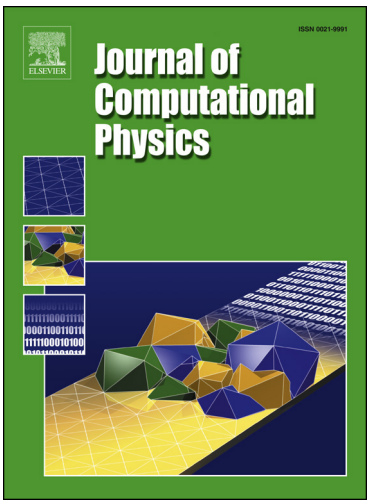
Journal Pre-proof



Objective molecular dynamics for atomistic simulation of macroscopic fluid motion

Gunjan Pahlani, Thomas E. Schwartzentruber and Richard D. James

PII: S0021-9991(23)00033-5
DOI: https://doi.org/10.1016/j.jcp.2023.111938
Reference: YJCPH 111938

To appear in: *Journal of Computational Physics*

Received date: 19 June 2022
Revised date: 18 December 2022
Accepted date: 11 January 2023

Please cite this article as: G. Pahlani, T.E. Schwartzentruber and R.D. James, Objective molecular dynamics for atomistic simulation of macroscopic fluid motion, *Journal of Computational Physics*, 111938, doi: https://doi.org/10.1016/j.jcp.2023.111938.

Highlights

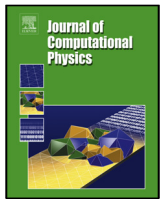
- OMD for the time-dependent translation group is developed as a computational tool.
- OMD provides a rigorous framework to perform non-equilibrium molecular dynamics.
- The widely used velocity-Verlet algorithm is consistent with OMD.
- In the regime of higher rates the Navier-Stokes-Fourier theory is no longer accurate.
- OMD can also deal with spontaneous phase transition and boundary driven flow.

Journal of Computational Physics (2023)

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp



Objective molecular dynamics for atomistic simulation of macroscopic fluid motion

Gunjan Pahlani[a], Thomas E. Schwartzentruber[a], Richard D. James[a,*]

[a]*Aerospace Engineering and Mechanics, University of Minnesota, Minneapolis, 55415, USA*

ARTICLE INFO

Article history:
Received XX Xxx XXXX
Received in final form XX Xxx XXXX
Accepted XX Xxx XXXX
Available online XX Xxx XXXX

Communicated by G. Pahlani

Keywords: Objective Molecular Dynamics, Nano-channel flow, Slip-flow, Non-equilibrium Molecular Dynamics, Far-from-equilibrium flows, Phase transformation, Gas Dynamics

ABSTRACT

The method of Objective Molecular Dynamics (OMD) provides exact solutions of the equations of molecular dynamics for atoms filling all of space in non equilibrium situations. The method can be used to simulate families of incompressible, compressible and unsteady flows, also with time-dependent vorticity. In this paper we develop OMD as an efficient computational tool and introduce some of its applications. We apply the method to the evolution of compressible heat-conducting monoatomic gas under general incompressible flow, and to flow in a nanochannel having realistic atomistic solid boundaries. The macroscopic flows of OMD are exact solutions of the Navier-Stokes equation with Newtonian and Fourier models, and we make comparisons with these solutions. We also report the simulation of homoenergetic dilatational flow exhibiting condensation using OMD techniques.

1. Introduction

Objective Molecular Dynamics (OMD) is an exact method of molecular dynamics (MD) simulation for non-equilibrium flows. A finite number of atoms are actually simulated, and the motions of the remaining (typically infinitely many) nonsimulated atoms are given by explicit formulas based on the positions of the simulated atoms. There are no restrictions on the number of simulated atoms, and each simulated atom can be given arbitrary initial conditions of position and velocity. The formulas that relate the positions of the nonsimulated atoms to those of the simulated atoms have an explicit dependence on time, which gives a macroscopic flow. The underlying theorem that justifies this method, i.e., that each nonsimulated atom also satisfies exactly the equations of molecular dynamics for its forces, relies only on the structure of the equations of molecular dynamics and the basic invariance of (non relativistic) quantum mechanics. The invariance used is the frame-indifference and permutation invariance of the potential energy.

*Corresponding author: Tel.: +0-000-000-0000; fax: +0-000-000-0000;
e-mail: james@aem.umn.edu (Richard D. James)

If the simulated atoms are chosen near each other initially, then typically they diffuse quickly into the sea of nonsimulated atoms as the simulation proceeds. So, generally, each simulated atom is surrounded by both simulated and nonsimulated atoms within its cut-off. Since the simulations exhibit the complex dynamics ("chaos") that is typical of MD simulation, the main challenge for the method is to efficiently find all simulated and nonsimulated atoms within the cut-off of each simulated atoms. In this paper we develop efficient methods for this purpose, and apply the method to some flows of interest for the comparison of molecular dynamics to fluid dynamics.

The explicit formulas that relate the positions of simulated to nonsimulated atoms are formulated using isometry groups, i.e., groups of orthogonal transformations and translations. Many molecular structures present in nature are connected to this invariance [18, 16] and can be constructed using these isometries, and hence are amenable to OMD methods. But here we consider the simplest case of the time-dependent translation group. In that case we can obtain OMD solutions representing a family of unsteady macroscopic flows which are associated to this group [10]. These macroscopic flows necessaily have Eulerian velocity fields $\mathbf{v}(\mathbf{x}, t)$ of the form

$$\mathbf{v}(\mathbf{x}, t) = \mathbf{A}(\mathbf{I} + t\mathbf{A})^{-1}\mathbf{x}. \tag{1}$$

Here, $\mathbf{A}$ is an assignable 3×3 matrix that can be interpreted as Lagrangian velocity gradient, and t is the time. $\mathbf{A}$ also enters the formulas that give the positions of the nonsimulated atoms in terms of the simulated atoms. By choosing $\mathbf{A}$ in different ways, the velocity field (1) includes many examples of steady and unsteady compressible and incompressible flows, including cases with time-dependent vorticity and cases with strong singularities. The latter occurs when $\mathbf{A}$ is chosen such that $\det(\mathbf{I} + t\mathbf{A}) \to 0$ in finite time.

The comparison with fluid dynamics is much aided by the fact that (1) is an exact solution of the Navier-Stokes equations (compressible or incompressible) as well as every accepted continuum model of fluid flow, for every choice of the 3×3 matrix $\mathbf{A}$. These statements also apply to non-Newtonian and other complex fluids. Flows of the form (1) are therefore promising candidates as an approach to experimental fluid mechanics for which we have an exact atomistic analogue, and "OMD rheometers" have been designed based on some of these flows [11].

OMD can be rephrased as a (time-dependent) invariant manifold of the MD equations. This has the usual meaning: the manifold is a surface in the space of positions, momenta and time for all the atoms. If you give initial conditions on this manifold at $t = 0$, you remain on this manifold for $t > 0$. The manifold is given by an explicit analytical formula. It is interesting to note that in addition to continuum mechanics, the kinetic theory inherits the invariant manifold given by OMD exactly. The ansatz on molecular density function $f(t, \mathbf{x}, \mathbf{v}) = g(t, \mathbf{v} - \mathbf{A}(\mathbf{I} + t\mathbf{A})^{-1}\mathbf{x})$ corresponds to OMD and reduces the Maxwell-Boltzmann equation to an equation for $g(t, \mathbf{w})$ [3]. In important special cases this equation is shown to be well-posed [23], and associated asymptotic formulas for the H-function (minus the entropy) give interesting insight into far-from-equilibrium statistical mechanics [23, 22, 24]. These flows can play a crucial role in the future for validation of multiscale methods due to its connection with the theories at different scales. Thus, the method of OMD is unusual in being an exact method of molecular dynamics without boundaries that includes far-from-equilibrium flows and general (Born-Oppenheimer) atomic forces, having precise connections to both continuum mechanics and the Boltzmann equation (and its associated numerical methods, such as DSMC [6]).

The efficient numerical implementation of OMD presents difficulties usually not present in other time-dependent numerical methods, and there does not currently exist an efficient numerical strategy for implementing OMD. The purpose of this paper is a) to provide a useful implementation of OMD for the time-dependent translation group, and b) to explore some of the flow phenomena that are possible with the method. As explained below, the main issue one has to confront in implementation is that the simulated atoms quickly diffuse chaotically into the sea of nonsimulated atoms, and this requires that one develops an efficient method of finding neighbors.

There are various alternate approaches to OMD. First, we note that MD with periodic boundary conditions is a special case of OMD with $\mathbf{A} = 0$, as is Lees-Edwards boundary conditions where $\mathbf{A} = \mathbf{a} \otimes \mathbf{n}$, $\mathbf{a} \cdot \mathbf{n} = 0$ [30]. However, the typical implementation of the Lees-Edwards method using moving boxes does not generalize to OMD. A popular alternative method is NEMD where a fictitious external field is imposed which guarantees that desired velocity field is maintained [20]. This has the advantage of being able to treat some flows not of the macroscopic form (1). However, in NEMD the perturbation is applied by modifying the deterministic equations, termed the SLLOD equations. An

extensive review of the method is given in [42]. In OMD, the MD equations are not modified: the force on each atom is only produced by the other (simulated and non-simulated) atoms. One can say that the forces that "drive the motion" are applied at infinity, but these forces are not explicitly introduced but are inherent in the method. There is also an extensive literature on methods of molecular dynamics designed to simulate elongational flows [3, 17, 41]. These methods are not exact but can be useful to generate approximations in the situations of interest. The widely used Parinello-Rahman method and its generalizations [36, 32] is implemented using some ideas similar to those used here but is otherwise rather different. The aim of isobaric methods like Parinello-Rahman is to find an equilibrium state at constant assigned stress, whereas OMD operates in the non equilibrium regime, and is most interesting in the far-from-equilibrium regime.

Molecular level flow fields can also be induced by modeling the actual solid walls that confine the fluid. This induces generation of spatial density inhomogeneities near the walls which calls for the bigger system for simulation, unless one is specifically interested in nano-confined flows and these inhomogeneities. Like periodic boundary conditions, this makes homogeneous methods like OMD a more appropriate choice for computing bulk properties where boundary effects can be eliminated. On the other hand, OMD for fluids is compatible with the introduction of certain types of solid boundaries, either modelled as atomistic solids or with certain confining potentials. In this paper we do not develop efficient methods for the case of such walls, but we give some examples of simulations of this type.

To achieve efficiency in OMD simulation, we make use of ideas from the theory of lattice invariant deformations of crystallography and adapt concepts from neighbor list generation, such as minimum image convention and the cell list method used in traditional molecular dynamics codes. We examine its validity by carrying out the simulations on Lennard-Jones (LJ) systems for incompressible flows of a monoatomic gas, and we compare atomistic simulations with the continuum Navier-Stokes-Fourier (NSF) model. OMD can also work well for many other complex phenomena in fluids such as phase transition, chemical dissociation [35], electronic transition, etc. To show its reach, we report in this work, the homogeneous phase transition in a super-critical Lennard-Jones Argon driven by high rate expansion. It is also appropriate for modeling friction and complex motion and interaction of dislocations in crystalline solids, which we show in forthcoming work.

We solve the equations of OMD using the Velocity-Verlet algorithm. In addition to its well-known desirable properties such as being symplectic and time-reversible, we prove in Section 4 that the invariant manifold of OMD is inherited exactly at the discrete level with the Velocity-Verlet algorithm.

The paper is organized as follows: Section 2 describes the basics of Objective Molecular Dynamics. Section 3 gives the details of the numerical method. Section 4 shows the satisfaction of the basic theorem of OMD at the discrete level by the velocity Verlet algorithm. Sections 5, 6 and 7 show validation and some applications of OMD. Finally, the conclusions are contained in Section 8.

2. Objective Molecular Dynamics

Objective molecular dynamics makes use of discrete groups of isometries. These are groups $\mathbf{G} = \{g_1, g_2, \dots\}$ of orthogonal transformations and translations with elements typically written in the notation

$$g_j = (\mathbf{Q}_j|\mathbf{c}_j) \in \mathbf{G}, \quad j = 1, ..., N, k = 1, ..., M, \tag{2}$$

where $\mathbf{Q}_j \in \mathrm{O}(3)$ and $\mathbf{c}_j \in \mathbb{R}^3$. The multiplication rule for isometries is

$$(\mathbf{Q}_j|\mathbf{c}_j)(\mathbf{Q}_k|\mathbf{c}_k) = (\mathbf{Q}_j\mathbf{Q}_k|\mathbf{c}_j + \mathbf{Q}_j\mathbf{c}_k), \tag{3}$$

the inverse of $(\mathbf{Q}|\mathbf{c})$ is $(\mathbf{Q}^T| - \mathbf{Q}^T\mathbf{c})$ and the identity is $(\mathbf{I}|0)$. For OMD, the translational part $\mathbf{c}_k$ is allowed to depend on time but this dependence must be affine [10]: $\mathbf{c}_k = \mathbf{a}_k t + \mathbf{b}_k$.

