Development of explicit algebraic models of Reynolds stresses for flows in channels using gene expression programming

Haoyan Li, Mikhail P. Tokarev, Rustam I. Mullyadzhanov, and Sergey N. Yakovenko*
Kutateladze Institute of Thermophysics, Siberian Branch, Russian Academy of Sciences, 630090 Novosibirsk, Russian Federation

Abstract. To build a data-driven approximation for the Reynolds-stress anisotropy (RSA), the symbolic regression method of gene expression programming (GEP) is applied. Two tensor-basis terms from the algebraic expansion for the RSA tensor are used in the GEP algorithm. A new RANS-GEP model is tested in several flows in channels without/with bumps at different physical and geometrical parameters, where DNS data of high fidelity involved as a target for RSA are available. The results of RANS-DNS runs are also obtained where the RSA values in the mean momentum equation are taken directly from DNS to show ability to improve the model performance versus the conventional linear eddy viscosity model (LEVM).

1 Introduction

Application of eddy-resolving techniques (DNS, LES, measurement), which generate high-fidelity (HF) data, to industrial and environmental problems is restricted by high computing cost. In contrast, RANS closure models are computationally efficient so still contribute greatly to practical flow predictions [1], but often give visible deviations from HF data. Modification to RANS models improve predictions, however its discrepancy from HF data remains considerable [2]. Machine learning (ML) methods can help in data-driven turbulence modelling to enhance the model performance [3].

So far, several ML methods have been proposed to enhance RANS models: tensor basis neural network (TBNN) [4], field inversion and machine learning [5], physics-informed machine learning [6], convolutional neural network [7], tensor basis random forest (TBRF) [8, 9], universally interpretable machine learning [10], tensor basis gradient boosting [11]. Its workability is limited due to the black-box character of ML tools. On the other hand, to build explicit non-linear relations for the RSA tensor from data, symbolic regression can be used [12-15]. In particular, multi-dimensional gene expression programming (MGEP) [12, 13] and sparse regression [14] have been built as symbolic regression techniques to show possibilities to enhance the model performance in flows in channels of different geometry.

In the present work, the first two tensor-basis terms for the RSA components are taken to generate a new algebraic model by MGEP in canonical channel flows with periodic hills (PH) at different slopes and Reynolds numbers, and that in a square duct (SD). It has extensive data of DNS, LES, RANS computations [16-20] to be used for input features and for a comparative analysis.

2 Methodology

To close the RANS equations, the non-linear expression for the Reynolds stress tensor \( \tau_{ij} \) has been generated, for calibration of which, the MGEP tool [12] can be taken. It allows one to write RSA, \( a_g = 2k\partial_g = \tau_{ij} - (2/3)k\delta_{ij} \), as

\[
\begin{align*}
\alpha_i &= -2\nu_s \delta_{ij} + 2(k \delta_{ij} + g \delta_{ij})^2, \\
\gamma &= k/\omega \\
\sigma_i &= \frac{S_n}{\omega} = \frac{1}{2\omega} \left( \frac{\partial U_j}{\partial x_i} + \frac{\partial U_i}{\partial x_j} \right), \\
\beta_j &= -\sigma_i + g \delta_{ij}^2 + g \delta_{ij}^2, \\
\tau_{ij} &= \sigma_i - \beta_j.
\end{align*}
\]

where \( g(\lambda_1, \lambda_2) \) are linear and quadratic functions of scalar invariants \( \lambda_1 = \delta_{ij} \delta_{ij} \) and \( \lambda_2 = \delta_{ij} \delta_{ij} \), tensors \( T \) are functions of the strain \( (\sigma_i) \) and rotation \( (\tau_{ij}) \) tensors. Note that the first term, \( \beta_j \), is the LEVM model, is written separately, and the \( k-\omega \) model [21] is used to find the turbulent kinetic energy \( k \), and its dissipation \( \omega \).

The tensor basis \{\( T^1, T^2 \)\} is short in comparison to other MGEP techniques having taken three to four terms \{\( T^1, ..., T^4 \)\} [12, 13] or to the TBNN and TBRF tools with the complete basis \{\( T^1, ..., T^{10} \)\} [4, 8] since the first two terms contribute more than others [10].

