



A quantitative study on the approximation error and speed-up of the multi-scale MCMC (Monte Carlo Markov chain) method for molecular dynamics



Jie Liu^a, Qinglin Tang^b, Jisheng Kou^c, Dingguo Xu^d, Tao Zhang^{a,*}, Shuyu Sun^{a,*}

^a Physical Science and Engineering Division (PSE), Computational Transport Phenomena Laboratory, King Abdullah University of Science and Technology (KAUST), Thuwal 23955-6900, Saudi Arabia

^b School of Mathematics, Sichuan University, Chengdu 610064, China

^c School of Civil Engineering, Shaoxing University, Shaoxing 312000, Zhejiang, China

^d College of Chemistry, Sichuan University, Chengdu 610064, China

ARTICLE INFO

Article history:

Received 26 May 2022

Received in revised form 10 July 2022

Accepted 15 July 2022

Available online 21 July 2022

Keywords:

Molecular dynamics

Monte Carlo method

Coarsening

Multi-scale method

ABSTRACT

The past two decades have borne remarkable progress in the application of the molecular dynamics method in a number of engineering problems. However, the computational efficiency is limited by the massive-atoms system, and the study of rare dynamically-relevant events is challenging at the timescale of molecular dynamics. In this work, a multi-scale molecular simulation algorithm is proposed with a novel toy model that can mimic the state transitions in extensive scenarios. The algorithm consists of two scales, including producing the realistic particle trajectory and probability transition matrix in the molecular dynamics scale and calculating the state distribution and residence time in the Monte Carlo scale. A new state definition is proposed to take the velocity direction into consideration, and different coarsening models are established in the spatial and time scales. The accuracy, efficiency, and robustness of our proposed multi-scale method have been validated, and the general applicability is also demonstrated by explaining two practical applications in the shale gas adsorption and protein folding problems respectively.

© 2022 Elsevier Inc. All rights reserved.

1. Introduction

In the past decades, the molecular dynamics (MD) simulation method has been successfully applied in a number of engineering problems thanks to the development of a full-atom model and accurate force field data [1,2]. For example, the special mechanisms in unconventional reservoirs can be described well by using MD simulations [3], especially the dynamic sorption in the extremely tight rocks that can be captured precisely [4]. In addition, the protein folding problem, one of the major challenges in our understanding of the transformation of biological information in our body [5], has been accurately simulated by using the MD method to calculate the interaction of protein molecules [6], which is usually hard to be obtained in the laboratory experiments [7].

* Corresponding authors.

E-mail addresses: tao.zhang.1@kaust.edu.sa (T. Zhang), shuyu.sun@kaust.edu.sa (S. Sun).

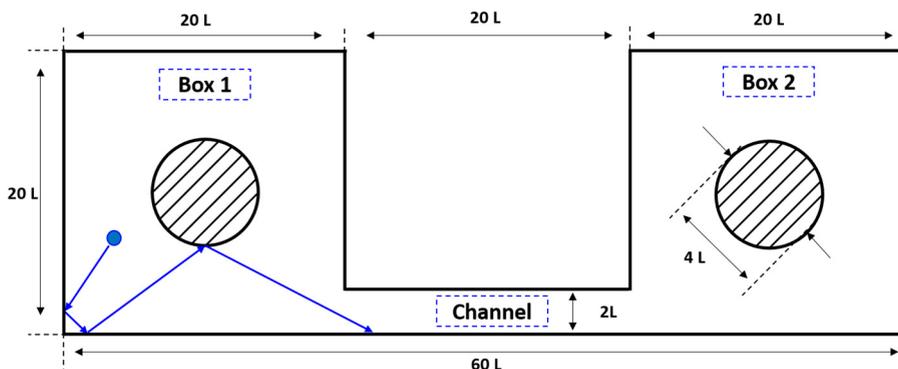


Fig. 1. The toy model of the MD. The blue ball denotes the particle, the blue arrow lines denote the particle trajectory, and the solid slash lines denote the round areas that cannot be penetrated. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

The classical MD method based on Newtonian mechanics is developed to simulate the complex molecular behaviors and get accurate state transitions in the phase space [8–11]. However, due to the accurate description of all atoms, such a classical MD method has two major problems. The first one is that the number of ordinary differential equations (ODE) is huge, thus more computational resources and memory load are needed, especially in the study of protein macromolecules [12]. The second problem is that not all atomistic-scale events are crucial to the dynamics, where the crucial events are termed as dynamically-relevant events [13,14], and it is challenging to study the rare dynamically-relevant events at the MD timescale [15,16]. Thus, such efforts can be saved if proper techniques are involved to directly obtain the dynamically-relevant events without traversing all the atomistic-scale events.

The Monte Carlo (MC) method was proposed in the 1940s for the development of nuclear weapons [17,18], and has been developed to be a better choice to handle state transition problems. However, an accurate probability distribution function (PDF) is often needed to keep consistency with the conventional MD results [19]. In general, the accurate molecular trajectory can be provided by MD simulations and the probability transition matrix can be generated accordingly [20]. Based on that, the state transition can be recreated efficiently by using the Markov chain scheme in the MC method [21], and good consistency with the MD results can be ensured as long as the transition matrix retains most of the information of the trajectory in the phase space.

Generally speaking, higher accuracy can be realized if more specific states are taken into consideration, for example, when the velocity is adopted to show the effect on state decompositions [22]. However, the transition matrix may be enlarged if the trajectory is decomposed into too many states [23], and similar problems may occur subsequently in MD simulations. To further enhance the computational efficiency, the coarsening MC method is developed in both spatial and time scales. The direct method is spatial coarsening to reduce the state number, which can be calculated in a coarser mesh in the physical space [24]. Besides, the time coarsening is achieved by merging several transition steps into one step [25,26].

In this work, a multi-scale molecular simulation method is proposed, and a novel toy model is designed to examine the algorithm's performance. The exact solution MD method is adopted to produce the accurate trajectory in phase space and the probability transition matrix can be created accordingly. The accuracy of the proposed multi-scale method is verified on the residence time and state distributions, while the improved efficiency of the coarsening models is validated by proper comparisons. The remainder of this paper is organized as follows. The multi-scale method is introduced in Section 2, with a clear description of the novel toy model. Scheme validation is presented in Section 3, followed by several numerical experiments to show the new state definition and coarsening methods. Two practical cases, i.e., the dynamic sorption in shale gas reservoirs and the protein folding problems, are explained on the potential applications of our algorithm and model to demonstrate the general adaptability. Concluding remarks are summarized at the end.

2. Methodology

2.1. Molecular dynamics simulation method

To construct a toy model to discuss the state transitions, as shown in Fig. 1, three boxes are designed to represent three states. Regions of box1 and box2 mimic the stable large-scale states, and the channel plays a transitional role between the states of box1 and box2. The sizes of box1, box2, and channel are $20L \times 20L$, $20L \times 20L$, and $20L \times 2L$, respectively, where L denotes the representative unit length in practical applications. The small blue ball denotes the initial particle, the solid slash lines denote the circle areas that cannot be penetrated, the blue solid lines denote the particle trajectory, and the blue arrows denote the velocity directions in each free path. Two round boundaries, with the radius of $2L$, are set in the center of box1 and box2 respectively, making the particle trajectory chaotic [27,28]. In the conventional MD simulation, at least three atoms are needed to generate the chaotic trajectory [29], but it also brings complexity to the implementation. In this study, to construct a toy MD model as simple as possible, only one particle, without considering its

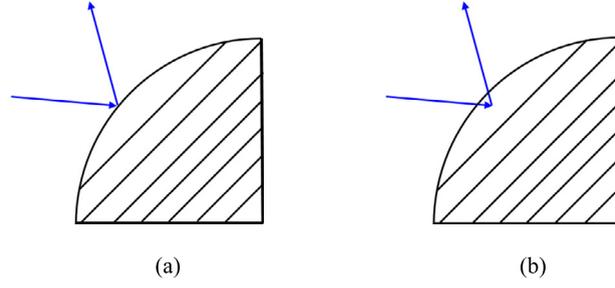


Fig. 2. The schematic diagram of wall rebound with (a) the exact solution MD algorithm and (b) the classical MD algorithm.

radius, is used to create the MD trajectory, and the interaction force is not involved, which can benefit further researches in wider scenarios. Energy dissipation is not adopted in the collision and reflection of this particle, suggesting that the change of the potential energy is the same as the change of the kinetic energy in this system. The general Newton's formulas of motion are formulated as follows,

$$\begin{pmatrix} x^{k+1} \\ y^{k+1} \end{pmatrix} = \begin{pmatrix} x^k + u\Delta t + 0.5a_x\Delta t^2 \\ y^k + v\Delta t + 0.5a_y\Delta t^2 \end{pmatrix} \quad (1)$$

where u and v are the velocity in x and y directions respectively, a_x and a_y the acceleration in x and y directions respectively, k and $k+1$ the k step and the $k+1$ step, and Δt the time step.

