The 8th European Congress on Computational Methods in Applied Sciences and Engineering
ECCOMAS Congress 2022
5–9 June 2022, Oslo, Norway

REYNOLDS STRESS CORRECTION BY MACHINE LEARNING METHODS WITH PHYSICAL CONSTRAINTS

Thomas Philibert[1,2,3], Angelo Iollo[2,3], Andrea Ferrero[1] and Francesco Larocca[1]

[1] Department of Mechanical and Aerospace Engineering (DIMEAS)
Politecnico di Torino
Corso Duca degli Abruzzi, 24, 10129 Torino, Italy
e-mail: direttore.dimeas@polito.it, www.dimeas.polito.it

[2] National Institute for Research in Digital Science and Technology (INRIA)
Inria Bordeaux-Sud-Ouest
Avenue de la Vieille Tour, 200, 33405 Talence, France
email: contact-bordeaux@inria.fr, www.inria.fr

[3] Institut de Mathématiques de Bordeaux UMR 5251
Université de Bordeaux
Cours de la Libération, 351, 33405 Talence, France
email: institut@math.u-bordeaux.fr, www.math.u-bordeaux.fr

Key words: Machine learning, Turbulence modeling, RANS, Realizability constraints

Abstract. For the past three decade, Reynolds Average Navier-Stokes models have been widely used in the industry to simulate complex flows. However, these models suffer from limitations. Indeed there are still large discrepancies in the Reynolds stresses between the RANS model and high-fidelity data provided by DNS or experiments. This paper presents a strategy to correct the Menter SST model using an explicit algebraic model and two different neural networks: an multilayer perceptron (MLP) and a generative adversarial network (GAN). Moreover, in order to preserve the physical properties of the Reynolds stress tensor, we introduce a penalisation term in the loss of the GAN.

1 INTRODUCTION

Nowaday, with the increase in computationnal power, artificial intelligence plays a major role in various fields such as facial recognition, creating art and a broad range of non-linear physical systems. In the field of turbulence modeling, data-driven methods represent a promising research area since they have the potential to improve classical closure models for Reynolds Averaged Navier-Stokes (RANS) equations. Furthermore, the available computational power allows to perform scale-resolving simulations on industrial problems characterised by significantly high values of the Reynolds number. The high-fidelity data obtained by these expensive simulations can be exploited not just to

predict the performances of a system but to extract modelling knowledge on the physical phenomena which govern the flow field and develop new RANS models. Every improvement in RANS modelling capability has a significant impact on the industry since RANS models are still widely used for design, analysis and optimization of many aerodynamic components. In particular, RANS models can fail in the presence of separation, transition or high streamline curvature: several machine learning strategies can be exploited to compute corrections for existing RANS models in order to overcome their limitations. In the present work, a non-linear correction to the Boussinesq assumption is evaluated by means of artificial neural networks. The goal is to express the correction as a function of local flow features and to impose some physical constraints on the obtained correction. A database obtained by a high-fidelity DNS simulation available in the literature [3] is analyzed in order to investigate the potential of the proposed strategy.

2 PRESENTATION OF THE REFERENCE DATABASE

The flow over a periodic hill has become a very useful benchmark test case. It allows to evaluate performance of numerical methods, turbulence closure for RANS or LES or meshing strategy. In this configuration, there are some complex flow features appear like separation and reattachment that RANS models have difficulty in predicting. Xiao and al. performed a Direct Numerical Simulation[3] on different geometric configurations assuming incompressible flow and setting the Reynolds number to 5600. The data are produced using a six-order compact scheme for spatial discretization and third-order Adams-Bashforth scheme for time discretization on a Cartesian mesh.

