

### 3.2 Control of MD in a 3D complex-geometry mixing channel

We now consider a microscale mixing channel of three-inlet one-outlet design, taken from [20] (Figure 8 inset) and reduce its scale to nanometres so that reasonable mixing time scales may be simulated (Figure 8).

The nano mixer geometry is initially drawn in Pro/ENGINEER®, a commercial computer-aided design drawing application, and exported as a STEP format (.stp)

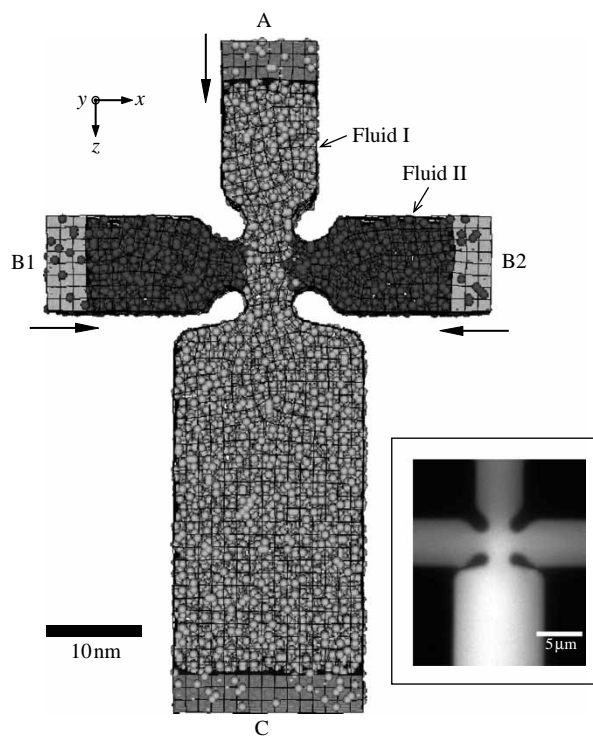


Figure 8. The nano mixer MD case shown in its initial state. NPBCs are applied at all three inlets (A, B1, B2) and the outlet (C), depicted by the shaded regions. Note: only 10,000 random molecules are shown of the  $\sim 200,000$  molecules that occupy the simulation domain. Inset: an image of the original fabricated microscale mixer taken from [20]; reproduced with permission from copyright holder.

to GAMBIT® (a meshing utility normally used for FLUENT® CFD). The geometry is meshed using hexahedral cells, and further exported in mesh format (.msh) into OpenFOAM, where it is filled with LJ molecules of two species. The two fluids are essentially isotopes of argon that have identical properties but different identification (id) number, so that mixing can be observed and measured. The NPBCs in inlet A (Figure 8) supply molecules of fluid I at a constant rate,  $\mathbf{u}_A^* = (0, 0, 0.25)$ , while inlets B1 and B2 supply molecules of fluid II at the same rate:  $\mathbf{u}_{B1}^* = (0.07, 0, 0)$ ,  $\mathbf{u}_{B2}^* = (-0.07, 0, 0)$ . These velocities were selected based on the parameter sensitivity guidelines for flow rate ratios described in [20]. Density and temperature at the inlets are set to  $\rho^* = 0.6$  and  $T^* = 2.4$ , respectively.

At the outlet, no control is applied since the target values of density, velocity and temperature are not known a priori; the complex constrictions in the central part of the domain and the boundary walls introduce compressibility effects. So, instead we apply a new mass flux boundary model that removes molecules of fluids I and II at a rate computed from the error between the mean density within the global system and a target density,  $\rho^* = 0.55$ .

The channel wall boundary is 3D – no PBCs are applied at all within this simulation. We model the outer wall using an implicit isothermal stochastically reflective boundary model that resamples velocities of colliding molecules from a Maxwell–Boltzmann distribution at a temperature of  $T^* = 2.4$ , and zero mean velocity, and imposes the constraint  $(\mathbf{v}_i \cdot \hat{\mathbf{n}}_f) \leq 0$ . In addition, potential energy and external forces are also applied to molecules near the boundary to take into effect the missing liquid–wall molecular interactions. The applied mean force and potential energy distributions (Figure 9) are sampled from a small equilibrium MD simulation next to a face-centred cubic MD wall lattice at a temperature of  $T^* = 2.4$  and density  $\rho^* = 0.8$ .

