

# RANS predictions of trailing-edge slot flows using heat-flux closures developed with CFD-driven machine learning

## Original article

### Article history:

Submission date: 29 September 2020

Final revision date: 15 January 2021

Acceptance date: 6 February 2021

Publication date: 25 May 2021



### \*Correspondence:

CL: chitrarth.lav@unimelb.edu.au

### Peer review:

Single blind

### Copyright:

© 2021 Lav et al. © This is an open access article distributed under the Creative Commons Attribution License (CC-BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited and its authors credited.

### Keywords:

turbulence; machine learning; RANS; heat flux models

### Citation:

Lav C., Haghiri A., and Sandberg R. D. (2021). RANS predictions of trailing-edge slot flows using heat-flux closures developed with CFD-driven machine learning. *Journal of the Global Power and Propulsion Society. Special Issue: Data-Driven Modelling and High-Fidelity Simulations: 1–13.*  
<https://doi.org/10.33737/jgpps/133114>

Chitrarth Lav<sup>1,\*</sup>, Ali Haghiri<sup>1</sup>, Richard D. Sandberg<sup>1</sup>

<sup>1</sup>Department of Mechanical Engineering, Grattan Street, University of Melbourne, Parkville, 3010, Australia

## Abstract

Accurate prediction of the wall temperature downstream of the trailing-edge slot is crucial to designing turbine blades that can withstand the harsh aerothermal environment in a modern gas turbine. Because of their computational efficiency, industry relies on low-fidelity tools like RANS for momentum and thermal field calculations, despite their known underprediction of wall temperature. In this paper, a novel framework using a branch of machine learning, gene-expression programming (GEP) [Zhao et al. 2020, *J. Comp. Physics*, 411:109413] is used to develop closures for the turbulent heat-flux to improve upon this underprediction. In the original use of GEP ("frozen" approach), the turbulent heat-flux from a high-fidelity database was used to evaluate the fitness of the candidate closures during the symbolic regression, however, the resulting closure had no information of the temperature field during the optimisation process. In this work, the regression process of the GEP instead incorporates RANS calculations to evaluate the fitness of the candidate closures. This allows the inclusion of the temperature field from RANS to advance the iterative regression, leading to a more integrated heat-flux closure development, and consequently more accurate and robust models. The GEP-based CFD-driven framework is demonstrated on a trailing edge slot configuration with three blowing ratios. Full *a posteriori* predictions from the new closures are compared to high-fidelity reference data and both conventional RANS closures and closures obtained from the "frozen" approach.

## Introduction

In the modern gas turbine engines of today, the high-pressure turbine (HPT) blades are exposed to harsh aerothermal environments, with the temperatures of the working fluid in excess of the metal melting point temperature. The need for effective cooling of the metal surface is crucial to maintaining the component life as well as the safe operation of the engine. One such area of cooling that is a challenge is the trailing-edge slot (also known as the pressure-side bleed), as shown in [Figure 1](#). The aerodynamic requirements dictate the trailing-edge be tapered to increase the aerodynamic efficiency, which prevents internal cooling solutions. Instead, the cooling is provided by injecting the coolant fluid out of the slot, forming a film on the metal cutback surface. The interaction of this film with the extraneous working fluid leads to interesting flow dynamics, producing either wall-jet or wall-wake

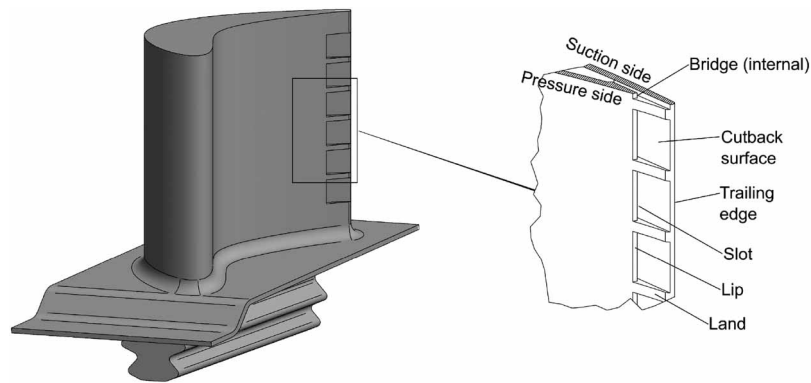


Figure 1. Schematic of the trailing-edge slot, adapted from Wong et al. (2016).

type flows. The flow physics for the trailing-edge slot has received attention both experimentally (Kacker and Whitelaw, 1969, 1971) and numerically (Medic and Durbin, 2005; Joo and Durbin, 2009; Haghiri and Sandberg, 2020) in order to understand how effective the slot fluid is in shielding the metal surface, quantified by a non-dimensional adiabatic wall effectiveness  $\eta$

$$\eta = \frac{T_w - T_{fs}}{T_s - T_{fs}}, \quad (1)$$

where  $T_w$ ,  $T_{fs}$  and  $T_s$  represent the adiabatic wall, freestream and slot fluid temperatures, respectively. Lower values of  $\eta$  indicate higher wall temperatures, on account of the negative denominator in Equation 1. From a design perspective, predicting these wall temperatures numerically is fraught with challenges, stemming from either the excessive cost of high-fidelity simulations or unacceptable inaccuracies from low-fidelity tools like Reynolds-Averaged Navier-Stokes (RANS) (Launder and Rodi, 1983; Holloway et al., 2002a) or unsteady RANS (URANS) (Holloway et al., 2002b; Ivanova and Laskowski, 2014). The poor prediction of  $\eta$  for these flows when using RANS-based closures is attributed to the incorrect momentum and thermal mixing modelled, which ties into the closures for the turbulence stress (Boussinesq approximation) and turbulent heat-flux (gradient diffusion hypothesis). With the availability of high-fidelity data for these flows, building better models of the closures for the stress and the heat-flux through machine-learning approaches is gaining increasing attention. In a RANS context, studies such as Sandberg et al. (2018) and Haghiri and Sandberg (2020) developed closures for the turbulent heat-flux by fixing the velocity field from high-fidelity data while the URANS closures in the studies of Lav et al. (2019a) and Lav and Sandberg (2020) were developed for both the turbulence stress tensor and the heat-flux vector. These studies obtained the closures with the aid of a genetic evolutionary algorithm, the gene-expression programming for RANS (Weatheritt and Sandberg, 2016) and for URANS (Lav et al., 2019b). This machine-learning algorithm performs symbolic regression to obtain a mathematical expression that attempts to match the reference quantity of interest, obtained from high-fidelity data, and thus far has been used to develop expressions for the turbulence stress tensor (via tensorial regression) and the turbulent heat-flux vector (via scalar regression). It is noteworthy that the expressions are obtained by regressing against the high-fidelity quantity of interest, leaving the utilisation of RANS or URANS out of the closure development process, also known as “training”. While gene-expression programming (GEP) in the present form leads to quick closure development times (of the order of minutes), one crucial issue with this approach is the lack of any information related to the velocity or temperature fields during the closure development process. Implementing these closures into RANS/URANS solvers *a posteriori* has shown to improve the flow prediction remarkably over the commonly used closures but improvement of the turbulent stress tensor or heat flux vector prediction does not always lead to equally improved mean flow predictions. In that vein, the GEP algorithm was modified by Zhao et al. (2020) to include contributions of the velocity field into the closure development process to build more robust turbulence stress closures for high-pressure and low-pressure turbine wake flow predictions. This was achieved by running RANS calculations within a loop during the algorithm’s execution to develop turbulence stress closures that improved the flow prediction instead of just the turbulence stress. The modification of the algorithm in this way was labelled as CFD-driven machine learning (or “looped” training) while the conventional use of the GEP is denoted as data-driven machine-learning (or “frozen” training).

