

Dimensional homogeneity constrained gene expression programming for discovering governing equations

Wenjun Ma¹, Jun Zhang^{1,†}, Kaikai Feng¹, Haoyun Xing¹ and Dongsheng Wen¹

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

(Received 10 April 2023; revised 24 January 2024; accepted 12 March 2024)

Data-driven discovery of governing equations is of great significance for helping us understand intrinsic mechanisms and build physical models. Recently, numerous highly innovative algorithms have emerged, aimed at inversely discovering the underlying governing equations from data, such as sparse regression-based methods and symbolic regression-based methods. Along this direction, a novel dimensional homogeneity constrained gene expression programming (DHC-GEP) method is proposed in this work. The DHC-GEP simultaneously discovers the forms and coefficients of functions using basic mathematical operators and physical variables, without requiring preassumed candidate functions. The constraint of dimensional homogeneity is capable of filtering out the overfitting equations effectively. The key advantages of DHC-GEP compared with the original gene expression programming, including being more robust to hyperparameters, the noise level and the size of datasets, are demonstrated on two benchmark studies. Furthermore, DHC-GEP is employed to discover the unknown constitutive relations of two representative non-equilibrium flows. Galilean invariance and the second law of thermodynamics are imposed as constraints to enhance the reliability of the discovered constitutive relations. Comparisons, both quantitative and qualitative, indicate that the derived constitutive relations are more accurate than the conventional Burnett equations in a wide range of Knudsen numbers and Mach numbers, and are also applicable to the cases beyond the parameter space of the training data.

Key words: machine learning

1. Introduction

Machine-learning-assisted modelling has become a new paradigm of research in a variety of scientific and engineering disciplines (Brunton, Noack & Koumoutsakos 2020;

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

Weinan 2021). A very incomplete list includes Ling, Kurzwaski & Templeton (2016), Koch-Janusz & Ringel (2018), Bergen *et al.* (2019), Sengupta *et al.* (2020), Ma, Zhang & Yu (2021), Guastoni *et al.* (2021), Park & Choi (2021), Karniadakis *et al.* (2021), Yu *et al.* (2022) and Juniper (2023), the vast majority of which present improved performance, but a prominent critique is that the resulting models are ‘black boxes’. They cannot be explicitly expressed in mathematical forms. This not only sacrifices interpretability but also makes the resulting models difficult to disseminate between end users (Beetham & Capece-laturo 2020).

In contrast to the aforementioned studies, Bongard & Lipson (2007) and Schmidt & Lipson (2009) proposed using stratified symbolic regression and genetic programming to discover governing equations from data in low-dimensional systems. While this approach is innovative, it is difficult to scale up to high-dimensional systems. Genetic programming encodes equations using the nonlinear parse trees with varying sizes and shapes, which can become bloated in high-dimensional problems, making evolutions computationally expensive (Vaddireddy *et al.* 2020). Subsequently, Brunton, Proctor & Kutz (2016) proposed a seminal work called sparse identification of nonlinear dynamics (SINDy), which employs sparse regression to identify the most informative subset from a large predetermined library of candidate functions and determines the corresponding coefficients. The SINDy is advantageous in deriving models that are explicit and concise, and has been widely used to discover equations in the form of first-order ordinary differential equations, alternatively with linear embedding (Lusch, Kutz & Brunton 2018; Champion *et al.* 2019a), for applications in fluid systems (Loiseau & Brunton 2018; Loiseau, Noack & Brunton 2018; Zhang & Ma 2020), predictive control of nonlinear dynamics (Kaiser, Kutz & Brunton 2018) and multi-time-scale systems (Champion, Brunton & Kutz 2019b). To date, there have been many extensions to SINDy, such as partial differential equation functional identification of nonlinear dynamic (Rudy *et al.* 2017; Schaeffer 2017), implicit sparse regression (Mangan *et al.* 2016; Kaheman, Kutz & Brunton 2020), physics-constrained sparse regression (Loiseau & Brunton 2018) and sparse relaxed regularized regression (Zheng *et al.* 2018). Basically, the sparse regression-based methods are confronted with two issues: the difficulty in accurately computing the derivative of noisy data and the requirement that all variables in the equation are observable. Gurevich, Reinbold & Grigoriev (2019), Reinbold, Gurevich & Grigoriev (2020) and Alves & Fiuza (2022) employed the weak formulation of differential equations to decrease the noise sensitivity and eliminate the dependence on unobservable variables. Furthermore, Reinbold & Grigoriev (2019), Reinbold *et al.* (2021) and Gurevich, Reinbold & Grigoriev (2021) considered three appropriate physical constraints, including locality, smoothness and symmetries, to dramatically constrain the size of candidate library to be concise and effective.

More recently, in addition to sparse regression-based methods, two promising categories of data-driven methods have been proposed for discovering explicit models. The first category is the neural network-based method, such as PDE-Net (Long *et al.* 2018; Long, Lu & Dong 2019) and equation learner (EQL) (Sahoo, Lampert & Martius 2018). The PDE-Net approximates differential operators with convolutions and employs a symbolic multilayer neural network for model recovery, resulting in high expressivity and flexibility. The EQL uses a special neural network structure whose activation functions are symbolic operators, and it was demonstrated that the derived models can be generalized to the parameter spaces not covered by the training dataset. However, both neural network-based methods are criticized for the resulting equations being overly complex.

Another category is gene expression programming (GEP)-based methods (Ferreira 2001; Vaddireddy *et al.* 2020; Xing *et al.* 2022), which learn the forms of functions and

their corresponding coefficients concurrently. The preselected elements for GEP include only mathematical operators, physical constants and physical variables. The resulting equations are constructed by randomly combining basic elements while satisfying the syntactic requirements of the mathematical expression, rather than by linearly combining the predetermined candidate functions. Unlike the genetic programming method used in Schmidt & Lipson (2009), GEP encodes equations with fixed length linear strings that have unfixed open reading frames (ORFs). This feature of fixed length prevents bloating issues and excessive computational costs when dealing with complex problems. Additionally, variable ORFs ensure the variety of expression products, thus providing strong expressivity. Moreover, GEP performs a global exploration in the space of mathematical expressions, tending to obtain good results in a reasonable time. Therefore, in terms of data fitting, GEP has almost all the advantages of the aforementioned methods, featuring explicitness, concision, enhanced expressivity and flexibility without assuming the forms of candidate functions. Generally, GEP endeavours to discover the equations with less error; however, dimensional homogeneity cannot be guaranteed, particularly for problems with a variety of variables. Without any constraints or assumptions on the function forms, it is highly probable to obtain some overfitting and unphysical equations, which are sensitive to hyperparameters.

Considering the pros and cons of GEP, we propose a novel dimensional homogeneity constrained GEP (DHC-GEP) method for discovering governing equations. To the best of our knowledge, this is the first time that the constraint of dimensional homogeneity has been introduced to GEP. The constraint is implemented via a dimensional verification process before evaluating loss, without altering the fundamental features of the original GEP (Original-GEP) (Ferreira 2001), including the structure of chromosomes, the rules of expression, selection and genetic operators. Therefore, DHC-GEP inherits all the advantages of Original-GEP. More importantly, through two benchmark studies, we demonstrate that DHC-GEP has three critical improvements over Original-GEP: (a) robust to the size and noise level of datasets; (b) less sensitive to hyperparameters; (c) lower computational costs.

Furthermore, we extend the application of DHC-GEP to discover the unknown constitutive relations for non-equilibrium flows, including one-dimensional shock wave and rarefied Poiseuille flow. The conventional governing equations for fluid flows are the Navier–Stokes–Fourier (NSF) equations, which are derived based on the conservation of mass, momentum and energy, as well as the empirical assumptions of linear constitutive relations for the viscous stress and heat flux. Note that in strong non-equilibrium flows, these linear constitutive relations breakdown, and thus NSF equations are no longer applicable. Although high-order constitutive relations can be derived based on kinetic theory (Chapman & Cowling 1990), such as Burnett equations (Burnett 1936), their applicability is still very limited. Instead, our data-driven strategy is to derive the unknown constitutive relations from the data generated with molecular simulations. A general flowchart is shown in figure 1. Considering that constitutive relations describe the local transport mechanisms of momentum and energy, we regard local non-equilibrium parameters as key factors, and meticulously select the imported variables to satisfy the Galilean invariance (Han *et al.* 2019; Huang *et al.* 2021). Besides, the constraint of the second law of thermodynamics is embedded by adding an additional loss term, which is related to entropy production, to the loss function. The derived equations are more accurate than conventional Burnett equations over a wide range of Knudsen number and Mach number.

The remainder of this paper is organized as follows. In § 2.1, we briefly introduce the molecular simulation method, i.e. the direct simulation Monte Carlo (DSMC) method.

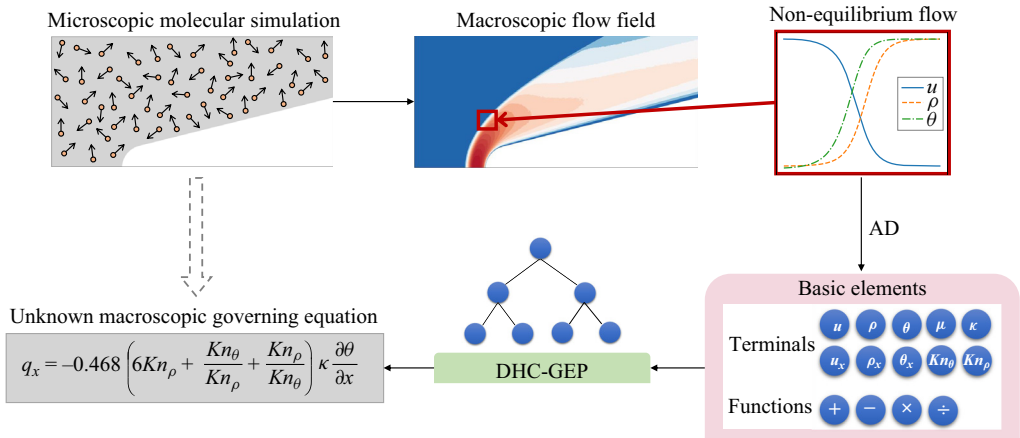


Figure 1. Flowchart of discovering unknown governing equations from data. First, macroscopic flow field data is generated via microscopic molecular simulation. Then, the basic elements preselected for DHC-GEP are constituted with state variables, mathematical operators and other variables of concern. Taking the one-dimensional shock wave case as an example, the terminal set involves fundamental physical parameters (viscosity (μ), heat conductivity (κ)), state variables (velocity (u), density (ρ), temperature (θ)), gradient terms (u_x, ρ_x, θ_x) and gradient-length local (GLL) Knudsen number (Kn_ρ, Kn_θ). The target variable is the heat flux in the x direction (q_x). The gradient terms are computed with auto differentiation (AD) based on neural networks, which is introduced in [Appendix A](#). Finally, based on the terminal and function sets, DHC-GEP conducts a global search in the space of mathematical expressions until a satisfactory equation that fits well with the target variable is obtained.

In § 2.2, the DHC-GEP method is introduced in detail. In § 3, we demonstrate the improved performance of DHC-GEP on two benchmark studies. Then, in § 4, we extend its applications to discovering unknown constitutive relations for two non-equilibrium flows. Conclusions and discussions are drawn in § 6.

2. Methodology

2.1. Direct simulation Monte Carlo

Generally, the Knudsen number (Kn), which is defined as the ratio of the molecular mean free path to the characteristic length scale of system, is used to classify various flow regimes, with the continuum regime being in the range of $Kn \leq 0.01$, the slip regime being $0.01 < Kn \leq 0.1$, the transition regime being $0.1 < Kn \leq 10$ and the free molecular flow being $Kn > 10$. While the computational fluid dynamics (CFD) methods based on the NSF equations have been successfully applied to the continuum regime and can also be used for the slip regime with appropriate slip boundary conditions, they cannot accurately simulate transitional and free molecular gas flows. On the contrary, the Boltzmann equation can describe gas flows in all flow regimes, which can be written as

$$\frac{\partial f}{\partial t} + \mathbf{c} \frac{\partial f}{\partial \mathbf{x}} + \mathbf{G} \frac{\partial f}{\partial \mathbf{c}} = \mathbf{S}. \quad (2.1)$$

Here, f is the molecular velocity distribution function, which represents the density of molecules in phase-space (i.e. the space spanned by position and velocity). Here \mathbf{c} is molecular velocity, \mathbf{G} is external force and \mathbf{S} is a complex integral term, which represents the change of distribution function due to the interaction between molecules. Owing to the presence of \mathbf{S} , the Boltzmann equation is difficult to solve numerically.

The DSMC is a stochastic molecule-based method that approximates the molecular velocity distribution function in the Boltzmann equation with simulation molecules (Oran, Oh & Cybyk 1998). It has been demonstrated that DSMC converges to the solution of the Boltzmann equation in the limit of a large number of simulation molecules (Wagner 1992). In DSMC, each simulation molecule represents F real molecules and F is the so-called simulation ratio. The DSMC tracks the simulation molecules as they move, collide with other molecules and reflect from boundaries. The macroscopic physical quantities are obtained via sampling corresponding molecular information and making an average at the sampling cells. Specifically, density (ρ) and macroscopic velocity (v_i) are computed with

$$\rho = \frac{mFN_p}{V_{cell}}, \quad \text{and} \quad v_i = \frac{1}{N_p} \sum_{cell} c_i, \quad (2.2a,b)$$

where m is molecular mass, N_p is the number of simulation molecules in the sampling cell, and V_{cell} is the volume of the sampling cell. The velocity of each molecule can be regarded as a sum of two parts, i.e. the macroscopic velocity of the cell to which it belongs and the molecular thermal velocity (which is defined as $C_i = c_i - v_i$). Based on the molecular thermal velocity, temperature (T) is computed with

$$T = \frac{1}{3k_B N_p} \sum_{cell} mC^2, \quad (2.3)$$

where k_B is the Boltzmann constant, and $C^2 = C_1^2 + C_2^2 + C_3^2$. Pressure tensor (p_{ij}) is computed with

$$p_{ij} = \frac{mF}{V_{cell}} \left(\sum_{cell} c_i c_j - N_p v_i v_j \right), \quad (2.4)$$

and is generally split into its trace and trace-free parts, i.e. pressure (p) and stress tensor (τ_{ij}), $p_{ij} = p\delta_{ij} + \tau_{ij}$. Here, δ_{ij} is Kronecker delta (Heinbockel 2001). Therefore, pressure (p) and stress tensor (τ_{ij}) are computed with

$$p = \frac{1}{3}(p_{11} + p_{22} + p_{33}) \quad \text{and} \quad \tau_{ij} = p_{ij} - p\delta_{ij}. \quad (2.5a,b)$$

The heat flux (q_i) is computed with

$$q_i = \frac{F}{2V_{cell}} \sum_{cell} mC^2 C_i. \quad (2.6)$$

The local viscosity coefficient is computed with

$$\mu = \mu_0 \left(\frac{T}{T_0} \right)^\omega, \quad (2.7)$$

where $\mu_0 = 2.117 \times 10^{-5} \text{ kg m}^{-1} \text{ s}^{-1}$ is the reference viscosity at the reference temperature $T_0 = 273 \text{ K}$. Here ω is the viscosity exponent and is determined by the molecular model employed. For the hard sphere (HS) model (Bird 1994), the Maxwell molecule model (Bird 1994) and the variable hard sphere (VHS) model (Bird 1981), ω is equal to 0.5, 1.0 and 0.81, respectively. The local heat conductivity coefficient is computed according to the local viscosity coefficient using

$$\kappa = \frac{5}{2} \frac{\mu}{Pr}, \quad (2.8)$$

where Pr is the Prandtl number and is equal to $2/3$ for a monatomic gas.

