J. Fluid Mech. (2020), *vol.* 892, A5. © The Author(s), 2020.
Published by Cambridge University Press
doi:10.1017/jfm.2020.184

Data-driven discovery of governing equations for fluid dynamics based on molecular simulation

Jun Zhang[1,†] **and Wenjun Ma**[1]

[1]School of Aeronautic Science and Engineering, Beihang University, Beijing 100191, PR China

(Received 28 October 2019; revised 3 February 2020; accepted 2 March 2020)

The discovery of governing equations from data is revolutionizing the development of some research fields, where the scientific data are abundant but the well-characterized quantitative descriptions are probably scarce. In this work, we propose to combine the direct simulation Monte Carlo (DSMC) method, which is a popular molecular simulation tool for gas flows, and machine learning to discover the governing equations for fluid dynamics. The DSMC method does not assume any macroscopic governing equations *a priori* but just relies on the model of molecular interactions at the microscopic level. The data generated by DSMC are utilized to derive the underlying governing equations using a sparse regression method proposed recently. We demonstrate that this strategy is capable of deriving a variety of equations in fluid dynamics, such as the momentum equation, diffusion equation, Fokker–Planck equation and vorticity transport equation. The data-driven discovery not only provides the right forms of the governing equations, but also determines accurate values of the transport coefficients such as viscosity and diffusivity. This work proves that data-driven discovery combined with molecular simulations is a promising and alternative method to derive governing equations in fluid dynamics, and it is expected to pave a new way to establish the governing equations of non-equilibrium flows and complex fluids.

Key words: molecular dynamics

1. Introduction

Historically, the macroscopic governing equations of fluid dynamics, i.e. the well-known Navier–Stokes equations, were derived from the basic principles of continuity of mass and momentum, with the assumption that the fluid at the macroscopic scale is a continuous substance (Batchelor 2000). Note that the general form of the Navier–Stokes equations is essentially not closed, as the stress tensor is unknown except that a constitutive relation is assumed *a priori*. For Newtonian fluids, a reasonable assumption is that the stress tensor is linearly proportional to the local strain rate. For complex fluids or non-equilibrium flows, the constitutive relation is not so straightforward and usually depends on phenomenological models.

† Email address for correspondence: jun.zhang@buaa.edu.cn

https://doi.org/10.1017/jfm.2020.184 Published online by Cambridge University Press

Although the Navier–Stokes equations have been widely applied to describe fluid flows, it is commonly believed that they fail in the description of flows at large Knudsen numbers ($Kn > 0.1$), where the non-equilibrium gas effect plays an important role (Bird 1994). The Knudsen number is defined as the ratio between the molecular mean free path, i.e. the average distance travelled by one molecule between two subsequent collisions, and the characteristic chord length of the system, e.g. the chord length of an airfoil or the diameter of a cylinder. For non-equilibrium gas flows such as micro-flows and near-space flights, the Knudsen numbers are prone to be larger than 0.1, and thus a reliable set of equations above the Navier–Stokes level is highly desirable.

Theoretically, the macroscopic transport equations for non-equilibrium gas flows can be derived from the Boltzmann equation, which describes the microscopic behaviour of gases from a statistical point of view by accounting for molecular movements and collisions. One classical method for the derivation of macroscopic equations is the Chapman–Enskog method (Chapman & Cowling 1990), which expands the distribution function around equilibrium state in a series of the Knudsen number. To the first order of Knudsen number, the Navier–Stokes equations are reproduced, while expansions to the second and third orders of Knudsen number give rise to the so-called Burnett and super-Burnett equations, respectively. Another method is Grad's moment method (Struchtrup 2005), which expands the distribution function in Hermite polynomials, the coefficients of which are linear combinations of the moments. In Grad's seminal work, he truncated the distribution function at the third order in Hermite polynomials and derived the well-known 13 moment equations (Struchtrup 2005). Afterwards, a lot of efforts have been made in this research direction, including the regularized 13 moment equations (Struchtrup & Torrilhon 2003) and the regularized 26 moment equations (Gu & Emerson 2009; Gu *et al.* 2019). The applicability of these derived macroscopic governing equations to moderate non-equilibrium gas flows has been well validated, while the applicability to highly non-equilibrium gas flows is still limited and obscure.

Advances in machine learning (Jordan & Mitchell 2015; Duraisamy, Iaccarino & Xiao 2019; Brunton, Noack & Koumoutsakos 2020) and data science (Marx 2013; Brunton & Kutz 2019) have provided engineers and scientists across all disciplines new opportunities for data-driven discovery, which has been referred to as the fourth paradigm of scientific discovery (Hey, Tansley & Tolle 2009). Particularly, many concepts and methods from statistical learning can be employed to develop accurate models for complex dynamical systems directly from data. Methods for data-driven discovery of dynamical systems include equation-free modelling (Kevrekidis *et al.* 2003), artificial neural networks (Gonzalez, Rico & Kevrekidis 1998) and automated inference of the dynamics (Daniels & Nemenman 2015), just to name a few. In this series of developments, a seminal breakthrough was made by Bongard & Lipson (2007) and Schmidt & Lipson (2009), who used symbolic regression to determine a nonlinear dynamic system from data directly. The disadvantage of symbolic regression is that it is usually expensive and prone to overfitting.

More recently, sparsity (Tibshirani 1996; Loiseau & Brunton 2018) has been used to determine, in a highly efficient computational manner, the governing equations of a dynamical system. Significant progress in this direction has been made by Brunton, Proctor & Kutz (2016), who combined a sparsity-promoting technique and machine learning with nonlinear dynamical systems to discover ordinary differential equations (ODEs) from noisy measurement data. In particular, the sparse regression avoids overfitting by selecting parsimonious models that balance model accuracy with complexity. Only those terms that are most informative about the dynamics are

https://doi.org/10.1017/jfm.2020.184 Published online by Cambridge University Press

selected as part of the discovered ODEs. Afterwards, Rudy *et al.* (2017) extended this method to handle spatio-temporal data or high-dimensional measurements and to discover the governing partial differential equations (PDEs) of the system. The algorithm they proposed is called PDE functional identification of nonlinear dynamics (PDE-FIND), which is computationally efficient, robust and has been successfully applied to a variety of canonical problems including fluid dynamics governed by Navier–Stokes equations. Note that a similar algorithm to PDE-FIND has also been proposed by Schaeffer (2017) independently.

In the applications of PDE-FIND provided in the original work of Rudy *et al.* (2017), the data used to derive the PDEs were virtually generated by the numerical solutions of the associated governing equations. A more convincing demonstration would consider data from experimental observations or numerical simulations, which are independent of the derived governing equations. In this work, we generate spatio-temporal data of flow fields through a popular molecular simulation method, specifically, the direct simulation Monte Carlo (DSMC) method. DSMC method employs a large number of representative molecules to model gas flows. Note that, in the DSMC method, there is no need to assume any macroscopic governing equations *a priori*. The macroscopic quantities, such as density and velocity, are obtained by sampling molecular information and making an average at the computational cells. Theoretically, DSMC has been proved to be a particle method for solving the Boltzmann equation (Wagner 1992). Based on the spatio-temporal data obtained by DSMC, we derive the macroscopic governing equations for a variety of fluid flows using the PDE-FIND method.

Our ultimate objective is to develop general macroscopic governing equations for a wide range of non-equilibrium gas flows based on the data generated by DSMC, for which applicability to the whole range of Knudsen number flows has been validated. As a first step, in this work we focus on data-driven discovery of macroscopic governing equations at the Navier–Stokes level, i.e. in the continuum regime, where the theoretical macroscopic governing equations have been well established. The purpose is to verify that the data-driven discovery combined with molecular simulations is an alternative method to derive governing equations in fluid dynamics, besides the Chapman–Enskog expansion from the Boltzmann equation. To the best of our knowledge, this is the first time that macroscopic governing equations are derived by molecular simulations combined with a machine learning method.