The mixing-channel case is decomposed and solved on eight processors for a duration of  $t^* = 2500$ . Results of steady-state mixing are shown in Figures 10 and 11. The partial densities of both fluids are sampled in a region close to the outlet (Figure 11) and show that mixing occurs throughout the width of the outlet channel. Mixing is more complete in the central part of the channel; at the sides, we see discrepancies of  $\Delta\rho \approx 0.03$  relative to the mean (target) partial density.

### 3.3 Hybrid continuum MD simulations with a complex coupling region

We can further demonstrate the capabilities of our controllers, and the NPBCs in which they operate, by applying them in a complex coupling region of a hybrid MD-CFD simulation. The case we choose is an isothermal shear flow (Couette type) that has a complex protrusion in the stationary wall (see Figure 12). The wall and adjacent layer of liquid are modelled by MD, while the rest of the domain and moving wall are simulated by CFD. The sonicLiquidFlow solver already in OpenFOAM (the open

source CFD toolbox; available online: [www.openfoam.org](http://www.openfoam.org)) is used for the continuum subdomain; it models liquid flow using the compressible Navier–Stokes equations. A moving wall velocity of  $\mathbf{u}_w^* = (0.5, 0, 0)$ , with a no-slip boundary condition, is applied to the top boundary of the domain, and cyclic boundary conditions are applied in the other two directions. At the molecular-continuum interface, a 3D overlap region is present so that coupling between CFD and MD formulations can occur at regular time intervals of the simulation. The global mesh (see Figure 12(a)) is segmented into two separate MD and CFD sub-meshes, such that the coupling region is common to both sub-meshes (see Figure 12(b)). This sub-meshing technique ensures that the coupling region on both meshes are identical.

The coupling region is made up of two sub-regions,  $M \rightarrow C$  (molecular-to-continuum) and  $C \rightarrow M$  (continuum-to-molecular), in which velocity boundary conditions are passed between CFD and MD formulations. In the  $M \rightarrow C$  sub-region (on the MD mesh), velocity fields are averaged from molecular data and are passed as Dirichlet boundary conditions to the CFD mesh. Similarly, in the  $C \rightarrow M$  sub-region (on the CFD mesh) the convective velocity fields are passed as target fields to the MD mesh. These fields are used by our NPBC description, that is, via the controllers. Although no density or temperature coupling is performed, controllers are still applied in the  $C \rightarrow M$  region to fix the boundary state to  $\rho^* = 0.6$  and  $T^* = 1.8$ . Accurate coupling is achieved by setting the viscosity ( $\eta = 0.899(\sqrt{\epsilon m}/\sigma^2)$ ) and pressure ( $p = 1.98(\sigma^3/\epsilon)$ ) of the CFD formulation to match those sampled from an equilibrium MD simulation at the desired state point. The viscosity is determined using the Green–Kubo relationship [2], by averaging the shear-stress autocorrelation function over  $2 \times 10^6$  MD time steps.

