

Aerothermal Optimization of a Ribbed U-Bend Cooling Channel Using the Adjoint Method

Ping He, Charles A. Mader, Joaquim R. R. A. Martins, and Kevin J. Maki
University of Michigan, Ann Arbor, MI, 48109

Abstract

Aerothermal optimization is a powerful technique for automating the design of turbine internal cooling passages, where both pressure loss and heat transfer are considered. Existing optimization studies commonly adopt gradient-free algorithms, which can handle only a few design variables. However, to enhance heat transfer, internal cooling designs use complex geometries consisting of ribbed serpentine channels, which need to be parameterized by using a large number of design variables. To address this need, we develop herein an approach for aerothermal optimization that uses a gradient-based optimizer in conjunction with a discrete adjoint method to efficiently compute the required gradients with respect to numerous design variables. We apply this approach to the design of a ribbed U-bend channel, which is representative of a section of turbine internal cooling passages. The objective function combines heat transfer and pressure loss as a weighted sum. We find the Pareto front for these two objectives by running five optimizations with different weights. We consider both a rib-free and a ribbed U-bend configuration. For the rib-free configuration, we use 113 design variables to parameterize the U-bend shape. We compare our optimization results with those from gradient-free methods and demonstrate that the proposed method leads to lower pressure loss while enhancing heat transfer. For the ribbed configuration, we use 146 design variables and allow the ribs to change their arrangement independently (shape, height, pitch, and angle). Each rib adopts a requisite arrangement to balance heat transfer and aerodynamics, depending on the local flow conditions. Optimizing the U-bend shape is shown to be more effective for improving overall heat transfer than optimizing the rib arrangement. However, optimizing ribs is more effective for improving local heat transfer. The results demonstrate that the proposed optimization framework has the potential to handle general turbine heat transfer designs, not only for internal cooling but also for other design problems, such as film cooling and jet impingement cooling.

1 Introduction

To maximize thermal efficiency and power output, gas turbine designs typically adopt a turbine-inlet temperature that is hundreds of degrees higher than the upper limit of the turbine metal temperature. Therefore, effective cooling of the turbine blades is critical because it directly impacts the durability and safety of gas turbines. A common way to reduce the turbine metal temperature is to inject cooling air from the root of turbine blades. The cooling air first passes through a ribbed serpentine internal cooling passage and then serves for other cooling activities such as jet impingement and film cooling [1]. The ribbed serpentine passage generates large flow separation and vortices to enhance turbulence mixing and heat transfer. However, the total pressure loss also increases, which degrades the effectiveness of jet impingement and film cooling. In the worst-case scenario, the cooling air has insufficient momentum to mix into the main stream because of a large loss of total pressure, so the hot main stream is ingested into the turbine blades, resulting in severe damage. Therefore, we need to balance heat transfer and pressure loss in the design of turbine internal passages [1]. Aerothermal design optimization provides an automated way to explore designs and find the optimal combination of these two objectives.

Owing to the complex flow conditions in ribbed serpentine internal cooling passages, aerothermal design has relied heavily on experiments [2–10]. Han et al. [1] wrote a comprehensive review on the topic and summarized how passage shape, rib arrangement, Reynolds number, and rotation affect aerothermal performance. In recent years, computational fluid dynamics (CFD) has become a useful tool that provides valuable insights for turbine heat transfer design. This advance is partially due to the increase in computing power, which has enabled the simulation of a full ribbed serpentine cooling passage including all details [11–13]. More importantly, heat transfer CFD results have become more accurate and reliable, as shown by recent comparisons between CFD and experiments for internal cooling passage heat transfer [13–17].

Numerous studies have used CFD to understand flow and heat transfer characteristics and to improve the design of internal cooling passages. Early studies first used ribbed straight channels [18–21] and, later, more complicated configurations were considered, such as ribbed U-bend channels with rotation [22–24]. These studies focused on evaluating how various turbulence models, discretizations, and wall functions affect CFD performance. With more sophisticated CFD models, researchers evaluated the impact of passage shape and rib arrangement based on numerical results. For instance, Wang and Chyu [25] numerically investigated the secondary flow structures for three U-bend turning configurations. Su et al. [26] evaluated how three U-bend aspect ratios affect aerothermal performance. Saha and Acharya [27] simulated a U-bend channel and compared the aerothermal performance of nine different bend shapes. Finally, Gao et al. [28] improved the heat transfer performance by manually exploring eight different combinations of rib orientation, rib angle, and channel shape.

Although manually exploring design options provides useful insights, it is a time-consuming process and the improved design can be suboptimal given the limited samples. To further increase the efficiency of CFD-based design, we can automate the design

process by using an optimization algorithm. The CFD-based optimization approach has been demonstrated to effectively reduce the duration of the design process while achieving the same design quality as sophisticated human-supervised design tools [29]. Given this advantage, the interest in CFD-based optimization has been increasing.

We can divide optimization algorithms into two broad categories: gradient-free and gradient-based. Gradient-free algorithms use the values of the objective and constraint functions to navigate the design space, so they can be used when the CFD solver is a black box. This feature makes gradient-free algorithms easy to use, which is one of the reasons that they have been widely adopted in internal cooling optimization studies [13, 30–36]. The major limitation of the gradient-free algorithms is that their computational cost grows exponentially with respect to the number of design variables [37, 38]. Therefore, the aforementioned studies used only a small number of design variables (between 3 and 26) to parameterize the geometry of internal cooling passages [13, 30–36].

Since the geometry configuration of internal cooling passages is complex, they should be parameterized by using a large number of design variables to improve the performance to the maximum extent possible. In this paper, we address this need by combining a gradient-based optimization algorithm with an efficient method for computing the required gradients. We use the adjoint method [39, 40] to efficiently compute the derivatives because its computational cost does not depend on the number of design variables. The coupled gradient-based optimization and adjoint derivative computation approach has been widely used in design optimization for aircraft [41–47], cars [48–50], gas turbines [51–54], wind turbines [55, 56], ships [57, 58], and hydrofoils [59, 60], as well as in multidisciplinary design optimization [61, 62], such as aerostructural [63–65], aeropropulsive [66, 67], and hydrostructural [60, 68] optimization.

Given its usefulness and the applications cited above, it is surprising that the adjoint method was only recently adopted in a handful of turbine cooling optimization studies. For example, Zhang et al. [69] developed an adjoint solver to treat aerothermal design problems for a transonic turbine cascade. Gkaragkounis et al. [70] optimized the conjugate heat transfer design for a two-dimensional turbine by using their coupled adjoint solver. Verstraete et al. [71] minimized the pressure loss in a turbine internal cooling passage by using the adjoint method. Finally, Hayek et al. [72] used the adjoint method to optimize the eddy viscosity prediction of the k - ω shear stress transport (SST) turbulence model for an internal cooling passage.

Adjoint methods have not been adopted as fast as expected for turbine cooling applications because they require the source code for the CFD solver and because a significant effort is required to develop the adjoint code. Turbine cooling optimization studies commonly rely on commercial CFD solvers whose source codes are not publicly available.

Another difficulty arises because of the amount of separated flow that is inherent in turbine cooling applications. As elaborated in Sec. 3.3, optimization results exhibit large regions of separated flow to enhance heat transfer. This is not the case for most of the aircraft optimization studies, where drag minimization naturally eliminates flow separation. The flow separation and the associated unsteadiness cause CFD to converge poorly. This violates the assumption in the derivation of the adjoint equation that the flow residuals are well converged (see Sec. 2.3.1). Consequently, the accuracy of

the adjoint derivatives is degraded. Moreover, the poorly converged flow results in ill-conditioned Jacobians, and the adjoint equations become very stiff to solve [53, 73, 74], which eventually causes the optimization to abort.

In previous work, we applied gradient-based aerothermal optimizations to a rib-free U-bend cooling channel, which is representative of a simplified turbine internal cooling configuration [75]. To alleviate the adverse impact of flow separation on the optimization, we used the first-order discretization scheme to stabilize both flow and adjoint computations. In the present study, we enhance the robustness of our adjoint solver by using an improved preconditioning strategy (see details in Sec. 2.3.3), which allows us to use the second-order scheme for a more accurate flow simulation and adjoint derivative computation and to consider a more complex internal cooling configuration—a ribbed U-bend channel. Moreover, we evaluate how CFD configurations (mesh density and turbulence model) affect the simulations and optimizations and provide detailed flow analyses to better interpret the optimization results.

The objective of this study is to demonstrate the advantage of using the gradient-based optimization approach for turbine internal cooling design. We use the U-bend channel developed at the Von Karman Institute (VKI) for Fluid Dynamics [76] as our baseline configuration. We parameterize the U-bend by using 113 design variables, which represents significant freedom for geometry modification. We use the weighted heat transfer and pressure loss as the objective function and run five optimizations to construct a Pareto front with different weights. We then evaluate the benefit of gradient-based optimization by comparing our results with those optimized by the gradient-free methods. To further demonstrate the power of the proposed optimization framework, we add nine squared ribs on the top wall of the U-bend and use 146 design variables to parameterize the rib design. We allow all the ribs to change their arrangement independently (shape, height, pitch, and angle), as opposed to forcing them to have the same arrangement (a common treatment adopted in gradient-free optimization studies).

The remainder of this paper is structured as follows: In Sec. 2, we introduce the proposed optimization framework and the theoretical background of its components. The aerothermal optimization results and the detailed flow analysis are described in Sec. 3, followed by our conclusions in Sec. 4.

2 Method

In this section, we first introduce the overall adjoint framework for gradient-based optimization (DAFoam) and then briefly discuss the theoretical background of its components and their interaction. Finally, we elaborate on the implementation of the most important module: the adjoint derivative computation. We use the discrete adjoint approach because its derivatives are consistent with the flow simulations, as shown in our previous studies [49, 77].

2.1 DA Foam: A discrete adjoint framework for gradient-based optimization

As mentioned in the introduction, we use gradient-based optimization algorithms with derivatives ($df/d\mathbf{x}$) computed by using the adjoint method. In these total derivatives, f is the function of interest; in our case, f can be the total pressure loss, the Nusselt number, or a weighed combination of both. The design variables \mathbf{x} are shape variables that parameterize the surface geometry and ribs.

In our previous work, we developed an aerodynamic design optimization framework based on OpenFOAM using the adjoint method [49]. In the present study, we extend this framework to include heat transfer design. We first introduce the overall optimization framework and then briefly describe the theoretical background of its modules and their interaction.

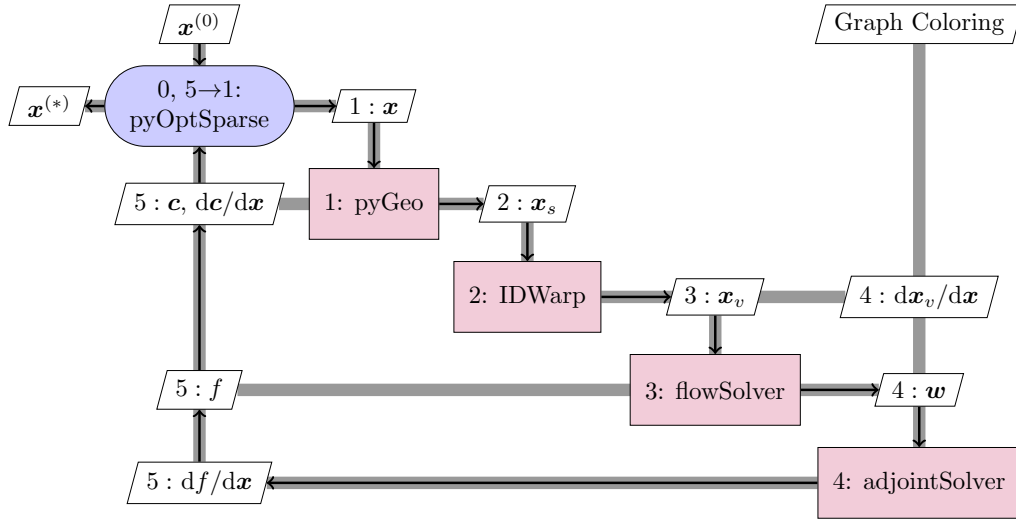


Figure 1: Process and data flow for the DA Foam optimization framework [49]: \mathbf{x} is the vector of design variables, $\mathbf{x}^{(0)}$ and $\mathbf{x}^{(*)}$ are the baseline and optimized designs, respectively, \mathbf{c} are the geometric constraints, \mathbf{w} are the state variables, f represents the objective and constraint functions, and \mathbf{x}_s and \mathbf{x}_v are the coordinates of the design surface and volume mesh, respectively.

Figure 1 illustrates the process and data flow for the DA Foam optimization framework using the extended design structure matrix representation developed by Lambe and Martins [78]. The diagonal and off-diagonal nodes are the modules and data, respectively. The black lines represent the process flow in the optimization, whereas the data flow is shown by the thick gray lines. The number in each node represents the execution order.

The DA Foam framework comprises two layers: OpenFOAM and Python [49]. The OpenFOAM layer contains three solvers: `flowSolver`, `adjointSolver`, and `coloringSolver`. `flowSolver` is based on OpenFOAM’s incompressible flow solver `simpleFoam`. To consider heat transfer, we add a scalar transport equation for temperature.

Instead of using the OpenFOAM’s built-in continuous adjoint solver [79], we developed a discrete adjoint solver for aerodynamic design optimization in our previous work [80]. Recently, we extended the adjoint capability for heat transfer by using an object-oriented adjoint code structure [77]. It lowers the barrier for time-consuming adjoint development and enables us to rapidly implement an adjoint solver for any set of partial differential equations. This salient feature allows DAfoam to handle multi-disciplinary design optimization, including aerodynamics, hydrodynamics, heat transfer, and structures. The `coloringSolver` module uses a parallel, heuristic graph-coloring scheme to accelerate the partial derivative computation for the `adjointSolver` module. The details of these three solvers are presented in Secs. 2.2 and 2.3.

The Python layer controls the three solvers in the OpenFOAM layer and also calls the external Python modules `pyGeo`, `IDWarp`, and `pyOptSparse` to perform optimization.

`pyGeo` uses the free-form deformation (FFD) approach to parameterize the surface geometry and surface mesh [81].¹ The surface mesh is first embedded into a tri-variate B-spline volume and then the surface mesh is manipulated by moving the surface points of the given volume. The benefit of using the tri-variate B-spline volume is that we can easily compute the gradient of any FFD point. We also use `pyGeo` to compute the values and derivatives of the geometric constraints (\mathbf{c} and $d\mathbf{c}/d\mathbf{x}$). Figure 2 shows an example of a FFD and control points for a U-bend channel.

