Journal of Computational Physics 424 (2021) 109859

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Application of Gene Expression Programming to a-posteriori LES modeling of a Taylor Green Vortex

Maximilian Reissmann [a,b], Josef Hasslberger [b], Richard D. Sandberg [a], Markus Klein [b,*]

[a] Department of Mechanical Engineering, University of Melbourne, Parkville, VIC, 3010, Australia
[b] Department of Aerospace Engineering, Universitaet der Bundeswehr Muenchen, Neubiberg, 85577, Germany

ARTICLE INFO

Article history:
Received 10 March 2020
Received in revised form 27 July 2020
Accepted 17 September 2020
Available online 22 September 2020

Keywords:
Evolutionary algorithm
Gene Expression Programming
In-the-loop optimization
A-posteriori LES
Taylor Green Vortex

ABSTRACT

Gene Expression Programming (GEP), a branch of machine learning, is based on the idea to iteratively improve a population of candidate solutions using an evolutionary process built on the survival-of-the-fittest concept. The GEP approach was initially applied with encouraging results to the modeling of the unclosed tensors in the context of RANS (Reynolds Averaged Navier–Stokes) turbulence modeling. In a subsequent study it was demonstrated that the GEP concept can also be successfully used for modeling the unknown Sub-Grid Stress (SGS) tensor in the context of Large Eddy Simulations (LES). This was done in an a-priori analysis, where an existing Direct Numerical Simulation (DNS) database was explicitly filtered to evaluate the unknown stresses and to assess the performance of model candidates suggested by GEP. This paper presents the next logical step, i.e. the application of GEP to a-posteriori LES model development. Because a-posteriori analysis, using in-the-loop optimization, is considered the ultimate way to test SGS models, this can be considered an important milestone for the application of machine learning to LES based turbulence modeling. GEP is here used to train LES models for simulating a Taylor Green Vortex (TGV) and results are compared with existing standard models. It is shown that GEP finds a model that outperforms known models from literature as well as the no-model LES. Although the performance of this best model is maintained for resolutions and Reynolds numbers different from the training data, this is not automatically guaranteed for all other models suggested by the algorithm.

1. Introduction

Data from experiments and DNS have historically been used to calibrate turbulence models. The increasing size and quality of the datasets, together with algorithmic innovations and advances in computer hardware have allowed the use of more complex algorithms that infer not just closure constants but also functional forms. As a result machine-learning algorithms have become a popular tool for improving turbulence models. For a recent overview the reader is referred to [1]. Machine learning includes a wide range of techniques, essentially sophisticated curve-fitting algorithms, within the broader field of artificial intelligence [1]. This concept is by no means new in turbulence modeling (e.g. [2,3]) but it has recently attracted considerable attention. Typically, the algorithms have been applied to closing the RANS equations, e.g.

* Corresponding author.
E-mail address: markus.klein@unibw.de (M. Klein).

https://doi.org/10.1016/j.jcp.2020.109859
0021-9991/

[4–7], although the scalar flux [8,9] equations and hybrid RANS/LES approaches [10] have also been tackled. Activities in LES include the use of neural networks to model subgrid-scale stresses [11,12], to estimate turbulent sub-grid scale reaction rates [13,14], sub-grid curvature effects in two phase flow simulations [15] or LES wall modeling [16]. GEP has been used recently [17] for modeling the unknown sub-grid stress tensor in the context of a-priori analysis. It has been also demonstrated in [18] that the sparse regression technique STRidge and the evolutionary optimization algorithm GEP are effective tools for identifying hidden physical laws from observed data. The aforementioned studies can be split into two categories: those that are transparent and those that are not. Essentially, this dictates whether the non-linearity of the machine-learnt model exists on a level of description interpretable by a human [17] or not. For methodologies such as random forests and neural networks, the non-linearity is built over hundreds, if not thousands of interactions. The existence of complexity at the lowest level forces the user to treat the model as a non-transparent black box. Diagnosing problems and sharing models with the community are thus not straightforward. For symbolic regression, such as in GEP, the model inferred is a mathematical expression. Therefore, using GEP for turbulence modeling is an emerging field of research. This method has the advantage, relative to other machine learning models such as artificial neural networks, that it produces a turbulence model as a function of key physical parameters. The obtained function can be readily implemented, and is also repeatable and provides insight into the phenomenon of interest [19]. For these reasons the GEP approach is adopted in this study as the model can be interpreted and easily implemented in existing LES solvers. The work presented in [17] was based on a-priori analysis, i.e. filtering of DNS data. Such analysis, however, has limitations [20], and it cannot be guaranteed that findings based on a-priori analysis necessarily result in improved performance in an actual LES simulation. The ultimate test of an LES must therefore consist of comparing actual simulation results to measurements or DNS; this is known as a posteriori testing [21]. Hence, the present optimization strategy is based on a-posteriori analysis, i.e. it uses tens of thousands in-the-loop LES runs to find a good LES representation of the DNS based reference data. It is worth mentioning that model development based on a-posteriori LES is not only more expensive, but also much more intricate than in the case of a-priori LES, because of the close interaction of modelling and numerical errors. Furthermore, there is no guarantee that every candidate model suggested by GEP will result in a stable LES. This paper is structured as follows: first the methodology is presented, outlining the overall algorithm for in-the-loop optimization and providing a short summary of the GEP approach. Then the performance of the trained models is presented on the training case, assessing their generalizability to other grid resolutions and flow conditions.

2. Methodology

This section outlines the numerical methodology and the computational configuration as well as details regarding the GEP framework used for the optimization.

2.1. Numerical method and computational configuration

One of the most demanding test cases for SGS models is laminar-turbulent transition [22]. As an example for a transitional flow, a Taylor Green vortex is analysed in this work. The configuration consists of a cube with side length 2π and periodic boundaries in all directions. The velocity field is initialised as follows and subsequently integrated forward in time (see Fig. 1):

$$u(x,y,z) = \cos(x)\sin(y)\sin(z), \quad v(x,y,z) = -\sin(x)\cos(y)\sin(z), \quad w(x,y,z) = 0\,. \tag{1}$$

After initialization of the flow, vortices roll up and start interacting with each other and then break down and transition to turbulence. In the final stage the flow can be considered fully turbulent. The flow is treated incompressible with constant density. Two different resolutions have been employed, a benchmark DNS conducted on a 256^3 cube and a relatively coarse 32^3 cube for all a-posteriori LES runs. In the initial state, the Reynolds number is 1600. By comparing the present simulation results with the spectral simulation of Brachet et al. [23] an excellent agreement was found for the dissipation of kinetic energy, apart from a small under-prediction of the peak dissipation value. In order to test the robustness of the model for parameters different to the training dataset, two additional Reynolds numbers have been considered $Re = 400, 3000$ and two additional LES resolutions of $64^3, 128^3$. The non-dimensional simulation time ranges from $t = 0$ to $t = T = 25$. It is admitted that the TGV with $Re = 3000$ would require a slightly higher resolution in the DNS. However, this moderate underresolution does not really matter in the context of this work and in particular for the statistics considered here. The open-source code PARIS [24] has been employed for the simulations. It uses a projection method, including a second-order predictor-corrector technique for time integration for solving the incompressible Navier–Stokes equations. Spatial discretization is realized by the finite-volume approach on a regular, cubic, staggered grid with second-order centred difference schemes.