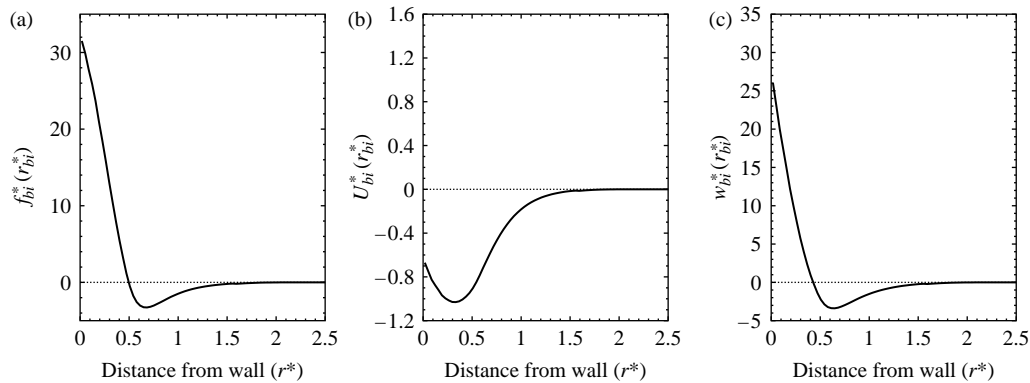


Figure 9. Distributions of mean (a) force, (b) potential energy and (c) virial sampled next to face-centred cubic wall in an equilibrium MD simulation at  $\rho^* = 0.8$  and  $T^* = 2.4$ . Wall molecules are tethered in space and a harmonic spring potential is applied between a tether point and its corresponding wall molecule,  $U_h = 1/2K_s(\mathbf{r}_i - \mathbf{r}_i^{\text{teth}})^2$ , where  $K_s = 150(\epsilon/\sigma^2)$  is the spring constant.

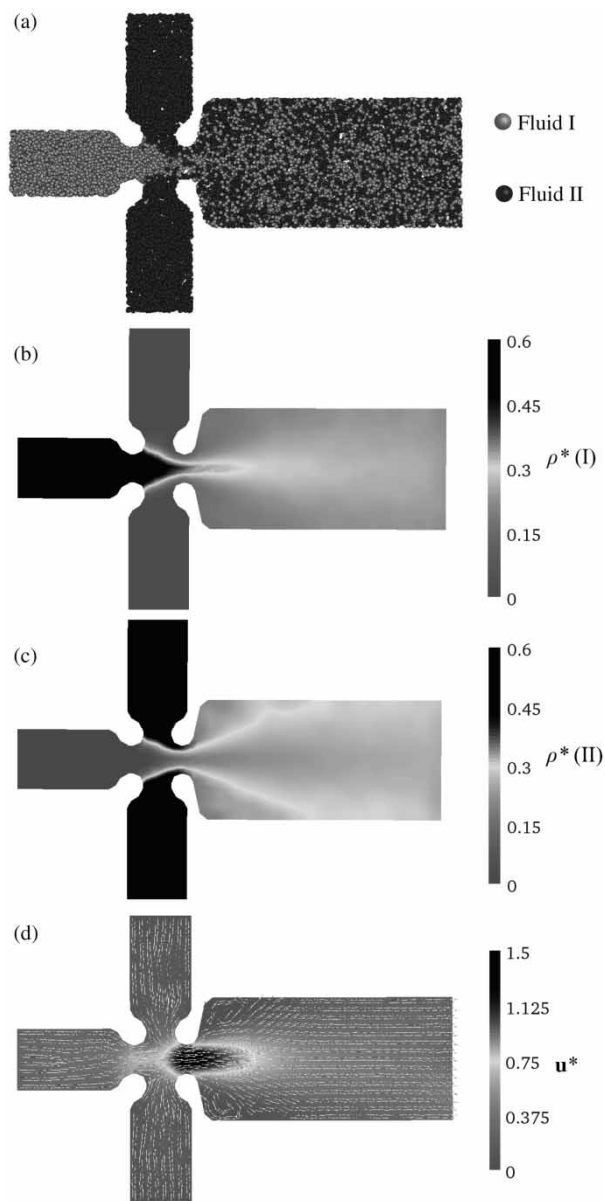


Figure 10. Mid-channel cross-sectional fields showing (a) the Lagrangian field of molecules, (b) partial density for fluid I, (c) partial density for fluid II and (d) total velocity.

A coupling time framework is used that advances the MD and CFD in a sequential manner by a common coupling time interval  $\Delta t_{\text{coupling}} = \tau_C \Delta t_c = \tau_M \Delta t_m = 20(\sqrt{m\sigma^2/\varepsilon})$ :

- (1) Advance continuum solution  $t \rightarrow t + \Delta t_{\text{coupling}}$  by  $\tau_C = 200$  time steps. MD waits.
- (2) Apply  $C \rightarrow M$  BCs.
- (3) Advance MD  $t \rightarrow t + \Delta t_{\text{coupling}}$  by  $\tau_M = 4000$  time steps. CFD waits.
- (4) Apply  $M \rightarrow C$  BCs.
- (5) Repeat the dual time-marching scheme until the end of the simulation.