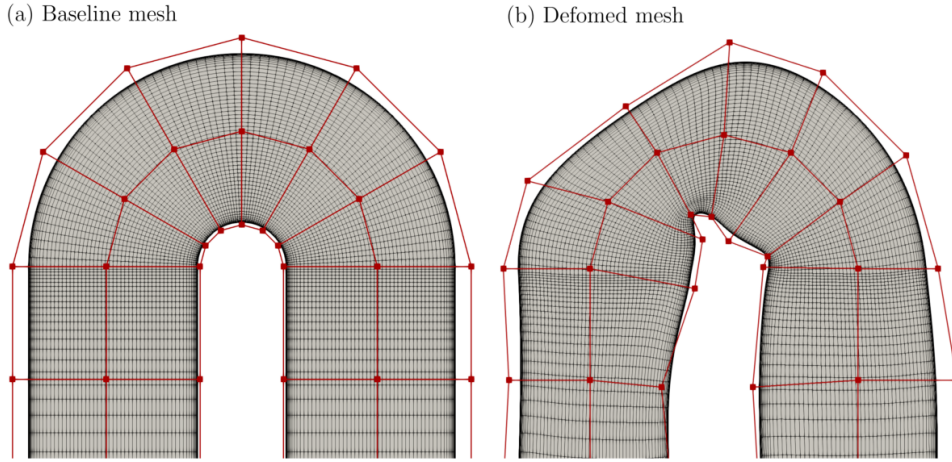


Figure 2: Example of surface mesh deformation for the U-bend channel obtained by using the FFD control points (red squares) in `pyGeo` and the corresponding volume mesh deformation from `IDWarp`.

`IDWarp` uses the analytic inverse distance algorithm proposed by Luke et al. [82] to deform the volume mesh.² One advantage of this algorithm is that it supports both structured and unstructured meshes. It also does a better job of preserving the mesh orthogonality near the wall compared with methods based on radial basis functions.

¹<https://github.com/mdolab/pygeo>

²<https://github.com/mdolab/idwarp>

Figure 2 shows an example of baseline and deformed volume meshes for the U-bend channel.

`pyOptSparse` is a Python module for formulating and solving large-scale constrained optimization problems.³ `pyOptSparse` is extended from `pyOpt` [83] and provides APIs for defining the design variables and constraint and objective functions, as well as an interface for several optimization packages. We use the SNOPT [84] optimizer in this study. SNOPT implements the sequential quadratic programming algorithm for the optimization, which uses a quasi-Newton method to solve the nonlinear equations resulting from the Karush–Kuhn–Tucker optimality conditions.

To summarize, the gradient-based optimization starts with an initial design variable $\mathbf{x}^{(0)}$ and a baseline U-bend mesh and passes them to `pyGeo`. Based on $\mathbf{x}^{(0)}$, `pyGeo` deforms the surface mesh and passes the updated surface mesh coordinates \mathbf{x}_s to `IDWarp`, while also computing the value and derivative of the geometric constraints. To avoid negative volume and preserve mesh quality, `IDWarp` deforms the volume mesh and passes the updated volume mesh coordinates \mathbf{x}_v to `flowSolver`, which then simulates the flow field and passes the converged state variables \mathbf{w} to `adjointSolver`. `flowSolver` also computes the constraint and objective function f . Based on \mathbf{w} , `adjointSolver` uses the adjoint approach to efficiently compute the total derivatives $df/d\mathbf{x}$. To accelerate the computation of the partial derivatives, `adjointSolver` uses the graph-coloring information provided by `coloringSolver`. Finally, `pyOptSparse` receives the values and derivatives of the objective and constraint functions (f , $df/d\mathbf{x}$, \mathbf{c} , and $d\mathbf{c}/d\mathbf{x}$), solves the sequential quadratic programming problem, updates the design variables \mathbf{x} , and passes them to `pyGeo` again. This process is repeated until the optimization converges. Data exchange between the OpenFOAM and Python layers is done through input and output files. The most expensive steps are the flow simulation and adjoint derivative computation; the file-based data exchange accounts for less than 1% of the total computational time.

2.2 Flow simulation

In this paper, we simulate three-dimensional, steady turbulent flow with heat transfer by using a modified version of OpenFOAM’s standard solver `simpleFoam` (primal solver). The flow is governed by the Navier–Stokes equations and the temperature equation:

$$\nabla \cdot \mathbf{U} = 0, \quad (1)$$

$$\nabla \cdot (\mathbf{U}\mathbf{U}) + \nabla p - \nabla \cdot [(\nu + \nu_t)(\nabla \mathbf{U} + \nabla \mathbf{U}^T)] = 0, \quad (2)$$

$$\nabla \cdot (T\mathbf{U}) - (\alpha + \alpha_t)\Delta T = 0, \quad (3)$$

where p is the pressure, T is the temperature, $\mathbf{U} = [u, v, w]$ is the velocity vector, ν is the kinematic viscosity, ν_t is the turbulent eddy viscosity, α is the thermal diffusivity, and α_t is the turbulent thermal diffusivity. The viscosity and thermal diffusivity are connected through $Pr = \nu/\alpha = 0.7$, and $Pr_t = \nu_t/\alpha_t = 0.7$. We ignore body forces and internal heat sources.

³<https://github.com/mdolab/pyoptsparse>

To connect the turbulent viscosity to the mean flow variables, we adopt the Spalart–Allmaras model,

$$\nabla \cdot (\mathbf{U}\tilde{\nu}) - \frac{1}{\sigma} \nabla \cdot [(\nu + \tilde{\nu})\nabla\tilde{\nu}] + \frac{1}{\sigma} C_{b2} |\nabla\tilde{\nu}|^2 - C_{b1} \tilde{S}\tilde{\nu} + C_{w1} f_w \left(\frac{\tilde{\nu}}{d}\right)^2 = 0. \quad (4)$$

The turbulent eddy viscosity ν_t is computed from $\tilde{\nu}$ via

$$\nu_t = \tilde{\nu} \frac{\chi^3}{\chi^3 + C_{v1}^3}, \quad \chi = \frac{\tilde{\nu}}{\nu}. \quad (5)$$

Spalart and Allmaras [85] provide a detailed description of the terms and parameters in Eq. (4). We include all the turbulence variables in our adjoint implementation, so we do not assume frozen turbulence, as is typically done when implementing the continuous adjoint approach.

The governing equations (1)–(5) are discretized by using the finite-volume method. More specifically, we use the SIMPLE algorithm (semi-implicit method for pressure-linked equations) to solve these equations in a segregated manner. We first solve the momentum and temperature equation based on the old pressure p and surface flux fields, ϕ . Next, the pressure is updated by solving a pressure Poisson equation, followed by an update for ϕ that uses the Rhie–Chow interpolation scheme. Based on the new p and ϕ , we then update \mathbf{U} such that it satisfies both the mass and momentum equations. Finally, we solve the turbulence equation (4) to update $\tilde{\nu}$.

2.3 Adjoint derivative computation

An efficient adjoint implementation is critical because it directly impacts the overall performance of optimization framework. In this subsection, we first derive the discrete adjoint formulations and then elaborate on the strategies for computing partial derivatives and solving the adjoint equations.

2.3.1 Adjoint equations

As previously mentioned, we use the discrete adjoint method to compute the total derivative $df/d\mathbf{x}$. This assumes that a discretized form of the governing equations (1)–(4) is available through the flow solver, and that, for a given design vector $\mathbf{x} \in \mathbb{R}^{n_x}$, the discretized equations are solved for the flow state variable vector $\mathbf{w} \in \mathbb{R}^{n_w}$ such that

$$\mathbf{R}(\mathbf{x}, \mathbf{w}) = 0, \quad (6)$$

where $\mathbf{R} \in \mathbb{R}^{n_w}$ is the residual vector. The \mathbf{R} and \mathbf{w} vectors contain the residuals and state variables for all mesh cells, respectively. Therefore, Eq. (6) contains nonlinear equations that involve millions of state variables and require specialized iterative solvers.

The functions of interest are then functions of both the design variables and the flow variables; namely,

$$f = f(\mathbf{x}, \mathbf{w}). \quad (7)$$

This computation does not require iteration and is much cheaper than the nonlinear (primal) solution.

In general, we have multiple functions of interest (the objective and multiple design constraints), but in the following derivations we consider f to be a scalar without loss of generality. As shown below, each additional function requires the solution of another adjoint system.

To obtain the total derivative $df/d\mathbf{x}$, we apply the chain rule as follows:

$$\underbrace{\frac{df}{d\mathbf{x}}}_{1 \times n_x} = \underbrace{\frac{\partial f}{\partial \mathbf{x}}}_{1 \times n_x} + \underbrace{\frac{\partial f}{\partial \mathbf{w}}}_{1 \times n_w} \underbrace{\frac{d\mathbf{w}}{d\mathbf{x}}}_{n_w \times n_x}, \quad (8)$$

where the partial derivatives $\partial f/\partial \mathbf{x}$ and $\partial f/\partial \mathbf{w}$ are relatively cheap to evaluate because they only involve explicit computations. The matrix for the total derivative $d\mathbf{w}/d\mathbf{x}$, on the other hand, is expensive because \mathbf{w} and \mathbf{x} are implicitly determined by the residual equations $\mathbf{R}(\mathbf{w}, \mathbf{x}) = 0$.

To solve for $d\mathbf{w}/d\mathbf{x}$, we can apply the chain rule to \mathbf{R} . We then use the fact that the governing equations should always hold, independently of the values of design variables \mathbf{x} . Therefore, the total derivative $d\mathbf{R}/d\mathbf{x}$ must be zero, so

$$\frac{d\mathbf{R}}{d\mathbf{x}} = \frac{\partial \mathbf{R}}{\partial \mathbf{x}} + \frac{\partial \mathbf{R}}{\partial \mathbf{w}} \frac{d\mathbf{w}}{d\mathbf{x}} = 0. \quad (9)$$

Rearranging the above equation, we get the linear system

$$\underbrace{\frac{\partial \mathbf{R}}{\partial \mathbf{w}}}_{n_w \times n_w} \underbrace{\frac{d\mathbf{w}}{d\mathbf{x}}}_{n_w \times n_x} = - \underbrace{\frac{\partial \mathbf{R}}{\partial \mathbf{x}}}_{n_w \times n_x}. \quad (10)$$

We can solve a column of $d\mathbf{w}/d\mathbf{x}$ by using the same column of $d\mathbf{R}/d\mathbf{x}$ as the right-hand-side. We then repeatedly solve the linear equation n_x times until all the columns are computed in the $d\mathbf{w}/d\mathbf{x}$ matrix. Finally, we substitute $d\mathbf{w}/d\mathbf{x}$ into Eq. (8) to compute the total derivative $df/d\mathbf{x}$ for any objective function f . This approach is known as the *direct* method, and its computational cost is proportional to the number of design variables.

Alternatively, we can substitute the solution for $d\mathbf{w}/d\mathbf{x}$ from Eq. (10) into Eq. (8) to get

$$\underbrace{\frac{df}{d\mathbf{x}}}_{1 \times n_x} = \underbrace{\frac{\partial f}{\partial \mathbf{x}}}_{1 \times n_x} - \overbrace{\underbrace{\frac{\partial f}{\partial \mathbf{w}}}_{1 \times n_w} \underbrace{\frac{\partial \mathbf{R}^{-1}}{\partial \mathbf{w}}}_{n_w \times n_w}}^{\psi^T} \underbrace{\frac{\partial \mathbf{R}}{\partial \mathbf{x}}}_{n_w \times n_x}. \quad (11)$$

Typically, we do not actually find the inverse of the Jacobian $\partial \mathbf{R}/\partial \mathbf{w}$ explicitly. Instead, we solve the corresponding linear system with the appropriate right-hand-side vector.

The direct method consists in solving with $-\partial \mathbf{R}/\partial \mathbf{x}$ as the right-hand side. Alternatively, we can transpose the Jacobian and solve with $[\partial f/\partial \mathbf{w}]^T$ as the right-hand side,

which yields the *adjoint equations*,

$$\underbrace{\frac{\partial \mathbf{R}^T}{\partial \mathbf{w}}}_{n_w \times n_w} \underbrace{\boldsymbol{\psi}}_{n_w \times 1} = \underbrace{\frac{\partial f^T}{\partial \mathbf{w}}}_{n_w \times 1}, \quad (12)$$

where $\boldsymbol{\psi}$ is the *adjoint vector*. We can then compute the total derivative by substituting the adjoint vector into Eq. (11):

$$\frac{df}{d\mathbf{x}} = \frac{\partial f}{\partial \mathbf{x}} - \boldsymbol{\psi}^T \frac{\partial \mathbf{R}}{\partial \mathbf{x}}. \quad (13)$$

For each function of interest, we solve the adjoint equations only once because the design variable is not explicitly present in Eq. (12). Therefore, its computational cost is independent of the number of design variables. This approach is also known as the *adjoint method* and its computational cost is proportional to the number of objective functions.

For turbine cooling design, the adjoint method is advantageous because we typically have a few functions of interest but use several hundred design variables.

To summarize, a discrete adjoint method consists of four major steps: First, we compute the partial derivatives $[\partial \mathbf{R} / \partial \mathbf{w}]^T$ and $[\partial f / \partial \mathbf{w}]^T$. Next, we solve the linear equation (12) for the adjoint vector $\boldsymbol{\psi}$. Finally, we compute $\partial \mathbf{R} / \partial \mathbf{x}$ and $\partial f / \partial \mathbf{x}$ and use Eq. (13) to compute the total derivative $df / d\mathbf{x}$. An effective adjoint implementation requires efficient computation of these partial derivatives, and an efficient solver for the adjoint equations. We elaborate the details of these two tasks in the following subsections.

2.3.2 Partial derivative computation

We use the finite-difference method to compute $[\partial \mathbf{R} / \partial \mathbf{w}]^T$ and $[\partial f / \partial \mathbf{w}]^T$. However, naively computing these partial derivatives by using finite differences requires calling the residual functions n_w times—once for each column in $[\partial \mathbf{R} / \partial \mathbf{w}]^T$ and $[\partial f / \partial \mathbf{w}]^T$. This becomes computationally prohibitive for three-dimensional problems because n_w is at least a few million for useful problems.