The remainder of this paper is organized as follows. In § 2, we first describe the DSMC method for generating data, and then introduce the methodology for data-driven discovery, i.e. the PDE-FIND algorithm. In § 3, a variety of benchmark cases are provided to check the validity of our strategy for data-driven discovery of macroscopic governing equations, including the momentum equation, diffusion equation, Fokker–Planck equation and vorticity transport equation. Conclusions are given in § 4.

2. Methodology

2.1. *DSMC method*

The direct simulation Monte Carlo method was first proposed by Bird in the 1960s and is a stochastic particle-based algorithm to solve the Boltzmann equation by approximating the continuous molecular velocity distribution function with a discrete number of simulation molecules (Oran, Oh & Cybyk 1998; Zhang *et al.* 2019*a*). Figure 1 is a schematic of the DSMC method, where each simulation molecule

https://doi.org/10.1017/jfm.2020.184 Published online by Cambridge University Press

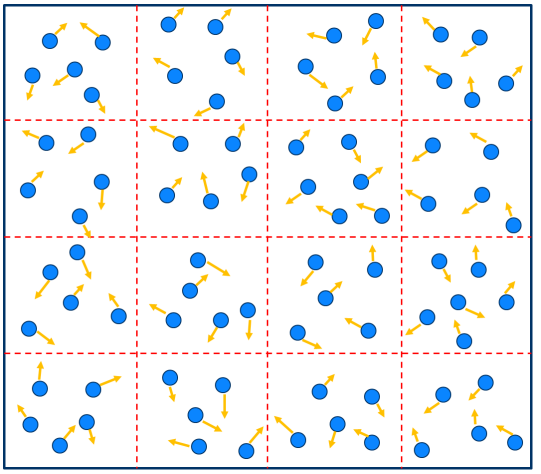

FIGURE 1. Schematic diagram of DSMC method.

displayed by a blue sphere is randomly selected from a large number of real molecules. These simulation molecules are tracked as they move with their velocities, collide with other molecules and reflect from boundaries within a computational domain. Macroscopic gas properties are obtained by sampling corresponding molecular information and making an average at the computational cells, which are marked by red dashed lines in figure 1. Specifically, density (ρ) and macroscopic velocity (u) are given by,

$$\rho = \frac{\sum_{i=1}^{N} m_i}{\delta V}, \tag{2.1}$$

$$u = \frac{\sum_{i=1}^{N} C_i}{N}, \tag{2.2}$$

where m_i and C_i are molecular mass and molecular velocity, respectively, δV is the volume of the computational cell and N is the number of molecules in the computational cell for sampling and averaging. To reduce statistical errors, a time average and ensemble average are usually used.

For any application of the DSMC method, the first step is initializing simulation molecules according to the density distribution in the computational domain, and the following steps implement two sequential processes in each calculated time interval, i.e. molecular motions and inter-molecular collisions, which are assumed uncoupled during one time step. The molecular motions are implemented in a deterministic way, that is, every molecule moves ballistically from its original position to a new position, and the displacement is equal to the product of its velocity times the time step. If the predicted new position of one molecule crosses any boundary of the computational domain, an appropriate boundary condition needs to be applied. Specifically, the time at which the molecule hits the wall is first identified according to the distance between the molecule and the boundary divided by the molecular velocity, and then

https://doi.org/10.1017/jfm.2020.184 Published online by Cambridge University Press

the molecular velocity reflected from the boundary is determined by the imposed gas–wall interaction model, such as specular, diffuse and Maxwell reflection models. Afterwards, the molecule moves according to its new reflected velocity within the remaining time of one calculating time step.

The inter-molecular collisions in the DSMC method are implemented in a probabilistic way, which is inherently different from other deterministic methods such as molecular dynamics. Several algorithms for the modelling of inter-molecular collisions have been proposed within the framework of DSMC. The most widely used model now is the no-time-counter (NTC) technique proposed by Bird (1994). In the NTC method, molecules in the same cells are randomly chosen as collision partners. The probability of selecting a collision pair is proportional to the relative speed between these two selected molecules. The post-collision velocities of molecules depend on the molecular model employed, which plays an important role for the accurate modelling of the real gas flows. The DSMC method allows for the introduction of phenomenological models such as the hard sphere (HS) and variable hard sphere models (Bird 1994). For the argon gas at a fixed temperature used in this work, we employ the simplest HS model to describe interactions between gaseous molecules. Specifically, the molecular diameter is set to $d = 3.659 \times 10^{-10}$ m, which is determined by the Chapman–Enskog theory and has been proven to give a correct prediction of viscosity.

The DSMC method was first applied to the simulation of high-speed gas flows in the context of aerospace engineering. Afterwards, the DSMC method has been successfully extended to investigate a variety of gas flows at the molecular level, such as micro-flows (Sun & Boyd 2002), flow instability (Stefanov, Roussinov & Cercignani 2002; Zhang & Fan 2009; Zhang, Fan & Fei 2010; Manela & Zhang 2012) and even turbulence (Gallis *et al.* 2017; Zhang *et al.* 2019*b*). Theoretically, the DSMC method can be applied to simulate the whole regime of gas flows. It should be emphasized that, for an accurate simulation using the DSMC method, the time step needs to be smaller than the molecular mean collision time, and the cell sizes for the selection of collision pairs need to be smaller than the molecular mean free path (Alexander, Garcia & Alder 1998; Garcia & Wagner 2000).

2.2. *PDE-FIND method*

We employ the PDE-FIND algorithm proposed by Rudy *et al.* (2017) to derive the macroscopic governing equations based on the spatio-temporal results obtained by the DSMC method. Here, we just provide a brief description of the basic algorithm of the PDF-FIND method, and we refer readers to the original paper and supplementary materials (Rudy *et al.* 2017) for details.

The spatial and temporal evolution of flow fields such as the velocity $u(x, t)$ are obtained through DSMC calculations. We assume that the evolution of the flow field satisfies a PDE in terms of a general form as

$$u_t = N(u, u_x, u_{xx}, \ldots) = \sum_{j=1}^{d} N_j(u, u_x, u_{xx}, \ldots)\xi_j, \tag{2.3}$$

where $N(\cdot)$ is a complex nonlinear function that can be expanded as a sum of simple monomial basis functions N_j of u and its derivatives multiplied by the corresponding coefficient ξ_j. The derivatives of the data with respect to time and space can be obtained using either a finite difference or polynomial interpolation method. Generally,

https://doi.org/10.1017/jfm.2020.184 Published online by Cambridge University Press

the polynomial interpolation method performs better than the finite difference method if the data are associated with noise. Considering that the data generated from DSMC have inherent noise as DSMC is a stochastic particle method, the polynomial interpolation method is employed in this work to determine the derivatives.

Given a dataset comprising of m time steps and n grid points, the data and their derivatives are constructed to form a linear problem as follows:

$$\underbrace{\begin{pmatrix} u_t(x_1,t_1) \\ u_t(x_2,t_1) \\ \vdots \\ u_t(x_n,t_m) \end{pmatrix}}_{\boldsymbol{U}_t} = \underbrace{\begin{pmatrix} 1 & u(x_1,t_1) & u^2(x_1,t_1) & \cdots & u^3u_{xxx}(x_1,t_1) \\ 1 & u(x_2,t_1) & u^2(x_2,t_1) & \cdots & u^3u_{xxx}(x_2,t_1) \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & u(x_n,t_m) & u^2(x_n,t_m) & \cdots & u^3u_{xxx}(x_n,t_m) \end{pmatrix}}_{\Theta(\boldsymbol{U})} \xi. \tag{2.4}$$

In (2.4), each row represents an observation of the dynamics at a specific point in time and space, and ξ is a vector of coefficients which need to be determined. If we assume Θ on the right-hand side of (2.4) is an over complete library, which has sufficient potential terms, then the dynamics of the system should be well described by the assumed governing equation. Theoretically, the candidate functions have arbitrary options including any order of nonlinearities and partial derivatives. Considering the characteristics of the fluid dynamics problems studied in this work, the highest-order term in Θ is composed of the third power of u multiplied by the third derivative of u. For the sake of clarity, here, we list all the candidate functions considered for a one-dimensional problem, that is, $1, u, u^2, u^3, u_x, uu_x, u^2u_x, u^3u_x, u_{xx}, uu_{xx}, u^2u_{xx}, u^3u_{xx}, u_{xxx}, uu_{xxx}, u^2u_{xxx}$ and u^3u_{xxx}.

