Data-driven turbulence modelling using symbolic regression

Article *in* Journal of Physics Conference Series · November 2021

DOI: 10.1088/1742-6596/2099/1/012020

2 authors, including:

Sergey N. Yakovenko
Khristianovich Institute of Theoretical and Applied Mechanics
87 PUBLICATIONS **199** CITATIONS

Data-driven turbulence modelling using symbolic regression

Data-driven turbulence modelling using symbolic regression

A Chakrabarty[1] **and S N Yakovenko**[2]

[1] Novosibirsk State University, Pirogova str., 1, Novosibirsk, 630090, Russia
[2] Khristianovich Institute of Theoretical and Applied Mechanics, Siberian Branch of Russian Academy of Sciences, Institutskaya str., 4/1, Novosibirsk, 630090, Russia

E-mail: s.yakovenko@mail.ru

Abstract. The study is focused on the performance of machine-learning methods applied to improve the velocity field predictions in canonical turbulent flows by the Reynolds-averaged Navier–Stokes (RANS) equation models. A key issue here is to approximate the unknown term of the Reynolds stress (RS) tensor needed to close the RANS equations. A turbulent channel flow with the curved backward-facing step on the bottom has the high-fidelity LES data set. It is chosen as the test case to examine possibilities of GEP (gene expression programming) of formulating the enhanced RANS approximations. Such a symbolic regression technique allows us to get the new explicit expressions for the RS anisotropy tensor. Results obtained by the new model produced using GEP are compared with those from the LES data (serving as the target benchmark solution during the machine-learning algorithm training) and from the conventional RANS model with the linear gradient Boussinesq hypothesis for the Reynolds stress tensor.

1. Introduction

The development of machine learning and artificial intelligence techniques has opened up avenues in computational fluid dynamics (CFD) that were not well explored before specifically in the area of data-driven turbulence modelling [1]. Solving the Navier–Stokes equations for turbulent flows still remains one of the biggest challenges faced in CFD. With the use of direct numerical simulation (DNS), very accurate results can be obtained due to the fact that no turbulence model is needed but it is not a preferred solution since DNS is very computationally expensive scaling as Re^3 where Re is a typical Reynolds number in a flow. In large eddy simulation (LES), turbulent motions are modelled on small scales, which reduces the computational complexity in comparison with DNS. However, the LES application is still challenging, especially for high-Reynolds-number separated flows with thin near-wall boundary layers [2]. The more popular way to solve the Navier–Stokes equations is to analyze their Reynolds-averaged form [3-6]. Solving the Reynolds averaged Navier–Stokes (RANS) equations is widely used in engineering applications because of its relatively low computational costs compared to other methods. RANS simulations do not resolve turbulent fluctuations explicitly, but employ empirical Reynolds-stress approximations, which is computationally efficient. The RANS approximations are often inaccurate, leading to errors and uncertainties that should be quantified.

There is a significant possibility to improve RANS models with the use of machine learning (ML) algorithms [1] which have been first introduced in turbulence modelling [7, 8] to create markers detecting the regions of high RANS model uncertainty and to model the Reynolds stress anisotropy (RSA) tensor. While the progress has been made to apply ML tools to predict some properties of flows in a data-driven approach, e.g. using the high-fidelity (LES, DNS, measurement) data for training, wider applications of ML models still remain vastly unexplored. With the increase in computational

capabilities, high-fidelity data for larger Reynolds numbers become more frequent aiding in the development of data-driven models of turbulence [9-16] using different machine learning methods like neural networks, random forests, sparse regression, evolutionary algorithms, etc.

The present study aims to improve predictions of the RSA tensor $b_{ij} \equiv [\tau_{ij}/(2k) - (1/3)\delta_{ij}]$ (where the velocity fluctuation correlation $\tau_{ij} \equiv \langle u_i' u_j' \rangle$ is the Reynolds stress tensor, its trace $k \equiv 0.5\tau_{ii} \equiv 0.5\langle u_i' u_i' \rangle$ is the turbulent kinetic energy) using the ML technique to decrease the discrepancy in comparison with high fidelity data for canonical turbulent wall-bounded flow cases as in [8-16]. The workability of gene expression programming (GEP) which belongs to symbolic regression is explored here for a channel flow with the curved backward-facing step having the high-fidelity LES data set [17]. This is a new test case in comparison with those in [10] where the novel GEP algorithm was proposed, trained and tested for channel flows with the rectangular backstep and periodic hills placed on the bottom.