We consider any number of atoms labeled $1, ..., M$ with positive masses $m_1, ..., m_M$. These are called *simulated atoms*. Let $\mathbf{y}_k(t), t > 0, k = 1, \dots, M$, be the motions of these *simulated atoms*. Then there exist nonsimulated atoms whose motions are given in terms of the simulated atoms by

$$\mathbf{y}_{j,k}(t) = g_j(\mathbf{y}_k(t)) \tag{4}$$

where the elements $g_j(t) \in \mathbf{G}$ is a discrete group of isometries. The allowed time dependence of the g_j implies that

$$\frac{d^2\mathbf{y}_{j,k}(t)}{dt^2} = \frac{d^2}{dt^2} g_j(\mathbf{y}_k(t)) = \mathbf{Q}_j \frac{d^2\mathbf{y}_k(t)}{dt^2}, \quad g_j = (\mathbf{Q}_j|\mathbf{c}_j) \in \mathbf{G}, j = 1, ..., N, k = 1, ..., M \tag{5}$$

The fact that the non-simulated atoms of OMD satisfy the equations of MD rests on the the standard conditions of invariance satisfied by the atomic forces. That is, with the force on atom i, k denoted by the suggestive notation $-\partial\varphi/\partial\mathbf{y}_{i,k} : \mathbb{R}^{3MN} \to \mathbb{R}^3$, we assume that this function is smooth and frame-indifferent, i.e., for all $\mathbf{Q} \in \mathrm{O}(3)$ and $\mathbf{c} \in \mathbb{R}^3$,

$$\begin{aligned} &\mathbf{Q}\frac{\partial\varphi}{\partial\mathbf{y}_{i,k}}(\dots, \mathbf{y}_{i_1,1}, \dots \mathbf{y}_{i_1,M}, \dots, \mathbf{y}_{i_2,1}, \dots \mathbf{y}_{i_2,M}, \dots) \\ &\quad = \frac{\partial\varphi}{\partial\mathbf{y}_{i,k}}(\dots, \mathbf{Q}\mathbf{y}_{i_1,1} + \mathbf{c}, \dots \mathbf{Q}\mathbf{y}_{i_1,M} + \mathbf{c}, \dots, \mathbf{Q}\mathbf{y}_{i_2,1} + \mathbf{c}, \dots \mathbf{Q}\mathbf{y}_{i_2,M} + \mathbf{c}, \dots), \end{aligned} \tag{6}$$

and also that it is permutation invariant,

$$\begin{aligned} &\frac{\partial\varphi}{\partial\mathbf{y}_{\Pi(i,k)}}(\dots, \mathbf{y}_{i_1,1}, \dots \mathbf{y}_{i_1,M}, \dots, \mathbf{y}_{i_2,1}, \dots \mathbf{y}_{i_2,M}, \dots) \\ &\quad = \frac{\partial\varphi}{\partial\mathbf{y}_{i,k}}(\dots, \mathbf{y}_{\Pi(i_1,1)}, \dots \mathbf{y}_{\Pi(i_1,M)}, \dots, \mathbf{y}_{\Pi(i_2,1)}, \dots \mathbf{y}_{\Pi(i_2,M)}, \dots), \end{aligned} \tag{7}$$

where Π is any permutation that preserves species. Here, preservation of species means that if $(i, k) = \Pi(j, \ell)$ then the species (i.e., atomic mass and number) of atom i, k is the same as the species of atom j, ℓ.

The basic theorem of OMD then says that if $\mathbf{y}_k(t)$ are subjected to the equations of molecular dynamics, i.e.,

$$\begin{aligned} m_k\ddot{\mathbf{y}}_k(t) &= -\frac{\partial\varphi}{\partial\mathbf{y}_{1,k}}(\dots, \mathbf{y}_{i,1}(t), \dots, \mathbf{y}_{i,M}(t), \mathbf{y}_{i+1,1}(t), \dots, \mathbf{y}_{i+1,M}(t), \dots) \\ &= -\frac{\partial\varphi}{\partial\mathbf{y}_{1,k}}(\dots, g_i(\mathbf{y}_{1,1}(t), t), \dots, g_i(\mathbf{y}_{1,M}(t), t), g_{i+1}(\mathbf{y}_{1,1}(t), t), \dots, g_{i+1}(\mathbf{y}_{1,M}(t), t), \dots), \\ &\quad \mathbf{y}_k(0) = \mathbf{y}_k^0, \;\; \dot{\mathbf{y}}_k(0) = \mathbf{v}_k^0, \;\; k = 1, \dots, M, \end{aligned} \tag{8}$$

then the equations of molecular dynamics are exactly satisfied by non simulated atoms $\mathbf{y}_{j,k}(t)$ in spite of the fact that their motion is coming from an explicit formula (4) :

$$m_k\ddot{\mathbf{y}}_{j,k}(t) = -\frac{\partial\varphi}{\partial\mathbf{y}_{j,k}}(\dots, \mathbf{y}_{i,1}(t), \dots, \mathbf{y}_{i,M}(t), \mathbf{y}_{i+1,1}(t), \dots, \mathbf{y}_{i+1,M}(t), \dots)$$

Notice that because of the substitution of the group elements acting on simulated atoms on the right hand side, (8) is a system of ordinary differential equations in standard form for the simulated atoms.

In the case of the translation group given by $G_T = \{(\mathbf{I}|\nu^1\hat{\mathbf{e}}_1 + \nu^2\hat{\mathbf{e}}_2 + \nu^3\hat{\mathbf{e}}_3) : \nu = (\nu^1, \nu^2, \nu^3) \in \mathbb{Z}^3\}$, the basic method is the following. Give a 3×3 matrix $\mathbf{A}$ and three linearly independent vectors $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$, $(\mathbf{e}_1 \times \mathbf{e}_2) \cdot \mathbf{e}_3 > 0$, choose $\hat{\mathbf{e}}_i = (\mathbf{I} + t\mathbf{A})\mathbf{e}_i$, $i = 1, 2, 3$. This choice preserves the group properties and the affine time-dependence. The motions of the nonsimulated atoms are given in terms of the simulated atoms by

$$\mathbf{y}_{\nu,k}(t) = g_\nu(\mathbf{y}_k(t)), \quad \mathbf{y}_{\nu,k}(t) = \mathbf{y}_k(t) + \sum_{i=1}^{3} \nu^i(\mathbf{I} + t\mathbf{A})\mathbf{e}_i, \quad g_\nu \in G_T \tag{9}$$

where, $\mathbf{y}_k(t) = \mathbf{y}_{(0,0,0),k}(t), t > 0, k = 1, \dots, M$, are the motions of the *M simulated atoms*. This simulation fills all of space but in certain cases it can be confined by rigid boundaries [11]. The matrix $\mathbf{A}$ is the same as that appearing in the formula (1) for the macroscopic motion.

3. Numerical Method

Depending on the choice of $\mathbf{A}$, it can happen that at some $t = T > 0$, $(\mathbf{I} + t\mathbf{A})$ ceases to be invertible. Then we stop the simulation before time T. So, below we assume that $0 \le t < T$ so that $(\mathbf{I} + t\mathbf{A})$ is invertible.

During a typical simulation, the simulated atoms quickly diffuse into the nonsimulated atoms. Thus, the simulated atoms can become highly strung-out. When we say that the motion $\mathbf{y}_k(t)$ of simulated atom k satisfies the equations of molecular dynamics, we note that the force on this atom generally arises from both simulated and nonsimulated atoms. We will assume that, in addition to the usual invariances, the force on any atom is only produced by other (simulated or nonsimulated) atoms within a cut-off d_{cut}. Thus, the critical issue for making an efficient simulation is to find all atoms within d_{cut} of each simulated atom at each time step. Below, we develop an efficient method for doing this based on 1) occasionally redefining which atoms are the simulated atoms, 2) using basic (time-dependent) periodicity of the motion, 3) using an integer Gram-Schmidt process to redefine the unit cell to be "fat" and then redefining the simulated atoms in a consistent way. The method outlined below keeps the simulated atoms in a deforming unit cell by occasional redefinition of simulated atoms, until that cell becomes elongated.

3.1. Starting the simulation

At $t = 0$ assign initial positions and initial velocities of simulated atoms in a parallelepiped defined by $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$, and extend them periodically using $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ to get the nonsimulated atoms as given by (9). Start the simulation.

3.2. Nonsimulated atoms

A basic property of OMD is that one can redefine the simulated atoms at any particular time step, by choosing a new set of simulated atoms that are images of the the given set of simulated atoms under the group. The group element chosen can be different for different simulated atoms. When restarting the simulation, the new simulated atoms are given velocities that are obtained by time differentiating (9). The methods described here to speed up OMD rely on occasionally choosing new simulated atoms that are in some sense close together.

At every time step use (9) to get the positions of the nonsimulated atoms which can also be understood as images of simulated atoms.

3.3. Criterion that a simulated atom lies in a unit cell

If a simulated atom j passes out of the unit cell $\mathcal{U}(t_1) = \{\lambda_1(\mathbf{I} + t_1\mathbf{A})\mathbf{e}_1 + \lambda_2(\mathbf{I} + t_1\mathbf{A})\mathbf{e}_2 + \lambda_3(\mathbf{I} + t_1\mathbf{A})\mathbf{e}_3 : 0 \leq \lambda_1, \lambda_2, \lambda_3 < 1\}$ at time t_1, then we declare it no longer to be a simulated atom and replace it by the $\mathbf{y}_{\nu,j}$ (note: same j), which has just entered $\mathcal{U}(t_1)$. The passing in and out of $\mathcal{U}(t_1)$ and the value of ν can be detected in the following way. Let $\mathbf{e}^1, \mathbf{e}^2, \mathbf{e}^3$ be the reciprocal vectors, i.e., the unique vectors that satisfy $\mathbf{e}^i \cdot \mathbf{e}_j = \delta^i_j$. Outward normals of the faces of $\mathcal{U}(t_1)$ are

$$\pm(\mathbf{I} + t_1\mathbf{A})^{-T}\mathbf{e}^1, \quad \pm(\mathbf{I} + t_1\mathbf{A})^{-T}\mathbf{e}^2, \quad \pm(\mathbf{I} + t_1\mathbf{A})^{-T}\mathbf{e}^3. \tag{10}$$

Thus, because $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ has been chosen as linearly independent and right handed, a point $\mathbf{x} \in \mathcal{U}$ if and only if

$$0 \leq \mathbf{x} \cdot (\mathbf{I} + t_1\mathbf{A})^{-T}\mathbf{e}^i < 1, \tag{11}$$

So, if any of the inequalities in (11) are violated by $\mathbf{x} = \mathbf{y}_j(t_1)$, then atom j has passed out of $\mathcal{U}(t_1)$, and it should be replaced by (ν, j) that has just entered $\mathcal{U}(t_1)$. If, say, (11) is violated at $i = 2$ and at 1, then the new simulated atom is $((0, -1, 0), j)$. Fig. 1 illustrates this procedure. The green atom goes outside the domain and at the same time its image (blue atom) enters the domain from the neighboring cell of non simulated atoms. In this case the inequalities in (11) are violated at i=3 and at 0. Therefore, the ν of the redefined new simulated atom is $(0, 0, 1)$. As noted above the new simulated atom has a different velocity than the old one depicted by the red vector in the Fig. 1 – their velocities are related by the time derivative of (9):

$$\dot{\mathbf{y}}_{\nu,k}(t) = \dot{\mathbf{y}}_k(t) + \nu^i \mathbf{A}\mathbf{e}_i. \tag{12}$$

So, when using the velocity Verlet algorithm (see below) for the new simulated atom, the new position and the new velocity have to be used.

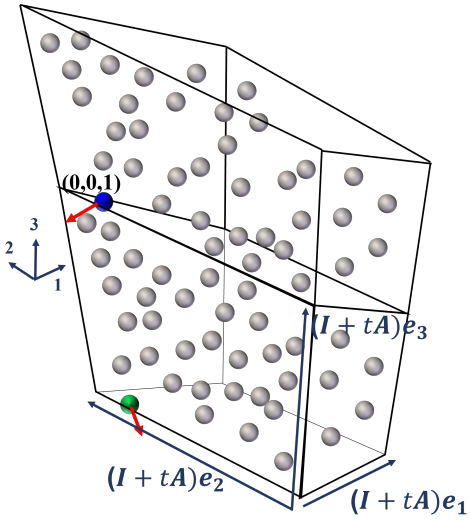


Fig. 1: Criteria for redefinition: Simulated atom (green) goes outside the domain and is replaced by the non-simulated atom (blue) which enters at the same time

3.4. Computation of forces

The force on a simulated atom is calculated using either minimum image convention or cell linked-list method which are commonly used techniques in molecular dynamics to accelerate the computation of potential and force evaluation [1, 14]. The choice of the method depends on the number of simulated atoms in the system. Both the above methods are modified here to take into account the time-dependence of the parallelepiped associated to the three vectors $((\mathbf{I} + t\mathbf{A})\mathbf{e}_1, (\mathbf{I} + t\mathbf{A})\mathbf{e}_2, (\mathbf{I} + t\mathbf{A})\mathbf{e}_3)$. These vectors constitute the fundamental domain comprising of simulated atoms.