To circumvent this issue, we accelerate the partial derivative computation by using graph coloring. In this approach, we group all the states (columns) into different colors such that, for each color, no two states impact the same residual (row). By using the coloring information, we can simultaneously perturb multiple columns that have the same colors and compute their partial derivatives by calling the residual function only once. As an example, consider a five-point-stencil Jacobian matrix obtained from a 5×5 two dimensional mesh, as shown in Fig. 3. If we successively perturb each state (column) and compute the partial derivatives by using the forward difference method, we need 26 residual function evaluations. With graph coloring, we can simultaneously perturb columns that have the same color because they affect distinct residuals. Thus, the entire Jacobian is computed by calling the residual functions only six times.

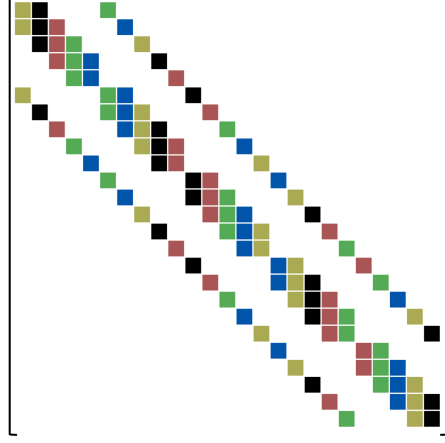


Figure 3: Example of graph coloring for a five-point-stencil Jacobian matrix obtained from a 5×5 two-dimensional mesh. Naively computing the Jacobian by using finite differencing (column by column) would require 26 residual function evaluations. With graph coloring, we simultaneously perturb columns that have the same color, and we reduce the number of function evaluations to six.

The actual Jacobian matrices for the three-dimensional Navier–Stokes equations are much more complex than the five-point-stencil matrix discussed above. We developed a heuristic graph coloring algorithm to compute such Jacobians [49]. The central idea of the algorithm is to tentatively assign colors for the local meshes and then resolve conflicts by exchanging coloring information between the local meshes. Our coloring algorithm is applicable to structured and unstructured meshes and reduces the number of residual evaluations to $\mathcal{O}(1000)$. This number does not depend on the size of the meshes or the number of CPU cores [49].

We do not use the coloring scheme for $\partial \mathbf{R} / \partial \mathbf{x}$ and $\partial f / \partial \mathbf{x}$. Instead, we compute $d\mathbf{x}_v / d\mathbf{x}$ by first perturbing each \mathbf{x} and then deforming the surface and volume meshes by using `pyGeo` and `IDWarp`. We then pass $d\mathbf{x}_v / d\mathbf{x}$ to `adjointSolver` to compute $\partial \mathbf{R} / \partial \mathbf{x}$ and $\partial f / \partial \mathbf{x}$ directly, as shown in Fig. 1.

2.3.3 Adjoint equation solution

After computing $[\partial \mathbf{R} / \partial \mathbf{w}]^T$ and $[\partial f / \partial \mathbf{w}]^T$, we solve the adjoint equations (12) by using the PETSc software library (Portable, Extensible Toolkit for Scientific computation) [86–88]. We use the generalized minimal residual (GMRES) method as the top-level linear equation solver and adopt a nested preconditioning strategy. The global preconditioner is based on the additive Schwarz method with one level of overlap. The additive Schwarz method divides the linear system into sub-blocks so that they can be solved in parallel. The local preconditioner in each sub-block is based on incomplete lower and upper (ILU) factorization, where we use one level of extra fill-in to improve its effectiveness.

As mentioned in the introduction, the flow separation in turbine internal cooling passages makes the adjoint equations very stiff to solve. Therefore, we need an effective

preconditioner matrix to enhance the eigenvalue clustering for better convergence. The right-preconditioned adjoint equation can be written as

$$\left(\frac{\partial \mathbf{R}^T}{\partial \mathbf{w}} \left[\frac{\partial \mathbf{R}^T}{\partial \mathbf{w}_{PC}} \right]^{-1} \right) \left(\frac{\partial \mathbf{R}^T}{\partial \mathbf{w}_{PC}} \psi \right) = \frac{\partial f}{\partial \mathbf{w}}^T, \quad (14)$$

where $[\partial \mathbf{R} / \partial \mathbf{w}]_{PC}^T$ is the preconditioner matrix and $([\partial \mathbf{R} / \partial \mathbf{w}]_{PC}^T)^{-1}$ is the approximated inverse of $[\partial \mathbf{R} / \partial \mathbf{w}]^T$. The preconditioner $[\partial \mathbf{R} / \partial \mathbf{w}]_{PC}^T$ should be an approximation of $[\partial \mathbf{R} / \partial \mathbf{w}]^T$ but easily invertible. To this end, we make approximations to the residuals and their linearizations for $[\partial \mathbf{R} / \partial \mathbf{w}]_{PC}^T$. We compute the convective terms by using the first-order scheme. This reduces the stiffness of the preconditioner matrix. We then shrink the residual stencils by reducing the maximum level of connected states for p and ϕ residuals from three to two. We also ignore the turbulence-production term to further decrease the linear system stiffness, especially for the viscous layer near the wall when using a $y^+ = 1$ mesh. Finally, to ensure a better diagonal dominance for both $[\partial \mathbf{R} / \partial \mathbf{w}]_{PC}^T$ and $[\partial \mathbf{R} / \partial \mathbf{w}]^T$, we normalize the cell and face residuals by their volume and face area, respectively. We also normalize each column of state Jacobians by their reference values at the far field [49]. The above preconditioning strategy reliably converges the adjoint equations, even in the case of large flow separation regions.

3 Results and Discussion

In this section, we start by describing the flow simulation configurations and verify our CFD solver by comparing our simulation results with the experimental data reported by Coletti et al. [89]. Next, we elaborate on the optimization setup and show the results for rib-free and ribbed optimizations with a range of trade-offs between aerodynamic and heat transfer performance. In addition, we compare our optimization results with the results of previous work [33, 90].

3.1 Flow simulation configurations and solver verification

As mentioned in the introduction, we use the Von Karman Institute U-bend channel [76] as our baseline configuration. As shown in Fig. 4, the U-bend has a square cross section with a hydraulic diameter $D_h = 0.075$ m. According to Verstraete [76], the simulation domain consists of an upstream section going from $x = 0$ (inlet) to $x = 10D_h$, a 180°-bend section, and a downstream section from $x = 10D_h$ back to $x = 0$ (outlet). Because the channel is symmetric with respect to the $z = 0$ plane, we simulate only half of the geometry. We generate the structured mesh shown in Fig. 5(a), which has 409 600 cells for the rib-free case. For the ribbed case, we add nine squared ribs on the top wall, which are aligned orthogonally to the flow direction and have a height corresponding to 5% of D_h . The mesh for the ribbed case has 1 339 200 cells and is shown in Fig. 5(b). The averaged y^+ for these two cases is 0.9. We use ANSYS ICEM to construct the U-bend geometry and generate the meshes. To ensure that these mesh densities are sufficient, we

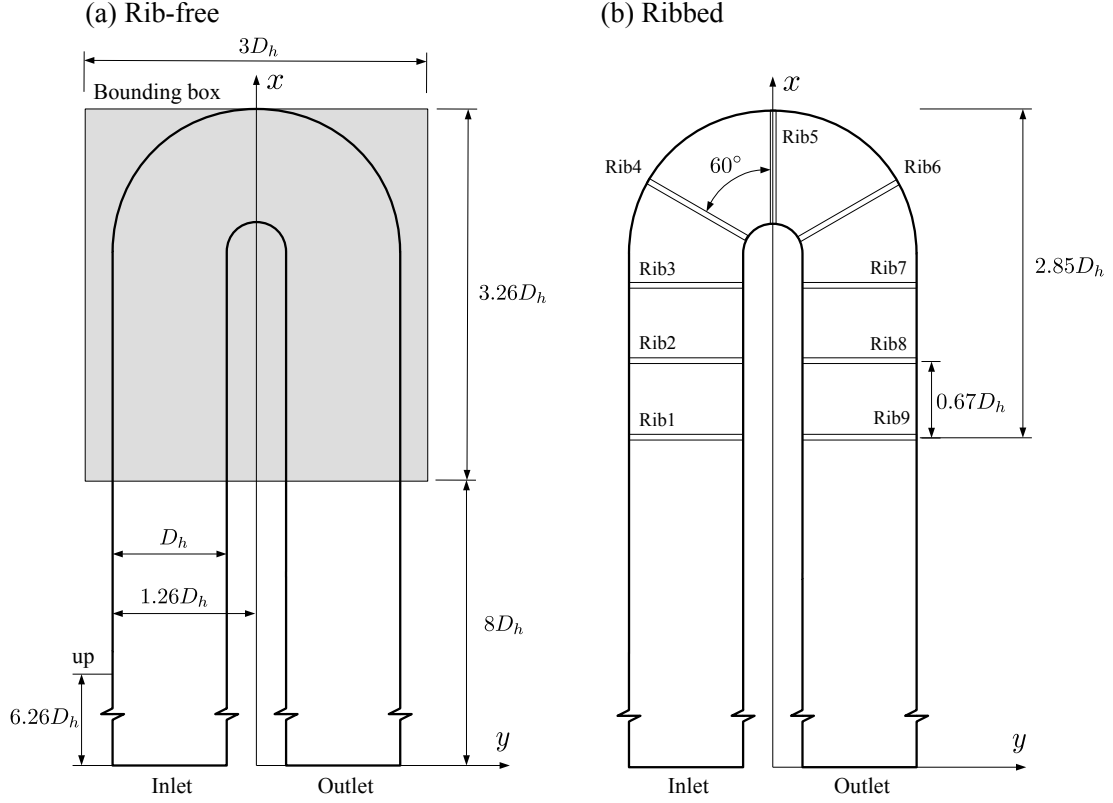


Figure 4: Geometry dimensions for the rib-free and ribbed U-bend configurations, which have a hydraulic diameter of $D_h = 0.075$ m. For the rib-free case, we confine the U-bend surfaces within the shaded bounding box and impose the limit $-0.6D_h \leq z \leq 0.6D_h$. For the ribbed case, we have nine squared ribs on the top wall with rib height $H_r/D_h = 0.05$.

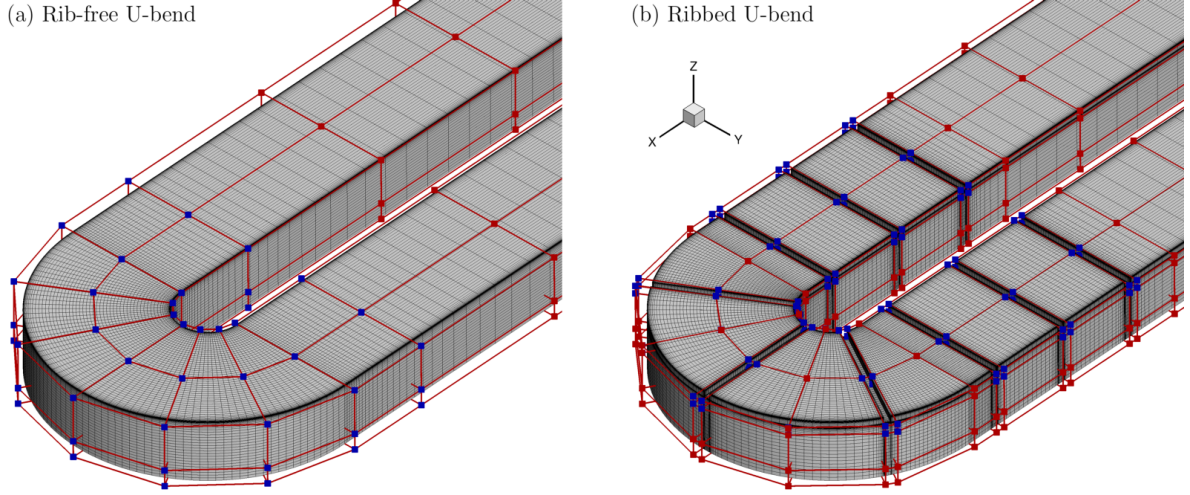


Figure 5: Structured mesh for the rib-free and ribbed U-bend channels. The red and blue squares are the FFD points that manipulate the U-bend surfaces. We select only the blue FFD points as the design variables and keep the red FFD points fixed during the optimizations.

evaluate how mesh size affects the simulation and optimization results, which is detailed in the appendix.

To be consistent with Verstraete [76], we impose a velocity profile at the inlet to reproduce the boundary layer development that occurs in the experiment. The reference bulk velocity is 8.4 m/s, and the Reynolds number is 4.2×10^4 , based on D_h . The inlet temperature is 293.15 K, whereas the wall temperature is fixed at 303.15 K to produce a 10 K temperature difference to drive the heat transfer. The symmetry boundary condition is imposed at the $z = 0$ plane and a zero-gradient boundary condition is imposed at the outlet. We use the second-order linear upwind scheme for the convective terms and a central differential scheme for the diffusion terms. Turbine heat transfer simulations are highly sensitive to the turbulence model. Spalart–Allmaras, k - ε , and k - ω SST are three popular turbulence models that have been used in recent turbine heat transfer studies [12, 15, 27, 28, 36, 69, 71, 90, 91]. We evaluated the impact of these turbulence models on the simulation results and find that the Spalart–Allmaras model is the best option for our case (see appendix). Therefore, we use the Spalart–Allmaras model in this work.

We quantify the aerodynamic loss by the total pressure loss coefficient C_{PL} , which is defined as

$$C_{PL} = \frac{p_0^{\text{up}} - p_0^{\text{down}}}{0.5\rho U_0^2}, \quad (15)$$

where $U_0 = 8.4$ m/s is the reference velocity and p_0 is the averaged total pressure. The superscripts “up” and “down” denote the upstream and downstream locations where we compute the total pressure. We follow the experimental configuration of Coletti et al. [89] and compute the upstream total pressure at $x = 6.26D_h$ and the downstream total pressure at the outlet surface (see Fig. 4 for reference).

