Black-box optimization for the design of a jet plate for impingement cooling

Lorenzo Cocchi* Filippo Marini† Margherita Porcelli‡§ Elisa Riccietti¶

February 17, 2025

Abstract

In this work, we propose a novel black-box formulation of the impingement cooling system for a nozzle in a gas turbine. Leveraging on a well-known model that correlates the design features of the cooling system with the efficiency parameters, we develop NOZZLE, a new constrained black-box optimization formulation for the jet impingement cooling design. Then we illustrate how to use derivative-free algorithms for finding a solution of NOZZLE and validate the proposed model on real-word test cases.

Keywords Cooling systems, gas turbine, simulation optimization, direct search algorithm.

1 Introduction

The improvement of the cooling system of an existing gas turbine model is a crucial issue for the turbine design and maintenance process. For the time being, a generalized and rigorous approach to this issue is not available and, in most cases, the matter is entrusted to the experience of the engineers that are working at the moment on that particular machine. Moreover, whenever a new design for the impingement insert is proposed, its expected performance has to be tested by a Computational Fluid Dynamics and Thermodynamics simulation, also for checking that any violations of engineering constraints are made. This kind of simulation often requires a huge computational effort.

The main purpose of this work is to present a fast and automatic methodology to improve the efficiency of an impingement cooling system using tools from *black-box* and *derivative-free* optimization. The term black-box optimization (BBO) refers to problems where the structure of the objective and constraint functions cannot be exploited, as it is often the case when they are obtained through numerical simulations; derivative-free optimization (DFO) refers to methods that do not use derivative information, being in fact the most suitable approach for black-box problems, see e.g. [2, 6, 7, 21, 18].

Starting from the well-known model by Florschuetz *et al.* [17, 16], we develop NOZZLE, a numerical model that simulates the functioning of an impingement cooling system in a fixed nozzle. The simulation also includes the estimation of temperature distribution both on the internal and external

*Baker Hughes, Via Felice Matteucci, 2, 50127, Firenze, Italia. Email: lorenzo.cocchi2@bakerhughes.com

†Dipartimento di Matematica, Alma Mater Studiorum - Università di Bologna, Piazza di Porta San Donato 5, 40126 Bologna, Italia. Email: filippo.marini5@unibo.it

‡Dipartimento di Ingegneria Industriale, Università degli Studi di Firenze, Viale Morgagni 40/44, 50134 Firenze, Italia. Email: margherita.porcelli@unifi.it

§ISTI–CNR, Via Moruzzi 1, Pisa, Italia

¶Univ Lyon, ENS de Lyon, UCBL, CNRS, Inria, LIP, F-69342, LYON Cedex 07, France. Email: elisa.riccietti@ens-lyon.fr

wall of the nozzle by solving a steady Heat Equation, and the estimation of the outlet pressure. Once the simulator is defined, we embed it in a BBO framework in order to optimize the design of the impingement system so that the highest possible cooling efficiency is achieved.

In more detail, we look for an insert design that maximizes the Heat Transfer Coefficient (HTC) h_c of the coolant in the feasible set $V \subset \mathbb{R}^n$ defined by engineering constraints. If we identify the main geometric variables that characterize the design of the impingement insert with the vector $\mathbf{v} \in \mathbb{R}^n$, then the optimization problem takes the form:

$$\begin{aligned} \max_{\mathbf{v}\in\mathbb{R}^n} \quad & H(\mathbf{v}) \\ \text{s.t.} \quad & \mathbf{v} \in V, \end{aligned} \tag{1}$$

where the function $H : \mathbb{R}^n \to \mathbb{R}$ is a scalar valued function that models the correlation between the geometric variables $\mathbf{v}$ and the value of the HTC h_c. Specifically, since the HTC is a non constant distribution within the cooling system, we chose H as the root mean square (RMS) of the HTC distribution.

The numerical solution of (1) poses several challenges. Indeed, the overall BBO problem is a mixed variable problem: some geometric variables are continuous and one is categorical, that is non-numeric, unconstrained and implicitly unordered. Moreover, the feasible set V is determined by black-box constraints. Taking into account the mentioned problem characteristics, various DFO solvers might be used to solve it, e.g. [2] and references therein. Here we chose to use a new flexible and robust penalized DFO approach that handles the constraints using an ℓ_1-penalty function and the Brute Force Optimizer (BFO) [19, 20] as inner solver. In our numerical experience, the proposed approach returned solutions judged reliable by practitioners and was general enough to represent a basis for further model for impingement cooling systems.

The main contributions of this work are the following:

- A new versatile BBO model for the optimization of the design of an impingement cooling system for the nozzle of a gas turbine.
- NOZZLE, a simple but still fairly accurate numerical simulator. NOZZLE has been implemented and validated and can be used by the scientific community as a test case for any kind of BBO method. A standalone Matlab version of NOZZLE is available in the S2PMJ [11] format on `GitHub` page `https://github.com/GrattonToint/S2MPJ/blob/main/matlab_problems/NOZZLEfp.m`.
- A general DFO approach that, coupled with the use of our black-box function, defines an automatic and reliable procedure for the optimization of the efficiency of a cooling system in a gas turbine.

The paper is organized as follows. In Section 2 we develop the NOZZLE, a model for optimizing the efficiency of an impingement cooling system in a nozzle as a constrained BBO problem. In details, in Sections 2.1 and 2.2 the geometric variables $\mathbf{v}$, the function H and the inequalities characterizing the feasible set V in problem (1) are defined, while in Section 2.3 the construction of the function H and of the constraints as black-box functions is described.

A DFO approach for solving the NOZZLE is proposed in Section 3. Finally, we numerically illustrate in Section 4 that our strategy allows to automatically find an improved design for the cooling system taking into account the main engineering requirements.

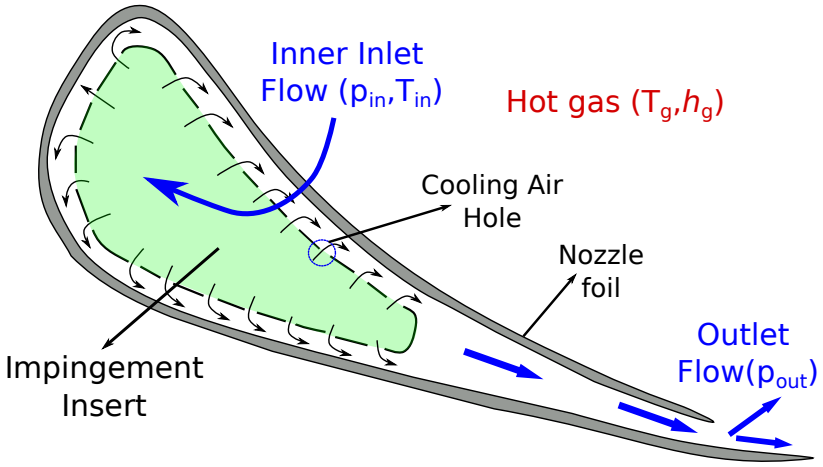

Figure 1: Section of an impingement cooling system of a nozzle

2 NOZZLE: an optimization model for the impingement cooling system

The cooling system of a gas turbine nozzle is broadly structured as shown in Figure 1. The nozzle is surrounded by hot gas, characterized by a temperature T_g and a HTC h_g, coming from the combustion chamber. Inside, a duct, called the impingement insert, takes place. The coolant fluid flows inside the insert at pressure p_c^{in} and temperature T_c and then exits the insert through orifices, hits the inner wall of the nozzle plate and finally exits the nozzle through a gap at the tail of the nozzle with pressure p_c^{out}.

When the cool fluid impinges on the inner wall of the nozzle, there is a heat exchange between the surface and the fluid, whereby the cool air subtracts heat from the nozzle wall that has been heated by the hot gas on the outside. Because of the thermal conductivity of the wall, the external wall of the nozzle can be then cooled by subtracting heat from inside the nozzle. In this way, the damage caused to the nozzle by the high temperature of the surrounding external gas is reduced.

The main component of an impingement cooling system is the impingement insert; in particular its efficiency depends on the position of the insert inside the nozzle and on the size and disposition of the orifices on its surface.

A fluid that is often used in turbine cooling systems is air, which is drawn in from the surrounding environment. Most of this flow is used in the fuel combustion process, while a portion is diverted into the cooling system. Throughout our discussion we assume that the cooling fluid used is air.

The air employed in the cooling system does not actively contribute to work generation by the gas turbine engine. Moreover, coolant ejection in the main flow can generate secondary flows and mixing losses which may reduce the aerodynamic efficiency of the airfoil [12]. This evidence demonstrates the need to maximize the efficiency of cooling systems, i.e., obtain the desired cooling effect using the minimum coolant mass flow rate.

2.1 The objective function

A major parameter for evaluating heat transfer coefficients is the nondimensional *Nusselt number* Nu, which is the ratio of the heat flux exchanged by convection to that exchanged by conductivity, in this way the measurement of the HTC is related only to the properties of the cooling air. The number Nu

is related to the coolant HTC h_c by the following relation:

$$\mathrm{Nu} = \frac{h_c d}{k_c}; \tag{2}$$

where the constant k_c is the thermal conductivity of the cooling air and d is the diameter of the orifices through which the coolant flow occurs. Therefore, once we get the distribution of Nu, h_c can be obtained by inverting (2), giving

$$h_c = \frac{k_c \mathrm{Nu}}{d}. \tag{3}$$

Therefore, the core of the objective function H is a model that correlates the geometric parameters and (the distribution of) the Nusselt number within the cooling system.

Given the wide use of impingement cooling systems, many mathematical models have been developed over time to describe their functioning and study the correlations between design features and performance. An extensive collection of impingement heat transfer correlations can be found in the work by Zuckerman and Lior [26].

The mathematical model we choose to build the objective function H is the experimental correlation developed by L. W. Florschuetz *et al.* [15, 16, 17]. Such correlation is a relatively simple model, and although it is not able to provide the level of detail of a more complex modeling approach (e.g. CFD) it turns out to be sufficiently accurate because the component to be cooled down is a metal wall. Thus the thermal conductivity of the material smooths down most of the nonuniformities of the heat transfer distribution provided by the internal cooling system: as so, the most significant parameter to be considered for the optimization is the area averaged heat transfer coefficient value over the whole heat transfer surface, and thus a high level of detail is not required for this analysis. In this framework, heat transfer correlations are the most suitable model to be employed, since they are usually capable of providing very robust outcomes (accuracy within 10-20%), given that they are applied in the proper validity range. At the same time, it is not granted that more detailed models are capable to significantly improve the prediction of such parameter, thus nullifying their advantage in terms of spatial resolution. Moreover, it was developed for an array of orifices placed on a single plate, which is the configuration closest to that of our interest for the design of the impingement insert; in fact, the insert is made from a metal plate that is drilled following the desired layout and then it is bent to obtain the final shape (see Figure 1). In this case, referring to a plate to define a model for a bent geometry does not cause a significant loss of accuracy. This is because, in actual gas turbine nozzles cooled via jet impingement, the curvature radius of the outer surface is usually at least one order of magnitude larger than the gap between the perforated wall and the target surface. As a consequence, the curved surface can be suitably represented by a flat wall, since the discrepancy introduced by this approximation is considered to be lower than the correlation accuracy (usually 10-20%).

Thus, the correlation by Florschuetz represents a good trade-off between low computational costs and meaningful modeling of the impingement cooling system. This fact makes it suitable for its use in an optimization procedure that requires the baseline model to be applied a large number of times. On the other hand, using more complex models would likely exponentially increase the computational cost of the overall procedure, making it useless for a practical application.

2.1.1 Problem geometry and variables

Let us now introduce the geometry of the cooling system referred to in Florschuetz's work. We refer to [16, 17] for further details.