In the minimum image convention, every simulated atom is at most interacting with one image of other simulated atoms in the fundamental domain as long as the minimum distance between points on opposite faces of domain is less than twice the cut-off radius d_{cut} of the interatomic potential. The main idea then is to minimize the distances between each pair of simulated atoms in the domain to find the nearest image.

The distance between simulated atom k and m at time t can be written as:

$$\begin{aligned} \mathbf{r}_{k,m} &= \mathbf{y}_k - \mathbf{y}_m = \{\lambda_1(\mathbf{I} + t\mathbf{A})\mathbf{e}_1 + \lambda_2(\mathbf{I} + t\mathbf{A})\mathbf{e}_2 + \lambda_3(\mathbf{I} + t\mathbf{A})\mathbf{e}_3\} \\ \mathbf{\Lambda} &= [\lambda_1, \lambda_2, \lambda_3]^{\mathrm{T}} = [\mathbf{r}_{k,m} \cdot (\mathbf{I} + t\mathbf{A})^{-\mathrm{T}}\mathbf{e}^1,\ \mathbf{r}_{k,m} \cdot (\mathbf{I} + t\mathbf{A})^{-\mathrm{T}}\mathbf{e}^2,\ \mathbf{r}_{k,m} \cdot (\mathbf{I} + t\mathbf{A})^{-\mathrm{T}}\mathbf{e}^3]^{\mathrm{T}} \end{aligned} \tag{13}$$

$\mathbf{r}_{k,m}$ is the shortest distance if and only if $|\lambda_i| \leq 0.5$. If $|\lambda_i| > 0.5$ then simulated atom k interacts with an image of simulated atom m and the distance between them is given by $\mathbf{r}_{k,m} - \lfloor(\lambda_i)\rfloor(\mathbf{I} + t\mathbf{A})\mathbf{e}_i$ where $\mathbf{r}_{k,m}$ and λ_i is known from the calculation above and $\lfloor x \rfloor$ is the closest integer to x. This is illustrated in the Fig. 2(a). In other words, it is similar to representing an alike fundamental domain centered around each atom (blue atom in the Fig. 2(a)) and computing the forces using atoms within cutoff d_{cut}.

The cell list algorithm is another effective method which is used here when the number of simulated atoms in the domain is large. Here, the fundamental domain is subdivided into cells. All atoms are assigned to the cells according to their positions, and the interactions are computed between particles in the same or neighbouring cells. So rather than looping over each particle pair in the simulation domain, one only loops over the particle pairs in these 27 cells (or 9 cells in 2d). This is true only when minimum distance between opposite faces of each cell is greater than the cut-off radius d_{cut}. This is 'minimum distance requirement'. The number of cells in each direction $(\mathbf{I} + t\mathbf{A})\mathbf{e}_i$ can be changed on the fly depending on the dimensions of the domain and particle density, which can vary during a simulation. This number is computed using $n_i = \{\lfloor d_{i,min}/d_{cut} \rfloor - 1\}$, where $d_{i,min}$ is the minimum distance between two neighboring parallel faces of the fundamental domain. The formula for its computation is given in Section 3.4. Since $d_{i,min}$ decreases as the flow evolves, the total number of cells changes in order to satisfy the minimum distance requirement. For dilute systems with a fairly big fundamental domain, n_i given by the above expression doesn't give optimized load balancing when parallelized. This happens because of the presence of very few atoms in each cell. In that case, the cell size is increased to achieve a good speedup.

The method is depicted in the Fig. 2(b). Here, different colors of the atoms correspond to different cells. Only a few subdivisions are shown here for clarity. The motion of the fundamental domain depends on the choice of $\mathbf{A}$.

When the number of atoms in the system is comparatively fewer, an image convention is used. This is because the relative overhead of the generation of cells and sorting increases with a decrease in the number of atoms. Both the methods are general and work with any $\mathbf{A}$.

3.5. Criterion for excessive distortion of the unit cell

As mentioned above, for the minimum image convention and the cell-linked list, it is necessary that the distance between two neighboring parallel faces should maintain a particular minimum length. Since the unit cell follows the macroscopic motion of the flow, it distorts with time and hence the minimum distance decreases. We need a criteria to decide that when the unit cell has distorted enough and the new unit cell is needed. To formulate this criterion, let $\mathcal{N}$ be the neighbors of $\mathcal{U}$, where $\mathcal{N}$ and $\mathcal{U}$ are unit cells. Suppose the minimum distance between $\mathcal{U}$ and $\partial(\mathcal{U} \cup \mathcal{N})$ is achieved at the two points $\mathbf{x} \in \partial\mathcal{U}$ and $\mathbf{y} \in \partial(\mathcal{U} \cup \mathcal{N})$ (Clearly we can assume both points are on their respective boundaries.) We have that $\mathbf{y} - \mathbf{x}$ must be a generalized outward normal to $\mathcal{U}$, i.e., $(\mathbf{y} - \mathbf{x}) \cdot (\mathbf{x}' - \mathbf{x}) \leq 0$ for all $\mathbf{x}' \in \partial(\mathcal{U} \cup \mathcal{N})$ because otherwise we could reduce the distance $|\mathbf{y} - \mathbf{x}|$ by perturbing $\mathbf{x}$. Also, $\mathbf{y}$ cannot be on an edge of $\partial(\mathcal{U} \cup \mathcal{N})$ because all edges of $\mathcal{U} \cup \mathcal{N}$ have interior acute angles and so the distance $|\mathbf{y} - \mathbf{x}|$ could be reduced by perturbing $\mathbf{y}$. Thus, $\mathbf{y}$ must be on a face and $\mathbf{x} - \mathbf{y}$ must be perpendicular to that face. Hence, $|\mathbf{y} - \mathbf{x}|$ must be the minimum distance between two neighboring parallel faces, which is

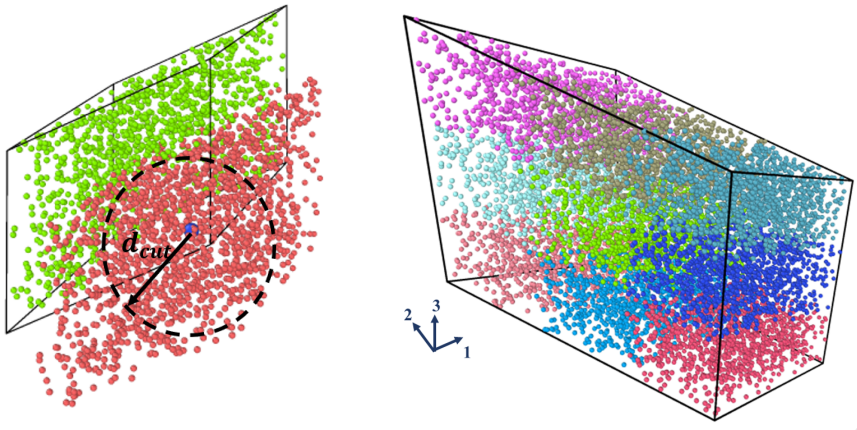
$$\min_{i=1,2,3} \frac{1}{|(\mathbf{I} + t\mathbf{A})^{-T}\mathbf{e}^i|} \tag{14}$$

Algorithm 1 Cell list: Assign atoms to cells

for i=1 to 3 **do** ▷ Number of cells in direction i
  $n_i \leftarrow \{\lfloor d_{i,min}/d_{cut} \rfloor - 1\}$
end for
for k=1 to N **do** ▷ Assign simulated atoms to different cells
  $\lambda_1 \leftarrow \mathbf{y}_k \cdot (\mathbf{I} + t\mathbf{A})^{-\mathrm{T}}\mathbf{e}^1$
  $\lambda_2 \leftarrow \mathbf{y}_k \cdot (\mathbf{I} + t\mathbf{A})^{-\mathrm{T}}\mathbf{e}^2$
  $\lambda_3 \leftarrow \mathbf{y}_k \cdot (\mathbf{I} + t\mathbf{A})^{-\mathrm{T}}\mathbf{e}^3$
  $(c_1[k], c_2[k], c_3[k]) \leftarrow (\lfloor n_1[\lambda_1 - \frac{1}{2n_1}] \rfloor, \lfloor n_2[\lambda_2 - \frac{1}{2n_2}] \rfloor, \lfloor n_3[\lambda_3 - \frac{1}{2n_3}] \rfloor)$
  if $c_i[k]$ is less than 0 **then** ▷ $c_i[k] \in \{0, , , n_i - 1\}, c_i[k] \in \mathbb{Z}^+, i \in \{1, 2, 3\}$
    $c_i[k] \leftarrow n_i - 1$
    $\mathbf{y}_k \leftarrow \mathbf{y}_k + (\mathbf{I} + t\mathbf{A})\mathbf{e}_i$ ▷ Redefine non-simulated atom as simulated atom
    $\mathbf{v}_k \leftarrow \mathbf{v}_k + \mathbf{A}\mathbf{e}_i$
  end if
  if $c_i[k]$ is greater than or equal to n_i **then**
    $c_i[k] \leftarrow 0$
    $\mathbf{y}_k \leftarrow \mathbf{y}_k - (\mathbf{I} + t\mathbf{A})\mathbf{e}_i$
    $\mathbf{v}_k \leftarrow \mathbf{v}_k - \mathbf{A}\mathbf{e}_i$
  end if
  $c \leftarrow (n_1 n_2 c_3[k]) + (n_1 c_2[k]) + (c_1[k]) + 1$ ▷ Cell identifier, $c \in \{0, 1, ...(n_1 n_2 n_3 - 1)\}$
  $natoms[c] \leftarrow natoms[c] + 1$ ▷ Counter for the number of simulated atoms in c^{th} cell
  $list(c, natoms[c]) \leftarrow k$ ▷ List of the simulated atoms in c^{th} cell
end for

Algorithm 2 Cell list: Compute forces

for k=1 to N **do** ▷ Looping over total number of simulated atoms in the fundamental domain
  for $l_i = c_i[k] - 1$ to $c_i[k] + 1$ **do** ▷ Looping over neighboring cells in i^{th} direction, $i \in \{1, 2, 3\}$
    if l_i is less than 0 **then** ▷ Accounting for atoms near the domain boundary
      $rshift[i] \leftarrow -1$
    else if l_i is greater than or equal to n_i **then**
      $rshift[i] \leftarrow +1$
    else
      $rshift[i] \leftarrow 0$
    end if
    $c_1 \leftarrow ((l_1 + n_1) \bmod \mathrm{n}_1) + ((\mathrm{l}_2 + \mathrm{n}_2) \bmod \mathrm{n}_2)\mathrm{n}_1 + ((\mathrm{l}_3 + \mathrm{n}_3) \bmod \mathrm{n}_3)\mathrm{n}_1\mathrm{n}_2 + 1$
    for $p = 1$ to $natom[c_1]$ **do** ▷ Looping over number of atoms in c_1^{th} cell
      $\tilde{k} \leftarrow list(c_1, p)$
      $\mathbf{r} \leftarrow (\mathbf{y}_{\tilde{k}} + rshift[i](\mathbf{I} + t\mathbf{A})\mathbf{e}_i) - \mathbf{y}_k$ ▷ Minimum interatomic distance
      if $|\mathbf{r}|$ is less than cutoff **then**
        Compute forces
      end if
    end for
  end for
end for



(a) (b)

Fig. 2: Different techniques for computation of forces (a) Minimum Image Convention adapted to consider deformation of domain (b) Cell List: Simulation domain is divided into cells. Both domain and cells are identically deformed.

So the criterion for restarting the simulation with a new unit cell is when the following inequality is first violated

$$\min_{i=1,2,3} \frac{1}{|(\mathbf{I} + t\mathbf{A})^{-T}\mathbf{e}^i|} \leq m d_{cut}. \tag{15}$$

Here, $m = 2$ for the whole fundamental domain when using minimum image convention and m depends on the maximum number of cells $\max_{i=1,2,3} n_i$ when cell list method is used to maintain 'minimum distance requirement'. Fig. 3 shows a fat cubical domain ($t = 0$) which later gets deformed to a highly distorted cell (different projections shown) at time t_1. Here the criterion (15) is violated and some part of the cutoff region is located out of the box.