To quantify the heat transfer, we use the Nusselt number, which is defined as

$$Nu = \frac{qD_h}{(T_w - T_B)k}, \quad (16)$$

where q is the heat flux, k is the thermal conductivity, and T_w and T_B are the wall temperature and bulk temperature, respectively. To compute the bulk temperature, we assume a linear increase of the mainstream temperature from the inlet to outlet. Next, T_B is linearly interpolated based on the normalized streamwise location. The Nusselt number averaged over the U-bend walls is then computed as

$$\overline{Nu} = \frac{1}{S} \sum_{i=1}^N Nu_i dS_i, \quad (17)$$

where the subscript i denotes the index of a discrete surface mesh, dS is the area of surface mesh, and S is the total surface area. The averaging is performed on all the U-bend and rib walls, starting from the “up” location to the outlet. To normalize, we use the reference Nusselt number Nu_0 computed by using the Dittus–Boelter correlation, $Nu_0 = 0.023Re^{0.8}Pr^{0.4} = 99.6$.

We first validate our CFD solver by comparing the simulated velocity field with the experimental data measured by Coletti et al. [89], as shown in Fig. 6. Our definition of z/D_h differs from that of Coletti et al. [89] because of the symmetry boundary condition that we use. The inner-wall separation bubble is well captured and the velocity distributions at the symmetry ($z/D_h = 0$) and upper ($z/D_h = 0.47$) planes are consistent with the experiments. In terms of averaged quantities, the simulated C_{PL} (1.20) is 16.5% higher than the value of 1.03 measured by Coletti et al. [89]. A similar overestimation also occurred in other U-bend studies [33, 71], which we suspect is partially because we use a reduced simulation domain such that the three-dimensional velocity distribution at the inlet is not identical to the distribution in experiments.

We validate the capability of our CFD solver to simulate heat transfer, as shown in Fig. 7. Overall, the agreement for heat transfer is poorer than for velocity. In the experiments, there is a high Nu in the downstream section of the U-bend, which is caused by flow separation and enhanced turbulence mixing. The simulated Nu captures this trend but overestimates the magnitude of Nu at the top wall. Moreover, the experiments have a high Nu at the corner between the inner and top walls in the downstream section, which is not quantitatively captured by our simulation. Overall, the simulated \overline{Nu}/Nu_0 is 1.43, which is 13.3% lower than the experimental value of 1.65. Given that the measurement uncertainty for \overline{Nu} ranges from 7% to 11% [89], our CFD prediction is acceptable.

3.2 Formulation and convergence of optimization problem

As mentioned in Sec. 2.1, we use the FFD points to manipulate the geometries of the U-bend and its ribs. Figure 5 shows the FFD setup for the rib-free and ribbed cases. Only the blue FFD points are allowed to move, whereas the red FFD points are fixed during the optimization process.

For the rib-free case, we use 63 FFD points, as shown in Fig. 5(a). We allow the top FFD points to move in the z direction and the side FFD points to move in both the x and y directions. In total, we have 113 degrees of freedom (i.e., design variables) for this case. According to Verstraete [76], for structural reasons, the U-bend surfaces should be within the bounding box: $8D_h \leq x \leq 11.26D_h$, $-1.5D_h \leq y \leq 1.5D_h$, and $-0.6D_h \leq z \leq 0.6D_h$ [see Fig. 4(a) for reference]. This is done by setting proper lower and upper bounds for each FFD displacement. To ensure a smooth transition, we impose 29 linear constraints such that the surface slope with respect to z is zero at the symmetry plane. In addition, we set nine linear constraints to avoid a potential negative mesh volume caused by the intersection of inner walls. Although there are other design considerations in internal cooling designs (e.g., curvature constraints to lower manufacturing cost [49]), they are beyond the scope of this paper and are left to future work. Table 1 summarizes the optimization configuration.

For the ribbed case, we use 108 FFD points to morph the rib geometries, while the U-bend channel remains unchanged. Table 2 summarizes the formulation of the optimization problem. For each rib, we use two FFD points in the streamwise direction, three FFD points in the spanwise direction, and two FFD points in the vertical direction

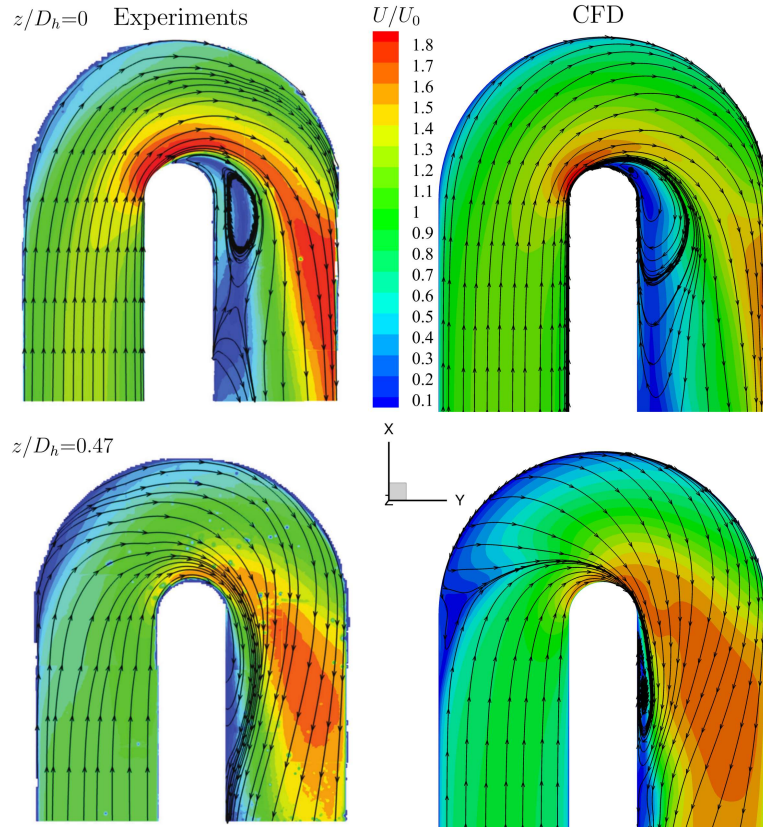


Figure 6: The simulated velocity fields are consistent with the experimental data measured by Coletti et al. [89].

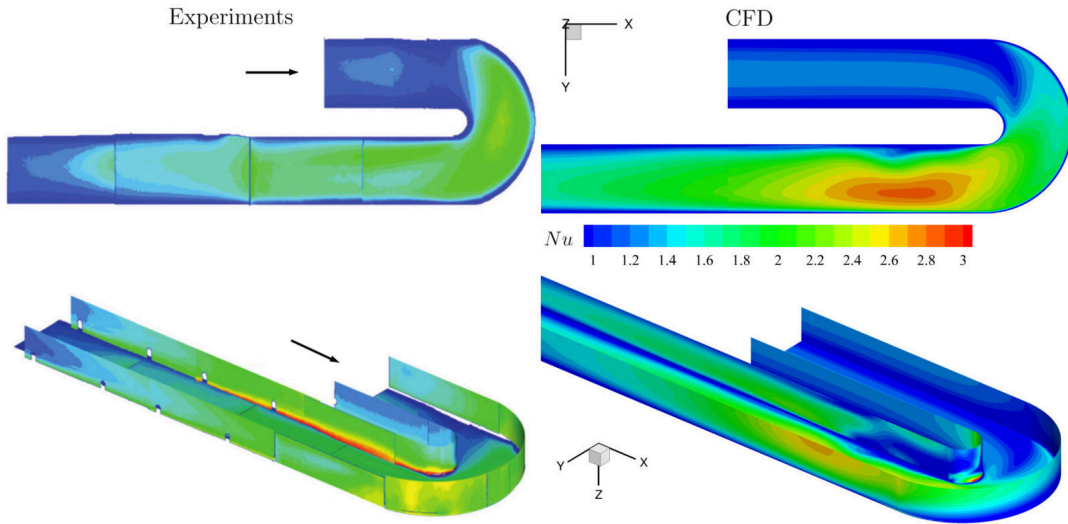


Figure 7: The simulated Nusselt number distribution is qualitatively consistent with the experimental data measured by Coletti et al. [89].

Table 1: Formulation of optimization problem for the rib-free case. The U-bend channel is parameterized by using 113 design variables, and 38 constraints ensure feasible designs.

	Function or variable	Description	Quantity
Minimize	f	Weighted C_{PL} and \overline{Nu}	
with respect to	$\Delta x, \Delta y,$ and Δz	Displacement of FFD points	113
subject to	$g_z^{\text{sym}}=0$	Zero slope at symmetry plane	29
	$\Delta y_{\text{inner}1} + \Delta y_{\text{inner}2} > t_{\text{min}}$	Nonoverlapping inner walls	9
	bound($\Delta x, \Delta y, \Delta z$)	Design-variable bounds to confine the design surfaces within the bounding box	
		Total constraints	38

[see blue FFD points in Fig. 5(b)]. We allow all FFD points to move in the z direction. However, to ensure that the U-bend geometry is unchanged, we restrict the z displacement to be zero for the upper FFD points ($k = 4$). For ribs 1 to 3 and 7 to 9, we allow their FFD points to move in the x direction, whereas for ribs 4 to 6, we allow only the midspan FFD points ($j = 2$) to move in the x and y directions.

In total, we have 146 degrees of freedom (i.e., design variables) for this case. The above FFD setup parameterizes the ribs so that the optimization can change their shape, height, pitch, and angle. None of the variables are linked among the ribs and each rib is allowed to change independently. To avoid potential negative mesh volume, we set 46 linear constraints to link the x and y displacements of the FFD points between $k = 3$ and $k = 4$; the cross section of the ribs remains rectangular. Moreover, we set bounds for the FFD displacement to ensure that each rib does not intersect and that the rib height falls in the range $0.03D_h \leq H_r \leq 0.14D_h$. Changing only the ribs isolates the impact of the U-bend shape and the rib design. A combined U-bend shape and rib optimization can potentially further improve performance, as reported by Gao et al. [28]; however, this topic is beyond the scope of this paper.

Table 2: Formulation of optimization problem for the ribbed case. The ribs are parameterized by using 146 design variables, and 46 constraints ensure feasible designs.

	Function or variable	Description	Quantity
Minimize	f	Weighted C_{PL} and \overline{Nu}	
with respect to	$\Delta x, \Delta y,$ and Δz	Displacement of FFD points	146
subject to	$\Delta x_{k=3}=\Delta x_{k=4}$	Linear constraints to link x	46
	$\Delta y_{k=3}=\Delta y_{k=4}$	and y displacements	
	$-0.533 \leq \Delta x/D_h \leq 0.533$	Design-variable bounds to avoid rib	
	$-0.533 \leq \Delta y/D_h \leq 0.533$	intersection and limit rib height	
	$-0.267 \leq \Delta z/D_h \leq 0.047$		

To consider both aerodynamics and heat transfer, we use a weighted objective func-

tion

$$f = w \frac{C_{PL}}{C_{PL}^b} - (1 - w) \frac{\overline{Nu}}{\overline{Nu}^b}, \quad (18)$$

where the superscript b denotes the value for the baseline design. We run multiple optimizations for different values of w (listed in Table 3). Opt0 corresponds to minimum pressure loss design, whereas Opt4 is the maximum heat transfer design. In Opt4, to improve the robustness of optimization process, we set a small w for C_{PL} instead of using $w = 0$ because, with $w = 0$, the optimizer tends to choose designs that have excessive flow separation and unsteadiness such that the flow and adjoint solvers fail to converge. By using a small w for C_{PL} , we allow the optimizer to maximize heat transfer while maintaining the flow separation at a manageable level.

Table 3: Weights for aerothermal optimizations.

	Opt0	Opt1	Opt2	Opt3	Opt4
w (rib free)	1.0	0.5	0.2	0.1	0.002
w (ribbed)	1.0	0.5	0.3	0.1	0.002

We run all the optimizations on TACC Stampede2. Each job uses 96 CPU cores on 4 Skylake nodes. The Skylake nodes are equipped with Intel Xeon Platinum 8160 CPUs running at 2.1 GHz. The optimizations are terminated when the objective function changes less than 0.01% in five steps. The rib-free optimizations take between 25 and 36 iterations to converge, whereas the ribbed cases require 20 to 26 iterations, as shown in Fig. 8. Each optimization iteration consists of one adjoint derivative computation and one to three flow simulations. The reason for multiple flow simulations (function evaluations) is because the optimizer uses a backtracking line search to improve the design.

For simple cases such as Opt0 (rib-free), the optimizer eliminates the flow separation to minimize C_{PL} (Fig. 11b) and the optimization converges smoothly. However, when increasing the weight for Nu , the optimizer enhances heat transfer by increasing flow separation. As a result, the flow simulation converges poorly, which degrades the accuracy of adjoint derivatives, as explained in the introduction. Due to the degraded derivative accuracy, the line search requires more function evaluations to improve a design and the optimization problem becomes harder to converge. In the worst case scenario, the optimization aborts because the optimizer fails to further improve a design (e.g., Opt4 rib-free).

For the rib-free case, each adjoint computation takes 310 s and uses 61.6 GB of memory, and each flow simulation takes 200 s and uses 10.2 GB of memory. Overall, the optimization converges after 3 to 5 hours, depending on the flow conditions. For the ribbed case, each adjoint computation takes 1100 s and uses 229.2 GB of memory and each flow simulation takes 580 s and uses 17.4 GB of memory. Overall, the optimization converges after 10 to 16 hours. The longer runtime for the ribbed case is due to the larger mesh size and stronger flow separation.