2. Turbulence model formulation

To describe the flow behaviour in terms of velocity and pressure fields, the momentum and continuity equations for incompressible fluid are considered here in the Reynolds (time or ensemble) averaged form leading to the widely used RANS equations where the unknown term of the Reynolds stress (RS) tensor is to be modelled for closing the equations. The Boussinesq hypothesis (gradient expression)

$$\tau_{ij} = (2/3)k\delta_{ij} - 2\nu_t S_{ij}, \quad \nu_t = C_\mu k^2/\varepsilon = C_\mu k/\omega, \quad C_\mu = 0.09, \quad S_{ij} = (1/2)\cdot(\partial U_i/\partial x_j + \partial U_j/\partial x_i)$$

is usually applied for the RS tensor based on the isotropic eddy-viscosity coefficient ν_t and related linearly to the strain tensor S_{ij} which is defined by the spatial derivatives of the mean velocity vector components U_i. Quantities of k and its viscous dissipation rate ε or ω are found from extra transport equtions formulated within the various versions of k-ε, k-ω and other turbulence models. The linear Boussinesq relation between the RS and strain tensors can be rewritten in terms of the RSA tensor as

$$b_{ij} = -(\nu_t/k)\, S_{ij} = -C_\mu s_{ij} \tag{1}$$

where the non-dimensional strain tensor $s_{ij} = (k/\varepsilon)\cdot S_{ij}$ can be introduced as in [8, 18] to develop the algebraic relation of the non-dimensional RSA tensor with the mean velocity derivatives.

As mentioned in Introduction, the conventional model (1) leads to the considerable errors and uncertainties, especially in separated turbulent flows bounded by complex-geometry surfaces. The most attempts to enhance the RANS approximations modify the expression for b_{ij} following to [8] where the generalized eddy viscosity model for incompressible flow is presented as function of the non-dimensionalized strain (s_{ij}) and rotation (r_{ij}) tensors decomposed into ten terms with the isotropic basis tensors (where coefficients depend on the scalar functions $\lambda_1, \ldots, \lambda_5$ being the s_{ij}, r_{ij} invariants):

$$b_{ij} = \sum g^{(n)}(\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5)\, T_{(n),ij} = g^{(1)} T_{(1),ij} + g^{(2)} T_{(2),ij} + \ldots + g^{(10)} T_{(10),ij} \tag{2}$$

$$T_{(1),ij} = s_{ij}, \quad T_{(2),ij} = s_{ik} r_{kj} - r_{ik} s_{kj}, \quad T_{(3),ij} = s_{ik} s_{kj} - (1/3)\cdot\delta_{ij} s_{km} s_{mk}, \quad T_{(4),ij} = r_{ik} r_{kj} - (1/3)\cdot\delta_{ij} r_{km} r_{mk}$$

$$T_{(5),ij} = r_{ik} s_{km} s_{mj} - s_{ik} s_{km} r_{mj}, \quad T_{(6),ij} = r_{ik} r_{km} s_{mj} + s_{ik} r_{km} r_{mj} - (2/3)\cdot\delta_{ij} s_{kl} r_{lm} r_{mk}$$

$$T_{(7),ij} = r_{ik} s_{km} r_{mn} r_{nj} - r_{ik} r_{km} s_{mn} r_{nj}, \quad T_{(8),ij} = s_{ik} r_{km} s_{mn} s_{nj} - s_{ik} s_{km} r_{mn} s_{nj}$$

$$T_{(9),ij} = r_{ik} r_{km} s_{mn} s_{nj} + s_{ik} s_{km} r_{mn} r_{nj} - (2/3)\cdot\delta_{ij} s_{kl} s_{lm} r_{mn} r_{nk}, \quad T_{(10),ij} = r_{ik} s_{kl} s_{lm} r_{mn} r_{nj} - r_{ik} r_{kl} s_{lm} s_{mn} r_{nj}$$