3 RANS FRAMEWORK AND BOUSSINESQ'S ASSUMPTION

In this work, we propose a correction of the Menter SST two equations model[4] which is one of the most used RANS model. The Menter SST two equations model is :

$$\frac{\partial(\rho k)}{\partial t} + \frac{\partial(\rho u_j k)}{\partial x_j} = P - \beta^* \rho \omega k + \frac{\partial}{\partial x_j}\left[(\mu + \sigma_k \mu_t)\frac{\partial k}{\partial x_j}\right] \quad (1)$$

$$\frac{\partial(\rho \omega)}{\partial t} + \frac{\partial(\rho u_j \omega)}{\partial x_j} = \frac{\gamma}{\nu_t} P - \beta \rho \omega^2 + \frac{\partial}{\partial x_j}\left[(\mu + \sigma_\omega \mu_t)\frac{\partial \omega}{\partial x_j}\right] + 2(1 - F_1)\frac{\rho \sigma_{\omega 2}}{\omega}\frac{\partial k}{\partial x_j}\frac{\partial \omega}{\partial x_j} \quad (2)$$

In the framework of this RANS model, the Boussinesq's assumption is used to evaluate the anisotropic part of the Reynolds stresses. That assumption link linearly the anisotropic part of the Reynolds stress tensor to the strain rate tensor:

$$\tau_{ij}^{SST} = 2\mu_t S_{ij} - \frac{2}{3}\rho k \delta_{ij} \quad (3)$$

Where k is the dynamic turbulent energy and S_{ij} is the strain rate tensor:

$$k = \frac{1}{2}\overline{u_i' u_j'} \quad (4)$$

$$S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \quad (5)$$

and the kinetic eddy viscosity μ_t is computed by means of the Menter SST two equations model :

$$\mu_t = \frac{\rho a_1 k}{\max(a_1\omega, \Omega F_2)} \tag{6}$$

With :

$$\Omega = \sqrt{2R_{ij}R_{ij}}, \qquad F_2 = \tanh(arg_2^2),$$
$$R_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i}\right), \qquad arg_2 = \max\left(2\frac{\sqrt{k}}{\beta^*\omega d}, \frac{500\nu}{d^2\omega}\right),$$
$$\beta^* = 0.09, \qquad a_1 = 0.31, \qquad \omega = \frac{\epsilon}{k}$$

The quatity d is the distance from the closest wall and ϵ is the rate of dissipation of the turbulent kinetic energy. In order to assess the quality of the Boussinesq's assumption, we can introduce an indicator γ[2] defined by the inner product between the anisotropic stress tensor a_{ij} and the strain rate tensor:

$$a_{ij} = \tau_{ij}^{DNS} + \frac{2}{3}k\rho\delta_{ij}$$
$$\gamma = \frac{a_{ij}S_{ij}}{\sqrt{a_{nm}a_{mn}S_{pq}S_{qp}}} \tag{7}$$

Where $\tau_{ij}^{DNS} = -\rho\overline{u_i' u_j'}$.
This indicator gamma can be applied to high-fidelity data to test the validity of the Boussinesq assumption: it provides a measure of the alignment between the anisotropic part of the Reynolds stress tensor and the strain rate tensor. Therefore, γ varies between -1 and 1. If $\gamma = -1$, it means that the two tensors are colinear in the opposite direction. A value of 0 means that the two tensors are perpendicular and a value of 1 means an alignement between the two and therfore the Boussinesq approximation holds.

The γ field obtained by applying Eq. 7 to the DNS data provided by [2] is reported

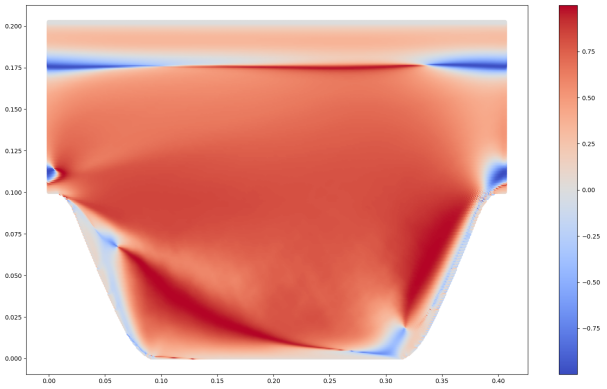

Figure 1: Alignment of the anisotropic stress tensor and the strain rate tensor

in Figure 1. Some regions with poor alignment can be seen. The objective of this work will be to add a corrective term in the Boussinesq's assumption in order to correct this discrepancy.