Thus, in the present study the CFD-driven machine learning approach of Zhao et al. (2020) is adapted to develop closures for the turbulent heat-flux, where RANS calculations for the temperature field are incorporated into the training and used as a target for improvement. High-fidelity datasets of three blowing ratios for a given lip thickness geometry from Haghiri and Sandberg (2020) are used to lead the model development and the closures obtained from this looped training are compared with the high-fidelity data, the conventional frozen trained closures in addition to the commonly used gradient diffusion hypothesis closure.

## Numerical setup

### Reference datasets

The highly-resolved large eddy simulation (LES) datasets of a fundamental trailing edge slot, performed by Haghiri and Sandberg (2020), are used in the present study as the reference for sake of model development. The data was produced using an in-house multi-block structured compressible Navier–Stokes solver HiPSTAR (Sandberg, 2015). Figure 2 shows a schematic of the flow configuration. At the inlet boundary, a cold wall jet and a plane hot co-flow with different speeds are specified, separated by a flat plate with thickness  $t$ . The inlet mean velocity profiles were adapted from the experimental study of Whitelaw and Kacker (1963). In the LES, only mean profiles were prescribed at the inlet boundary and no fluctuations were added. All walls were set to no-slip boundaries and adiabatic temperature conditions were used. The wall adapting local eddy viscosity (WALE) model (Nicoud and Ducros, 1999) was used for modelling of the sub-grid scale stresses. Table 1 summarises the details of the three cases. Interested readers can refer to the original paper of Haghiri and Sandberg (2020) for more details on the flow configuration and numerical setup.

### RANS-based scalar flux closures

Even with the recent rise of computational power, LES is prohibitively restrictive within an industrial design context, ensuring that the dependence on RANS continues. Unfortunately, RANS is known to suffer from two major sources of modelling errors: Reynolds stress and turbulent heat-flux closures. In the present study, the focus is placed only on the latter and it is assumed that the momentum field (velocity) and other inputs to heat-flux models are known accurately (from time-averaged LES in this case). The mean temperature transport equation for an incompressible flow is given (in non-dimensional form) below:

$$\frac{\partial \bar{T}}{\partial t} + \bar{u}_i \frac{\partial \bar{T}}{\partial x_i} = \frac{1}{Re Pr} \frac{\partial^2 \bar{T}}{\partial x_i \partial x_i} - \frac{\partial \overline{u_i T'}}{\partial x_i}, \tag{2}$$

where

$$x_i = \frac{x_i^*}{L_{ref}}, \quad \bar{u}_i = \frac{\overline{u_i^*}}{u_{ref}}, \quad t = \frac{t^* u_{ref}}{L_{ref}}, \quad \bar{T} = \frac{\overline{T^*}}{T_{ref}}. \tag{3}$$

In the above,  $x_i$  and  $t$  are the spatial coordinate vector and time, respectively. In addition,  $u_i$  is the velocity vector,  $T$  is the temperature,  $Re$  is the Reynolds number and  $Pr$  is the Prandtl number. It should also be noted that “asterisk” and “overbar” represent dimensional and time-averaged variables, respectively. The use of the incompressible framework here is justified due to the low Mach number (<0.3) in the freestream and the jet

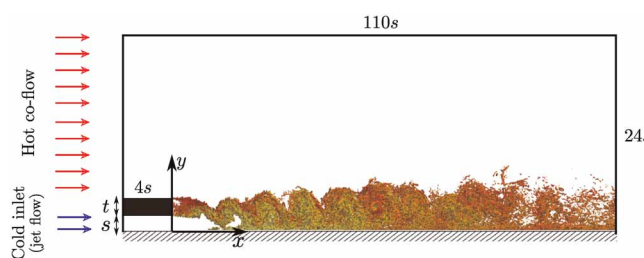


Figure 2. Computational domain (not to scale). The structures shown represent contours of the Q-criterion, coloured by the temperature field in the range [0.85, 1.0], non-dimensionalised by the freestream temperature.

Table 1. Simulation parameters.

CASE	$t/s$	$BR$	$Re_{slot}$	$U_{fs}$ (m/s)	$T_{slot}$ (K)	$T_{fs}$ (K)
A1	1.14	1.26	$\approx 12,000$	19.2	273	323
A2	1.14	1.07	$\approx 10,000$	19.2	273	323
A3	1.14	0.86	$\approx 8,000$	19.2	273	323

flows. Furthermore, the buoyancy effects are negligible due to the small temperature differences between the free-stream and the jet flows, implying that the energy equation can be decoupled from the momentum equations (Sandberg et al., 2018). As such, the temperature field can be solved passively.

The turbulent heat-flux is commonly modelled using the gradient diffusion hypothesis (GDH), assuming that the heat-flux terms are in the direction of their corresponding maximum mean temperature gradient

$$\overline{u'_i T'} = -\alpha_t \frac{\partial \bar{T}}{\partial x_i}, \quad (4)$$

where the diffusivity  $\alpha_t = \nu_t / Pr_t$  is modelled as eddy viscosity ( $\nu_t$ ) over turbulent Prandtl number ( $Pr_t$ ). The eddy viscosity concept is commonly used in closures of Reynolds stress tensor while the turbulent Prandtl number is assumed to be constant, with  $Pr_t = 0.9$  widely used in the literature. The GDH implies that the turbulence isotropically diffuses the heat, which is known not to be the case, except perhaps for simple free shear flows (Fabris, 1979). There are indeed more accurate approximations to model the turbulent heat-flux vector, such as the Generalised Gradient Diffusion Hypothesis (Daly and Harlow, 1970) or the Higher Order Generalised Gradient Diffusion Hypothesis (Abe and Suga, 2001) which have been explored previously (Ling et al., 2015; Weatheritt et al., 2020). However, in this study the closures are exclusively developed for the GDH for two main reasons. First, the simplicity of the GDH closure can not be discounted, which helps in running stable calculations in the RANS solver. Second, the GDH closure developed in this study using the CFD-driven approach can serve as a direct comparison with the GDH trained closure obtained from the frozen approach in Haghiri and Sandberg (2020). Consequently, it offers the chance to evaluate whether further improvements using the GDH can be obtained in prediction of  $\eta$  when switching to a CFD-driven training approach. Thus, in the present study, the constancy of  $Pr_t$  in Equation 4 is upended in lieu of a scalar functional form obtained as a function of velocity and temperature gradients using the symbolic regression-based machine learning framework outlined below.