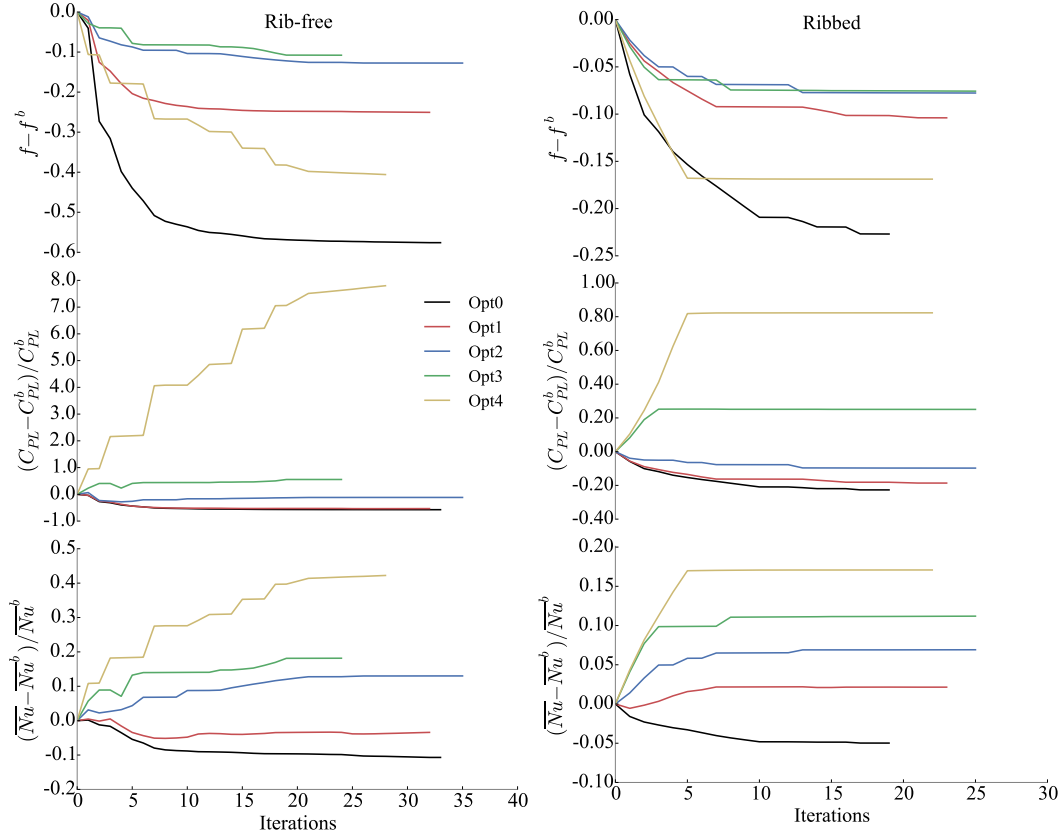


Figure 8: The rib-free optimizations take between 25 and 36 iterations to converge, whereas the ribbed cases require 20 to 26 iterations.

To summarize, the proposed framework can perform aerothermal optimizations with a large number of design variables in an overnight cycle. Such optimizations are not possible with gradient-free methods because their computational cost does not scale well with the number of design variables [38]. As expected, our discrete adjoint implementation requires a relatively large amount of memory because we explicitly store all the Jacobian matrices. However, this is not a severe limitation for small problems with a few million cells because modern high-performance computing systems commonly have more than 128 GB of memory per node. For larger problems with $\mathcal{O}(100 \text{ million})$ cells, the large memory usage becomes a bottleneck, but this memory limitation can be alleviated by using a Jacobian-free adjoint approach [92].

3.3 Design optimization results

In this section, we report the U-bend optimization results with various weights for aerodynamics and heat transfer. We also analyze the flow structures to interpret the optimization results. We start with the rib-free case, followed by the ribbed case.

3.3.1 Rib-free case

Figure 9 shows the Pareto front based on five optimizations with varying weights. For the minimum pressure loss case (Opt0), we obtain a 57.5% C_{PL} reduction, similar to the 58.7% gradient-based optimization reduction reported by Verstraete et al. [71]. Our gradient-based C_{PL} reduction is greater than the 36% reduction reported by Verstraete et al. [90], who used a gradient-free method (a metamodel-assisted differential evolution algorithm). Willeke and Verstraete [93] arrived at a similar conclusion when they compared gradient-based and gradient-free methods for a two-dimensional U-bend channel. This additional C_{PL} reduction is primarily due to the large number of design variables, which provides more design freedom. Because the turbulence mixing is suppressed when minimizing the pressure loss, \overline{Nu} decreases by 10.7%.

When decreasing w to 0.002 (maximum heat transfer case; Opt4), we obtain a 42.0% increase in \overline{Nu} . Again, we obtain a greater increase in \overline{Nu} than the 16.9% bulk heat transfer enhancement reported by Verstraete and Li [33]. However, the Opt4 design generates excessive flow separation and C_{PL} increases by 770.9%.

For the Opt2 case, we obtain a 13.0% increase in \overline{Nu} and a 11.7% reduction in C_{PL} , which indicates that the design is appropriate in terms of both aerodynamics and heat transfer. The other two useful designs are Opt1 and Opt3. For Opt1, we obtain a slightly smaller C_{PL} reduction (53.4%) than for the minimum pressure loss case (57.5% in Opt0). However, the penalty in \overline{Nu} is only 3.4%, which is much lower than the 10.7% penalty in Opt0. Similarly, for Opt3, \overline{Nu} improves by 18.2% with only a 55.5% increase in C_{PL} , as opposed to the 770.9% increase in C_{PL} incurred in the maximum heat transfer case (Opt4). These results highlight the need to consider both aerodynamics and heat transfer in internal cooling designs and to make the appropriate design tradeoffs.

The two competing objectives C_{PL} and \overline{Nu} depend strongly on the velocity distribution in the turbine internal cooling channel [1]. The pressure loss (C_{PL}) is correlated with the velocity shear generated by flow acceleration and separation. As the velocity increases or the flow becomes separated, the friction loss increases and so does C_{PL} . However, the accelerated flow increases the convective heat transfer, and the flow separation enhances the turbulence mixing and results in a greater heat flux transferring to the wall surfaces. Both of these factors contribute to an increased \overline{Nu} .

We now compare in Fig. 10 the baseline and optimized U-bend shapes. Figure 11 shows the corresponding velocity distribution and streamline at the symmetry plane, Fig. 12 shows the local Nusselt number distribution on the U-bend walls, and Fig. 13 shows the average Nusselt number distribution along the streamwise direction.

For the minimum pressure loss case (Opt0), the channel expands in both the z and y directions [Fig. 10(b)], and the magnitude of the velocity decreases [Fig. 11(b)]. A similar trend occurred in previous U-bend-optimization studies [33, 71, 90]. The optimizer creates a smoother U-bend section to reduce the adverse pressure gradient [see Fig. 11(b)]. As a result, the inner wall separation bubble in the downstream section of the baseline design [Fig. 11(a)] is eliminated. These two factors reduce the friction loss, so C_{PL} decreases. However, the trade-off for reduced velocity and separation suppression is that the local Nu decreases on all walls [Figs. 12(b) and 13].

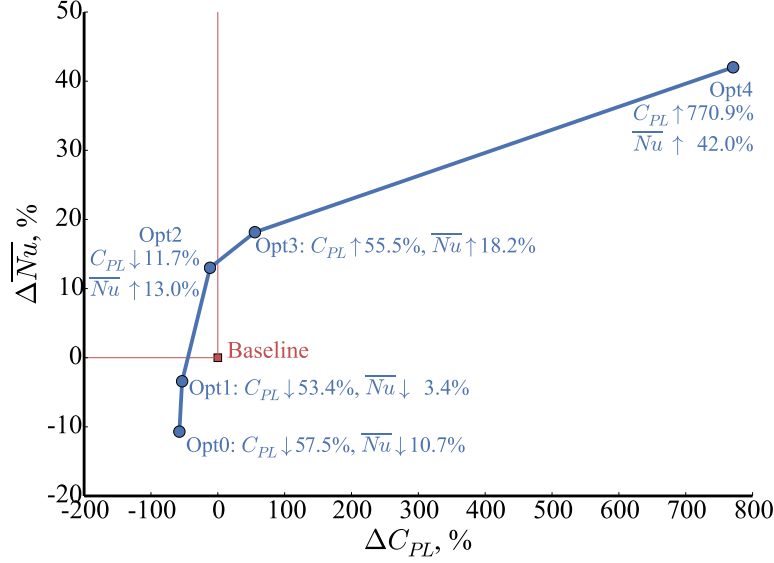


Figure 9: Pareto front for C_{PL} and \overline{Nu} (rib free). For the minimum pressure loss case (Opt0), we obtain 57.5% C_{PL} reduction, whereas for the maximum heat transfer case (Opt4), we increase \overline{Nu} by 42.0%. Opt2 is a balanced design with simultaneous improvement in aerodynamics (11.7% reduction in C_{PL}) and heat transfer (13.0% increase in \overline{Nu}). The reference values are computed from the baseline design without optimization.

For the maximum heat transfer case (Opt4), the channel shrinks in the z direction [Fig. 10(f)], forcing the flow to accelerate [Fig. 11(f)]. In addition, the optimization creates a tongue-like feature at the inner wall, which results in excessive flow separation downstream. The accelerated flow and large separation increase the heat transfer, especially for the outer and top walls [Figs. 12(f) and 13]. As expected, this design has a large C_{PL} due to excessive friction loss.

For Opt2, the optimization creates a shrink-expansion-shrink feature in which the channel shrinks, expands, and then shrinks in the z direction before, within, and after the U-bend section, respectively [Fig. 10(d)]. This intriguing feature effectively increases the velocity magnitude such that the heat transfer is enhanced around the U-bend section [Figs. 12(d) and 13(a)]. In addition, it creates a smoother U-bend transition to minimize flow separation, as shown in Fig. 11(d). Eventually, this balanced design leads to the simultaneous improvement in both aerodynamics and heat transfer.

As mentioned above, the Opt1 and Opt3 designs also balance the aerodynamics and heat transfer. Similarly to Opt0, the Opt1 design expands the channel in both the y and z directions [Fig. 10(c)] to reduce friction loss; however, there is a shrink feature in the bend section, which increases the velocity [Fig. 11(c)] and enhances the heat transfer [Fig. 12(c)]. For Opt3, there is a large flow separation region in the downstream region [Fig. 11(e)], which increases the heat transfer [Fig. 12(e)]. In addition, we observe a shrink-expansion-shrink feature [Fig. 10(e)] similar to Opt2. Compared with Opt4, this feature reduces the velocity in the bend section [Fig. 11(e)] and reduces the friction loss.

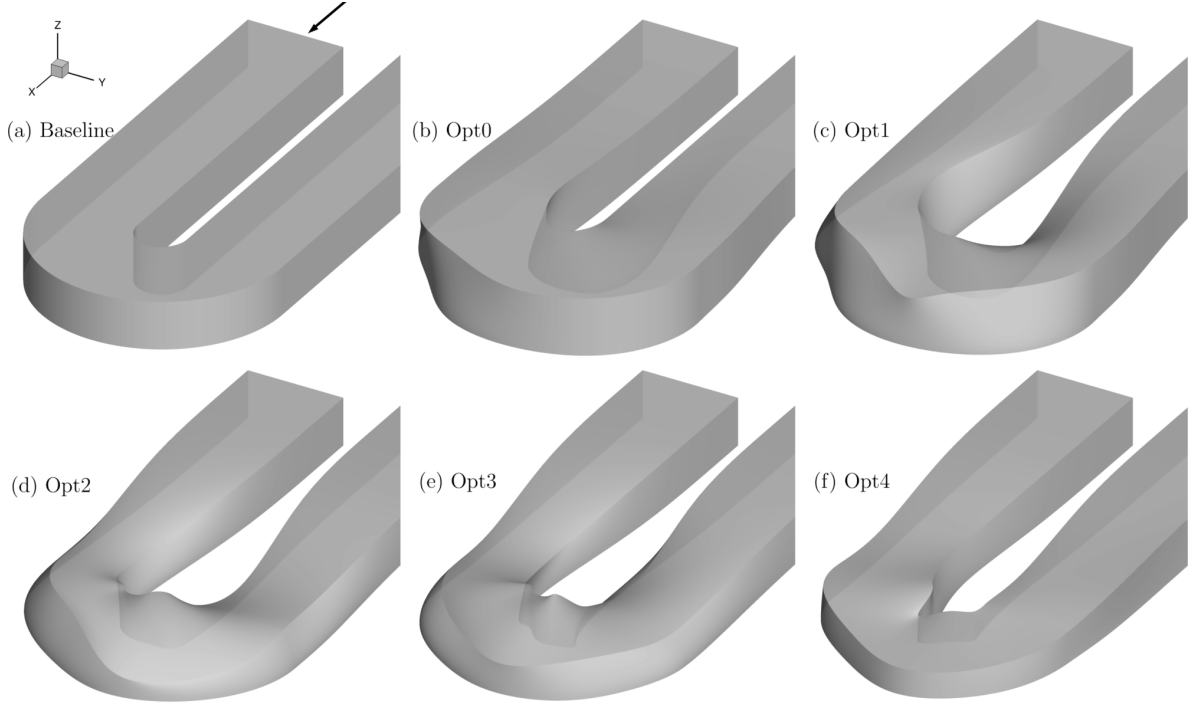


Figure 10: Comparison between baseline and rib-free optimized shapes. For the minimum pressure loss case (Opt0), the channel expands in the bend section, whereas the channel shrinks for the maximum heat transfer case (Opt4). Opt2 exhibits a shrink-expansion-shrink feature.

3.3.2 Ribbed case

Next, we focus on the ribbed U-bend case, where we optimize only the ribs to improve the aerodynamics and heat transfer, while the U-bend channel remains unchanged. In previous studies [12, 13] that used a gradient-free optimization algorithm, the optimization was limited to a small number of design variables such that all the rib design variables were linked, thus having the same shape, height, pitch, and angle. Although these optimization studies led to improved designs, the restricted design freedom limited the potential improvements in performance. In the present study, we overcome this limitation by using 146 design variables to parameterize the ribs, where each rib can have a unique shape (see Sec. 3.2). The goal is to further demonstrate the benefit of using the coupled adjoint derivative computation and gradient-based optimization approach for turbine internal cooling design.

Figure 14 shows the Pareto front constructed from the five optimizations with varying weights. Similarly to the rib-free case, optimizing only the aerodynamics incurs a large penalty for heat transfer (Opt0), and vice versa (Opt4). By setting the proper weights, we obtain more balanced designs (Opt1 and Opt2) that give simultaneous improvements in C_{PL} and \overline{Nu} .