3.1 Explicit algebraic model

In order to correct the Boussinesq's assumption, it is possible to follow the approach proposed in the framework of the Explicit Algebraic Reynolds Stress model [8](EARSM model) in which additional terms are introduces in order to form a non-linear constitutive stress-strain relationship that reduces the discrepancy showed before.
The EARSM model has been developed in the past invoking a weak equilibrium hypothesis as well as the use of the Cayley-Hamilton theorem in order to define a tensor basis V_{ij}^k and some scalar invariants I_k that can be used to correct the Reynolds stress tensor. This formulation assures that Galilean invariance is respected. In this case, Pope[1] has already derived the relevant integrity basis and already proven that in the most general incompressible case, an eddy viscosity model that is function of only S and R can be expressed as linear combination of 10 isotropic basis tensors :

$I_1 = \mathrm{Tr}(\tilde{S}^2)$	$I_4 = \mathrm{Tr}(\tilde{R}^2\tilde{S})$
$I_2 = \mathrm{Tr}(\tilde{R}^2)$	$I_5 = \mathrm{Tr}(\tilde{R}^2\tilde{S}^2)$
$I_3 = \mathrm{Tr}(\tilde{S}^3)$	

Table 1: Invariants

$V_1 = \tilde{S}$	$V_6 = \tilde{R}^2\tilde{S} + \tilde{S}\tilde{R}^2 - \frac{2}{3}I\,\mathrm{Tr}(\tilde{S}\tilde{R}^2)$
$V_2 = \tilde{S}\tilde{R} - \tilde{R}\tilde{S}$	$V_7 = \tilde{R}\tilde{S}\tilde{R}^2 - \tilde{R}^2\tilde{S}\tilde{R}$
$V_3 = \tilde{S}^2 - \frac{1}{3}I\,\mathrm{Tr}(\tilde{S}^2)$	$V_8 = \tilde{S}\tilde{R}\tilde{S}^2 - \tilde{S}^2\tilde{R}\tilde{S}$
$V_4 = \tilde{R}^2 - \frac{1}{3}I\,\mathrm{Tr}(\tilde{R}^2)$	$V_9 = \tilde{R}^2\tilde{S}^2 + \tilde{S}^2\tilde{R}^2 - \frac{2}{3}I\,\mathrm{Tr}(\tilde{S}^2\tilde{R}^2)$
$V_5 = \tilde{R}\tilde{S}^2 - \tilde{S}^2\tilde{R}$	$V_{10} = \tilde{R}\tilde{S}^2\tilde{R}^2 - \tilde{R}^2\tilde{S}^2\tilde{R}$

Table 2: Tensor Basis

$$\tilde{S} = \tau S \qquad \tilde{R} = \tau R \qquad \tau = \frac{k}{\epsilon}$$

The correction of the Reynold's stress tensor is defined as :

$$\tau_{ij}^{SST} = 2\mu_t S_{ij} - \frac{2}{3}\rho k\delta_{ij} + \rho k \sum_{k=1}^{10} f_k(I_1, I_2, ..., I_5)V_{ij}^k \tag{8}$$

Any correction which is of this form will satisfy Galilean invariance.
The goal then is to find the function : $f_k(I_1, ...I_5)$. However, it has been demonstred [1] that only two invariants and three tensors basis are sufficient to have good results in a two-dimensional flow and the time for the training is greatly reduced.

From (8), we deducted that for each point of the mesh we have:

$$\rho k \sum_{k=1}^{3} f_k(I_1, I_2)V_{ij}^k = \tau_{ij}^{SST} - 2\mu_t S_{ij} + \frac{2}{3}\rho k\delta_{ij} \tag{9}$$

And we can compute f_k by solving a linear system since all the variables are available from the reference database (we impose $\tau_{ij}^{SST} = \tau_{ij}^{DNS}$).
However, if f_k is directly evaluated it is possible to obtain some outliers values (Figure

2, 3, 4) which will deteriorate the following training process: the origin of this outliers could be related to the discretization error introduced in the computation of the velocity gradients from the reference database. To limit this behavior, we assume that the solution of the system on each points should be close to the solution of the system at neighbouring points. From (9), we can transform this equation as :

$$Ax - b = 0 \tag{10}$$

If we consider this equation on the point P_0 and $P_1, ..., P_n$ are the neighbors of P_0, then we want to resolve the minimization problem :

$$\arg\min_x ||A_0 x - b_0||^2 + \alpha \sum_{i=1}^{n} ||A_i x - b_i||^2 \tag{11}$$

As reported in Figures 5, 6, 7, there are still outliers values but the scale is much lower, which will help the training of the neural networks.