$$\lambda_1 = s_{ij} s_{ji}, \quad \lambda_2 = r_{ij} r_{ji}, \quad \lambda_3 = s_{ij} s_{jk} s_{ki}, \quad \lambda_4 = r_{ij} r_{jk} s_{ki}, \quad \lambda_5 = r_{ij} r_{jk} s_{km} s_{mi}$$

$$R_{ij} = (1/2)\cdot(\partial U_i/\partial x_j - \partial U_j/\partial x_i), \quad r_{ij} = (k/\varepsilon)\cdot R_{ij}$$

The ML tool is directed to find the optimal scalar coefficients $g^{(n)}(\lambda_1, \ldots, \lambda_5)$. Once the coefficients are found during the training of a model with the high-fidelity data, serving as a target solution and taken from available datasets, then (2) can be used to obtain the new RSA tensor b_{ij} and can be inserted into the RANS solver to improve its performance for new flow cases in comparison with that for (1).

3. Methodology

As a machine learning algorithm, gene expression programming (GEP) which belongs to symbolic regression is used here as in [10, 11]. GEP has been first introduced in [19] and has since attracted research communities to further develop and enhance such a method for new fields. It is a type of evolutionary algorithms closely related to genetic algorithms (GA) and genetic programming (GP). GEP is a special field of evolutionary computation that builds models and programs automatically to solve problems in any domain without depending on or making assumptions about the domain. The fundamental difference between the GA, GP, GEP algorithms resides in the nature of the individuals which are linear strings of fixed length (chromosomes) in GA, or nonlinear entities of different sizes and shapes (parse trees) in GP. On the other hand, in GEP the individuals are encoded as linear strings of fixed length (the genome or chromosomes) which are afterwards expressed as nonlinear entities of different sizes and shapes (i.e., simple diagram representations or expression trees).

The GEP process begins with the random generation of the chromosomes of the initial population. Then the chromosomes are expressed, and the fitness of each individual is evaluated. The individuals are then selected according to the fitness to reproduce with modification. The reproduction with modification is to ensure a creation of individuals with the necessary genetic diversity that facilitates evolution in each generation. This reproduction includes replication, mutation and transposition which imitate the real-life behaviour of genes. In other words, the modifications are in fact the modifications of functions that are reproduced every generation. Diversification in such functions helps increase the search space to find the coefficient of best fit and make sure the optimisation does not get stuck in local maxima or minima.

The implementation for GEP in the RANS turbulence modelling is as follows:

- define the Reynolds stress anisotropy tensor b_{ij} from the high-fidelity DNS or LES dataset;
- calculate scalars λ_m and tensors $\mathbf{T}^{(n)}$ from values of k, ε, $\partial U_i/\partial x_j$ of the high-fidelity data set;
- start GEP based on these λ_m, $\mathbf{T}^{(n)}$ and using *a priori* form of $b_{ij} = b_{ij}$ ($\mathbf{T}^{(1)}$, $\mathbf{T}^{(2)}$, … , λ_1, λ_2, …);
- receive optimal algebraic equations from GEP as a combination of some or all input functions;
- test the new algebraic model for b_{ij} in different test cases to see its predictability.

Advantages of such an approach are as follows [10]:

- no particular model needs to be assumed;
- little human bias in selecting functions is applied as functions are chosen randomly;
- low computation cost as long as search space is restricted to relevant functions.

In the present study, the GEP tool is explored with fixed length chromosomes to find the b_{ij} values. For the GEP implementation, the geppy library (https://geppy.readthedocs.io/en/latest) is used where an evolutionary algorithm framework designed for GEP in Python is built on top of the more general evolutionary computation framework DEAP. The latter does not completely support GEP by itself.