The key point in PDE-FIND is to select a sparse subset of active terms from the candidate functions, in other words, to determine the values of the vector of coefficients ξ. If the coefficient of one specific term is non-zero, then the corresponding candidate function is selected. For an unbiased representation of the dynamics, one straightforward method for determining ξ is to solve the least-squares problem. However, as reported by Rudy *et al.* (2017), the least-squares solution may be inaccurate even only with numerical round-off errors. In particular, ξ tends to have non-negligible values, suggesting a PDE with all the prescribed functional forms in the library. In this way, the derived PDE is too complicated to be used, although it is mathematically correct. More importantly, the least-squares problem is essentially poorly conditioned for regression problems. Numerical error in computing the derivatives of the data could be magnified when inverting Θ. On the other hand, sparse regression has been demonstrated to be an efficient method, ensuring that the coefficients of the candidate functions which do not appear in the governing equations are exactly zero (Brunton *et al.* 2016). Recently, Rudy *et al.* (2017) proposed utilizing sparse regression to approximate the solution of ξ as follows:

$$\xi = \arg\min_{\hat{\xi}} \|\Theta\hat{\xi} - \boldsymbol{U}_t\|_2^2 + \lambda\|\hat{\xi}\|_0. \tag{2.5}$$

This means that the prescribed terms only show up in the derived PDE if their effect on the error $\|\Theta\hat{\xi} - \boldsymbol{U}_t\|$ is greater than their addition to $\|\hat{\xi}\|_0$. The ℓ^0 term, i.e. the last term on the right-hand side of (2.5), makes this problem np-hard.

Specifically, the convex relaxation of the ℓ^0 optimization problem in (2.5) can be written as

$$\xi = \arg\min_{\hat{\xi}} \|\Theta\hat{\xi} - \boldsymbol{U}_t\|_2^2 + \lambda\|\hat{\xi}\|_1. \tag{2.6}$$

https://doi.org/10.1017/jfm.2020.184 Published online by Cambridge University Press

This convex optimization problem can be solved by the least absolute shrinkage and selection operator (LASSO) (Tibshirani 1996). However, previous studies demonstrated that LASSO tends to have difficulty in finding a sparse basis if the columns in the matrix Θ are correlated. Recently, Rudy *et al.* (2017) proposed an alternative method, called the sequentially threshold least-squares (STLS) method. In STLS, a least-squares predictor is obtained and a hard threshold is performed on the regression coefficients. The process is repeated recursively on the remaining non-zero coefficients.

As reported by Rudy *et al.* (2017), STLS performed better than LASSO in most cases but still has the challenge of correlation in the data. In order to overcome this problem, ridge regression with an ℓ^2 regularizer has been proposed by Rudy *et al.* (2017) to replace the least squares in STLS, that is,

$$\hat{\xi} = \arg\min_{\xi} \|\Theta\xi - \boldsymbol{U}_t\|_2^2 + \lambda\|\xi\|_2^2 = (\Theta^{\mathrm{T}}\Theta + \lambda I)^{-1}\Theta^{\mathrm{T}}\boldsymbol{U}_t. \tag{2.7}$$

This method is called sequential threshold ridge regression (STRidge) (Rudy *et al.* 2017). A variation of test cases in the recent works Rudy *et al.* (2017, 2019) have demonstrated that STRidge has the best empirical performance. Note that a different threshold tolerance would result in a different level of sparsity in the final solution. To find the best tolerance, predictors are trained for varying tolerances and their performances are evaluated.

3. Results and discussion

3.1. *Shear flow and momentum equation*

We simulate two-dimensional unbounded flow of argon gas at standard conditions, i.e. pressure $p = 1.01 \times 10^5$ Pa and temperature $T = 273$ K. According to the equation of state for perfect gases, the number density is $n = 2.69 \times 10^{25}$ m^{-3}. The computational domain is a square of side length $L = 100\lambda$, where λ is the molecular mean free path with the definition of $\lambda = 1/(\sqrt{2}\pi d^2 n)$ for HS gas molecules. Consequently, the Knudsen number ($Kn = \lambda/L$) is 0.01, and the flow can be considered to be in the continuum regime. Periodic boundary conditions are assumed in both directions. This means that, in the process of molecular movements, one molecule that gets through a specific boundary will re-enter the computational domain from the opposite boundary.

The whole computational domain is divided into 256×256 cells, and in each cell approximately 4000 simulation molecules are randomly distributed at the initial time, with one simulation molecule representing approximately 4×10^6 real molecules. The initial thermal velocities of the simulation molecules are sampled randomly from a Maxwell distribution function at 273 K. The computational time step is set to 0.1τ, where τ is the molecular mean collision time with the definition of $\tau = \lambda/\bar{c}$. Note that $\bar{c}$ is the molecular mean speed, i.e. $\bar{c} = \sqrt{8kT/\pi m}$, where k and m are the Boltzmann constant and molecular mass, respectively. For the sake of clarity, the length scale and the time scale are normalized by the mean free path λ and the mean collision time τ, respectively. Correspondingly, the velocity and kinematic viscosity coefficient can be normalized by λ/τ and λ^2/τ, respectively. In the following description, all the non-dimensional quantities are denoted with a superscript asterisk, for instance, y^* represents the non-dimensional coordinate in the vertical direction.

To simulate a velocity decay process caused by viscous shear stress, we impose a spatially periodic macroscopic velocity field with a form of $\boldsymbol{u}^* = u_0^*(1 - \cos(2\pi y^*/L^*))\boldsymbol{e_x}$ at the initial time, where $u_0^* = 0.13$, and $\boldsymbol{e_x}$ is the unit vector in the x direction.

https://doi.org/10.1017/jfm.2020.184 Published online by Cambridge University Press

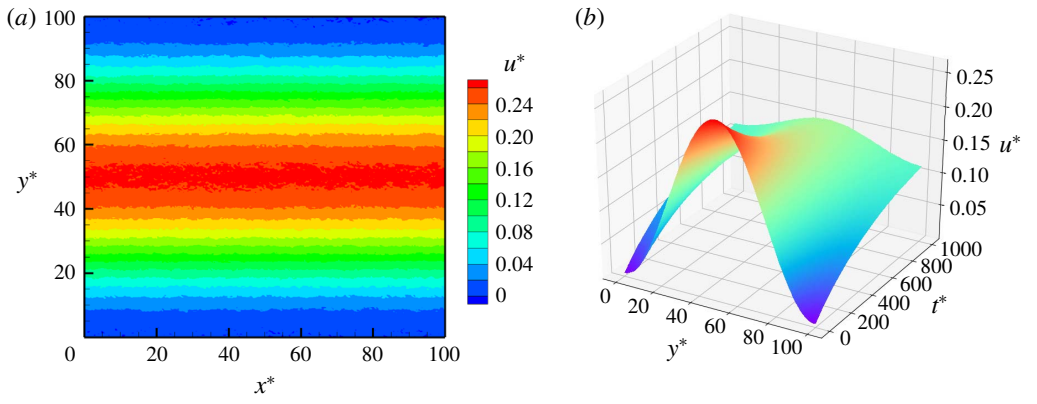

FIGURE 2. (*a*) The velocity contours at the initial time obtained by DSMC simulations; (*b*) temporal evolution of the velocity distribution along the vertical direction.