## The machine-learning framework

To perform the optimisation in the present study, we use gene-expression programming (GEP), which is a symbolic regression algorithm returning mathematical equations (Ferreira, 2001). The biggest advantage of using GEP over other algorithms is the portability and transparency of returning an equation, allowing for inspection of the generated terms and interpretation of the optimisation outputs. In GEP (drawing an analogy with Darwin's survival-of-the-fittest theory), an initial population of candidate solutions is randomly generated at generation step  $i = 0$ . From this pool of candidate solutions, the fitness of each candidate is evaluated based on a cost function. Subsequently, selection (survival-of-the-fittest) directs the search away from poorer candidates and weights the chance of survival towards fitter functional forms. This is achieved through tournament selection, whereby the candidate solutions compete with each other to survive based on their fitness. The genetic operators then (e.g. mutation, crossover) provide the search aspect of the algorithm, applying variations to the existing solutions, updating the population of candidate solutions at the next generation  $i + 1$ . The search then iteratively continues through several generations until a functional form that "best" fits the dataset (the optimum solution) is obtained. The functional form returned by the algorithm depends on the random initialisation when generating the population of candidate solutions at generation  $i = 0$ , i.e. it is a non-deterministic process. However, the fitness of the functional forms have been observed to be not too dissimilar, leading to similar performances when the closures are tested *a priori* and *a posteriori* (Haghiri and Sandberg, 2020). Interested readers can refer to Koza (1994) and Ferreira (2001) for introductory information of the GEP algorithm, whereas the details of using GEP in turbulence modelling was specified in detail in Weatheritt and Sandberg (2016).

### Frozen training process

The data-driven closure for the turbulent heat-flux in the frozen approach is developed by adopting the gradient-diffusion hypothesis by minimising the cost function

$$\text{CostFun}(\alpha^{\text{sep}}) = \frac{1}{V} \sum_{m \in V} \left\{ \overline{u'_i T'} + \alpha^{\text{sep}} \nu_t \frac{\partial \bar{T}}{\partial x_i} \right\}^{(m)} \left\{ \overline{u'_i T'} + \alpha^{\text{sep}} \nu_t \frac{\partial \bar{T}}{\partial x_i} \right\}^{(m)}, \tag{5}$$

which is the mean-squared error of Equation 4 where  $\alpha_t = \alpha^{\text{sep}} \nu_t$ . Note that  $V$  represents the points within the training region. The training region refers to the spatial extent of the domain from where the data points are picked to undergo the training. Therefore, the target for optimisation is a non-dimensional parameter  $\alpha^{\text{sep}}$ , which can be seen as the inverse of a *non-constant* turbulent Prandtl number with a functional dependence on the velocity and temperature gradients,

$$\alpha^{\text{sep}} = f(I^1, I^2, J^1, \dots, J^5), \tag{6}$$

where, considering the present statistically two-dimensional flow, the function  $\alpha^{\text{sep}}$  is formed by constructing independent invariants from  $s_{ij} = S_{ij}/\omega$ ,  $w_{ij} = \Omega_{ij}/\omega$ , and  $\vartheta_i = (\sqrt{k}/\beta^* \omega)(\partial \bar{T}/\partial x_i)$  given as follows:

$$\begin{aligned} I^1 &= s_{ij}s_{ji}, & I^2 &= w_{ij}w_{ji}, & J^1 &= \vartheta_i \vartheta_i, & J^2 &= \vartheta_i s_{ij} \vartheta_j, & J^3 &= \vartheta_i s_{ij} s_{jk} \vartheta_k, \\ J^4 &= \vartheta_i w_{ij} w_{jk} \vartheta_k, & J^5 &= \vartheta_i w_{ij} s_{jk} \vartheta_k. \end{aligned} \tag{7}$$

The invariants used in this study represent an integrity basis (Smith, 1965) to ensure the coefficient  $\alpha^{\text{sep}}$  remains Galilean invariant. Owing to the statistically two-dimensional nature of the flow, Equation 7 represents a subset from the full list of invariants as listed in Milani et al. (2018) and discussed in Pope (1975). In Equations 5 and 6 the turbulence eddy viscosity ( $\nu_t$ ) and the specific turbulence dissipation ( $\omega$ ) are, respectively, needed in the data-driven solutions. As such,  $\nu_t$  and  $\omega$  are obtained through a frozen approach in which the “true” (high-fidelity) velocity and Reynolds stress is held constant and only the  $\omega$  transport equation is solved, resulting in  $\nu_t = k/\omega$  consistent with both the high-fidelity database used and the turbulence model equations. Hence, apart from the frozen approach to obtain  $\nu_t$ , there is no CFD involved in the machine learning process here. A detailed investigation by Haghiri and Sandberg (2020) on different BRs and lip thicknesses found that the turbulent heat-flux closure developed for the thickest lip ( $t/s = 1.14$ ) and the highest BR of 1.26 performed extremely well on itself, other flow conditions and lip thicknesses. Thus, in this paper, we consider the BR = 1.26 case for comparison and closure development. The developed machine-learnt heat-flux closure ( $\alpha_t^{\text{mod}} = \alpha^{\text{sep}} \nu_t$ ) generated with the above described framework, using the LES case of BR = 1.26, is given as follows

$$\alpha_t^{\text{mod}} = \{6.806I_2 - 109.407J_1 + 2.0J_2 + 2.368\} \nu_t. \tag{8}$$

A box domain training region was chosen for developing this closure, where the wall normal extent ranged from  $0 \leq y \leq 3$  and the streamwise extent covered  $30 \leq x \leq 85$ . The functional form obtained here constitutes the “Frozen” model and will be used later on to compare the new closures developed via the looped training approach, described next.