The geometry of the impingement cooling system studied by Florschuetz is schematically depicted in Figure 2 and consists of a plate of jets of size $L_x \times L_y$ placed at a distance z_n from the target surface. We have set up the reference system as in Figure 2, i.e. such that the cooling air, once it leaves the jets, flows out of the duct made by the impingement plate and the target surface in the direction of the x-axis in our reference system. Because of this the x-direction is called stream-wise, while the $y-$direction is called span-wise.

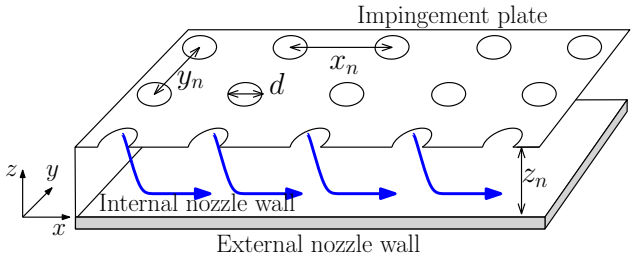

Figure 2: Reference geometry of the impingement cooling system

On the plate, round orifices of diameter d are arranged to have distance between centers x_n along the direction of the abscissa and y_n along the direction of the ordinate. The distances to the edges of the first row are also imposed as $\frac{x_n}{2}$ in the x direction and $\frac{y_n}{2}$ in the y direction (see Figure 3).

Concerning the direction stream-wise, given two points $A(x_A, y_A)$ and $B(x_B, y_B)$ on the plate, A is said to be 'upstream of' B if $x_A < x_B$ and at the same time B is said to be 'downstream of' A.

Holes could be arranged in two different ways on the plate: inline or staggered. In both cases we have $N_x := \lfloor \frac{L_x}{x_n} \rfloor$ span-wise rows each containing $N_y := \lfloor \frac{L_y}{y_n} \rfloor$ orifices. In the inline layout the centers of the orifices on the same span-wise row have the same $x-$coordinate and the ones on the same stream-wise row have the same $y-$coordinate (see Figure 3, left). Staggered layout derives from inline layout by shifting by $\frac{y_n}{2}$ the span-wise rows of even position, counting from upstream (see Figure 3, right).

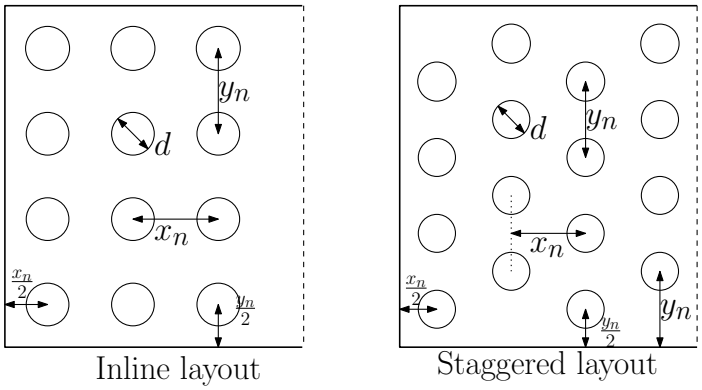

Figure 3: The two possible layouts of the holes on the jet plate: inline (left) and staggered (right).

We are now ready to define the design variables as components of the input vector $\mathbf{v}$ of our objective function H. These variables are

- d: the diameter of the impingement holes;
- x_n: stream-wise distance between the centers jet holes;

- y_n: span-wise distance between the centers jet holes;
- z_n: distance of the impingement plate from the target surface (meatus width);
- *layout*: specifies the hole pattern.

We must notice that, while d, x_n, y_n and z_n are positive real continuous variables, *layout* is a non-ordinal categorical variable that can take two values: `inline` and `staggered` [1].

2.1.2 Florschuetz correlation

The correlation between the design variables $\mathbf{v} = (x_n, y_n, z_n, d, layout)$ of the impingement plate introduced in the previous section and the stream-wise distribution of the Nusselt number Nu is defined by the following equation:

$$\mathrm{Nu}(x_i) = A\mathrm{Re}_j^{\alpha}(x_i)\mathrm{Pr}_c^{\frac{1}{3}}\left(1 - B\left(\frac{z_n}{d}\left[\frac{G_c(x_i)}{G_j(x_i)}\right]\right)^{\beta}\right), \quad \text{for } x_i = x_n\left(i - \frac{1}{2}\right), \text{ with } i = 1, ..., N_x; \quad (4)$$

where x_i is the x-coordinate of the centers of the holes of the i-th stream-wise row and Pr_c denotes the Prandtl number of the coolant; the coefficients A, α, B e β depend on the geometric parameters x_n, y_n, z_n, d and *layout* according to the following relationship:

$$\star(x_n, y_n, z_n, d) = C_\star\left(\frac{x_n}{d}\right)^{\gamma_{\star x}}\left(\frac{y_n}{d}\right)^{\gamma_{\star y}}\left(\frac{z_n}{d}\right)^{\gamma_{\star z}}, \quad \text{with } \star \in \{A, \alpha, B, \beta\}; \quad (5)$$

where the constants $C_\star$, $\gamma_{\star x}$, $\gamma_{\star y}$ e $\gamma_{\star z}$ were estimated empirically and are displayed in Table 1 (see [16]). Note that the values of the constants presented in the table differ depending on the value taken by the categorical variable *layout* (i.e. `inline` or `staggered`).

	Inline pattern				Staggered pattern			
$\star$	$C_\star$	$\gamma_{\star x}$	$\gamma_{\star y}$	$\gamma_{\star z}$	$C_\star$	$\gamma_{\star x}$	$\gamma_{\star y}$	$\gamma_{\star z}$
A	1.18	-0.944	-0.642	0.169	1.87	-0.771	-0.999	-0.257
α	0.612	0.059	0.032	-0.022	0.571	0.028	0.092	0.039
B	0.437	-0.095	-0.219	0.275	1.03	-0.243	-0.307	0.059
β	0.092	-0.005	0.599	1.04	0.442	0.098	-0.003	0.304

Table 1: Coefficients A, α, B e β for (5)

The *layout* variable is also important in defining the feasible set V and this issue is covered in Section 2.2.

Equation (4) involves the quantities $G_j(x_i)$, $G_c(x_i)$ and $\mathrm{Re}_j(x_i)$ distributed along the x-coordinate and dependent on the geometric variables.

- $G_j(x_i)$ is the mass velocity (unit: $kg \cdot m^{-2} \cdot s^{-1}$) of the flow of cooling air passing through a single jet of abscissa x_i referred to the area of the jet hole. In particular, let us consider that the cooling system receives a certain flow rate $\dot{m}_{tot}$ (unit: $kg \cdot s^{-1}$) of cooling air that is distributed

[1] We remark that in the current problem formulation, the *layout* variable can take two values and therefore it could be treated as a binary variable. For the sake of generality, we prefer to treat it as a categorical variable as it is of engineering interest to investigate models that admit orifice arrangements other than inline and staggered (see Remark 2.1).

among the span-wise rows of jets. If a row having abscissa x_i and N_y jets of area A_j has a mass flow rate $\dot{m}_j$, then we have that

$$\dot{m}_j(x_i) = G_j(x_i) A_j N_y \;\Rightarrow\; G_j(x_i) = \frac{\dot{m}_j(x_i)}{A_j N_y}, \quad x_i = x_n \left(i - \frac{1}{2}\right), \text{ for } i = 1, ..., N_x; \tag{6}$$

moreover, G_j is considered constant along the dimension y, so each span-wise row is characterized by a single value of G_j.

- G_c is the crossflow mass velocity and it is thus related to the area of the cross section of the duct, given by the product $z_n L_y$. Therefore, if we have a transverse flow rate $\dot{m}_c$ at $x-$coordinate x_i of a certain row of jets we have

$$\dot{m}_c(x_i) = G_c(x_i) L_y z_n \;\Rightarrow\; G_c(x_i) = \frac{\dot{m}_c(x_i)}{L_y z_n}, \quad x_i = x_n \left(i - \frac{1}{2}\right), \text{ for } i = 1, ..., N_x. \tag{7}$$

  We assume that G_c is constant along $y-$direction.

- $\mathrm{Re}_j(x_i)$ is the distribution of the Reynolds number of the cooling air flowing through an orifice in position x_i. The Reynolds number is the nondimensional ratio between inertia forces and internal viscous forces of a fluid, and it is related with the jet mass velocity $G_j(x_i)$ by the relation

$$\mathrm{Re}_j(x_i) = \frac{G_j(x_i) d}{\mu_c}, \quad x_i = x_n \left(i - \frac{1}{2}\right), \text{ for } i = 1, ..., N_x; \tag{8}$$

  where μ_c is the dynamic viscosity coefficient of the cooling air and d the diameter of the jet hole. The coefficient μ_c depends on the coolant's characteristics and mainly on its temperature T_c and it can be estimated via interpolation, since the cooling fluid is air and it is known the behaviour of μ_c with respect to temperature.

Remark that in (4) the distribution of the Nusselt number depends on the ratio

$$\left[\frac{G_c}{G_j}\right](x_i) := \frac{G_c(x_i)}{G_j(x_i)}, \quad x_i = x_n \left(i - \frac{1}{2}\right), \text{ for } i = 1, ..., N_x. \tag{9}$$

Therefore, in order to determine the Nusselt number distribution with (4) it necessary to estimate the distributions of G_j, G_c and Re_j.

The distribution of G_j can be derived directly from a mathematical model developed by Florschuetz [16] for the stream-wise distribution of the ratio of the jet mass velocity G_j to the average jet mass velocity $\overline{G}_j$ (see [16])

$$\left[\frac{G_j}{\overline{G}_j}\right](x_i) = \frac{\delta N_x \cosh\left(\delta \frac{x_i}{x_n}\right)}{\sinh\left(\delta N_x\right)}, \quad x_i = x_n \left(i - \frac{1}{2}\right), \text{ for } i = 1, ..., N_x; \tag{10}$$

where δ is a constant defined by

$$\delta = \frac{C_D \sqrt{2}}{y_n z_n} \frac{\pi d^2}{4} = \frac{C_D \sqrt{2} A_j}{y_n z_n};$$

with C_D being the discharge coefficient for every hole.

Note that $\overline{G_j}$ is constant and depends only on the total flow rate of cooling air $\dot{m}_{tot}$ supplied to the system and the sum of the areas of all jets on the plate A_j^{tot} according to the relationship:

$$\overline{G}_j = \frac{\dot{m}_{tot}}{A_j^{tot}}; \tag{11}$$

where $A_j^{tot} = N_x N_y A_j$.

By substituting (11) in (10) we obtain the following analytical model for the distribution of the jet mass velocity:

$$G_j(x_i) = \frac{\dot{m}_{tot}}{A_j^{tot}} \frac{\delta N_x \cosh\left(\delta \frac{x_i}{x_n}\right)}{\sinh\left(\delta N_x\right)}, \quad x_i = x_n \left(i - \frac{1}{2}\right), \text{ for } i = 1, ..., N_x. \tag{12}$$

To determine the mass velocity distribution of the transverse flow, one must refer to the geometry in Figure 2. In this representation, the duct between the impingement plate and the target surface is closed at one end; this implies that the crossflow has only one direction. Thus it is reasonable to assume that the crossflow at a point of abscissa x_i is due to the contribution of the flows passing through all the jets upstream from x_i; in particular, we can assume that the crossflow mass velocity at x_i is due to the sum of all the jet mass velocities coming from upstream, so for G_c we have

$$G_c(x_1) = 0, \quad G_c(x_i) = \frac{\dot{m}_c(x_i)}{L_y z_n} = \frac{1}{L_y z_n} \sum_{k=1}^{i-1} \dot{m}_j(x_k) = \frac{A_j N_y}{L_y z_n} \sum_{k=1}^{i-1} G_j(x_k), \text{ for } i = 2, ..., N_x. \tag{13}$$

From the equations (12) and (13), we can estimate the distribution of the ratio G_c/G_j following the simple procedure shown in Algorithm 1.