The MGEP algorithm can be applied to handle with scalar, vector and tensor fields, for example to generate an explicit RSA relation based on HF data for canonical turbulent flows, following the steps:

- Define rates of genetic operations; choose a fitness function to evaluate the offspring, set the population size, the gene head length, the number of generations (iterations), those of genes in chromosome and plasmid.
- For the first generation, randomly generate the population of multi-gene chromosomes.

* Corresponding author: s.yakovenko@mail.ru

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).
• **Iteration start:** find the fitness of each individual and then choose the individuals satisfying the criteria; besides, collect the best individual separately.
• Apply genetic operations for chosen chromosomes, except the best chromosome.
• Gather plasmids of chromosomes, which encounter the genetic operations; exploit the genetic operations for the plasmids; distribute them to chromosomes.
• Produce the new generation using the chosen chromosomes and the best chromosome.
• **Iteration end:** go back to iteration start until the finishing the iterations due to their prescribed number.

Eight genetic operations to optimize chromosomes from generation to generation, are: mutation, inversion; one-point, two-point, gene recombination; transposition of insertion sequence elements, root/gene transposition.

More details of MGEP can be found in [12, 22].

For target values taken in the fitness function, the HF data for components of non-dimensional or dimensional RSA ($b_j$ or $a_{ij}$) are considered.

Input features (scalars $\lambda_1$, tensors $T'$) can be either from HF data as in [12], or from RANS solutions as in TBNN, TBRF [4, 8], or from both HF data and RANS solutions [13, 14].

### 3 Training and testing cases

The DNS data [17] in a PH flow at $Re = U_b H/v = 5600$ ($H$ is hill height, $U_b$ is bulk velocity) and slope $a = 1$ have initially been taken for training, with the baseline $k$-$\omega$ model results found using OpenFOAM [18]. For testing, PH flows at other slopes $a = 0.5, 1.5$ are available too [17, 18], as well as the LES data at $Re = 10595$ [16].

Next, a developed flow in a square duct at different Reynolds numbers $Re = U_b D/v = 10320$ (where $D$ is duct width) [19] and $2154 \leq Re \leq 7000$ [20] are examined as training and testing cases to build MGEP models.

### 4 Results

The way of [12] (called MGEP-WS), with quantities in (1) from the available HF data, is realized first (Fig. 1).

At the beginning, ten different runs were made using the MGEP-WS algorithm with input features taken from DNS [17], where ten different $b_j$ approximations were obtained. Here $N_t = 148$ data points located at $y/H < 1.58$ were selected from the whole data set having 14960 points, to save computing resources (increasing greatly for MGEP with $N_t$). Note that even smaller $N_t = 100$ was taken in [12] for the training dataset.

Moreover, the population size (i.e. number of individuals) has been set to 300 which is 1.5 times higher than that in [12, 13]. Finally, number of generations (iterations) was assigned to $N_g = 1000$ which is between 300 as in [12] and 1500 as in [13]. Such a number of $N_g$ was enough to get the convergence of the fitness function.

The gene head length was chosen to be 3 due to recommendation of the original GEP approach [22], whereas numbers of genes in chromosome and plasmid were taken to be 2 which corresponds as in [12] to numbers of basis tensors $\{T', T^2\}$ and scalar invariants $(\lambda_1, \lambda_2)$.

Then, five of ten expressions (having a minimum root mean square error (RMSE) determined from the difference between MGEP and DNS data for $b_j$ [4]) were chosen, and the average expression for the RSA tensor was found with the following functions

$$g^{(1)} = 0.1438 + 1.8\lambda_1 - 0.8\lambda_1\lambda_2$$

$$g^{(2)} = -1.72 + 2.17\lambda_1 - 0.4\lambda_2\lambda_1 - 0.75\lambda_2 - 0.375\lambda_2\lambda_2$$

A priori estimation, i.e. substitution of the data of [17] into the LEVM model, where $g^{(1)} = g^{(2)} = 0$, and into the MGEP model, demonstrates (Table 1) the ability of ML techniques to enhance the baseline turbulence model.