2.2. GEP algorithm

The GEP algorithm is implemented in Python, which is executed by an interpreter. The individual instructions are translated at run time. The sequence of instructions of this software component and its communication with the CFD solver can be summarised as follows:

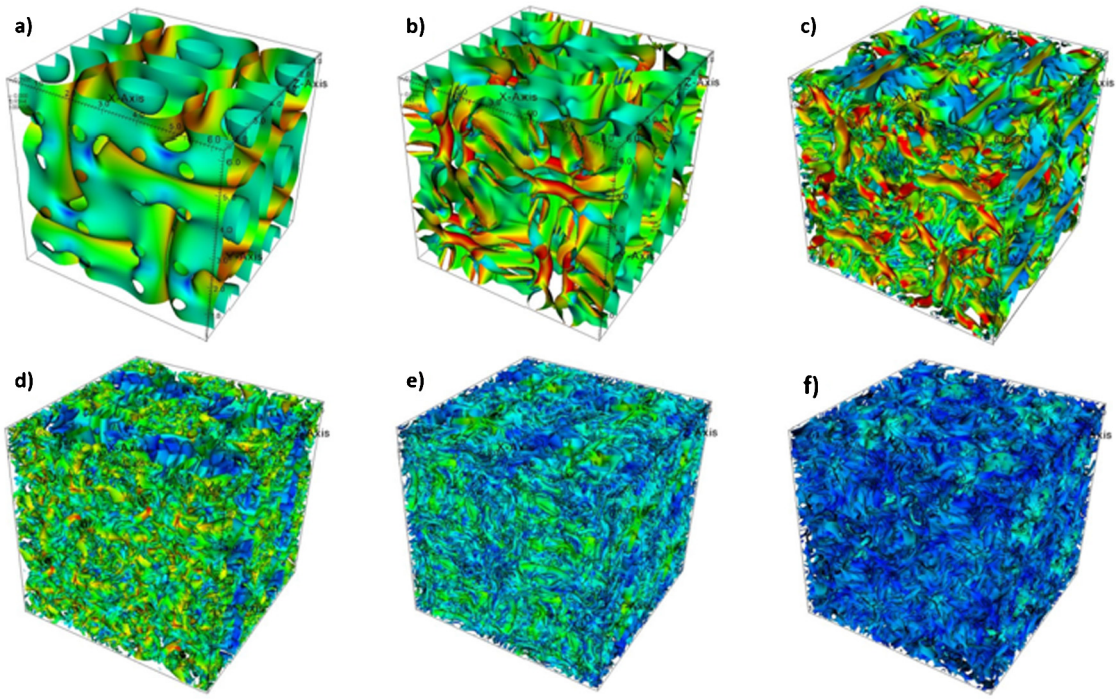

Fig. 1. Instantaneous views of the second invariant of the velocity gradient tensor $Q = 0$ iso-contours extracted from the reference DNS and coloured with velocity magnitude for the TGV at non-dimensional simulation times (a)-(f): $t = 2.5, 5.0, 7.5, 10, 15, 25$. (For interpretation of the colours in the figure(s), the reader is referred to the web version of this article.)

1. Start program by specifying a parameter set consisting of a function set, random numerical constants, probabilities describing the evolution process, etc.
2. Run GEP
   (a) Generate population
   (b) Send functions (phenotype of individuals) to simulation
   (c) Wait for a signal (LES finished)
   (d) Evaluate the results from the simulation, by using the DNS and LES statistics files and a cost function and pass them to the GEP algorithm
   (e) Update population
   (f) Go to step 2(a)
3. Write best function to file

Steps 2(b) and 2(d) of the above algorithm require an interface between the GEP code and the CFD software. Based on the evaluation of a randomly generated population, the functions (phenotype of an individual) are transferred to the interface. The interface translates them into a Fortran code, creates a number of solver instances and overwrites the subgrid scale term at the corresponding location. This implies conducting an LES for each candidate model. Then the Fortran modules are compiled and the scheduler is activated. Steps 2(a) and 2(e) are sketched in the flowchart in Fig. 2. All simulations have been performed on a small Linux Cluster such that all model candidates within one generation can be assessed in parallel.

The hyper parameters chosen for the simulation are given in Table 1. They have been selected by relying on previous work and experiences [4,9,10,17,19,25,26] and no extensive tuning was required and done. Only the number of individuals and the number of generations was adjusted to the available computational resources and time frame for this project.

2.3. Gene expression programming

GEP will be used in this work for modeling the unclosed SGS stress tensor in the context of LES. In order to derive the LES formalism the Navier–Stokes equations are usually filtered with a linear commutative filter, i.e. $\partial\overline{\varphi} = \overline{\partial\varphi}$ where φ denotes a general variable and $\bar{\cdot}$ a filtering operation

$$\frac{\partial}{\partial t}\overline{u_i} = -\frac{\partial}{\partial x_j}\left(\overline{u_i u_j}\right) + \frac{\partial}{\partial x_j}\nu\left(\frac{\partial \overline{u_i}}{\partial x_j} + \frac{\partial \overline{u_j}}{\partial x_i}\right) - \frac{1}{\rho}\frac{\partial \overline{p}}{\partial x_i} - \frac{\partial \tau_{ij}}{\partial x_j}\,, \tag{2}$$

where the SGS stress is given by

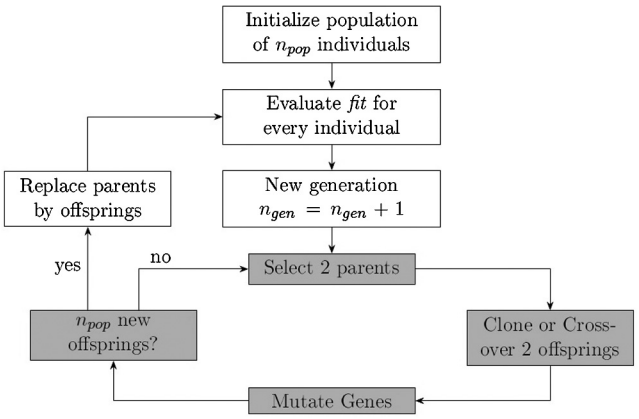

Fig. 2. General form of a genetic algorithm [19]. Note that here, 'Evaluate fit for every model' implies conducting an LES for each model and then evaluating the cost function based on the LES output.