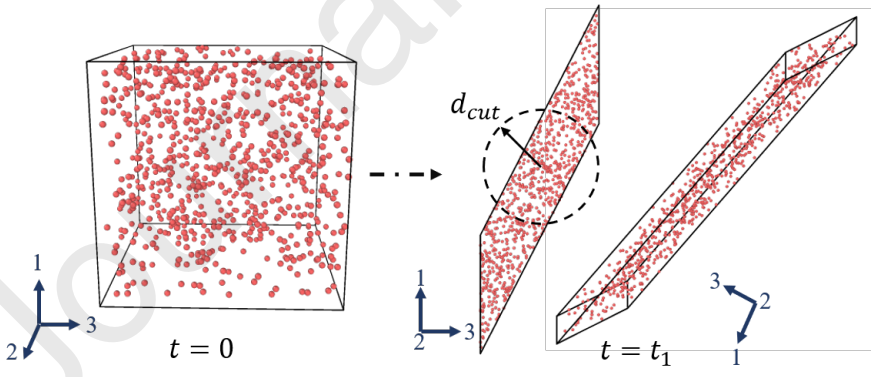


Fig. 3: Criteria for distortion: A sufficiently fat box at $t = 0$ gets highly deformed at $t = t_1$.

3.6. Approximate orthogonalization by an integer Gram-Schmidt method

In principle, the simulation may be done in a previously defined suitably distorted box, but for a number of practical reasons this may be unattractive. For example, the cut-off sphere may then be located in many boxes at the same time. Also, we cannot use the idea of the minimum image convention and the cell list because then simulated atoms would be interacting with more than one image of non simulated atoms. To improve this situation, a given lattice basis that has become highly distorted is transformed into a "fat" lattice basis which is closer to orthogonal. One needs a suitable mathematical definition of "fat basis" to achieve this. Orthogonal basis can be determined by

considering the lengths of the Gram-Schmidt vectors. Orthogonality of a lattice basis is closely tied to the lengths of the Gram-Schmidt vectors [19].

Gram-Schmidt orthogonalization: Given a basis $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_m\}$ of a subspace $\mathbb{H}_m$ of $\mathbb{R}^n$, this method gives back an orthogonal basis $\{\mathbf{b}_1^*, \mathbf{b}_2^*, \ldots, \mathbf{b}_m^*\}$ of $\mathbb{H}_m$.

$$\begin{aligned}
\mathbf{b}_1^* &= \mathbf{b}_1, \\
\mathbf{b}_2^* &= \mathbf{b}_2 - \mu_{1,2}\mathbf{b}_1, \quad \text{where}, \mu_{1,2} = \frac{\mathbf{b}_2 \cdot \mathbf{b}_1^*}{\mathbf{b}_1^* \cdot \mathbf{b}_1^*} \\
\mathbf{b}_m^* &= \mathbf{b}_m - \sum_{i<m} \mu_{i,m}\mathbf{b}_i, \quad \text{where}, \mu_{i,m} = \frac{\mathbf{b}_m \cdot \mathbf{b}_i^*}{\mathbf{b}_i^* \cdot \mathbf{b}_i^*}
\end{aligned} \tag{16}$$

In matrix form, $\mathbf{B} = \mathbf{B}^*\mathbf{U}$, where basis vectors are columns in $\mathbf{B}$ and $\mathbf{B}^*$ and $\mathbf{U}$ is an upper triangular matrix with diagonal elements 1.

$$\begin{bmatrix} \mathbf{b}_1 & \mathbf{b}_2 & \ldots & \mathbf{b}_m \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1^* & \mathbf{b}_2^* & \ldots & \mathbf{b}_m^* \end{bmatrix} \begin{bmatrix} \mu_{1,1} & \mu_{1,2} & \cdots & \mu_{1,m} \\ 0 & \mu_{2,2} & \cdots & \mu_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mu_{m,m} \end{bmatrix}$$

Since the coefficients $\mu_{i,j}$ do not usually lie in $\mathbb{Z}$, the resulting vectors are not usually elements of the lattice and hence Gram-Schmidt process is not useful, in general, for lattices. A. Lenstra, H. Lenstra, and L. Lovasz proposed the celebrated LLL algorithm [31], an approximation of basis reduction which exploits a Gram-Schmidt orthogonalization GSO. LLL algorithm produces a nearly orthonormal integer basis spanning the space of a given, possibly non-integer basis. Below, we term this method an integer Gram-Schmidt process.

The LLL Algorithm contains two steps [12]:

1. **Normalization:** Lattice invariant operations are applied to transform the upper triangular matrix $\mathbf{U}$ to as close as possible to the identity matrix. If $\mathbf{U}$ were the identity, then $\mathbf{B}$ itself would be orthogonal. Let $\mu_{i,j}$ be the j_{th} entry of the i_{th} row of $\mathbf{U}$. By subtracting $\lfloor \mu_{i,j} \rfloor$ times the i_{th} column of $\mathbf{U}$ from the j_{th} column, the new entry $\mu'_{i,j}$ at position i, j will satisfy $-(1/2) < \mu'_{i,j} \leq (1/2)$. The entries in a row below the i_{th} row of $\mathbf{U}$ remain unchanged. By following these steps from the last to the first row, we obtain a basis $\mathbf{B}' = \mathbf{B}^*\mathbf{U}'$ with $|\mu'_{i,j}| \leq 1/2$ for every $1 \leq i < j \leq m$ in GSO. This step is also called size reduction.
2. **Swapping:** If there exists a j such that

$$\|\mathbf{b}_{j+1}^* + \mu_{j+1,j}\mathbf{b}_j^*\|^2 < \frac{3}{4}\|\mathbf{b}_j^*\|^2, \tag{17}$$

swap $\mathbf{b}_j$ and $\mathbf{b}_{j+1}$. Then return to Normalization.

The constant $\delta = 3/4$ in (17) is chosen for simplicity. Any constant between $1/4$ and 1 can guarantee that the algorithm terminates in polynomial time. If the basis vectors $\mathbf{b}_j^*$ violate (17) then the algorithm terminates. The LLL-algorithm alternates the normalization and swapping steps: it normalizes the basis and then searches for two consecutive basis elements which should be swapped. This is continued until (17) does not hold. The final $\mathbf{B}'$ matrix is called the δ-LLL reduced matrix.

3.7. Remapping the simulated atoms

Once the new nice lattice basis is obtained from the LLL algorithm, then atoms are mapped into the parallelepiped formed by this new set of basis vectors. Every simulated atom in the distorted unit cell which does not belong to new unit cell has a corresponding non simulated atom in the new unit cell. These images are then redefined as the new simulated atoms and simulation is continued. One has to be careful to restart with the correct positions and velocities, noting that some of the new simulated atoms were previously nonsimulated atoms. Some methods available in the literature rely on the possibility of finding a reproducible lattice where the lattice points occupy the same points as those of the initial lattice [3, 21, 29]. We don't impose that condition on the simulation cell. Rather, the idea is to get a basis which is close to being orthogonal and hence can construct a sufficiently fat unit cell which obeys the minimum distance criteria. It is seen in Section 5 that there is no discontinuity in the macroscopic properties of the system

during remapping, as must be true. The resulting simulation is exactly the same as if the original set of simulated atoms had been used for all time.

Let $\tilde{\mathbf{e}}_\mathbf{i}$ be the new basis vectors. If the remapping is done at time t_1 then $\mathbf{e}_i$ changes to,

$$(\mathbf{I} + t_1\mathbf{A})\mathbf{e}_i = \tilde{\mathbf{e}}_i \Rightarrow \mathbf{e}_i = (\mathbf{I} + t_1\mathbf{A})^{-1}\tilde{\mathbf{e}}_i \tag{18}$$

This changes the corresponding reciprocal vectors $\mathbf{e}^i$. (It is important to change the values of $\mathbf{e}_i$ and $\mathbf{e}^i$ whenever remapping happens in the code.) The positions and velocities of the non-simulated atoms which are redefined as simulated atoms at time t_1 are then given by

$$\mathbf{y}_{\nu,k}(t_1) = \mathbf{y}_k(t_1) + \sum_{i=1}^{3} \nu^i \tilde{\mathbf{e}}_i \tag{19}$$

$$\dot{\mathbf{y}}_{\nu,k}(t_1) = \dot{\mathbf{y}}_k(t_1) + \sum_{i=1}^{3} \nu^i \mathbf{A}\mathbf{e}_i \tag{20}$$

where the value of triplet of integers $\nu^\mathbf{i}$ for the non simulated atoms lying in the fat cell are

$$\nu^i = -\lfloor(\mathbf{y}_k(t_1) \cdot (\mathbf{I} + t_1\mathbf{A})^{-T}\mathbf{e}^i - 0.5)\rfloor \tag{21}$$

Fig. 4 shows an example of remapping for a general incompressible flow. A new basis is defined at this instant using the LLL Algorithm where the size of the domain constituted by the new basis is different than the original domain. Orange atoms are remapped to blue atoms which are redefined as simulated atoms. Depending on the value of the matrix $\mathbf{A}$, and especially for certain highly distorting compressible flows, it can happen that, at some point in time, there might not exist a unit cell which is sufficiently fat and which obeys the minimum distance criteria. At this point the cutoff necessarily forces atoms to lie in several – or in particularly unfavorable cases many – copies of any fundamental domain. In those cases, sole use of integer Gram-Schmidt approach may not be sufficient. Depending on the size of this domain, one can expand the size of fundamental domain by including non-simulated atoms in the simulation (non-simulated atoms are redefined as simulated atoms) such that the domain satisfy minimum distance criteria. The simulation can also be continued in the original domain but approach for finding neighbors is altered slightly. In this case, other neighboring domains of non-simulated atoms are explicitly constructed and forces are computed from the atoms lying in these neighboring and center domains.

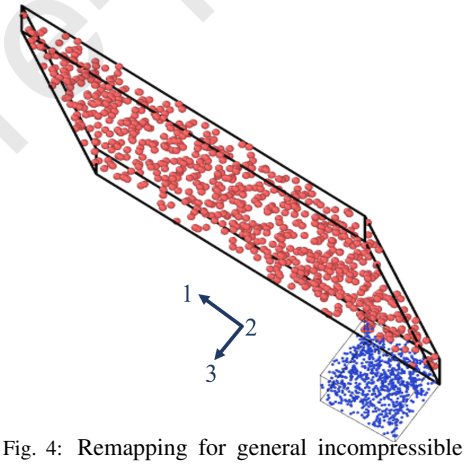


Fig. 4: Remapping for general incompressible flows: orange atoms are remapped to blue atoms.

Fig. 5 shows a final flowchart for single time stepping in the OMD simulation. Note that we heavily perform the redefinition of non-simulated atoms as simulated atoms in the simulation. After solving for the trajectory of simulated atoms using velocity Verlet, few simulated atoms can move out of the cell. At this stage, the corresponding non-simulated atoms which enter are redefined as simulated atoms and the simulation is continued. Ovito software has been used for all the visualization purpose in this work [40].

4. Velocity Verlet algorithm and Objective MD

The propagation scheme of the velocity Verlet algorithm is widely used in traditional MD codes. We show that it is nicely consistent with Objective MD. It exactly inherits the invariant manifold of (continuous) molecular dynamics: if you start on the manifold, you stay on the manifold exactly, even with discrete time steps. Or, equivalently, if you use the velocity Verlet algorithm only for the simulated atoms then it is automatically being used for all the atoms.

This is explained here in the general case of time dependent isometry groups. The time dependent translation group is a special case.

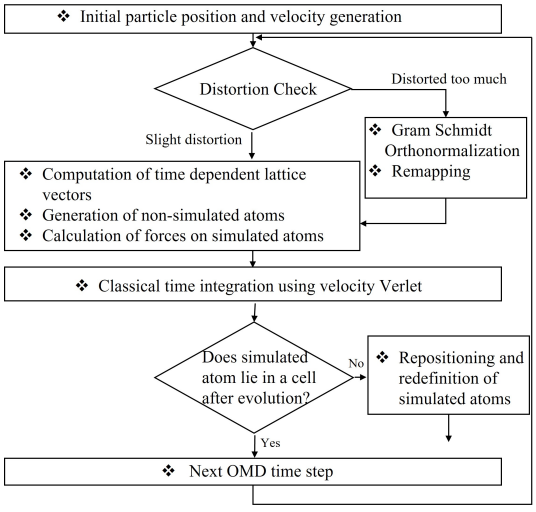


Fig. 5: Flow chart of single time stepping in OMD.