### Looped training process

The looped training process discussed here is an adaptation of the method described in Zhao et al. (2020), extended to scalar regression. While the cost function in the previous section was obtained by computing the error difference between the reference and model heat-fluxes, the cost function evaluation here is performed by running RANS calculations in the loop to obtain the difference between the reference and the model. A key advantage of running RANS in the loop is the ability to modify the quantity used to evaluate the cost function. Thus, the looped training process can be used to find a turbulent heat-flux closure that improves the temperature field, instead of the turbulent heat-flux as was considered in the frozen training process. An example of the cost

function used here is given by

$$\text{CostFun}(\alpha^{\text{gEP}}) = \frac{1}{V} \sum_{m \in V} \{\bar{T}^{\text{ref}} - \bar{T}^{\text{gEP}}\}^m \{\bar{T}^{\text{ref}} - \bar{T}^{\text{gEP}}\}^m. \quad (9)$$

$\bar{T}^{\text{ref}}$  is obtained from a reference dataset while  $\bar{T}^{\text{gEP}}$  is obtained after a RANS is run with a heat-flux closure from GEP. The RANS runs the scalar transport equation (Equation 2) for the temperature field, using the velocity field and  $\nu_t$  (obtained from the frozen procedure) from the reference time-averaged LES data. The temperatures in Equation 9 could be from any spatial location, using the whole domain or select spatial positions. The latter is advantageous when the reference dataset is limited, as is often the case in experiments. The GEP algorithm produces a candidate pool of models for the heat-flux, generated randomly, which are used to run RANS calculations. Upon reaching a converged RANS solution, a cost function is evaluated (like in Equation 9) and the error returned back to the algorithm. Based on the models with the lowest returned errors, a new pool of models is generated using the evolution principles also used in the frozen training. These new models are evaluated again by running RANS calculations and the cycle repeats until the best surviving model remains, based on some stopping criterion such as the number of generations or an error threshold. This approach should generally lead to better closures compared to the frozen training because it implicitly includes the effect of other modelling errors present when solving the RANS equations. The RANS calculations are performed with OpenFOAM v2.4 which interfaces with the GEP algorithm. The scalarTransportFoam solver is modified to read the GEP-generated model form as a wordList string which the algorithm pipes to a file in a given generation. This string is used to construct the scalar expression in Equation 6 and thus solve Equation 2. At the end of the converged run, the RANS result is sampled at the spatial locations of interest and an error value using Equation 9 is evaluated. If the RANS fails to converge, the error is set to a very large number to automatically filter out unstable models. The difference between the two training approaches is summarised in Figure 3 via a schematic.

## Results and discussion

### Grid independence

Since the looped training relies on running multiple RANS calculations simultaneously and iteratively, the runtime of these RANS can require large computational resources and dramatically increase the model development time. Thus, it is crucial to have as cheap a RANS setup as possible for the training. This is attained by coarsening the grid resolution without compromising the solution accuracy. A grid independence test is performed on the frozen GEP trained case (referred to as “Frozen”), i.e. with  $\alpha_t$  given by Equation 8 in OpenFOAM v2.4. The temperature equation was solved using a smoothSolver with the GaussSeidel smoother and a tolerance of  $10^{-16}$ . A relaxation factor of 0.7 was used on the temperature equation as well as the final temperature at each iteration. The discretisation schemes used for the gradient, divergence, laplacian and interpolation operators were Gauss linear, bounded Gauss linearUpwind grad(T), Gauss linear corrected and linear respectively.

Figure 4 shows the adiabatic wall effectiveness  $\eta$  for the BR = 1.26 case tested on multiple grid resolutions while Table 2 shows the details of the grid resolutions tested. The results indicate that the coarsest grid tested, i.e. G5, shows excellent agreement with the reference LES grid G1 for the frozen case. With a runtime of  $\sim 15$  s running on a single core laptop, grid G5 is ideally placed to conduct the looped training. Additionally, the cost of the training process can be further minimised by executing the GEP algorithm for a smaller number of generations, chosen to be 100 here, running 4 RANS calculations per generation (representing the tournament size) followed by 20 mating operations per generation, resulting in a total model development time of approximately 4 h. This means that at any given generation,  $4 \times 20 = 80$  candidate solutions are being evaluated before proceeding to the next generation. It will be shown later that even with a reduced number of generations, the closures that improve the prediction can still be obtained. The number of RANS in the loop was chosen such that it matches the number of cores on the compute node. In contrast, the frozen training had been conducted for 3,000 generations, with 2 tournaments per generation followed by 50 mating operations per generation (i.e.  $2 \times 50 = 100$  candidate solution evaluations per generation), which took about 20 min. It must be pointed out that the purpose of the looped training process here is to demonstrate the effect of including RANS calculations in the optimisation process, rather than finding the best parameters to run the GEP algorithm. As a rule of thumb, increasing the number of generations, which represents the iterations, leads to a model with better fitness values. Consequently, a longer looped run was also conducted on a larger compute node with 12 cores for 400

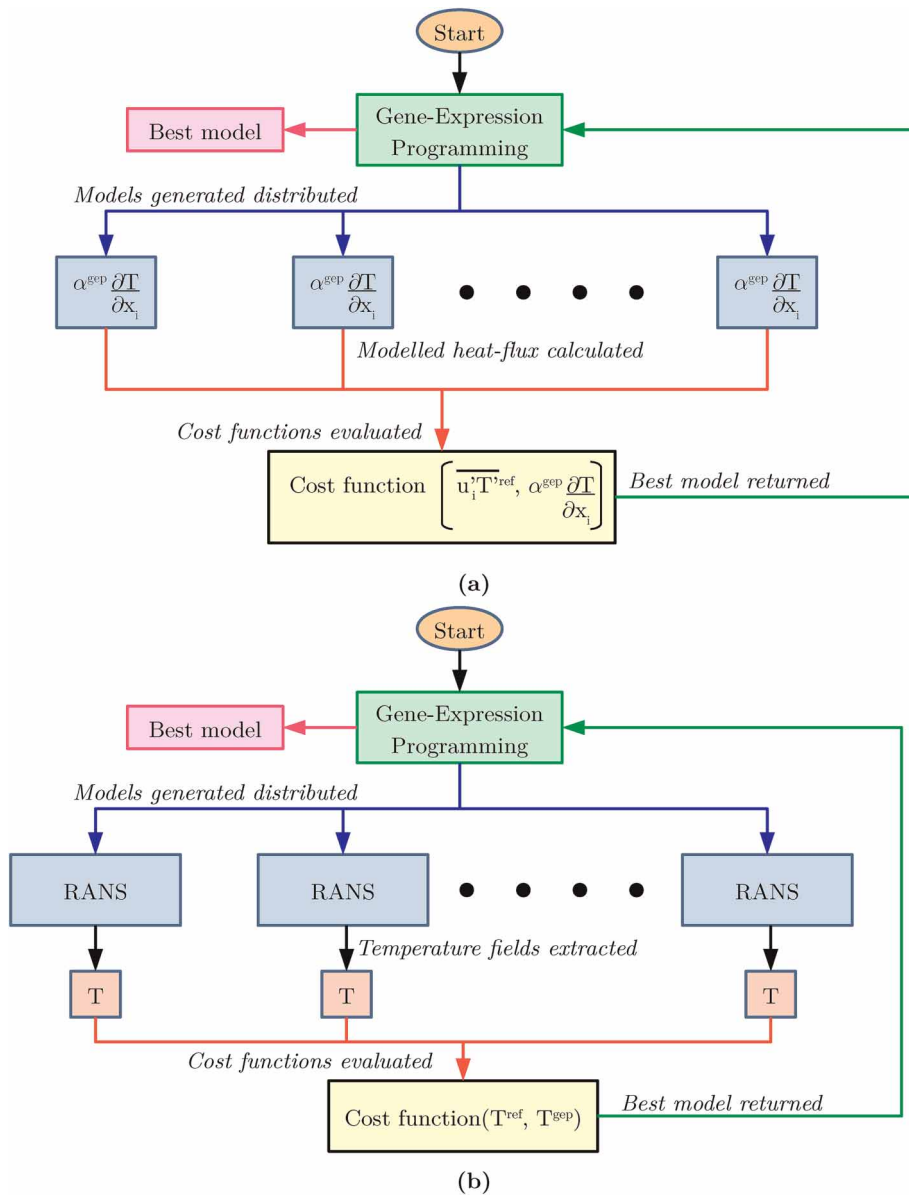


Figure 3. Schematic of the (a) frozen and (b) looped training process.