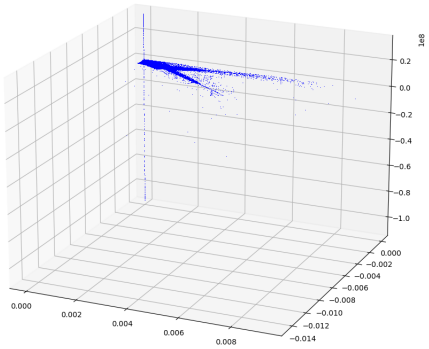

Figure 2: $f_1(I_1, I_2)$ without accounting for neighbours

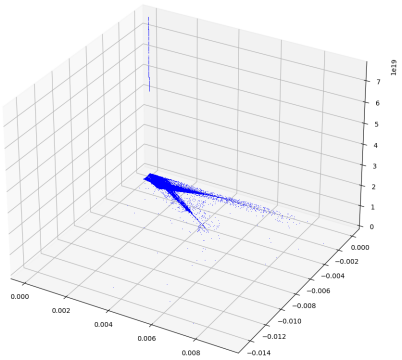

Figure 3: $f_2(I_1, I_2)$ without accounting for neighbours

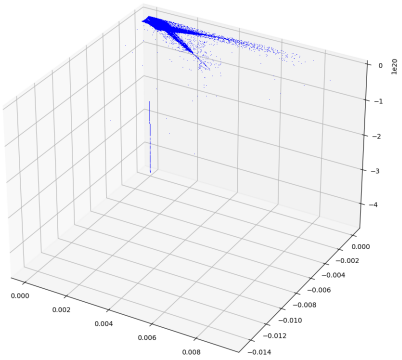

Figure 4: $f_3(I_1, I_2)$ without accounting for neighbours

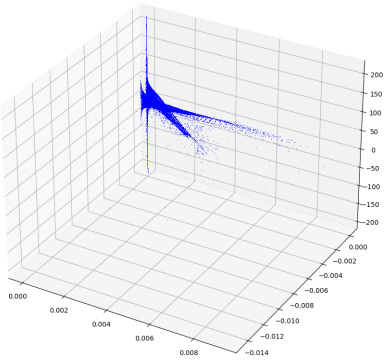

Figure 5: $f_1(I_1, I_2)$ taking into account the neighbours

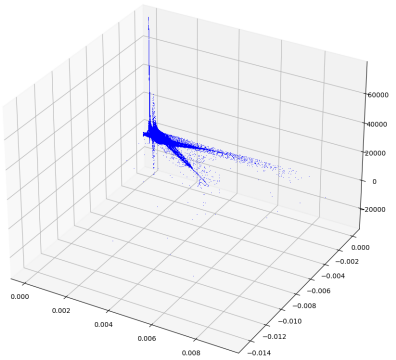

Figure 6: $f_2(I_1, I_2)$ taking into account the neighbours

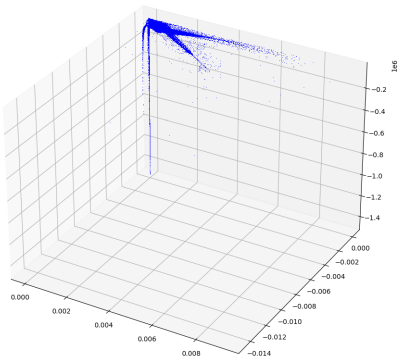

Figure 7: $f_3(I_1, I_2)$ taking into account the neighbours

4 MACHINE LEARNING

4.1 Neural network

As explained in the previous section, the goal is to determine the function $f_k(I_1, I_2)$ from (8). In order to do so, we will use three multilayer perceptrons. Each one of the neural networks will take as input the two invariant and as output the value of one of the function f_k. The mean squared error will be used as loss function.