For each simulation molecule, its initial velocity is a sum of two parts, i.e. the imposed macroscopic velocity and the random thermal velocity. In the simulation process, we obtain the macroscopic velocities for each cell by sampling of the molecular velocities in a short time period (10 mean collision times) and output the velocities on-the-fly, that is, at the time instants $t^* = 0, 10, 20, \ldots, 1000$. Figure 2(*a*) shows the initial macroscopic velocity field generated in DSMC simulations. It can be seen from figure 2(*a*) that the flow is along the horizontal direction, and the amplitude of the horizontal velocity is uniform in the horizontal direction and only changes in the vertical direction. Note that fluctuations in the velocity field are caused by the molecular thermal motions, which are inevitable due to molecular thermal velocities in molecular simulations. To reduce the statistical fluctuations to an acceptable level, we make an average along the horizontal direction and focus on the changes in the vertical direction. Figure 2(*b*) shows the temporal evolution of the distribution of the horizontal velocity along the vertical direction. It can be seen that the initial distribution in terms of the cosine function gradually becomes smoother over time due to viscous dissipation.

We employ the values of the horizontal velocity at the discretized space–time points, i.e. 256 computational cells and 100 time instants, to construct the input dataset. Using the PDE-FIND method, we derive the governing equation in a parsimonious form, as shown in table 1. The coefficient of the term on the right-hand side is approximately 0.49 with a tolerance of ± 0.01. Note that, for the one-dimensional incompressible shear flow studied here, the non-dimensional Navier–Stokes equation can be simplified as follows:

$$\frac{\partial u^*}{\partial t^*} = \nu^* \frac{\partial^2 u^*}{\partial y^{*2}}, \tag{3.1}$$

where ν^* is the non-dimensional viscosity coefficient. According to the Chapman–Enskog theory, the viscosity coefficient for a HS gas at the first-order approximation reads as (Chapman & Cowling 1990)

$$\nu = \frac{5\pi}{32} \lambda \bar{c} = \frac{5\pi}{32} \frac{\lambda^2}{\tau}. \tag{3.2}$$

Therefore, the non-dimensional viscosity coefficient ν^* is $5\pi/32 \approx 0.49$. Comparing the derived governing equation with the theoretical counterpart in table 1, we can

https://doi.org/10.1017/jfm.2020.184 Published online by Cambridge University Press

Derived governing equation	$\dfrac{\partial u^*}{\partial t^*} = (0.49 \pm 0.01)\dfrac{\partial^2 u^*}{\partial y^{*2}}$
Theoretical momentum equation	$\dfrac{\partial u^*}{\partial t^*} = 0.49\dfrac{\partial^2 u^*}{\partial y^{*2}}$

TABLE 1. Governing equation for one-dimensional incompressible shear flow of argon gas.

conclude that the derived governing equation not only has the expected form determined from the theoretical counterpart, but also gives an accurate estimation of the viscous coefficient.

It should be noted that the PDE-FIND method is sensitive to noisy data, and the data generated by molecular simulations inevitably have statistical fluctuations or errors. According to the theory of statistical mechanics, the fractional error of the velocity is inversely proportional to the square root of the sampling size (Hadjiconstantinou *et al.* 2003), that is,

$$E_u \approx \frac{1}{\sqrt{N}Ma}, \tag{3.3}$$

where N is the sampling size, and Ma is the Mach number. Note that, in the preceding case of shear flow, there are approximately 4000 simulation molecules in each computational cell. To reduce fraction errors, we employ a short-time average (100 calculating time steps) and spatial average (256 computational cells in the horizontal direction) to obtain the temporal evolution of the velocity field in the vertical direction. Specifically, for each computational cell, the sampling size is $4000 \times 100 \times 256 \approx 10^8$, and thus the fraction error at the initial time ($Ma \approx 0.3$) according to (3.3) is approximately 0.03 %. As the simulation progresses, the macroscopic velocities and the Mach number decrease due to viscous dissipation, and hence the fraction error increases continuously. In this numerical case, the maximum fraction error of the viscosity coefficient in the derived equation is approximately 2 %, as shown in table 1.

We run three other cases of the shear flow with different sampling sizes of 10^5, 10^6 and 10^7 for each computational cell, and the fractional errors of the velocity fields at the initial time are 1.0 %, 0.3 % and 0.1 %, respectively. If the initial fractional error is as large as 1.0 %, we cannot obtain an expected governing equation. For the next two cases, the PDE-FIND method can derive governing equations in the right forms based on DSMC data, and the maximum fraction errors of the viscosity coefficient in the derived equations are 6.0 % and 4.0 %, respectively. It is demonstrated that the accuracy of the derived governing equation using the PDE-FIND method is quite sensitive to noisy data. To obtain an expected derived equation, the fraction errors of the data generated by the molecular simulations must be reduced to an acceptable level.

3.2. *Diffusion problem and diffusion equation*

The simulation model for diffusion is also at the standard conditions, and it has the same geometry and boundary conditions as those employed for the shear flow in § 3.1. The differences are that two species of gas are employed to mimic the diffusion process, and the initial macroscopic velocity in the computational domain is set to

https://doi.org/10.1017/jfm.2020.184 Published online by Cambridge University Press

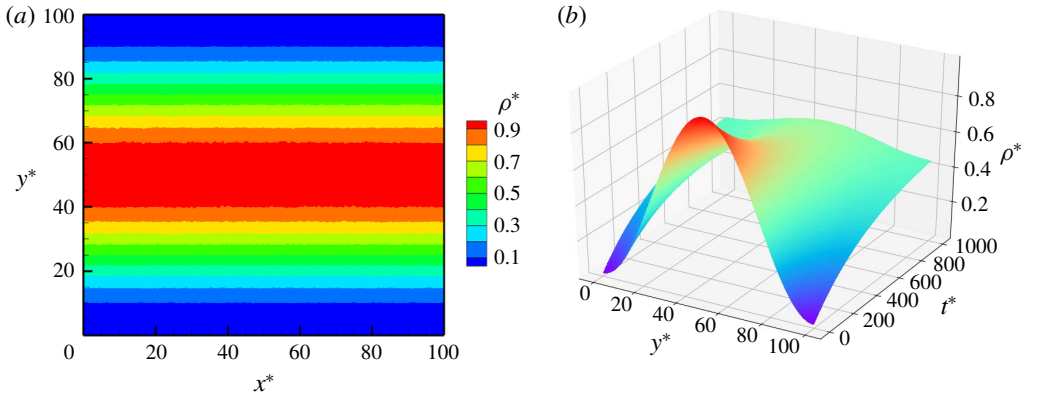

FIGURE 3. (*a*) The density contours of species-A at the initial time obtained by DSMC simulations; (*b*) temporal evolution of density distribution of species-A along the vertical direction.

Derived governing equation	$\dfrac{\partial \rho^*}{\partial t^*} = (0.59 \pm 0.01)\dfrac{\partial^2 \rho^*}{\partial y^{*2}}$
Theoretical diffusion equation	$\dfrac{\partial \rho^*}{\partial t^*} = 0.59\dfrac{\partial^2 \rho^*}{\partial y^{*2}}$

TABLE 2. Governing equation for the diffusion of argon gas.

zero uniformly. For the sake of simplicity, the two species are virtually like argon gas with the same molecular diameters, but are denoted as species-A and species-B. The initial distribution of non-dimensional densities for species-A and species-B are $\rho_A^* = 0.5(1 - \cos(2\pi y^*/L^*))$ and $\rho_B^* = 0.5(1 + \cos(2\pi y^*/L^*))$, respectively. Figure 3(*a*) shows the density contours of species-A at the initial time. In the whole simulation process, the macroscopic velocity is always zero and the total density of the two species remains uniform in the computational domain, while the respective densities of species-A and species-B change with space and time. Therefore, the simulation model can be considered as a pure diffusion problem.