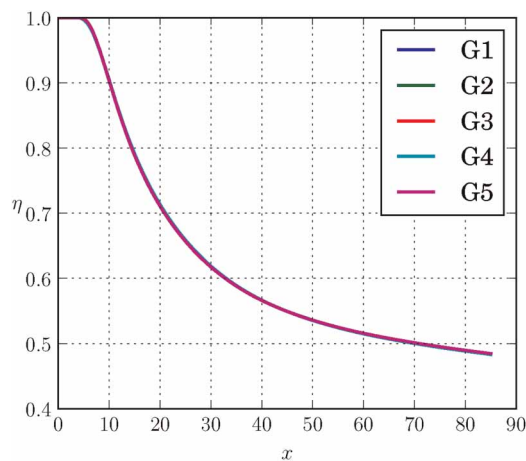


Figure 4. Grid convergence test on grids from Table 2 using the frozen closure from Equation 8 on BR = 1.26 case.

Table 2. Grid independence study.

Grid Caption	$N_x \times N_y$	Processors	Runtime (s)
G1	1681 × 386	24	750
G2	841 × 194	12	150
G3	421 × 98	6	50
G4	211 × 50	3	20
G5	106 × 26	1	15

generations, running 12 RANS calculations per generation followed by 50 mutation operations per generation, bringing the total model creation cost to approximately 30 h.

### Effect of cost function

With the availability of RANS calculations within the looped training process, a clear advantage over the frozen training is the ability to choose any quantity as a cost function to develop the closure for the heat-flux. Using the BR = 1.26 case as the reference for model development, two cost functions are tested, the temperature distribution along the wall, i.e.  $y = 0$ , and the wall-normal temperature distribution at 3 streamwise locations, i.e.  $x = 10, 30$  and  $50$ . The resulting closures obtained are given by

$$\alpha_t^{ST} = \{2.57 - I_1\}v_t, \quad (10)$$

$$\alpha_t^{WT} = \{3.15 - J_4 + J_5\}v_t, \quad (11)$$

where superscripts  $ST$  and  $WT$  refer to the closures developed using the streamwise temperature profiles and wall temperature as cost functions, respectively. Compared with Equation 8, the looped closures in Equations 10 and 11 are simpler in form and have larger values of the constants within the expression. The simpler expression forms are a feature of the looped training process and was first discussed in Zhao et al. (2020) for the turbulence stress closures while the high value of the constants indicate a requirement of an even lower  $Pr_t$  than what the frozen trained model returned to improve the prediction. Interestingly, the contribution from the invariants in Equations 10 and 11 relative to the constant term were found to be smaller in magnitude, indicating that the major contribution to the turbulent heat-flux comes from the constant term. The prediction of  $\eta$  and the temperature profiles using these closures are presented in Figure 5. The predictions from calculations of the baseline and the frozen models are also added to the figure to highlight the effect of incorporating RANS into the data-driven model development process.

It is evident from the figure that looped training leads to an improvement over frozen training using both cost functions. The use of the streamwise temperature profiles as a cost function improves the temperature prediction everywhere except for points near the wall. The wall temperature cost function performs better for points near the wall, leading to an excellent match with the  $\eta$  profile, however compromising the streamwise temperature profile prediction for  $y \in [0.5, 2]$ . Given the excellent agreement in  $\eta$  using the  $WT$  closure, the next sections will use this particular cost function for any subsequent model development.

### Generalisability

Based on the excellent improvements obtained with the wall temperature closure (Equation 11) for the BR = 1.26 case, its potential is also tested on separate flow conditions to evaluate its generalisability. Blowing ratios of 0.86 and 1.07 are run with this closure and the  $\eta$  predictions are presented in Figure 6. For the sake of consistency, the BR = 1.26 frozen closure was used to test the frozen closure's generalisability across the other BRs.

There is a slight overprediction in  $\eta$  for  $x \in [15, 40]$  for BR = 0.86 and for  $x \in [15, 35]$  for BR = 1.07, followed by slight underprediction beyond  $x = 60$  with the looped closure. However, the looped closure clearly performs better than the frozen closure for both the BRs. To further investigate improvements in  $\eta$ , a looped



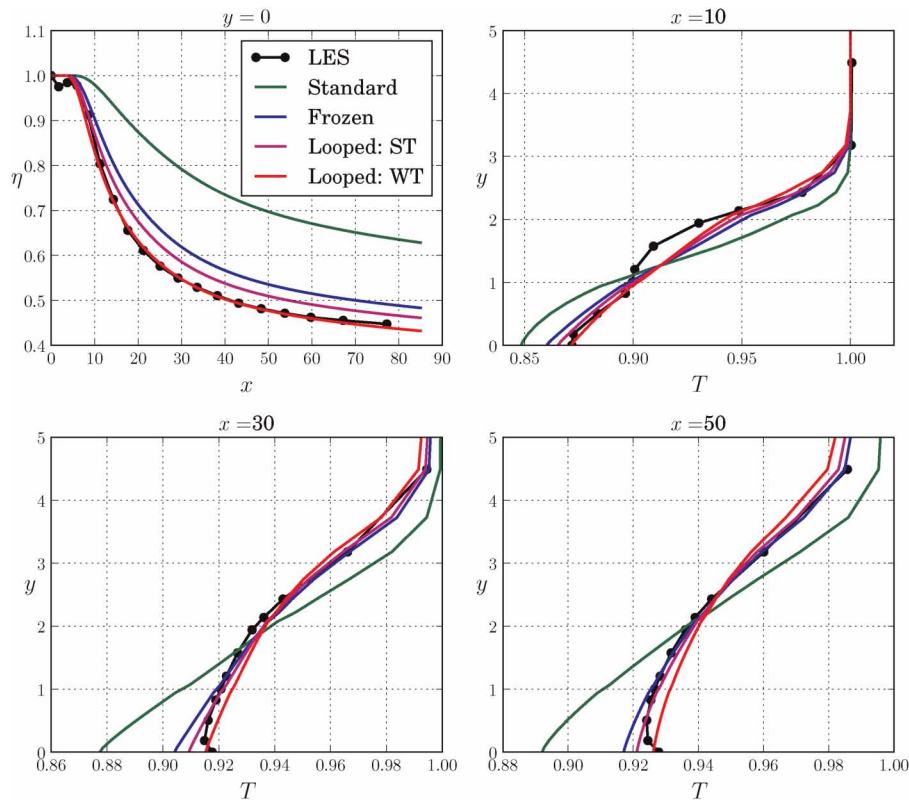


Figure 5. Comparison of predictions from standard, frozen and looped closures with LES data, trained and tested on the BR = 1.26 case.

closure development for BR = 0.86 was conducted, first using the shorter 100 generation 4 RANS run and then using a longer 400 generation 12 RANS run, resulting in the following models:

$$\alpha_t^S = \{3.0 - 2J_4\}\nu_t, \tag{12}$$

$$\alpha_t^L = \{3.43 - I_1\}\nu_t, \tag{13}$$

where the superscripts *S*, *L* correspond to the short and long runs of the looped training algorithm. The resulting  $\eta$  profiles are plotted in Figure 7 where the two models are compared with the looped BR = 1.26 closure.