The three neural networks have the same architecture. They are composed of 7 layers of 100 neurons each. We chose this architecture using a Bayesian hyperparameters optimization [7]. Furthermore, those networks will use connection skipping to help with the vanishing gradient problem and stabilize training and convergence. Connection skipping in deep architecture is a technique which allow to skip some layers in the neural network and feed the output of one layers as the input of the next layers (not only the very next one). There are multiple ways to do a connection skipping but in this case, we will concatenate the output of one layer with the output of the next layer to input in the next third layer.

To train this network, we will use 80% of the mesh of one geometry as training set and the 20% left will be used as validation set. Then, we will test it on a different geometry from the database described before. The difference between the training set (geometry A) and the test set (geometry B) is the width of the domain.

4.2 Results

In the Figures 8, 9, 10, the blue points are the different coefficients we extracted with the minimization problem and the red points are the coefficients we get by the neural networks described in the previous section from the training data.

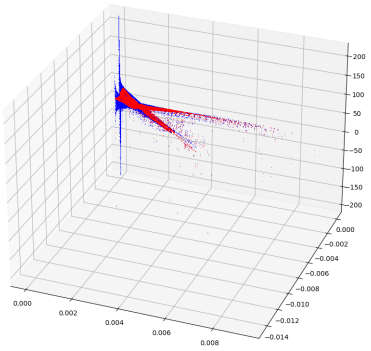

Figure 8: $f_1(I_1, I_2)$ on the training set on geometry A

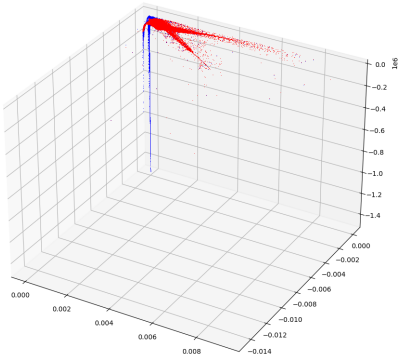

Figure 9: $f_2(I_1, I_2)$ on the training set on geometry A

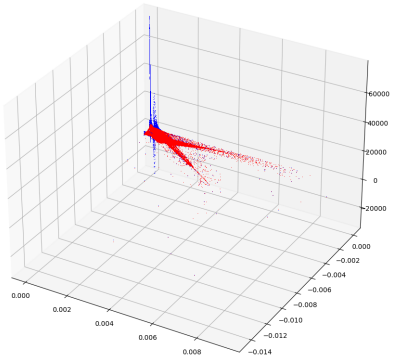

Figure 10: $f_3(I_1, I_2)$ on the training set on geometry A

Thoses figures show that the network has learned the general shape of the data except of the outliers.
Figures 11a and 15a represents the error of the Reynolds stress tensor computed with the SST model and the correction on the training set (Figure 11) and the test set (Figure 15). The Figure 11b and 15b represents the error of the Reynolds stress tensor computed with just the SST model. The Errors are computed as follow :

$$||\tau^{DNS} - \tau^{SST+correction}||_{L2} \tag{12}$$

$$||\tau^{DNS} - \tau^{SST}||_{L2} \tag{13}$$

On the Figure 11 and 15, there is an overall improvement except on one region on the right. This may be to numerical errors that were introduced when the gradients were computed from the available high-fidelity data.

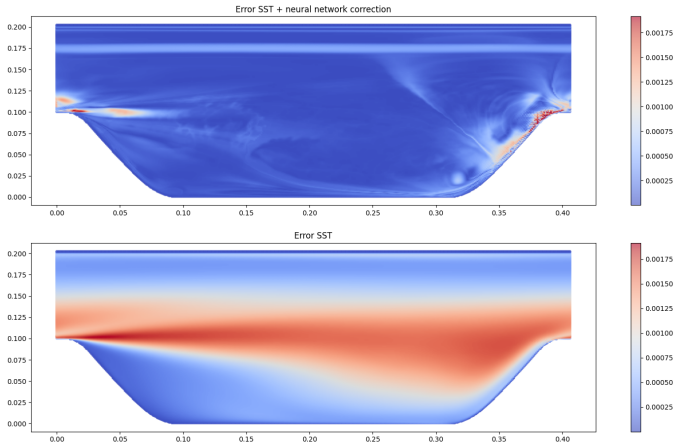

Figure 11: Error of the Reynold stress tensor on the training set on geometry A

The Figure 12, 13, 14 are the coefficient extracted from the minimization problem (blue) and from the neural networks (red) on the training set.