4.1. Velocity Verlet algorithm

Let $\mathcal{G} = \{g_1, g_2, \ldots, g_N\}$, $g_1 = id$, be a time-dependent discrete group of isometries with affine time dependence:

$$g_i = (\mathbf{Q}_i|\mathbf{a}_i t + \mathbf{b}_i), \quad \mathbf{Q}_i \in \mathrm{O}(3), \quad \mathbf{a}_i, \mathbf{b}_i \in \mathbb{R}^3, \quad i = 1, \ldots, N. \tag{22}$$

Suppose the atomic forces satisfy frame-indifference and permutation invariance. Consider a continuous OMD simulation defined by

$$\begin{aligned} m_k \ddot{\mathbf{y}}_{1,k}(t) &= -\frac{\partial \varphi}{\partial \mathbf{y}_{1,k}}(\ldots, \mathbf{y}_{i,1}(t), \ldots, \mathbf{y}_{i,M}(t), \mathbf{y}_{i+1,1}(t), \ldots, \mathbf{y}_{i+1,M}(t), \ldots) \\ &= -\frac{\partial \varphi}{\partial \mathbf{y}_{1,k}}(\ldots, g_i(\mathbf{y}_{1,1}(t), t), \ldots, g_i(\mathbf{y}_{1,M}(t), t), g_{i+1}(\mathbf{y}_{1,1}(t), t), \ldots, g_{i+1}(\mathbf{y}_{1,M}(t), t), \ldots), \\ &\quad \mathbf{y}_{1,k}(0) = \mathbf{y}_k^0, \quad \dot{\mathbf{y}}_{1,k}(0) = \mathbf{v}_k^0, \quad k = 1, \ldots, M, \end{aligned} \tag{23}$$

Here the notation $g_i(\mathbf{y}, t)$ stands for $\mathbf{Q}_i\mathbf{y}(t) + \mathbf{a}_i t + \mathbf{b}_i$. From now on we simplify the notation and write $\mathbf{y}_i(t) = \mathbf{y}_{1,i}(t)$. The $\mathbf{y}_i(t)$, $i = 1, \ldots, M$, are called the simulated atoms.

Consider a sequence of equal time steps, $0, t_1, t_2, \ldots$, with $t_{i+1} - t_i = \Delta t$, The velocity Verlet algorithm applied to this "small" system, i.e., the equations for the simulated atoms, is

$$\begin{aligned} \mathbf{y}_k(t_{i+1}) &= \mathbf{y}_k(t_i) + \mathbf{v}_k(t_i)\Delta t - \frac{(\Delta t)^2}{2m_k}\frac{\partial \varphi}{\partial \mathbf{y}_{1,k}}(\ldots, g_i(\mathbf{y}_1(t_i), t_i), \ldots, g_i(\mathbf{y}_M(t_i), t_i), \ldots), \\ \mathbf{v}_k(t_{i+1}) &= \mathbf{v}_k(t_i) - \frac{\Delta t}{2m_k}\left(\frac{\partial \varphi}{\partial \mathbf{y}_{1,k}}(\ldots, g_i(\mathbf{y}_1(t_i), t_i), \ldots, g_i(\mathbf{y}_M(t_i), t_i), \ldots)\right. \\ &\quad \left. + \frac{\partial \varphi}{\partial \mathbf{y}_{1,k}}(\ldots, g_i(\mathbf{y}_1(t_{i+1}), t_{i+1}), \ldots, g_i(\mathbf{y}_M(t_{i+1}), t_{i+1}), \ldots)\right) \end{aligned} \tag{24}$$

Here, we have consolidated the usual velocity Verlet algorithm into two equations.

4.2. Proof that the velocity Verlet algorithm for the simulated atoms implies its satisfaction for the full system

Denote the full set of atom motions by $\mathbf{y}_{n,k}(t) = g_n(\mathbf{y}_k(t), t)$. In the continuous case these satisfy the full system of MD, according to the basic theorem of OMD.

Fix $m \in \{1, \ldots, N\}$ throughout this section. Define the permutation Π (which depends on m but not on t) by

$$\mathbf{y}_{\Pi(\ell,k)}(t) = g_m^{-1}(\mathbf{y}_{\ell,k}(t), t) = g_m^{-1} g_\ell(\mathbf{y}_k(t)). \tag{25}$$

Recalling that $g_1 = id$ note that $\Pi(m,k) = (1,k)$. Apply the group element $g_m(\cdot\,, t_{i+1})$ to the first equation in (24). This gives

$$\begin{aligned}
\mathbf{y}_{m,k}(t_{i+1}) &= g_m(\mathbf{y}_k(t_{i+1}), t_{i+1}) \\
&= g_m\Big(\mathbf{y}_k(t_i) + \mathbf{v}_k(t_i)\Delta t - \frac{(\Delta t)^2}{2m_k}\frac{\partial \varphi}{\partial \mathbf{y}_{1,k}}(\dots, \mathbf{y}_{\ell,1}(t_i), \dots, \mathbf{y}_{\ell,M}(t_i), \dots),\ t_{i+1}\Big) \\
&= \mathbf{Q}_m\mathbf{y}_k(t_i) + \mathbf{Q}_m\mathbf{v}_k(t_i)\Delta t - \frac{(\Delta t)^2}{2m_k}\mathbf{Q}_m\frac{\partial \varphi}{\partial \mathbf{y}_{\Pi(m,k)}}(\dots, \mathbf{y}_{\ell,1}(t_i), \dots, \mathbf{y}_{\ell,M}(t_i), \dots) \\
&\qquad + \mathbf{a}_m t_{i+1} + \mathbf{b}_m, \\
&= \mathbf{Q}_m\mathbf{y}_k(t_i) + \mathbf{a}_m t_i + \mathbf{b}_m + \mathbf{Q}_m\mathbf{v}_k(t_i)\Delta t + \mathbf{a}_m\Delta t \\
&\qquad - \frac{(\Delta t)^2}{2m_k}\mathbf{Q}_m\frac{\partial \varphi}{\partial \mathbf{y}_{m,k}}(\dots, \mathbf{y}_{\Pi(\ell,1)}(t_i), \dots, \mathbf{y}_{\Pi(\ell,M)}(t_i), \dots) \\
&= \mathbf{y}_{m,k}(t_i) + \mathbf{v}_{m,k}(t_i)\Delta t - \frac{(\Delta t)^2}{2m_k}\mathbf{Q}_m\frac{\partial \varphi}{\partial \mathbf{y}_{m,k}}(\dots, g_m^{-1}(\mathbf{y}_{\ell,1}(t_i), t_i), \dots, g_m^{-1}(\mathbf{y}_{\ell,M}(t_i), t_i), \dots) \\
&= \mathbf{y}_{m,k}(t_i) + \mathbf{v}_{m,k}(t_i)\Delta t \\
&\qquad - \frac{(\Delta t)^2}{2m_k}\mathbf{Q}_m\frac{\partial \varphi}{\partial \mathbf{y}_{m,k}}(\dots, \mathbf{Q}_m^T\mathbf{y}_{\ell,1}(t_i) - \mathbf{Q}_m^T(\mathbf{a}_m t_i + \mathbf{b}_m), \dots, \mathbf{Q}_m^T\mathbf{y}_{\ell,M}(t_i) - \mathbf{Q}_m^T(\mathbf{a}_m t_i + \mathbf{b}_m), \dots) \\
&= \mathbf{y}_{m,k}(t_i) + \mathbf{v}_{m,k}(t_i)\Delta t - \frac{(\Delta t)^2}{2m_k}\frac{\partial \varphi}{\partial \mathbf{y}_{m,k}}(\dots, \mathbf{y}_{\ell,1}(t_i), \dots, \mathbf{y}_{\ell,M}(t_i), \dots).
\end{aligned} \tag{26}$$

This is the first step in the velocity Verlet algorithm for the full system. The last few lines above use permutation invariance and frame-indifference and follow the lines of the continuous proof.

In the above argument the quantity $\mathbf{v}_{m,k}(t)$ is simply defined (for obvious reasons) by the formula $\mathbf{v}_{m,k}(t) = \mathbf{Q}_m\mathbf{v}_k(t) + \mathbf{a}_m$.

Now apply the transformation $\mathbf{v} \to \mathbf{Q}_m\mathbf{v} + \mathbf{a}_m$ (no t dependence) to both sides of the second step of the algorithm (24). We get

$$\begin{aligned}
\mathbf{v}_{m,k}(t_{i+1}) &= \mathbf{v}_{m,k}(t_i) - \frac{\Delta t}{2m_k}\mathbf{Q}_m\Big(\frac{\partial \varphi}{\partial \mathbf{y}_{1,k}}(\dots, \mathbf{y}_{\ell,1}(t_i), \dots, \mathbf{y}_{\ell,M}(t_i), \dots) \\
&\qquad + \frac{\partial \varphi}{\partial \mathbf{y}_{1,k}}(\dots, \mathbf{y}_{\ell,1}(t_{i+1}), \dots, \mathbf{y}_{\ell,M}(t_{i+1}), \dots)\Big) \\
&= \mathbf{v}_{m,k}(t_i) - \frac{\Delta t}{2m_k}\Big(\frac{\partial \varphi}{\partial \mathbf{y}_{m,k}}(\dots, \mathbf{y}_{\ell,1}(t_i), \dots, \mathbf{y}_{\ell,M}(t_i), \dots) \\
&\qquad + \frac{\partial \varphi}{\partial \mathbf{y}_{m,k}}(\dots, \mathbf{y}_{\ell,1}(t_{i+1}), \dots, \mathbf{y}_{\ell,M}(t_{i+1}), \dots)\Big).
\end{aligned} \tag{27}$$

This completes the argument.

5. Connection with Continuum Mechanics

The macroscopic motion simulated by OMD in Lagrangian and Eulerian description is given by $\mathbf{y}(x,t) = (\mathbf{I} + t\mathbf{A})\mathbf{x}$ and $\mathbf{v}(\mathbf{y},t) = \mathbf{A}(\mathbf{I} + t\mathbf{A})^{-1}\mathbf{y}$, respectively. These are termed affine motions. The Cauchy stress ($\mathbf{T}$) based on an affine motion is uniform in space for all accepted constitutive relations in continuum mechanics. The balance of linear momentum thus becomes,

$$\rho(\mathbf{v}_t + \nabla\mathbf{v}\mathbf{v}) = \rho(-\mathbf{A}(\mathbf{I} + t\mathbf{A})^{-1}\mathbf{A}(\mathbf{I} + t\mathbf{A})^{-1}\mathbf{y} + \mathbf{A}(\mathbf{I} + t\mathbf{A})^{-1}\mathbf{A}(\mathbf{I} + t\mathbf{A})^{-1}\mathbf{y}) = 0 = \boldsymbol{\nabla}\cdot\mathbf{T} = 0. \tag{28}$$

Thus, this family of flows satisfies equation of balance of linear momentum motion identically for all accepted constitutive laws of continuum mechanics of solids and fluids, e.g., Navier-Stokes fluids, general non-Newtonian fluids, nonlinear elastic solids, plastic solids, etc.. The other two mass and energy balance laws are

$$\begin{aligned}
\rho_t + \nabla\cdot(\rho\mathbf{v}) &= 0, \\
\rho(e_t + \nabla e\cdot\mathbf{v}) &= \mathbf{T}\cdot\nabla\mathbf{v} - \boldsymbol{\nabla}\cdot\mathbf{q}.
\end{aligned} \tag{29}$$

The first of these determines the density $\rho(t) = \rho_0 \exp\left(-\int_0^t \mathrm{tr}\left(\mathbf{A}(\mathbf{I}+s\mathbf{A})^{-1}\right)\mathrm{d}s\right)$. With the motion $\mathbf{v}(\mathbf{y},t) = \mathbf{A}(\mathbf{I}+t\mathbf{A})^{-1}\mathbf{y}$ the temperature field θ becomes uniform and time dependent $\theta(t)$. All accepted constitutive relations then make stress and energy also independent of position, $\mathbf{T} = \mathbf{T}(t)$, $e = e(t)$. This pure time dependence gives the following parameter $\mathbf{A}$-dependent ordinary differential equation for the temperature after incorporating the Navier-Stokes-Fourier constitutive model:

$$e = c_v\theta,$$

$$\mathbf{T} = p\mathbf{I} - \mu(\nabla\mathbf{v} + \nabla\mathbf{v}^T - \frac{2}{3}(\nabla\cdot\mathbf{v})\mathbf{I}),$$

$$\frac{d\theta}{dt} = \frac{-R}{c_v}\mathrm{tr}\left(\mathbf{A}(\mathbf{I}+t\mathbf{A})^{-1}\right)\theta(t) + \frac{\mu(\theta(t))}{\rho_0 c_v}(|\mathbf{A}(\mathbf{I}+t\mathbf{A})^{-1}|^2 + \mathrm{tr}\left((\mathbf{A}(\mathbf{I}+t\mathbf{A})^{-1})^2\right) - \frac{2}{3}(\mathrm{tr}(\mathbf{A}(\mathbf{I}+t\mathbf{A})^{-1}))^2)$$
$$\exp\left((-\int_0^t \mathrm{tr}\,\mathbf{A}(\mathbf{I}+s\mathbf{A})^{-1}\mathrm{d}s)\right), \quad (30)$$

$$\theta(0) = \theta_0.$$

5.1. General Incompressible flow

A general incompressible flow (shear in three directions) gives a family of choices of $\mathbf{A}$ out of many flows possible. Incompressibility imposes the condition, $\det(\mathbf{I}+t\mathbf{A}) = 1$, $t > 0$. The characteristic equation in t then yields the condition,

$$\det\mathbf{A} = \mathrm{tr}\,\mathbf{A} = \mathrm{tr}\,\mathbf{A}^2. \quad (31)$$

A necessary and sufficient condition for (31) is that there exists an orthonormal basis such that, in this basis,

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & \kappa \\ \gamma_1 & 0 & \gamma_3 \\ 0 & 0 & 0 \end{bmatrix}. \quad (32)$$

In general, this is matrix of rank 2. We note that there are many of these isochoric affine flows which are not viscometric flows [9]. In abstract form, $\mathbf{A} = \kappa\mathbf{e_1}\otimes\mathbf{e_3} + \mathbf{e_2}\otimes\mathbf{g}$ and $\mathbf{v}(\mathbf{x},t) = \mathbf{A}\mathbf{x} - \kappa t\gamma_1\gamma_3\mathbf{e_2}$ where $\mathbf{e_1},\mathbf{e_2},\mathbf{e_3}$ are orthonormal and $\mathbf{g} = \gamma_1\mathbf{e_1} + \gamma_3\mathbf{e_3}$.