Algorithm 1 Estimation of the distribution of G_c/G_j

Initialization

Input: the variables $\mathbf{v} = (x_n, y_n, z_n, d, layout)$ and the parameters L_x, L_y, $\dot{m}_{tot}$, C_D.

Compute $N_x = \lfloor \frac{L_x}{x_n} \rfloor$, $N_y = \lfloor \frac{L_y}{y_n} \rfloor$ and $A_j = \frac{\pi d^2}{4}$.

Define the vector $x_j = \frac{1}{2}x_n : x_n : (N_x - \frac{1}{2})x_n$ of the x-coordinate of the centers of the span-wise rows.

Jet mass velocity distribution

Obtain the distribution $G_j(x_i)$ for $i = 1, ..., N_x$ with (12).

Crossflow mass velocity distribution

Set $G_c(x_1) = 0$, because there is no flow coming from upstream.

for $i = 2, ..., N_x$ **do**

Using (13) obtain crossflow mass velocity

$$G_c(x_i) = \frac{1}{L_y z_n} \sum_{k=1}^{i-1} \dot{m}_j(x_k).$$

end for

Evaluation of the ratio $\frac{G_c}{G_j}$

for $i = 1, ..., N_x$ **do**

Set

$$\left[\frac{G_c}{G_j}\right](x_i) = \frac{G_c(x_i)}{G_j(x_i)}$$

end for

To finally determine the distribution of Nu we still need to derive the Prandtl number Pr_c and the dynamic viscosity μ_c of the refrigerant fluid. Moreover, in order to use (3) to get the distribution of h_c we must estimate the thermal conductivity k_c of the cooling air. These three coefficients depend mainly on temperature T_c and fluid composition and can be easily estimated by nonlinear interpolation.

The overall procedure for determining the stream-wise distribution of HTC h_c is outlined in Algo-

rithm 2.

Algorithm 2 HTC stream-wise distribution

Initialization

1: Input: the variables $\mathbf{v} = (x_n, y_n, z_n, d, layout)$ and the parameters L_x, L_y, T_c, T_g, h_g, $\dot{m}_{tot}$, C_D.
2: Compute $N_x = \lfloor \frac{L_x}{x_n} \rfloor$, $N_y = \lfloor \frac{L_y}{y_n} \rfloor$ and $A_j = \frac{\pi d^2}{4}$.
3: Define the vector $x_j = \frac{1}{2}x_n : x_n : (N_x - \frac{1}{2})x_n$ of the x-coordinate of the centers of the span-wise rows.
4: Calculate by nonlinear interpolation of $\mu_c := \mu_c(T_c)$, $\mathrm{Pr}_c := \mathrm{Pr}_c(T_c)$ and $k_c := k_c(T_c)$ of the coolant fluid.

Mass velocity distributions

5: Estimate the distributions of G_j and $\frac{G_c}{G_j}$ using Algorithm 1.

Jet Reynolds number distribution

6: Compute the distributions of jet Reynolds number $Re_j(x_i)$ using (8).

HTC h_c distribution

7: Compute the stream-wise distribution of the Nusselt number Nu using (4) and (5).
8: Compute the distribution of h_c with (3):

$$h_c(x_i) = \frac{\mathrm{Nu}(x_i)k_c}{d}, \quad i = 1, ..., N_x.$$

Algorithm 2 is straightforward; after receiving the input $\mathbf{v} = (x_n, y_n, z_n, d, layout)$ and the parameters derived from the boundary conditions, the number N_x of span-wise rows, the number N_y of holes for every span-wise row and the area of a single hole A_j are computed; then the vector containing the $x-$coordinate of the centers of the holes on the span-wise rows is computed (Step 1 - Step 3). In Step 4 the inlet temperature T_c of the cooling air is used as a query point to interpolate the dynamic viscosity coefficient μ_c, the Prandtl number Pr_c and the thermal conductivity k_c. Step 5 computes the distribution of the jet mass velocity G_j and of the ratio $\frac{G_c}{G_j}$ via Algorithm 1. Step 6 computes the distribution of the jet Reynolds number $Re_j(x_i)$ using (8). Finally, in Step 7 and Step 8 the algorithm uses Florschuetz's model (4) to get the distribution of the Nusselt number Nu and uses (3) to estimate the distribution h_c of the HTC of the cooling air.

Let us remark that Algorithm 2 returns a one-dimensional distribution of h_c, so $h_c(\mathbf{v}) \in \mathbb{R}^{N_x}$. In order to define the scalar objective function $H(\mathbf{v})$ we use the root mean square (RMS) of $h_c(\mathbf{v})$, then

$$H(\mathbf{v}) := (h_c(\mathbf{v}))_{RMS} = \frac{\|h_c(\mathbf{v})\|_2}{\sqrt{N_x}}. \tag{14}$$

2.2 The constraints

The design of a cooling system, like many other industrial applications, is subject to constraints that arise from the need to have solutions that are actually applicable in a real-world context or at least retain a minimum of relevance to the physics of the problem we are solving.

In our setting, the constraints arise from several requirements. First, the cooling system must be efficient enough to ensure a minimal durability of the nozzle, this means that the design of the impingement insert must avoid configurations that can cause excessive damage to the nozzle walls. Secondly, it is necessary that the system is actually manufacturable so, for example, the solution to the problem cannot lead to an impingement plate with holes that are too small or too close. Finally, since our objective function is derived from the empirically developed mathematical model (4), the variables must be constrained in a space in which the model has been validated; this is because outside that space the validity of the model is not guaranteed. All these requirements are represented by a set of constraint functions that involves the design variables $\mathbf{v} = (x_n, y_n, z_n, d, layout)$, the distributions of

the temperatures on the internal and external nozzle walls and the (outlet) pressure of the cooling air.

In this section, the constraints necessary for the final formulation of the problem are defined and explained.

2.2.1 Temperature constraints

To get an idea of the efficiency of the cooling system, we need to quantify how much it can cool the inner and outer nozzle wall, so we need to estimate the stream-wise distributions of the internal and external nozzle wall temperature, which we denote as T_{wi} and T_{we}, respectively.

To do this, we can assume that heat transfer occurs from the external of the nozzle, where we have the hot gas with temperature T_g and HTC h_g, to the inside of the nozzle where we have the cool air with temperature T_c and HTC h_c. We can further assume that the heat transfer from the external to the internal of the nozzle consists of three phases. In the first stage, heat from the external gas is transferred by convection to the external wall of the nozzle, then heat is transferred to the inside of the nozzle by thermal conductivity from the external wall to the inner one (not considering the thermal conductivity that occurs perpendicularly to this direction), and finally heat is transferred from the inside wall to the cooling fluid again by convection. This process is represented in Figure 4.

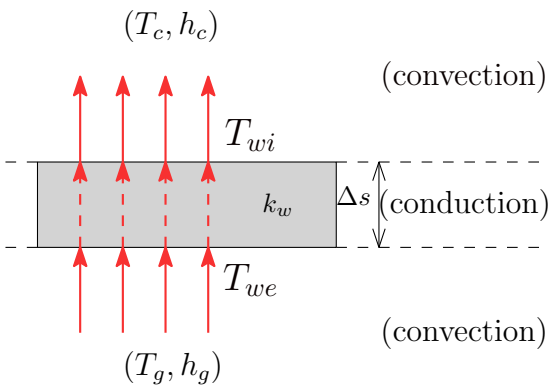

Figure 4: Scheme of the heat transfer through the nozzle wall.

Given these assumptions, and having derived the heat transfer coefficient of the cooling air, we can calculate the distributions of T_{wi} and T_{we} by solving for every $i = 1, ..., N_x$ the following linear system

$$\begin{cases} h_c(x_i)\left(T_{wi}(x_i) - T_c\right) = \frac{k_w}{\Delta s}\left(T_{we}(x_i) - T_{wi}(x_i)\right) \\ \frac{k_w}{\Delta s}\left(T_{we}(x_i) - T_{wi}(x_i)\right) = h_g\left(T_g - T_{we}(x_i)\right) \end{cases} ; \tag{15}$$

where k_w and Δs are the respectively the thermal conductivity and the thickness of the nozzle wall. This formulation comes from the time-independent heat equation

$$\begin{cases} \nabla^2 T_{in} = 0 & \text{on } \Omega, \\ \frac{\partial T_{in}}{\partial \mathbf{n}} = q_g & \text{on } \Gamma_g, \\ \frac{\partial T_{in}}{\partial \mathbf{n}} = q_c & \text{on } \Gamma_c, \\ \frac{\partial T_{in}}{\partial \mathbf{n}} = 0 & \text{on } \partial\Omega \setminus (\Gamma_g \cup \Gamma_c)\,. \end{cases} \tag{16}$$

Here T_{in} is the temperature distribution inside the nozzle wall, the domain Ω is the rectangular section of the nozzle wall with size $L_x \times \Delta s$, the boundaries Γ_g and Γ_c are the surfaces subject to convective heat flow of the hot gas and cooling air respectively. The remaining boundary of Ω is supposed to

be adiabatic [9]. If we discretize Ω in N_x rectangular elements of size $x_n \times \Delta s$ and we use the Finite Differences (FD) method assuming that there is no heat flow between two contiguous elements we obtain the system (15).

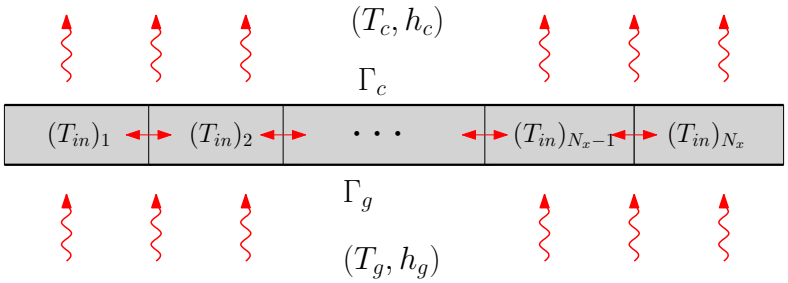

Figure 5: Discretization of the heat transfer on the nozzle wall.

A more accurate, but computationally more expensive, estimation of T_{wi} and T_{we} can be obtained in two steps. First we solve (16) with FD taking into account the heat transfer between adjacent elements in order to obtain the one dimensional distribution of T_{in} which is the stream-wise temperature distributions inside the nozzle wall (see Figure 5); then we substitute the distribution of T_{in} in system (15), in particular we put T_{in} in place of T_{we} in the first equation and in place of T_{wi} in the second equation, obtaining two separate equations:

$$h_c(x_i)\left(T_{wi}(x_i) - T_c\right) = \frac{k_w}{\Delta s}\left(T_{in}(x_i) - T_{wi}(x_i)\right) \tag{17}$$

$$\frac{k_w}{\Delta s}\left(T_{we}(x_i) - T_{in}(x_i)\right) = h_g\left(T_g - T_{we}(x_i)\right). \tag{18}$$

We use the distributions T_{wi} and T_{we} to define two constraint functions. The first one means to bound the external wall temperature T_{we} and it is defined as

$$\left(T_{we}(\mathbf{v})\right)_{RMS} \leq T_{we}^{\max}. \tag{19}$$

By setting this constraint we guarantee that the mechanical properties of the material are sufficient to allow the component to reach the expected life span.

The second constraint function is defined to bound the temperature gradient between external and internal wall; in our case it means to set a bound for the distribution of the difference between T_{we} and T_{wi}, thus

$$\left(\Delta T(\mathbf{v})\right)_{RMS} \leq \Delta T^{\max}; \tag{20}$$

where $\Delta T(\mathbf{v}) = T_{we}(\mathbf{v}) - T_{wi}(\mathbf{v})$. This constraint is necessary to prevent structural damage to the nozzle wall caused by thermal deformation due to an excessive difference between the external and internal temperatures on the wall.