Table 1
Parameters used for the genetic optimization.

Name	Value
Number of individuals	50
Generations	50
Rate of one-point recombination	0.2
Rate of two-point recombination	0.6
Mutation rate	0.2
Inversion rate	0.01
Random numerical constants: min; max; number of RNC	-0.1; 0.1; 5

$$\tau_{ij} = \overline{u_i u_j} - \overline{u_i}\,\overline{u_j}\,. \tag{3}$$

As usual u_i denotes the ith velocity component and p pressure. Without loss of generality viscosity ν and density ρ are assumed to be constant. It is assumed in this work that the subgrid-scale stress tensor can be expressed as a function depending only on the strain and rotation rate tensors, as well as the LES filter width Δ:

$$\tau_{ij}^{sgs} = \tau_{ij}^{sgs}(\overline{S}_{ij}, \overline{\Omega}_{ij}, \Delta) \tag{4}$$

with

$$\overline{S}_{ij} = \frac{1}{2}\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right) \quad \overline{\Omega}_{ij} = \frac{1}{2}\left(\frac{\partial \bar{u}_i}{\partial x_j} - \frac{\partial \bar{u}_j}{\partial x_i}\right). \tag{5}$$

Since typical symbolic regression algorithms do not involve restrictions with respect to units, $\overline{S}$ and $\overline{\Omega}$ have been non-dimensionalized by a suitable inverse time scale $|\overline{S}|$, yielding the non-dimensional matrices s and ω:

$$s_{ij} = \frac{\overline{S}_{ij}}{|\overline{S}|} \quad \omega_{ij} = \frac{\overline{\Omega}_{ij}}{|\overline{S}|} \quad |\overline{S}| = \sqrt{\bar{S}_{mn}\bar{S}_{mn}}. \tag{6}$$

It is important to note that, for ease of notation, the overbar is omitted for the normalised quantities s, ω. Any candidate model f_{ij} found by the GEP algorithm to approximate τ_{ij}^{sgs} has to be re-dimensionalized at the end of the process. For dimensional reasons this has been accomplished by:

$$\tau_{ij}^{mod} = -2\Delta^2|\overline{S}|^2 \cdot f_{ij}\left(s_{ij}, \omega_{ij}\right). \tag{7}$$

Making use of the Caley–Hamilton theorem as well as the work of Spencer et al. [27] and Pope [28] the unknown function can be written as a finite series of tensor basis functions T^{α} with scalar coefficients G_{α}. These coefficients are functions of invariants and are subject to optimization by the GEP algorithm:

$$f_{ij} = \sum_{\alpha=1}^{n} G_{\alpha} \cdot T_{ij}^{\alpha}; \quad G_{\alpha} = G_{\alpha}(I_1, I_2, \ldots I_m). \tag{8}$$

In order to reduce the computational cost of the search only the first four, at least quasi-quadratic, basis functions have been retained in the above series. Therefore, the function f_{ij} is spanned by the basis

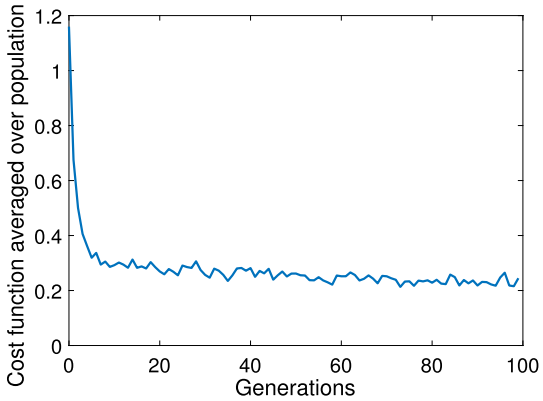

Fig. 3. Cost function averaged over population versus generation. Diverging models are discarded from the average.

$$
\begin{aligned}
t^1_{ij} &= s_{ij} \\
t^2_{ij} &= s_{ik}\omega_{kj} - \omega_{ik}s_{kj} \\
t^3_{ij} &= s_{ik}s_{kj} - s_{mk}s_{km}\frac{\delta_{ij}}{3} \\
t^4_{ij} &= \omega_{ik}\omega_{kj} - \omega_{mk}\omega_{km}\frac{\delta_{ij}}{3}
\end{aligned}
\tag{9}
$$

and the scalar coefficients G_α are functions of the four invariants:

$$
\begin{aligned}
I_1 &= s_{mn}s_{nm} \\
I_2 &= \omega_{mn}\omega_{nm} \\
I_3 &= s_{km}s_{mn}s_{nk} \\
I_4 &= \omega_{km}\omega_{mn}s_{nk}\,.
\end{aligned}
\tag{10}
$$

It is noted that the trace of s equals zero in incompressible flow and hence cannot be used as an invariant in the context of this work. Further it is worth remarking that the invariants I_1, I_2, I_3, I_4 are related to the physical mechanisms of dissipation of kinetic energy (I_1), enstrophy (I_2), generation of kinetic energy dissipation (I_3), and enstrophy production (I_4). Finally, the dimensional basis tensors T^α are given by

$$
T^1 = t^1 \cdot |\overline{S}|, \quad T^2 = t^2 \cdot |\overline{S}|^2, \quad T^3 = t^3 \cdot |\overline{S}|^2, \quad T^4 = t^4 \cdot |\overline{S}|^2\,. \tag{11}
$$

The quality of a candidate solution is expressed by a scalar value, called the fitness. In the context of this work the fitness has been calculated as the relative mean absolute error (RMAE) between a scalar output quantity φ of a candidate LES solution and the respective DNS value:

$$
RMAE(\varphi) = \frac{1}{T}\int_0^T \frac{|\varphi_{DNS}(t) - \varphi_{LES}(t)|}{|\varphi_{DNS}(t)|}dt\,. \tag{12}
$$

The function f_{ij} is optimized using an evolutionary analogy. For this purpose the function is represented as a string that can be recursively decoded into a non-linear function [29]. The process starts with an initial set of random strings. Subsequently a variety of operators analogous to reproduction and mutation are applied to the population of strings which modify the value of the strings. This in turn modifies the mathematical form (known as the phenotype). Many changes will result in a poorer representation of the true subgrid-scale stress tensor and in many cases the LES will even become unstable and diverge. Via a selection operator and the cost function, bad models (i.e. those where the LES diverges) are discarded such that only the better models in the population remain in the next generation. Using an analogy to Darwin's survival-of-the-fittest theory, the quality of the models evolves over many generations in exactly the same way as observed in natural systems. Fig. 3 shows an example of the evolution of the cost function averaged over the population for 100 generations. For an excellent and concise introduction on the GEP algorithm the reader is referred to the work of Ferreira [30]. For application of the methodology to fluid mechanics, and in particular RANS turbulence modeling, see the work of Weatheritt and Sandberg [4,10]. A-priori LES modeling using GEP is discussed in [17] and the methodology is outlined in detail in [19].