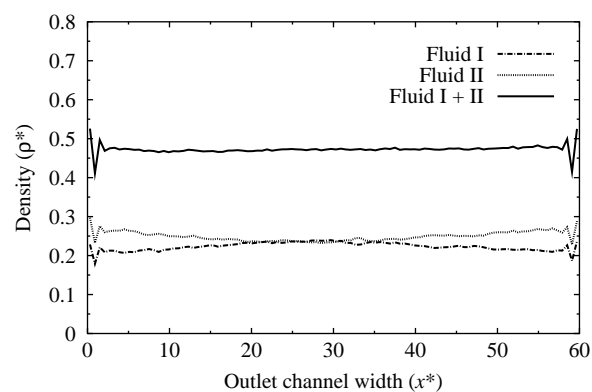


Figure 11. The linear partial-density distribution for both fluids taken from a sampling region at the outlet of the mixer. The figure shows complete mixing in the middle part of the channel.

The hybrid and full MD simulations are each solved in parallel on two processors, so that timings can be compared. Steady-state results are shown in Figures 13 and 14, where general agreement of the velocity field is observed between the full MD and hybrid MD-CFD simulations. However, the hybrid simulation is approximately two times faster over the full MD simulation. Speed-up may also be estimated from the term  $N_{\text{mol}}^{\text{FM}}/N_{\text{mol}}^{\text{H}}$ ,

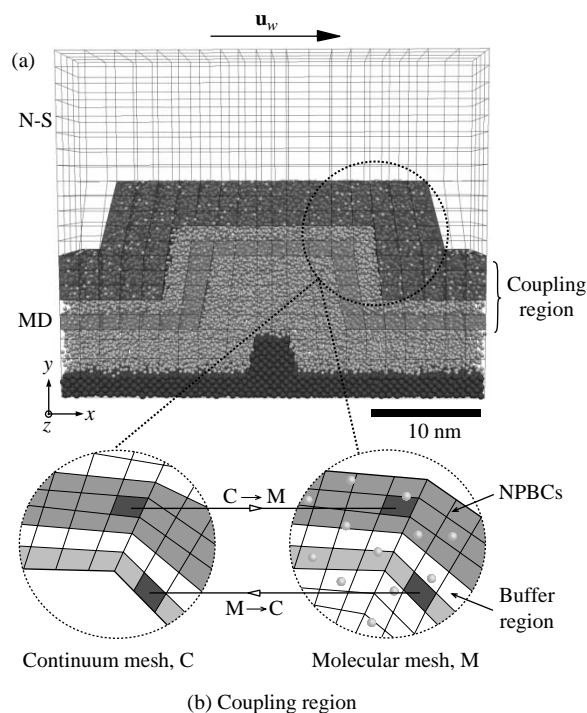


Figure 12. (a) Hybrid simulation domain of a shear flow with a complex fixed-wall topology. (b) Schematic of the coupling region between continuum and molecular subdomain meshes.  $M \rightarrow C$  and  $C \rightarrow M$  velocity boundary conditions are transferred between pairs of coupled cells, as shown in the two highlighted regions.  $C \rightarrow M$  coupling is achieved via the controllers.