2.2.2 Pressure constraints

Another factor that affects the performance of an impingement cooling system there is the pressure of the cooling air. In particular in our case we consider the ratio of the outlet pressure p_c^{out} to the inlet pressure p_c^{in} of the cooling air

$$r_p := \frac{p_c^{out}}{p_c^{in}}. \tag{21}$$

In our case $r_p \in (0, 1]$. This is because, since fluids move in the opposite direction of the pressure gradient, to have air flow there must be a pressure difference. If $p_c^{in} = p_c^{out}$, that is, if $r_p = 1$, we have no flow of cooling air through the system. It is not possible to have the case $p_c^{out} > p_c^{in}$, which means $r_p > 1$, because if so the flow would be in the opposite direction, and this is ruled out by the way we have defined the model that simulates the impingement cooling. On the other hand $r_p > 0$ since pressure is strictly positive by definition.

The ratio r_p is related to the flow rate of air through the cooling system and thus its velocity. The lower r_p, the higher the mass flow rate and the jet velocity. This is true until the velocity approaches the speed of sound. In particular, if the flow becomes sonic in the nozzle the so-called choked condition is reached, corresponding to the maximum flow rate: further reducing the discharge pressure does not lead to an increase in coolant flow rate but results in an underexpanded supersonic jet. In this regime, complex shock patterns and a recirculation pattern at the stagnation point occur, resulting in a degradation of heat transfer performance [30].

In order to keep the ratio r_p away from zero we set the following constraint

$$r_p \geq r_{\max}; \tag{22}$$

with $r_{\max} \in (0, 1)$. Recall that in the case of our interest the inlet pressure p_c^{in} of the cooling air is constant and is given as a boundary condition, while the outlet pressure p_c^{out} is unknown, hence to check constraint (22) it is necessary to estimate p_c^{out} with respect to the design variables vector $\mathbf{v}$. We assume that $p_c^{out}(\mathbf{v})$ is equal to the outlet pressure at the most downstream row of orifices, i.e. the row with centers of abscissa x_{N_x}. Focusing on this last row, from the theory of isentropic flow the pressure ratio r_p is related to the mass flow rate of cooling air through the last row $\dot{m}_j\left(x_{N_x}\right)$ by the equation

$$\dot{m}_j\left(x_{N_x}\right) = N_y A_j C_D \left(\frac{p_c^{out}}{p_c^{in}}\right)^{\frac{1}{\gamma}} \sqrt{\frac{2\gamma}{\gamma - 1}\frac{p_c^{in}}{\rho_c^{in}}\left[1 - \left(\frac{p_c^{out}}{p_c^{in}}\right)^{\frac{\gamma-1}{\gamma}}\right]}, \tag{23}$$

where $\gamma = \frac{c_P}{c_V}$ is ratio of specific heats of the air, C_D is the jet discharge coefficient, $A_j = \pi d^2/4$ is the surface of a single orifice, N_y is the number of holes in every row and $\rho_c^{in} = \frac{p_c^{in}}{RT_c}$ is the density of inlet cooling air, with R being the ideal gas constant [30]. On the other hand we know from (6) that

$$\dot{m}_j\left(x_{N_x}\right) = A_j N_y G_j(x_{N_x}); \tag{24}$$

where the jet flow mass velocity $G_j(x_{N_x})$ for the last row can be evaluated using (12).

Thus, substituting (24) in (23) and simplifying we obtain the equation

$$G_j(x_{N_x}) = C_D \left(\frac{p_c^{out}}{p_c^{in}}\right)^{\frac{1}{\gamma}} \sqrt{\frac{2\gamma}{\gamma - 1}\frac{p_c^{in}}{\rho_c^{in}}\left[1 - \left(\frac{p_c^{out}}{p_c^{in}}\right)^{\frac{\gamma-1}{\gamma}}\right]}. \tag{25}$$

It is possible to estimate the value of $p_c^{out}(\mathbf{v})$ for a given design vector $\mathbf{v}$ as the solution of a scalar non linear equation

$$f(p) := p^{\frac{1}{\gamma}}\sqrt{\left(p_c^{in}\right)^{\frac{\gamma-1}{\gamma}} - p^{\frac{\gamma-1}{\gamma}}}\left(p_c^{in}\right)^{\frac{\gamma-1}{2\gamma}} - \frac{G_j(x_{N_x})}{C_D}\sqrt{\frac{\gamma - 1}{2\gamma}RT_c} = 0, \tag{26}$$

where the function $f : \left[0, p_c^{in}\right] \to \mathbb{R}$ is derived from (25). It can be shown that problem (26) admits

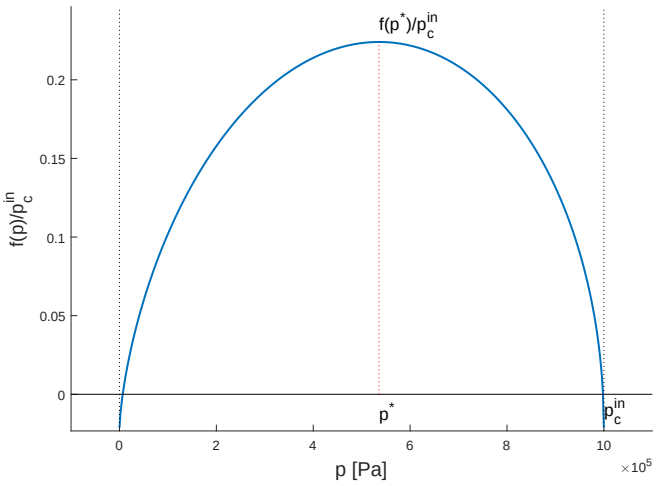

Figure 6: Graphical representation of the function $f(p)/p_c^{in}$, with $f(p)$ defined in (26) in the interval $\left[0, p_c^{in}\right]$, for $p_c^{in} = 2 \cdot 10^6$ Pa.

two solutions in $\left(0, p_c^{in}\right)$ with one in the open sub-interval (p^*, p_c^{in}) (see e.g. Figure 6), where p^* is the critic pressure defined as

$$p^* = \left(\frac{2}{\gamma + 1}\right)^{\frac{\gamma}{\gamma - 1}} p_c^{in}. \tag{27}$$

We look for the solution in (p^*, p_c^{in}), since for $p_c^{out} \leq p^*$ supersonic flow surely occurs somewhere in the cooling system. Once we have estimated p_c^{out} we can check the constraint (22), which can be rearranged as a constraint on pressure difference in the following way

$$\Delta p_c\left(\mathbf{v}\right) := p_c^{in} - p_c^{out}(\mathbf{v}) \leq \left(1 - r_{\max}\right) p_c^{in} =: \Delta p_c^{\max}. \tag{28}$$

2.2.3 Feasibility linear constraints

These constraints are meant to ensure a meaningful and applicable solution. We do not want to get an uncraftable or physically meaningless design for an impingement plate. Some unacceptable designs are, for example, one with holes too small, one with jet rows too close to each other or one with overlapping holes. Most of these unwanted results can be avoided by setting suitable box constraints for the continuous variables (x_n, y_n, z_n, d), while the variable *layout* is "unconstrained", since it is a non-ordinal categorical variable which admits only two values.

Moreover, the validity of Florschuetz's model (4) has to be ensured then it is important to keep the variables in a subspace where (4) has been validated. To this end, the following linear constraints

$$1 \leq \frac{z_n}{d} \leq 3, \tag{29}$$

$$4 \leq \frac{y_n}{d} \leq 8, \tag{30}$$

$$6.25 \cdot 10^{-1} \leq \frac{x_n}{y_n} \leq 3.75, \tag{31}$$

$$5 \leq \frac{x_n}{d} \leq \begin{cases} 15 & \text{if } \textit{layout} = \texttt{inline} \\ 10 & \text{if } \textit{layout} = \texttt{staggered} \end{cases}. \tag{32}$$

have to be satisfied. The coefficients and the form of the inequalities above were empirically defined in [17, 16]. Note that the depending on the value of the variable *layout*, the constraint (32) changes. It is easy to show that when constraints (29)-(32) are fulfilled, overlapping holes are avoided in the design

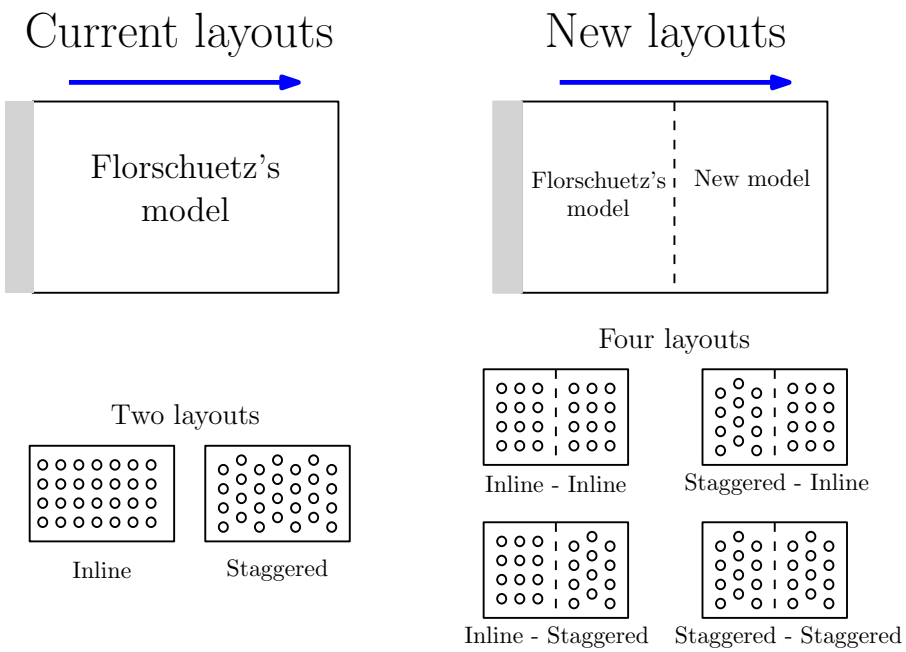

Figure 7: **Left:** the current choice used to define NOZZLE with two layouts. **Right:** a hint for a (possible) new idea to allow four layouts.

of the impingement plate, both for inline and staggered layout.

Remark 2.1. *We remark that the Florschuetz model used to define our BBO model admits only two values for the categorical variable layout. Therefore both the problem formulation and solution may be considerably simplified. On the other hand, the possibility to explore, at least numerically, further values for this variable represents an interesting new perspective for engineering design and motivates our formulation of the problem.*

Enlarging the number of values of layout requires overcoming the Florschuetz model, or at least, its integration in order to exploit its simplicity and reliability. We present now an alternative manufacturable pattern layout under study that motivated our proposed general framework. Figure 7 displays on the left a representation of the current NOZZLE version: we have the impingement plate where we use Florschuetz's model, thus we have only two possible layouts, inline and staggered. Alternatively, the right Figure 7 shows the impingement plate split into two regions to allow one layout per region. In this way, considering only the `inline` *and* `staggered`*, we can have four possible layouts for the whole plate given by the combinations of* `inline` *and* `staggered` *on every half-plate. As the figure shows, we could apply Florschuetz's model to the half of the plate that is upstream of the cooling air flow (whose direction is represented by the blue arrow), while for the downstream half, it is necessary to find another model. This is because downstream we are no longer in the exact configuration for which the Florschuetz model was developed, in fact in the first row of jets we will have a nonzero crossflow mass velocity since we will have the contribution coming from the upstream half. Recall that the Florschuetz model was instead developed for a configuration such as that shown on the left in Figure 7, in which we have no crossflow coming from upstream of the plate. Therefore, a different model needs to be applied to the downstream half that is currently under study.*

2.3 Black-box definition

In this section we merge all the ingredients defined in Sections 2.1 and 2.2 to present the black-box formulation that models the impingement cooling system for a nozzle in a gas turbine. By black-box

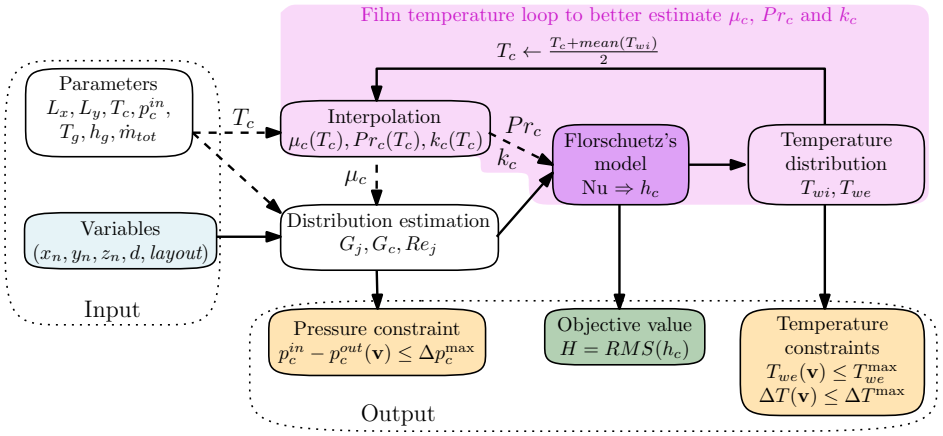

Figure 8: Flow chart for the structure of the black-box.

we mean a set of computational models for the optimization problem (objective and constraints) that can be evaluated to simulate the cooling system under consideration.