For a specific application, DSMC first initializes the simulation molecules according to the initial distributions of density, temperature and macroscopic velocity. Then, the molecular motions and intermolecular collisions are sequentially conducted in each computational time step. The molecular motions are implemented in a deterministic way. Each 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 trajectory crosses any boundaries, an appropriate gas–wall interaction model would be employed to determine the reflected velocity. Common gas–wall interaction models include specular, diffuse and Maxwell reflection models. In our simulations, the diffuse model is employed. The intermolecular collisions are implemented in a probabilistic way. Among the several algorithms for modelling intermolecular collisions, the no-time-counter technique (Bird 1994) is the most widely used. In the no-time-counter, any two molecules in the same computational cell are selected as a collision pair with a probability that is proportional to the relative speed between them. Then, the post-collision velocities of molecules are determined depending on the molecular model employed. In this work we employ the HS model for the first two benchmark cases (diffusion flow and Taylor–Green vortex), the Maxwell molecule model for the shock wave case and the VHS model for the Poiseuille flow case. It is noteworthy that because the molecular motions and intermolecular collisions are conducted in a decoupled manner, the time step in DSMC needs to be smaller than the molecular mean collision time, and the cell size for the selection of collision pairs needs to be smaller than the molecular mean free path.

The DSMC has been successfully applied to simulate gas flows in the whole flow regimes, and the results have been well validated (Stefanov, Roussinov & Cercignani 2002; Sun & Boyd 2002; Zhang, Fan & Fei 2010; Gallis *et al.* 2017). The advantage of the DSMC method is that 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 sampling cells. In this work, the aim of using data generated by DSMC is to prove the ability of DHC-GEP to discover the governing equations that could be unknown.

2.2. Dimensional homogeneity constrained gene expression programming

2.2.1. General framework

The DHC-GEP is an enhanced GEP method, which compiles mathematical expressions into chromosomes and iteratively discovers the equation that is suitable for describing the training data by mimicking the natural law of evolution. The flowchart of DHC-GEP is shown in figure 2(a). It starts with creating N_i random individuals of initial population. Each individual has two forms: genotype (chromosome (CS)) and phenotype (expression tree (ET)). Phenotype is the expression product of genotype. Each CS is composed of one or more genes. A specific schematic diagram of a gene is shown in figure 2(b). There are two parts in a gene, i.e. head and tail. The head consists of the symbols from the function set or terminal set, while the tail consists of only symbols from the terminal set. The function set and terminal set are both predefined according to the specific problem. For the problems in this work, the function set includes basic mathematical operators (+, −, ×, ÷). If necessary for other problems, the function set can also include nonlinear functions such as sin, cos and even user-defined functions. The terminal set includes the symbols of variables and physical constants. Taking the first symbol of the gene as the root node, we can translate the gene into the ET (shown in figure 2b) through level-order traversal according to the argument requirement of each function. Note that the final

DHC-GEP for discovering governing equations

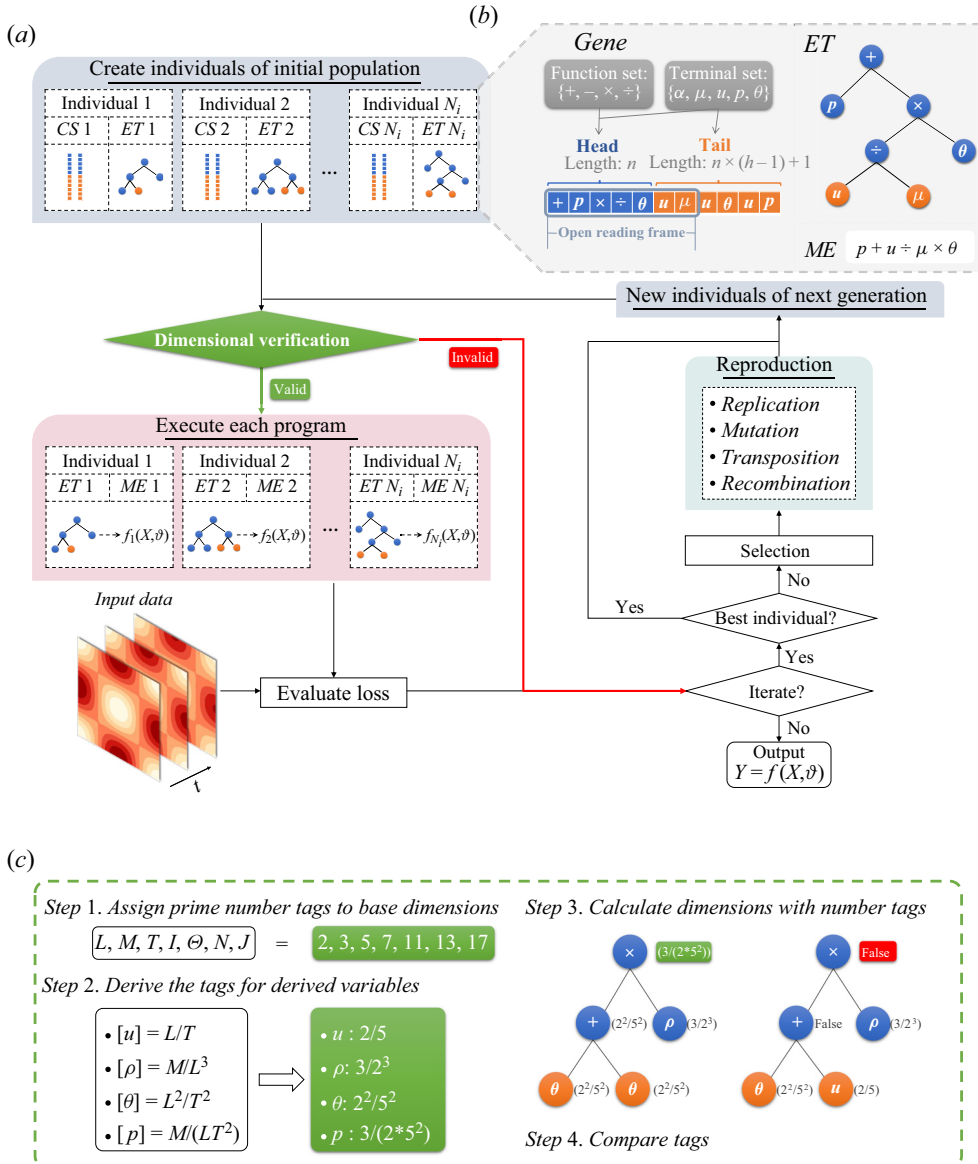


Figure 2. Main characteristics of DHC-GEP. (a) Flowchart of DHC-GEP. (b) Schematic diagram of a gene and its corresponding expression tree and mathematical expression. (c) Strategy of dimensional verification.

four symbols are not expressed, and the region before them is called ORF. The length of ORF is variable, which is determined by the argument requirement of the function nodes in the head. On the contrary, the length of gene is fixed, preventing the individuals from bloating. Assuming the length of the head is n , then the length of the tail must be $n \times (h - 1) + 1$, where h is the maximum number of arguments in the functions. Four genes with different ORFs are shown in figure 3. Although the four genes have the same length, the complexities of their expression products are different. Therefore, ORF is the decisive factor that ensures the diversity of expression products and the strong expressivity of DHC-GEP. This is also the reason that the length of the head determines only the upper

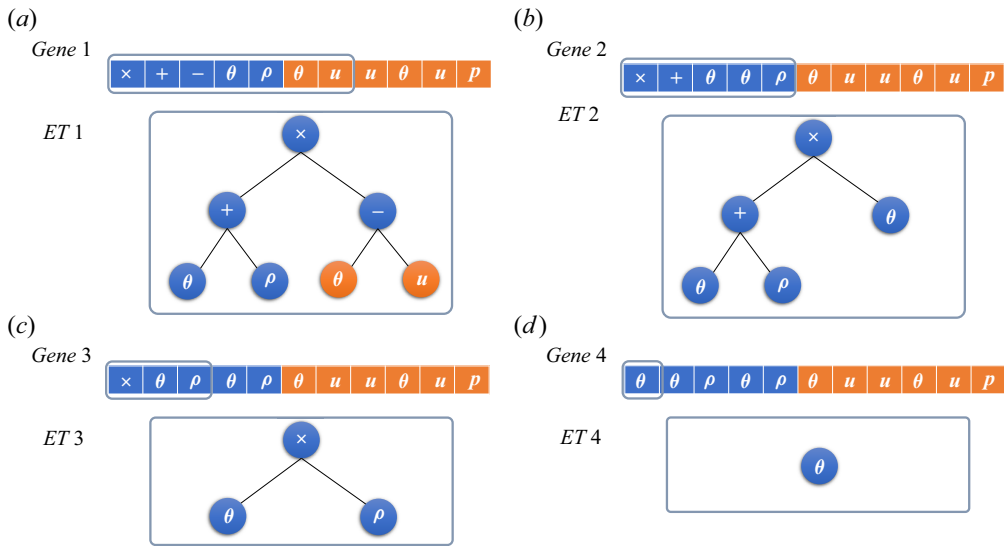


Figure 3. Four genes with different ORFs.

limit of complexity, not the lower limit. When discovering unknown governing equations, a relatively long head would be a good choice. Because with ORF, a long gene can represent either a simple equation or a complex equation.

Then, via dimensional verification (which will be introduced in §2.2.2), all the individuals are classified into valid individuals and invalid individuals according to whether they satisfy the dimensional homogeneity. The valid individuals would be translated into mathematical expressions (MEs) and evaluated for losses with input data according to the loss function, which is defined using the mean relative error (MRE) as

$$L_{MRE} = \frac{1}{N} \sum_{i=1}^N \left| \frac{\hat{Y}_i - Y_i}{Y_i} \right|. \tag{2.9}$$

Here, the variable with a superscript \wedge is the predicted variable, and N is the total number of data points. The invalid individuals would be directly assigned a significant loss and eliminated in the subsequent iterations. The best individual (with the lowest loss) is replicated to the next generation directly. Other individuals are selected as superior individuals with probabilities that are inversely proportional to their losses. Based on the superior individuals, the population of the next generation is reproduced via genetic operators. Common genetic operators include replication, mutation, transposition and recombination. The schematic diagrams of them are shown in figure 4; the details are introduced as follows.

- (i) Replication is responsible for copying the selected superior individuals to the next generation. After replication, the population maintains the same size as the initial population.
- (ii) Mutation is a change in some symbols of the genes. To maintain the structural organization of the genes, the symbols in the head can be changed into either functions or terminals, while the symbols in the tail can only be changed into terminals. In addition, note that mutations in genes do not necessarily lead to changes in expression products, unless the mutation occurs in ORF.

DHC-GEP for discovering governing equations

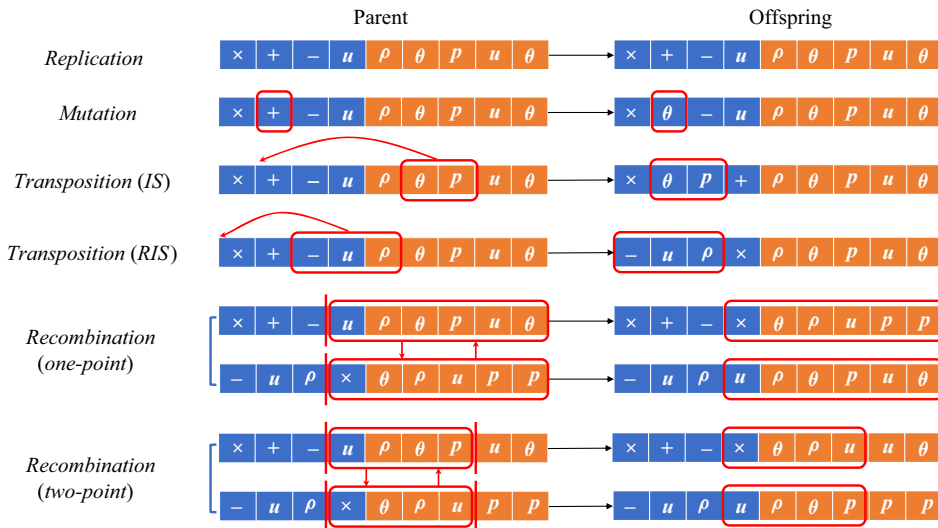


Figure 4. Schematic diagrams of genetic operators.

- (iii) Transposition is the replacement of a gene segment with another position. Specifically, it includes insertion sequence (IS) transposition, root insertion sequence (RIS) transposition and gene transposition. For IS transposition, a short segment with a function or terminal in the first position is transposed to the head of the genes. The length of the transposed segment and the insertion site are randomly selected. The sequence before the insertion site remains unchanged. The sequence after the insertion site is shifted backwards as a whole, and the last symbols (as many as the length of the transposed segment) of the head are deleted. The RIS transposition is similar to IS transposition, except that the first position of the transposed segment must be a function, and the insertion site must be the root. Gene transposition occurs in the CSs that consist of multiple genes. During gene transposition, one gene switches position with another gene.
- (iv) Recombination is an exchange of gene segments between two selected parent CSs, including one-point recombination, two-point recombination and gene recombination. For one-point recombination, one bond is randomly selected as the crossover point. Then, both selected CSs are cut at this point, and exchange the segments after the crossover point. For two-point recombination, two crossover points are selected, and the exchanged segments are between these two points. For gene recombination, the exchanged segment is an entire gene.

Note that the genetic operators, except replication, are not necessarily invoked in every CS, but are invoked with certain probabilities. In this work, the probabilities of the genetic operators being invoked refer to Ferreira (2006), listed in table 1, which were concluded from various examples via the trial-and-error approach.

The above processes are iteratively conducted until a satisfactory individual is obtained. Specifically, in benchmark studies, we manually terminate the evolution when the target equation is discovered. In cases of discovering the unknown constitutive relations, the evolution is terminated automatically if the optimal equation remains unchanged for 1000 consecutive generations.

Genetic operator	Probabilities
Mutation	0.05
Inversion	0.1
IS transposition	0.1
RIS transposition	0.1
Gene transposition	0.1
One-point recombination	0.3
Two-point recombination	0.2
Gene recombination	0.1

Table 1. Probabilities for genetic operators being invoked.

Other advanced knowledge is provided in the appendices, including the Dc domain (Appendix B) for generating numerical constants and the linear scaling (Appendix C) for discovering numerical coefficients.

2.2.2. Dimensional verification

The dimensional verification is the additional process to implement the constraint of dimensional homogeneity, and is introduced as follows (a simplified version is shown in figure 2c). The general form of a governing equation is assumed to be

$$Y = f(X, \vartheta). \tag{2.10}$$

Here, Y is the target variable, and $f(\cdot)$ is the mathematical expression coded by a CS. Here X and ϑ represent variables and physical constants, respectively. Dimensional homogeneity means that $f(\cdot)$ should have the same dimension as Y , and every part in $f(\cdot)$ should satisfy the constraint that the parameters for operators $+$ or $-$ must have the same dimensions.

Generally, dimensional verification includes calculating the dimension of $f(\cdot)$ and comparing it with the dimension of Y . For calculating dimensions, there are five principles that should be noted.

- (i) Computers cannot directly deal with symbolic operations, but only numerical operations.
- (ii) In SI units, there are seven base dimensions: length (L); mass (M); time (T); electric current (I); thermodynamic temperature (Θ); amount of substance (N); luminous intensity (J). The dimensions of any other physical quantities can be derived by powers, products or quotients of these base dimensions.
- (iii) Base dimensions are independent of each other. Any dimension cannot be derived by other base dimensions.
- (iv) Physical quantities with different dimensions cannot be added or subtracted. Adding or subtracting the physical quantities with the same dimension does not change the dimension.
- (v) When physical quantities are multiplied or divided, the corresponding dimensions are multiplied or divided equally.

Considering the above principles, we propose to assign number tags to physical quantities and replace the dimensional calculations with numerical calculations. For the seven base dimensions, the tags are 2, 3, 5, 7, 11, 13 and 17, respectively. These numbers are prime numbers, ensuring that the base dimensions are independent of each other. The tags

Algorithm 1 Modified addition function

```

1  Input: tags for the input variables, denoted by a and b
2  Modified_addition(a,b):
3    If a or b is bool value:  (# Indicates that this equation has been proved to be invalid at previous nodes.)
4      Return False  (# Continue transfer the message that this equation is invalid.)
5    If a is integer:  (# Indicates that the input variables is a numerical constant.)
6      Assign 1/1 (i.e. 1 in the form of fraction) to a  (# Assign the number tag 1 to the numerical constant.)
7    If b is integer:  (# Indicates that the input variables is a numerical constant.)
8      Assign 1/1 (i.e. 1 in the form of fraction) to b  (# Assign the number tag 1 to the numerical constant.)
9    If a is equal to b:  (# Indicates that the input variables have the same dimensions.)
10     Return a  (# Return the tag for any input variables as the tag for this node.)
11   Else:  (# Indicates that the input variables do not have the same dimensions.)
12     Return False  (# Judge that this equation is invalid.)