The results show that the  $\alpha_t^S$  closure for BR = 0.86 (Equation 12) slightly overpredicts the  $\eta$  and thus performs less well than the BR = 1.26 closure (Equation 11). This is however not proof the algorithm could not find a better closure, just that it was not run for a long enough number of generations. Doing so does lead to a better prediction of  $\eta$ , as demonstrated by the  $\alpha_t^L$  closure (Equation 13).

### Multidata training

The ability of running RANS in the loop to develop a closure leads to an interesting possibility, i.e. developing closures for multiple datasets at once. A short run with 100 generations using 6 RANS per generation, with 2 RANS per BR and a long run with 400 generations using 36 RANS per generation, with 12 RANS per BR were performed with the resulting closures given by:

$$\alpha_t^{S,MD} = \{3.0 - I_1\}\nu_t, \tag{14}$$

$$\alpha_t^{L,MD} = \{3.383 - I_1\}\nu_t, \tag{15}$$

where the superscripts *S*, *MD* and *L*, *MD* correspond to the short and long multidata runs of the looped training algorithm. The cost function remains the same as Equation 9, where the  $T^{ref}$  for a given RANS run depends on the BR and is chosen accordingly. Effectively, all three BRs are made to compete simultaneously in a given

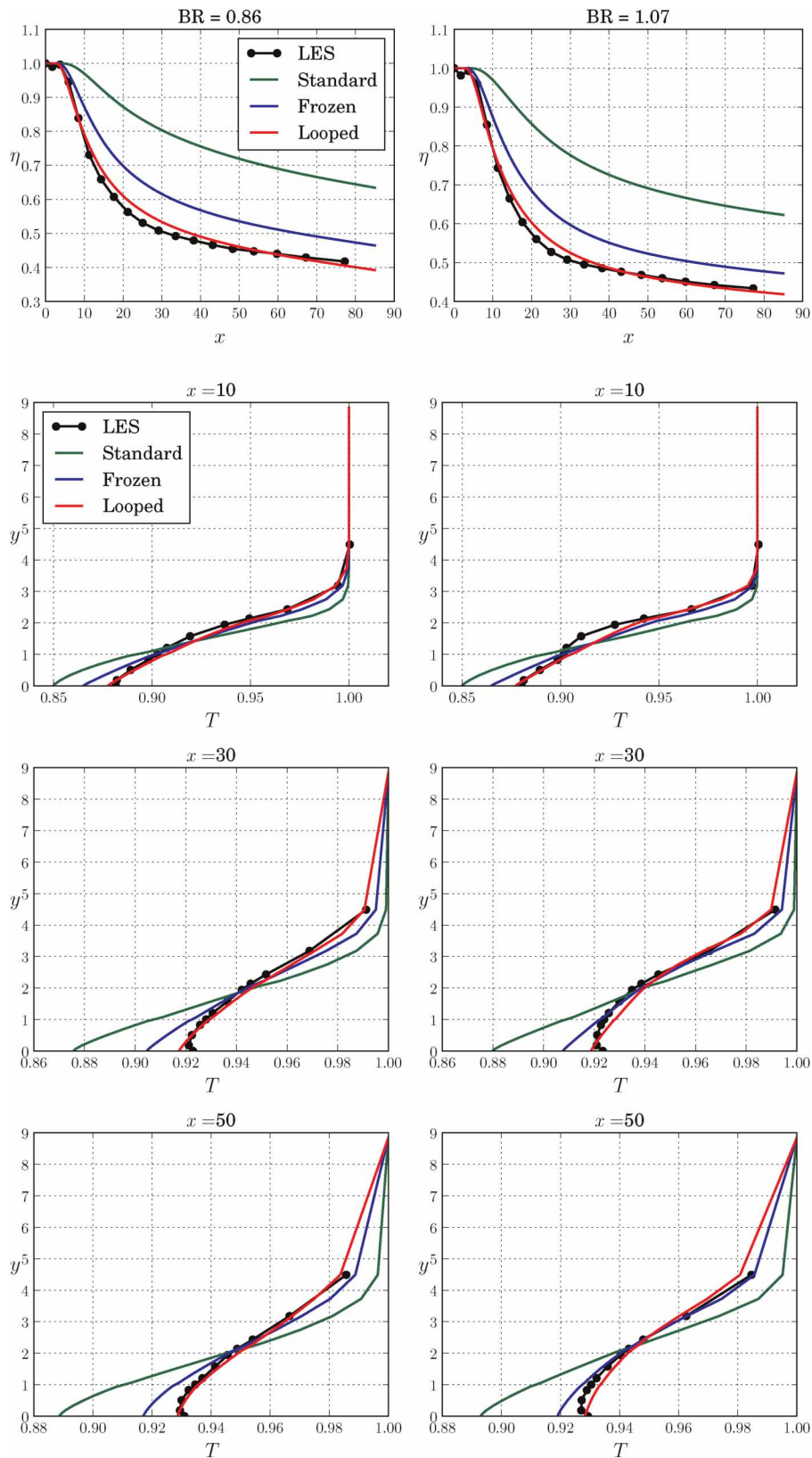


Figure 6. Comparison of predictions from standard, frozen and looped closures trained on the BR = 1.26 case and applied on the BR = 0.86 and 1.07 cases.

generation with different candidate solutions, eventually leading to the selection of the candidate with the best fitness. The predictions with these new closures are presented in Figure 8, where the short run (Equation 14) performs slightly worse than the looped BR = 1.26 closure (Equation 11) while the longer run closure (Equation

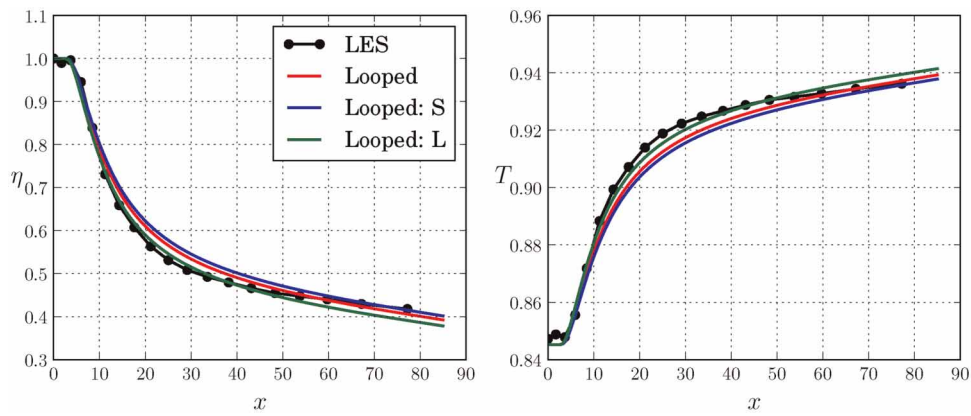


Figure 7. Comparison of the predictions from the looped closures obtained from the BR = 1.26 case (Looped), the short (Looped: S) and the long (Looped: L) looped runs from the BR = 0.86 case, applied to the BR = 0.86 case.