3.1. Test case

First, a turbulent flow with the curved backward-facing step (CBFS) on the channel bottom which has the extensive data set of the high-fidelity computations [17] is chosen as the test case (figure 1). The tensor basis is initially calculated from (2), taking all values obtained from the available data of LES (https://turbmodels.larc.nasa.gov/Other_LES_Data/curvedstep.html), and using only four tensors $\mathbf{T}^{(1)}$, $\mathbf{T}^{(2)}$, $\mathbf{T}^{(3)}$, $\mathbf{T}^{(4)}$ and two scalar invariants λ_1, λ_2. Taking more tensors causes only a negligible increase in calculation accuracy while increasing computation time considerably [10]. In the two-dimensional consideration, each input tensor $\mathbf{T}$ contains four components (T_{11}, T_{12}, T_{21}, T_{22}) flattened as in [13] to give a single vector as the input into the GEP algorithm. With the input features ready, the GEP algorithm is trained on the anisotropy tensor b_{ij} to receive the new equations for b_{ij}. Next, these equations can be inserted into the RANS solver and verified in the *a-posteriori* study for the same flow at different Reynolds numbers or for a new flow case of similar geometry.

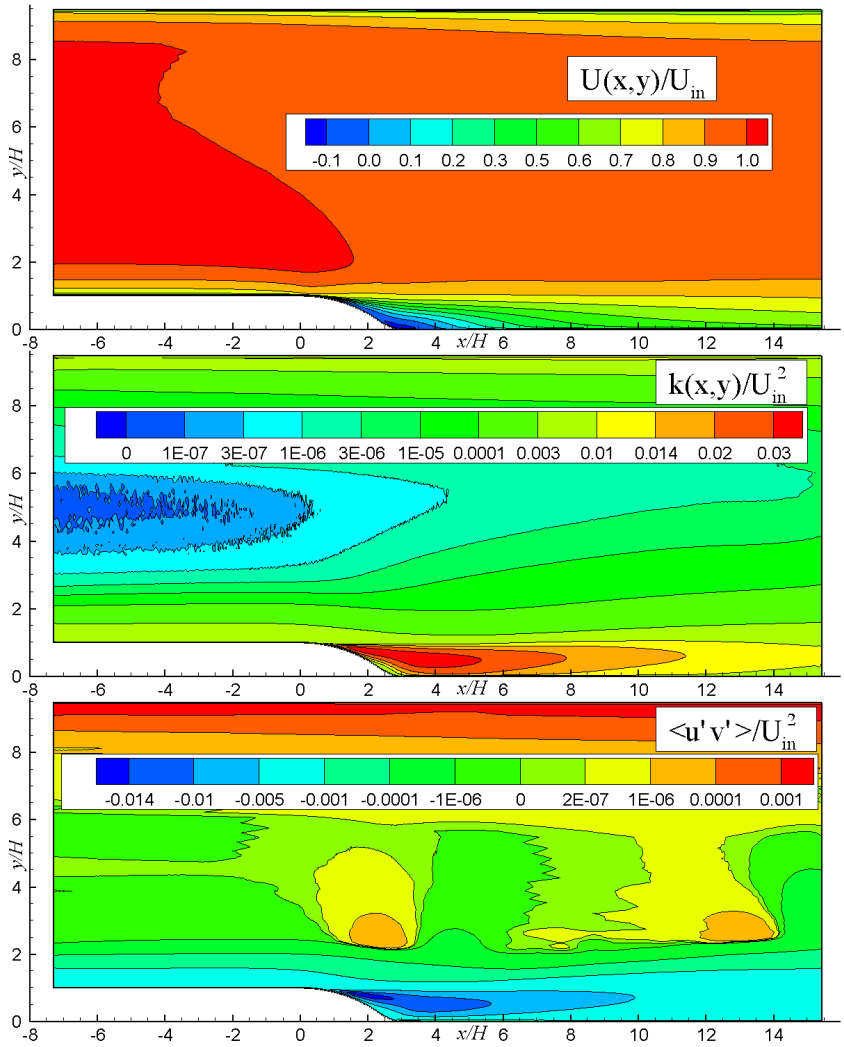

Figure 1. The LES data [17] at Reynolds number Re = $U_{in}H/\nu$ = 13 700 presented by contours of non-dimensional horizontal mean velocity vector component $U(x,y)/U_{in}$, the turbulent kinetic energy $k(x,y)/(U_{in})^2$, the shear Reynolds-stress tensor component $\langle u'v'\rangle/(U_{in})^2$, where U_{in} is the inlet free-stream velocity, H is the curved step height.