```

Algorithm 2 Modified multiplication function

```

1  Input: tags for the input variables, denoted by a and b
2  Modified_multiplication(a,b):
3    If a or b is bool value:  (# Indicates that this equation has been proved to be invalid at previous nodes.)
4      Return False  (# Continue transfer the message that this equation is invalid.)
5    If a is integer:  (# Indicates that the input variables is a numerical constant.)
6      Assign 1/1 (i.e. 1 in the form of fraction) to a  (# Assign the number tag 1 to the numerical constant.)
7    If b is integer:  (# Indicates that the input variables is a numerical constant.)
8      Assign 1/1 (i.e. 1 in the form of fraction) to b  (# Assign the number tag 1 to the numerical constant.)
9    Return a×b in the form of fraction  (# Return the product of the tags for the input variables as the tag for this node.)

```

for other physical quantities are derived according to their dimensions. For example, as the dimension of velocity is length divided by time (L/T), its tag is defined as $2/5$. It is noteworthy that the tags are always in the form of fractions rather than floats to avoid truncation errors. The tags for numerical constants and dimensionless quantities are set to 1. In addition, according to the last two principles, we modify the operation rules of mathematical operators. The pseudocodes for addition (+) and multiplication (\times) are provided in Algorithm 1 and Algorithm 2, respectively. Subtraction ($-$) and division (\div) are defined in similar ways. With the number tags and modified mathematical operators, we can compute the dimension of each node in the ET from the bottom up. If the parameters for one function node (+ or $-$) do not have the same dimension, the individual is directly identified as invalid one. Otherwise, we can calculate the specific dimension of the individual, i.e. the number tag for the root node. Comparing it with the tag for the target variable, we can conclude whether the individual is dimensionally homogeneous.

Note that the datasets and source codes used in this work are available on GitHub at <https://github.com/Wenjun-Ma/DHC-GEP>.

3. Demonstration on benchmark studies

3.1. Diffusion equation

We employ the DSMC method to simulate a two-dimensional diffusion flow in a square domain, as shown in figure 5(a). The side length of the square is $L = 100\lambda$, and λ is the mean free path of argon gas molecules under standard conditions (pressure $p = 1.01 \times 10^5$ Pa and temperature $T = 273$ K), i.e. $\lambda = 6.25 \times 10^{-8}$ m. As a result, the Knudsen number ($Kn = \lambda/L$) is 0.01. All the four sides are set to periodic boundaries. The simulation includes only argon gas, which is divided into two species, denoted as species-A and species-B, respectively. At the initial instant, the density distributions of these two species are $\rho_A = 0.5\rho_0(1 - \cos(2\pi y/L))$ and $\rho_B = 0.5\rho_0(1 + \cos(2\pi y/L))$, respectively. Here, $\rho_0 = 1.78 \text{ kg m}^{-3}$ is the density of argon gas under standard conditions, and y is the spatial coordinate in the vertical direction. The initial macroscopic velocity is

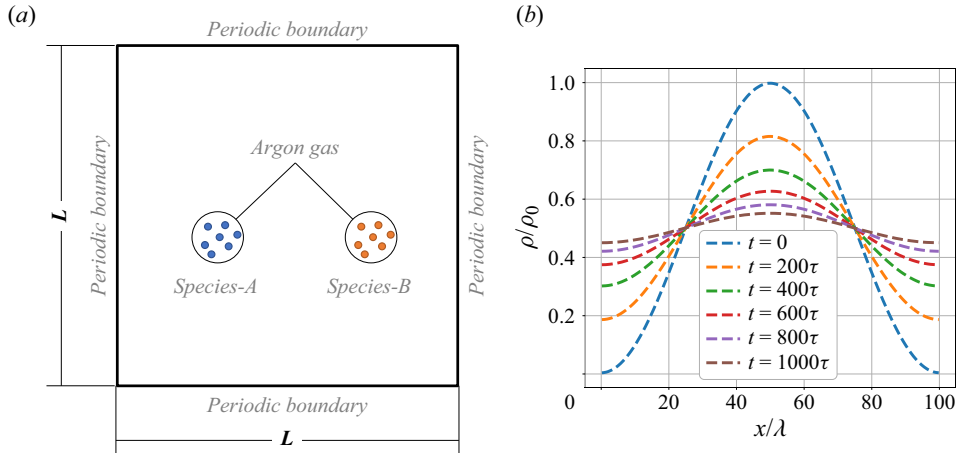


Figure 5. (a) Schematic diagram of the computational domain for the diffusion flow. (b) Distributions of density at the instants of $t = 0, 200\tau, 400\tau, 600\tau, 800\tau, 1000\tau$.

uniformly zero. During simulation, the macroscopic velocity and the total density of the two species remain uniform and unchanged, while the respective densities of these two species vary with space and time.

The computational domain is divided into 256×256 cells, which means that the cell size $(\Delta x, \Delta y)$ is approximately 0.4λ . In each cell, 4000 simulation molecules are randomly distributed at the initial instant, with one simulation molecule representing 4×10^6 real molecules. The initial thermal velocity for each simulation molecule is sampled randomly from a Maxwell distribution at 273 K. The computational time step (Δt) is 0.1τ , where τ is the mean collision time of argon gas molecules under standard conditions with the definition of $\tau = \lambda/\bar{c}$. Here, \bar{c} is the molecular mean velocity, i.e. $\bar{c} = \sqrt{8k_B T/\pi m}$, where k_B and m are the Boltzmann constant and molecular mass, respectively.

Note that the density varies only along the vertical direction. We average the data along the horizontal direction to reduce the statistical errors. During simulation, we sample the macroscopic densities of species-A at the instants of $t = 0, 10\tau, 20\tau, \dots, 1000\tau$. The temporal evolution of density distributions is shown in figure 5(b). The final dataset is composed of 25 856 (256×101) data points that are distributed in 256 cells along the vertical direction at 101 instants.

The theoretical governing equation of this flow is the diffusion equation, i.e.

$$\frac{\partial \rho}{\partial t} = D \frac{\partial^2 \rho}{\partial y^2}. \tag{3.1}$$

Here, D is the diffusion coefficient. According to the Chapman–Enskog theory (Chapman & Cowling 1990), the diffusion coefficient for an HS gas at the first-order approximation is

$$D = \frac{3\pi}{16} \frac{\lambda^2}{\tau}. \tag{3.2}$$

Therefore, for the argon gas in our simulation, D equals $1.40 \times 10^{-5} \text{ m}^2 \text{ s}^{-1}$.

Before using the dataset generated by DSMC, we test DHC-GEP and Original-GEP on a clean dataset. The only difference between the DSMC dataset and the clean dataset lies in the target variable $(\partial \rho / \partial t)$. In the clean dataset, $\partial \rho / \partial t$ is directly computed based

on D and $\partial^2 \rho / \partial y^2$, according to (3.1). We define the function set as $\{+, -, \times, \div\}$, and the terminal set as $\{\mu, D, \partial \rho / \partial y, \partial^2 \rho / \partial y^2, \partial^3 \rho / \partial y^3\}$. Here, μ is the viscosity coefficient, and is introduced as a distraction term. According to the Chapman–Enskog theory (Chapman & Cowling 1990), μ equals $(5\pi/32)(\lambda^2/\tau)\rho = 2.08 \times 10^{-5} \text{ kg m}^{-1} \text{ s}^{-1}$. All the hyperparameters are set to be the same for both GEP methods across all cases (see Appendix E). The derived equations are shown in table 2. It can be observed that the derived equation of DHC-GEP is consistent with the theoretical equation, while the derived equation of Original-GEP is partially correct. Specifically, Original-GEP identifies the correct function form and coefficient (1.40×10^{-5}). In the view of data fitting, this is a correct result. However, Original-GEP fails to recognize that the coefficient (1.40×10^{-5}) preceding the diffusion term represents the diffusion coefficient. Consequently, while the resulting equation may be numerically accurate, it lacks complete physical significance. For another diffusion flow with a different diffusion coefficient, the derived equation of Original-GEP is no longer valid, while the derived equation of DHC-GEP is generally applicable.

Subsequently, we test DHC-GEP and Original-GEP on the DSMC dataset. All settings are consistent with those in the test on the clean dataset. The derived equations are shown in table 2. The coefficient of the equation discovered by DHC-GEP has a minor deviation from that of the theoretical equation, and the loss is not zero. This is because DSMC is a stochastic molecule-based method, the data of which are inherently noisy. It is impossible for DSMC to simulate a flow with the diffusion coefficient being exactly $1.40 \times 10^{-5} \text{ m}^2 \text{ s}^{-1}$. Small fluctuations around the theoretical value are acceptable. In addition, calculating derivatives also introduces errors. Therefore, we believe that DHC-GEP has discovered the correct equation, while the derived equation of Original-GEP is clearly wrong for not satisfying the fundamental dimensional homogeneity. Besides, we generate another dataset of diffusion flow with a diffusion coefficient being $3.0 \times 10^{-5} \text{ m}^2 \text{ s}^{-1}$ and calculate the losses of the two derived equations based on the new dataset. The losses for the derived equations of Original-GEP and DHC-GEP are 0.55 and 0.06, respectively. Therefore, the equation obtained by Original-GEP is clearly overfitting, while the equation obtained by DHC-GEP is generally applicable in different diffusion flows.

To test the sensitivity of DHC-GEP to hyperparameters, we conduct a parametric study on the length of the head, the number of genes in a CS and the number of individuals in a population. Specifically, the parameter spaces for these three hyperparameters are set to $\{10, 11, 12, 13, 14, 15\}$, $\{1, 2\}$ and $\{800, 1000, 1200, 1400, 1600, 1660\}$, respectively, resulting in a total of 72 ($6 \times 2 \times 6$) distinct parameter combinations. Other hyperparameters are kept unchanged, including the probabilities of invoking the genetic operators and the maximum number of evolution generations. In all 72 sets of parameter combinations, DHC-GEP consistently discovers the correct equation. In contrast, Original-GEP discovers 71 completely different results, all of which are overfitting and fail to satisfy the dimensional homogeneity constraint despite their lower losses than the correct equations.

Considering that DSMC is a simulation method at mesoscale, which coarse grains the molecular description to the hydrodynamic regime (Hadjiconstantinou 2000), the influence of the coarse graining on the derived equations of DHC-GEP is discussed. For the same computational domain, we employ seven sets of sampling cells with different resolutions (256×256 , 128×128 , 64×64 , 32×32 , 16×16 , 8×8 , 4×4) to acquire macroscopic physical quantities. The derived equations based on these datasets are summarized in table 3. It is encouraging to note that DHC-GEP is still capable of discovering the correct equation when the resolution is 16×16 and the size of the

Case	Theoretical equation	Data	Method	Derived equation	Loss
Diffusion problem	$\frac{\partial \rho}{\partial t} = D \frac{\partial^2 \rho}{\partial y^2}$,	Clean dataset	DHC-GEP	$\frac{\partial \rho}{\partial t} = 1.0D \frac{\partial^2 \rho}{\partial y^2}$	0
	$D = 1.40 \times 10^{-5} \text{ m}^2 \text{ s}^{-1}$	DSMC dataset	Original-GEP	$\frac{\partial \rho}{\partial t} = 1.40 \times 10^{-5} \frac{\partial^2 \rho}{\partial y^2}$	0
			DHC-GEP	$\frac{\partial \rho}{\partial t} = 0.984D \frac{\partial^2 \rho}{\partial y^2}$	0.027
			Original-GEP	$\frac{\partial \rho}{\partial t} = 1.91 \times 10^{-7} \left(72 \frac{\partial^2 \rho}{\partial y^2} + 72 \frac{\partial^2 \rho}{\partial y^2} \frac{\partial \rho}{\partial y} + \frac{1}{D^2(D - \rho + 1)} \right)$	0.024
Taylor–Green vortex	$\frac{\partial \omega_z}{\partial t} = \nu \left(\frac{\partial^2 \omega_z}{\partial x^2} + \frac{\partial^2 \omega_z}{\partial y^2} \right) - \left(u \frac{\partial \omega_z}{\partial x} + v \frac{\partial \omega_z}{\partial y} \right)$,	Clean dataset	DHC-GEP	$\frac{\partial \omega_z}{\partial t} = 1.0\nu \left(\frac{\partial^2 \omega_z}{\partial x^2} + \frac{\partial^2 \omega_z}{\partial y^2} \right)$	0
	$\nu = 1.17 \times 10^{-5} \text{ m}^2 \text{ s}^{-1}$		Original-GEP	$\frac{\partial \omega_z}{\partial t} = 1.17 \times 10^{-5} \left(\frac{\partial^2 \omega_z}{\partial x^2} + \frac{\partial^2 \omega_z}{\partial y^2} \right)$	0
		DSMC dataset	DHC-GEP	$\frac{\partial \omega_z}{\partial t} = 0.981\nu \left(\frac{\partial^2 \omega_z}{\partial x^2} + \frac{\partial^2 \omega_z}{\partial y^2} \right)$	0.112
			Original-GEP	$\frac{\partial \omega_z}{\partial t} = -60.53 \left(\frac{\partial^2 u}{\partial y^2} \left(u + \frac{\partial v}{\partial x} + \frac{\partial \omega_z}{\partial y} + 3 \frac{\partial v}{\partial y} \frac{\partial^2 u}{\partial y^2} \frac{\partial \omega_z}{\partial y} \right) + \frac{\partial v}{\partial x} \right)$	0.096
Hybrid problem	$\frac{\partial \omega_z}{\partial t} = \nu \left(\frac{\partial^2 \omega_z}{\partial x^2} + \frac{\partial^2 \omega_z}{\partial y^2} \right) - \left(u \frac{\partial \omega_z}{\partial x} + v \frac{\partial \omega_z}{\partial y} \right)$,	Hybrid dataset	DHC-GEP	$\frac{\partial \omega_z}{\partial t} = 0.988\nu \left(\frac{\partial^2 \omega_z}{\partial x^2} + \frac{\partial^2 \omega_z}{\partial y^2} \right) - 0.988 \left(u \frac{\partial \omega_z}{\partial x} + v \frac{\partial \omega_z}{\partial y} \right)$	0.082

Table 2. Derived equations of DHC-GEP.

Resolution of sampling cells	Size of sampling cells	Derived equation	Loss
256 × 256	0.4λ	$\frac{\partial \rho}{\partial t} = 0.984D \frac{\partial^2 \rho}{\partial y^2}$	0.027
128 × 128	0.8λ	$\frac{\partial \rho}{\partial t} = 0.986D \frac{\partial^2 \rho}{\partial y^2}$	0.024
64 × 64	1.6λ	$\frac{\partial \rho}{\partial t} = 0.994D \frac{\partial^2 \rho}{\partial y^2}$	0.027
32 × 32	3.1λ	$\frac{\partial \rho}{\partial t} = 0.990D \frac{\partial^2 \rho}{\partial y^2}$	0.045
16 × 16	6.3λ	$\frac{\partial \rho}{\partial t} = 1.001D \frac{\partial^2 \rho}{\partial y^2}$	0.038
8 × 8	12.5λ	$\frac{\partial \rho}{\partial t} = 0.791 \left(D + \frac{\mu}{\rho} \right) \frac{\partial^2 \rho}{\partial y^2}$	0.445
4 × 4	25λ	$\frac{\partial \rho}{\partial t} = -0.075 \left(D \frac{\partial \rho}{\partial y} \frac{\partial^2 \rho}{\partial y^2} + (\mu + D\rho) \frac{\partial^3 \rho}{\partial y^3} \right) / \frac{\partial \rho}{\partial y}$	0.516
Theoretical equation		$\frac{\partial \rho}{\partial t} = D \frac{\partial^2 \rho}{\partial y^2}$	

Table 3. Derived equations of DHC-GEP based on the datasets with different resolutions.

sampling cells is up to 6.3λ. From the results, it can be concluded that DHC-GEP is robust to coarse graining, as long as the sampled data can accurately capture the gradients of macroscopic quantities.