Overall, we obtain less improvement in heat transfer and aerodynamics compared with the rib-free case. For example, for Opt2, we obtain a 9.7% decrease in C_{PL} and a 6.9% increase in \overline{Nu} , which are lower than the values for the rib-free case (11.7% and

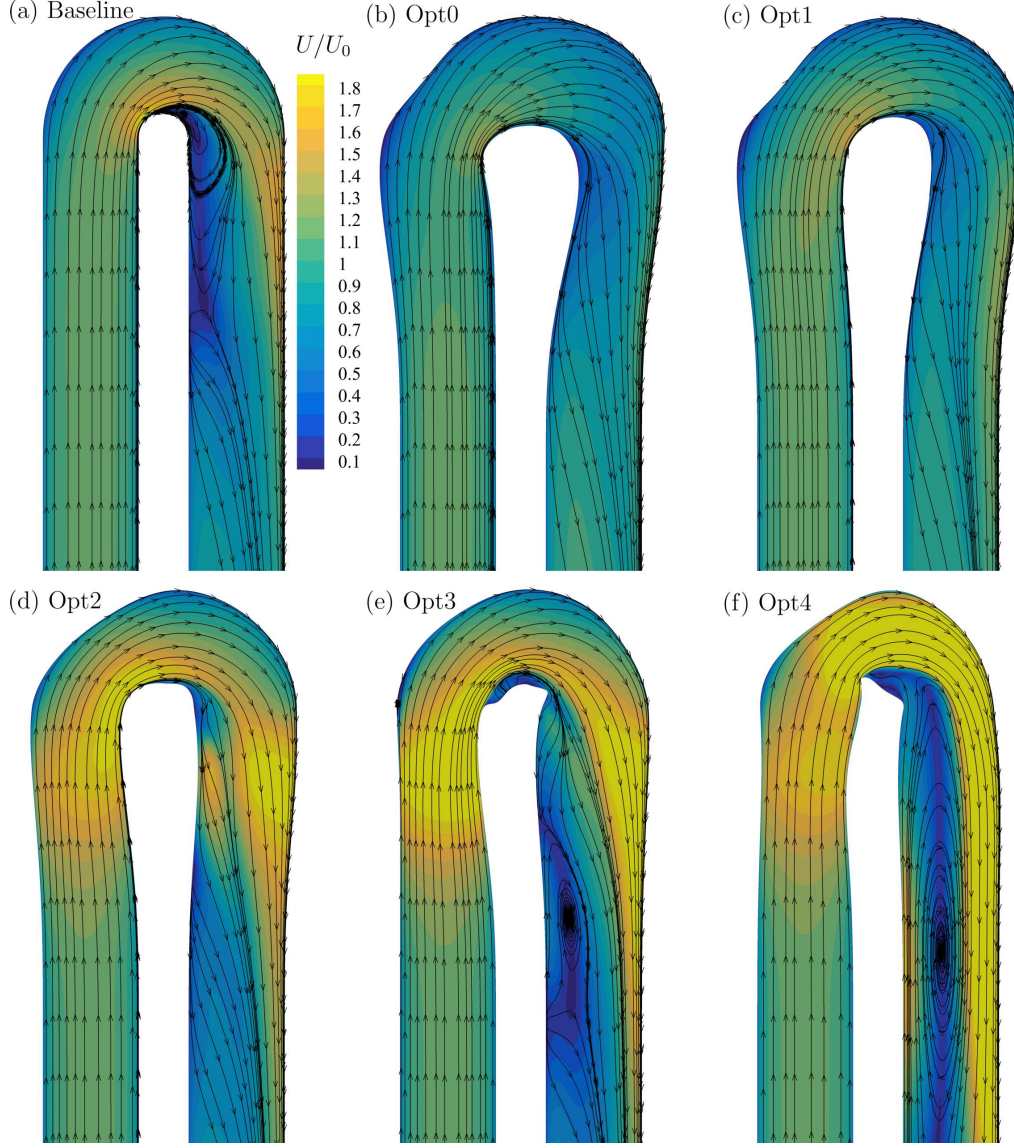


Figure 11: Velocity distribution and streamlines at the symmetry plane (rib free). For the minimum pressure loss case (Opt0), velocity decreases due to the expansion, whereas velocity increases for the maximum heat transfer case (Opt4), and a large separation region appears in the downstream section. For Opt2, the velocity increases but no large separation occurs.

13.0% for C_{PL} and \overline{Nu} , respectively; see Fig. 9). This indicates that, to improve the overall heat transfer and aerodynamic performance, changing the U-bend shape is more effective than changing only the ribs. As expected, adding ribs is more effective for improving the local heat transfer. For example, by comparing the baseline designs in Figs. 13 and 18, we see that adding ribs increases the local heat transfer, especially on the top wall in the upstream section.

We now compare the baseline and optimized rib shapes (Fig. 15). Overall, increasing the weight for \overline{Nu} results in higher ribs. This is expected, because higher ribs increase

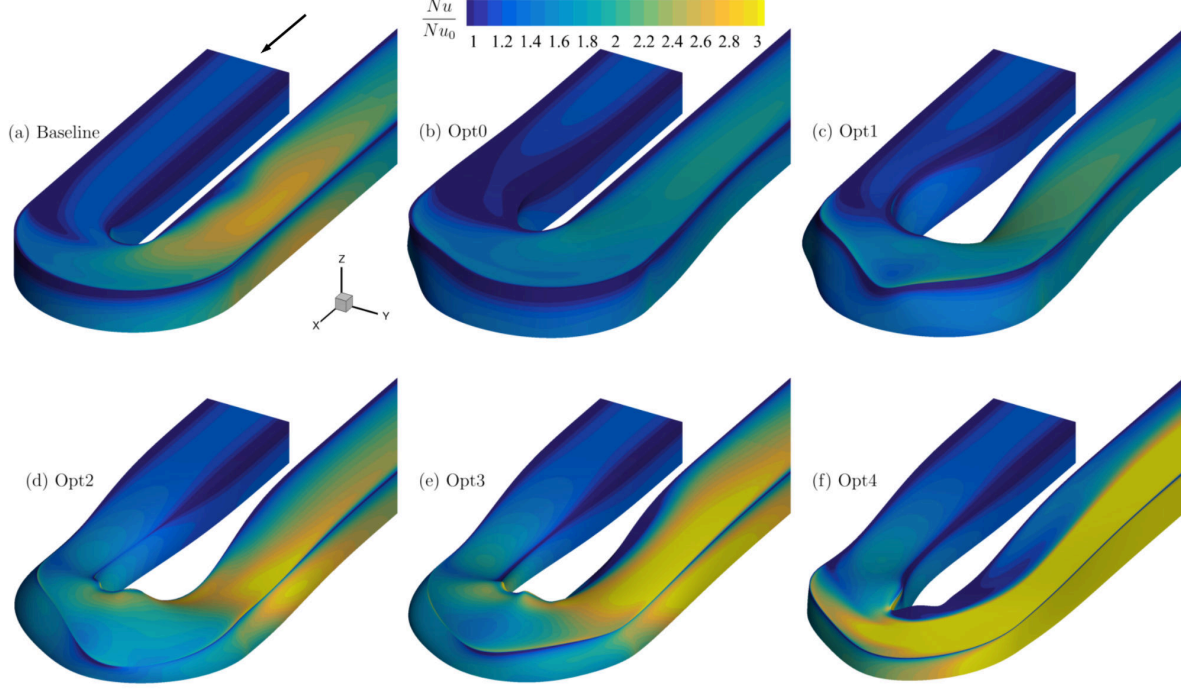


Figure 12: Comparison of \overline{Nu} between the baseline and rib-free optimized shapes. For the minimum pressure loss case (Opt0), \overline{Nu} decreases due to the reduced velocity in the channel, whereas for the maximum heat transfer case (Opt4), \overline{Nu} is large on the top and outer walls because of the increased velocity and large separation. The shrink-expansion-shrink feature in Opt2 effectively increases \overline{Nu} on all walls.

flow separation and generate stronger streamwise vortices, which enhance heat transfer. However, to balance the pressure loss, every rib adopts a completely different design that depends on the local flow conditions in the U-bend channel. In other words, the shape, height, pitch, and angle differ from rib to rib and do not follow any of the patterns reported in the literature. This result underscores the value of allowing each rib to change independently.

To analyze the three-dimensional vortex structures in the ribbed channel, we plot the isosurface of Q , which is a metric to identify the vortex core and is defined as

$$Q = \frac{1}{2}(\Omega_{i,j}\Omega_{i,j} - S_{i,j}S_{i,j}), \quad (19)$$

where S and Ω are the strain and rotation rates, respectively. Figure 16 compares Q isosurfaces between the baseline and optimized shapes. Figures 17 and 18 show the distribution of the corresponding local Nusselt number and the averaged Nusselt number along the streamwise direction, respectively.

The baseline design leads to rib-generated secondary flows and the corresponding strong vorticity near all the ribs [Fig. 16(a)]. For ribs 1 to 4, these secondary flows do not interact with each other because of a relatively large pitch-to-height ratio (13.4). In addition, a strong vortex core originates at the inner wall in the bend section and propa-

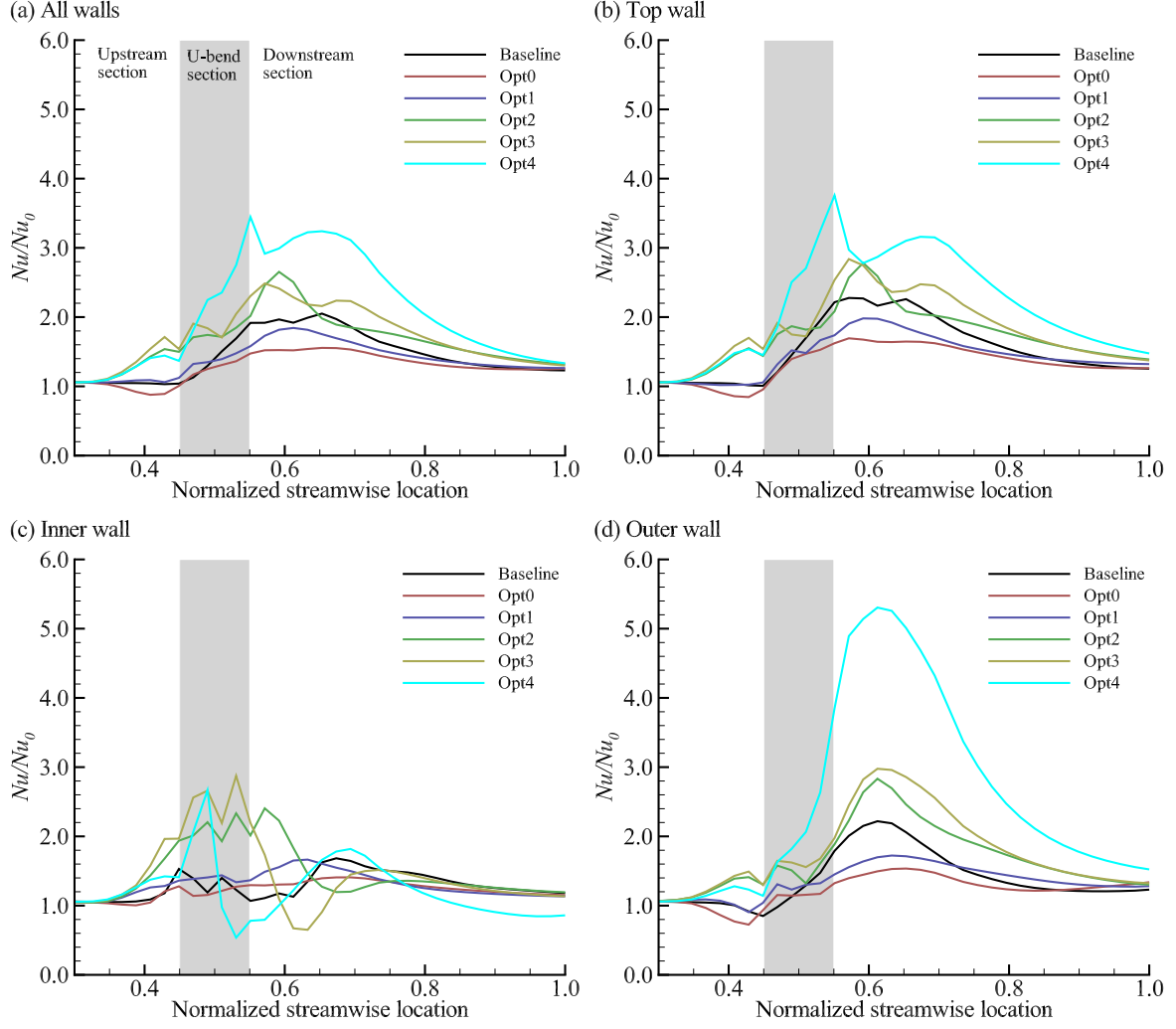


Figure 13: Averaged Nusselt number along the streamwise direction (rib free).

gates further downstream. This vortex core is attributed to the interaction between the inner wall separation bubble and the secondary flow from the ribs. In Opt0, the optimizer reduces the rib heights so that this interaction becomes weaker, and the size of the inner wall vortex core decreases slightly. Moreover, Opt0 uses a positive angle (rotated counterclockwise with respect to the streamwise direction) for ribs 7–9, which has the effect of further reducing the interaction between the rib-generated secondary flows and the inner wall vortex in the downstream section, as observed in the experimental study by Mochizuki et al. [6]. As a result, Nu decreases, especially on the top and outer walls (Figs. 17 and 18).

Increasing the weight for Nu (from Opt1 to Opt4) leads to a clear interaction of secondary flows between the ribs, which is primarily due to the increase in rib height. For example, starting from Opt2, a second strong vortex core originates from rib 3 and propagates downstream (Fig. 16). This second vortex also interacts with the inner

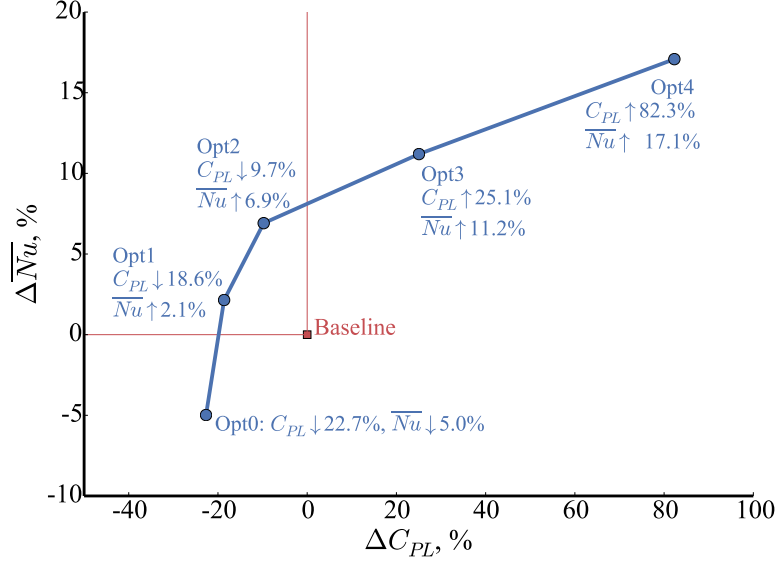


Figure 14: Pareto front for C_{PL} and \overline{Nu} (ribbed). For the minimum pressure loss case (Opt0), C_{PL} is reduced by 22.7%, whereas \overline{Nu} increases by 17.1% for the maximum heat transfer case (Opt4). Opt1 and Opt2 are more balanced designs that give simultaneous improvement in aerodynamics (18.6% and 9.7% reduction in C_{PL} , respectively) and heat transfer (2.1% and 6.9% increase in \overline{Nu} , respectively). The reference values correspond to the baseline design without optimization.

wall separation bubble and strengthens its intensity. In addition, the optimizer uses a negative angle (rotated clockwise with respect to the streamwise direction) for ribs 7–9 to further strengthen the rib-generated secondary flow. The stronger interaction between the inner wall separation bubble and the rib secondary flows leads to a larger inner wall vortex core. The enhanced vorticity increases the local Nu , especially on the top wall, as shown in Figs. 17 and 18.