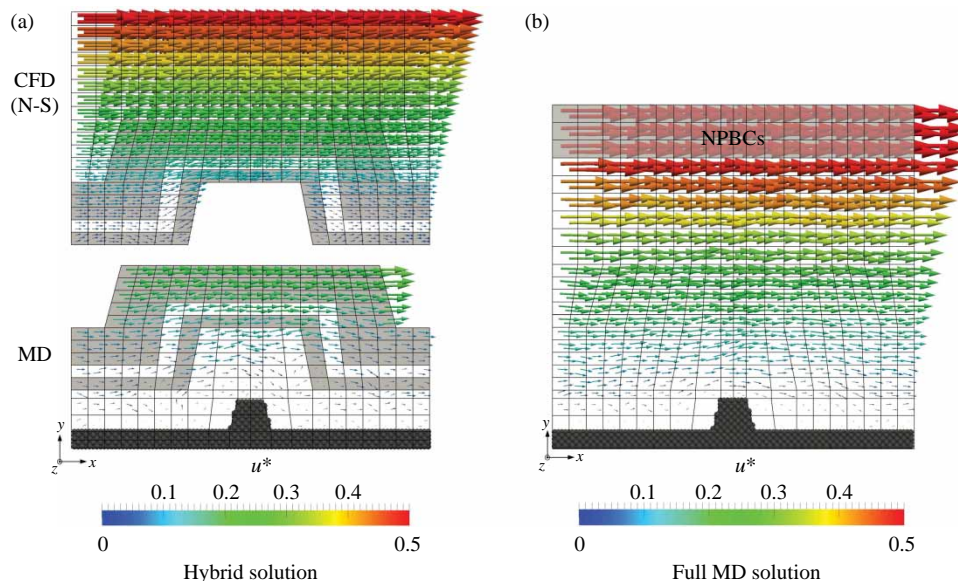


Figure 13. Comparisons of the velocity vector field between (a) hybrid (top – Navier–Stokes solution; bottom – MD solution) and (b) full MD simulations of the same case. Fluctuations of velocity measurements in the M → C coupling sub-region on the CFD mesh are caused by the small coupling interval,  $\Delta t_{av}^* = \Delta t_{coupling}^* = 20$ , which is used to average velocity from the MD cells. A longer averaging period  $\Delta t_{av}^* = 1000$  is used for presenting the MD velocity solutions.

where  $N_{mol}^H = 34,394$  and  $N_{mol}^{FM} = 68,944$  are the average number molecules in the hybrid MD subdomain and the full MD domain, respectively.

#### 4. Conclusions

The ability to control MD simulations of nano-scale fluid flows is an essential requirement for the rapid research and development of many engineering applications, such as lab-on-a-chip components, desalination membranes and nanoelectromechanical systems. In this paper, we have proposed a set of controllers that are able to converge continuum-derived properties in an arbitrary geometry region of an MD

simulation. We have shown that the perturbations introduced by our control algorithm do not influence the dynamic or liquid structure when sampled close to the controlling region. Furthermore, control, which is based on local cells of the mesh, has been shown to be a more flexible and generalised technique for applying non-periodic MD boundary conditions in 3D complex geometries (such as a three-inlet one-outlet mixing channel), and also hybrid MD-CFD simulations that contain complex coupling regions (such as the Couette-type flow case). Each full MD and hybrid MD-CFD case have been decomposed on multiple CPUs and executed in parallel.

New continuum-to-MD controller models based on this control system may easily be implemented, since they only require the explicit definition of each of the control functions. The underlying generic code of the controllers is common to all and has only been implemented once for the simulations in this paper.

#### Acknowledgements

The authors would like to thank Chris Greenshields and William Nicholls of the University of Strathclyde for useful discussions. This work is funded in the UK by the Engineering and Physical Sciences Research Council under grant EP/F002467/1 and by the James Weir Foundation.

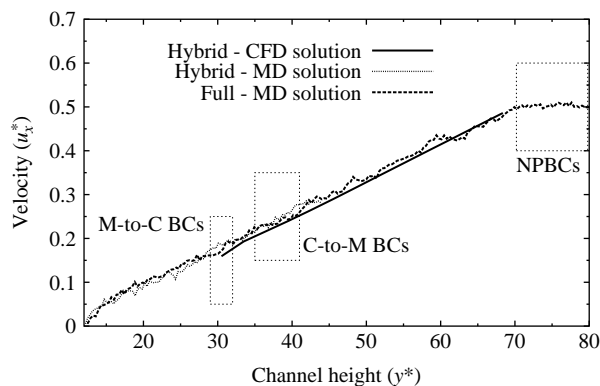


Figure 14. Comparisons between the hybrid and full MD simulations of the y-direction velocity profiles ( $u_x^*$ ) taken from a central part of the computational domain.

#### References

- [1] D.C. Rapaport, *The Art of Molecular Dynamics Simulation*, 2nd ed., Cambridge University Press, Cambridge, 2004.
- [2] M.P. Allen and D.J. Tildesley, *Computer Simulation of Liquids*, Oxford University Press, Oxford, 1987.