The basic structure of the black-box is represented in the flow chart in Figure 8 and described with more detail in Algorithm 3. The black-box takes as input the variable vector $\mathbf{v} = (x_n, y_n, z_n, d, layout)$ that defines the design of the impingement plate, and the fixed parameters given by assumptions and boundary conditions: the inlet temperature T_c and pressure p_c^{in} of the cooling air, the total mass flow $\dot{m}_{tot}$ of cooling air coming from upstream of the cooling system, the temperature T_g and HTC h_g of the hot gas that surrounds the nozzle, the discharge coefficient C_D of the holes and the sizes L_x and L_y of the rectangular impingement plate (Step 1).

The inlet temperature T_c is used to interpolate the values of the dynamic viscosity μ_c, the thermal conductivity k_c and Prandtl number Pr_c for the cooling air (Step 4).

The design variables $\mathbf{v}$, together with the dynamic viscosity μ_c and the parameters L_x, L_y, $\dot{m}_{tot}$ and C_D are given as inputs to Algorithm 1 to estimate the stream-wise distributions at every row of holes of the flow mass velocities to the jets G_j and to the cross section G_c and of the jet Reynolds number Re_j. Distributions of G_j and G_c are then used to estimate the outlet pressure p_c^{out} of the cooling air by solving the non linear equation (26) (Steps 5-7).

The HTC distribution $h_c(\mathbf{v})$ of the cooling air is estimated using the variables $\mathbf{v}$, the distributions G_j, G_c and Re_j and the coefficients k_c and Pr_c with Florschuetz's model (4) and (3) (Steps 8-10).

The HTC distribution $h_c(\mathbf{v})$ is then used to calculate the objective value $H(\mathbf{v}) = (h_c(\mathbf{v}))_{RMS}$ and to estimate the wall temperature distributions $T_{wi}(\mathbf{v})$ and $T_{we}(\mathbf{v})$ using one of the two approaches explained in Section 2.2.1 (Step 24).

The black-box returns as outputs the HTC distribution h_c to obtain the objective value $H(\mathbf{v})$ and the temperature distributions T_{wi} and T_{we} and the value of p_c^{out} to verify the constraints (19)-(20) and (28) respectively.

We observe that the wall temperature distributions are not only used to define constraints. Indeed, the distribution of T_{wi} is also used for better estimation of the thermal conductivity k_c of the cooling air. As discussed in Section 2.1.2, the value of this coefficient is obtained by interpolation using T_c as query value; however, assuming T_c as the temperature of the cooling air in the boundary layer near the inner nozzle wall is quite inaccurate, because in that region the cooling air is affected by the temperature of the wall, heated by conduction. So, as query value, we use the film temperature T_f which is an approximation of the temperature of a fluid inside a convection boundary layer [17, 16],

and it is defined as

$$T_f = \frac{\text{mean}\,(T_{wi}) + T_c}{2}. \tag{33}$$

After the a first interpolation of $k_c := k_c(T_c)$ it is necessary to estimate again the distributions of h_c, T_{wi} and T_{we}. We recall that k_c is involved directly in (3) for the evaluation of the distribution h_c and consequently in (15) for the estimation of T_{wi} and T_{we}. We include all this steps into a loop that in every iterate generates new estimations h_c, T_{wi} and T_{we} and it ends when the relative error between two subsequent estimations of h_c is below a certain tolerance tol_h, i.e.

$$\frac{\|h_c - (h_c)_{old}\|_2}{\|(h_c)_{old}\|_2} \leq \text{tol}_h. \tag{34}$$

This loop is described in Steps 9-23 of Algorithm 3. Note that this loop does not involve the other coefficients μ_c and Pr_c, that is because in that case assuming T_c as query for the interpolation is acceptable. Furthermore the loop does not involve directly the distribution of the external wall temperature T_{we}.

We note that the evaluation of the feasibility constraints (29)-(32) is not included in Algorithm 3 as they can can be easily treated outside the black-box.

Finally, we note that the implementation of the black-box results in an evaluation cost of the objective function and of the constraints that is rather cheap. This is due to the small number of variables (four continuous and one categorical) and to the small size of the distributions handled by the function (e.g. T_{we}, T_{wi}, h_c). Moreover, most of the auxiliary quantities and distributions needed are obtained straightforwardly, with the only exception of the solution of the nonlinear equation (26) to get the outlet pressure p_c^{out}, which uses an iterative method. Thus a single evaluation of the black-box requires a small amount of computational time and memory.

3 Derivative-free optimization for the solution of NOZZLE

In this section we embed the NOZZLE model described in the previous sections in the DFO framework. We therefore describe the main model features and propose a DFO based procedure for its minimization.

3.1 The overall constrained black-box optimization formulation

In Section 2.1 we have described the formulation of the objective function $H(\mathbf{v})$ while in Section 2.2 we have defined and motivated the constraint functions on wall temperature distributions $T_{wi}(\mathbf{v})$, $T_{we}(\mathbf{v})$, on the outlet pressure $p_c^{out}(\mathbf{v})$ and on the design variables $\mathbf{v} = (x_n, y_n, z_n, d, \mathit{layout})$. We can gather all

Algorithm 3 HTC stream-wise distribution with film temperature loop

Initialization

Input: the variables $\mathbf{v} = (x_n, y_n, z_n, d, layout)$ and the parameters L_x, L_y, T_c, T_g, h_g, $\dot{m}_{tot}$, C_D.

Compute $N_x = \lfloor \frac{L_x}{x_n} \rfloor$, $N_y = \lfloor \frac{L_y}{y_n} \rfloor$ e $A_j = \frac{\pi d^2}{4}$.

Define the vector $x_j = \frac{1}{2}x_n : x_n : (N_x - \frac{1}{2})x_n$ of the x-coordinate of the centers of the span-wise rows.

Calculate by nonlinear interpolation of $\mu_c := \mu_c(T_c)$, $\mathrm{Pr}_c := \mathrm{Pr}_c(T_c)$.

Mass velocity distributions

Estimate the distributions of G_j and $\frac{G_c}{G_j}$ using Algorithm 1.

Jet Reynolds number distribution

Compute the distributions of jet Reynolds number $Re_j(x_i)$ using (8).

Outlet pressure estimation

Estimate p_c^{out} as a solution of (26).

Nu distribution

Using (4) and (5), obtain the stream-wise distribution of the Nusselt number Nu.

First interpolation of k_c and first evaluation of h_c, T_{wi} and T_{we}

Calculate by nonlinear interpolation of $(k_c)_{old} := k_c(T_c)$ of the coolant fluid.

Compute the distribution of $(h_c)_{old}$ using (3):

$$(h_c)_{old}(x_i) = \frac{\mathrm{Nu}(x_i)\,(k_c)_{old}}{d}, \quad i = 1, ..., N_x.$$

Estimate of the distribution of $(T_{wi})_{old}$ and $(T_{we})_{old}$ via solving (15).

Film temperature loop

for $it = 1, 2, ...$ **do**

Set

$$T_f = \frac{\overline{(T_{wi})_{old}} + T_c}{2}.$$

Interpolate $k_c := k_c(T_f)$.

Evaluate

$$h_c(x_i) = \frac{\mathrm{Nu}(x_i)k_c}{d}, \quad i = 1, ..., N_x.$$

Estimate new distribution T_{wi} and T_{we} via solving (15) using h_c.

Compute the relative error ε_{rel} as

$$\varepsilon_{rel} = \frac{\|h_c - (h_c)_{old}\|_2}{\|(h_c)_{old}\|_2}.$$

if $\varepsilon_{rel} \leq \mathrm{tol}_h$ **then**

Break.

else

Set

$$(h_c)_{old} = h_c;$$
$$(T_{wi})_{old} = T_{wi};$$
$$(T_{we})_{old} = T_{we}.$$

end if

end for

Objective value

Set $H = (h_c)_{RMS}$

Outputs

Return H, T_{wi}, T_{we} and p_c^{out}.

the functions defined so far to formulate the NOZZLE as a standard minimization problem as follows.

$$
\begin{aligned}
\min_{\mathbf{v}} \quad & -H(\mathbf{v}) \\
\text{s.t.} \quad & c_1(\mathbf{v}) := \left(T_{we}(\mathbf{v})\right)_{RMS} - T_{we}^{\max} \leq 0; \\
& c_2(\mathbf{v}) := \left(\Delta T(\mathbf{v})\right)_{RMS} - \Delta T^{\max} \leq 0; \\
& c_3(\mathbf{v}) := \Delta p_c(\mathbf{v}) - \Delta p_c^{\max} \leq 0; \\
& c_4(\mathbf{v}) := 1 - \frac{z_n}{d} \leq 0; \\
& c_5(\mathbf{v}) := \frac{z_n}{d} - 3 \leq 0; \\
& c_6(\mathbf{v}) := 4 - \frac{y_n}{d} \leq 0; \\
& c_7(\mathbf{v}) := \frac{y_n}{d} - 8 \leq 0; \\
& c_8(\mathbf{v}) := \frac{x_n}{y_n} - 3.75 \leq 0; \\
& c_9(\mathbf{v}) := 6.25 \cdot 10^{-1} - \frac{x_n}{y_n} \leq 0; \\
& c_{10}(\mathbf{v}) := 5 - \frac{x_n}{d} \leq 0; \\
& c_{11}(\mathbf{v}) := \begin{cases} \frac{x_n}{d} - 15 \leq 0 & \text{if } \textit{layout} = \texttt{inline} \\ \frac{x_n}{d} - 10 \leq 0 & \text{if } \textit{layout} = \texttt{staggered} \end{cases}; \\
& (x_n, y_n, z_n, d) \in \mathcal{B} \subset \mathbb{R}^4_{>0}; \quad \textit{layout} \in \{\texttt{inline}, \texttt{staggered}\}.
\end{aligned}
\tag{35}
$$

Formulation (35) represents a constrained black-box optimization problem, where the objective $-H(\mathbf{v})$ is the negative RMS of the HTC distribution h_c defined in (14) and it is returned as one of the outputs of the black-box function defined in Section 2.3. The other outputs of the black-box are used to define the constraint functions $c_1(\mathbf{v})$, $c_2(\mathbf{v})$, $c_2(\mathbf{v})$ which are respectively derived from (19), (20) and (28). The constraint functions $c_i(\mathbf{v})$ with $i = 4, ..., 11$ are derived directly from the feasibility constraints (29)-(32).