To summarize, the ribs and U-bend channel generate complex vortex structures. The optimizations exploit this by adjusting the rib arrangements to control the vortex interaction and thereby balance the aerodynamics and heat transfer.

4 Conclusions

We discuss herein the use of the adjoint method for aerothermal optimizations for a ribbed U-bend channel. To allow a large design freedom, we use more than 100 design variables to parameterize the rib and channel geometries. We develop an improved preconditioning strategy that involves scaling the Jacobians and reducing the preconditioner stencil, which allows us to reliably converge the adjoint equations, even under conditions of large flow separation.

We find that optimizing only for heat transfer incurs a large penalty on pressure loss, and vice versa. For example, in the rib-free case, \overline{Nu} increases by 42.0% when optimizing only heat transfer, whereas C_{PL} increases by 770.9%. Similarly, for the

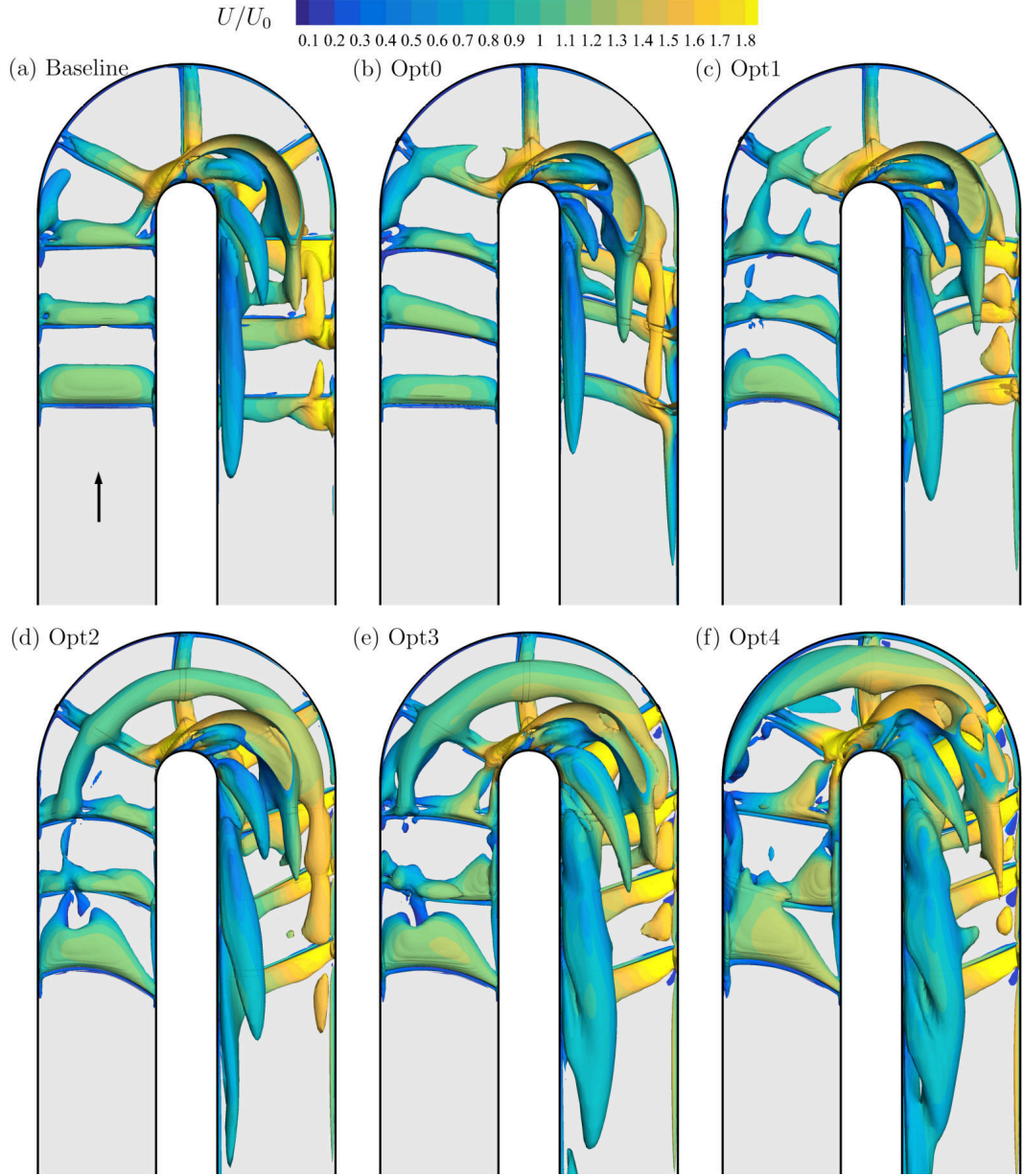


Figure 16: Comparison between baseline and optimized vortex structures. The isosurface is based on the Q criterion ($Q = 1 \times 10^4$) and the contour is based on the velocity magnitude. Overall, the velocity increases with increasing weight for Nu , which increases the heat transfer in the channel.

et al. [90]. Moreover, for the maximum heat transfer case, Nu increases by 42.0%, which is also higher than the previously reported bulk heat transfer enhancement of 16.9% [33]. The greater improvement in C_{PL} and \bar{Nu} in our study is attributed primarily to the large design freedom enabled by the proposed optimization framework.

To further demonstrate the proposed optimization framework, we add nine squared ribs on the top wall of the U-bend and allow all independent changes in rib arrangement.

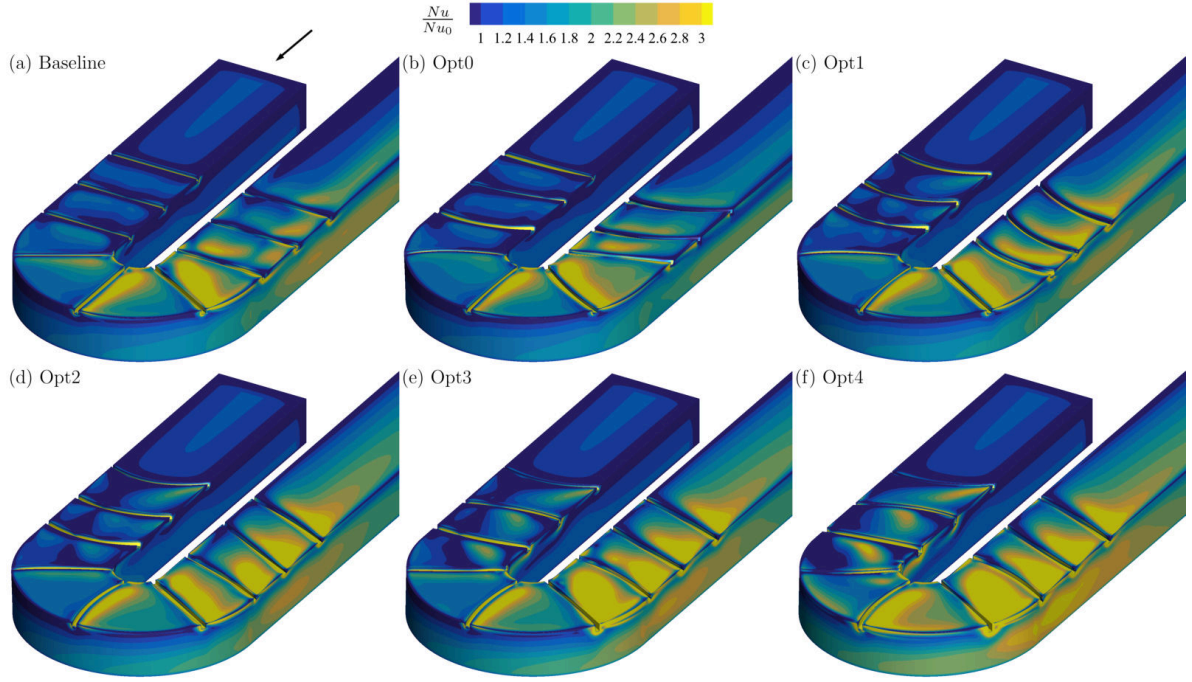


Figure 17: Comparison between baseline and optimized Nu distribution. For the minimum pressure loss case (Opt0), both the rib height and Nu decrease. For the maximum heat transfer case (Opt4), both the rib height and Nu increase, especially on the top walls.

We find that, depending on the local flow conditions, each rib adopts a different arrangement (shape, height, pitch, and angle) to balance heat transfer and aerodynamics. Generally, when adding more weight to Nu , the optimization increases the rib height to enhance the rib secondary flow. The rib secondary flow then interacts with the inner wall separation bubble, which enhances heat transfer, especially on the top wall. The results indicate that, to improve the overall heat transfer and aerodynamic performance, optimizing the U-bend shape is more effective than optimizing only the ribs. However, to improve the local heat transfer, adding ribs is more effective, especially in the upstream section.

We have developed and demonstrated powerful methods for designing internal cooling passages that can optimize designs parameterized using a large number of design variables to parameterize complex cooling configurations, thanks to gradient-based optimization and with coupled adjoint derivative computation. The proposed optimization framework has the potential to optimize general turbine heat transfer designs, such as the coupled design of channel shape and ribs and the design of film cooling and jet impingement cooling processes.

Acknowledgment

The computations were done in the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation Grant No.

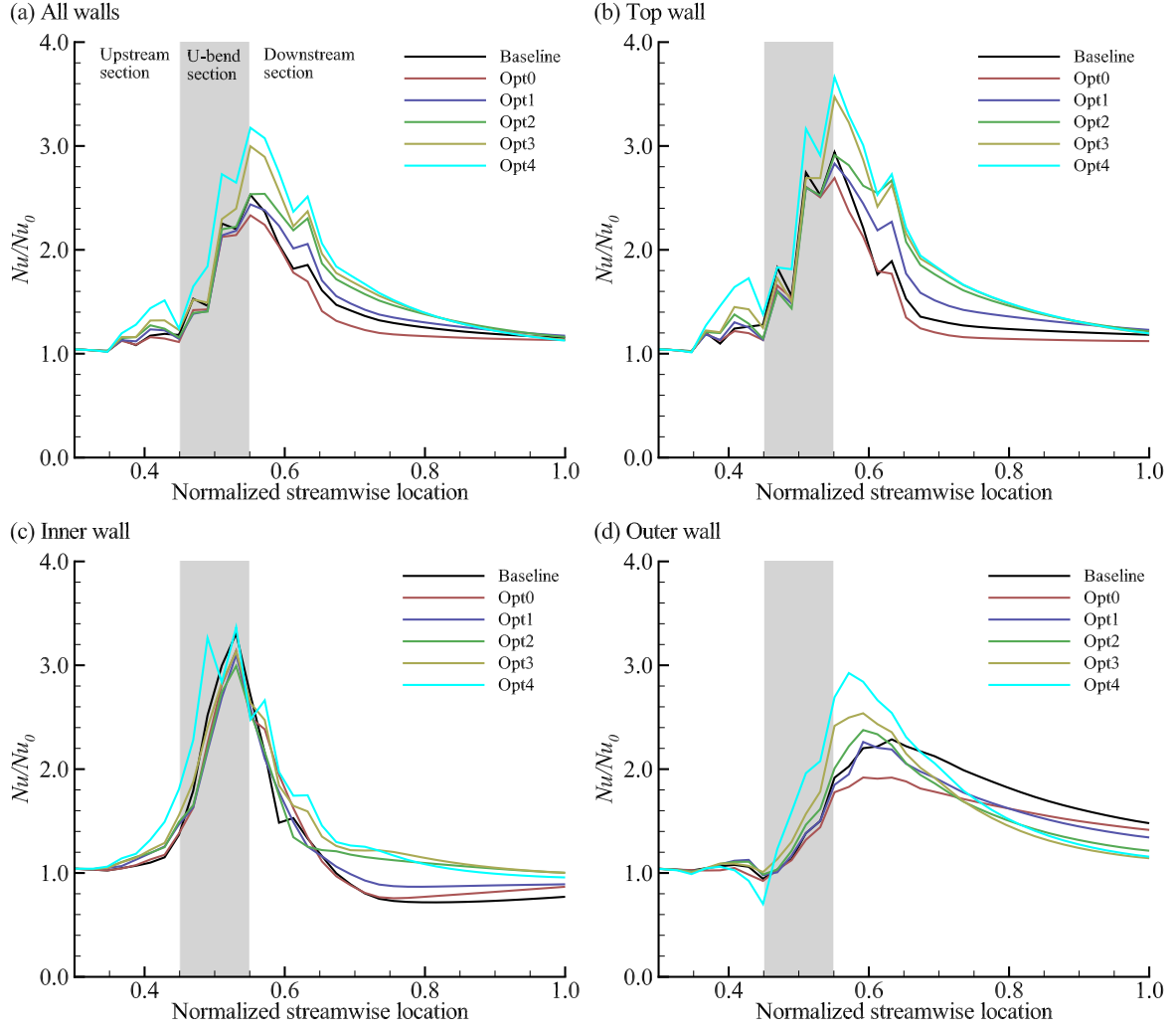


Figure 18: Averaged Nusselt number along the streamwise direction (ribbed).

ACI-1548562.