- [3] S.T. O'Connell and P.A. Thompson, *Molecular dynamics-continuum hybrid computations: A tool for studying complex fluid flows*, Phys. Rev. E 52 (1995), pp. R5792–R5795.
- [4] X.B. Nie, S.Y. Chen, W. E, and M.O. Robbins, *A continuum and molecular dynamics hybrid method for micro- and nano-fluid flow*, J. Fluid Mech. 500 (2004), pp. 55–64.
- [5] E.G. Flekkøy, G. Wagner, and J. Feder, *Hybrid model for combined particle and continuum dynamics*, Europhys. Lett. 52 (2000), pp. 271–276.
- [6] R. Delgado-Buscalioni and P.V. Coveney, *Continuum-particle hybrid coupling for mass, momentum, and energy transfers in unsteady fluid flow*, Phys. Rev. E 67 (2003), pp. 1–13, 046704.
- [7] T. Werder, J.H. Walther, and P. Koumoutsakos, *Hybrid atomistic-continuum method for the simulation of dense fluid flows*, J. Comput. Phys. 205 (2005), pp. 373–390.
- [8] M. Sun and C. Ebner, *Molecular-dynamics simulation of compressible fluid flow in two-dimensional channels*, Phys. Rev. A 46 (1992), pp. 4813–4818.
- [9] J. Koplik, J.R. Banavar, and J.F. Willemsen, *Molecular dynamics of Poiseuille flow and moving contact lines*, Phys. Rev. Lett. 60 (1988), pp. 1282–1285.
- [10] J. Li, D. Liao, and S. Yip, *Imposing field boundary conditions in MD simulation of fluids: Optimal particle controller and buffer zone feedback*, Mat. Res. Soc. Symp. Proc. 538 (1998), pp. 473–478.
- [11] E.M. Kotsalis, J.H. Walther, and P. Koumoutsakos, *Control of density fluctuations in atomistic-continuum simulations of dense liquids*, Phys. Rev. E 76 (2007), 016709.
- [12] E.M. Kotsalis, J.H. Walther, E. Kaxiras, and P. Koumoutsakos, *A control algorithm for multiscale flow simulations of water*, Phys. Rev. E 79 (2009), 045701.
- [13] G.B. Macpherson, N. Nordin, and H.G. Weller, *Particle tracking in unstructured, arbitrary polyhedral meshes for use in CFD and molecular dynamics*, Commun. Numer. Methods Eng. 25 (2009), pp. 263–273.
- [14] G.B. Macpherson and J.M. Reese, *Molecular dynamics in arbitrary geometries: Parallel evaluation of pair forces*, Mol. Simulat. 34 (2008), pp. 97–115.
- [15] N. Hadjiconstantinou, A. Garcia, M. Bazant, and G. He, *Statistical error in particle simulations of hydrodynamic phenomena*, J. Comput. Phys. 187 (2003), pp. 274–297.
- [16] R. Delgado-Buscalioni and P.V. Coveney, *USHER: An algorithm for particle insertion in dense fluids*, J. Chem. Phys. 119 (2003), pp. 978–987.
- [17] N. Hadjiconstantinou and A. Patera, *Heterogeneous atomistic-continuum methods for dense fluid systems*, Int. J. Modern Phys. C 8 (1997), pp. 967–976.
- [18] G. Ciccotti and A. Tenenbaum, *Canonical ensemble and non-equilibrium states by molecular dynamics*, J. Stat. Phys. 23 (1980), pp. 767–772.
- [19] G.B. Macpherson, M.K. Borg, and J.M. Reese, *Generation of initial molecular dynamics configurations in arbitrary geometries and in parallel*, Mol. Simulat. 33 (2007), pp. 1199–1212.
- [20] D.E. Hertzog, B. Ivorra, B. Mohammadi, O. Bakajin, and J.G. Santiago, *Optimization of a microfluidic mixer for studying protein folding kinetics*, J. Anal. Chem. 78 (2006), pp. 4299–4306.