To reduce the statistical fluctuations, we also make an average of the density along the horizontal direction. Figure 3(*b*) shows the temporal evolution of the density distribution of species-A along the vertical direction. It is shown that the initial density distribution tends to be evenly distributed as the simulation proceeds due to the diffusion mechanism. Using the density of species-A at the discretized 256 computational cells and 100 time instants as the input dataset, we employ the PDE-FIND method to derive the corresponding governing equation, as shown in table 2. The coefficient of the term on the right-hand side is approximately 0.59 with a tolerance of ± 0.01. It is known that the non-dimensional diffusion equation with a constant diffusion coefficient is,

$$\frac{\partial \rho^*}{\partial t^*} = D^* \frac{\partial^2 \rho^*}{\partial y^{*2}}. \tag{3.4}$$

https://doi.org/10.1017/jfm.2020.184 Published online by Cambridge University Press

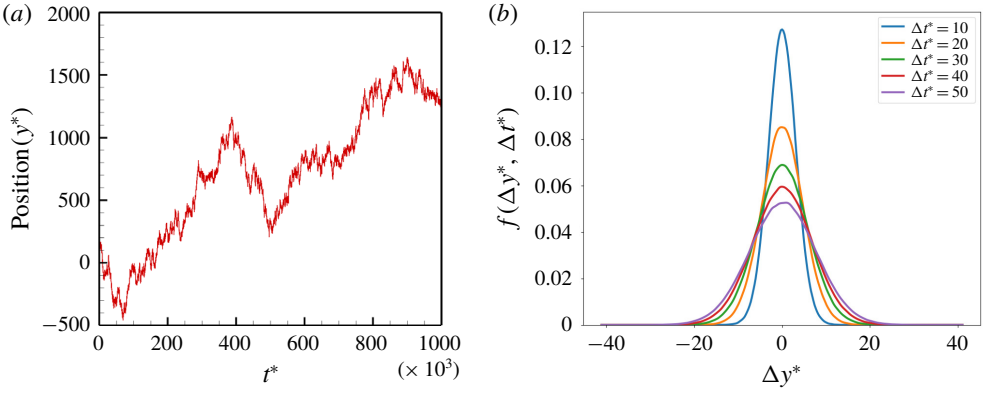

FIGURE 4. (*a*) A single trace of molecular motion; (*b*) probability density function of molecular displacements for different time intervals.

According to the Chapman–Enskog theory, the diffusion coefficient for a HS gas at the first-order approximation reads as (Chapman & Cowling 1990)

$$D = \frac{3\pi}{16}\lambda\bar{c} = \frac{3\pi}{16}\frac{\lambda^2}{\tau}. \tag{3.5}$$

Therefore, the non-dimensional diffusion coefficient is $3\pi/16 \approx 0.59$. It can be concluded from the results shown in table 2 that the form of the derived governing equation for the diffusion problem agrees well with the theoretical diffusion equation, and gives an accurate estimation of the diffusion coefficient.

The advantage of molecular simulations like the DSMC method is that they not only can obtain the macroscopic quantities by sampling and averaging, but also provide molecular information such as the velocities and positions in detail. Our previous studies (Zhang *et al.* 2010; Zhang & Önskog 2017; Zhang *et al.* 2019*b*) have demonstrated that on a time scale larger than the molecular mean collision time, the characteristics of molecular motion conform to Brownian motion. Therefore, it is intriguing to investigate whether the diffusion equation can be derived directly from molecular movements. To this end, in the simulation case of the diffusion problem, we randomly select 100 representatives from all the simulation molecules and record their positions every 10τ. Figure 4(*a*) shows one representative simulation molecule's trajectory, that is, the position versus time, which seems like Brownian motion qualitatively.

In order to analyse the characteristics of molecular motions quantitatively, we first determine the molecular displacements Δy^* over any time interval Δt^* based on 100 trajectories of selected simulation molecules, and then we obtain the probability density function of displacements by taking a statistical average for a specific time interval, i.e. $f(\Delta y^*, \Delta t^*)$, as shown in figure 4(*b*). It can be seen that the probability density functions conform to a normal distribution, and their variance increases with the time interval. These are virtually the typical characteristics of Brownian motion.

Using $f(\Delta y^*, \Delta t^*)$ as the input dataset, we further employ the PDE-FIND method to derive the governing equation of the probability density function, as shown in table 3. Note that the mathematical symbol (delta) in front of y^* and t^* is dismissed for the sake of simplicity. According to the theory of stochastic processes, the Brownian

https://doi.org/10.1017/jfm.2020.184 Published online by Cambridge University Press

Derived governing equation	$\dfrac{\partial f}{\partial t^*} = (0.59 \pm 0.01)\dfrac{\partial^2 f}{\partial y^{*2}}$
Simplified Fokker–Planck equation	$\dfrac{\partial f}{\partial t^*} = 0.59\dfrac{\partial^2 f}{\partial y^{*2}}$

TABLE 3. Governing equation for the molecular diffusion of argon gas.

motion can be described by the Wiener process using a Langevin-type equation, or equivalently, by a Fokker–Planck-type equation in terms of the probability density function. Specifically, the non-dimensional Fokker–Planck equation for Brownian motion with a constant diffusion coefficient and without any macroscopic velocities reads as,

$$\frac{\partial f}{\partial t^*} = D^* \frac{\partial^2 f}{\partial y^{*2}}, \tag{3.6}$$

where D^* is the non-dimensional diffusion coefficient, which has the same physical meaning as that in the diffusion equation and has the value of 0.59 for HS gas molecules. Comparing the two equations shown in table 3, we can conclude that the molecular motions in DSMC, over a time scale larger than the molecular collision time, can be described using a Fokker–Planck-type equation, with a well-defined diffusion coefficient based on kinetic theory.

3.3. *Taylor–Green vortex and vorticity transport equation*

The above two cases in §§ 3.1 and 3.2 are virtually one-dimensional as there are no gradients in the horizontal direction, and in the following we study one real two-dimensional problem of fluid dynamics. It is well known that, for two-dimensional incompressible viscous flow, the non-dimensional Navier–Stokes equations read as,

$$\left.\begin{aligned} \frac{\partial u^*}{\partial x^*} + \frac{\partial v^*}{\partial y^*} &= 0, \\ \frac{\partial u^*}{\partial t^*} + u^*\frac{\partial u^*}{\partial x^*} + v^*\frac{\partial u^*}{\partial y^*} &= -\frac{\partial p^*}{\partial x^*} + \nu^*\left(\frac{\partial^2 u^*}{\partial x^{*2}} + \frac{\partial^2 u^*}{\partial y^{*2}}\right), \\ \frac{\partial v^*}{\partial t^*} + u^*\frac{\partial v^*}{\partial x^*} + v^*\frac{\partial v^*}{\partial y^*} &= -\frac{\partial p^*}{\partial y^*} + \nu^*\left(\frac{\partial^2 v^*}{\partial x^{*2}} + \frac{\partial^2 v^*}{\partial y^{*2}}\right). \end{aligned}\right\} \tag{3.7}$$

Note that the length scale and the time scale are also normalized by the mean free path and the mean collision time, respectively; u^* and v^* are the non-dimensional velocities in the horizontal and vertical directions, respectively. By defining the vorticity $\overrightarrow{\omega} = \boldsymbol{\nabla} \times \overrightarrow{v}$ and taking the curl of the Navier–Stokes equations, we can obtain the non-dimensional vorticity equation for two-dimensional incompressible flows as follows:

$$\frac{\partial \omega_z^*}{\partial t^*} + u^*\frac{\partial \omega_z^*}{\partial x^*} + v^*\frac{\partial \omega_z^*}{\partial y^*} = \nu^*\left(\frac{\partial^2 \omega_z^*}{\partial x^{*2}} + \frac{\partial^2 \omega_z^*}{\partial y^{*2}}\right), \tag{3.8}$$

where ω_z^* is the non-dimensional vorticity component in the z direction, and $\omega_x^* = \omega_y^* = 0$. In the community of fluid dynamics, the Taylor–Green vortex is

https://doi.org/10.1017/jfm.2020.184 Published online by Cambridge University Press

widely used as a benchmark case in validating solvers and formulations in numerical computations, as it has an exact analytical solution to (3.7) and (3.8), that is,

$$\left.\begin{aligned} u^* &= U_0^* \cos(2\pi x^*/L_x^*)\sin(2\pi y^*/L_y^*)\exp(-2v^*t^*), \\ v^* &= -U_0^* \sin(2\pi x^*/L_x^*)\cos(2\pi y^*/L_y^*)\exp(-2v^*t^*). \end{aligned}\right\} \tag{3.9}$$