Referring to the black-box optimization constraint taxonomy presented in [32] we can identify two kinds of constraint functions in (35). Functions c_1, c_2 and c_3 are black-box simulation based, thus any kind of (sub-)gradient is unavailable, and they are also relaxable, since an impingement plate design that violates these constraints is still meaningful and can be post-processed. Remaining functions, from c_4 to c_{11}, are algebraic since they are expressed in an explicit form but they also are unrelaxable because, as we explained in Section 2.2.3, they describe the validity space of Florschuetz's model used to define the black-box.

3.2 DFO solution framework

The structure of NOZZLE as optimization problem in (35) clearly calls for DFO tools. Two main issues influences the choice of the DFO approach: the presence of black-block constraints and the presence of mixed variables, that is continuous and categorical variables.

Focusing on DFO for constrained black-box optimization, three different approaches can be found in the literature: filter approaches, model-based approaches and penalty approaches. For a more complete overview refer to [13].

Filter methods aim to address black-box constraints, and also algebraic constraints, by concurrently minimizing both the objective and constraint violation. Audet and Dennis introduced in [3] a

pattern-search technique for general constrained optimization that accepts steps that improve either the objective or the violation of black-box constraints. Further approaches can be found in [25, 34].

Model-based approaches defines a surrogate problem by building models to replace the simulation-based functions (objective and constraints). Powell in [24] develops a direct search method for constrained optimization which approximates the objective and constraint functions using linear models defined over a simplex. Bürmen *et al.* in [1] presented a version of Mesh Adaptive Direct Search (MADS) applied to a surrogate problem defined using strongly convex quadratic model for the objective function and linear models for the constraints. In [27], a trust-region method employing fully linear models of both constraint and objective functions was developed. An alternative method employs interpolating radial basis function surrogates of the objective and constraint functions (CONORBIT) [29]. Finally, in [8] a simplex-gradient-based approach is considered for approximating normal cones when the black-box constraints are quantifiable.

Regarding penalty methods, Audet and Dennis [4] propose a progressive-barrier method within MADS method with quadratic penalty for relaxable black-box constraints and an extreme-barrier penalty for unrelaxable ones; in [31], an extreme-barrier penalty is used again to handle unrelaxable constraints, while an exact penalty is used for relaxable ones, everything within a Directional Direct Search (DDS) framework; the paper [10] proposes a line-search method with a sequence of quadratic penalty functions to address non-differentiable constraint and objective functions; Sampaio and Toint propose a derivative-free variant of trust-funnel method to deal equality constraints without using neither merit functions nor filters, see [28]. Finally, an augmented Lagrangian framework was developed in [35] where the merit function is defined using Gaussian process models of the objective and constraint functions.

The literature comprising DFO algorithms that handles mixed variables is not very extensive. Papers by Audet and Dennis [5], by Lucidi *et al.* [33] and by Abramson [23] consider the presence of categorical variables. In particular, the work [23] extends the MADS algorithms for solving constrained mixed variable optimization problems. These algorithms have been successfully applied to relevant engineering applications, see e.g. [18, 22]. Finally, we mention the recent work [20] where the pattern search method Brute Force Optimizer (BFO) proposed in [19] for solving problems with continuous and discrete variables, has been extended to handle categorical variables. More details on BFO are postponed to next section.

3.3 Our DFO proposal: the ℓ_1-penalty BFO method

We propose a new DFO penalty methods leveraging on the general ℓ_1-penalty method for derivative-based optimization, see e.g. [14]. Indeed we consider problem (35), and define the penalty function $\phi_1(\mathbf{v}, \varepsilon)$ as

$$\phi_1(\mathbf{v}, \varepsilon) = -H(\mathbf{v}) + \varepsilon C(\mathbf{v}), \tag{36}$$

where $\varepsilon > 0$ is the penalty parameter and the constraint violation function gathers all the constraint functions $C(\mathbf{v})$ as follows

$$C(\mathbf{v}) := \sum_{j=1}^{11} \max\{0, c_j(\mathbf{v})\}. \tag{37}$$

By choosing an increasing sequence of penalty parameters $\{\varepsilon_k\}_{k\in\mathbb{N}}$ such that $\varepsilon_k \to \infty$ we define a sequence of unconstrained minimization problems of the form

$$\begin{aligned} \min_{\mathbf{v}} \quad & \phi_1(\mathbf{v}, \varepsilon_k) = -H(\mathbf{v}) + \varepsilon_k C(\mathbf{v}) \\ \text{s.t.} \quad & (x_n, y_n, z_n, d) \in \mathcal{B} \subset \mathbb{R}^4_{>0}, \quad \textit{layout} \in \{\texttt{inline}, \texttt{staggered}\}, \end{aligned} \tag{38}$$

where we penalize constraint violations more severely, thereby forcing the minimizer of the penalty function closer to the feasible region for problem (35). Then, we use a derivative-free algorithm for solving (38) for every value of ε_k. In particular, our choice for the inner solver is the Brute Force Optimizer (BFO) [19, 20].

BFO is a simple random pattern search algorithm specifically designed for black-box optimization as it can deal bound-constrained optimization problems without any regularity or convexity assumption on the objective function. In particular, BFO is suitable for the minimization of the nonsmooth black-box function (36).

As a pattern search method, for every iterate $\overline{v}$ BFO creates a polling set of directions $\mathcal{P}$ that defines a finite local mesh around $\overline{v}$, BFO searches for any improvement of the objective function on this mesh by evaluating all the points on the mesh and if it succeed in finding a better value for the objective function at a new point $\hat{v}$ the iterate is updated; otherwise if BFO fails in finding an improvement on the local mesh, the mesh is refined (i.e. a new mesh is defined closer to $\overline{v}$) and a new search is performed.

BFO can handle different types of variables as continuous, integer, discrete, mixed or categorical. If the optimization problem has mixed variables, e.g. continuous and categorical variables, the search phase is more articulated and is called tree-search strategy, see further details in [19, 20]. In fact, the search phase is firstly performed involving only the continuous components of the iterate $\overline{v}$ while the discrete ones are kept fixed, then, if there are no improvements on the continuous mesh, instead of refining the mesh a further search is performed by exploring the meshes defined around an iterate defined by fixing successively each of the non-continuous variables to a value neighboring that present in $\overline{v}$. As an example let us consider the vector of the design variables $\overline{\mathbf{v}} = \left(\overline{x_n}, \overline{y_n}, \overline{z_n}, \overline{d}, \texttt{staggered}\right)$; at first the the mesh is build around the continuous part $\left(\overline{x_n}, \overline{y_n}, \overline{z_n}, \overline{d}\right)$ while keeping fixed $\overline{\textit{layout}} = \texttt{staggered}$, if BFO fails in finding an improvement another search is done on the same mesh for the continuous part but setting $\overline{\textit{layout}} = \texttt{inline}$. [2]

The overall proposed algorithm is detailed in Algorithm 4.

Algorithm 4 The ℓ_1-penalty BFO scheme

1: Given $\mathbf{v}_0^s$, $\varepsilon_0 > 0$, $\nu > 1$, $\tau > 0$, $k_{\max} > 0$
2: **for** $k = 0, 1, ..., k_{\max}$ **do**
3: Use BFO to find an approximate minimizer $\mathbf{v}_k$ of $\phi_1(\mathbf{v}, \varepsilon_k)$, starting from $\mathbf{v}_k^s$.
4: **if** $C(\mathbf{v}_k) \leq \tau$ **then**
5: Stop and return $\mathbf{v}_k$.
6: **end if**
7: Set $\varepsilon_{k+1} = \nu\varepsilon_k$.
8: Set $\mathbf{v}_{k+1}^s = \mathbf{v}_k$.
9: **end for**

At the beginning, an initial guess $\mathbf{v}_0^s$ and an initial value $\varepsilon_0 > 0$ for the penalty parameter are chose. Also the update coefficient $\nu > 1$ to increase the penalty, the tolerance $\tau > 0$ for the constraint

[2] Although problem (38) might be solved by interpreting the categorical variable as a binary one, thus considerably simplifying the problem solution, we chose to treat it as truly categorical for the sake of generality (see Remark 2.1).

violation and a maximum number of iteration $k_{\max}$ are set. At every kth iteration BFO is called to find an approximate minimizer of the penalty function $\phi_1(\mathbf{v}, \varepsilon_k)$, where the penalty parameter is kept fixed, returning a point $\mathbf{v}_k$. Then the value of the constraint violation function $C(\mathbf{v}_k)$ is checked and if $C(\mathbf{v}_k) \leq \tau$ then $\mathbf{v}_k$ is accepted as an acceptable solution and the procedure stops. Otherwise the penalty parameter ε_{k+1} is increased by a factor ν and the new starting point $\mathbf{v}^s_{k+1}$ is set equal to the minimizer just found $\mathbf{v}_k$ and a new iteration starts. The update strategy for the new starting point at Step 8 is motivated by the fact that with a good choice of the initial penalty parameter ε_0, it is possible to obtain from the first iteration an approximate minimizer that does not excessively violate the constraints, so it will be sufficient to search for an admissible solution in a neighborhood of the last minimizer found, having chosen an appropriate parameter ν for the penalty update.

4 Experimental results

In this section we numerically solve the NOZZLE model formulated as problem (35) by exploiting the the black-box functions described in the previous sections and by employing the ℓ_1-penalty BFO algorithm.

Recalling that we aim to find "better" geometric variables x_n, y_n, z_n, d and *layout* that define the design for an impingement plate for a fixed nozzle of a gas turbine, we consider two different problem settings obtained considering two different sets of boundary conditions. The first, called here the "laboratory case", represents a situation that is encountered on a laboratory test bench, that is conditions similar to those under which Florschuetz and collaborators carried out the experiments to derive the model (4) (see [17]) are reproduced. The second has boundary conditions that reflect the typical values of an actual gas turbine and, for this reason, it will be referred as the "industrial case"[3]. The boundary conditions and the upper bounds for both experimental cases are gathered in Table 2. At the moment we emphasize that for both situations we use the same value for the discharge coefficient C_D and mass flow rate $\dot{m}_{tot}$. In addition, the upper limit for the pressure difference $\Delta p_c^{\max}$ is also somewhat the same, specifically the two values for $\Delta p_c^{\max}$ are chosen such that the ratio of the pressure difference to the inlet pressure has as an upper limit equal to 0.04, i.e. we look for a configuration that allow a pressure difference lower than the 4% of p_c^{in}.

For the evaluation of the distributions of the wall temperature we solve the general problem defined in (16) using finite differences as described in Section 2.2.1.

All the experiments have been carried out using Matlab R2023a on a Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz machine with 16 GB RAM and the new release 2.0 of the Matlab BFO package available at `https://github.com/m01marpor/BFO`. Default parameters have been set for BFO while $\varepsilon_0 = 1.5$, $\nu = 10$, $\tau = 10^{-3}$ and $k_{\max} = 15$ have been set in Algorithm 4.