3. Results

In this section the results of the optimization process will be presented and the models suggested by GEP will be compared to standard models from literature. In the following subsection a sensitivity study on the model parameters will be presented. Finally, the robustness of the models with respect to filter size (i.e. grid size) and Reynolds number variations will be discussed.

3.1. Performance of trained models

A TGV has been selected as a demanding test case for LES model development. The LES turbulence kinetic energy $K = \overline{u}_i\overline{u}_i/2$ and its dissipation rate $\varepsilon = \partial K/\partial t$ are benchmarked against a reference DNS, performed with the same code, which provides consistent DNS data for evaluating the cost function given by $\frac{3}{5}RMAE(K) + \frac{2}{5}RMAE(\varepsilon)$. This choice of the cost function is to some extent arbitrary and provides a reasonable weight between the value of K and its derivative at the same time. Each model training has been conducted for 100 generations with a population size of 50. Due to the non-deterministic nature of the GEP tool, the training process (called creation in this framework) was performed 30 times, resulting in 30 different models. The precise number of a-posteriori model evaluations is unknown, because it depends on a number of random parameters, but is of the order of 100000. Out of the 30 creation processes performed, resulting in 30 best models, two models have been selected for further analysis: The model GEP1 due to its compact and elegant formulation,

$$\tau_{ij}^{GEP1} = -2\Delta^2\left\{-(I_3 + 0.04)\cdot T_{ij}^2\right\} \tag{13}$$

and the model GEP2 because it resulted in the smallest value of the cost function.

$$\tau_{ij}^{GEP2} = -2\Delta^2\left\{C_1\cdot|\overline{S}|\cdot T_{ij}^1 - C_2T_{ij}^2 + C_3T_{ij}^3 - C_4T_{ij}^4\right\} \tag{14}$$
$$C_1 = 0.01,\ C_2 = 0.146,\ C_3 = 0.01,\ C_4 = 0.11$$

The GEP1 model has a very simple structure consisting only of the tensor T^2 which, according to Horiuti [31], is responsible for a high correlation with the stresses and furthermore is relevant for the vortex stretching and generation of backward scatter of the SGS energy into the grid scale. The GEP2 model consists of an eddy viscosity type contribution (tensor T^1) and a structural model with contributions from T^2, T^3, T^4 and hence can be considered a mixed model. It is worth noting that the well-known gradient model due to Clark [32],

$$\tau_{ij}^{Clark} = \frac{\Delta^2}{12}\frac{\partial\overline{u}_i}{\partial x_k}\frac{\partial\overline{u}_j}{\partial x_k} \tag{15}$$

can be equivalently written as:

$$\tau_{ij}^{Clark} = \frac{\Delta^2}{12}(-T_{ij}^2 + T_{ij}^3 - T_{ij}^4). \tag{16}$$

Because of the pre-factor -2 in Eqn. (7) this corresponds to the coefficients $C_2 = C_3 = C_4 = -1/24 = -0.041\bar{6}$. In order to assess the performance of models GEP1 and GEP2 these models are not only compared to DNS data but also to four standard models from literature as well as a simulation without explicit model (no model LES). The well known Smagorinsky [33] model is given by

$$\tau_{ij}^{Smago} = -2\nu_t\overline{S}_{ij}, \quad \nu_t = (C_s\Delta)^2\sqrt{2\overline{S}_{ij}\overline{S}_{ij}}, \ \ C_S = 0.17 \tag{17}$$

and a more recent eddy viscosity model [34], the so called Sigma model, reads:

$$\tau_{ij}^{Sigma} = -2\nu_t\overline{S}_{ij} \quad \nu_t = (C_\sigma\Delta)^2\frac{\sigma_3(\sigma_1-\sigma_2)(\sigma_2-\sigma_3)}{\sigma_1^2},\ \sigma_1\geq\sigma_2\geq\sigma_3,\ \ C_\sigma = 1.35 \tag{18}$$

where σ_i are the square roots of the eigenvalues of the matrix $(\partial_j\bar{u}_i\cdot\partial_i\bar{u}_j)$. Finally a mixed model is considered which is given by the following two equivalent expressions:

$$\tau_{ij}^{Mixed} = (0.5^2)\tau_{ij}^{Smago} + \tau_{ij}^{Clark} = -2\Delta^2\left\{0.01\cdot|\overline{S}|\cdot T_{ij}^1 + \frac{1}{24}(T_{ij}^2 - T_{ij}^3 + T_{ij}^4)\right\} \tag{19}$$

In Eqn. (19) the modified Smagorinsky constant $C_S^* = \sqrt{0.01}/\sqrt[4]{2} \approx 0.084 \approx 0.5\cdot C_S$ has been used for two reasons: First of all the TGV case runs stable with a pure Clark model and there is no need for strong dissipative action potentially resulting from the theoretical value of $C_S = 0.17$. Secondly this choice of model parameter corresponds to the value C_1 appearing in the GEP2 model.

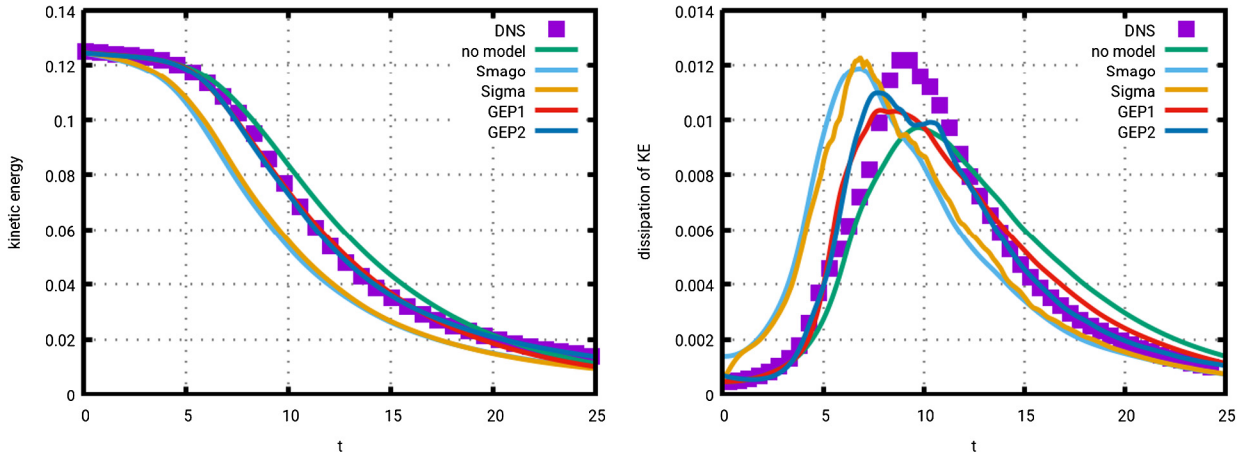

Fig. 4. Volume averaged mean kinetic energy and its dissipation rate versus non-dimensional time for the reference DNS and LES using no model, the Smagorinsky model, the Sigma model as well as the GEP1 and GEP2 models.