Here, we employ the DSMC method to simulate argon gas flow with initial conditions provided by the Taylor–Green vortex, and then we utilize the dataset obtained by DSMC to derive the underlying governing equation. Specifically, we simulate argon gas flow in a square of side length $L_x^* = L_y^* = 100$, and periodic boundary conditions are applied for all the boundaries. The initial macroscopic velocity is given by (3.9) at $t^* = 0$, and we choose $U_0^* = 0.08$ to ensure that the gas flow conforms to the assumption of incompressibility. The initial reference state is at standard conditions with uniform density and a small variation of pressure as $p^* = p_0^* - (\rho_0^* U_0^{*2})/4(\cos(4\pi x^*/L_x^*) + \cos(4\pi y^*/L_y^*))$, to ensure that the initial conditions completely satisfy the solution of (3.7).

The whole computational domain is divided into 64×64 sampling cells, and each cell has approximately 4×10^5 simulation molecules. The macroscopic quantities such as velocities are obtained by sampling the molecular velocities in the sampling cells. Each sampling cell is divided into enough sub-cells within which collision pairs are selected, and the sizes of the sub-cells are guaranteed to be less than the mean free path. We record the velocity field every 10 mean collision times (100 computational time steps) and obtain the vorticity field by taking the curl of the velocity vectors. To reduce the statistical errors, 10 independent runs with different random number sequences are performed to make an ensemble average. Figure 5 shows the contours of vorticity at $t^* = 0$, 100, 200, 300 obtained by DSMC. At the initial time ($Ma \approx 0.1$), the fraction error of the velocities according to (3.3) is approximately 0.05 %. As shown in figure 5(*a*), our DSMC results are consistent with the theoretical results shown by the solid black lines, which are determined by (3.9). Due to viscous dissipation, the magnitude of the velocities and hence the Mach number would decrease as the simulation progresses. Consequently, the fractional errors increase continuously, as shown in figures 5(*b*), 5(*c*) and 5(*d*). Basically, our DSMC results agree well with theoretical results, and the maximum fraction error is less than 1 %.

Using the velocity and vorticity fields at 64×64 sampling cells and 65 time instants obtained by DSMC as the input dataset, we derive the governing equation shown in table 4. At first glance, the derived governing equation lacks two convective terms on the left-hand side compared with the theoretical vorticity transport equation shown in table 4. On the other hand, the two viscous terms on the right-hand side of the derived equation have the same forms as those in the theoretical equation, and their coefficients are quite close. Virtually, the derived governing equation is not wrong, but just in a simplified form for the specific problem of the Taylor–Green vortex.

It should be mentioned that the DSMC results are consistent with the exact analytical solution, equation (3.9), for the Taylor–Green vortex, except that the DSMC results are somewhat noisy. Substituting the velocity and vorticity fields determined by (3.9) into the two convective terms $u^*(\partial\omega_z^*/\partial x^*)$ and $v^*(\partial\omega_z^*/\partial y^*)$, it is obvious that their sum is always zero. In this case, they are automatically eliminated in the derived governing equation, since the PDE-FIND method aims to find the most parsimonious form for the underlying governing equation.

https://doi.org/10.1017/jfm.2020.184 Published online by Cambridge University Press

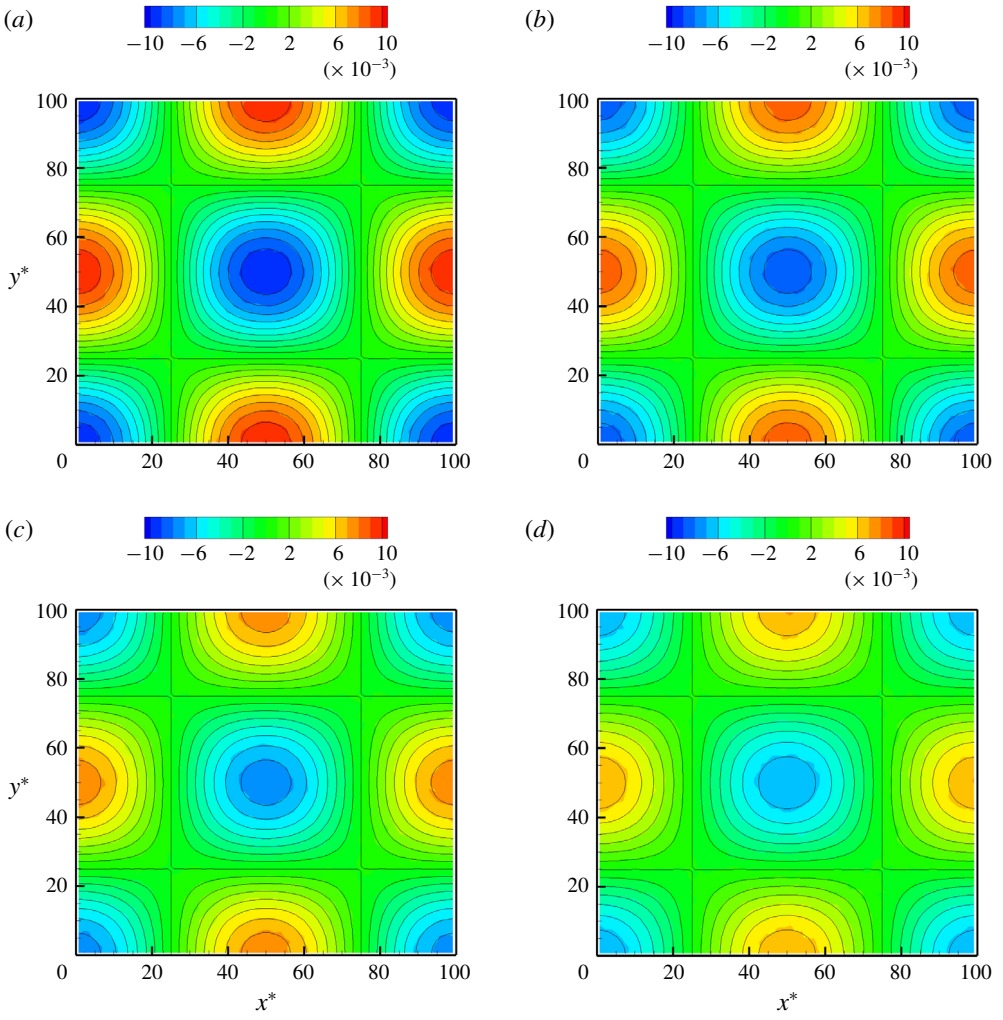

FIGURE 5. Contours of vorticity for the Taylor–Green vortex obtained by the DSMC method at different time instants: (*a*) $t^* = 0$; (*b*) $t^* = 100$; (*c*) $t^* = 200$; (*d*) $t^* = 300$. The solid black lines represent the theoretical solutions of the Taylor–Green vortex.