15) improves the prediction even further. There are locations, particularly beyond  $x = 40$  where the long run leads to an underprediction of  $\eta$  for all BRs. This directly ties into how the selection of the best model is made during the training process. Even though the training includes multiple BRs, the model with the lowest fitness across a given generation survives, which usually occurs for one BR only, in this case BR = 1.26. This leads to a bias toward a particular model during the training process geared toward one BR only. A possible rectification to this bias is part of future investigations.

Nevertheless, the results with the looped training show that the closures developed do perform better than their counterparts from frozen training, are shorter in their mathematical extent and generalise well across other flow conditions.

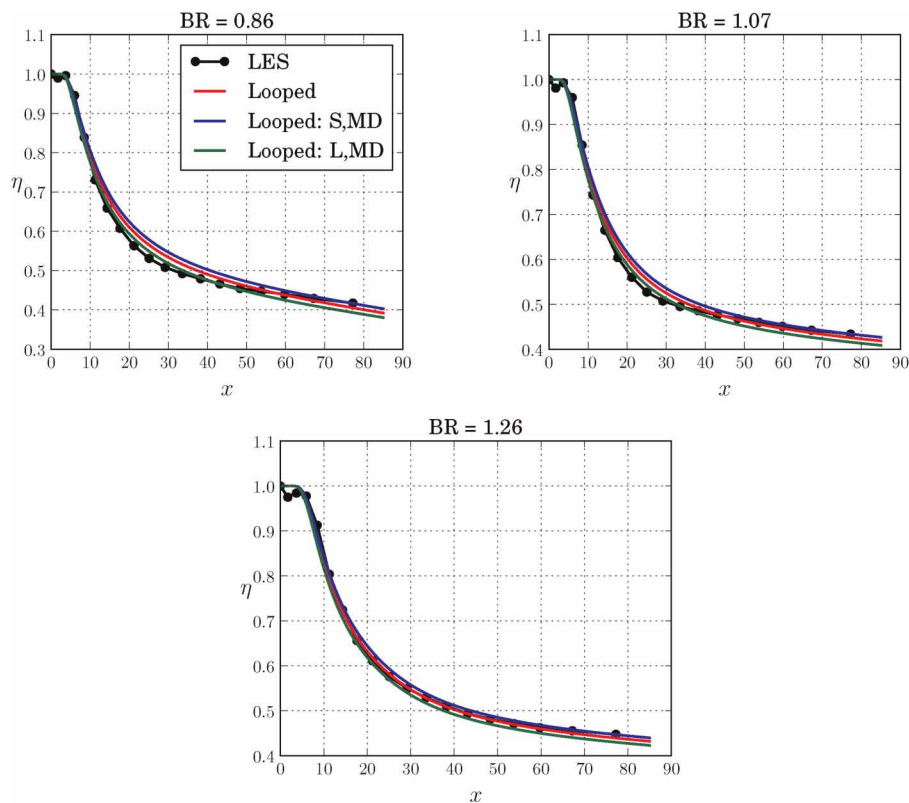


Figure 8. Comparison of the predictions from the looped closures obtained from the BR = 1.26 case (Looped) and the short (Looped: S,MD) and long (Looped: L,MD) looped multidata runs using BR = 0.86, 1.07 and 1.26 cases, and applied on the three BRs.

## Conclusions

In this study, a novel symbolic scalar regression machine-learning algorithm was applied to improve the prediction of the wall-temperature in a fundamental trailing-edge slot. The prediction of the temperature is improved by developing a scalar mathematical expression to represent a non-constant turbulent Prandtl number within the gradient diffusion hypothesis for the turbulent heat-flux. The conventional use of the machine-learning algorithm, also known as the “frozen” training process, obtained this expression by minimising the error between the true turbulent heat-flux from high-fidelity data and the model expressions generated by gene-expression programming (GEP). In the novel approach, i.e. the “looped” training process, the scalar expression is derived by conducting RANS calculations in a loop within GEP. This leads to promoting the expressions which minimise the error between the true temperature field from the high-fidelity data and the RANS solution. Thus, the closure for the turbulent heat-flux which improves the temperature field prediction can be obtained instead of the closure that improves the turbulent heat-flux prediction. This approach also allows for the use of any type of cost function and for the largest blowing ratio of 1.26, two such functions were tested, with the one based on the wall-temperature value leading to the best prediction. The resulting closure from the looped approach was also found to be mathematically succinct compared with its frozen counterpart and improved upon the prediction of the wall-temperature significantly further. The closure showed that the effective turbulent Prandtl number was required to be even smaller ( $\sim 0.32$ ) than the frozen closure’s value of 0.42 to realise improvements and generalised well upon application to the other blowing ratios. Finally, the looped approach was also demonstrated to build closures by combining datasets from multiple blowing ratios, which led to further improvements in terms of the generalisation.

## Funding sources

This work was partially funded by the Australian Research Council Linkage Projects LP160100228, LP180100712, Prof Richard Sandberg.

## Competing interests

Chitrarth Lav declares that he has no conflict of interest. Ali Haghiri declares that he has no conflict of interest. Richard Sandberg declares that he has no conflict of interest.