In DHC-GEP, the overfitting results are automatically filtered out due to not satisfying the dimensional homogeneity. On the contrary, Original-GEP always favours the equations with smaller loss, so it tends to converge to overfitting results if data are noisy. Our numerical experiences show that Original-GEP may discover the correct equations only when tuning the hyperparameters repeatedly and terminating the evolution at a proper generation (when overfitting equations have not appeared).

We compare the computational cost needed per 1000 generations of evolution. Based on the 3.5 GHz Intel Xeon E5-1620 processor, the average CPU runtime of Original-GEP (443 s) is almost twice that of DHC-GEP (216 s). The main reason is that DHC-GEP can identify some individuals as invalid through dimensional verification and then skip the process of evaluating losses for these individuals. It can still save computational time despite the extra expense of dimensional verification. Additionally, as the complexity of the problem increases, the number of invalid individuals also increases, so the advantage of the computational efficiency of DHC-GEP becomes more significant.

Furthermore, we conduct a much more challenging test in which we do not include the transport coefficients in the terminal set. Specifically, the terminal set is $\{\tau, \lambda, \partial \rho / \partial y, \partial^2 \rho / \partial y^2, \partial^3 \rho / \partial y^3\}$, which involves only the fundamental physical property parameters of gas (molecular mean collision time (τ) and molecular mean free path (λ)), but excludes the diffusion coefficient (D). Other hyperparameters are the same as those in

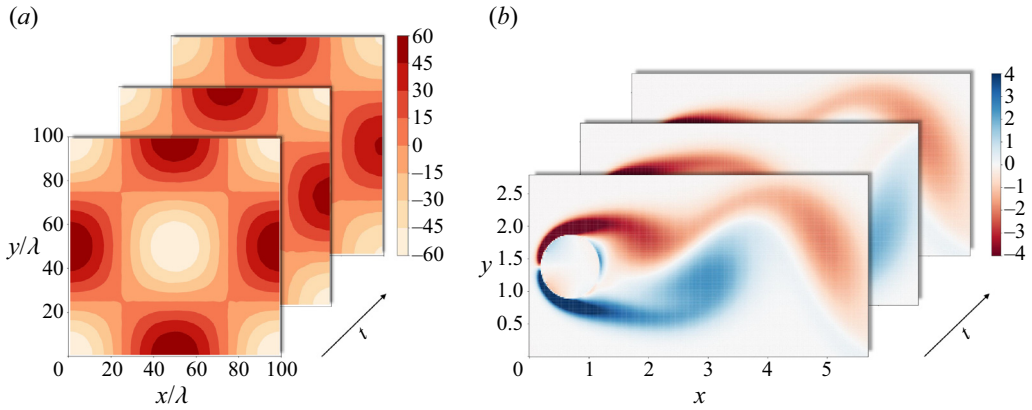


Figure 6. Temporal evolution of vorticity. (a) Taylor–Green vortex at the instants of $t = 0, 100\tau, 200\tau$. Here, τ and λ are the mean collision time and mean free path of argon gas molecules at standard condition, respectively. (b) Viscous flow around a cylinder at the instants of $t = 0 \text{ s}, 3 \text{ s}, 6 \text{ s}$.

the original test. The derived equation is

$$\frac{\partial \rho}{\partial t} = 0.579 \frac{\lambda^2}{\tau} \frac{\partial^2 \rho}{\partial y^2}. \quad (3.3)$$

According to (3.2), the coefficient $(0.579(\lambda^2/\tau))$ is essentially the diffusion coefficient. This result demonstrates that DHC-GEP has the capability to derive accurate equations with minor prior knowledge of the essential transport coefficients, and can deduce the essential transport coefficients from fundamental physical property parameters using dimensional homogeneity as a constraint.

3.2. Vorticity transport equation

We employ DSMC to simulate the temporal evolution of a Taylor–Green vortex, as shown in figure 6(a). The simulation model is approximately the same as that of diffusion flow, including simulation conditions, geometry and boundary conditions. The major difference is that the initial distribution of macroscopic velocity is a Taylor–Green vortex,

$$\begin{cases} u = v_0 \cos(x) \sin(y), \\ v = -v_0 \sin(x) \cos(y), \end{cases} \quad (3.4)$$

where u and v are the velocities in the horizontal (x) and vertical (y) directions, respectively, and v_0 is the initial amplitude of velocity and equals 30 ms^{-1} in our simulation.

The computational cell and time step are set the same as those in the diffusion flow. Additionally, we introduce sampling cells. In this case, each sampling cell consists of 16 computational cells, i.e. the whole computational domain has 64×64 sampling cells. The macroscopic physical quantities obtained with sampling cells are cleaner than those obtained with computational cells, due to more molecules being sampled in each cell. In addition, to further reduce the statistical errors, 10 independent simulations are conducted with different random number sequences to make an ensemble average. During simulation, we sample the macroscopic velocities (u, v) and vorticities (ω_z) at the instants of $t = 0, 10\tau, 20\tau, \dots, 310\tau$. The final dataset is composed of 131 072 ($64 \times 64 \times 32$) data points distributed in 64×64 sampling cells at 32 instants.

The theoretical governing equation is the vorticity transport equation,

$$\frac{\partial \omega_z}{\partial t} = \nu \left(\frac{\partial^2 \omega_z}{\partial x^2} + \frac{\partial^2 \omega_z}{\partial y^2} \right) - \left(u \frac{\partial \omega_z}{\partial x} + v \frac{\partial \omega_z}{\partial y} \right). \quad (3.5)$$

Here, ω_z is the vorticity in the z direction. ν is the kinematic viscosity and approximately equals $1.17 \times 10^{-5} \text{ m}^2 \text{ s}^{-1}$. Compared with the diffusion equation (3.1), this equation is more complex, involving multiple variables and nonlinear terms.

In this case, we define the target variable as $\partial \omega_z / \partial t$, the function set as $\{+, -, \times, \div\}$, and the terminal set as $\{v, u, v, \omega_z, \partial u / \partial x, \partial^2 u / \partial x^2, \partial u / \partial y, \partial^2 u / \partial y^2, \partial v / \partial x, \partial^2 v / \partial x^2, \partial v / \partial y, \partial^2 v / \partial y^2, \partial \omega_z / \partial x, \partial^2 \omega_z / \partial x^2, \partial \omega_z / \partial y, \partial^2 \omega_z / \partial y^2\}$.

We also test the performances of DHC-GEP on the clean dataset and the DSMC dataset sequentially. The results are provided in table 2. An interesting observation is that the derived equations of DHC-GEP miss two convective terms ($u(\partial \omega_z / \partial x)$, $v(\partial \omega_z / \partial y)$). This is caused by the unique feature of the Taylor–Green vortex. The analytical solution of the Taylor–Green vortex is

$$\begin{cases} u = v_0 \cos(x) \sin(y) \exp(-2\nu t), \\ v = -v_0 \sin(x) \cos(y) \exp(-2\nu t), \\ \omega_z = -2v_0 \cos(x) \cos(y) \exp(-2\nu t). \end{cases} \quad (3.6)$$

Substituting (3.6) into the convective terms ($u(\partial \omega_z / \partial x)$, $v(\partial \omega_z / \partial y)$), it is clear that the sum of the convective terms is automatically zero. Therefore, the equations without convective terms are also correct for the Taylor–Green vortex. It can be considered as a specialized variant of the complete vorticity transport equation in the Taylor–Green vortex.

To discover the complete vorticity transport equation and validate the performance of DHC-GEP on discovering nonlinear terms, we further consider a viscous flow around a cylinder at Reynolds number being 100, as shown in figure 6(b). The dataset is the open access dataset provided by Rudy *et al.* (2017), and the theoretical governing equation is

$$\frac{\partial \omega_z}{\partial t} = 0.01 \cdot \left(\frac{\partial^2 \omega_z}{\partial x^2} + \frac{\partial^2 \omega_z}{\partial y^2} \right) - \left(u \frac{\partial \omega_z}{\partial x} + v \frac{\partial \omega_z}{\partial y} \right). \quad (3.7)$$

For the sake of simplicity, we regard the data as dimensional data with kinematic viscosity $\nu = 0.01 \text{ m}^2 \text{ s}^{-1}$. We randomly sampled 200 spatiotemporal data points from the Taylor–Green vortex dataset and another 200 spatiotemporal data points from the flow around cylinder dataset, forming a hybrid dataset consisting of 400 spatiotemporal data points. The hyperparameters of DHC-GEP are consistent with those in the case of the Taylor–Green vortex. The results are provided in table 2. The DHC-GEP discovers the correct and complete vorticity transport equation. This demonstrates that in some special cases, where the overfitting equations may satisfy the dimensional homogeneity and fit well with the training data, employing multiple datasets can help mitigate overfitting. Overfitting equations are generally dataset specific. While an overfitting equation may adequately describe a particular dataset, it cannot be expected to generalize well to multiple datasets. Besides, note that the training dataset consisting of 400 spatiotemporal data points is relatively small. Generally, if the size of dataset is small, the information carried by data is sparse, leading to the derived equations overfitting to specific phenomena. However, in DHC-GEP, the overfitting equations are automatically filtered out by the constraint of dimensional homogeneity.

4. Application on discovering unknown constitutive relations

The general governing equations for fluid flows are the conservation equations of mass, momentum and energy, as follows:

$$\begin{cases} \frac{D\rho}{Dt} + \rho \frac{\partial v_k}{\partial x_k} = 0, \\ \rho \frac{Dv_i}{Dt} + \frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ik}}{\partial x_k} = \rho F_i, \\ \frac{3}{2} \frac{D\theta}{Dt} + \frac{\partial q_k}{\partial x_k} = -(p\delta_{ij} + \tau_{ij}) \frac{\partial v_i}{\partial x_j}. \end{cases} \quad (4.1)$$

Here, $D/Dt = \partial/\partial t + v_i(\partial/\partial x_i)$ is the substantial derivative, and $\theta = k_B T/m$ is the temperature in energy/mass units. Besides the five basic physical variables ρ , v_i and θ , there are eight additional variables, i.e. the viscous stress τ_{ij} and heat flux q_i . To numerically solve (4.1), the additional constitutive relations that close the viscous stress and heat flux are needed. In the continuum regime, the NSF equations are widely employed, which assumes that the viscous stress/heat flux is linearly proportional to the local strain rate/temperature gradient. However, for strong non-equilibrium flows, the NSF equations are not valid (Boyd, Chen & Candler 1995).

Alternatively, high-order constitutive relations have been derived from the Boltzmann equation using the Chapman–Enskog method (Chapman & Cowling 1990), including Burnett equations (Burnett 1936), super-Burnett (Su-Burnett) equations (Shavaliyev 1993) and augmented-Burnett (Au-Burnett) equations (Zhong, MacCormack & Chapman 1993), to account for the non-equilibrium effects. However, despite being proven to be superior to NSF equations, these equations are still unsatisfactory in strong non-equilibrium flows. In this work, we employ DHC-GEP to discover the unknown constitutive relations in two representative non-equilibrium flows, as the examples to illustrate how to apply DHC-GEP to discover unknown governing equations.

To avoid any potential misleading, it is important to emphasize that we do not employ DHC-GEP to discover the complex governing equations like (4.1). In this work, we are focused on discovering the constitutive relations for the two unclosed variables within (4.1), i.e. the viscous stress and heat flux.

4.1. One-dimensional shock wave

In the research community of non-equilibrium flows, the one-dimensional shock wave is a benchmark to validate solvers and formulations in numerical computations. We employ DSMC to simulate the one-dimensional shock wave of argon gas, the general structure of which is shown in figure 7.

The one-dimensional shock essentially connects two equilibrium states. In this work, for upstream (i.e. the free stream), the temperature (T_1) is 300 K, the density (ρ_1) is $1.067 \times 10^{-4} \text{ kg m}^{-3}$ and the mean free path (λ_1) is $1.114 \times 10^{-3} \text{ m}$. For downstream, the state

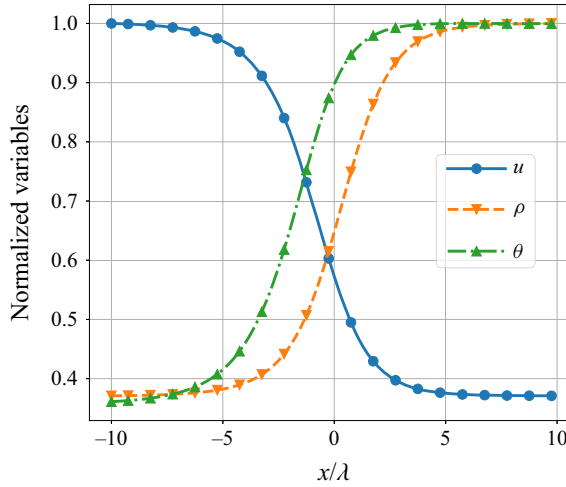


Figure 7. General structure of one-dimensional shock. Each variable is normalized with its maximum value in the computational domain.

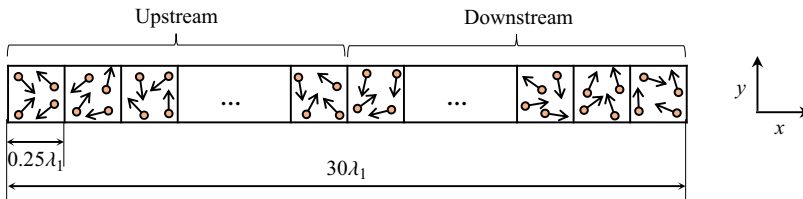


Figure 8. Schematic diagram of the computational domain for one-dimensional shock.

variables are computed with Rankine–Hugoniot relations

$$\begin{cases} T_2 = \frac{(5Ma_\infty^2 - 1)(Ma_\infty^2 + 3)}{16Ma_\infty^2} T_1, \\ \rho_2 = \frac{4Ma_\infty^2}{Ma_\infty^2 + 3} \rho_1, \\ v_2 = \frac{Ma_\infty^2 + 3}{4Ma_\infty^2} v_1. \end{cases} \quad (4.2)$$

Here, Ma_∞ and v_1 are the Mach number and velocity of the free stream, respectively. Note that for this one-dimensional problem, the velocity direction of each position in the flow field is along the x direction, and all the flow variables vary along only the x direction. Therefore, we set one cell along the y and z directions, but 120 cells along the x direction. A schematic diagram is shown in figure 8. The lengths of the computational domain and computational cell are $30\lambda_1$ and $0.25\lambda_1$, respectively. The time step is 0.1τ .

During simulation, we sample the macroscopic velocities (u), densities (ρ), temperatures (T) and heat flux in the x direction (q_x) at the instants of $t = 10\,000\tau, 10\,010\tau, 10\,020\tau, \dots, 50\,000\tau$. Then, we obtain the final data by averaging all the sample data.

In this case, the target variable is defined as the heat flux in the x direction (q_x). The function set is defined as $\{+, -, \times, \div\}$ as usual, while the terminal set is meticulously constructed to embed physics knowledge as follows.

- (i) Constitutive relations describe the local transport mechanisms of momentum and energy, and hence we regard local non-equilibrium parameters as key factors. Specifically, the GLL Knudsen number defined as $Kn_Q = (\lambda_l/Q)(\partial Q/\partial x)$ is selected. The local non-equilibrium characteristics intensify as the increase of the absolute value of Kn_Q . Here, λ_l is the local mean free path, and Q represents state variables, including temperature (θ) and density (ρ).
- (ii) The transports of momentum and energy are driven by the gradients of state variables. As a result, the gradient terms are selected, including $\partial u/\partial x$, $\partial \rho/\partial x$, $\partial \theta/\partial x$, $\partial^2 u/\partial x^2$, $\partial^2 \rho/\partial x^2$, $\partial^2 \theta/\partial x^2$, $\partial^3 u/\partial x^3$, $\partial^3 \rho/\partial x^3$ and $\partial^3 \theta/\partial x^3$.
- (iii) The state variables themselves are important factors. Besides, it is noteworthy that the constitutive relations should satisfy Galilean invariance (Han *et al.* 2019; Li, Dong & Wang 2021), which means that the constitutive relations cannot contain velocity (u) explicitly outside the partial differential operators. The proof is provided in Appendix F. Therefore, the state variables (θ and ρ) excluding velocity are selected. This is also the reason that we do not select the GLL Knudsen number of velocity.
- (iv) The parameters representing the physical properties of gas are selected, including local viscosity (μ), local heat conductivity (κ), heat capacity ratio ($\gamma = 5/3$ for argon gas) and viscosity exponent ($\omega = 1$ for Maxwell molecules).