As the interaction force is not taken into consideration, the time step needs to be small enough to avoid crossing the wall boundary, which may damage the accuracy of MD simulations. Different from the classical MD algorithms, the exact solution MD method is used in our work to calculate the trajectory changing with time. As shown in Fig. 2 (a), the exact solution method can avoid the invasion of the particle at the wall boundary, and the accurate rebound position can be captured. On the contrary, the trajectory generated by the classical MD method may result in penetration inside the wall boundary in Fig. 2 (b), and the particle needs to be pushed back manually. The exact solution of motion can be formulated as,

$$\begin{pmatrix} x^{k+1} \\ y^{k+1} \end{pmatrix} = \begin{pmatrix} x^k + ut_{exact} + 0.5a_x t_{exact}^2 \\ y^k + vt_{exact} + 0.5a_y t_{exact}^2 \end{pmatrix} \quad (2)$$

where t_{exact} denotes the exact solution of time. Equation (2) can be solved with the boundary conditions determined in equations (3)-(5),

$$(x^{k+1} - x_{sphere})^2 + (y^{k+1} - y_{sphere})^2 = R^2 \quad (3)$$

$$x^{k+1} = X \quad (4)$$

$$y^{k+1} = Y \quad (5)$$

where x_{sphere} and y_{sphere} are the locations of the ball center, the R the radius of the non-flow boundaries, and X and Y the boundary values in x -direction and y -direction respectively. After that, the initial location and interaction location at the boundary for one free path can be obtained, and the Newtonian trajectory can be sampled with Δt_{MD} . In this way, the accurate trajectory can still be obtained without a strict requirement on the very small time steps.

2.2. Monte Carlo simulation method

The MC method has been regarded as an effective tool to handle state transition problems [30], such as protein macromolecule folding [31]. In our toy model, the computational domain is decomposed into non-overlapped mesh regions, which correspond to the particle states. According to the MD results, the movement trajectory is produced, which can be used to generate the Markov chain. Consequently, it satisfies the Markov property: the probability in any state is only relevant to the previous state, and is irrelevant to other states [32]. By counting and normalizing the MD trajectory, the transition matrix can be obtained with a transition sample step Δt_{TM} satisfying,

$$\Delta t_{MD} < \Delta t_{TM} < \Delta t_{free} \quad (6)$$

where Δt_{MD} denotes the sampling timestep in the MD simulation, and Δt_{free} is the free path time, which is computed between step k and step $k+1$.

2.3. Coarsening

Although the MD method can generate the accurate trajectory and state distributions, it costs a huge amount of computational resources. To address this issue, the coarsening method targeting the MD and MC methods can be developed on the spatial and time scales. The details of the spatial coarsening and time coarsening are presented in the following.

2.3.1. Spatial coarsening

The spatial coarsening is carried out by merging the similar states in the transition matrix, e.g., merging state i_1 and state i_2 into state \bar{i} . According to the property of the probability transition matrix, each column summation equals one,

$$\sum_{i=1}^m P_{i,j} = 1 \tag{7}$$

where P is the probability transition matrix with the size of $m \times m$, and m is the total state number,

$$P = \begin{pmatrix} P_{1,1} & \cdots & P_{m,1} \\ \vdots & \ddots & \vdots \\ P_{1,m} & \cdots & P_{m,m} \end{pmatrix} \tag{8}$$

where the $P_{1,1}$ denotes the probability that the transition from state 1 to state 1, and $P_{m,m}$ denotes the probability that the transition from the state m to the state m .

Using the probability transition matrix P , the next state i_{t+t_M} can be predicted based on the state j_t , and $P_{j_t \rightarrow i_{t+t_M}}$ is used to denote the corresponding element in the transition matrix, which can also be written as $P_{i,j}$ in equation (9). Firstly, the states i_1 and i_2 can be merged by corresponding rows in the matrix. In general, in the probability transition matrix, two rows are merged into one row,

$$P_{i,j}^{New} = P_{i_1,j}^{Old} + P_{i_2,j}^{Old} \tag{9}$$

where P^{Old} denotes the transition matrix before coarsening, and P^{New} denotes the transition matrix after coarsening, in which the size is changed to $(m - 1) \times m$. For example, if we want to merge the second and third states in a 3×3 matrix, it can be obtained as follows,

$$P^{New} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix} P^{Old} \tag{10}$$

where the auxiliary matrix $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix}$ is denoted as M_1 . Similarly, the state j_1 and state j_2 also need to be merged by columns in the matrix, which can be represented as the following equation,

$$P_{i,\bar{j}}^{New} = \frac{q_{j_1}}{q_{j_1} + q_{j_2}} P_{i,j_1}^{Old} + \frac{q_{j_2}}{q_{j_1} + q_{j_2}} P_{i,j_2}^{Old} \tag{11}$$

where q is the summation of each row in the probability transition matrix, and q can be calculated by equation (12),

$$P \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} = \begin{bmatrix} q_1 \\ \vdots \\ q_m \end{bmatrix} \tag{12}$$

where $\begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$ is an $m \times 1$ unit column vector. Here, an example of how to merge the columns in a 3×3 matrix is given,

$$P^{New} = P^{Old} \begin{pmatrix} 1 & 0 \\ 0 & q_2/(q_2 + q_3) \\ 0 & q_3/(q_2 + q_3) \end{pmatrix} \tag{13}$$

where the auxiliary matrix is denoted as M_2 . Finally, the merged probability transition matrix can be generated in equation (14),

$$P^{New} = M_1 (P^{Old} M_2) = (M_1 P^{Old}) M_2 \tag{14}$$

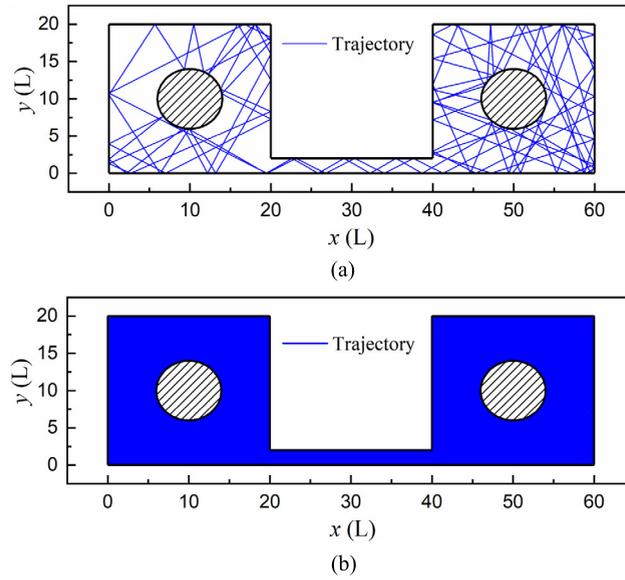


Fig. 3. (a) The MD trajectory after 100 collision counts, and (b) the trajectory after 10000 collision counts.

2.3.2. Time coarsening

In the spatial scale, the sampling is processed as one MC step. In the time scale, the states in the next k MC steps can be predicted at one time. In this case, the probability transition matrix turns out to be,

$$P^{New} = (P^{Old})^k \quad (15)$$

In fact, the time coarsening and spatial coarsening are usually combined for a faster MC calculation, and this is also one specific form of the Chapman-Kolmogorov equation [32,33].

3. Results and discussions

The multi-scale method developed above will be validated in this section, and further applied in practical engineering examples. The algorithm reliabilities are validated in Section 3.1 by simulating the meaningful trajectory and state properties. A new state definition method is developed in Section 3.2 to take the velocity effect into consideration. In Section 3.3, the coarsening method is applied to further accelerate the computation, and two cases are given in Section 3.4.

3.1. Validation of the multi-scale method

3.1.1. Validation in the MD scale

In the toy model, one particle is set in a random position beyond the non-flow area, with a random velocity less than $10 \text{ L}\cdot\text{T}^{-1}$. As shown in Fig. 3 (a), in the short-time MD simulation, the trajectory cannot cover the entire computational domain. However, in the long-time simulation, Fig. 3 (b) presents that the MD trajectory covers almost every point in the model, suggesting the validation of the trajectory produced by one particle.

Fig. 4 illustrates the free path time versus the collision count. The dramatic fluctuations can be observed, which indicates that it is a chaotic movement. Due to the regular-shape channel region, the time solution in the channel is always constant, which induces the free path time constantly in one path. Moreover, the free path time in the channel is various if the velocity angle is changed in the next time when the particle crosses the channel. The upper limit value of free path time is controlled by the model's size and the particle's velocity. As a result, the free path time is always fluctuant between zero and upper limit values, and the chaotic fluctuation also verifies the accuracy of the trajectory.