4.1 Laboratory case

Referring to the third column of Table 2, the temperature T_g of the hot gas is around 100°C with low HTC h_g while the inlet temperature of the cooling air T_c is around 50°C and inlet pressure p_c^{in} is about twice the atmospheric pressure. The impingement plate is nearly square with approximately 12 cm per side, and the target surface is 1 cm thick with good thermal conductivity. In this experiment, we start from an initial guess $\mathbf{v}_0^s$ chosen with $x_n = 1.75 \cdot 10^{-2}$ m, $y_n = 8.40 \cdot 10^{-3}$ m, $z_n = 6.30 \cdot 10^{-3}$ m,

[3]The black-box functions implementing the industrial case are available at `https://github.com/GrattonToint/S2MPJ/blob/main/matlab_problems/NOZZLEfp.m`

Parameter name	Description [Unit]	Laboratory case value	Industrial case value
L_x	Plate stream-wise length [m]	$1.27 \cdot 10^{-1}$	$5 \cdot 10^{-2}$
L_y	Plate span-wise length [m]	$1.22 \cdot 10^{-1}$	$5 \cdot 10^{-2}$
T_c	Cooling air inlet temperature [K]	$2.93 \cdot 10^{2}$	$7.73 \cdot 10^{2}$
p_c^{in}	Cooling air inlet pressure [Pa]	$2.03 \cdot 10^{5}$	$1.01 \cdot 10^{6}$
T_g	External hot gas temperature [K]	$3.73 \cdot 10^{2}$	$1.27 \cdot 10^{3}$
h_g	External hot gas HTC [W m^{-2}K^{-1}]	$1 \cdot 10^{2}$	$1 \cdot 10^{3}$
$\dot{m}_{tot}$	Cooling air mass flow rate [Kg s^{-1}]	$1.00 \cdot 10^{-2}$	$1.00 \cdot 10^{-2}$
C_D	Jet discharge coefficient [-]	0.85	0.85
k_w	Wall thermal conductivity [W m^{-1}K^{-1}]	$1 \cdot 10^{2}$	$2 \cdot 10^{1}$
Δs	Wall thickness [m]	$1.00 \cdot 10^{-2}$	$3.00 \cdot 10^{-3}$
$\Delta T^{\max}$	Upper bound for $T_{we} - T_{wi}$ [K]	$3.00 \cdot 10^{1}$	$6.00 \cdot 10^{1}$
$T_{we}^{\max}$	Upper bound for T_{we} [K]	$3.43 \cdot 10^{2}$	$1.07 \cdot 10^{3}$
$\Delta p_c^{\max}$	Upper bound for $p_c^{in} - p_c^{out}$ [Pa]	$8.11 \cdot 10^{3}$	$4.04 \cdot 10^{4}$

Table 2: Parameters for boundary conditions and bounds for black-box constraints for laboratory (3rd column) and industrial (4th column) cases.

$d = 2.10 \cdot 10^{-3}$ m and *layout* = `staggered` as it shown in Figure 9. The initial guess has a value for the objective value $H(\mathbf{v}_0^s) = 2.03 \cdot 10^2$W m^{-2}K^{-1}. In this way $\mathbf{v}_0^s$ satisfies the algebraic constraints (29)-(32) and the box constrains defined by the set $\mathcal{B} \subset \mathbb{R}^4_{>0}$

$$\mathcal{B} := \left\{ (x_n, y_n, z_n, d) \in \left[\frac{L_x}{30}, \frac{L_x}{2}\right] \times \left[\frac{L_y}{30}, \frac{L_y}{5}\right] \times \left[10^{-3}\text{m}, 10^{-2}\text{m}\right] \times \left[2 \cdot 10^{-3}\text{m}, \frac{L_y}{2}\right] \right\}. \tag{39}$$

This definition of $\mathcal{B}$ allows a very simple design for the impingement plate. After only one iteration of

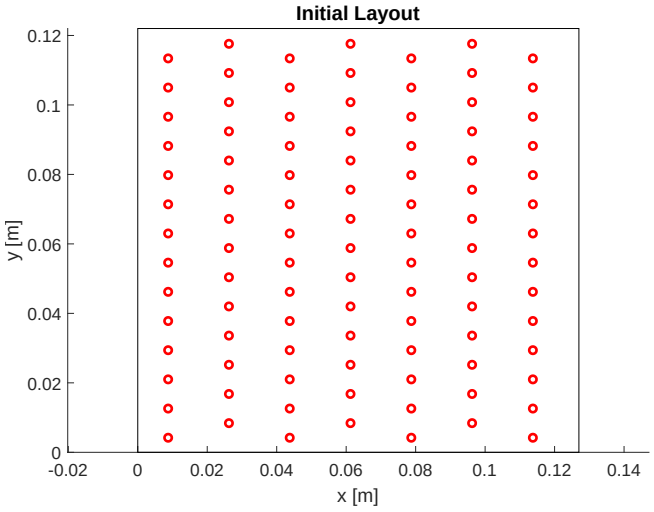

Figure 9: 2-D representation of the initial guess for the laboratory case.

ℓ_1-penalty BFO method and 346 evaluations of the black-box the procedure converges to a solution $\tilde{\mathbf{v}}$ such that $C(\tilde{\mathbf{v}}) \leq \tau$. In particular the geometric variables of this solution are $x_n = 2.54 \cdot 10^{-2}$ m, $y_n = 1.53 \cdot 10^{-2}$ m, $z_n = 4.60 \cdot 10^{-3}$ m, $d = 2.00 \cdot 10^{-3}$ m and *layout* = `inline` with a corresponding objective value $H(\tilde{\mathbf{v}}) = 4.16 \cdot 10^2$W m^{-2}K^{-1}; in Figure 11 (right) we show in 2-D the resulting layout.

In Figures 10 and 11 there are some plots to show the distribution of the HTC of the cooling air h_c (Figure 10, left), the distributions of the wall temperatures (Figure 10, right) and the distribution of the wall temperature difference (Figure 11, left). This last plot shows very low values for the difference between external and internal temperature, and this is due to the thickness of the wall combined

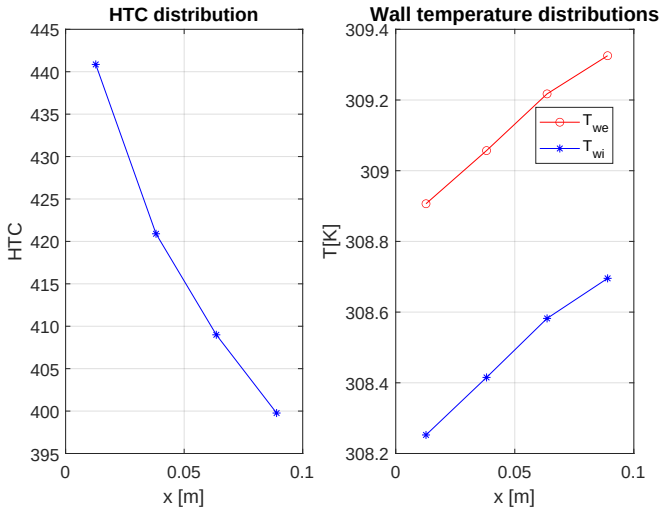

Figure 10: **Laboratory case:** On the left the 1-D distribution of the HTC h_c of the cooling air. On the right the 1-D distribution of the internal and external wall temperature distributions.

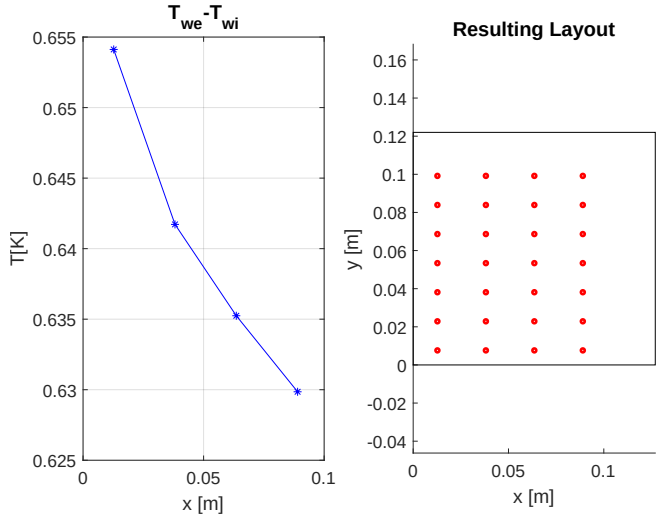

Figure 11: **Laboratory case:** On the left the 1-D distribution of the difference between external and internal wall temperatures. On the right a 2-D representation of the final layout of the impingement plate.

with its thermal conductivity. From the other plots it is possible to see that there are no violation in temperature constraints.

4.2 Industrial case

In this case the temperatures are significantly higher. In fact the temperature T_g of the external hot gas is 1000°C and the inlet temperature T_c is 500°C. From the 4th column of Table 2, we can see that also the HTC of the hot gas h_g is higher and that the inlet pressure p_c^{in} of the cooling air is ten times the atmospheric pressure. On the other hand we are considering a smaller impingement plate (a square with with 5cm-long side) and a thinner target surface (only 3mm thick) with a lower thermal conductivity. Since the boundary conditions change, the upper bounds for the temperature constraints must be increased (see Table 2, 4th column). Since the plate is smaller, also the box constraints $\mathcal{B}$ are changed a bit, in particular we allow the diameter d to be smaller, thus $d \in \left[5 \cdot 10^{-4}\text{m}, \frac{L_y}{2}\right]$.

The initial guess $\mathbf{v}_0^s$ has values $x_n = 5.00 \cdot 10^{-3}$ m, $y_n = 4.00 \cdot 10^{-3}$ m, $z_n = 3.00 \cdot 10^{-3}$ m, $d = 1.00 \cdot 10^{-3}$ m and *layout* = `staggered` and has an objective value of $H\left(\mathbf{v}_0^s\right) = 1.01 \cdot 10^3 \text{W m}^{-2}\text{K}^{-1}$. The initial layout is shown in Figure 12.

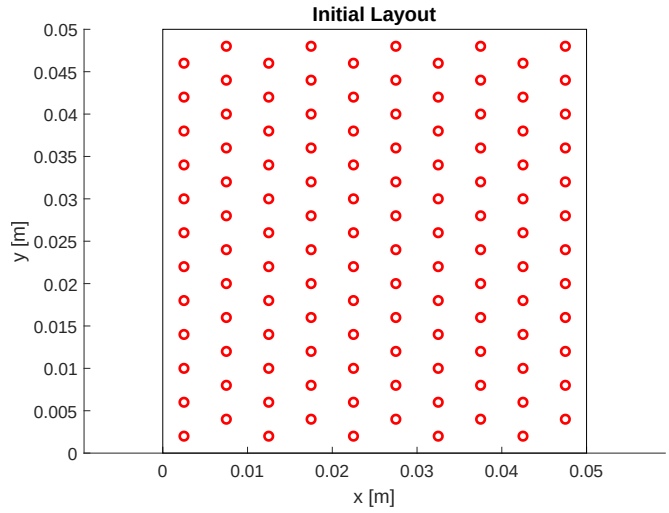

Figure 12: 2-D representation of the initial guess for the industrial case.

Again, with only one iteration of ℓ_1-penalty BFO method and 252 evaluations of the black-box the procedure converges to a solution $\tilde{\mathbf{v}}$ with $H(\tilde{\mathbf{v}}) = 2.71 \cdot 10^3 \text{W m}^{-2}\text{K}^{-1}$. The geometric variables have the following values $x_n = 6.28 \cdot 10^{-3}$ m, $y_n = 3.87 \cdot 10^{-3}$ m, $z_n = 1.46 \cdot 10^{-3}$ m, $d = 5.00 \cdot 10^{-4}$ m and $layout =$ `inline`. The resulting layout is shown in Figure 14 (right).

In Figure 13 are plotted the distribution of HTC h_c (left) and the distributions of the wall temperatures (right); in Figure 14, on the left, we have the distribution of the temperature difference on the target wall. Also in this case there is no violation of the temperature constraints.

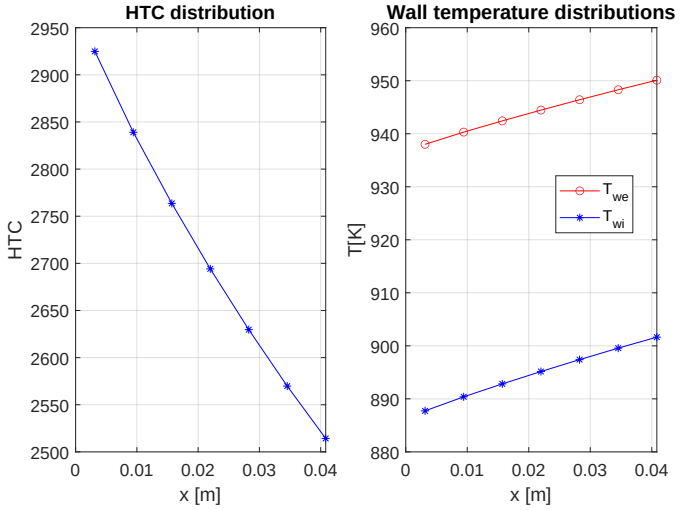

Figure 13: **Industrial case:** On the left the 1-D distribution of the HTC h_c of the cooling air. On the right the 1-D distribution of the internal and external wall temperature distributions.