In order to check whether the data generated by DSMC can derive the complete vorticity transport equation, we simulate another case with an artificial initial condition as follows:

$$\left.\begin{aligned} u^* &= U_0^* \cos(4\pi x^*/L_x^*)\sin(2\pi y^*/L_y^*), \\ v^* &= -2U_0^* \sin(4\pi x^*/L_x^*)\cos(2\pi y^*/L_y^*). \end{aligned}\right\} \tag{3.10}$$

Note that the wavenumber in the horizontal direction in (3.10) is double that of the standard Taylor–Green vortex. In order to satisfy the continuity equation of fluid dynamics, the amplitude of the velocity in the vertical direction is also doubled. The simulation domain is a square of side length $L_x^* = L_y^* = 200$, and other computation parameters are the same as those in the Taylor–Green vortex.

https://doi.org/10.1017/jfm.2020.184 Published online by Cambridge University Press

Derived equation for Taylor–Green vortex	$\dfrac{\partial \omega_z^*}{\partial t^*} = (0.49 \pm 0.03)\left(\dfrac{\partial^2 \omega_z^*}{\partial x^{*2}} + \dfrac{\partial^2 \omega_z^*}{\partial y^{*2}}\right)$
Derived equation for artificial vortex	$\dfrac{\partial \omega_z^*}{\partial t^*} + (1.01 \pm 0.06)u^* \dfrac{\partial \omega_z^*}{\partial x^*} + (1.01 \pm 0.06)v^* \dfrac{\partial \omega_z^*}{\partial y^*} = (0.49 \pm 0.03)\left(\dfrac{\partial^2 \omega_z^*}{\partial x^{*2}} + \dfrac{\partial^2 \omega_z^*}{\partial y^{*2}}\right)$
Theoretical vorticity transport equation	$\dfrac{\partial \omega_z^*}{\partial t^*} + u^* \dfrac{\partial \omega_z^*}{\partial x^*} + v^* \dfrac{\partial \omega_z^*}{\partial y^*} = 0.49\left(\dfrac{\partial^2 \omega_z^*}{\partial x^{*2}} + \dfrac{\partial^2 \omega_z^*}{\partial y^{*2}}\right)$

TABLE 4. Governing equation for the diffusion of argon gas.

Figure 6 shows the contours of the vorticity field obtained by DSMC for the case with the artificial initial condition defined by (3.10) at $t^* = 0$, 100, 200, 300. Using the dataset generated by DSMC, we also derive the governing equation, as shown in table 4. It has complete terms, including convective and viscous terms, as with the theoretical equation. The coefficients of the convective terms in the horizontal and vertical directions are 1.01 with a tolerance of ± 0.06, while the coefficients of the viscous terms are 0.49 with a tolerance of ± 0.03. The maximum relative error is approximately 6 % compared with the theoretical value. Considering that the DSMC results inevitably have noise as DSMC is a statistical simulation method, the prediction of the coefficients is acceptable. It is expected that we can get more accurate coefficients if we have larger sampling sizes, but this needs more computational resources.

Note that, in this work, we employ the simulation cases with an initial condition in terms of the Taylor–Green vortex and its variant to derive the vorticity transport equation. Essentially, the discovery of the vorticity transport equation is not limited to these special cases, and it can be realized as long as the flow problem is able to provide the spatial-temporal evolution of the velocity and vorticity fields, such as flow around a cylinder with shedding vortices.

4. Conclusions

In this work, we employed the DSMC method on the molecular level to simulate three benchmark cases of fluid dynamics and obtained the data of the spatial-temporal evolution of the flow fields. The generated data are used to derive the macroscopic governing equations via the PDE-FIND method. Our simulation results of shear flow, diffusion problem and the Taylor–Green vortex obtained by DSMC are successfully applied to discover the momentum equation, diffusion equation and vorticity transport equation, respectively. For the diffusion problem, we also demonstrate that it is possible to derive the macroscopic equation using molecular information, such as the trajectories of the molecules instead of macroscopic quantities. The equations derived by the data-driven discovery method not only have the same form as the theoretical ones, but also provide accurate predictions of the transport coefficients contained in the governing equations.

This work provides strong proof that microscopic molecular movements and macroscopic flow phenomena governed by the underlying macroscopic equations have a close relationship, in terms of data-driven discovery. It should be noted that

https://doi.org/10.1017/jfm.2020.184 Published online by Cambridge University Press

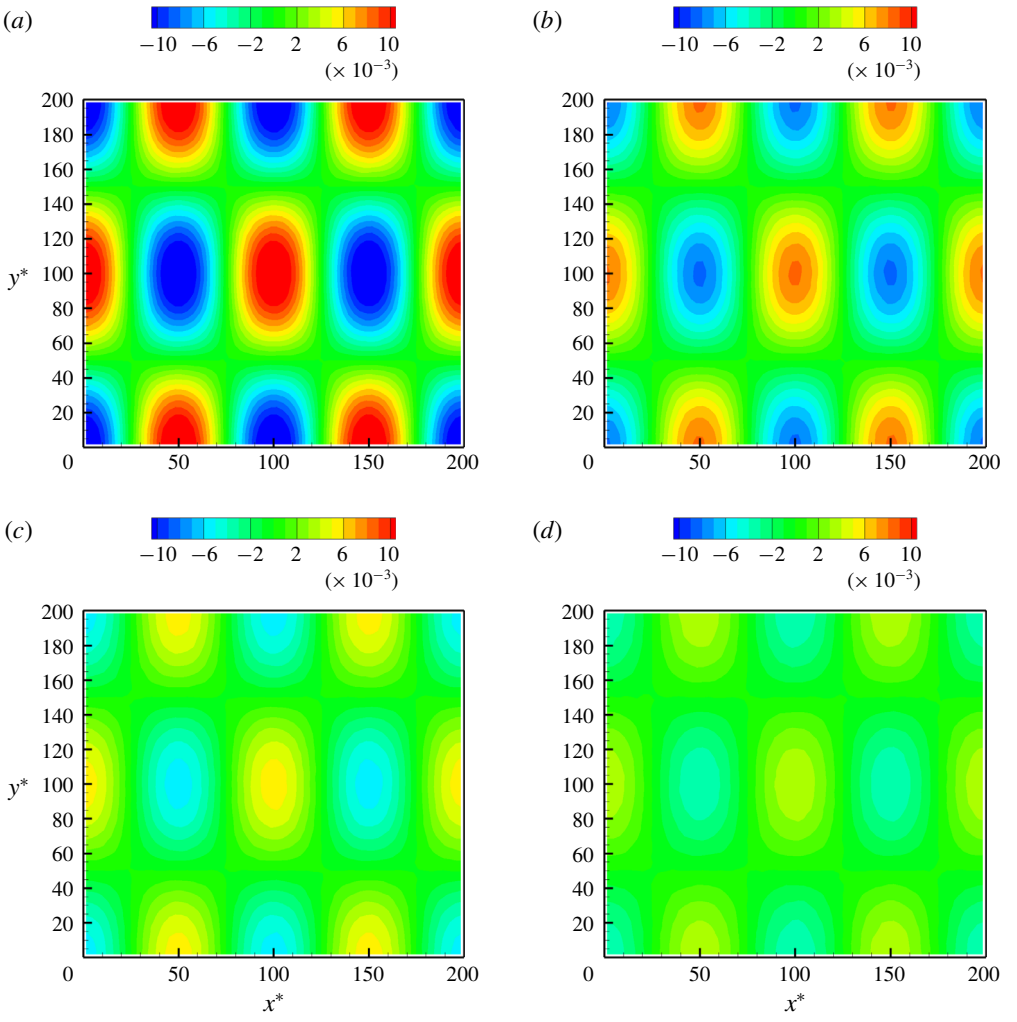

FIGURE 6. Contours of vorticity fields for the case with artificial initial conditions obtained by the DSMC method at different time instants: (*a*) $t^* = 0$; (*b*) $t^* = 100$; (*c*) $t^* = 200$; (*d*) $t^* = 300$.

we have only focused on simple flow problems where the theoretical governing equations are well established so far, but the strategy proposed in this work can be extended to more complex problems such as rarefied gas flows, where the validity of the conventional Navier–Stokes equations is questionable and a variety of higher-order equation sets have been proposed. However, no single higher-order equation set has demonstrated universal superiority in the prediction of rarefied gas flows, especially for the capture of the Knudsen layer behaviour (Lockerby, Reese & Gallis 2005). Due to the complexity of the Knudsen layer and boundary conditions, the application of data-driven discovery of the governing equations to rarefied gas flows would be quite challenging. Research work in this direction is expected to be carried out in the future.

https://doi.org/10.1017/jfm.2020.184 Published online by Cambridge University Press

Acknowledgements

We thank S. L. Brunton and S. H. Rudy for providing the code of PDE-FIND method and stimulating discussions. This work was supported by the National Natural Science Foundation of China (grant no. 11772034). Results were obtained using the Tianhe-2 supercomputer.