Fig. 5 (a) and (b) depict the free path time distributions in box1 and box2, respectively. Two peaks can be captured in two histograms, and the calculated free path time is mainly located between 2 T and 4 T . To understand the effect of movement types on the free path time, the free path time counted with the collision on the circle boundary is also plotted as red histograms in Fig. 5. It can be referred that the free path time with boundary collisions occupies most of the counts around the first peak, suggesting that the trajectory is mainly related to the circle boundary, which means the starting point or collision point is on the circle wall. To explain the free path time distribution more clearly, Fig. 6 (a) and (b) are depicted. Three different movement types are defined as A, B, and C, denoting the particle movement near the domain corner, collision with the circle boundary, and collision between two domain boundaries, respectively. The counting result is reasonable, as the A-type free path time should be smaller as the path is shorter, B-type free path time takes the dominant

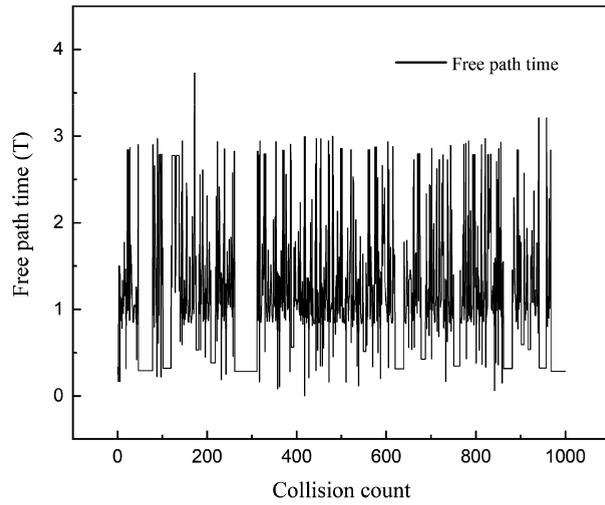


Fig. 4. The free path time with various collision counts.

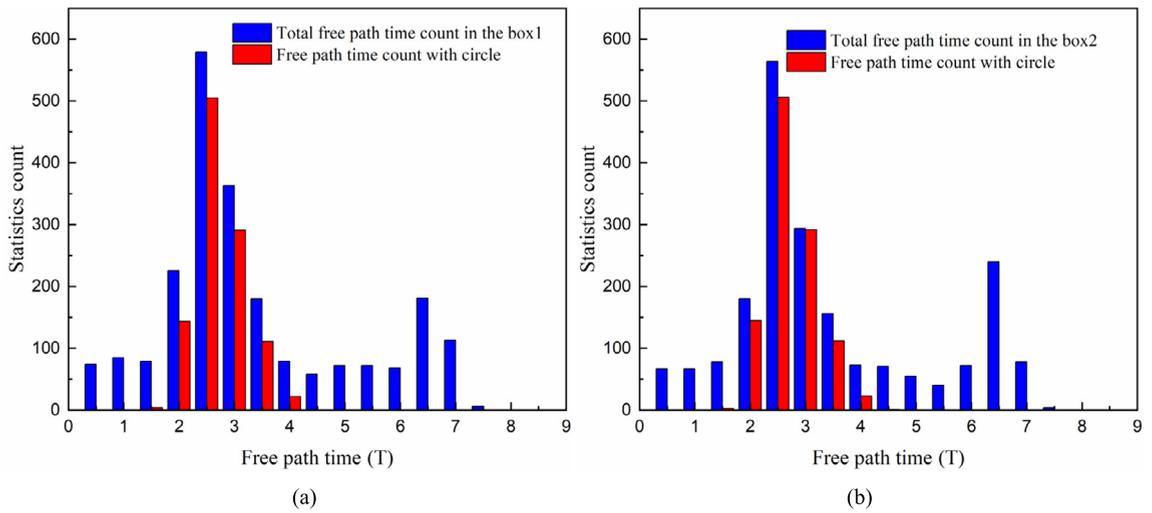


Fig. 5. (a) The free path time count that in box1. (b) The free path time counts that in box2.

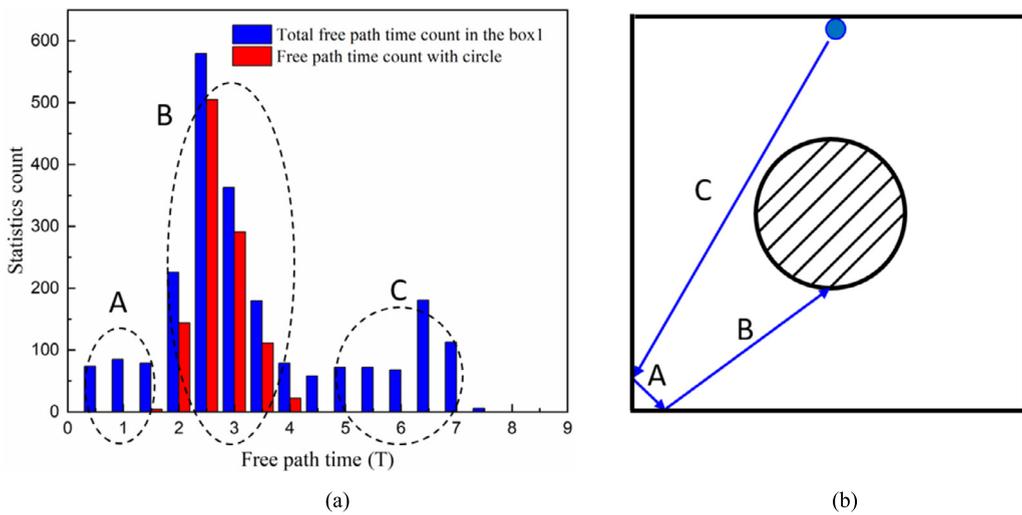


Fig. 6. (a) The different types of free path time that in box1. (b) The schematic diagram for different types of the free path.

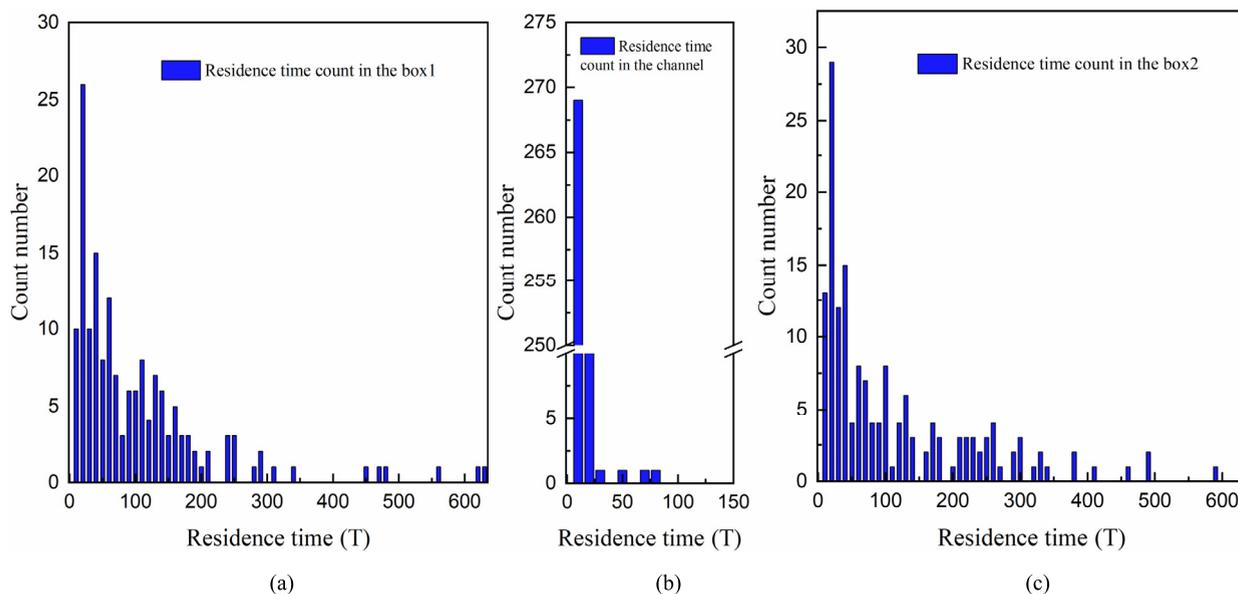


Fig. 7. The residence time produced by the MD in the (a) box1, (b) channel, and (c) box2.

counts, and C-type free path time concentrates near the second peak. In fact, free path time represents the fine state in the macro state (box1 and box2), and the accuracy of free path time can ensure the reliability of macro state transitions.

The particle makes state transitions between two boxes, and the residence time is the total time that the particle stays in the box before transition. As shown in Fig. 7 (a) and (c), the count number is relatively small at short residence times, because the particle always stays in the box for a while before leaving this box. The most frequent residence time is 20 T, and the general trend is that the count number declines with the increase of residence time. Because the longer residence time means a lower probability that the particle makes state transitions between two boxes. Therefore, the frequency is lower at longer residence time. In the long residence time, for example, around 600 T, the particle can still be trapped in the box. Since the channel is the only routine that must be passed when the particle takes a transition between the state box1 and state box2, the count number in the channel is concentrated at short residence times, as shown in Fig. 7 (b).

3.1.2. Validation in the MC scale

For the MD simulation, the trajectory points are counted and indexed on corresponding states, which are generated by meshing the computational domain, and each mesh is assumed to be one state. Totally, there are 840 states in this model, and the state distribution calculated by the MD method is shown in Fig. 8 (a). In Fig. 8 (b), by using the MC method, the state distribution can be reproduced with the probability transition matrix created by the MD trajectory. To compare the MD and MC results clearly, Fig. 8 (c) is plotted with the state count number, and the result exhibits a good agreement.