These flows have an extra feature which is not present in other viscometric flows. Here, vorticity (independent of position) grows linearly in time given by

$$\boldsymbol{\nabla}\times\mathbf{v} = (\gamma_3 - \kappa\gamma_1 t)\mathbf{e_1} - \kappa\mathbf{e_2} - \gamma_1\mathbf{e_3}. \quad (33)$$

Conservation of mass determines the density:

$$\rho = \rho(0)\exp\left(-\int_0^t E\mathrm{d}s\right) = \rho(0)\exp\left(-\int_0^t \mathrm{tr}\mathbf{A}(\mathbf{I}+s\mathbf{A})^{-1}\mathrm{d}s\right) = \rho(0)\exp(0) = \rho(0) = \rho_0 \quad (34)$$

Conservation of energy determines the temperature. We solve the resulting ordinary differential equation using a Runge-Kutta solver for the given viscosity model,

$$\dot\theta = \frac{\mu(\theta)}{c_v\rho_0}(\gamma_1^2 + \kappa^2 + (\gamma_3 - \gamma_1\kappa t)^2) \quad (35)$$

5.2. Numerical Validation

We simulate a general incompressible flow ($\mathbf{A}$ given by (32)) for a model of Argon gas and compare the temperature evolution from continuum and atomistic calculations. Initially, the simulated atoms are defined on a domain constructed by linearly independent vectors $\mathbf{e}_1, \mathbf{e}_2$ and $\mathbf{e}_3$. The initial coordinates and momenta correspond to a specified equilibrium state of a gas. (Here, ρ=1.78 kg/m^3, $\theta(0)$ = 350 K). The initial velocity of each simulated atom is drawn from a Maxwell–Boltzmann distribution whose variance is determined by the temperature. A pairwise additive short-range Lennard-Jones potential is used to define the interaction between atoms, given by

$$\phi(r_{ij}) = 4\epsilon\left[\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right], \tag{36}$$

where r_{ij} is the distance between atoms, $\epsilon = 1.65 \times 10^{-21}$J, and $\sigma = 3.4 \times 10^{-10}$m. It is important for comparison that the input parameters of the continuum model are consistent with the force field used in MD. We use here Lennard-Jones (12-6) temperature dependent viscosity model ($\mu(\theta)$ in (35)) which was computed directly from the kinetic theory [26]. It is in agreement with the Newtonian constitutive model used here to compute the continuum temperature profile. Reduced units are used for the OMD simulations by making distance and energy dimensionless by use of the molecular diameter σ, and characteristic interaction energy ϵ, respectively. The time step varies from 0.05 to 1 femtosecond depending on the temperature of the system.

The mean fluid velocity is obtained using the mass averaged position: $\langle \mathbf{v} \rangle = \frac{\sum_{i=1}^{N} m_i \mathbf{v}_i}{\sum_{i=1}^{N} m_i}$ where N is the total number of simulated atoms; this follows the well-studied multiscale idea that one should average the momentum, not the velocity. The macroscopic temperature is computed as:

$$\theta = \frac{m}{3k}[\langle v_1^2 + v_2^2 + v_3^2 \rangle - (\langle v_1 \rangle^2 + \langle v_2 \rangle^2 + \langle v_3 \rangle^2)], \tag{37}$$

where m is a mass of an atom and k is Boltzmann's constant.

We used 1600 simulated atoms for the simulation of general-incompressible flow. We compare the evolution of temperature coming from OMD and NSF computations. For OMD computation, we use the instantaneous temperature definition given in (37) and for NSF computation, we use the ODE derived in (35) with γ_1, κ =, and γ_3 set to 0.0001(reduced units) to obtained time-dependent uniform temperature field $\theta(t)$. The density ($\rho(t) = \rho(0)$) and specific heat c_v of Argon gas are given by 1.78 kg/m^3 and 312 J/kg.K respectively. OMD and NSF agree very well as shown in the Fig. 6(a). The remapping of the fundamental domain is performed several times during the simulation due to significant deformation. We notice that there is no discontinuity in the temperature field due to remapping, as must be true. For the strain rate considered in this case, $\gamma_1 = \kappa = \gamma_3 = 0.0001$ (in reduced units), the fundamental assumptions behind the conventional linear constitutive law (Fourier's law, Navier-Stokes) works very well. This is evident from the comparison between velocity distribution function (VDF) determined from OMD and from the Chapman-Enskog method [6, 8], computed using the local moments $(\rho, \theta, \mathbf{T})$ obtained from simulations as shown in Fig. 7. The computed VDF follows the near equilibrium conditions and deviates little from the equilibrium Maxwell-Boltzmann distribution. For higher rates, we can expect that the linear constitutive law will no longer remain valid and a non linear constitutive laws will be needed. This is very well shown in the comparison (Fig. 6(b)) made for a much higher value of velocity gradient ($\gamma_1 = \kappa = \gamma_3 = 0.05$ (in reduced units)). Here, the discrepancy is quite apparent.

The velocity gradient considered in this comparison study is not constant in time. It takes some time for the simulation to attain the gradient imposed on it. We wait for the transient stage to settle before extracting any data. We noticed that one can take the advantage of basic theorem of OMD by starting with a much lower number of simulated atoms and once the system comes out of the transient regime, the number of simulated atoms can be increased to improve the statistics. This can be done by simply designating some non-simulated atoms as simulated atoms and restarting the simulation with the initial conditions given by the last time-step. The extreme sensitivity of nonlinear dynamical systems to perturbations of initial conditions in practice assures better statistics, i.e., better approximation of the invariant manifold.

In Fig. 8 we compare the gradient imposed ($\partial v_2/\partial x_1$, $\partial v_1/\partial x_3$,) and the one attained in the simulations. Fluctuations increase with time due to increase in temperature of the system. Comparison with NSF is made once the system

is out of this transient regime 6(b). Note that for a comparatively bigger system, it takes more time to attain fully developed gradient field.

OMD works well in highly non-equilibrium settings where there is no well developed constitutive equation. It can provide interesting insight on the response of the gas in that regime. An alternative is to use the Direct Simulation Monte Carlo (DSMC) (Bird [5]) method, which represents a valuable and efficient tool to investigate the nonequilibrium structure of the gas, but limited to the dilute regime and relies on a variety of collision models. A variant of the DSMC method, called Direct Molecular Simulation (DMS) [27, 38, 43] eliminates the need for collision models by relying on a set of PES, but is still limited to the dilute gas regime.

Next, we explore the effect of remapping on temperature evolution. Fig. 9(a) shows two independent OMD simulations, one where remapping is done (20 instances of remapping) and another where it is not performed. The simulation is done for comparatively higher density state (ρ = 674.3 kg/m^3) than the previous one to get many occurrences of the criterion for remapping. The temperature fields agree well with each other. Hence, remapping is only performed from the computational perspective since it makes the computation less intensive. We also look at the effect of another important parameter, number of simulated atoms, on the predictions. The method works for any number of simulated atoms, but the question is to find an appropriate lower bound so that it represents the correct physics of the system. Fig. 9(b) shows the comparison between different simulations which use various numbers of simulated atoms. Surprisingly, even 200 atoms represents the macroscopic system remarkably well in these cases and gives correct trends. The inherent length scale of the phenomenon needs to be considered when fixing the number of atoms to be simulated. For our system the size of fundamental domain needs to be bigger than mean free path of the gas to eliminate nonphysical effects. We choose 1600 atoms to be an optimum number for the simulation considered. Note that to capture much lower gradients one needs to increase the number of simulated atoms further to reduce the surrounding statistical noise of the system. Ensemble averaging by running many instances of OMD also improves statistics for applications involving macroscopically homogeneous simulations.

One can place the simulated atoms at any positions, map these to other locations using the group, simulate the equations of molecular dynamics using just this original set of atoms while calculating the forces on these from all other atoms within the cut-off. We found that, as expected, reproducible macroscopic behavior was achieved more quickly if we gave the simulated atoms initial velocities consistent on average with the macroscopic Eulerian velocity $\mathbf{v}(\mathbf{x}, t) = \mathbf{A}(\mathbf{I} + t\mathbf{A})^{-1}\mathbf{x}$.

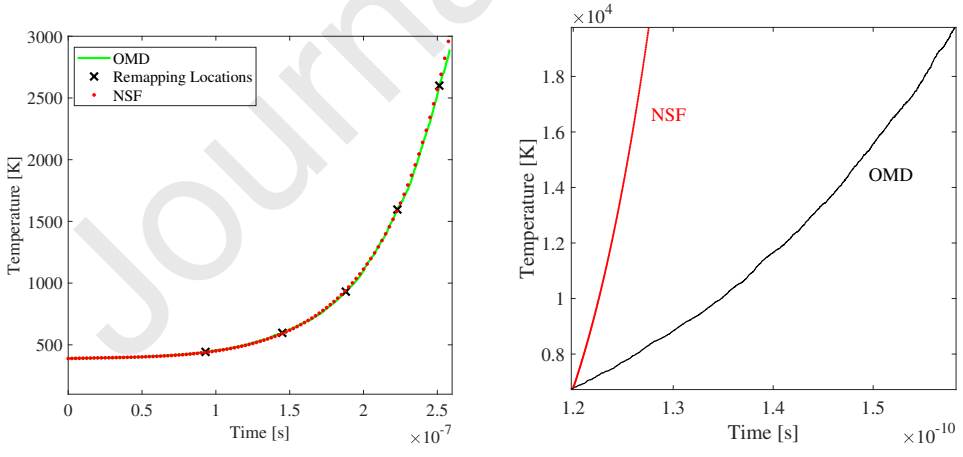


Fig. 6: Evolution of temperature for general incompressible flow of Argon gas (a) $\gamma_1 = \kappa = \gamma_3 = 0.0001$ (b) $\gamma_1 = \kappa = \gamma_3 = 0.05$.

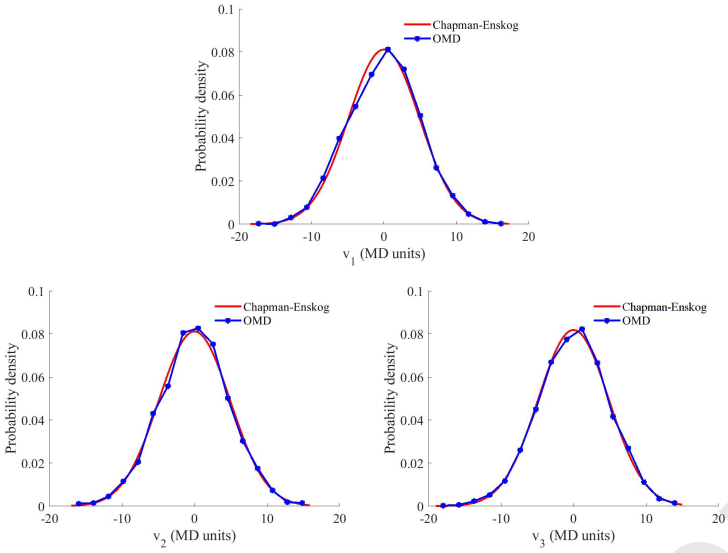


Fig. 7: Comparison of velocity distribution function for different velocity components ($\gamma_1 = \kappa = \gamma_3$ set to 0.0001).