### Table 1. RMSE values relative to the DNS data [17], obtained from a priori estimation for PH flow.

<table>
<thead>
<tr>
<th>Model</th>
<th>$b_{11}$</th>
<th>$b_{12}$</th>
<th>$b_{22}$</th>
<th>$b_{13}$</th>
<th>$b_{33}$</th>
<th>$b_{23}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MGEP</td>
<td>0.100</td>
<td>0.059</td>
<td>0.083</td>
<td>0.059</td>
<td>0.083</td>
<td></td>
</tr>
<tr>
<td>LEVM</td>
<td>0.151</td>
<td>0.096</td>
<td>0.133</td>
<td>0.096</td>
<td>0.133</td>
<td></td>
</tr>
</tbody>
</table>

Next, the same procedure was performed to train the MGEP model for $a_{ij}$ with input features from DNS [17] collected in the shortened data set as mentioned above, fifty $b_j$ approximations were obtained, then six were selected with minimum RMSE values, and the average expression was generated with the following functions

$$g^{(1)} = 0.095 + 1.165\lambda_1 - 0.142\lambda_2 - 0.167\lambda_1\lambda_2 + 0.167\lambda_1\lambda_2$$

$$g^{(2)} = -1.524 + 1.842\lambda_1 - 0.083\lambda_2 - 0.167\lambda_1\lambda_1$$

Again, a priori estimation shows (Table 1, Fig. 2) the ability of ML tools to enhance the LEVM model.

However, after propagation of the expressions for both $b_j$ and $a_{ij}$ into the momentum, $k$ and $\omega$ equations (in the modified simpleFoam solver in OpenFOAM), no visible improvements were observed (Fig. 3). A possible reason of such a disadvantage is that using the HF data for training as in [12] may lead to inconsistency after its propagation, since the $k$ and $\omega$ values generated by the $k$-$\omega$ model are quite far [7, 9] from their DNS and LES...
counterparts used for training. So we have examined other ways to implement the MGEP techniques.

Fig. 2. Contours of RSA component $a_{11}$, a priori estimation.

First, the RANS-DNS runs (denoted also as RDNS) are made for both PH and SD flows, where the ‘frozen’ $a_{ij}$ values taken directly from [17, 19] are inserted into the RANS equations, and the velocity prediction is visibly improved versus that by LEVM (Fig. 4-5).

Note that the simple RSA propagation with the ‘explicit’ treatment of the ‘frozen’ $a_{ij}$ values yields spurious waves on the distributions and often leads to divergence seen by us and in [6, 8, 23]. Instead, we apply the ‘implicit’ treatment for the RSA propagation, which gives stable computations [6, 8, 23] and implies splitting

$$a_y = -2y_S + 2(0.77 + 1.23\lambda)T_y$$  \hspace{1cm} (7)

where $k$ and $\omega$ are obtained from the standard $k$-$\omega$ model [23], and the last term is frozen from DNS data [17].

Then, a posteriori study is performed by inserting the RSA model obtained by MGEP-AFA into a RANS solver (such a way follows the ‘implicit’ RSA treatment in RANS-DNS [23]) and testing it for the PH and SD cases at different physical and geometrical parameters.

In particular, for the SD case at Re = 10320 (with input features from DNS [19] collected in the shortened data set with $N_r = 145$ points from the duct cross-section), the MGEP model which turns to have a quite simple form

$$a_y = -(k/\omega)(U_{i,i} + U_{j,j}) + a_y$$  \hspace{1cm} (6)

5 Conclusions

The present study explores different ways to build the explicit non-linear algebraic models for the Reynolds stress anisotropy tensor by means of the ML technique of multi-dimensional gene expression programming.

The visible improvement of predictions can indeed be achieved using the MGEP models for flows in channels of different geometries.

As this work continuation, it is of interest to compare the obtained results with those of [4-15], as well as to find the best ML methods and optimal flow cases for
both training and testing, number of sample points taken for training, suitable ML hyperparameters, tensor basis terms \( \{T_n\} \), and input features \( \{\lambda_1, \lambda_2, \ldots\} \).

Ways to build turbulence models by ML methods for other complex-geometry turbulent flows having lateral and longitudinal obstacles, with heat and mass transfer, is to be considered too in the future studies.

The study is supported by a grant No. 22-19-00587 of Russian Science Foundation, https://rscf.ru/en/project/22-19-00587/.

References

22. C. Ferreira, Complex Systems. 13, 87 (2001)
23. A. Amarloo, P. Forooghi, A. Abkar, Phys. Fluids 34, 115102 (2022)