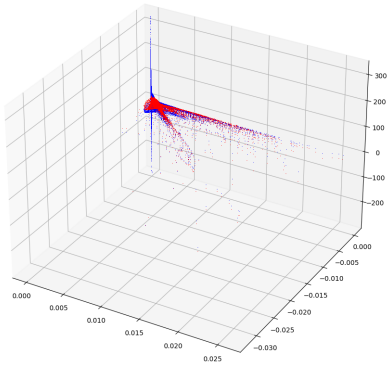

Figure 12: $f_1(I_1, I_2)$ on the test set on geometry B

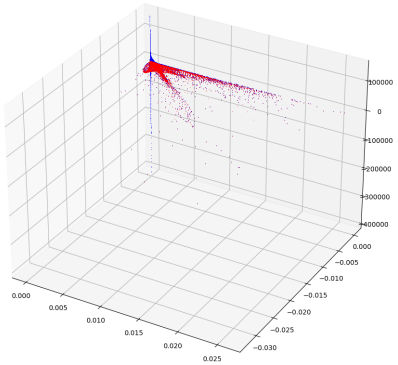

Figure 13: $f_2(I_1, I_2)$ on the test set on geometry B

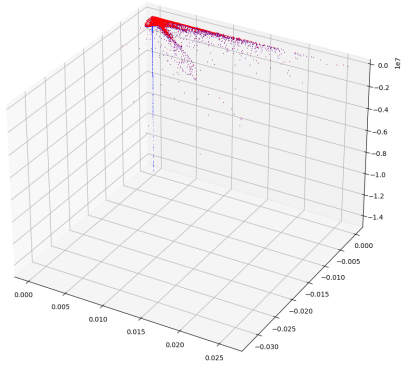

Figure 14: $f_3(I_1, I_2)$ on the test set on geometry B

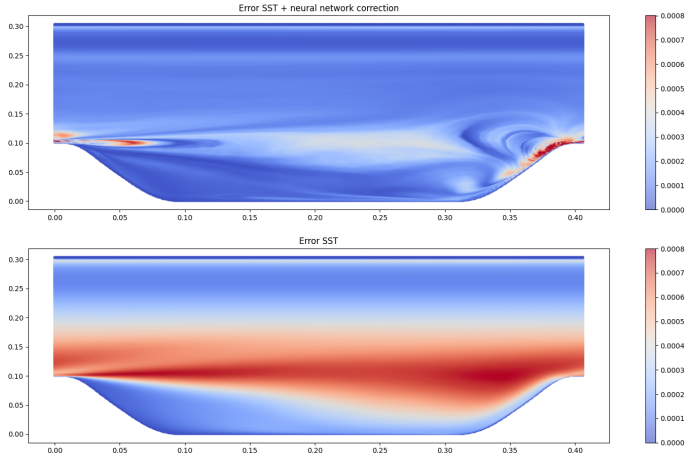

Figure 15: Error of the Reynold stress tensor on the test set on geometry B

4.3 Generative adversarial network

A Generative adversarial network [6] (GAN) is a network composed by two neural networks. One of them is called the discriminator, whose task is to discriminate generated from real data. And the other is the generator, which has to generate data (Figure 16).