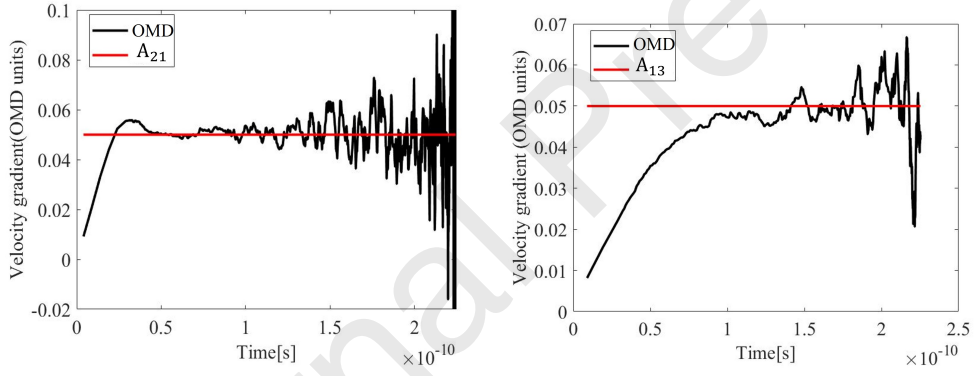


Fig. 8: Time taken by the system to adapt to the boundary conditions.

6. Phase change

There are no fundamental restrictions on the density of the fluid in the simulations, and phase change can occur spontaneously during a simulation. To illustrate the phenomenon of phase change, we perform the dilatation of the infinite system of supercritical Argon using the Lennard-Jones potential. Here, $\mathbf{A} = \kappa\mathbf{e}_1\otimes\mathbf{e}_1 + \kappa\mathbf{e}_2\otimes\mathbf{e}_2 + \kappa\mathbf{e}_3\otimes\mathbf{e}_3$, $\kappa > 0$. This flow field comes under the general nine parameter family of compressible flows where density is given by

$$\rho(t) = \frac{\rho(0)}{(\kappa t + 1)^3} \tag{38}$$

The values of initial density and dilatation rate in OMD units are chosen to be $\rho = 674.3$ kg/m^3 and $\kappa = 0.01$ (OMD units) respectively. Fig. 11(a) illustrates the phase transition driven by high rate expansion where clusters of varying sizes appear spontaneously during the simulation. Fig. 11(b) shows a zoomed view of a cluster. Note that the clusters are composed of both simulated and non-simulated (redefined as simulated atoms at this time instant) atoms. The $\rho - T$ diagram is shown in the Fig. 10. As the simulation proceeds, the temperature of the system decreases due to rapid

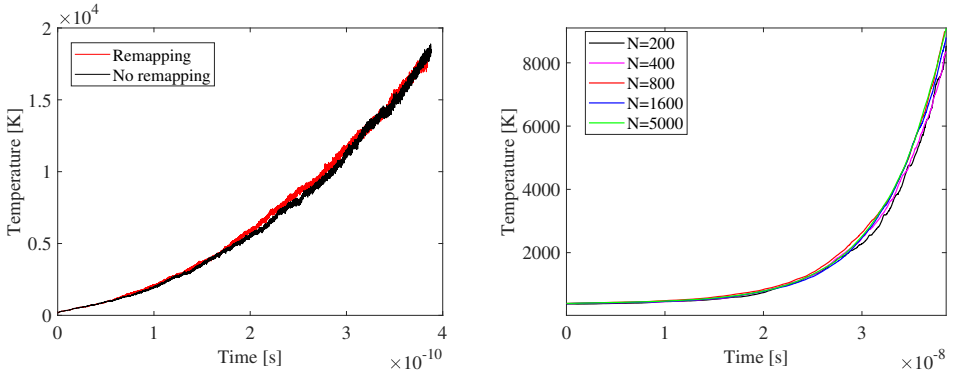


Fig. 9: Effect of (a) remapping and (b) number of simulated atoms on the temperature evolution.

adiabatic expansion.

A slight rise in the temperature due to the release of latent heat shows the onset of condensation. At this instant, the potential energy starts decreasing due to clustering of atoms. Note that, macroscopic variables (density, temperature, stress, etc.) cease to be macroscopically uniform once the two phase system is fully developed. Also, in order to examine the properties of a heterogeneous system like this, it is necessary to use many more simulated atoms than in the case of a single phase system.

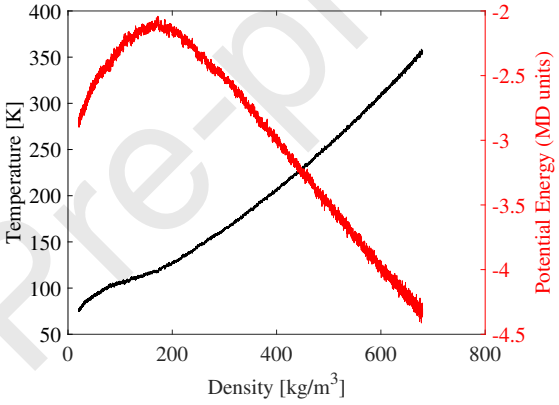


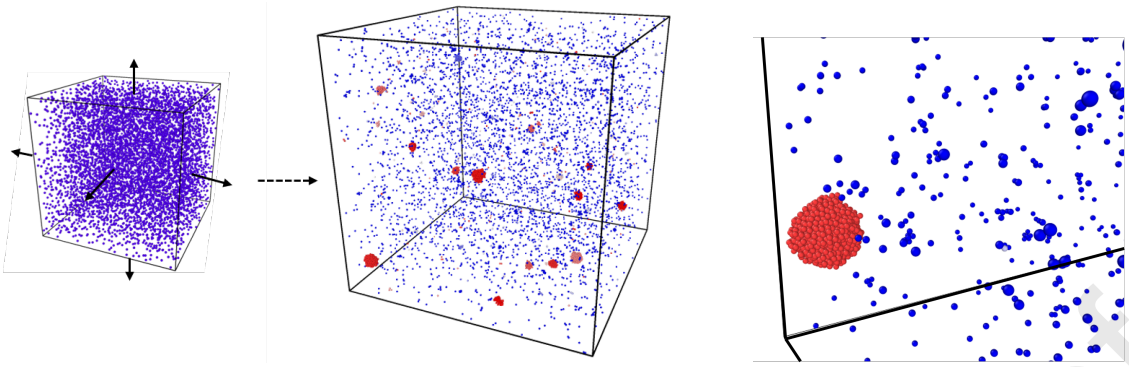
Fig. 10: $\rho - T$ diagram with evolution of the potential energy and temperature.

These studies can be highly useful in studying the actual dynamics of the birth of the new phase, exploring the actual morphology of clusters produced in the expansions and to produce benchmark results for testing various explicit nucleation models. This will appear in forthcoming work, but here we simply show that this is a feasible possibility with the OMD method as described here.

Several studies report molecular dynamics simulation of phase change in Argon using various cooling protocols. Kraska studied homogeneous nucleation of argon from a supersaturated vapor phase using a NVE ensemble [28]. Diemand et al. studied large-scale MD simulations of homogeneous vapor-to-liquid nucleation under NVT ensemble [13]. These techniques are associated with reproducing equilibrium probability density at a given external environment whereas OMD operates in highly non-equilibrium environment. Ashurst and Holian studied the expansion and fragmentation of a 3D system without free boundaries [2]. Our present system where the fundamental domain edge length grows like $L(t) = (\mathbf{I} + t\mathbf{A})L_0$ is similar to the one considered in [39] with $\mathbf{A}$ being a diagonal matrix with diagonal entries given by κ.

7. Shear-driven gas flow in nano channel

In this section OMD is used to study confined flows where walls are modeled explicitly using Lennard-Jones force field. We study flow of Lennard-Jones argon subjected to boundary driven shear where the gas is confined between two thin face-centered cubic infinite walls a distance H apart. Fig. 12 illustrates the fundamental domain composed of simulated atoms which only include three layer atomically thin lower wall (red atoms). The other wall is composed of non-simulated atoms and moves with a net velocity in $\mathbf{e}_1$ direction is not shown. Wall atoms



(a) (b)

Fig. 11: Phenomenon of phase change showing (a) sudden dilatation of box leading to cluster (red) formation and (b) Zoomed view of a cluster.

have lattice parameter, mass and diameter equivalent to platinum ($a_p = 0.392$ nm, $m_w = 3.2398$ x 10^{-22} kg, $\sigma_w = 2.4626$ x 10^{-10} m) and depth of potential well is $\epsilon_w = 31.36$ kJ/mol. For the cross interactions between walls and gas, the length parameter is determined from the Lorentz-Berthelot mixing rule [41] $\sigma_c = (\sigma_w + \sigma_{Ar})/2$ and energy parameter $\epsilon_c = 0.6580$ kJ/mol is taken from the literature [39]. We emphasize that both the wall and gas are treated in one simulation with the given **A**, all atoms satisfying the MD equations; the platinum atoms happen to move macroscopically as a block because they are strongly bonded. The initial system is equilibrated to a temperature of 300K and then shear motion is induced by choosing $\mathbf{A} = \kappa \mathbf{e}_1 \otimes \mathbf{e}_2$ where κ is of the order $10^9 s^{-1}$. To reduce time in achieving the fully developed flow, a macroscopic constant velocity gradient κ is imposed on the initial velocity field.

The velocity profile of the gas in the channel as a function of non-dimensional channel height is plotted in Fig. 13(a). Near the walls, we see the formation of a Knudsen layer of thickness of the order of few mean free paths ($\lambda = 16$ nm). In the middle of the channel the velocity varies linearly, whereas in Knudsen layer the velocity gradient differs significantly from that of the mainstream. In this region, the paths of gas molecules are severely affected by the presence of solid walls resulting in significant amount of velocity slip v_s. The existence of velocity slip was first predicted by Maxwell [25].

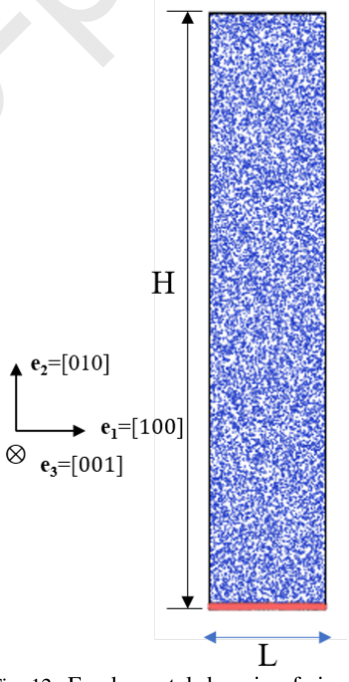


Fig. 12: Fundamental domain of simulated atoms (Wall: Orange atoms, Gas: Blue atoms)

The quantity v_s is computed by averaging the velocity of atoms lying in a bin adjacent to the wall, minus the velocity of wall itself; for the lower wall this velocity is zero. Different sizes of computational domain LxWxH and density of the gas are chosen to simulate the flow at varying Knudsen number Kn = λ/H, where λ is defined in terms of viscosity using Cercignani's definition [7], and $\lambda = \frac{\mu}{p}\frac{\pi m}{2k_bT}$. For the viscosity model we use the Newtonian viscosity which was computed for the Lennard-Jones force field using the Chapman-Enskog expansion of the velocity distribution function [26]. The simulation details is listed in Table 1. κ is varied for different simulations to maintain a constant Mach number Ma = $\frac{v_w}{\sqrt{\gamma k_b T/m}}$ of ≈ 1.5 with varying Knudsen number. Here v_w is the wall velocity given by $v_w = \kappa$H.