3.2. Computation details

During running the GEP algorithm, the number of data points were kept in the feasible range from 2500 to 10000 to have an optimized solution along with optimal training time. The latter increases drastically with increase in the number of points so a training dataset region is initially reduced to the area (in the recirculation zone downstream the step near the lower surface) where there is the highest discrepancy between the high- and lower-fidelity data of LES and RANS simulations. The population size and number of generations are the criteria that ensure that there is sufficient search space and enough number of iterations respectively for the algorithm to converge to an optimum solution. The population size was kept between 500 and 1000, and the better-optimized equations were obtained with population sizes near the higher threshold. The equations converge from 50-70 generations after which the drop in loss is not observed to be large enough when compared to time efficiency.

4. Results

The final convergent equation for the RSA tensor obtained after 20 runs with 50 iterations (i.e. 50 generations) of GEP for the reduced training region ($2.0 < x/H < 3.2$, $0 < y/H < 0.45$) in CBFS flow is:

$$\mathbf{b} = -0.027\mathbf{T}^{(1)} - 0.009\mathbf{T}^{(2)} + 0.009(\lambda_1 + \lambda_2)\mathbf{T}^{(3)} - 0.036\mathbf{T}^{(4)} \quad (3)$$

Next, we once again trained the GEP algorithm with 50 iterations and 20 runs for a larger CBFS flow area ($-5.0 < x/H < 10.0$, $0 < y/H < 1.25$) and have the modified explicit relation for the RSA tensor:

$$\mathbf{b} = -(0.007 + 0.003\lambda_1 + 0.003\lambda_2)\mathbf{T}^{(1)} - 0.003\mathbf{T}^{(2)} + 0.007\mathbf{T}^{(3)} \quad (4)$$

Due to GEP being a symbolic regression algorithm, a few resulting equations obtained after 20 runs of this algorithm contain a constant term (without **T**) in the right-hand side which makes the equation unphysical. These results had to be discarded. We also had to carefully select the equations (and then perform their averaging) that closely resembled the form given by [10, 11] where each input tensor **T** is represented. There were, however, instances where the best results were obtained without all the tensors being represented as shown by the expression (4) for the larger training area.

The Reynolds-stress anisotropy tensor components for normal stress $\langle u'u' \rangle$ and shear stress $\langle u'v' \rangle$ obtained by the baseline Boussinesq model (1) giving

$$b_{11} = -C_\mu s_{11} = -0.09 \cdot (k/\varepsilon) \cdot (\partial U/\partial x), \quad b_{12} = -C_\mu s_{12} = -0.45 \cdot (k/\varepsilon) \cdot (\partial U/\partial y + \partial V/\partial x)$$

are compared in figures 2, 3 to the corresponding values $b_{11} = \langle u'u' \rangle/(2k) - (1/3)$ and $b_{12} = \langle u'v' \rangle/(2k)$ from the LES computations and to those from the model equations (3) and (4) discovered from GEP.

To calculate the b_{ij} components by (1), (3) and (4), all quantities are taken from the high-fidelity data [17]. Our primary goal for such *a priori* study is to examine whether improvement of the RSA tensor predictions against the baseline model indeed takes place, and whether new explicit expressions that better fit the anisotropy tensor have been found. One can observe (figures 2−4) that the Boussinesq model (1) is poor in modelling the anisotropy tensor, and the machine learning model equations (3) and (4) resemble the RSA tensor components more closely than the baseline model.