Finally, the terminal set is $\{Kn_\theta, Kn_\rho, \partial u/\partial x, \partial \rho/\partial x, \partial \theta/\partial x, \partial^2 u/\partial x^2, \partial^2 \rho/\partial x^2, \partial^2 \theta/\partial x^2, \partial^3 u/\partial x^3, \partial^3 \rho/\partial x^3, \partial^3 \theta/\partial x^3, \rho, \theta, \mu, \kappa, \gamma, \omega\}$.

Moreover, constitutive relations ought to satisfy the second law of thermodynamics. Theoretically, whether a constitutive relation satisfies the second law of thermodynamics can be determined through the Clausius–Duhem inequality (Comeaux 1995)

$$\rho \frac{Ds_{eq}}{Dt} + \nabla \cdot \frac{\mathbf{q}}{\theta} = -\frac{1}{\theta} \boldsymbol{\tau} : \nabla \mathbf{v} - \frac{1}{\theta^2} \mathbf{q} \cdot \nabla \theta \geq 0. \tag{4.3}$$

The two terms on the left-hand side are the local increase rate of entropy and the reversible outflow of entropy, respectively. The sum of the two terms on the right-hand side is the total entropy production. The second law of thermodynamics demands that the total entropy production must be non-negative. Note that the total entropy production is the sum of subentropy productions of all high-order macroscopic variables (the viscous stress τ_{ij} and heat flux q_i). If each subentropy production is non-negative, the total entropy production is naturally non-negative. This work deals with the discovery of the constitutive equation of a single high-order macroscopic variable, rather than all of them. Hence, the entropy production discussed in this work refers to the contribution of a single high-order macroscopic variable’s constitutive relation. Specifically, in the one-dimensional shock wave case, the target variable is the heat flux in the x direction (q_x), and thus the entropy production considered here is

$$S_{p,q_x} = -\frac{1}{\theta^2} q_x \frac{\partial \theta}{\partial x}. \tag{4.4}$$

To ensure that the resulting constitutive equation satisfies the second law of thermodynamics, we introduce this law as a constraint into the loss function (2.9). Specifically, we incorporate the constraint in a soft form by adding a loss term (L_{S_p}) that

represents the ratio of the number of the data points with negative entropy production to the total number of data points. The modified loss function is

$$\text{Loss} = L_{MRE} + \alpha L_{S_p} = \frac{1}{N} \sum_{i=1}^N \left| \frac{\hat{Y}_i - Y_i}{Y_i} \right| + \alpha \frac{\langle S_p \rangle_-}{N}. \quad (4.5)$$

Here, the variable with a superscript \wedge is the predicted variable, and N is the total number of data points. Here $\langle S_p \rangle_-$ is the operator that counts the number of data points with negative entropy production, and α is the weighting factor that controls the importance of the constraint.

We employ the DHC-GEP, combined with the constraint of the second law of thermodynamics, to discover the underlying constitutive relation based on the data of two cases with the free stream Mach number $Ma_\infty = 3.0$ and 4.0 . We sequentially set α to $0.01, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9$ and 1.0 . Each α corresponds to an independent training run and a resulting equation with relatively low L_{MRE} . Among them, the equation with the smallest L_{S_p} is ultimately selected as the optimal equation. For this case, the optimal equation is

$$q_x = -0.468 \left(6Kn_\rho + \frac{Kn_\theta}{Kn_\rho} + \frac{Kn_\rho}{Kn_\theta} \right) \kappa \frac{\partial \theta}{\partial x}, \quad (4.6)$$

which is derived with $\alpha = 0.3$. The entropy production is

$$S_{p,q_x} = \frac{0.468\kappa}{\theta^2} \left(6Kn_\rho + \frac{Kn_\theta}{Kn_\rho} + \frac{Kn_\rho}{Kn_\theta} \right) \left(\frac{\partial \theta}{\partial x} \right)^2. \quad (4.7)$$

According to [figure 7](#), for the one-dimensional shock wave case, the gradients of density and temperature are non-negative. Additionally, according to the definition of the local Knudsen number ($Kn_Q = (\lambda_l/Q)(\partial Q/\partial x)$), both Kn_ρ and Kn_θ are thus non-negative. Therefore, from a mathematical perspective, the entropy production (4.7) must be non-negative. As a comparison, whether the Burnett equation, augmented-Burnett equation and super-Burnett equation satisfy the second law of thermodynamics is obscure (Comeaux 1995).

We compare the results predicted by (4.6) with those predicted by the NSF equation, the Burnett equation, the augmented-Burnett equation and the super-Burnett equation in [figure 9](#). For a quantitative comparison, we define the relative error as

$$\text{error} = \frac{\sqrt{\sum_{i=1}^N (\hat{q}_{x,i} - q_{x,i})^2}}{\sqrt{\sum_{i=1}^N q_{x,i}^2}}. \quad (4.8)$$

The relative errors for each constitutive relation are summarized in [table 4](#). It can be found that the derived constitutive relation of DHC-GEP exhibits higher accuracy than other equations over a wide range of Ma_∞ , and is applicable to cases beyond the parameter space of the training data, i.e. $2.0 \leq Ma_\infty < 3.0$ and $4.0 < Ma_\infty \leq 5.5$.

Furthermore, to investigate the sensitivity to hyperparameters in non-equilibrium cases, we conduct a parametric study. Specifically, we focus on the sensitivity to the length of

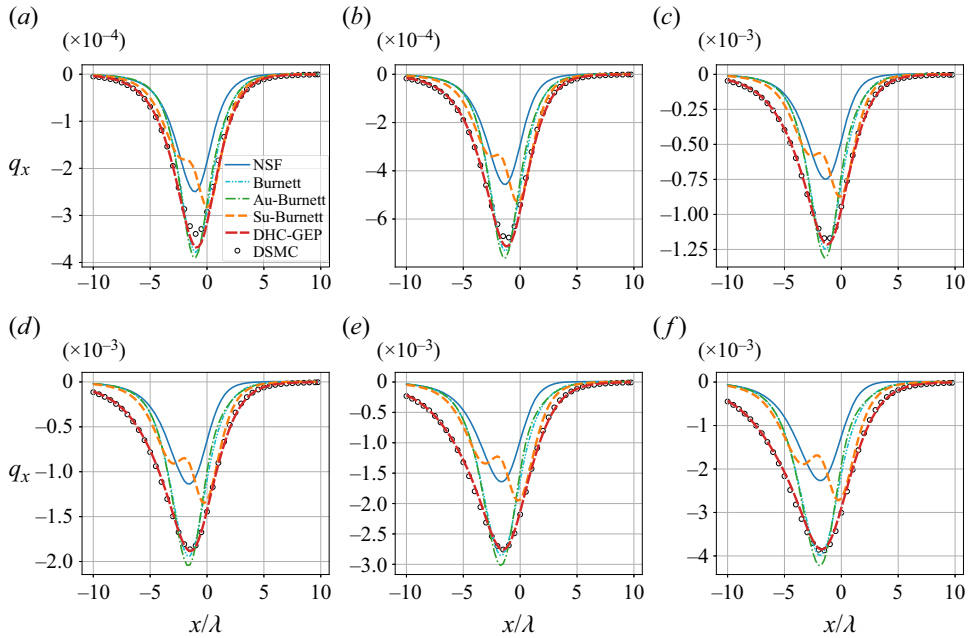


Figure 9. Comparisons of the NSF equation, Burnett equation, augmented-Burnett equation (Au-Burnett), super-Burnett equation (Su-Burnett), derived equation of DHC-GEP and DSMC (exact solution) at $Ma_\infty = 2.5, 3.0, 3.5, 4.0, 4.5$ and 5.0 for the one-dimensional shock wave case.

the head, the number of genes in a CS and the number of individuals in a population. The parameter spaces for these three hyperparameters are set to $\{10, 11, 12, 13, 14, 15\}$, $\{1, 2\}$ and $\{800, 1000, 1200, 1400, 1600, 1660\}$, respectively, resulting in a total of 72 ($6 \times 2 \times 6$) distinct hyperparameter combinations. Other hyperparameters, including the probability of invoking the genetic operators, are kept unchanged. Among the total of 72 distinct hyperparameter combinations, DHC-GEP successfully derives the same equation as (4.6) based on 38 of these hyperparameter combinations. In contrast, Original-GEP discovers 70 completely different results, all of which fail to satisfy even the fundamental dimensional homogeneity constraint, let alone other physical constraints. It can be concluded that in non-equilibrium cases, while DHC-GEP is not entirely insensitive to model hyperparameters, it does exhibit significantly reduced sensitivity compared with Original-GEP.

4.2. Rarefied Poiseuille flow

Poiseuille flow is a flow confined between two infinite, parallel and relative static plates. The gas is driven by an external force in the x direction. The external force is uniformly distributed along the y direction. A schematic diagram of Poiseuille flow is shown in figure 10. The global Knudsen number (Kn) is defined as

$$Kn = \frac{\lambda}{H}. \quad (4.9)$$

Here, H is the distance between two plates, and λ is the mean free path of argon gas molecules at standard condition. For different Kn , λ remains the same, while H changes according to (4.9). The uniform external force is implemented through a uniform

	$Ma_\infty = 2.0$	$Ma_\infty = 2.3$	$Ma_\infty = 2.5$	$Ma_\infty = 3.0$	$Ma_\infty = 3.2$	$Ma_\infty = 3.5$	$Ma_\infty = 4.0$	$Ma_\infty = 4.2$	$Ma_\infty = 4.5$	$Ma_\infty = 5.0$	$Ma_\infty = 5.5$
DHC-GEP	0.101	0.093	0.086	0.057	0.053	0.045	0.041	0.044	0.046	0.050	0.055
NSF	0.251	0.321	0.357	0.431	0.451	0.477	0.509	0.518	0.530	0.545	0.555
Burnett	0.120	0.150	0.164	0.204	0.219	0.238	0.266	0.274	0.284	0.298	0.306
Au-Burnett	0.127	0.164	0.180	0.221	0.237	0.257	0.284	0.293	0.303	0.316	0.325
Su-Burnett	0.206	0.255	0.282	0.333	0.349	0.370	0.396	0.406	0.415	0.428	0.437

Table 4. Relative errors of the derived equation of DHC-GEP, NSF equation, Burnett equation, Au-Burnett equation and Su-Burnett equation for the one-dimensional shock wave case.

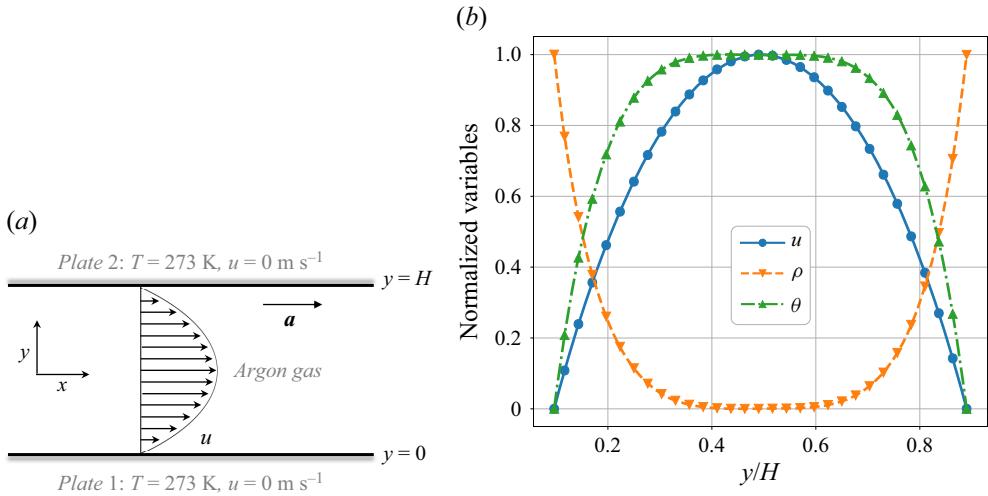


Figure 10. (a) Schematic diagram of Poiseuille flow. (b) Profiles of velocity, density and temperature along the direction normal to the plate. Each variable (U) is normalized with $U_{Normal} = (U - U_{min}) / (U_{max} - U_{min})$.

	$Kn = 0.01$	$Kn = 0.1$	$Kn = 0.2$	$Kn = 0.3$	$Kn = 0.35$	$Kn = 0.4$	$Kn = 0.5$	$Kn = 0.6$
Acceleration ($\times 10^9 \text{ m s}^{-2}$)	3.31	320	444	716	856	986	1240	1490

Table 5. Accelerations for different Kn .

acceleration in molecular simulations. The accelerations for different Kn are listed in table 5.

All the flow variables vary along only the y direction. Therefore, we set one cell along the x and z directions, but 3000 cells along the y direction. The time step is 0.2τ . During simulation, we sample the macroscopic velocities (u), densities (ρ), temperatures (T) and viscous shear stress (τ_{xy}) at the instants of $t = 100\,000\tau, 102\,000\tau, 104\,000\tau, \dots, 1\,000\,000\tau$. Then, we obtain the final data by averaging all the sample data. In addition, to avoid the influence of boundary conditions, we subsample the data points in the region of $0.1H \leq y \leq 0.9H$ to form the training dataset.

In this case, the target variable is defined as the viscous shear stress (τ_{xy}), and the function set is defined as $\{+, -, \times, \div\}$. The terminal set is almost the same as that in the case of the one-dimensional shock wave, i.e. $\{Kn_\theta, Kn_\rho, \partial u / \partial y, \partial \rho / \partial y, \partial \theta / \partial y, \rho, \theta, \mu, \gamma, \omega\}$, except that the high-order gradients are removed. The motivation for removing high-order gradients is that the derived constitutive relation should be applicable in CFD. If containing high-order gradients, the constitutive relation would be unstable and require additional boundary conditions (Bobylev 1982; Zhong *et al.* 1993; Struchtrup & Torrilhon 2003; Torrilhon & Struchtrup 2004; Singh, Jadhav & Agrawal 2017), which are the common problems with the kind of Burnett equations. Besides, note that the GLL Knudsen number in this case is defined as $Kn_Q = (\lambda_l / Q)(\partial Q / \partial y)$.

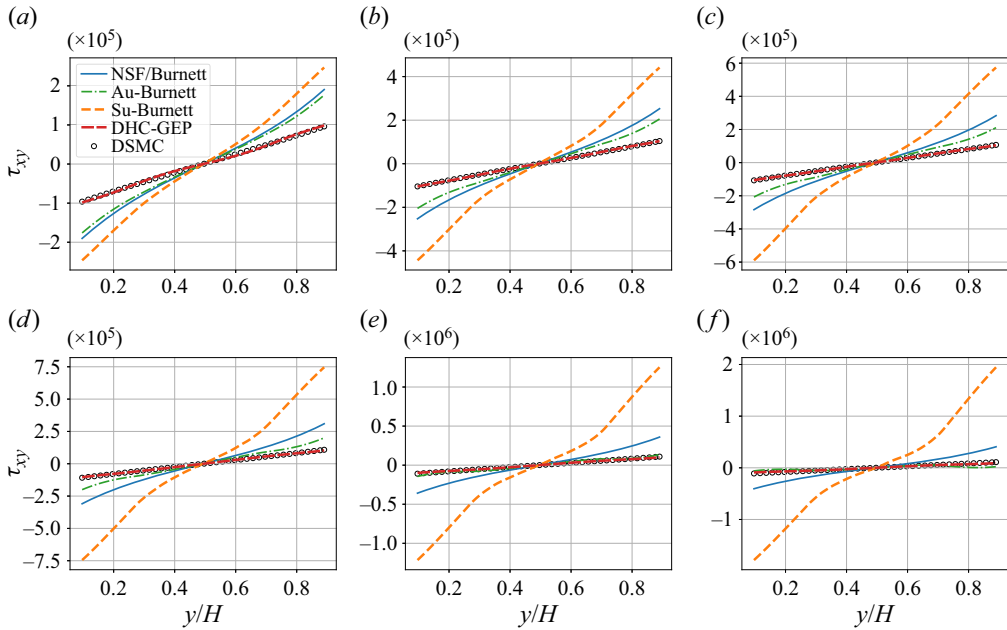


Figure 11. Comparisons of the NSF equation, Burnett equation, augmented-Burnett equation (Au-Burnett), super-Burnett equation (Su-Burnett), derived equation of DHC-GEP and DSMC (exact solution) at $Kn = 0.2, 0.3, 0.35, 0.4, 0.5$ and 0.6 for the rarefied Poiseuille flow case. Note that the NSF equation is equal to the Burnett equation in this case.