The Knudsen number characterizing these flows is in the transition regime. For sufficiently small Knudsen num-

Table 1: Summary of different sizes of nano channel considered

H (nm)	ρ (kg/m^3)	κ (ps^{-1})
150	8.32	0.0046
150	8.07	0.0046
150	8.0	0.0046
150	5.0	0.0046
100	5.0	0.0069
100	2.0	0.0069

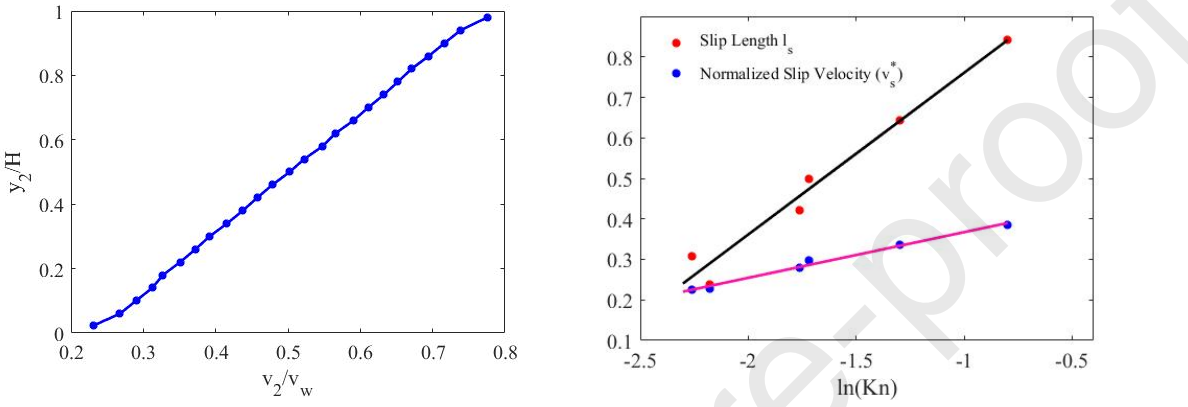


Fig. 13: Slip flow over smooth platinum surface (a) Normalized velocity profile (b) Variation of slip length and normalized slip velocity with the Knudsen number.

bers Maxwell's theory of slip accurately predicts a non-dimensional slip length for isothermal flow given by

$$l_s^* = \frac{v_s}{\mathrm{H}(\mathrm{dv_2}/\mathrm{d_{y_2}})|_\mathrm{w}} = \frac{2-\sigma}{\sigma}\mathrm{Kn}$$

where σ is the tangential momentum accommodation coefficient (TMAC).

The interesting question is how the slip velocity behaves in the transition regime. Fig. 13(b) shows the OMD prediction of slip velocity normalized by wall velocity $v_s^* = v_s/v_w$ and normalized slip length l_s^* as a function of Knudsen number. It can be seen from the plot that the normalized slip velocity and slip length vary as the log of the Knudsen number, when the Newtonian viscosity and Cercignani's mean free path definition is used. It therefore deviates from the Maxwell first order slip model which predicts a linear dependence. A similar finding for the hard sphere definition of mean free path $\lambda = \frac{m}{\sqrt{2}\pi\rho d^2}$ was predicted by Bhattacharya et al [4]. This logarithmic dependence can prove to be important in proposing new simple slip models which only depend on average temperature, density and Newtonian viscosity and which could work well also in the transition regime. This result provides motivation to validate this simple slip model for a wide range of flow system geometries and regime of (Kn, Ma). Another approach is to improve the prediction of mean free path [15] and viscosity which goes into the slip model. It was shown in prior work that when an effective viscosity is obtained from the shear stress at a thermal wall, then the Maxwell model works reasonably well, even in the transient regime [33]. On the other hand there exist higher order velocity slip boundary conditions which are shown to improve the flow field predictions in some situations of interest. A review can be found in Reese and Zhang [37].

8. Summary

In summary, this work details computational aspects of a novel objective molecular dynamics (OMD) method for fluids. The method can simulate three parameter family of general incompressible and nine parameter family of compressible flows at the atomistic scale. The framework developed is quite easy to implement. One only needs to provide a 3×3 matrix $\mathbf{A}$ to consider different flows, and every atom in the infinite system satisfies the equations of molecular dynamics exactly for its forces. We also prove this in a discrete sense by showing the consistency of OMD with the velocity Verlet algorithm widely used in MD simulations. The trend of the temperature field for general incompressible flows is in good agreement with linear continuum theory in the regime of small velocity gradients where conventional hydrodynamic closure for the transport fluxes (Fourier's law, Newton's law, etc.) remains valid. In the regime of higher rates, the Navier-Stokes-Fourier theory is no longer accurate and there is a need to develop new constitutive equations. Atomistic methods can very well cater to this growing need since it rests on a Potential Energy Surface (PES) that can be produced by first principles computational chemistry calculations under the Born-Oppenheimer approximation. Additionally, with the availability of the framework of simulating different flows in OMD, the method is very well suited to test the validity of constitutive models under various flows [34]. OMD solutions are possible in any material and fluid with an arbitrary number of simulated atoms. We also report in this work how OMD also provides a structure for simulating non homogeneous phenomena of phase transition and boundary driven shear flow. It can also deal with other phenomena of chemical reaction, electronic excitation and shock wave dynamics quite easily and accurately.

Acknowledgments

G. Pahlani and R. D. James acknowledge funding from the Multidisciplinary University Research Initiative (MURI) under Grant No. FA9550-18-1-0095 and a Vannevar Bush Faculty Fellowship. T. E. Schwartzentruber acknowledges funding from NASA under Grant 80NSSC20K1061.

References

[1] Michael P Allen and Dominic J Tildesley. Computer simulation of liquids. Oxford university press, 2017.
[2] Wm T Ashurst and Brad Lee Holian. Droplet formation by rapid expansion of a liquid. Physical Review E, 59(6):6742, 1999.
[3] András Baranyai and Peter T Cummings. Steady state simulation of planar elongation flow by nonequilibrium molecular dynamics. The Journal of chemical physics, 110(1):42–45, 1999.
[4] DK Bhattacharya and GC Lie. Nonequilibrium gas flow in the transition regime: a molecular-dynamics study. Physical Review A, 43(2):761, 1991.
[5] Graeme A Bird. Molecular gas dynamics and the direct simulation of gas flows. Molecular gas dynamics and the direct simulation of gas flows, 1994.
[6] Iain D Boyd and Thomas E Schwartzentruber. Nonequilibrium gas dynamics and molecular simulation, volume 42. Cambridge University Press, 2017.
[7] Carlo Cercignani et al. Mathematical methods in kinetic theory, volume 1. Springer, 1969.
[8] Sydney Chapman and Thomas George Cowling. The mathematical theory of non-uniform gases: an account of the kinetic theory of viscosity, thermal conduction and diffusion in gases. Cambridge university press, 1990.
[9] Bernard D Coleman, Hershel Markovitz, and Walter Noll. Viscometric flows of non-Newtonian fluids: theory and experiment, volume 5. Springer Science & Business Media, 2012.
[10] Kaushik Dayal and Richard D James. Nonequilibrium molecular dynamics for bulk materials and nanostructures. Journal of the Mechanics and Physics of Solids, 58(2):145–163, 2010.
[11] Kaushik Dayal and Richard D James. Design of viscometers corresponding to a universal molecular simulation method. Journal of Fluid Mechanics, 691:461–486, 2012.
[12] Xinyue Deng. An introduction to lenstra-lenstra-lovasz lattice basis reduction algorithm, 2016.
[13] Jürg Diemand, Raymond Angélil, Kyoko K Tanaka, and Hidekazu Tanaka. Large scale molecular dynamics simulations of homogeneous nucleation. The Journal of chemical physics, 139(7):074309, 2013.
[14] Matthew Dobson, Ian Fox, and Alexandra Saracino. Cell list algorithms for nonequilibrium molecular dynamics. Journal of Computational Physics, 315:211–220, 2016.
[15] Nishanth Dongari, Yonghao Zhang, and Jason M Reese. The importance of mean free path in determining gas micro flow behaviour. In International Conference on Nanochannels, Microchannels, and Minichannels, volume 54501, pages 481–490, 2010.
[16] Traian Dumitrică and Richard D James. Objective molecular dynamics. Journal of the Mechanics and Physics of Solids, 55(10):2206–2236, 2007.
[17] Federico Frascoli, BD Todd, and Debra J Searles. Boundary condition independence of molecular dynamics simulations of planar elongational flow. Physical Review E, 75(6):066702, 2007.
[18] Daan Frenkel and Berend Smit. Understanding molecular simulation: from algorithms to applications, volume 1. Elsevier, 2001.
[19] Steven D Galbraith. Mathematics of public key cryptography. Cambridge University Press, 2012.

[20] William G Hoover, Denis J Evans, Richard B Hickman, Anthony JC Ladd, William T Ashurst, and Bill Moran. Lennard-jones triple-point bulk and shear viscosities. green-kubo theory, hamiltonian mechanics, and nonequilibrium molecular dynamics. Physical Review A, 22(4):1690, 1980.
[21] Thomas A Hunt, Stefano Bernardi, and BD Todd. A new algorithm for extended nonequilibrium molecular dynamics simulations of mixed flow. The Journal of chemical physics, 133(15):154116, 2010.
[22] Richard D James, Alessia Nota, and Juan JL Velázquez. Long-time asymptotics for homoenergetic solutions of the boltzmann equation: collision-dominated case. Journal of Nonlinear Science, 29(5):1943–1973, 2019.
[23] Richard D James, Alessia Nota, and Juan JL Velázquez. Self-similar profiles for homoenergetic solutions of the boltzmann equation: particle velocity distribution and entropy. Archive for Rational Mechanics and Analysis, 231(2):787–843, 2019.
[24] Richard D James, Alessia Nota, and Juan JL Velázquez. Long time asymptotics for homoenergetic solutions of the boltzmann equation. hyperbolic-dominated case. Nonlinearity, 33(8):3781, 2020.
[25] Earle H Kennard et al. Kinetic theory of gases, volume 483. McGraw-hill New York, 1938.
[26] Sun Ung Kim and Charles W Monroe. High-accuracy calculations of sixteen collision integrals for lennard-jones (12–6) gases and their interpolation to parameterize neon, argon, and krypton. Journal of Computational Physics, 273:358–373, 2014.
[27] Katsuhisa Koura. 4 carlo direct simulation of rotational relaxation of diatomic molecules using classical trajectory calculations: Nitrogen shock wave. Physics of Fluids, 9(11):3543–3549, 1997.
[28] T Kraska. Molecular-dynamics simulation of argon nucleation from supersaturated vapor in the nve ensemble. The Journal of chemical physics, 124(5):054507, 2006.
[29] AM Kraynik and DA Reinelt. Extensional motions of spatially periodic lattices. International journal of multiphase flow, 18(6):1045–1059, 1992.
[30] AW Lees and SF Edwards. The computer study of transport processes under extreme conditions. Journal of Physics C: Solid State Physics, 5(15):1921, 1972.
[31] Arjen K Lenstra, Hendrik Willem Lenstra, and László Lovász. Factoring polynomials with rational coefficients. Mathematische annalen, 261(ARTICLE):515–534, 1982.
[32] Ronald E Miller, Ellad B Tadmor, Joshua S Gibson, Noam Bernstein, and Fabio Pavia. Molecular dynamics at constant cauchy stress. The Journal of chemical physics, 144(18):184107, 2016.
[33] David L Morris, Lawrence Hannon, and Alejandro L Garcia. Slip length in a dilute gas. Physical review A, 46(8):5279, 1992.
[34] Gunjan Pahlani, Thomas E Schwartzentruber, and Richard James. Investigation of the breakdown of navier-stokes equation using objective molecular dynamics. In AIAA SCITECH 2022 Forum, page 1012, 2022.
[35] Gunjan Pahlani, Erik Torres, Thomas E Schwartzentruber, and Richard D James. Objective molecular dynamics of dissociating nitrogen under high temperature conditions. In AIAA Scitech 2021 Forum, page 0707, 2021.
[36] Michele Parrinello and Aneesur Rahman. Crystal structure and pair potentials: A molecular-dynamics study. Physical review letters, 45(14):1196, 1980.
[37] Jason M Reese and Yonghao Zhang. Simulating fluid flows in micro and nano devices: the challenge of non-equilibrium behaviour. Journal of Computational and Theoretical Nanoscience, 6(10):2061–2074, 2009.
[38] Thomas E Schwartzentruber, Maninder S Grover, and Paolo Valentini. Direct molecular simulation of nonequilibrium dilute gases. Journal of Thermophysics and Heat Transfer, 32(4):892–903, 2018.
[39] Peter Spijker, Albert J Markvoort, Silvia V Nedea, and Peter AJ Hilbers. Computation of accommodation coefficients and the use of velocity correlation profiles in molecular dynamics simulations. Physical Review E, 81(1):011203, 2010.
[40] Alexander Stukowski. Visualization and analysis of atomistic simulation data with ovito–the open visualization tool. Modelling and simulation in materials science and engineering, 18(1):015012, 2009.
[41] BD Todd and Peter J Daivis. A new algorithm for unrestricted duration nonequilibrium molecular dynamics simulations of planar elongational flow. Computer physics communications, 117(3):191–199, 1999.
[42] BD Todd and Peter J Daivis. Homogeneous non-equilibrium molecular dynamics simulations of viscous flow: techniques and applications. Molecular Simulation, 33(3):189–229, 2007.
[43] Erik Torres and Thomas E Schwartzentruber. Direct molecular simulation of oxygen dissociation across normal shocks. Theoretical and Computational Fluid Dynamics, 36(1):41–80, 2022.

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Author Contributions: Conceptualization, GP and RDJ; theory, GP, TES and RDJ; simulation, GP; writing, original draft preparation, GP; writing, review and editing, GP, TES and RDJ; funding acquisition, RDJ, TES. All authors have read and agreed to the published version of the manuscript.