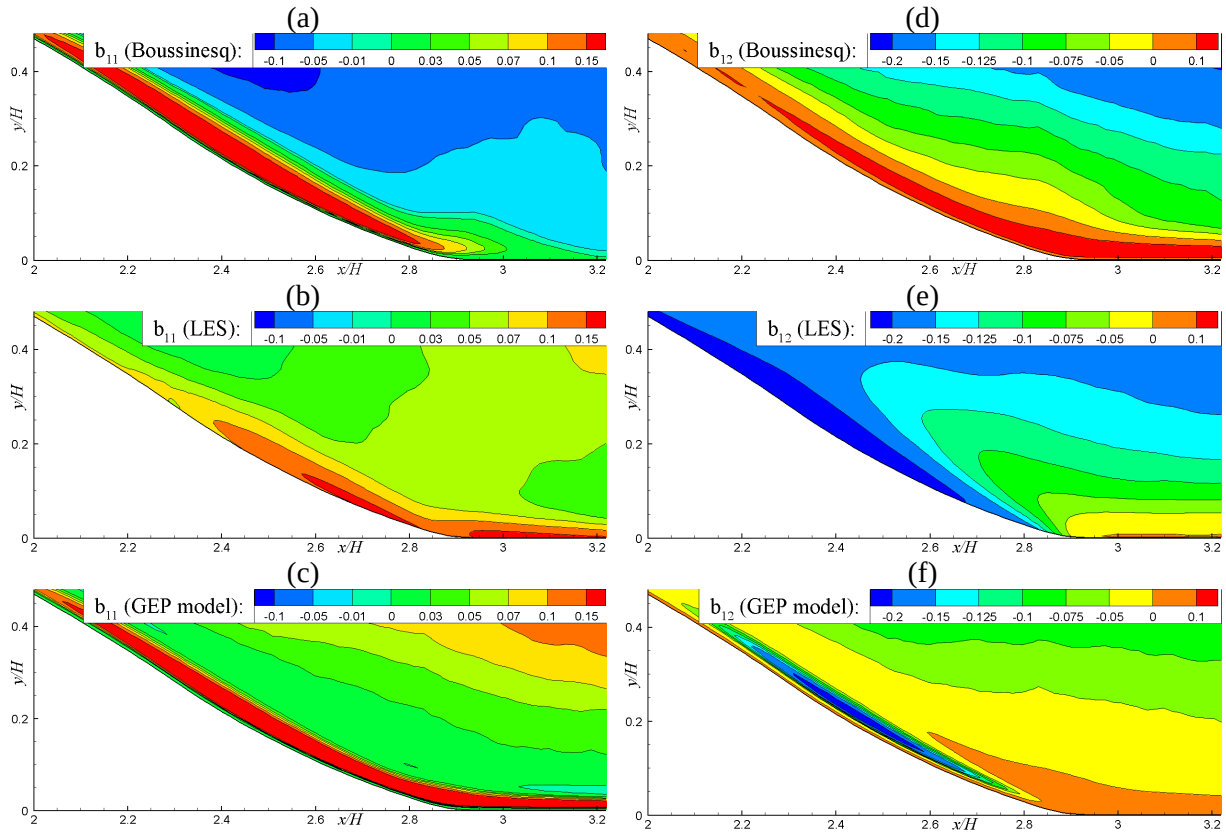

Figure 2. Contours of b_{11} (a, b, c) and b_{12} (d, e, f) estimated for area $2.0 < x/H < 3.2$, $0 < y/H < 0.45$: (a, d) the baseline Boussinesq model (1), (b, e) LES [17], (c, f) the present model (3) defined by GEP.