The generator takes random noise as input and generates data that the discriminator has to discriminate from the real data. The training of GANs involves both finding the parameters of a discriminator that maximize its classification accuracy, and finding the parameters of a generator which maximally confuse the discriminator. The cost of training is evaluated using a value function, $V(D, G)$ that depends on both the generator and the discriminator. The training involves solving :

$$\min_G \max_D V(D, G) = E_{x \sim p_{data}(x)} \log(D(x)) + E_{z \sim p_z(z)} \log(1 - D(G(z)) \tag{14}$$

Where $p_{data}(x)$ $p_z(z)$ denote the probability distributions of training data and latent space vector z, respectively, and E indicates expectations

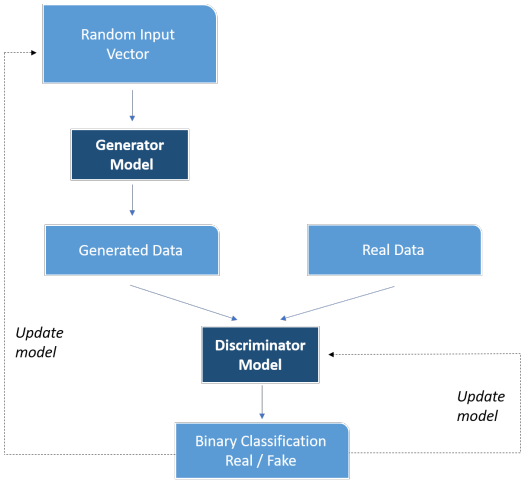

Figure 16: Architecture of a generative adversarial network

The goal is to train each of the network one after the other to improve the quality of the generated data. Indeed, when the generator is trained it forces the discriminator to generate better quality data. Furthermore, when the generator is trained, it forces the discriminator to improve its classification capability.

In our case, the generator will be composed by 4 layers of 50 neurons and the discriminator will be composed by 8 layers of 20 neurons. We chose this architecture using a Bayesian hyperparameters optimization [7]. Furthermore, in the generator loss[5], we will add a penalisation term. Indeed, we would like that the data generated follow the physics. And so we add a penalisation on the realizability conditions :

$$\overline{\rho u_i'^2} \geq 0 \tag{15}$$

$$\left(\overline{\rho u_i' u_j'}\right)^2 \leq \overline{\rho u_i'^2}\ \overline{\rho u_j'^2} \tag{16}$$

Since those conditions are on the Reynolds stress tensor, we have to create some random vectors to compute a Reynolds stress tensor with the SST model. These new data should follow the distribution of the DNS database. The input of the generator model includes the generated invariants, tensor basis and the generated Reynolds stress tensor we should have with the SST model. From here, the generator will try to find the functions f_k as previously but we compute the Reynolds stress tensor with the correction in the network in order to apply the penalisation. The functions created are then fed to the discriminator with some data obtain from DNS.

4.4 Results

As previously, this network will be trained on 80% of the mesh of one geometry and validated on the other 20% (geometry A) and then tested on a different geometry from

the DNS database (geometry B).

The Figure 17 and 18 show the error of the GAN on the training set (17a) and on the test set (18a) compared with the SST model (17b and 18b) As we can see, we have an averall improvement. However, this improvement is not as good as the one we had previously. But since we added the realizability condition, the results we got with this strategy tend to satisfy the realizability conditions. The Figure 19 show the constraint applied to the result of the classic neural network. On the right of the graphe, there is a zone that is not zero meaning that the constraint is not respected here. However, the constraint is always respected while using the GAN.

All the test performed in this work are based on a-priori analysis: future work will be dedicated to a-posteriori analysis in which the corrected models will be implemented and tested in a CFD solver.