4.3 Comments on the numerical results

Summarizing, in both the problem settings, our DFO approach allows to compute solutions that improve the performance of the cooling system with low computational effort (only a few hundred function evaluations). In particular, we improve the RMS heat transfer h_c from $2.03 \cdot 10^2 \text{W m}^{-2}\text{K}^{-1}$ to $4.16 \cdot 10^2 \text{W m}^{-2}\text{K}^{-1}$ for the "laboratory case" and from $1.01 \cdot 10^3 \text{W m}^{-2}\text{K}^{-1}$ to $2.71 \cdot 10^3 \text{W m}^{-2}\text{K}^{-1}$ for the "industrial case". In addition, the behavior of the temperature distributions shown in Figures 10, 11, 13 and 14 are consistent with the studies previously done (see [17, 16]). Let us notice that both

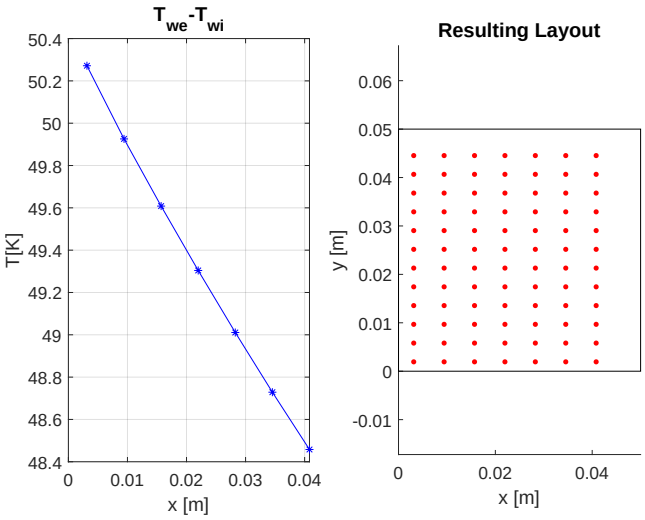

Figure 14: **Laboratory case:** On the left the 1-D distribution of the difference between external and internal wall temperatures. On the right a 2-D representation of the final layout of the impingement plate.

the computed solutions are not "far" from the initial guesses, that is because the procedure described in Algorithm 4 is a local optimization strategy, i.e. the algorithm strongly depends on the choice of the initial guess. On the other hand, we remark the switch in the value of the categorical *layout* from `staggered` to `inline`.

5 Conclusions and further developments

In this work, we defined a black-box function implementing a simple model for the the design of an impingement cooling system. This function simulates the functioning of the cooling system inside a turbine nozzle and returns as output the parameters to evaluate its efficiency. As a result, we proposed NOZZLE, a formulation of the problem of maximizing this efficiency in the form of a mixed variable constrained black-box optimization problem and we numerically validate the model using DFO algorithms. The overall proposed procedure allows for the design of an efficient impingement cooling system and for its improvement without having to rely on the operator's experience and also reduces the time required with respect to the standard procedure.

The obtained preliminary results form the basis for future developments involving both the black-box function formulation and the optimization method. The black-box function can be improved by starting from a problem with different boundary conditions. In particular, instead of knowing the total mass flow rate $\dot{m}_{tot}$ provided to the cooling system we impose a fixed value for the outlet pressure of the cooling air p_c^{out}. This change would allow us to remove the constraint on the pressure (28) so we do not need to solve the non linear equation (26), saving some computational effort. Another improvement of the simulator is to allow the layout variable to assume more than two values, so that it can be considered a true categorical variable and so that the possible choices for an improved design can be increased (see also Remark 2.1).

Regarding the optimization method, we observed in Section 4 that the ℓ_1-penalty BFO method returned reliable local solutions. On the other had, practitioners often need a global solution. Therefore, next step will be devoted to the implementation a global optimization strategy that is suitable for the problem under consideration.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Funding

FM and MP are member of the Gruppo Nazionale Calcolo Scientifico-Istituto Nazionale di Alta Matematica (GNCS-INdAM) and their work was partially supported by INdAM-GNCS under the INdAM-GNCS project CUP_E53C23001670001. The work of FM is supported by the program "Programma Operativo Nazionale Ricerca e Innovazione 2014-2020 (CCI2014IT16M2OP005)" - Azione IV.5 "Dottorati e contratti di ricerca su tematiche green" XXXVII ciclo, code DOT1303154-4, CUP J35F21003200006.

Availability of data and materials

Due to the nature of the research, due to commercial supporting data is not available.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

All authors contributed equally to the writing of this article. All authors reviewed the manuscript.

Acknowledgments

The authors would like to thank Philippe Toint for fruitful discussions and enlightening opinions and Luca Innocenti for his assistance regarding engineering aspects.

References

[1] Bűrmen A, Olenšek J, and Tuma T. "Mesh adaptive direct search with second directional derivative-based Hessian update". In: *Comput. Optim. Appl.* 62 (2015), pp. 693–715.

[2] Stéphane Alarie et al. "Two decades of blackbox optimization applications". In: *EURO Journal on Computational Optimization* 9 (2021), p. 100011.

[3] Audet C and Dennis JE Jr. "A pattern search filter method for nonlinear programming without derivatives". In: *SIAM J. Optim.* 14.4 (2004), pp. 980–1010.

[4] Audet C and Dennis JE Jr. "A progressive barrier for derivative-free nonlinear programming". In: *SIAM J. Optim.* 20.1 (2009), pp. 445–472.

[5] Audet C and Dennis JE Jr. "Pattern search algorithms for mixed variable programming". In: *SIAM Journal on Optimization* 11.3 (2001), pp. 573–594.

[6] Audet C and Kokkolaras M. "Blackbox and derivative-free optimization: theory, algorithms and applications". In: *Optimization and Engineering* 17 (2016), pp. 1–2.

[7] Audet C and Hare W. *Derivative-free and Blackbox Optimization*. Springer Series in Operations Research and Financial Engineering. Cham, Switzerland: Springer, 2017.

[8] Davis C and Hare W. "Exploiting Known Structures to Approximate Normal Cones". In: *Math. Oper. Res.* 38.4 (2013), pp. 665–681.

[9] Incropera FP et al. *Incropera's Principles of Heat and Mass Transfer*. John Wiley & Sons, Incorporated, 2017. ISBN: 9781119382911. URL: `https://books.google.it/books?id=PGIAMQAACAAJ`.

[10] Fasano G et al. "A linesearch-based derivative-free approach for nonsmooth constrained optimization". In: *SIAM J. Optim.* 24.3 (2014), pp. 959–992.

[11] S. Gratton and Ph. L. Toint. "S2MPJ and CUTEst optimization problems for Matlab, Python and Julia". In: *arXiv preprint arXiv:2407.07812* (2024).

[12] Han J, Dutta S, and Ekkad S. *Gas turbine heat transfer and cooling technology*. CRC press, 2012.

[13] Larson J, Menickelly M, and Wild SM. "Derivative-free optimization methods". In: *Acta Numerica* 28 (2019), pp. 287–404.

[14] Nocedal J and Wright SJ. *Numerical optimization*. Springer, 1999.

[15] Florschuetz LW, Truman CR, and Metzger DE. "Streamwise flow and heat transfer distributions for jet array impingement with crossflow". In: *Turbo Expo: Power for Land, Sea, and Air*. Vol. 79634. American Society of Mechanical Engineers. 1981.

[16] Florschuetz LW et al. *Jet array impingement with crossflow-correlation of streamwise resolved flow and heat transfer distributions*. Tech. rep. 1981.

[17] Florschuetz LW et al. *Multiple jet impingement heat transfer characteristic: Experimental investigation of in-line and staggered arrays with crossflow*. Tech. rep. 1980.

[18] Kokkolaras M, Audet C, and Dennis JE Jr. "Mixed variable optimization of the number and composition of heat intercepts in a thermal insulation system". In: *Optimization and Engineering* 2 (2001), pp. 5–29.

[19] Porcelli M and Toint PhL. "BFO, a trainable derivative-free brute force optimizer for nonlinear bound-constrained optimization and equilibrium computations with continuous and discrete variables". In: *ACM Trans. Math. Software* 44.1 (2017), pp. 1–25.

[20] Porcelli M and Toint PhL. "Exploiting Problem Structure in Derivative Free Optimization". In: *ACM Trans. Math. Software* 48.1 (2022), pp. 1–25.

[21] Pourbagian M et al. "Constrained problem formulations for power optimization of aircraft electro-thermal anti-icing systems". In: *Optimization and Engineering* 16 (2015), pp. 663–693.

[22] Abramson MA. "Mixed variable optimization of a load-bearing thermal insulation system using a filter pattern search algorithm". In: *Optimization and Engineering* 5 (2004), pp. 157–177.

[23] Abramson MA et al. "Mesh adaptive direct search algorithms for mixed variable optimization". In: *Optimization Letters* 3 (2009), pp. 35–47.

[24] Powell MJD. *A direct search optimization method that models the objective and constraint functions by linear interpolation*. Springer, 1994.

[25] Echebest N, Schuverdt ML, and Vignau RP. "An inexact restoration derivative-free filter method for nonlinear programming". In: *J. Comput. Appl. Math.* 36 (2017), pp. 693–718.

[26] Zuckerman N and Lior N. "Jet Impingement Heat Transfer: Physics, Correlations, and Numerical Modeling". In: ed. by Greene GA et al. Vol. 39. Advances in Heat Transfer. Elsevier, 2006, pp. 565–631. DOI: https://doi.org/10.1016/S0065-2717(06)39006-5. URL: https://www.sciencedirect.com/science/article/pii/S0065271706390065.

[27] "NOWPAC: a provably convergent derivative-free nonlinear optimizer with path-augmented constraints". In: *arXiv preprint arXiv:1403.1931* (2014).

[28] Sampaio PhR and Toint PhL. "Numerical experience with a derivative-free trust-funnel method for nonlinear optimization problems with general nonlinear constraints". In: *Optimization Methods and Software* 31.3 (2016), pp. 511–534.

[29] Regis RG and Wild SM. "CONORBIT: constrained optimization by radial basis function interpolation in trust regions". In: *Optim. Methods Softw.* 32.3 (2017), pp. 552–580.

[30] Fox RW, Pritchard PJ, and McDonald AT. *Introduction to Fluid Mechanics*. John Wiley & Sons, 2010. ISBN: 9780470547557. URL: https://books.google.it/books?id=kiFJRQAACAAJ.

[31] Gratton S and Vicente LN. "A merit function approach for direct search". In: *SIAM J. Optim.* 24.4 (2014), pp. 1980–1998.

[32] Le Digabel S and Wild SM. "A taxonomy of constraints in black-box simulation-based optimization". In: *Optimization and Engineering* (2023), pp. 1–19.

[33] Lucidi S, Piccialli V, and Sciandrone M. "An algorithm model for mixed variable programming". In: *SIAM Journal on Optimization* 15.4 (2005), pp. 1057–1084.

[34] Pourmohamad T. "Combining Multivariate Stochastic Process Models with Filter Methods for Constrained Optimization". PhD thesis. UC Santa Cruz, 2016.

[35] Picheny V et al. "Bayesian optimization under mixed constraints with a slack-variable augmented Lagrangian". In: *Advances in neural information processing systems* 29 (2016).