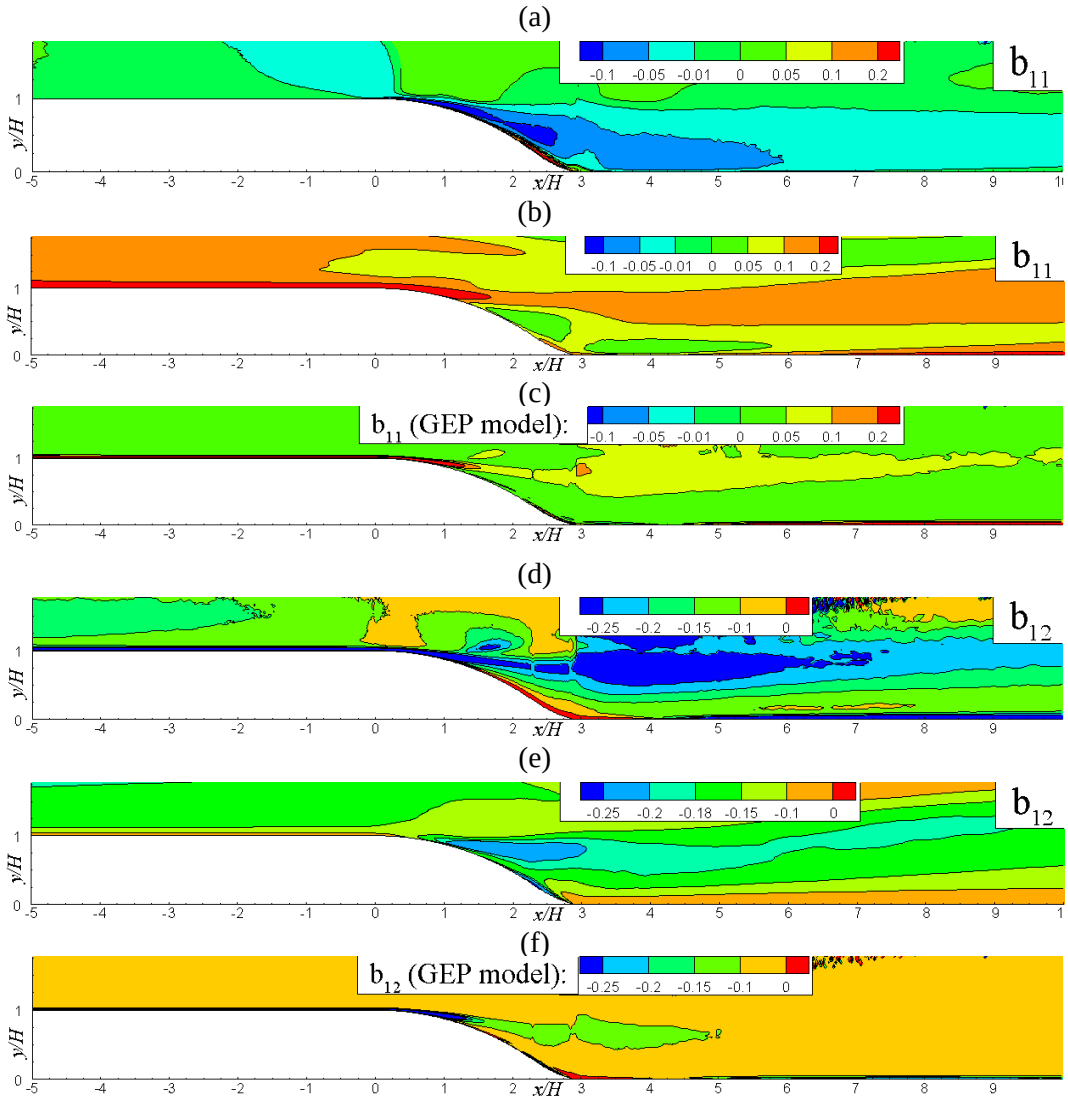

Figure 3. Contours of b_{11} (a, b, c) and b_{12} (d, e, f) estimated for $-5.0 < x/H < 10.0$, $0 < y/H < 1.25$: (a, d) the baseline Boussinesq model (1), (b, e) LES [17], (c, f) the present GEP model (4).