The states of box1 and box2 are connected by the channel, thus the channel size affects the transition probability between box1 and box2. Fig. 9 displays the residence time calculated by the MD and MC methods, changing with various channel widths. It can be referred that the wider size of the channel leads to the shorter residence time, suggesting that the wider channel makes the particle transition between the states more frequently. The narrower channel inhibits the particle transition, which causes a longer residence time. The MC results present a good agreement with the MD results, which verifies the accuracy of our algorithm concerning the transition probability.

In the simulation environment, the molecules are always affected by the external force field [34], such as gravity and electrostatic fields. To verify the algorithm's reliability in the conditions with the external force field, an acceleration in the positive y-axis direction is applied first. As shown in Fig. 10, the residence time in the states of box1 and box2 increases with a larger external acceleration. As the external force makes the particle prefer to stay in the box rather than the channel, the state transitions between box1 and box2 are limited, but the state transition also happens if the simulation covers a long time. In extreme conditions, the particle cannot get into the channel, which means that the state transition cannot happen. The algorithm accuracy and robustness have been validated by adding the external force field in our simulation, as indicated by the good match between MD and MC results in Fig. 10.

To further examine the results with external force in different directions, the acceleration direction is changed from the positive y-axis to the negative x-axis direction. Fig. 11 shows the particle residence time with different external accelerations in that condition. Due to the force towards the state box1, the residence time in box1 increases with the stronger external force, while in box2, the residence time shows a decline, because the transition to the state box2 needs to overcome greater resistance caused by the external force. Therefore, the particle prefers to stay in the state of box1, and the opposite trends in box1 and box2 further validate the reliability of our algorithm.

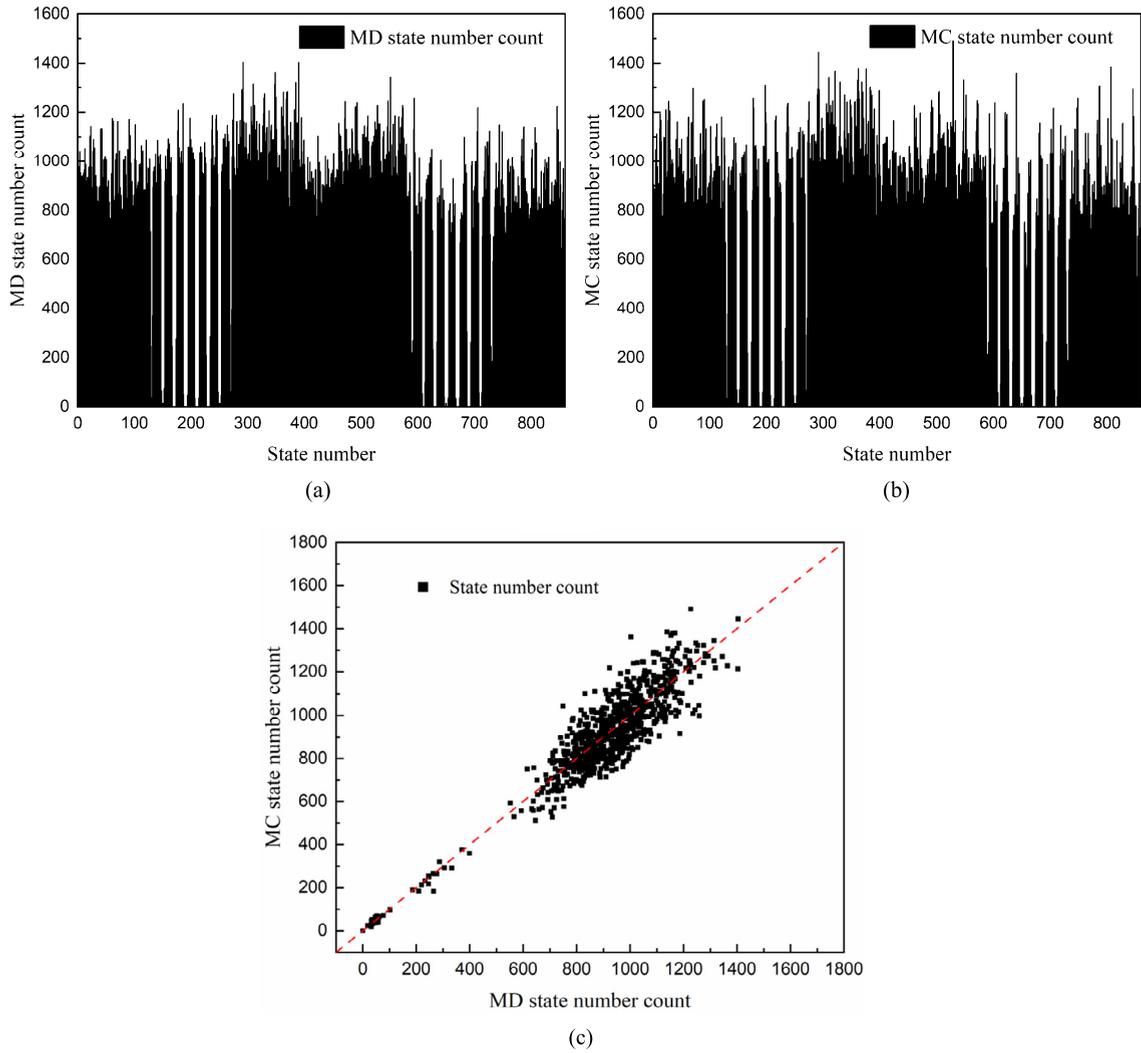


Fig. 8. The state distributions by using (a) the MD method and (b) MC method, and (c) the results comparison of the MD and MC methods.

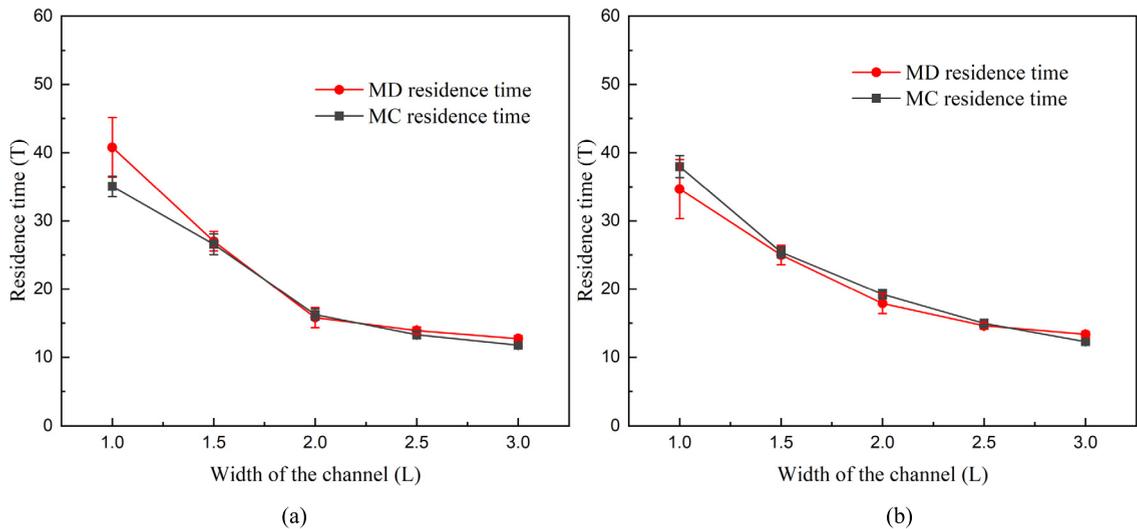


Fig. 9. The residence time without acceleration in the (a) box1 and (b) box2.

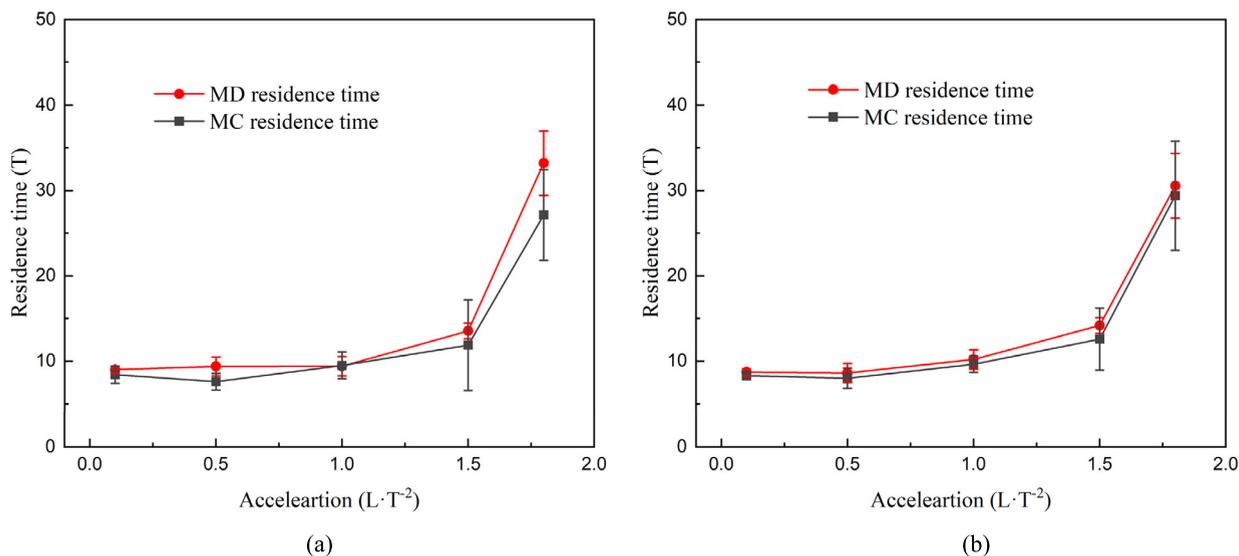


Fig. 10. The residence time with acceleration in the positive y-axis direction in the (a) box1 and (b) box2.