Similar to the one-dimensional shock wave case, we employ the DHC-GEP, combined with the constraint of the second law of thermodynamics, to discover the underlying constitutive relation based on the data for the cases of $Kn = 0.01, 0.3$ and 0.4 . The optimal equation is

$$\tau_{xy} = -0.247 \left(\frac{5}{Kn_\rho^2 + \gamma} - \frac{Kn_\theta}{Kn_\rho} \right) \mu \frac{\partial u}{\partial y}, \quad (4.10)$$

which is derived with $\alpha = 0.7$. The L_{S_p} for (4.10) is 0, which quantitatively indicates that the entropy productions at all data points are non-negative.

We compare the results predicted by (4.10) with those predicted by the NSF equation, the Burnett equation, the augmented-Burnett equation and the super-Burnett equation in figure 11. The relative errors for each constitutive relation are summarized in table 6. It can be found that the derived constitutive relation of DHC-GEP is much more accurate than other equations in a wide range of Kn , and is applicable to cases beyond the parameter space of the training data, i.e. $0.4 < Kn \leq 0.6$.

Moreover, note that for continuum flows, the GLL Knudsen numbers (Kn_ρ and Kn_θ) approach zero, and (4.10) reduces to

$$\tau_{xy} = -0.247 \left(\frac{5}{\gamma} - \frac{Kn_\theta}{Kn_\rho} \right) \mu \frac{\partial u}{\partial y} = -0.247 \left(3 - \frac{Kn_\theta}{Kn_\rho} \right) \mu \frac{\partial u}{\partial y}. \quad (4.11)$$

Here, we used the value of $\gamma = 5/3$ for monatomic gas in the above equation. For the Poiseuille flow in the continuum regime, the pressure along the y direction is observed to remain constant. Taking the flow of $Kn = 0.01$ as an example, the pressure oscillates

	$Kn = 0.01$	$Kn = 0.1$	$Kn = 0.2$	$Kn = 0.3$	$Kn = 0.35$	$Kn = 0.4$	$Kn = 0.5$	$Kn = 0.6$
DHC-GEP	0.028	0.083	0.044	0.019	0.025	0.039	0.076	0.137
NSF/Burnett	0.029	0.659	0.844	1.233	1.426	1.606	1.963	2.318
Au-Burnett	0.029	0.594	0.697	0.805	0.783	0.687	0.264	0.667
Su-Burnett	0.030	0.921	1.435	2.948	4.012	5.306	9.045	14.000

Table 6. Relative errors of the derived equation of DHC-GEP, NSF equation, Burnett equation, Au-Burnett equation and Su-Burnett equation for the rarefied Poiseuille flow case. Note that the NSF equation is equal to the Burnett equation in this case.

approximately 0.5 % of its absolute value. Considering the ideal gas equation of state $\rho = p/\theta$, we can obtain

$$\partial\rho = -\frac{p}{\theta^2}\partial\theta = -\frac{\rho}{\theta}\partial\theta. \tag{4.12}$$

Then, combining (4.12) with $Kn_\rho = (\lambda_l/\rho)(\partial\rho/\partial y)$ and $Kn_\theta = (\lambda_l/\theta)(\partial\theta/\partial y)$, we can conclude that $Kn_\theta/Kn_\rho = \partial\theta/\partial\rho \cdot \rho/\theta = -1$. Substituting this relation into (4.11) yields

$$\tau_{xy} = -0.247(3 + 1)\mu \frac{\partial u}{\partial y} \approx -\mu \frac{\partial u}{\partial y}. \tag{4.13}$$

Therefore, it can be concluded that in the continuum regime, the derived constitutive relation of DHC-GEP can be reduced to NSF equations. This makes it applicable to real complex flow problems, where flows tend to be multiscale, i.e. consisting of both continuum flows and non-equilibrium flows.

Finally, we emphasize that as containing only the first-order gradient of velocity, the derived constitutive relation is stable and requires the same boundary conditions as NSF equations. It is convenient to embed it into the well-developed CFD frameworks with minor modification.

5. Limitations and future works

Although DHC-GEP outperforms Original-GEP in the test cases investigated in this work, it still exhibits certain limitations. Consequently, we objectively outline several limitations associated with DHC-GEP and discuss the potential directions for future optimization as follows.

- (i) The DHC-GEP remains reliant on the prior knowledge of essential transport coefficients present in the target models. Lacking such knowledge could impede the discovery of meaningful relationships, rendering it impractical to address intricate and unknown problems. Note that dimensional transport coefficients can be derived from fundamental physical property parameters. Therefore, one alternative strategy is incorporating fundamental physical property parameters, such as the molecular mean collision time (τ) and molecular mean free path (λ), into the terminal set. Our advanced test presented in § 3.1, where the diffusion equation is discovered without incorporating the diffusion coefficient into the terminal set, has preliminarily demonstrated the feasibility of this strategy.
- (ii) The DHC-GEP necessitates computing the derivatives of training data, which can pose a notably ill-conditioned challenge, particularly in the presence of significantly noisy training data. In future investigations, a critical research direction would be

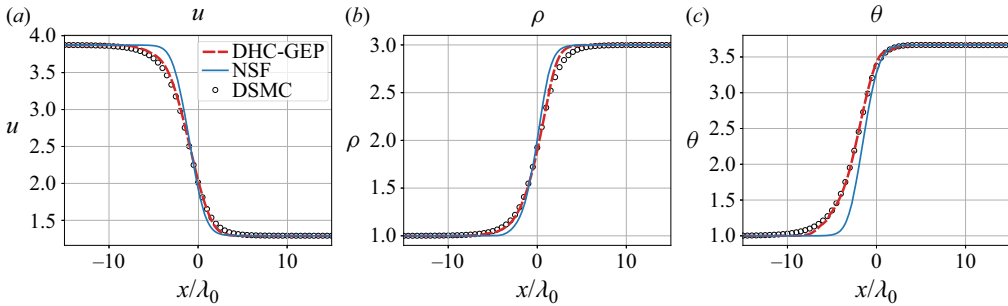


Figure 12. Comparisons of the results numerically computed with the DHC-GEP-derived/NSF constitutive relations for the one-dimensional shock wave with $Ma_\infty = 3.0$.

to explore the incorporation of the weak formulation utilized in the prior works by Gurevich *et al.* (2019), Reinbold *et al.* (2020) and Alves & Fiuza (2022) into DHC-GEP.

- (iii) As an initial study on DHC-GEP, the two non-equilibrium test cases investigated in this study are one-dimensional, and thus tensor symmetry and rotational invariance have not been considered. The target variables of DHC-GEP in the present work include scalars and one component of vectors (or tensors), which cannot be generalized to discover the equation constituting of tensor variables for higher-dimensional problems. A feasible avenue for future research is to incorporate the dimensional homogeneity constraint into multidimensional GEP (M-GEP) proposed by Weatheritt & Sandberg (2016), which modified Original-GEP for the purpose of tensor modelling. Furthermore, in our work, the constraint of dimensional homogeneity is integrated in a non-intrusive manner, which does not alter the fundamental features of Original-GEP, including CS structure, expression rules, selection and genetic operators. Therefore, the constraint of dimensional homogeneity can be conveniently incorporated into other variants of genetic programming such as M-GEP, as long as their individuals can be translated into mathematical expressions.
- (iv) The denominators of both (4.6) and (4.10) include the gradient-length local Knudsen number (Kn_ρ and Kn_θ). In numerical calculation, these two equations would become unstable and exhibit discontinuous jumps in the regions where density and temperature are uniform and both local Knudsen numbers vanish. One feasible solution is to use a relaxation technique. For instance, (4.6) can be relaxed to

$$q_x = -0.444 \left(6Kn_\rho + Kn_\theta + \frac{Kn_\theta Kn_\rho - 0.01}{Kn_\rho^2 + 0.01} + \frac{Kn_\theta Kn_\rho - 0.01}{Kn_\theta^2 + 0.01} \right) \kappa \frac{\partial \theta}{\partial x}. \quad (5.1)$$

By embedding (5.1) into an open-source CFD solver, SU2 (Economou *et al.* 2016), we can numerically solve the one-dimensional wave. Figure 12 shows the results for a case with free stream Mach number of 3.0. It can be observed that the results based on the DHC-GEP-derived constitutive relation exhibit excellent agreement with DSMC simulations. It should be noted that this merely serves as a manual correction method and does not fundamentally resolve the issue of possible singularities in the obtained equations. To enable DHC-GEP to automatically yield equations that require no further correction and can be employed in numerical computations, one possible approach is to couple the evolutionary process of DHC-GEP with the

numerical computation process. When evaluating the loss function of a particular equation, it involves more than simply computing the discrepancy between the stresses (or heat flux) predicted by the equation and those obtained from DSMC. Instead, the equation would be embedded into the CFD numerical computation program, and the equation is numerically solved. The computed velocity, density and temperature are then compared with the DSMC results to obtain the discrepancy between them as the loss value for the equation. This strategy provides a possible way to yield equations that require no further corrections and can be utilized for numerical computations, but also imposes higher demands on the computational performance of computers.

- (v) It is difficult for the present strategy (singly employing DHC-GEP) to discover a real universal governing equation. For example, the derived constitutive relation in the case of rarefied Poiseuille flow is definitely not universally valid for any non-equilibrium flows. This equation is essentially a model equation for a specific class of flows, the flow characteristics of which are similar to Poiseuille flow, instead of the real universal constitutive relation. Considering the non-equilibrium transport being complex, it is believed that the real universal constitutive relation tends to be correspondingly complex, for instance, containing high-order gradients. Although such constitutive relation is accurate, it is difficult or even impossible to embed it into the present CFD frameworks. Therefore, it is not suitable for practical engineering applications. Alternatively, we should focus on the model equations, each of which is valid for a specific class of problem and is easy to use in practice. One promising direction is combining clustering algorithms (Schmid *et al.* 2011; Callahan *et al.* 2021) with DHC-GEP. Based on a complex flow that contains a variety of flow characteristics, clustering algorithms can be first employed to divide the flow into several subflows. Then, DHC-GEP is employed in each subflow to discover the corresponding model equations. The derived constitutive relations in the present work are two examples of all model equations. During specific numerical computations, for each mesh point, we can first determine which subflow it belongs to, and then apply corresponding model equations. It is noteworthy that although the above discussions are based on non-equilibrium constitutive relations, they can also be extended to other fields.

6. Conclusions

In this work, an improved algorithm for GEP is proposed, referred to as DHC-GEP. The constraint of dimensional homogeneity is introduced to the Original-GEP method through an additional dimensional verification process. The major features of Original-GEP are not changed, including the structure of CSs, the rules of expression, selection and reproduction. Therefore, DHC-GEP inherits the advantages of Original-GEP. Specifically, DHC-GEP discovers the forms of functions and their corresponding coefficients simultaneously. The resulting equations are constructed by randomly combining basic elements in the terminal set and function set while satisfying the syntactic requirements of the mathematical expression, rather than by linearly combining the predetermined candidate functions, leading to great flexibility. The CSs in DHC-GEP have fixed length, avoiding bloating and unaffordable computational costs that are common in other evolutionary algorithms when dealing with complex problems. On the other hand, the length of open reading frame is variable, ensuring strong expressivity. The DHC-GEP is tested on two benchmark cases, including the diffusion equation and vorticity transport equation. It is demonstrated that DHC-GEP is capable of discovering the right

equations from both the views of data fitting and revealing physical principles, and the result of DHC-GEP is more robust to hyperparameters, the noise level and the size of datasets, compared with that of Original-GEP. When the data are noisy or scarce, Original-GEP tends to converge to the overfitting results with lower loss. However, these overfitting results can be automatically filtered out in DHC-GEP. Moreover, DHC-GEP is more computationally economical than Original-GEP, as DHC-GEP can identify some individuals as invalid individuals through dimensional verification and skip the process of evaluating losses for these individuals. The total cost decreases despite the extra expense of dimensional verification. These advantages make DHC-GEP a promising tool for discovering unknown governing equations from molecular simulation data.

We also present how to employ DHC-GEP to discover the unknown constitutive relations for two representative non-equilibrium flows, including a one-dimensional shock wave and rarefied Poiseuille flow. We generate the datasets by DSMC, which does not assume any governing equations. For other scientific and engineering disciplines, the datasets could be generated by experiments or first principle calculations without any assumptions of governing equations. Then, in these investigations for unknown equations, we meticulously design the terminal set to incorporate pertinent physical knowledge, such as Galilean invariance. Besides, the constraint of the second law of thermodynamics is embedded via adding an additional loss term, which is related to entropy production, to the loss function. Finally, based on the terminal and function sets, DHC-GEP conducts a global search in the space of mathematical expressions until a satisfying equation is obtained. For the two cases in our work, the derived constitutive relations are much more accurate than the conventional equations derived based on physics knowledge and phenomenological assumptions (including NSF, Burnett, augmented-Burnett equation and super-Burnet equations) in a wide range of Knudsen number and Mach number, and are even applicable to cases beyond the parameter space of the training data. In addition, the physical properties of the derived constitutive relations are excellent. Specifically, the derived constitutive relations contain only the first-order gradients, and hence are stable and require the same boundary conditions as NSF equations. As a comparison, the kind of Burnett equations are unstable, and cannot be exactly proven to satisfy the second law of thermodynamics. It is convenient to embed the derived constitutive relations into the well-developed CFD frameworks with minor modifications.

Acknowledgements. The authors thank D.A. Lockerby for providing stimulating discussions.

Funding. This work was supported by the National Natural Science Foundation of China (grant nos. 92052104 and 12272028). The results were obtained on the Zhejiang Super Cloud Computing Center M6 Partition.

Declaration of interests. The authors report no conflict of interest.

Data availability statement. The datasets and source codes used in this work are available on GitHub at <https://github.com/Wenjun-Ma/DHC-GEP>.

Author ORCIDs.

 Wenjun Ma <https://orcid.org/0000-0001-5520-1262>;

 Jun Zhang <https://orcid.org/0000-0002-3731-4594>.

Author contributions. W.M., J.Z. and D.W. contributed to the ideation and design of the research; W.M., K.F. and H.X. generated the datasets; W.M. performed the research; W.M., J.Z. and D.W. wrote the paper.

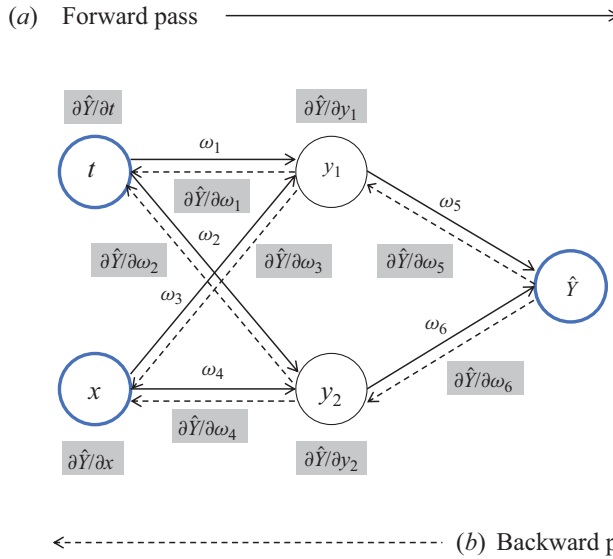


Figure 13. Schematic diagram of backpropagation. (a) The input variables are passed forward, and the corresponding target variables are predicted at the output layer. (b) The gradients are propagated backwards using the chain rule of differential calculus.