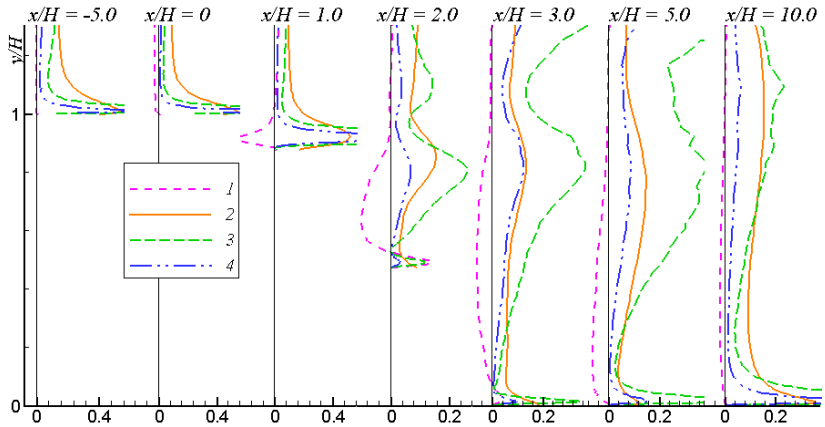

Figure 4. Vertical distributions of the RSA component b_{11} obtained from the Boussinesq model (1) (line *1*), LES [17] (line *2*), the present GEP models (3) and (4) (lines *3* and *4*).

5. Conclusion

The results for implementation of machine learning techniques to enhance the performance of the RANS turbulence models using the available high-fidelity data of LES are presented and discussed. For this, suitability of gene expression programming (GEP) which is a sort of symbolic regression is tested for the canonical case of a turbulent channel flow with the curved backward-facing step located on the bottom. Using GEP, the new explicit algebraic Reynolds-stress model is obtained, and the predictions for the Reynolds-stress anisotropy tensor components are improved in comparison with those for the baseline Boussinesq linear-gradient hypothesis as demonstrated in *a priori* study.

For future work, it can be checked whether the ML models obtained in the present study are consistently good for other flow cases and can be useful in general after inserting the new relation into the RANS solvers. Furthermore, it is of interest how the GEP technique can work when the tensor basis input is considered in its original shape and not flattened for computational convenience as has been made in the present research. Also, a more elegant method than averaging needs to be developed to obtain a faster converging equation so that limited runs of GEP return acceptable solutions.

Acknowledgements

The research was carried out within the state assignment of Ministry of Science and Higher Education of the Russian Federation (project No. 121030500149-8).

References

[1] Duraisamy K, Iaccarino G and Xiao H 2019 *Annu. Rev. Fluid Mech.* **51** 357
[2] Frohlich J and von Terzi D 2008 *Prog. Aerosp. Sci.* **44** 349
[3] Slotnick J, Khodadoust A, Alonso J, Darmofal D, Gropp W, Lurie E and Mavriplis D 2014 *NASA Technical Report* (NASA/CR-2014-21878)
[4] Durbin P 2018 *Annu. Rev. Fluid Mech.* **50** 77
[5] Yakovenko S N and Chang K C 2007 *Numer. Heat Transfer B* **51** 179
[6] Yakovenko S N and Chang K C 2019 *AIAA J.* **57** 279
[7] Ling J and Templeton J A 2015 *Phys. Fluids* **27** 085103
[8] Ling J, Kurzawski A and Templeton J 2016 *J. Fluid Mech.* **807** 155
[9] Parish E J, Duraisamy K 2016 *J. Comput. Phys.* **305** 758
[10] Weatheritt J and Sandberg R D 2016 *J. Comput. Phys.* **325** 22
[11] Zhao Y, Akolekar H D, Weatheritt J, Michelassi V and Sandberg R D 2020 *J. Comput. Phys.* **411** 109413
[12] Wu J-L, Xiao H and Paterson P 2018 Phys. Rev. Fluids **3** 074602
[13] Kaandorp M and Dwight R P 2020 *Comput. Fluids* **202** 104497
[14] Schmelzer M, Dwight R P and Cinnella P 2020 *Flow Turbul. Combust.* **104** 579
[15] Yakovenko S N and Razizadeh O 2020 *AIP Conf. Proc.* **2288** 030065
[16] Razizadeh O and Yakovenko S N 2020 Implementation of convolutional neural network to enhance turbulence models for channel flows *2020 Science and Artificial Intelligence conference (S.A.I.ence)* pp 1–4
[17] Bentaleb Y, Lardeau S and Leschziner M A 2012 *J. Turbul.* **13**(4) 1
[18] Pope S B 1975 *J. Fluid Mech.* **72**(2) 331
[19] Ferreira C 2001 *Complex Systems* **13**(2) 87