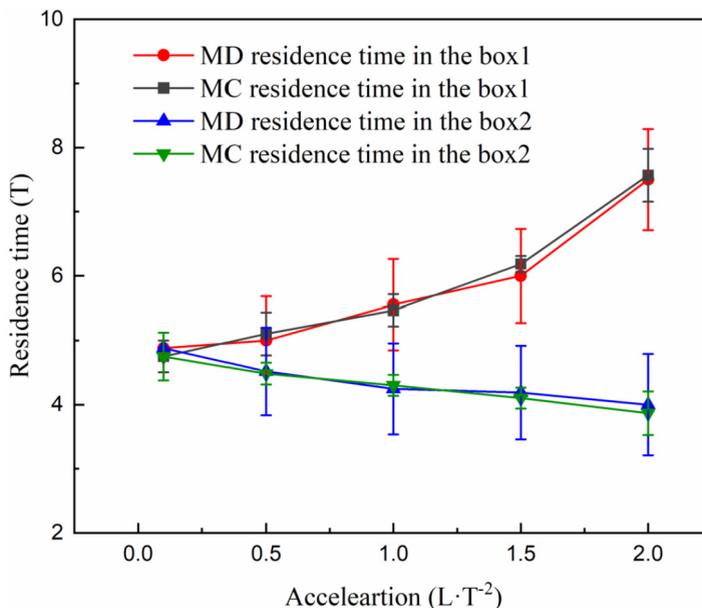


Fig. 11. The residence time with acceleration in the negative x-axis direction in box1 and box2.

3.2. The new state definition

The velocity also plays a role in the state transitions, especially in some velocity-sensitive systems [35], thus the velocity direction is decomposed into several angle ranges, as depicted in Fig. 12 (b). For example, two decomposition types denote that the velocity angle is decomposed into two states, which locate in the ranges of $0 - \pi$ and $\pi - 2\pi$ respectively. Thus, the total number of states turns to be two times bigger than the previous state number ($840 \times 2 = 1680$). Similar to Fig. 7, the count number on each state is calculated by the MD and MC methods, which are plotted on the x-axis and y-axis in Fig. 12 (a). It can be referred that a better match on the distribution of the state can be obtained, if the velocity angle is decomposed into more states. However, as the size of the transition matrix will grow exponentially with more states, the MC computational efficiency will also be damaged, as shown in Fig. 12 (c). After evaluating the simulation accuracy and computational cost, the eight decomposition type is a better choice.

Apart from the velocity angles, the sampling step, which is used to generate the transition matrix, also affects the consistency between the MD and MC results. As shown in Fig. 13 (a), the larger sampling step leads to a more dispersive

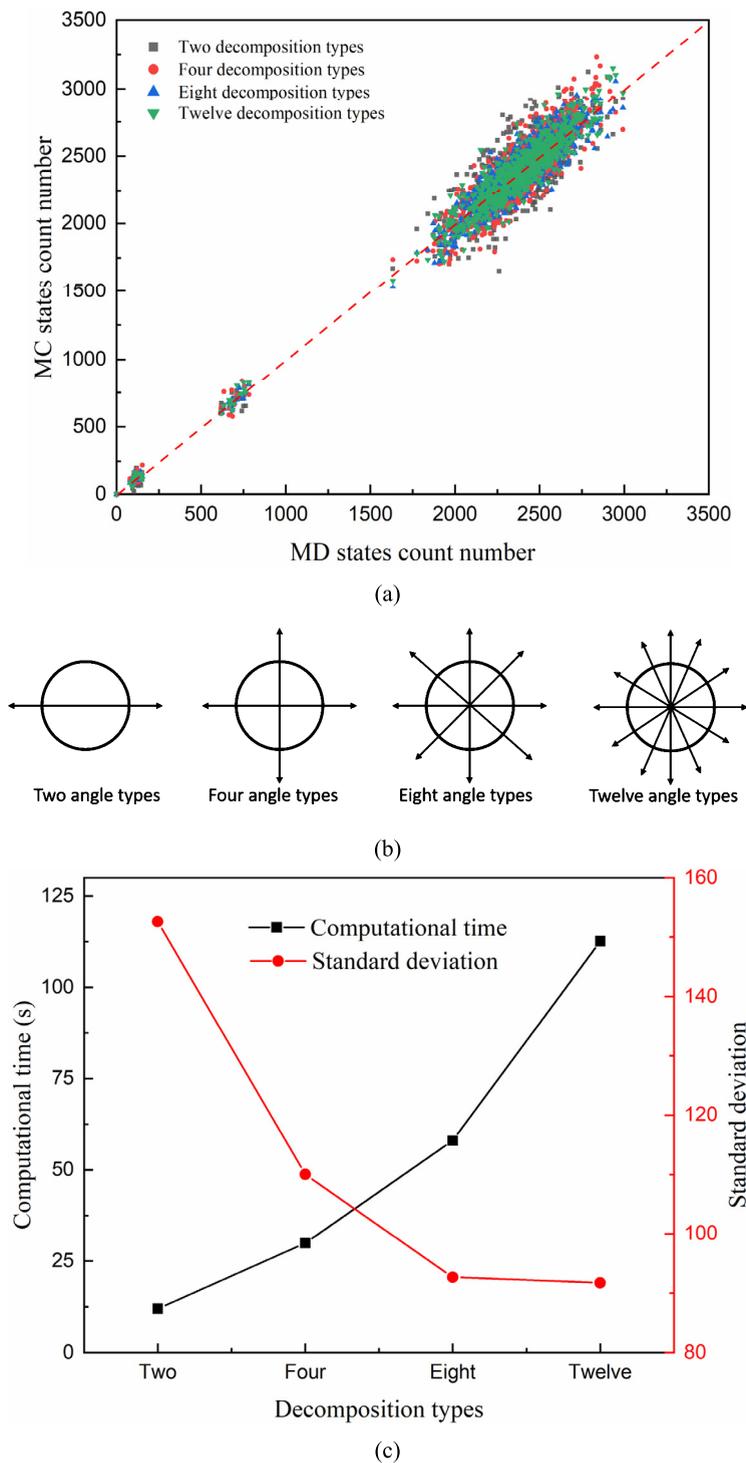


Fig. 12. (a) The Corresponding map of the MD and MC states counts with different decomposition types, (b) the schematic diagram of velocity decomposition types, and (c) the computational time and standard deviations of various decomposition types.

distribution, and the smaller step makes the scatters more concentrated, because the larger step will lose more information of the MD trajectory and induces more errors, as shown in Fig. 13 (b). However, it should be noticed that when the sampling step is very small, the probability trap may be formed in the transition matrix, which prompts the particle to make self-transitions and a misleading result may be obtained.

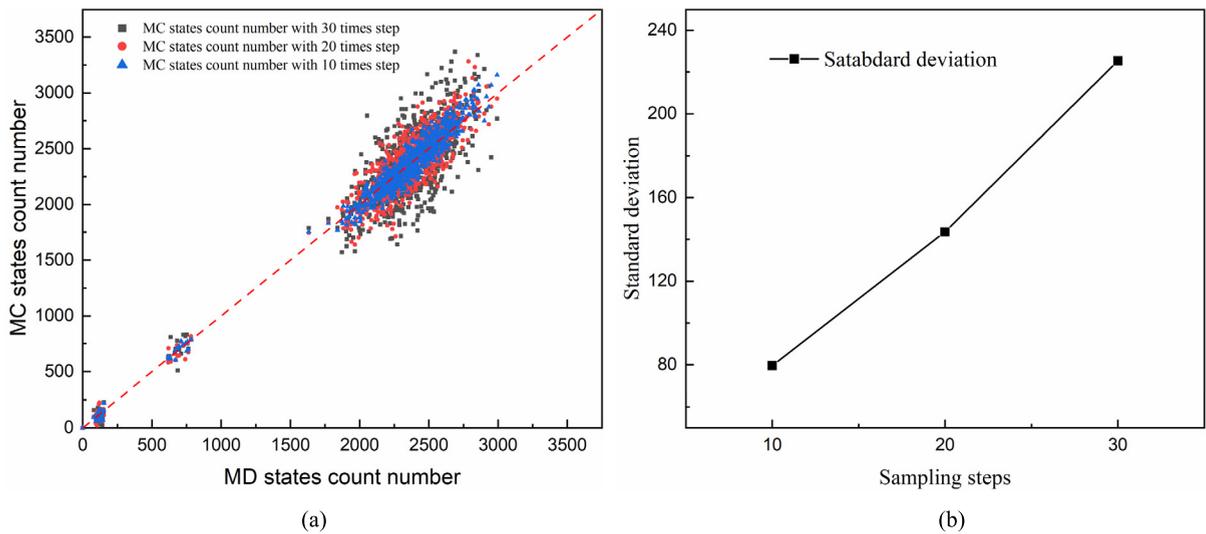


Fig. 13. (a) The corresponding map of the MD and MC states counts with different sampling steps. (b) The standard deviation of different sampling steps which are used to build the transition matrix.