Appendix A. Automatic differentiation

The spatial and temporal derivatives are computed with AD, which conducts a non-standard interpretation of a given computer program by replacing the domain of the variables to incorporate derivative values and redefining the semantics of the operators to propagate derivatives per the chain rule of differential calculus (Baydin *et al.* 2018). Theoretically, AD can be implemented using any regression algorithms. In this work, we utilize feedforward neural networks due to their exceptional fitting capabilities, which have been widely validated in various regression problems. In theory, multilayer feedforward neural networks can approximate any complex functional relationship (Hornik, Stinchcombe & White 1989).

Specifically, we begin by constructing a standard feedforward neural network, with spatial coordinates and time as inputs, and the physical variables to be fitted as outputs (such as horizontal velocity (u), vertical velocity (v) and vorticity (ω_z) in the case of cylinder flow). All the available data are fed into the network for training, and the loss function is defined using the mean squared error (MSE) as

$$L_{MSE} = \frac{1}{N} \sum_{i=1}^N (\hat{Y}_i - Y_i)^2, \quad (A1)$$

where the variable with a superscript \wedge is the predicted variable, and N is the total number of data points. Once training is completed, the derivatives of the physical variables with respect to the input variables can be obtained by the backwards propagation of the sensitivity of the objective value at the output layer, utilizing the chain rule of differential calculus (see in figure 13).

The AD calculates the derivatives by accumulating values during code execution to produce numerical derivative evaluations, instead of derivative expressions. This method enables precise evaluation of derivatives at machine precision, with only a small constant

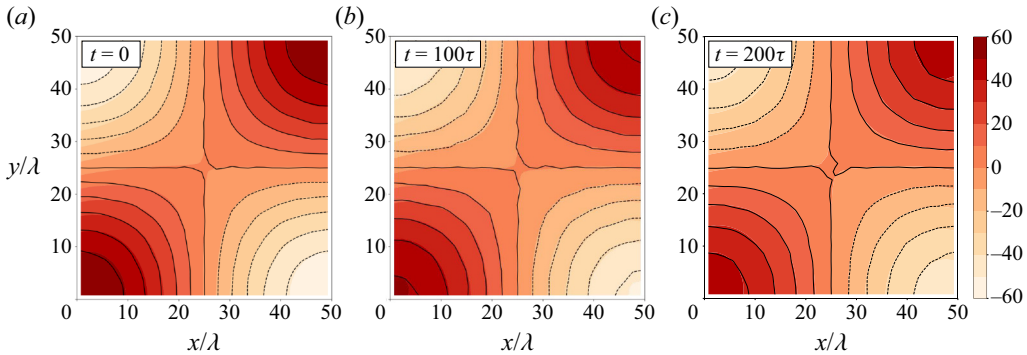


Figure 14. The theoretical values of $\partial^2 \omega_z / \partial x^2$ and the numerical values computed with automatic differentiation at the instants of $t = 0, 100\tau, 200\tau$. The contour lines represent the theoretical values, while the colour regions represent the numerical values obtained through automatic differentiation.

factor of overhead and ideal asymptotic efficiency (Baydin *et al.* 2018). It is reported in Raissi (2018) and Xu, Chang & Zhang (2019) that AD is more stable than finite differences and polynomial interpolation with respect to noise, which is beneficial for discovering the correct equations from noisy data. Another interesting point is that based on the trained neural networks, it is also convenient to combine the output variables and compute the derivatives of their combination with respect to input variables. Therefore, while not validated in this work, it is promising to incorporate partial differential operators into the function set by coupling the trained neural networks and DHC-GEP.

In this work, the feedforward neural networks have eight hidden layers, each with 20 neurons. During the training process, Adam optimization (Kingma & Ba 2014) is employed for the first 20 000 iterations, after which L-BFGS-B optimization (Byrd *et al.* 1995) is utilized until convergence.

Moreover, to demonstrate the accuracy of the derivatives computed with this method, we conduct a quantitative analysis based on the DSMC data of a Taylor–Green vortex. Specifically, according to the analytical solution of the Taylor–Green vortex given in (3.6), we can theoretically obtain the higher-order derivative term $\partial^2 \omega_z / \partial x^2$ as follows:

$$\frac{\partial^2 \omega_z}{\partial x^2} = 2v_0 \cos(x) \cos(y) \exp(-2vt) = -\omega_z. \tag{A2}$$

Figure 14 shows the theoretical values of $\partial^2 \omega_z / \partial x^2$, which equals $-\omega_z$, and the numerical values computed with automatic differentiation at the instants of $t = 0, 100\tau, 200\tau$.

At these three instants, the relative errors between the numerical and theoretical values are 0.04, 0.03 and 0.03, respectively. Hence, this analysis preliminarily indicates that the precision of the high-order derivatives obtained through automatic differentiation is satisfactory.

In addition, we generate the accurate flow field of Taylor–Green vortex based on (3.6), and then add noise of varying magnitudes into the accurate data. Specifically, the definition of noise data is consistent with the approach in Gurevich *et al.* (2021). Taking vorticity as an example, the vorticity data with noise is given by

$$\omega_{z,\sigma} = \omega_z + \sigma N_1 s_\omega. \tag{A3}$$

Here, σ is the noise level, N_1 is the sampled standard normal random variables at each point in space and time and s_ω is the sample standard deviation of ω_z . Based on the

Noise level	Relative error $\left(\frac{\partial^2 \omega_z}{\partial x^2}\right)$	Derived equation
2 %	0.053	$\frac{\partial \omega_z}{\partial t} = 1.048v \left(\frac{\partial^2 \omega_z}{\partial x^2} + \frac{\partial^2 \omega_z}{\partial y^2} \right)$
4 %	0.054	$\frac{\partial \omega_z}{\partial t} = 1.054v \left(\frac{\partial^2 \omega_z}{\partial x^2} + \frac{\partial^2 \omega_z}{\partial y^2} \right)$
6 %	0.058	$\frac{\partial \omega_z}{\partial t} = 1.040v \left(\frac{\partial^2 \omega_z}{\partial x^2} + \frac{\partial^2 \omega_z}{\partial y^2} \right)$
8 %	0.063	$\frac{\partial \omega_z}{\partial t} = 1.033v \left(\frac{\partial^2 \omega_z}{\partial x^2} + \frac{\partial^2 \omega_z}{\partial y^2} \right)$
10 %	0.134	$\frac{\partial \omega_z}{\partial t} = 1.006v \left(-\frac{\partial^2 \omega_z}{\partial x^2} + 3\frac{\partial^2 \omega_z}{\partial y^2} \right)$

Table 7. Relative error of AD-computed $\partial^2 \omega_z / \partial x^2$ and the corresponding derived governing equations at different noise levels.

noisy data, we use AD to compute the derivatives and use DHC-GEP to discover the underlying governing equations. Table 7 displays the relative error of AD-computed $\partial^2 \omega_z / \partial x^2$ compared with theoretical values, along with the corresponding derived governing equations. It can be observed that when the noise level does not exceed 8 %, AD is capable of accurately computing second-order derivative terms, and DHC-GEP can discover correct governing equations. Xu *et al.* (2019) conducted a similar study, and they found that with the assistance of the AD based on neural network without imposing additional constraints, the sparse regression could discover the correct equations from noisy data with a noise level of around 10 %. In comparison, the sparse regression using polynomial interpolation, as described by Rudy *et al.* (2017), could, at best, discover the correct equations from data with a noise level of around 1 %.

In general, given the remarkable fitting capabilities of neural networks, it is not difficult to achieve accurate fitting and derivative calculations, as long as the training data can represent the gradients. Nonetheless, it is imperative to clarify that if the training data are overly scarce and noisy, the learned neural networks would fail to calculate the derivatives correctly.

Appendix B. Dc domain

It is common to have numerical constants in governing equations. In the gene of GEP, the numerical constants are represented by a special terminal ‘?’. During expression, the ‘?’ would be replaced with the random numerical constants (RNCs) from a predefined array, under the guidance of a special domain called Dc domain. The Dc domain is an additional domain behind the gene, and consists of the indices that determine which RNCs are selected to replace the ‘?’. In this work, RNCs are randomly integers between -10 and 10. Indeed, RNCs could be selected from a broader range and are not inherently limited to integers. Nevertheless, considering the utilization of the linear scaling technique, such an extension appears unnecessary. This is because the multiplication of the scaling factor and RNCs allows for the generation of arbitrary numerical constants, which is introduced in Appendix C. Consequently, it is a common practice within the GEP community to define RNCs as integers within the specified range of (-10, 10). A simple schematic diagram is shown in figure 15. It is worth noting that, due to the introduction of negative constants,

DHC-GEP for discovering governing equations

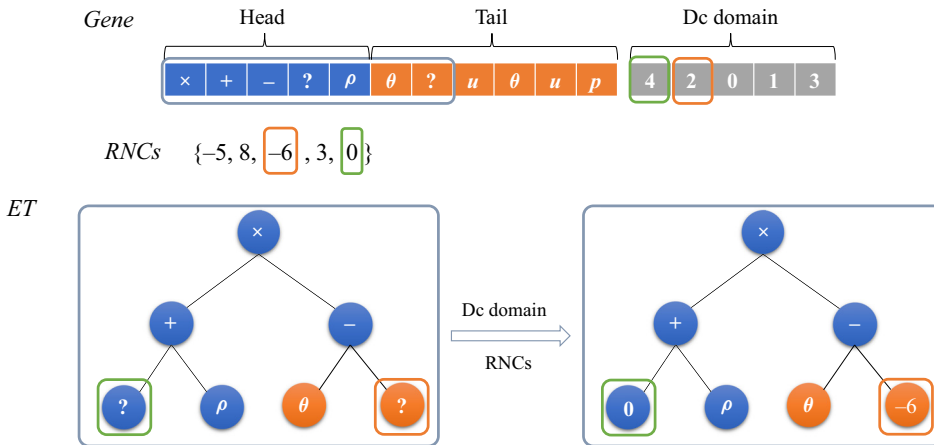


Figure 15. Schematic diagram of how the Dc domain works.

Genetic operator	Probabilities
Dc specific mutation	0.05
Dc specific inversion	0.1
Dc specific transposition	0.1
Random constant mutation	0.02

Table 8. Probabilities for genetic operators of the Dc domain being invoked.

the subtraction operator in the function set is redundant. The inclusion of the subtraction operator in this work is simply to maintain consistency with other researches relevant to GEP.

In GEP, there are several genetic operators specifically for the Dc domain and RNCs, including Dc inversion, Dc transposition and random constant mutation operators. These genetic operators have similar mechanisms to the genetic operators introduced in § 2.2.1, and are also invoked with certain probabilities, as listed in table 8.

Appendix C. Linear scaling

Generally, GEP discovers the structure of derived equations quickly, but struggles in optimizing the numerical constants. For example, if the target equation is $y = 9.7x^2$ (where y is the target variable and $9.7x^2$ is the mathematical expression that we want to derive), GEP would discover x^2 easily but struggles in discovering the numerical constant (9.7), because the number space is relatively large. To address this issue, a linear scaling technique was proposed in Keijzer (2003). Specifically, assuming that the mathematical expression of a CS is $g(x)$, then the scaling factor (ξ) can be directly obtained as

$$\xi = \frac{y}{g(x)}. \quad (C1)$$

Revisiting the previous example, as long as GEP generates a CS with its mathematical expression being ax^2 , where a can be any integer constant, the correct equation can be discovered simultaneously since the scaling factor ($9.7/a$) of this CS can be easily obtained

using (C1). In this fashion, GEP focuses on discovering the function structures, which is its *forte*, and the linear scaling technique facilitates the identification of the appropriate scaling factors.

It is important to note that in the example provided above, for the sake of simplification, $g(x)$ is represented as a simple monomial. However, it can be a linear combination of multiple functional terms, such as $g(x) = ax^3 + bx^2 + cx$. Here, a , b and c are integer constants, which are generated using the rules introduced in Appendix B. The final equation is $y = \xi g(x) = a\xi x^3 + b\xi x^2 + c\xi x$. Hence, the numerical constants are determined by a joint influence of the scaling factor (ξ) and the integer constants within the CS.

Appendix D. Loss function

In this work, to evaluate the performances of individuals, we define the loss function using the MRE as

$$L_{MRE} = \frac{1}{N} \sum_{i=1}^N \left| \frac{\hat{Y}_i - Y_i}{Y_i} \right|. \quad (D1)$$

Here, the variable with a superscript $\hat{}$ is the predicted variable, and N is the total number of data points. In MRE, each data point receives equal attention, making it highly suitable for this study. Since the constitutive relations reflect local momentum and energy transport characteristics, it is imperative to treat each data point with equal attention.

Moreover, we emphasize that the selection of loss function should be jointly determined based on the distribution of training data and the requirements of problems. It is challenging to ascertain that a particular loss function is always the optimal choice for any given problem. The MRE is suitable for this study, but it is not widely applicable to other problems. The MRE is sensitive to outliers in the data. A single outlier can significantly increase the relative error, affecting the stability and performance of the model during training. In problems where strict equal attention to each data point is not needed, the Huber loss (Huber 1992) is a better alternative, which is better equipped to handle outliers.

Appendix E. Hyperparameters

The three key hyperparameters in the GEP method are the length of the head, the number of genes in a CS and the number of individuals in a population. The length of the head and the number of genes in a CS determine the upper limit of the complexity of the derived equation. The number of individuals in a population determines the diversity of individuals in a population. Generally, a larger population means a greater diversity of individuals, and a higher computational cost in an evolution as well. According to the parametric study on the diffusion flow case (§ 3.1), we find that Original-GEP is quite sensitive to these three hyperparameters. However, this is not the issue of DHC-GEP. Therefore, we keep all hyperparameters consistent across all cases. Specifically, the length of the head is set to 15, and the number of genes a CS is set to 2, ensuring that the upper limit of the complexity of equations DHC-GEP could explore is sufficiently high. The number of individuals in a population is set to 1660 to ensure that there are enough varieties of equations in each generation.

The possibilities of the genetic operators being selected refer to Ferreira (2006), listed in tables 1 and 8, which were concluded from various examples via the trial-and-error approach.

Appendix F. Proof of Galilean invariance

Galilean invariance is a fundamental property of physical laws, which has been proven to be important for constructing constitutive relations with data-driven methods (Han *et al.* 2019, 2020; Huang *et al.* 2021; Li *et al.* 2021). Specifically, this means that the equation forms of physical laws remain invariant in all inertial frames. Assuming that one inertial frame ($x'y'z'$) moves at a constant speed (v_0) with respect to another inertial frame (xyz), the transformations of spatiotemporal coordinates between these two frames are

$$\begin{cases} \mathbf{r}' = \mathbf{r} - \mathbf{v}_0 t, \\ t' = t. \end{cases} \quad (\text{F1})$$

Here, \mathbf{r} and \mathbf{r}' are the radius vectors. Moreover, the macroscopic state variables satisfy

$$\begin{cases} \mathbf{v}' = \mathbf{v} - \mathbf{v}_0, \\ \rho', \theta', p' = \rho, \theta, p. \end{cases} \quad (\text{F2})$$

The partial differential operators have the following relations:

$$\begin{cases} \frac{\partial U}{\partial t'} = \frac{\partial U}{\partial t} \cdot \frac{\partial t}{\partial t'} + \frac{\partial U}{\partial \mathbf{r}} \cdot \frac{\partial \mathbf{r}}{\partial t} \cdot \frac{\partial t}{\partial t'} = \frac{\partial U}{\partial t} + \mathbf{v}_0 \frac{\partial U}{\partial \mathbf{r}}, \\ \frac{\partial U}{\partial \mathbf{r}'} = \frac{\partial U}{\partial t} \cdot \frac{\partial t}{\partial \mathbf{r}'} + \frac{\partial U}{\partial \mathbf{r}} \cdot \frac{\partial \mathbf{r}}{\partial \mathbf{r}'} = \frac{\partial U}{\partial \mathbf{r}}, \\ \frac{\partial^n U}{\partial \mathbf{r}'^n} = \frac{\partial}{\partial \mathbf{r}'} \left(\frac{\partial^{n-1} U}{\partial \mathbf{r}'^{n-1}} \right) = \frac{\partial}{\partial \mathbf{r}} \left(\frac{\partial^{n-1} U}{\partial \mathbf{r}^{n-1}} \right) = \dots = \frac{\partial^n U}{\partial \mathbf{r}^n}, \end{cases} \quad (\text{F3})$$

where U represents variables that are relevant to \mathbf{r} and t .