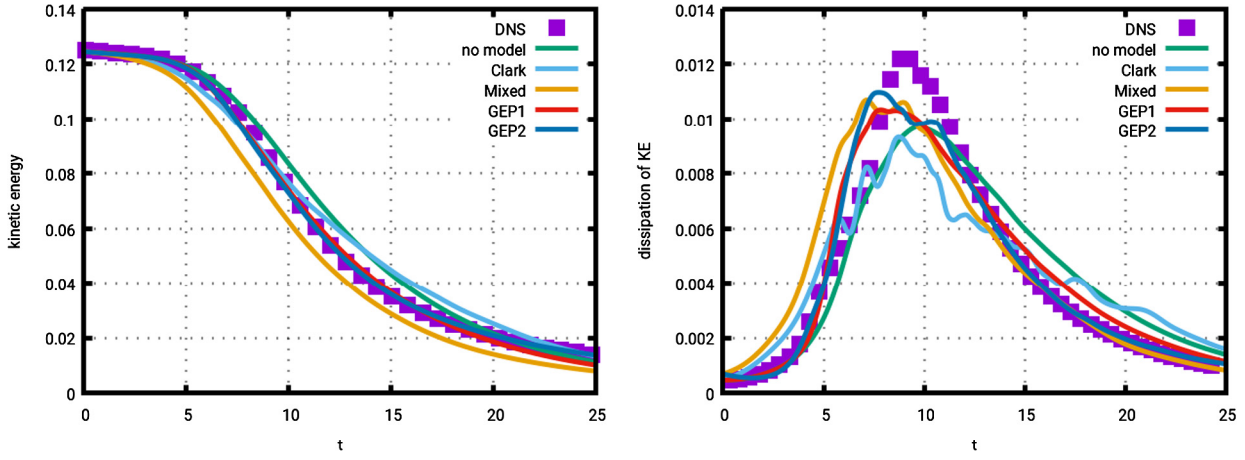

Fig. 5. Volume averaged mean kinetic energy and its dissipation rate versus non-dimensional time for the reference DNS and LES using no model, the Clark model and a mixed model as well as the GEP1 and GEP2 models.

Table 2
$RMAE(K)$ for the reference models as well as GEP1 and GEP2.

Model	$RMAE(K) \times 100$
No Model	9.67
Clark	11.78
Smago	21.45
Sigma	20.54
Mixed	18.64
GEP1	5.59
GEP2	1.51

Fig. 4 shows the volume averaged kinetic energy and its dissipation (note that, more precisely, here and in the subsequent figures the magnitude of dissipation rate is shown) for the reference DNS, a no model LES, the well-known Smagorinsky and Sigma models as well as GEP1 and GEP2. Fig. 4 shows that the no model run does not provide enough dissipation at non-dimensional time roughly $t = 10$ when the flow is in a stage between vortex breakdown and transition to turbulence (see Fig. 1). By contrast the Smagorinsky and Sigma models provide too much dissipation too early and consequently under-predict kinetic energy. Finally, the two models suggested by the GEP algorithm provide, considering the very coarse mesh, a good simulation accuracy and seem to capture the transition mechanism reasonably well. Fig. 5 compares the two GEP models with two structural models, i.e. the Clark model and the mixed model given by Eqn. (19). While the performance of these two structural models is better than the one from the eddy viscosity models they are clearly worse than the two GEP optimized models. Table 2 shows the $RMAE(K)$ for all models considered so far which clearly indicates the good performance of the GEP models.

Table 3
List of GEP2 model variants characterised by the changes in coefficients (indicated with bold font) and relative mean absolute error $RMAE(K)$. The baseline GEP2 coefficients are shown in the first (data) line.

Mod	C_1	C_2	C_3	C_4	$RMAE(K) \times 100$
Baseline	0.01	0.146	0.01	0.11	1.51
1	0.01	**0.11**	0.01	0.11	3.00
2	0.01	0.146	**0.0**	0.11	1.59
3	0.01	**0.11**	**0.11**	0.11	∞
4	0.01	**0.0**	**0.0**	**0.0**	15.97
5	**0.0**	0.146	0.01	0.11	33.07

3.2. Discussion of optimal model

One big advantage of the optimization by symbolic regression is that the model inferred is a mathematical expression (rather than a black box), that can be interpreted, analysed and modified. For example it can happen that GEP selects a model with a term T^α that has negligible contribution because the scalar function / coefficient G_α is very small. In another hypothetical scenario GEP might provide constant scalar coefficients which are of similar magnitude leading to the question as to whether this truly reflects the optimal functional form or it is because the optimization was stopped too early. Table 3 shows a list of parameter variations together with the model performance $RMAE(K)$.

Modification 1 assumes equal coefficients C_2 and C_4 which has been suggested by GEP as the optimal model for other creations. The $RMAE$ indicates that the performance is very similar, but slightly worse than the GEP2 model. It should be mentioned that the curves of K for modification 1 are more or less within the line thickness of the GEP2 results and thus they are not shown explicitly. Similarly for modification 2 the already small coefficient C_3 has been set to zero and the results demonstrate that this term has a small contribution and does not play a major role. In the case when the value of C_3 is increased and C_2 is slightly decreased such that $C_2 = C_3 = C_4 = 0.11$, the simulation diverges (indicated by an $RMAE$ of infinity, see modification 3). The combined action of terms $-T^2 + T^3 - T^4$ with the same scalar coefficient 0.11 corresponds to the Clark model multiplied with the factor -2.64. Due to the negative sign and large magnitude of model parameter it is not surprising that the model is unstable. Setting $C_2 = C_3 = C_4 = 0$ (modification 4) corresponds to a static Smagorinsky model with model coefficient $C_S^* = \sqrt{0.01}/\sqrt[4]{2} \approx 0.084$ (note the definition of $|\bar{S}|$ in Eqn. (6)). The results are considerably better than the results with the standard Smagorinsky model parameter because the transition process is captured considerably more closely (due to a smaller amount of dissipation). Nevertheless, they are a factor of 10 worse than those obtained by the GEP2 model. Finally, setting the eddy viscosity contribution to zero (modification 5) provides a stable simulation but with considerably overpredicted K which is reflected in the large MAE of this LES run. The foregoing discussion suggests that the important ingredients for the success of the GEP2 model are the eddy viscosity contribution T^1 together with the second order terms T^2 and T^4 with roughly equal, constant scalar coefficients.