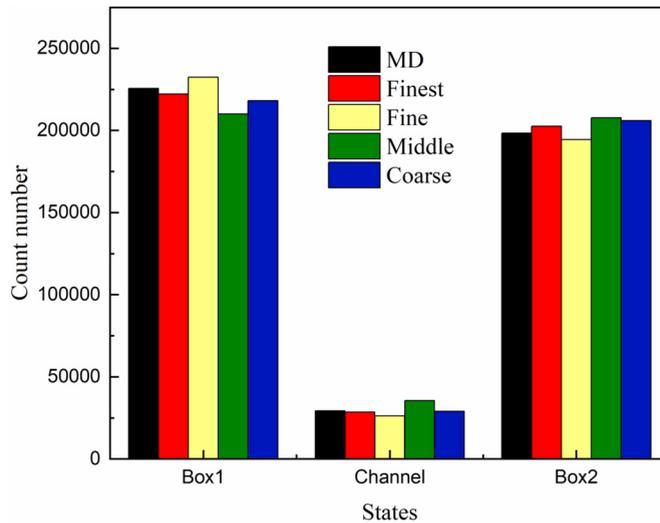


Fig. 14. The total state count number of coarsening models in three box states, with an external force from box2 to box1.

3.3. The coarsening

Using the idea of spatial coarsening, the coarsening models are developed based on the finest model, which takes the velocity states into consideration and 13440 states are included. The fine model is created without the velocity types, and 840 states are included. The middle model has 55 states by merging the transition matrix. The coarse model only has three states to represent three macro-scale states. The five scales, which include the MD scale, are calculated with the external force from box2 to box1. The state distribution is examined, and the count numbers of five models are plotted in Fig. 14. The results show that all the coarsening models present similar state count numbers with the MD results. As drawn in Fig. 15 (a), the residence time is counted in five scales, and the results of the finest model are the most similar to the MD results, suggesting that the finest model can retain most of the state information produced by the MD method. The fine and middle models show the relatively consistent residence time with acceptable errors. However, the coarse model exhibits overestimated residence time, which can be attributed to the three-by-three transition matrix. The probabilities on the diagonal elements are enlarged, which promotes the self-transition of states. Less computational time is needed if the scale is coarser, as shown in Fig. 15 (b), and the significant efficiency improvement can be captured from the finest to the fine scales. After the fine scale, the computational time slowly decreases if the model is further coarsened, which can be attributed to the small size of the transition matrix in the middle and coarse scales.

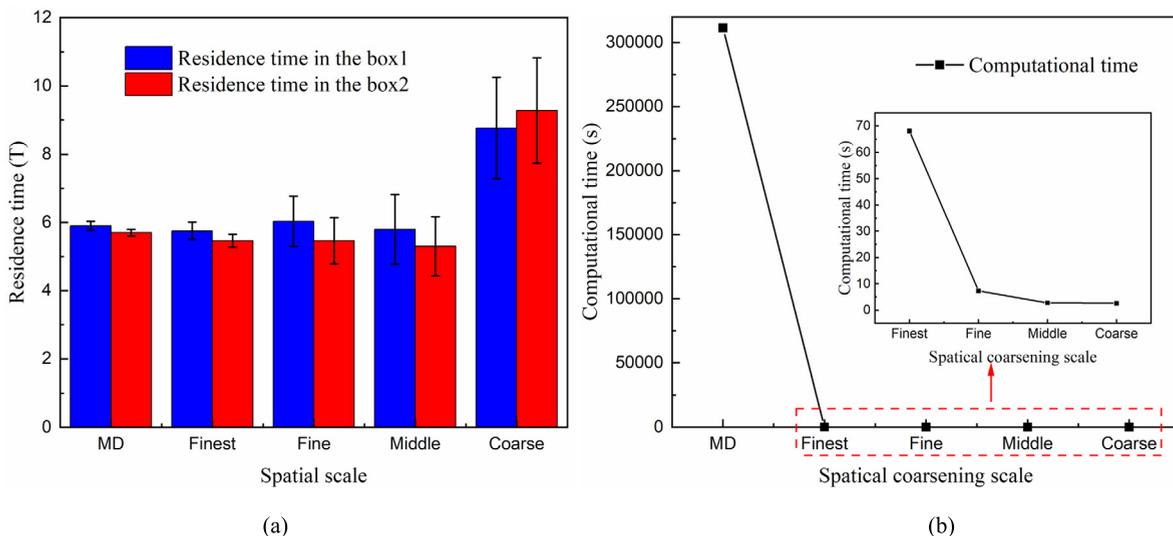


Fig. 15. (a) The residence time and (b) the computational time of the MC results in the states of box1 and box2 at different coarsening scales, with an external force from box2 to box1.

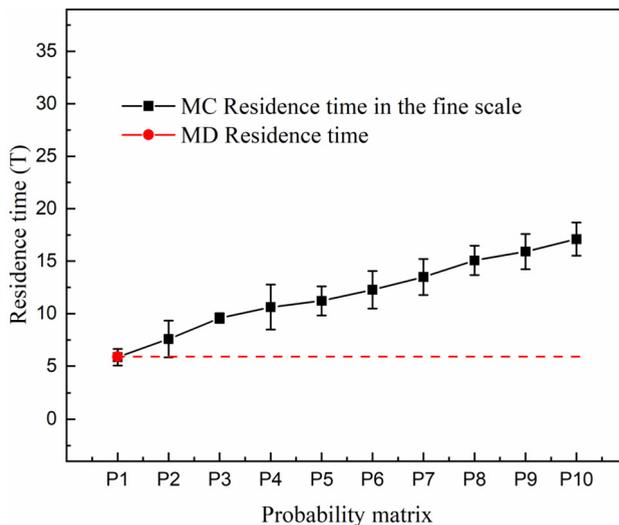


Fig. 16. The time coarsening results within tenfold state transition in box1, and the red dash line is the residence time value produced by the MD.

The spatial coarsening is accomplished by merging the states in the transition matrix, and then the state number is reduced. However, for the time coarsening, it is carried out by reducing the transition steps. As shown in Fig. 16, for example, the P3 denotes $P \times P \times P$, and the residence time is overestimated with a coarser transition matrix, as a part of the information of residence time is lost if the particle motions in several steps are merged in one step.

3.4. The cases in the energy and biochemical researches

Two practical engineering problems will be explained in this section on the adaptability of our proposed algorithm. The first example is methane adsorption, which is common in unconventional reservoirs [36,37]. One methane molecule is set in the center of the organic slit model, as shown in Fig. 17 (a). By using the MD method, the methane is motivated by thermal motion. Fig. 17 (b) presents the trajectory of the methane molecule's adsorption process, from the bulk phase (30 - 25 Å) to the adsorbed phase (25 - 20 Å). The methane keeps a stable state in the bulk phase, but after adsorbing on the wall, the methane collides with the wall frequently, which corresponds to previous studies [38–42]. Therefore, the bulk and adsorbed phases are two macro-states in this model, which are similar to the states of box1 and box2 in our model. What is more, due to the thermal motions, the trajectory shows tiny shakes, suggesting that the states can be refined furtherly.

Fig. 17 (c) shows the diffusion coefficients in the bulk, adsorbed phases, and the entire domain, which is one-sixth of the mean square displacement (MSD)'s slope,