Declaration of interests

The authors report no conflict of interest.

REFERENCES

ALEXANDER, F. J., GARCIA, A. L. & ALDER, B. J. 1998 Cell size dependence of transport coefficients in stochastic particle algorithms. *Phys. Fluids* **10** (6), 1540–1542.

BATCHELOR, G. K. 2000 *An Introduction to Fluid Dynamics*. Cambridge University Press.

BIRD, G. A. 1994 *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*. Clarendon Press.

BONGARD, J. & LIPSON, H. 2007 Automated reverse engineering of nonlinear dynamical systems. *Proc. Natl Acad. Sci. USA* **104** (24), 9943–9948.

BRUNTON, S. L. & KUTZ, J. N. 2019 *Data-driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press.

BRUNTON, S. L., NOACK, B. R. & KOUMOUTSAKOS, P. 2020 Machine learning for fluid mechanics. *Annu. Rev. Fluid Mech.* **52**, 477–508.

BRUNTON, S. L., PROCTOR, J. L. & KUTZ, J. N. 2016 Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proc. Natl Acad. Sci. USA* **113** (15), 3932–3937.

CHAPMAN, S. & COWLING, T. G. 1990 *The Mathematical Theory of Non-uniform Gases*. Cambridge University Press.

DANIELS, B. C. & NEMENMAN, I. 2015 Automated adaptive inference of phenomenological dynamical models. *Nat. Commun.* **6**, 8133.

DURAISAMY, K., IACCARINO, G. & XIAO, H. 2019 Turbulence modeling in the age of data. *Annu. Rev. Fluid Mech.* **51**, 357–377.

GALLIS, M. A., BITTER, N. P., KOEHLER, T. P., TORCZYNSKI, J. R., PLIMPTON, S. J. & PAPADAKIS, G. 2017 Molecular-level simulations of turbulence and its decay. *Phys. Rev. Lett.* **118** (6), 064501.

GARCIA, A. L. & WAGNER, W. 2000 Time step truncation error in direct simulation Monte Carlo. *Phys. Fluids* **12** (10), 2621–2633.

GONZALEZ, G. R., RICO, M. R. & KEVREKIDIS, I. G. 1998 Identification of distributed parameter systems: a neural net based approach. *Comput. Chem. Engng* **22**, S965–S968.

GU, X. J., BARBER, R. W., JOHN, B. & EMERSON, D. R. 2019 Non-equilibrium effects on flow past a circular cylinder in the slip and early transition regime. *J. Fluid Mech.* **860**, 654–681.

GU, X. J. & EMERSON, D. R. 2009 A high-order moment approach for capturing non-equilibrium phenomena in the transition regime. *J. Fluid Mech.* **636**, 177–216.

HADJICONSTANTINOU, N. G., GARCIA, A. L., BAZANT, M. Z. & HE, G. 2003 Statistical error in particle simulations of hydrodynamic phenomena. *J. Comput. Phys.* **187** (1), 274–297.

HEY, J. G., TANSLEY, S. & TOLLE, K. M. 2009 *The Fourth Paradigm: Data-intensive Scientific Discovery*, vol. 1. Microsoft Research.

JORDAN, M. I. & MITCHELL, T. M. 2015 Machine learning: trends, perspectives, and prospects. *Science* **349** (6245), 255–260.

KEVREKIDIS, I. G., GEAR, C. W., HYMAN, J. M., KEVREKIDID, P. G., RUNBORG, O. & THEODOROPOULOS, C. 2003 Equation-free, coarse-grained multiscale computation: enabling microscopic simulators to perform system-level analysis. *Commun. Math. Sci.* **1** (4), 715–762.

https://doi.org/10.1017/jfm.2020.184 Published online by Cambridge University Press

LOCKERBY, D. A., REESE, J. M. & GALLIS, M. A. 2005 Capturing the Knudsen layer in continuum-fluid models of nonequilibrium gas flows. *AIAA J.* **43** (6), 1391–1393.

LOISEAU, J.-C. & BRUNTON, S. L. 2018 Constrained sparse Galerkin regression. *J. Fluid Mech.* **838**, 42–67.

MANELA, A. & ZHANG, J. 2012 The effect of compressibility on the stability of wall-bounded Kolmogorov flow. *J. Fluid Mech.* **694**, 29–49.

MARX, V. 2013 The big challenges of big data. *Nature* **498**, 255–260.

ORAN, E. S., OH, C. K. & CYBYK, B. Z. 1998 Direct simulation Monte Carlo: recent advances and applications. *Annu. Rev. Fluid Mech.* **30** (1), 403–441.

RUDY, S., ALLA, A., BRUNTON, S. L. & KUTZ, J. N. 2019 Data-driven identification of parametric partial differential equations. *SIAM J. Appl. Dyn. Syst.* **18** (2), 643–660.

RUDY, S. H., BRUNTON, S. L., PROCTOR, J. L. & KUTZ, J. N. 2017 Data-driven discovery of partial differential equations. *Sci. Adv.* **3** (4), e1602614.

SCHAEFFER, H. 2017 Learning partial differential equations via data discovery and sparse optimization. *Proc. R. Soc. Lond.* A **473** (2197), 20160446.

SCHMIDT, M. & LIPSON, H. 2009 Distilling free-form natural laws from experimental data. *Science* **324** (5923), 81–85.

STEFANOV, S., ROUSSINOV, V. & CERCIGNANI, C. 2002 Rayleigh–Bénard flow of a rarefied gas and its attractors. I. Convection regime. *Phys. Fluids* **14** (7), 2255–2269.

STRUCHTRUP, H. 2005 *Macroscopic Transport Equations for Rarefied Gas Flows*. Springer.

STRUCHTRUP, H. & TORRILHON, M. 2003 Regularization of Grads 13 moment equations: derivation and linear analysis. *Phys. Fluids* **15** (9), 2668–2680.

SUN, Q. H. & BOYD, I. D. 2002 A direct simulation method for subsonic, microscale gas flows. *J. Comput. Phys.* **179** (2), 400–425.

TIBSHIRANI, R. 1996 Regression shrinkage and selection via the lasso. *J. R. Stat. Soc.* B **58** (1), 267–288.

WAGNER, W. 1992 A convergence proof for Bird's direct simulation Monte Carlo method for the Boltzmann equation. *J. Stat. Phys.* **66** (3–4), 1011–1044.

ZHANG, J. & FAN, J. 2009 Monte Carlo simulation of thermal fluctuations below the onset of Rayleigh–Bénard convection. *Phys. Rev.* E **79** (5), 056302.

ZHANG, J., FAN, J. & FEI, F. 2010 Effects of convection and solid wall on the diffusion in microscale convection flows. *Phys. Fluids* **22** (12), 122005.

ZHANG, J., JOHN, B., PFEIFFER, M., FEI, F. & WEN, D. S. 2019*a* Particle-based hybrid and multiscale methods for nonequilibrium gas flows. *Adv. Aerodyn.* **1** (1), 12.

ZHANG, J. & ÖNSKOG, T. 2017 Langevin equation elucidates the mechanism of the Rayleigh–Bénard instability by coupling molecular motions and macroscopic fluctuations. *Phys. Rev.* E **96** (4), 043104.

ZHANG, J., TIAN, P., YAO, S. Q. & FEI, F. 2019*b* Multiscale investigation of Kolmogorov flow: from microscopic molecular motions to macroscopic coherent structures. *Phys. Fluids* **31** (8), 082008.

https://doi.org/10.1017/jfm.2020.184 Published online by Cambridge University Press