3.3. Robustness of trained models

As mentioned before, roughly 100000 LES runs have been performed by the modelling framework to evaluate candidate models suggested by the GEP code. This was only possible by reducing the cost of an individual LES to a reasonable amount of CPU hours (roughly 0.75 CPU core hours, whereas the wall clock time for the GEP algorithm itself is minute in comparison to testing one model candidate) and by limiting the training to one single scenario. Nevertheless, there is no guarantee that the good performance of the models GEP1 and GEP2 can be maintained in scenarios different from the training configuration. Therefore, to assess the generalizability of the developed models described above, in this subsection additional LES runs of the models GEP1 and GEP2 have been performed on grids with resolutions 64^3 and 128^3, rather than the very coarse 32^3 used for the training. Furthermore, using the model training resolution of 32^3, the models have been tested for different Reynolds numbers of $Re = 400$ and $Re = 3000$ instead of $Re = 1600$. It can be seen from Fig. 6 that the performance of the GEP1 model deteriorates considerably when the LES is performed on finer grids. This shows that the good performance for the training configuration cannot be maintained on finer grids. In fact the LES results become even considerably worse with mesh refinement. In contrast the GEP2 model properly converges towards the DNS solution when the grid is refined and shows a very satisfactory performance (see Fig. 7).

Next, the Reynolds number has been changed to a lower and higher value, respectively. The results indicate a similar behaviour as for the variation of the filter size: Whereas the GEP1 model fails for the lower $Re = 400$ and the higher $Re = 3000$ Reynolds numbers on the standard 32^3 grid (see Fig. 8), the GEP2 model provides very satisfactory results for all Reynolds numbers (see Fig. 9). Thus overall, model GEP2 shows good robustness to varying grid sizes and Reynolds numbers, while model GEP1 fails to generalize. This indicates that model GEP2 better captures the salient physics of the problem by also including basis functions $T^1, (T^3)$ and in particular T^4. Nevertheless, the analysis in this subsection indicates that training of the model for one filter size and one Reynolds number might not be sufficient to ensure the robustness and the quality of the model for different scenarios. Therefore, ideally such variations should be included in the training phase in future work.

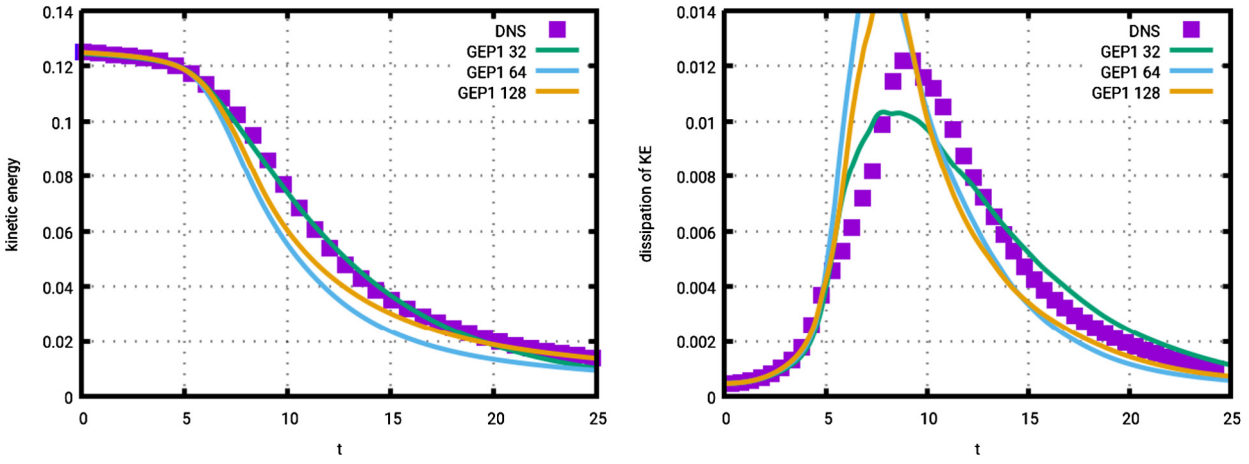

Fig. 6. Volume averaged mean kinetic energy and its dissipation rate versus non-dimensional time for the reference DNS and LES using the GEP1 model for three different grid resolutions.

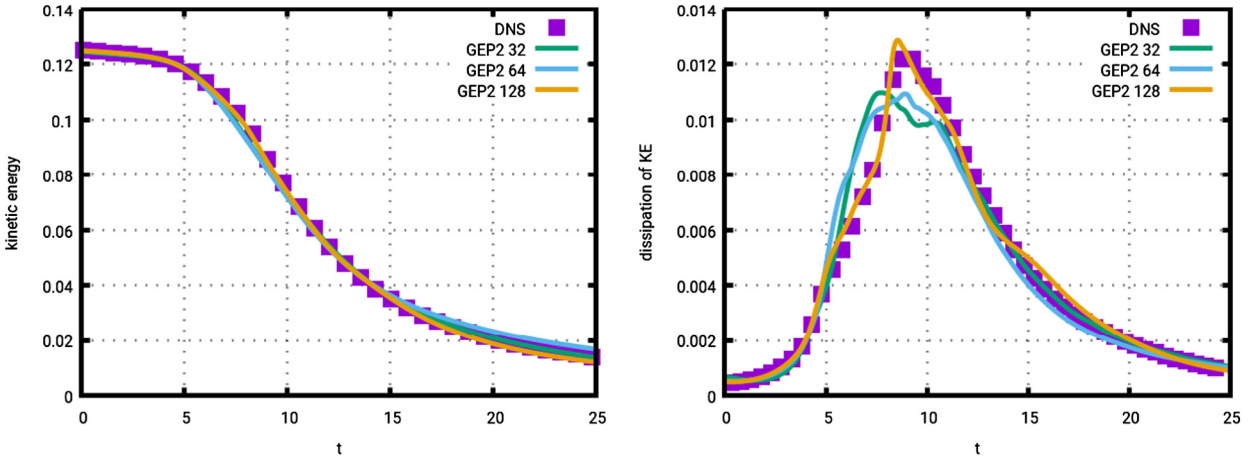

Fig. 7. Volume averaged mean kinetic energy and its dissipation rate versus non-dimensional time for the reference DNS and LES using the GEP2 model for three different grid resolutions.