Assuming that the constitutive relation for viscous stress contains velocity (\mathbf{v}) explicitly outside the partial differential operators, a simple but representative example is

$$\boldsymbol{\tau} = \mathbf{v} \frac{\partial U}{\partial \mathbf{r}}. \quad (\text{F4})$$

In the inertial frame ($x'y'z'$), it can be derived that

$$\boldsymbol{\tau}' = \mathbf{v}' \frac{\partial U}{\partial \mathbf{r}'} = (\mathbf{v} - \mathbf{v}_0) \frac{\partial U}{\partial \mathbf{r}} \neq \boldsymbol{\tau}, \quad (\text{F5})$$

which means that such a constitutive relation does not satisfy the Galilean invariance.

On the contrary, if the constitutive relations do not contain velocity (\mathbf{v}) explicitly outside the partial differential operators, it is straightforward to prove that they would remain invariant in different inertial frames, i.e. satisfying the Galilean invariance, according to (F2)–(F3).

REFERENCES

- ALVES, E.P. & FIUZA, F. 2022 Data-driven discovery of reduced plasma physics models from fully kinetic simulations. *Phys. Rev. Res.* **4** (3), 033192.
- BAYDIN, A.G., PEARLMUTTER, B.A., RADUL, A.A. & SISKIND, J.M. 2018 Automatic differentiation in machine learning: a survey. *J. Machine Learning Res.* **18**, 1–43.
- BEETHAM, S. & CAPECELATRO, J. 2020 Formulating turbulence closures using sparse regression with embedded form invariance. *Phys. Rev. Fluids* **5** (8), 084611.
- BERGEN, K.J., JOHNSON, P.A., DE HOOP, M.V. & BEROZA, G.C. 2019 Machine learning for data-driven discovery in solid earth geoscience. *Science* **363** (6433), eaau0323.

- BIRD, G.A. 1981 Monte-carlo simulation in an engineering context. *Prog. Astronaut. Aeronaut.* **74**, 239–255.
- BIRD, G.A. 1994 *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*. Clarendon.
- BOBYLEV, A.V. 1982 The Chapman-Enskog and grad methods for solving the Boltzmann equation. *Akad. Nauk SSSR Dokl.* **262** (1), 71–75.
- BONGARD, J. & LIPSON, H. 2007 Automated reverse engineering of nonlinear dynamical systems. *Proc. Natl Acad. Sci. USA* **104** (24), 9943–9948.
- BOYD, I.D., CHEN, G. & CANDLER, G.V. 1995 Predicting failure of the continuum fluid equations in transitional hypersonic flows. *Phys. Fluids* **7** (1), 210–219.
- 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.
- BURNETT, D. 1936 The distribution of molecular velocities and the mean motion in a non-uniform gas. *Proc. Lond. Math. Soc.* **2** (1), 382–435.
- BYRD, R.H., LU, P., NOCEDAL, J. & ZHU, C. 1995 A limited memory algorithm for bound constrained optimization. *SIAM J. Sci. Comput.* **16** (5), 1190–1208.
- CALLAHAM, J.L., KOCH, J.V., BRUNTON, B.W., KUTZ, J.N. & BRUNTON, S.L. 2021 Learning dominant physical processes with data-driven balance models. *Nat. Commun.* **12** (1), 1–10.
- CHAMPION, K., LUSCH, B., KUTZ, J.N. & BRUNTON, S.L. 2019a Data-driven discovery of coordinates and governing equations. *Proc. Natl Acad. Sci. USA* **116** (45), 22445–22451.
- CHAMPION, K.P., BRUNTON, S.L. & KUTZ, J.N. 2019b Discovery of nonlinear multiscale systems: sampling strategies and embeddings. *SIAM J. Appl. Dyn. Syst.* **18** (1), 312–333.
- CHAPMAN, S. & COWLING, T.G. 1990 *The Mathematical Theory of Non-Uniform Gases: An Account of the Kinetic Theory of Viscosity, Thermal Conduction and Diffusion in Gases*. Cambridge University Press.
- COMEAUX, K. 1995 *An Evaluation of the Second Order Constitutive Relations for Rarefied Gas Dynamics based on the Second Law of Thermodynamics*. Stanford University.
- ECONOMON, T.D., PALACIOS, F., COPELAND, S.R., LUKACZYK, T.W. & ALONSO, J.J. 2016 Su2: an open-source suite for multiphysics simulation and design. *AIAA J.* **54** (3), 828–846.
- FERREIRA, C. 2001 Gene expression programming: a new adaptive algorithm for solving problems. [arXiv:cs/0102027](https://arxiv.org/abs/cs/0102027).
- FERREIRA, C. 2006 *Gene Expression Programming: Mathematical Modeling by an Artificial Intelligence*. Springer.
- 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.
- GUASTONI, L., GÜEMES, A., IANIRO, A., DISCETTI, S., SCHLATTER, P., AZIZPOUR, H. & VINUESA, R. 2021 Convolutional-network models to predict wall-bounded turbulence from wall quantities. *J. Fluid Mech.* **928**, A27.
- GUREVICH, D.R., REINBOLD, P.A.K. & GRIGORIEV, R.O. 2019 Robust and optimal sparse regression for nonlinear pde models. *Chaos* **29** (10), 103113.
- GUREVICH, D.R., REINBOLD, P.A.K. & GRIGORIEV, R.O. 2021 Learning fluid physics from highly turbulent data using sparse physics-informed discovery of empirical relations (spider). [arXiv:2105.00048](https://arxiv.org/abs/2105.00048).
- HADJICONSTANTINOU, N.G. 2000 Analysis of discretization in the direct simulation Monte Carlo. *Phys. Fluids* **12** (10), 2634–2638.
- HAN, G.-F., LIU, X.-L., HUANG, J., NAWNIT, K. & SUN, L. 2020 Alternative constitutive relation for momentum transport of extended Navier–Stokes equations. *Chin. Phys. B* **29** (12), 124701.
- HAN, J., MA, C., MA, Z. & E, W. 2019 Uniformly accurate machine learning-based hydrodynamic models for kinetic equations. *Proc. Natl Acad. Sci. USA* **116** (44), 21983–21991.
- HEINBOCKEL, J.H. 2001 *Introduction to Tensor Calculus and Continuum Mechanics*, vol. 52. Trafford.
- HORNIK, K., STINCHCOMBE, M. & WHITE, H. 1989 Multilayer feedforward networks are universal approximators. *Neural Netw.* **2** (5), 359–366.
- HUANG, J., MA, Z., ZHOU, Y. & YONG, W.-A. 2021 Learning thermodynamically stable and galilean invariant partial differential equations for non-equilibrium flows. *J. Non-Equilib. Thermodyn.* **46** (4), 355–370.
- HUBER, P.J. 1992 Robust estimation of a location parameter. In *Breakthroughs in Statistics: Methodology and Distribution* (ed. S. Kotz & N.L. Johnson), pp. 492–518. Springer.
- JUNIPER, M.P. 2023 Machine learning for thermoacoustics. In *Machine Learning and its Application to Reacting Flows: ML and Combustion* (ed. N. Swaminathan & A. Parente), pp. 307–337. Springer.
- KAHEMAN, K., KUTZ, J.N. & BRUNTON, S.L. 2020 Sindy-pi: a robust algorithm for parallel implicit sparse identification of nonlinear dynamics. *Proc. R. Soc. A* **476** (2242), 20200279.

- KAISER, E., KUTZ, J.N. & BRUNTON, S.L. 2018 Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. *Proc. R. Soc. A* **474** (2219), 20180335.
- KARNIADAKIS, G.E., KEVREKIDIS, I.G., LU, L., PERDIKARIS, P., WANG, S. & YANG, L. 2021 Physics-informed machine learning. *Nat. Rev. Phys.* **3** (6), 422–440.
- KEIJZER, M. 2003 Improving symbolic regression with interval arithmetic and linear scaling. In *Genetic Programming: 6th European Conference, EuroGP 2003 Essex* (ed. C. Ryan, T. Soule, M. Keijzer, E. Tsang, R. Poli & E. Costa), pp. 70–82. Springer.
- KINGMA, D.P. & BA, J. 2014 Adam: a method for stochastic optimization. [arXiv:1412.6980](https://arxiv.org/abs/1412.6980).
- KOCH-JANUSZ, M. & RINGEL, Z. 2018 Mutual information, neural networks and the renormalization group. *Nat. Phys.* **14** (6), 578–582.
- LI, Z., DONG, B. & WANG, Y. 2021 Learning invariance preserving moment closure model for Boltzmann-bgk equation. [arXiv:2110.03682](https://arxiv.org/abs/2110.03682).
- LING, J., KURZAWSKI, A. & TEMPLETON, J. 2016 Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *J. Fluid Mech.* **807**, 155–166.
- LOISEAU, J.-C. & BRUNTON, S.L. 2018 Constrained sparse galerkin regression. *J. Fluid Mech.* **838**, 42–67.
- LOISEAU, J.-C., NOACK, B.R. & BRUNTON, S.L. 2018 Sparse reduced-order modelling: sensor-based dynamics to full-state estimation. *J. Fluid Mech.* **844**, 459–490.
- LONG, Z., LU, Y. & DONG, B. 2019 Pde-net 2.0: learning pdes from data with a numeric-symbolic hybrid deep network. *J. Comput. Phys.* **399**, 108925.
- LONG, Z., LU, Y., MA, X. & DONG, B. 2018 PDE-net: learning PDEs from data. In *Proceedings of the 35th International Conference on Machine Learning* (ed. J. Dy & A. Krause), **80**, pp. 3208–3216. PMLR.
- LUSCH, B., KUTZ, J.N. & BRUNTON, S.L. 2018 Deep learning for universal linear embeddings of nonlinear dynamics. *Nat. Commun.* **9** (1), 1–10.
- MA, W., ZHANG, J. & YU, J. 2021 Non-intrusive reduced order modeling for flowfield reconstruction based on residual neural network. *Acta Astronaut.* **183**, 346–362.
- MANGAN, N.M., BRUNTON, S.L., PROCTOR, J.L. & KUTZ, J.N. 2016 Inferring biological networks by sparse identification of nonlinear dynamics. *IEEE Trans. Mol. Biol. Multi-Scale Commun.* **2** (1), 52–63.
- ORAN, E.S., OH, C.K. & CYBYK, B.Z. 1998 Direct simulation Monte Carlo: recent advances and applications. *Annu. Rev. Fluid Mech.* **30**, 403.
- PARK, J. & CHOI, H. 2021 Toward neural-network-based large eddy simulation: application to turbulent channel flow. *J. Fluid Mech.* **914**, A16.
- RAISSI, M. 2018 Deep hidden physics models: deep learning of nonlinear partial differential equations. *J. Machine Learning Res.* **19** (1), 932–955.
- REINBOLD, P.A.K. & GRIGORIEV, R.O. 2019 Data-driven discovery of partial differential equation models with latent variables. *Phys. Rev. E* **100** (2), 022219.
- REINBOLD, P.A.K., GUREVICH, D.R. & GRIGORIEV, R.O. 2020 Using noisy or incomplete data to discover models of spatiotemporal dynamics. *Phys. Rev. E* **101** (1), 010203.
- REINBOLD, P.A.K., KAGEORGE, L.M., SCHATZ, M.F. & GRIGORIEV, R.O. 2021 Robust learning from noisy, incomplete, high-dimensional experimental data via physically constrained symbolic regression. *Nat. Commun.* **12** (1), 1–8.
- 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.
- SAHOO, S., LAMPERT, C. & MARTIUS, G. 2018 Learning equations for extrapolation and control. In *Proceedings of the 35th International Conference on Machine Learning* (ed. J. Dy & A. Krause), **80**, pp. 4442–4450. PMLR.
- SCHAEFFER, H. 2017 Learning partial differential equations via data discovery and sparse optimization. *Proc. R. Soc. A* **473** (2197), 20160446.
- SCHMID, P.J., LI, L., JUNIPER, M.P. & PUST, O. 2011 Applications of the dynamic mode decomposition. *Theor. Comput. Fluid Dyn.* **25** (1), 249–259.
- SCHMIDT, M. & LIPSON, H. 2009 Distilling free-form natural laws from experimental data. *Science* **324** (5923), 81–85.
- SENGUPTA, U., AMOS, M., HOSKING, S., RASMUSSEN, C.E., JUNIPER, M. & YOUNG, P. 2020 Ensembling geophysical models with Bayesian neural networks. *Adv. Neural Inf. Process. Syst.* **33**, 1205–1217.
- SHAVALIYEV, M.S. 1993 Super-Burnett corrections to the stress tensor and the heat flux in a gas of Maxwellian molecules. *J. Appl. Math. Mech.* **57** (3), 573–576.
- SINGH, N., JADHAV, R.S. & AGRAWAL, A. 2017 Derivation of stable Burnett equations for rarefied gas flows. *Phys. Rev. E* **96** (1), 013106.

- 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. & TORRILHON, M. 2003 Regularization of grad’s 13 moment equations: derivation and linear analysis. *Phys. Fluids* **15** (9), 2668–2680.
- SUN, Q. & BOYD, I.D. 2002 A direct simulation method for subsonic, microscale gas flows. *J. Comput. Phys.* **179** (2), 400–425.
- TORRILHON, M. & STRUCHTRUP, H. 2004 Regularized 13-moment equations: shock structure calculations and comparison to Burnett models. *J. Fluid Mech.* **513**, 171–198.
- VADDIREDDY, H., RASHEED, A., STAPLES, A.E. & SAN, O. 2020 Feature engineering and symbolic regression methods for detecting hidden physics from sparse sensor observation data. *Phys. Fluids* **32** (1), 015113.
- WAGNER, W. 1992 A convergence proof for bird’s direct simulation monte carlo method for the Boltzmann equation. *J. Stat. Phys.* **66** (3), 1011–1044.
- WEATHERITT, J. & SANDBERG, R. 2016 A novel evolutionary algorithm applied to algebraic modifications of the rans stress–strain relationship. *J. Comput. Phys.* **325**, 22–37.
- WEINAN, E. 2021 The dawning of a new era in applied mathematics. *Not. Am. Math. Soc.* **68** (4), 565–571.
- XING, H., ZHANG, J., MA, W. & WEN, D. 2022 Using gene expression programming to discover macroscopic governing equations hidden in the data of molecular simulations. *Phys. Fluids* **34** (5), 057109.
- XU, H., CHANG, H. & ZHANG, D. 2019 Dl-pde: deep-learning based data-driven discovery of partial differential equations from discrete and noisy data. [arXiv:1908.04463](https://arxiv.org/abs/1908.04463).
- YU, C., YUAN, Z., QI, H., WANG, J., LI, X. & CHEN, S. 2022 Kinetic-energy-flux-constrained model using an artificial neural network for large-eddy simulation of compressible wall-bounded turbulence. *J. Fluid Mech.* **932**, A23.
- 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. & MA, W. 2020 Data-driven discovery of governing equations for fluid dynamics based on molecular simulation. *J. Fluid Mech.* **892**, A5.
- ZHENG, P., ASKHAM, T., BRUNTON, S.L., KUTZ, J.N. & ARAVKIN, A.Y. 2018 A unified framework for sparse relaxed regularized regression: Sr3. *IEEE Access* **7**, 1404–1423.
- ZHONG, X., MACCORMACK, R.W. & CHAPMAN, D.R. 1993 Stabilization of the Burnett equations and application to hypersonicflows. *AIAA J.* **31** (6), 1036–1043.