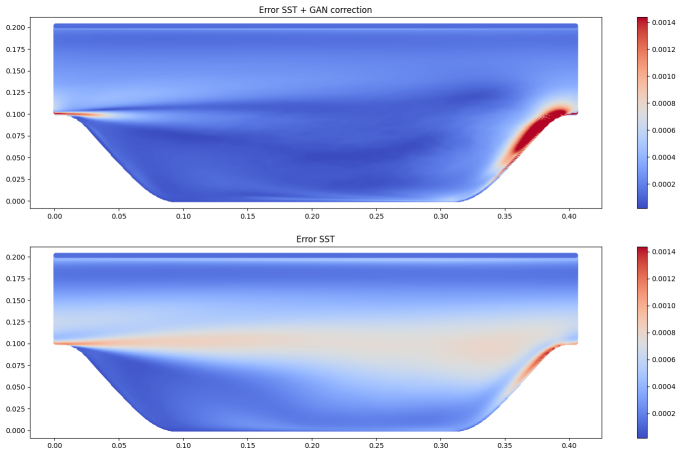

Figure 17: Error of the Reynold stress tensor on the training set on geometry A

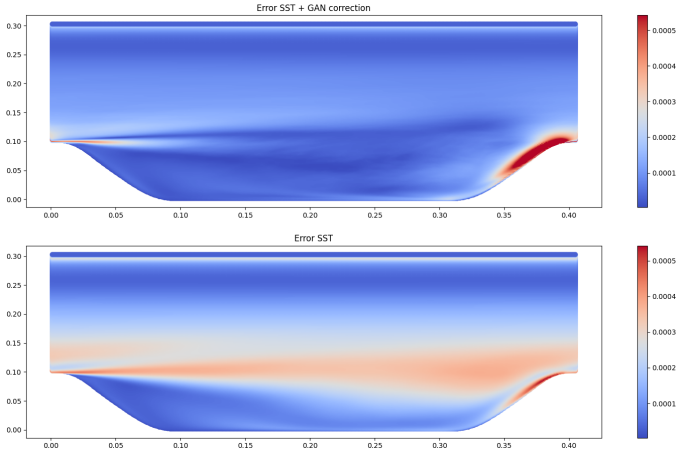

Figure 18: Error of the Reynold stress tensor on the test set on geometry B

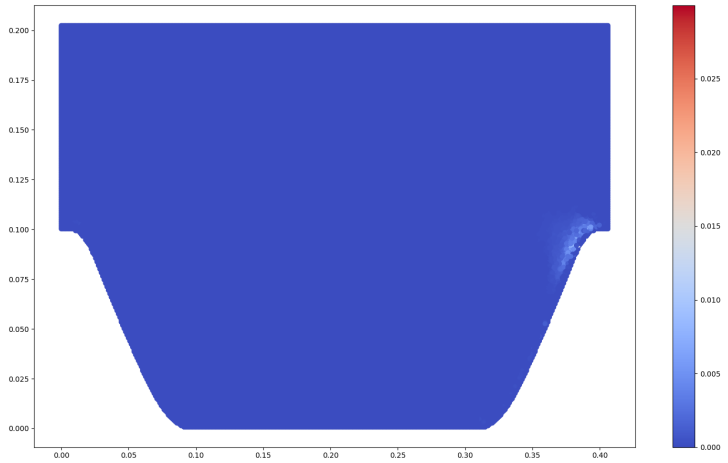

Figure 19: Constraint applied to the result of the Reynolds stress tensor from the Neural network

REFERENCES

[1] Pope, S. A more general effective-viscosity hypothesis. *J. Fluid Mech.*(1975) **72**:331–340.

[2] Sandberg, R.D. and Michelassi, V. The Current State of High-Fidelity Simulations for Main Gas Path Turbomachinery Components and Their Industrial Impact. *Flow, Turbulence and Combustion*(2019) **102**:797–848

[3] Xiao, H, Wu, J-L, Laizet, S and Duan, L. Flows over periodic hills of parameterized geometries: A dataset for data-driven turbulence modeling from direct simulations. *Computers and Fluids*(2020) **200**

[4] Menter, F.R. Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications. *AIAA Journal*(1994) **32**: 1598–1605

[5] Yang, Z, Wu, J-L and Xiao H. Enforcing Deterministic Constraints on Generative Adversarial Networks for Emulating Physical Systems.

[6] Goodfellow,I et al. Generative adversarial Nets. *Advances in Neural Information Processing Systems*(2014) **27**:2672–2680

[7] Snoek, J, Larochelle, H and Adams, R.P. Practical Bayesian Optimization of Machine Learning Algorithms. *Advances in Neural Information Processing Systems*(2012) **25**:2951–2959.

[8] Wallin, S and Johansson, A.V. An explicit algebraic Reynolds stress model for incompressible and compressible turbulent flows. *Journal of Fluid Mechanics*(2000) **403**:89–132.