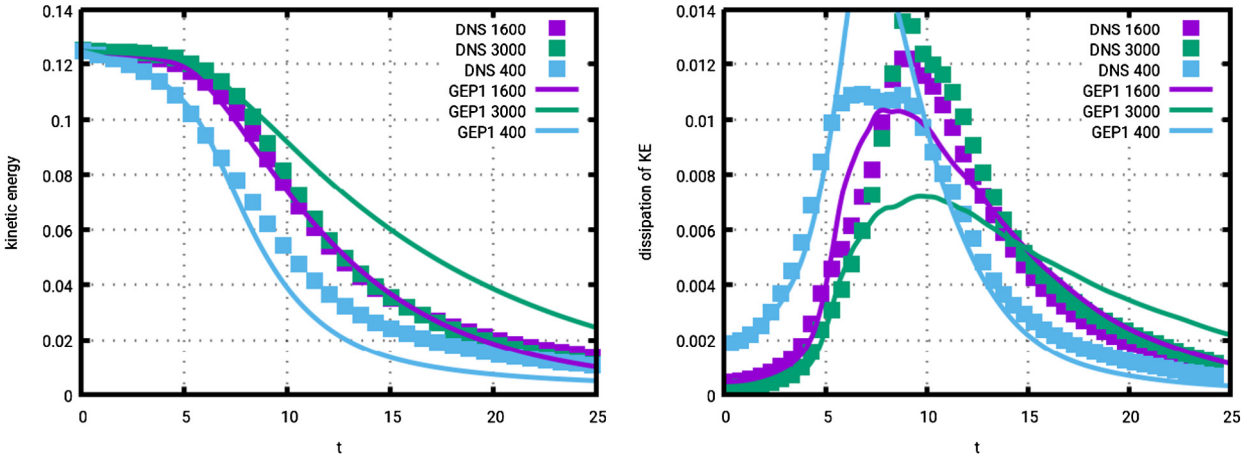

Fig. 8. Volume averaged mean kinetic energy and its dissipation rate versus non-dimensional time for the reference DNS and LES using the GEP1 model for three different Reynolds numbers.

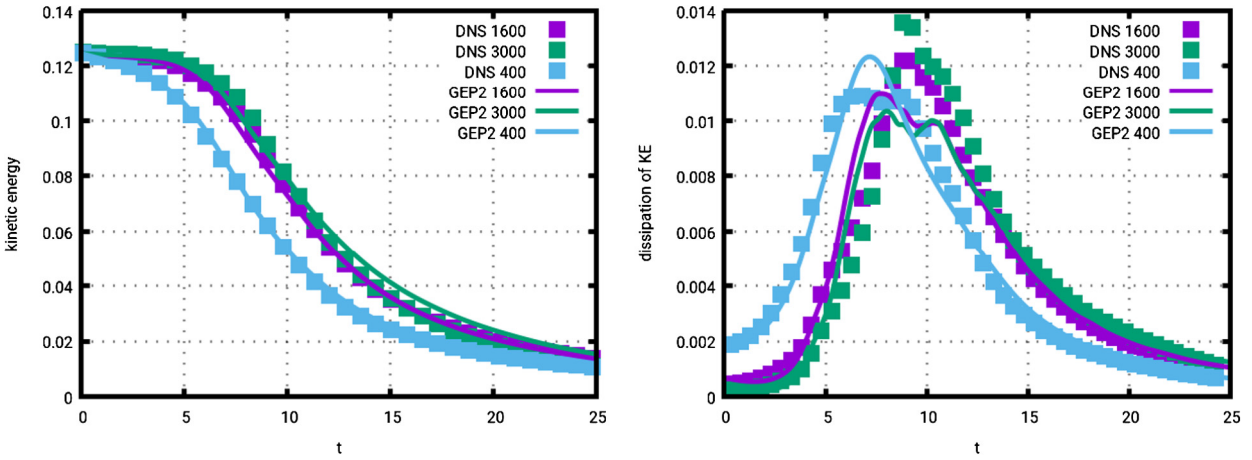

Fig. 9. Volume averaged mean kinetic energy and its dissipation rate versus non-dimensional time for the reference DNS and LES using the GEP2 model for three different Reynolds numbers.

4. Conclusions

The concept of Gene Expression Programming was applied to model the unknown subgrid stresses in the context of Large Eddy Simulation of a Taylor Green vortex. To the best knowledge of the authors this is the first application of GEP to in-the-loop, a-posteriori LES model development. A DNS of the same configuration has been performed, using the same code, to provide the reference data such that the cost function could be chosen as the weighted mean absolute error between DNS and LES of kinetic energy and its dissipation. About 100000 LES runs have been conducted resulting in 30 LES models as a result of 30 different runs of the stochastic optimization process. Out of these 30 models two have been selected for a detailed analysis. These two models GEP1 and GEP2 have been compared to standard LES models from literature as well as a no model simulation. For the training case they clearly outperform these existing models. When applied to different grid resolutions and different Reynolds numbers GEP1 did not perform satisfactorily while GEP2 proved to be very robust to these parameter variations. One particularly attractive feature of the proposed methodology is that it combines traditional modeling strategies, like the Caley–Hamilton theorem, with machine learning and that the result of the process is not a black box model, but a well defined mathematical expression. It has been demonstrated that this can be used to understand the important ingredients of the model and to possibly fine tune the model manually. Finally, it is important to stress that the focus of this work was not to put forward a particular new LES model but to demonstrate that evolutionary algorithms can successfully be used for in-the-loop a-posteriori optimization of LES models. Future work includes the application of GEP to other unclosed terms with additional complexity, like multiple phases or chemical reaction, using potentially different basis functions.

CRediT authorship contribution statement

Maximilian Reissmann: Data curation, Investigation, Software. **Josef Hasslberger:** Conceptualization, Formal analysis, Supervision, Validation, Writing - review & editing. **Richard D. Sandberg:** Methodology, Software, Supervision, Writing - review & editing. **Markus Klein:** Conceptualization, Formal analysis, Funding acquisition, Supervision, Writing - original draft, Writing - review & editing.

Declaration of competing interest

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

Acknowledgements

The visit to University of Melbourne of the first author was supported by the German Federal Ministry of Defence.