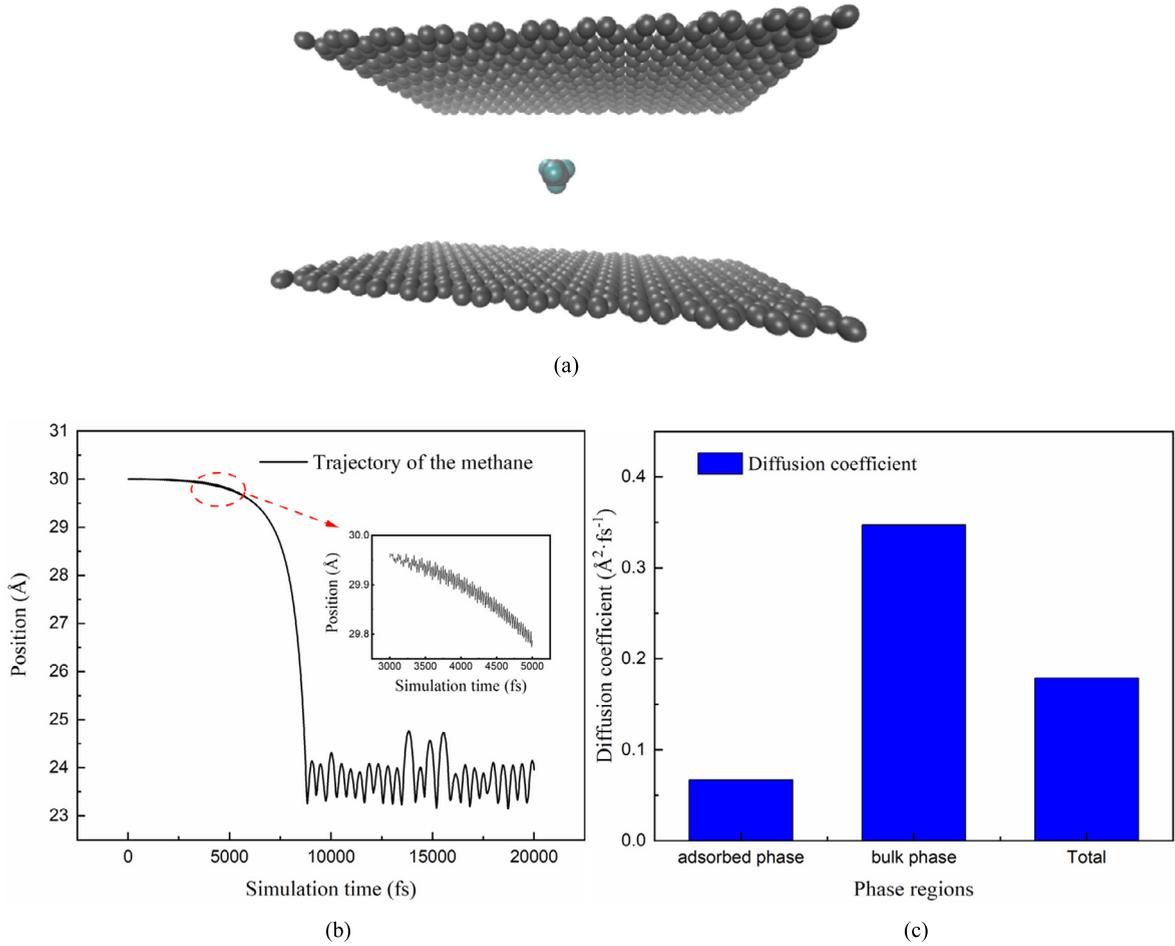


Fig. 17. (a) The shale gas adsorption model. (b) The methane trajectory in the states of bulk phase and adsorbed phase. (c) The diffusion coefficient of methane in the organic pore.

$$MSD = \frac{1}{N} \sum_{i=1}^N |x^i(t) - x^i(0)|^2 \quad (16)$$

where N denotes the number of particles to be averaged ($N = 1$), x^i the position of the i -th particle, $x^i(0)$ the initial position, and $x^i(t)$ the position at time t . The combined diffusion coefficient can be calculated as follows,

$$D_{combine}^{\alpha} = \frac{T_{surf}}{T_{bulk} + T_{surf}} D_{surf}^{\alpha} + \frac{T_{bulk}}{T_{bulk} + T_{surf}} D_{bulk}^{\alpha} \quad (17)$$

where T_{surf} denotes the residence time in the adsorbed phase, T_{bulk} denotes the residence time in the bulk phase, $\alpha = 1$ in this case, D_{surf}^{α} the diffusion coefficient in the adsorbed phase, D_{bulk}^{α} the diffusion coefficient in the bulk phase, and $D_{combine}^{\alpha}$ the diffusion coefficient in the entire pore space. The methane in the bulk phase moves with a high diffusion coefficient, while it is smaller in the adsorbed phase. The combined diffusion coefficient can be determined by the position and residence time in the bulk and adsorbed phases, suggesting that it can be studied by the multi-scale algorithm proposed in our work.

A protein folding case is also studied to verify the possibility of the algorithm's application in the biochemical research. A polypeptide in the protein 6LVN is relaxed by the MD method [43]. Fig. 18 provides the extended state and the folded state, suggesting that the protein problems can be studied by the state transition, where the state can be defined by the structure and free energy of the proteins [32]. In these two cases, the state transition and residence time in each state can be obtained, verifying the widely applicability of this multi-scale algorithm.

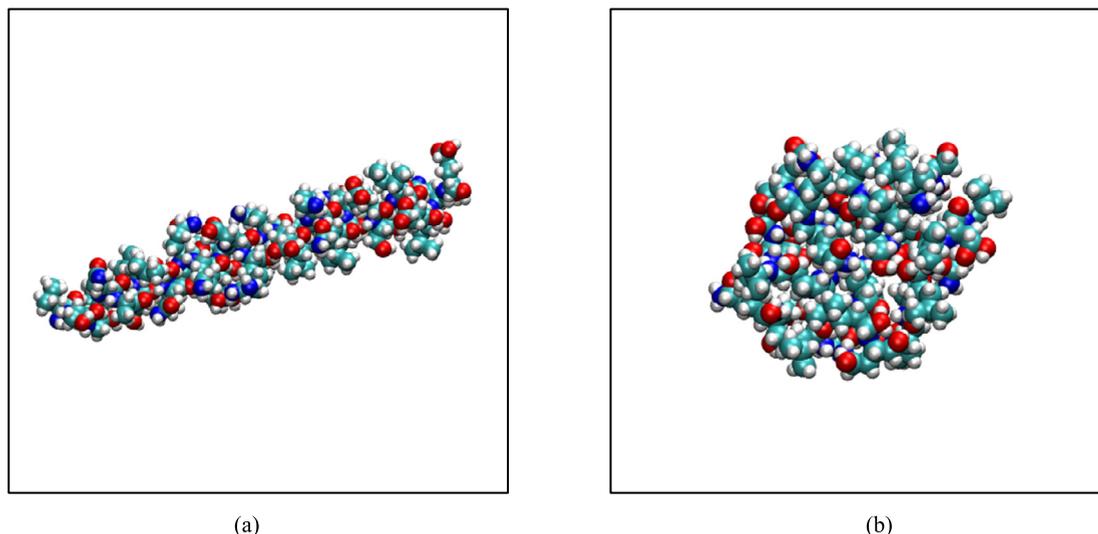


Fig. 18. The protein folding case with (a) the extended state and (b) the folded state.

4. Conclusions

In this work, a multi-scale molecular simulation algorithm is developed. A novel toy model is proposed for the first time to carry out typical state transition problems in various engineering scenarios. The accuracy, robustness, and efficiency of our algorithm have been proved in a number of numerical cases, and certain interesting phenomena have been captured with physically-meaningful analyses of the simulation results on different scales.

In the MD scale simulation, the exact solution MD algorithm is used to generate the accurate particle trajectory in the phase. It has been found that the free path time distribution contains two peaks, in which the circle-related time is the major contributor in the first peak, while the second peak is mainly contributed by the free path time between the opposite box boundaries. The reliability of the MD scale trajectory is validated by the physically-meaningful results on the free path time and residence time.

By using the realistic MD scale results, the probability transition matrix is created, which is further used in the MC simulation. It has been found that the wider size of the channel facilitates the state transitions between two boxes. The effect of the external forces on the state transition is also examined, and it is indicated in the results that the external force leads to the longer residence time for the particle, suggesting that some states can be more stable with the existence of external force field. The MC results present a good agreement with the MD results, verifying the accuracy of the algorithm. To describe the states more specifically, the velocity direction is taken into consideration. More velocity decomposition types can provide better accuracy, but the larger size of the transition matrix reduces the computational efficiency subsequently. According to the results, the four and eight velocity angles in the decomposition are selected as the appropriate choices.

To further accelerate the MC simulation, a coarsening idea is proposed, in which the spatial coarsening is carried out by merging the states in the transition matrix, and the time coarsening is proceeded by merging the transition steps. The coarser scale needs less computational time, but better accuracy is preserved in the fine scale. The time coarsening can reduce further the computational time by merging the transition steps. Furthermore, to demonstrate the general applicability of our proposed algorithm and toy model, two practical engineering cases, i.e., the adsorption of shale gas and protein folding, are explained on the potential application of our method.

In summary, a multi-scale molecular simulation algorithm, as well as a novel toy model, is proposed and verified in the MD and MC scales, which provides a reference method for further studies on a number of energy and biochemical researches, and promotes the development of relevant acceleration algorithms.

CRediT authorship contribution statement

Jie Liu: programming, visualization, writing.

Qinglin Tang: reviewing, validation.

Jisheng Kou: reviewing, editing.

Dingguo Xu: methodology, reviewing.

Tao Zhang: methodology, editing, supervision.

Shuyu Sun: conceptualization, methodology, supervision.

Declaration of competing interest

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.

Data availability

Data will be made available on request.

Acknowledgements

We would like to express appreciation to the following financial support: National Natural Scientific Foundation of China (Grants No. 51874262, 51936001, 21973064), King Abdullah University of Science and Technology (KAUST) through the grants BAS/1/1351-01, URF/1/4074-01, and URF/1/3769-01.