## References

- Abe K. and Suga K. (2001). Towards the development of a reynolds-averaged algebraic turbulent scalar-flux model. *International Journal of Heat and Fluid Flow*. 22 (1): 19–29. <http://www.sciencedirect.com/science/article/pii/S0142727X0000062X>
- Daly B. J. and Harlow F. H. (1970). Transport equations in turbulence. *Physics of Fluids*. 13 (11): 2634–2649. <https://doi.org/10.1063/1.1692845>.
- Nicoud F. and Ducros F. (1999). Subgrid-scale stress modelling based on the square of the velocity gradient tensor. *Flow, Turbul. Combust.* 62: 183–200.
- Fabris G. (1979). Conditional sampling study of the turbulent wake of a cylinder. Part 1. *Journal of Fluid Mechanics*. 94 (4): 673–709. <https://doi.org/10.1017/S0022112079001245>.
- Ferreira C. (2001). Gene expression programming: A new adaptive algorithm for solving problems. *Complex System*. 13 (2): 87–129.
- Haghiri A. and Sandberg R. D. (2020). Large eddy simulations of wall jets with coflow for the study of turbulent prandtl number variations and data-driven modeling. *Physical Review Fluids*. 5 (6): 064501. <https://doi.org/10.1103/PhysRevFluids.5.064501>.
- Holloway D. S., Leylek J. H., and Buck F. A. (2002a). Pressure-side Bleed Film Cooling: Part I - Steady Framework for Experimental and Computational Results. In *Turbomach. Tech. Conf. Expo. GT2002-30471*, ASME, Amsterdam, Netherlands, pp. 1–9.
- Holloway D. S., Leylek J. H., and Buck F. A. (2002b). Pressure-side Bleed Film Cooling: Part II – Unsteady Framework for Experimental and Computational Results. In *Turbomach. Tech. Conf. Expo. GT2002-30472*, ASME, Amsterdam, Netherlands, pp. 1–9.
- Ivanova E. and Laskowski G. (2014). LES and Hybrid RANS/LES of a Fundamental Trailing Edge Slot. In *Turbomach. Tech. Conf. Expo. GT2014-25906*, Dusseldorf, Germany, pp. 1–9.
- Joo J. and Durbin P. A. (2009). Simulation of turbine blade trailing edge cooling. *The Journal of Fluids Engineering*. 131 (2): 021102. <http://fluidsengineering.asmedigitalcollection.asme.org/article.aspx?articleid=1478218>
- Kacker S. C. and Whitelaw J. H. (1969). An experimental investigation of the influence of slot-lip-thickness on the impervious-wall effectiveness of the uniform-density, two-dimensional wall jet. *International Journal of Heat and Mass Transfer*. 12 (9): 1196–1201. [https://doi.org/10.1016/0017-9310\(69\)90129-X](https://doi.org/10.1016/0017-9310(69)90129-X).
- Kacker S. C. and Whitelaw J. H. (1971). The turbulence characteristics of two-dimensional wall-jet and wall-wake flows. *Journal of Applied Mechanics*. 38 (1): 239–252. <https://doi.org/10.1115/1.3408749>.
- Koza J. R. (1994). Genetic programming as a means for programming computers by natural selection. *Statistics and Computing*. 4 (2): 87–112. <https://doi.org/10.1007/BF00175355>.

- Launder B. E. and Rodi W. (1983). The turbulent wall jet measurements and modeling. *Annual Review of Fluid Mechanics*. 15 (1): 429–459. <https://doi.org/10.1146/annurev.fl.15.010183.002241>.
- Lav C. and Sandberg R. D. (2020). Unsteady Simulations of a Trailing-Edge Slot Using Machine-Learnt Turbulence Stress and Heat-Flux Closures. Proceedings of the ASME Turbo Expo 2020: Turbomachinery Technical Conference and Exposition. Volume 7B: Heat Transfer. Virtual, Online. September 21–25, 2020. V07BT12A012. ASME. <https://doi.org/10.1115/GT2020-14398>
- Lav C., Philip J., and Sandberg R. D. (2019a). A New Data-Driven Turbulence Model Framework for Unsteady Flows Applied to Wall-Jet and Wall-Wake Flows. Proceedings of the ASME Turbo Expo 2019: Turbomachinery Technical Conference and Exposition. Volume 2A: Turbomachinery. Phoenix, Arizona, USA. June 17–21, 2019. V02AT45A002. ASME. <https://doi.org/10.1115/GT2019-90179>
- Lav C., Sandberg R. D., and Philip J. (2019b). A framework to develop data-driven turbulence models for flows with organised unsteadiness. *Journal of Computational Physics*. 383: 148–165. <https://www.sciencedirect.com/science/article/pii/S0021999119300476?via%3DIihub#>
- Ling J., Ryan K. J., Bodart J., and Eaton J. K. (2015). Analysis of turbulent scalar flux models for a discrete hole film cooling flow. *The Journal of Turbomachinery*. 138 (1). <https://doi.org/10.1115/1.4031698>
- Medic G. and Durbin P. A. (2005). Unsteady effects on trailing edge cooling. *The Journal of Heat Transfer*. 127 (4): 388–392. <http://link.aip.org/link/JHTRAO/v127/i4/p388/s1&Agg=doi>
- Milani P. M., Ling J., Saez-Mischlich G., Bodart J., and Eaton J. K. (2018). A machine learning approach for determining the turbulent diffusivity in film cooling flows. *The Journal of Turbomachinery*. 140 (2): 021006. <https://doi.org/10.1115/1.4038275>.
- Pope S. B. (1975). A more general effective-viscosity hypothesis. *Journal of Fluid Mechanics*. 72 (2): 331–340. <https://doi.org/10.1017/S0022112075003382>.
- Sandberg R. D. (2015). Compressible-flow DNS with application to airfoil noise. *Flow, Turbulence and Combustion*. 95 (2–3): 211–229. <https://doi.org/10.1007/s10494-015-9617-1>.
- Sandberg R. D., Tan R., Weatheritt J., Ooi A., Haghiri A., et al. (2018). Applying machine learnt explicit algebraic stress and scalar flux models to a fundamental trailing edge slot. *The Journal of Turbomachinery*. 140: 1–13. <https://doi.org/10.1115/1.4041268>.
- Smith G. F. (1965). On isotropic integrity bases. *Archive for Rational Mechanics and Analysis*. 18 (4): 282–292. <https://doi.org/10.1007/BF00251667>.
- Weatheritt J. and Sandberg R. (2016). A novel evolutionary algorithm applied to algebraic modifications of the RANS stress–strain relationship. *Journal of Computational Physics*. 325: 22–37. <https://doi.org/10.1016/j.jcp.2016.08.015>.
- Weatheritt J., Zhao Y., Sandberg R. D., Mizukami S., and Tanimoto K. (2020). Data-driven scalar-flux model development with application to jet in cross flow. *International Journal of Heat and Mass Transfer*. 147: 118931. <http://www.sciencedirect.com/science/article/pii/S0017931019329187>
- Whitelaw S. and Kacker J. (1963). An Experimental investigation of the influence of slot lip thickness of the impervious wall effectiveness of the uniform density, two dimensional wall jet. *International Journal of Heat and Mass Transfer*. 542: 1196–1201. [https://doi.org/10.1016/0017-9310\(69\)90129-X](https://doi.org/10.1016/0017-9310(69)90129-X).
- Wong T., Ireland P., and Self K. (2016). Film cooling effectiveness downstream of trailing edge slots including cutback surface protuberances. *International Journal of Turbomachinery, Propulsion and Power*. 1 (1): 4. <https://doi.org/10.3390/ijtpp1010004>.
- Zhao Y., Akolekar H. D., Weatheritt J., Michelassi V., and Sandberg R. D. (2020). RANS turbulence model development using CFD-driven machine learning. *Journal of Computational Physics*. 411: 109413. <https://doi.org/10.1016/j.jcp.2020.109413>.