References

[1] K. Duraisamy, G. Iaccarino, H. Xiao, Turbulence modeling in the age of data, Annu. Rev. Fluid Mech. 51 (2019) 357–377.
[2] S. Wallin, A.V. Johansson, An explicit algebraic Reynolds stress model for incompressible and compressible turbulent flows, J. Fluid Mech. 403 (2000) 89–132.
[3] B.A. Younis, C.G. Speziale, T.T. Clark, A rational model for the turbulent scalar fluxes, Proc. R. Soc. Lond. A, Math. Phys. Eng. Sci. 461 (2005) 575–594.

[4] J. Weatheritt, R. Sandberg, A novel evolutionary algorithm applied to algebraic modifications of the RANS stress strain relationship, J. Comput. Phys. 325 (2016) 22–37.
[5] J. Ling, A. Kurzawski, J. Templeton, Reynolds averaged turbulence modelling using deep neural networks with embedded invariance, J. Fluid Mech. 807 (2016) 155–166.
[6] E.J. Parish, K. Duraisamy, A paradigm for data-driven predictive modeling using field inversion and machine learning, J. Comput. Phys. 305 (2016) 758–774.
[7] J. Wu, H. Xiao, E. Paterson, Physics-informed machine learning approach for augmenting turbulence models: a comprehensive framework, Phys. Rev. Fluids 3 (2018) 074602.
[8] P.M. Milani, J. Ling, G. Saez-Mischlich, J. Bodart, J.K. Eaton, A machine learning approach for determining the turbulent diffusivity in film cooling flows, in: ASME Turbo Expo 2017: Turbomachinery Technical Conference and Exposition, American Society of Mechanical Engineers, 2017, V05AT12A002.
[9] J. Weatheritt, Y. Zhao, R.D. Sandberg, S. Mizukami, K. Tanimoto, Data-driven scalar-flux model development with application to jet in cross flow, Int. J. Heat Mass Transf. 147 (2020) 118931.
[10] J. Weatheritt, R.D. Sandberg, Hybrid Reynolds-averaged/large-eddy simulation methodology from symbolic regression: formulation and application, AIAA J. (2017).
[11] A. Vollant, G. Balarac, C. Corre, Subgrid-scale scalar flux modelling based on optimal estimation theory and machine-learning procedures, J. Turbul. 18 (2017) 854–878.
[12] M. Gamahara, Y. Hattori, Searching for turbulence models by artificial neural network, Phys. Rev. Fluids 2 (2017) 054604.
[13] C.J. Lapeyre, A. Misdariis, N. Cazard, D. Veynante, T. Poinsot, Training convolutional neural networks to estimate turbulent sub-grid scale reaction rates, Combust. Flame 203 (2019) 255–264.
[14] A. Seltz, P. Domingo, L. Vervisch, Z.M. Nikolaou, Direct mapping from les resolved scales to filtered-flame generated manifolds using convolutional neural networks, Combust. Flame 210 (2019) 71–82.
[15] S. Ketterl, M. Reissmann, M. Klein, Large eddy simulation of multiphase flows using the volume of fluid method: Part 2—A-posteriori analysis of liquid jet atomization, Exp. Comput. Multiph. Flow 1 (2019) 201–211.
[16] X.I.A. Yang, S. Zafar, J.-X. Wang, H. Xiao, Predictive large-eddy-simulation wall modeling via physics-informed neural networks, Phys. Rev. Fluids 4 (2019) 034602.
[17] M. Schoepplein, J. Weatheritt, R. Sandberg, J. Mohsen, M. Klein, Application of an evolutionary algorithm to LES modelling of turbulent transport in premixed flames, J. Comput. Phys. 374 (2018) 1166–1179.
[18] H. Vaddireddy, A. Rasheed, A. Staples, O. San, Feature engineering and symbolic regression methods for detecting hidden physics from sparse sensor observation data, Phys. Fluids 32 (2020) 015113.
[19] M. Schoepplein, J. Weatheritt, M. Talei, M. Klein, R. Sandberg, Application of an evolutionary algorithm to LES modeling of turbulent premixed flames, in: H. Pitsch, A. Attili (Eds.), Data Analysis for Direct Numerical Simulations of Turbulent Combustion. From Equation-Based Analysis to Machine Learning, Springer, 2020, in press.
[20] C. Meneveau, Statistics of turbulence subgrid-scale stresses: necessary conditions and experimental tests, Phys. Fluids 6 (1994) 815–833.
[21] E. Bou-Zeid, Challenging the large eddy simulation technique with advanced a posteriori tests, J. Fluid Mech. 764 (2015) 1–4.
[22] S. Hickel, N. Adams, J. Domaradzki, An adaptive local deconvolution method for implicit LES, J. Comput. Phys. 213 (2006) 413–436.
[23] M. Bracheta, D. Meiron, S. Orszag, B. Nickel, R. Morf, U. Frisch, Small-scale structure of the Taylor–green vortex, J. Fluid Mech. 130 (1983) 411–452.
[24] Y. Ling, S. Zaleski, R. Scardovelli, Multiscale simulation of atomization with small droplets represented by a Lagrangian point-particle model, Int. J. Multiph. Flow 76 (2015) 122–143.
[25] K.A.D. Jong, An analysis of the behavior of a class of genetic adaptive systems, PhD thesis, Ann Arbor, MI, USA, 1975, AAI7609381.
[26] C. Ferreira, Mutation, transposition, and recombination: an analysis of the evolutionary dynamics, in: Joint Conference on Information Science, JCIS, 2002, pp. 614–617.
[27] A.J.M. Spencer, R.S. Rivlin, The theory of matrix polynomials and its application to the mechanics of isotropic continua, in: Collected Papers of R.S. Rivlin, Springer, 1997.
[28] S.B. Pope, A more general effective-viscosity hypothesis, J. Fluid Mech. 75 (1975) 331–340.
[29] J.R. Koza, Genetic programming as a means for programming computers by natural selection, Stat. Comput. 4 (1994) 87–112.
[30] C. Ferreira, Gene expression programming: a new adaptive algorithm for solving problems, Complex Syst. 13 (2001) 87–129.
[31] K. Horiuti, Alignment of eigenvectors for strain rate and subgrid-scale stress tensors, in: Direct and Large-Eddy Simulation IV, in: ERCOFTAC Series, vol. 8, 2001.
[32] R.A. Clark, J.H. Ferziger, W. Reynolds, Evaluation of subgrid-scale models using an accurately simulated turbulent flow, J. Fluid Mech. 91 (1979) 1–16.
[33] J. Smagorinsky, General circulation experiments with the primitive equations, Mon. Weather Rev. 91 (1963) 99–164.
[34] F. Nicoud, H. Toda, O. Cabrit, S. Bose, J. Lee, Using singular values to build a subgrid-scale model for large eddy simulations, Phys. Fluids 23 (2011) 085106.