References

- [1] J.-P. Ryckaert, G. Ciccotti, H.J. Berendsen, Numerical integration of the Cartesian equations of motion of a system with constraints: molecular dynamics of n-alkanes, *J. Comput. Phys.* 23 (1977) 327–341.
- [2] M. Karplus, J.A. McCammon, Molecular dynamics simulations of biomolecules, *Nat. Struct. Biol.* 9 (2002) 646–652.
- [3] Q. Feng, S. Xu, X. Xing, W. Zhang, S. Wang, Advances and challenges in shale oil development: a critical review, *Adv. Geo-Energy Res.* 4 (2020) 406–418.
- [4] Y. Yang, J. Liu, J. Yao, J. Kou, Z. Li, T. Wu, K. Zhang, L. Zhang, H. Sun, Adsorption behaviors of shale oil in kerogen slit by molecular simulation, *Chem. Eng. J.* 387 (2020) 124054.
- [5] P.L. Freddolino, C.B. Harrison, Y. Liu, K. Schulten, Challenges in protein-folding simulations, *Nat. Phys.* 6 (2010) 751–758.
- [6] S. Piana, J.L. Klepeis, D.E. Shaw, Assessing the accuracy of physical models used in protein-folding simulations: quantitative evidence from long molecular dynamics simulations, *Curr. Opin. Struct. Biol.* 24 (2014) 98–105.
- [7] A. Gershenson, L.M. Gierasch, Protein folding in the cell: challenges and progress, *Curr. Opin. Struct. Biol.* 21 (2011) 32–41.
- [8] T. Zhang, S. Sun, H. Song, Flow mechanism and simulation approaches for shale gas reservoirs: a review, *Transp. Porous Media* 126 (3) (2019) 655–681.
- [9] Huiying Guo, Ziqiang Wang, Bei Wang, Yuankai Zhang, Haojin Meng, Hongguang Sui, Molecular dynamics simulations of oil recovery from dolomite slit nanopores enhanced by CO₂ and N₂ injection, *Adv. Geo-Energy Res.* 6 (4) (2022) 306–313.
- [10] Veronica Salmaso, Stefano Moro, Bridging molecular docking to molecular dynamics in exploring ligand-protein recognition process: an overview, *Front. Pharmacol.* 9 (2018) 923.
- [11] J. Liu, Y. Yang, S. Sun, J. Yao, J. Kou, Flow behaviors of shale oil in kerogen slit by molecular dynamics simulation, *Chem. Eng. J.* (2022) 134682.
- [12] J.L. Klepeis, K. Lindorff-Larsen, R.O. Dror, D.E. Shaw, Long-timescale molecular dynamics simulations of protein structure and function, *Curr. Opin. Struct. Biol.* 19 (2009) 120–127.
- [13] A. Bhoutekar, S. Ghosh, S. Bhattacharya, A. Chatterjee, A new class of enhanced kinetic sampling methods for building Markov state models, *J. Chem. Phys.* 147 (2017) 152702.
- [14] V.J. Bhute, A. Chatterjee, Accuracy of a Markov state model generated by searching for basin escape pathways, *J. Chem. Phys.* 138 (2013) 084103.
- [15] A. Chatterjee, Accelerating rare events and building kinetic Monte Carlo models using temperature programmed molecular dynamics, *J. Mater. Res.* 33 (2018) 835–846.
- [16] D.E. Otzen, L.S. Itzhaki, N.F. ElMasry, S.E. Jackson, A.R. Fersht, Structure of the transition state for the folding/unfolding of the barley chymotrypsin inhibitor 2 and its implications for mechanisms of protein folding, *Proc. Natl. Acad. Sci. USA* 91 (1994) 10422–10425.
- [17] M.H. Kalos, P.A. Whitlock, *Monte Carlo Methods*, John Wiley & Sons, 2009.
- [18] N. Metropolis, S. Ulam, The Monte Carlo method, *J. Am. Stat. Assoc.* 44 (1949) 335–341.
- [19] T.D. Dreeben, S.B. Pope, Probability density function/Monte Carlo simulation of near-wall turbulent flows, *J. Fluid Mech.* 357 (1998) 141–166.
- [20] J.R. Errington, Direct calculation of liquid–vapor phase equilibria from transition matrix Monte Carlo simulation, *J. Chem. Phys.* 118 (2003) 9915–9925.
- [21] C.J. Geyer, Practical Markov chain Monte Carlo, *Stat. Sci.* (1992) 473–483.
- [22] U.V. Bhapkar, M.S. Shur, Monte Carlo calculation of velocity-field characteristics of wurtzite GaN, *J. Appl. Phys.* 82 (1997) 1649–1655.
- [23] A. Rambaut, N.C. Grass, Seq-Gen: an application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees, *Bioinformatics* 13 (1997) 235–238.
- [24] S. Langenbuch, W. Maurer, W. Werner, Coarse-mesh flux-expansion method for the analysis of space-time effects in large light water reactor cores, *Nucl. Sci. Eng.* 63 (1977) 437–456.
- [25] D. Molnar, R. Mukherjee, A. Choudhury, A. Mora, P. Binkele, M. Selzer, B. Nestler, S. Schmauder, Multiscale simulations on the coarsening of Cu-rich precipitates in α -Fe using kinetic Monte Carlo, molecular dynamics and phase-field simulations, *Acta Mater.* 60 (2012) 6961–6971.
- [26] D. Beeman, Some multistep methods for use in molecular dynamics calculations, *J. Comput. Phys.* 20 (1976) 130–139.
- [27] A.D. Stone, Chaotic billiard lasers, *Nature* 465 (2010) 696–697.
- [28] C. Gmachl, F. Capasso, E. Narimanov, J.U. Nöckel, A.D. Stone, J. Faist, D.L. Sivco, A.Y. Cho, High-power directional emission from microlasers with chaotic resonators, *Science* 280 (1998) 1556–1564.
- [29] P.T. Boyd, S.L. McMillan, Chaotic scattering in the gravitational three-body problem, *Chaos Interdiscip. J. Nonlinear Sci.* 3 (1993) 507–523.
- [30] U. Siebert, O. Alagoz, A.M. Bayoumi, B. Jahn, D.K. Owens, D.J. Cohen, K.M. Kuntz, State-transition modeling: a report of the ISPOR-SMDM modeling good research practices task force–3, *Med. Decis. Mak.* 32 (2012) 690–700.
- [31] U.H. Hansmann, Y. Okamoto, New Monte Carlo algorithms for protein folding, *Curr. Opin. Struct. Biol.* 9 (1999) 177–183.
- [32] J.D. Chodera, N. Singhal, V.S. Pande, K.A. Dill, W.C. Swope, Automatic discovery of metastable states for the construction of Markov models of macromolecular conformational dynamics, *J. Chem. Phys.* 126 (2007), 04B616.
- [33] J. Handschin, Monte Carlo techniques for prediction and filtering of non-linear stochastic processes, *Automatica* 6 (1970) 555–563.
- [34] M. Wittkop, J.-U. Sommer, S. Kreitmeyer, D. Göritz, Monte Carlo simulations of a single polymer chain under an external force in two and three dimensions, *Phys. Rev. E* 49 (1994) 5472.
- [35] Y. Shi, D.W. Brenner, Molecular simulation of the influence of interface faceting on the shock sensitivity of a model plastic bonded explosive, *J. Phys. Chem. B* 112 (2008) 14898–14904.
- [36] Tao Zhang, Shuyun Sun, A coupled Lattice Boltzmann approach to simulate gas flow and transport in shale reservoirs with dynamic sorption, *Fuel* 246 (2019) 196–203.

- [37] J. Cai, H. Hajibeygi, J. Yao, M. Hassanizadeh, Advances in porous media science and engineering from InterPore2020 perspective, *Adv. Geo-Energy Res.* 4 (2020) 352–355.
- [38] W. Song, B. Yao, J. Yao, Y. Li, H. Sun, Y. Yang, L. Zhang, Methane surface diffusion capacity in carbon-based capillary with application to organic-rich shale gas reservoir, *Chem. Eng. J.* 352 (2018) 644–654.
- [39] S. Wang, Q. Feng, F. Javadpour, Y.-B. Yang, Breakdown of fast mass transport of methane through calcite nanopores, *J. Phys. Chem. C* 120 (2016) 14260–14269.
- [40] S. Zhan, Y. Su, Z. Jin, M. Zhang, W. Wang, Y. Hao, L. Li, Study of liquid-liquid two-phase flow in hydrophilic nanochannels by molecular simulations and theoretical modeling, *Chem. Eng. J.* 395 (2020) 125053.
- [41] M. Zhang, S. Zhan, Z. Jin, Recovery mechanisms of hydrocarbon mixtures in organic and inorganic nanopores during pressure drawdown and CO₂ injection from molecular perspectives, *Chem. Eng. J.* 382 (2020) 122808.
- [42] T. Zhang, S. Sun, Thermodynamics-informed neural network (TINN) for phase equilibrium calculations considering capillary pressure, *Energies* 14 (22) (2021) 7724.
- [43] S. Xia, Y. Zhu, M. Liu, Q. Lan, W. Xu, Y. Wu, T. Ying, S. Liu, Z. Shi, S. Jiang, Fusion mechanism of 2019-nCoV and fusion inhibitors targeting HR1 domain in spike protein, *Cell. Mol. Immunol.* 17 (2020) 765–767.