References

- [1] Han, J.-C., Dutta, S., and Ekkad, S., *Gas turbine heat transfer and cooling technology*, CRC Press, 2012.
- [2] Han, J., “Heat transfer and friction characteristics in rectangular channels with rib turbulators,” *Journal of Heat transfer*, Vol. 110, No. 2, 1988, pp. 321–328.
- [3] Han, J., and Park, J. S., “Developing heat transfer in rectangular channels with rib turbulators,” *International Journal of Heat and Mass Transfer*, Vol. 31, No. 1, 1988, pp. 183–195.

- [4] Han, J., Zhang, Y., and Lee, C., “Augmented heat transfer in square channels with parallel, crossed, and V-shaped angled ribs,” *Journal of Heat Transfer*, Vol. 113, No. 3, 1991, pp. 590–596.
- [5] Park, J., Han, J., Huang, Y., Ou, S., and Boyle, R., “Heat transfer performance comparisons of five different rectangular channels with parallel angled ribs,” *International Journal of Heat and Mass Transfer*, Vol. 35, No. 11, 1992, pp. 2891–2903.
- [6] Mochizuki, S., Murata, A., and Fukunaga, M., “Effects of rib arrangements on pressure drop and heat transfer in a rib-roughened channel with a sharp 180 deg turn,” *Journal of Turbomachinery*, Vol. 119, No. 3, 1997, pp. 610–616.
- [7] Ekkad, S. V., and Han, J.-C., “Detailed heat transfer distributions in two-pass square channels with rib turbulators,” *International Journal of Heat and Mass Transfer*, Vol. 40, No. 11, 1997, pp. 2525–2537.
- [8] Al-Hadhrami, L., and Han, J.-C., “Effect of rotation on heat transfer in two-pass square channels with five different orientations of 45 angled rib turbulators,” *International Journal of Heat and Mass Transfer*, Vol. 46, No. 4, 2003, pp. 653–669.
- [9] Gupta, A., SriHarsha, V., Prabhu, S., and Vedula, R., “Local heat transfer distribution in a square channel with 90 continuous, 90 saw tooth profiled and 60 broken ribs,” *Experimental Thermal and Fluid Science*, Vol. 32, No. 4, 2008, pp. 997–1010.
- [10] Erelli, R., Saha, A. K., and Panigrahi, P., “Influence of turn geometry on turbulent fluid flow and heat transfer in a stationary two-pass square duct,” *International Journal of Heat and Mass Transfer*, Vol. 89, 2015, pp. 667–684.
- [11] Hagari, T., Ishida, K., Takeishi, K.-i., Oda, Y., and Kitada, H., “Investigation on Heat Transfer Characteristics of a Cooling Channel with Dense Array of Angled Rib Turbulators,” *ASME Turbo Expo 2012: Turbine Technical Conference and Exposition*, American Society of Mechanical Engineers, 2012, pp. 387–399.
- [12] Lott, P. T., Lepot, I., Chérière, E., Thirifay, F., Semmler, K., and Bénét, L., “Internal cooling channels design investigations: Aerothermal optimisation of ribbed U-bends,” *ASME Turbo Expo 2013: Turbine Technical Conference and Exposition*, American Society of Mechanical Engineers, 2013, pp. V03AT12A046–V03AT12A046.
- [13] Xi, L., Gao, J., Xu, L., Zhao, Z., and Li, Y., “Study on heat transfer performance of steam-cooled ribbed channel using neural networks and genetic algorithms,” *International Journal of Heat and Mass Transfer*, Vol. 127, 2018, pp. 1110–1123.
- [14] Xie, G., Liu, X., and Yan, H., “Film cooling performance and flow characteristics of internal cooling channels with continuous/truncated ribs,” *International Journal of Heat and Mass Transfer*, Vol. 105, 2017, pp. 67–75.

- [15] Wang, L., Wang, S., Wen, F., Zhou, X., and Wang, Z., “Heat transfer and flow characteristics of U-shaped cooling channels with novel wavy ribs under stationary and rotating conditions,” *International Journal of Heat and Mass Transfer*, Vol. 126, 2018, pp. 312–333.
- [16] Ruck, S., and Arbeiter, F., “Detached eddy simulation of turbulent flow and heat transfer in cooling channels roughened by variously shaped ribs on one wall,” *International Journal of Heat and Mass Transfer*, Vol. 118, 2018, pp. 388–401.
- [17] Baek, S., Lee, S., Hwang, W., and Park, J. S., “Experimental and Numerical Investigation of the Flow in a Trailing Edge Ribbed Internal Cooling Passage,” *Journal of Turbomachinery*, Vol. 141, No. 1, 2019, p. 011012.
- [18] Prakash, C., and Zerkle, R., “Prediction of Turbulent Flow and Heat Transfer in a Ribbed Rectangular Duct With and Without Rotation,” *Journal of Turbomachinery*, Vol. 117, No. 2, 1995, pp. 255–264.
- [19] Stephens, M., Shih, T., and Civinskas, K., “Computation of flow and heat transfer in a rectangular channel with ribs,” *33rd Aerospace Sciences Meeting and Exhibit*, 1995, p. 180.
- [20] Rigby, D. L., Steinthorsson, E., and Ameri, A., “Numerical prediction of heat transfer in a channel with ribs and bleed,” *ASME 1997 International Gas Turbine and Aeroengine Congress and Exhibition*, American Society of Mechanical Engineers, 1997, pp. V003T09A078–V003T09A078.
- [21] Iacovides, H., “Computation of flow and heat transfer through rotating ribbed passages,” *International Journal of Heat and Fluid Flow*, Vol. 19, No. 5, 1998, pp. 393–400.
- [22] Bonhoff, B., Tomm, U., Johnson, B. V., and Jennions, I., “Heat transfer predictions for rotating U-shaped coolant channels with skewed ribs and with smooth walls,” *ASME 1997 International Gas Turbine and Aeroengine Congress and Exhibition*, American Society of Mechanical Engineers, 1997, pp. V003T09A027–V003T09A027.
- [23] Iacovides, H., and Raisee, M., “Recent progress in the computation of flow and heat transfer in internal cooling passages of turbine blades,” *International Journal of Heat and Fluid Flow*, Vol. 20, No. 3, 1999, pp. 320–328.
- [24] Jang, Y.-J., Chen, H.-C., and Han, J.-C., “Computation of flow and heat transfer in two-pass channels with 60 deg ribs,” *Journal of Heat Transfer*, Vol. 123, No. 3, 2001, pp. 563–575.
- [25] Wang, T.-S., and Chyu, M. K., “Heat convection in a 180-deg turning duct with different turn configurations,” *Journal of Thermophysics and Heat Transfer*, Vol. 8, No. 3, 1994, pp. 595–601.

- [26] Su, G., Chen, H.-C., Han, J.-C., and Heidmann, J. D., “Computation of flow and heat transfer in rotating two-pass rectangular channels (AR= 1:1, 1:2, and 1:4) with smooth walls by a Reynolds stress turbulence model,” *International Journal of Heat and Mass Transfer*, Vol. 47, No. 26, 2004, pp. 5665–5683.
- [27] Saha, K., and Acharya, S., “Effect of bend geometry on heat transfer and pressure drop in a two-pass coolant square channel for a turbine,” *Journal of Turbomachinery*, Vol. 135, No. 2, 2013, p. 021035.
- [28] Gao, T., Zhu, J., Li, J., Gong, J., and Xia, Q., “Improving heat transfer performance in two-pass ribbed channel by the optimized secondary flow via bend shape modification,” *International Communications in Heat and Mass Transfer*, Vol. 103, 2019, pp. 43–53.
- [29] Puente, R., Corral, R., and Parra, J., “Comparison between aerodynamic designs obtained by human driven and automatic procedures,” *Aerospace Science and Technology*, Vol. 72, 2018, pp. 443–454.
- [30] Kim, H.-M., and Kim, K.-Y., “Design optimization of rib-roughened channel to enhance turbulent heat transfer,” *International Journal of Heat and Mass Transfer*, Vol. 47, No. 23, 2004, pp. 5159–5168.
- [31] Kim, H.-M., and Kim, K.-Y., “Shape optimization of three-dimensional channel roughened by angled ribs with RANS analysis of turbulent heat transfer,” *International Journal of heat and mass transfer*, Vol. 49, No. 21-22, 2006, pp. 4013–4022.
- [32] Kim, K.-Y., and Lee, Y.-M., “Design optimization of internal cooling passage with V-shaped ribs,” *Numerical Heat Transfer, Part A: Applications*, Vol. 51, No. 11, 2007, pp. 1103–1118.
- [33] Verstraete, T., and Li, J., “Multi-objective optimization of a U-bend for minimal pressure loss and maximal heat transfer performance in internal cooling channels,” *ASME Turbo Expo 2013: Turbine Technical Conference and Exposition*, American Society of Mechanical Engineers, 2013.
- [34] Mazaheri, K., Zeinalpour, M., and Bokaei, H., “Turbine blade cooling passages optimization using reduced conjugate heat transfer methodology,” *Applied Thermal Engineering*, Vol. 103, 2016, pp. 1228–1236.
- [35] Yang, Y.-T., Tang, H.-W., and Wong, C.-J., “Numerical simulation and optimization of turbulent fluids in a three-dimensional angled ribbed channel,” *Numerical Heat Transfer, Part A: Applications*, Vol. 70, No. 5, 2016, pp. 532–545.
- [36] Seo, J.-W., Afzal, A., and Kim, K.-Y., “Efficient multi-objective optimization of a boot-shaped rib in a cooling channel,” *International Journal of Thermal Sciences*, Vol. 106, 2016, pp. 122–133.

- [37] Martins, J. R. R. A., and Hwang, J. T., “Multidisciplinary Design Optimization of Aircraft Configurations—Part 1: A modular coupled adjoint approach,” Lecture series, Von Karman Institute for Fluid Dynamics, Rode Saint Genèse, Belgium, May 2016. ISSN0377-8312.
- [38] Yu, Y., Lyu, Z., Xu, Z., and Martins, J. R. R. A., “On the Influence of Optimization Algorithm and Starting Design on Wing Aerodynamic Shape Optimization,” *Aerospace Science and Technology*, Vol. 75, 2018, pp. 183–199. doi:[10.1016/j.ast.2018.01.016](https://doi.org/10.1016/j.ast.2018.01.016).
- [39] Pironneau, O., “On Optimum Profiles in Stokes Flow,” *Journal of Fluid Mechanics*, Vol. 59, No. 01, 1973, pp. 117–128. doi:[10.1017/S002211207300145X](https://doi.org/10.1017/S002211207300145X).
- [40] Jameson, A., “Aerodynamic Design via Control Theory,” *Journal of Scientific Computing*, Vol. 3, No. 3, 1988, pp. 233–260.
- [41] Nielsen, E. J., and Anderson, W. K., “Aerodynamic Design Optimization on Unstructured Meshes Using the Navier–Stokes Equations,” *AIAA Journal*, Vol. 37, No. 11, 1999, pp. 1411–1419.
- [42] Mavriplis, D. J., “Discrete Adjoint-Based Approach for Optimization Problems on Three-Dimensional Unstructured Meshes,” *AIAA Journal*, Vol. 45, No. 4, 2007, pp. 741–750. doi:[10.2514/1.22743](https://doi.org/10.2514/1.22743).
- [43] Mader, C. A., and Martins, J. R. R. A., “Stability-Constrained Aerodynamic Shape Optimization of Flying Wings,” *Journal of Aircraft*, Vol. 50, No. 5, 2013, pp. 1431–1449. doi:[10.2514/1.C031956](https://doi.org/10.2514/1.C031956).
- [44] Lyu, Z., Kenway, G. K. W., and Martins, J. R. R. A., “Aerodynamic Shape Optimization Investigations of the Common Research Model Wing Benchmark,” *AIAA Journal*, Vol. 53, No. 4, 2015, pp. 968–985. doi:[10.2514/1.J053318](https://doi.org/10.2514/1.J053318).
- [45] Chen, S., Lyu, Z., Kenway, G. K. W., and Martins, J. R. R. A., “Aerodynamic Shape Optimization of the Common Research Model Wing-Body-Tail Configuration,” *Journal of Aircraft*, Vol. 53, No. 1, 2016, pp. 276–293. doi:[10.2514/1.C033328](https://doi.org/10.2514/1.C033328).
- [46] Kenway, G. K. W., and Martins, J. R. R. A., “Multipoint Aerodynamic Shape Optimization Investigations of the Common Research Model Wing,” *AIAA Journal*, Vol. 54, No. 1, 2016, pp. 113–128. doi:[10.2514/1.J054154](https://doi.org/10.2514/1.J054154).
- [47] Secco, N. R., and Martins, J. R. R. A., “RANS-based Aerodynamic Shape Optimization of a Strut-braced Wing with Overset Meshes,” *Journal of Aircraft*, Vol. 56, No. 1, 2019, pp. 217–227. doi:[10.2514/1.C034934](https://doi.org/10.2514/1.C034934).
- [48] Othmer, C., “Adjoint methods for car aerodynamics,” *Journal of Mathematics in Industry*, Vol. 4, No. 1, 2014, p. 6. doi:[10.1186/2190-5983-4-6](https://doi.org/10.1186/2190-5983-4-6).

- [49] He, P., Mader, C. A., Martins, J. R. R. A., and Maki, K. J., “An Aerodynamic Design Optimization Framework Using a Discrete Adjoint Approach with OpenFOAM,” *Computers & Fluids*, Vol. 168, 2018, pp. 285–303. doi:[10.1016/j.compfluid.2018.04.012](https://doi.org/10.1016/j.compfluid.2018.04.012).
- [50] Papoutsis-Kiachagias, E., Asouti, V., Giannakoglou, K., Gkagkas, K., Shimokawa, S., and Itakura, E., “Multi-point aerodynamic shape optimization of cars based on continuous adjoint,” *Structural and Multidisciplinary Optimization*, 2018, pp. 1–20.
- [51] Wang, D., and He, L., “Adjoint aerodynamic design optimization for blades in multistage turbomachines—Part I: Methodology and verification,” *Journal of Turbomachinery*, Vol. 132, No. 2, 2010, p. 021011.
- [52] Wang, D., He, L., Li, Y., and Wells, R., “Adjoint Aerodynamic Design Optimization for Blades in Multistage Turbomachines—Part II: Validation and Application,” *Journal of Turbomachinery*, Vol. 132, No. 2, 2010, p. 021012.
- [53] Xu, S., Radford, D., Meyer, M., and Müller, J.-D., “Stabilisation of discrete steady adjoint solvers,” *Journal of Computational Physics*, Vol. 299, 2015, pp. 175–195. doi:[10.1016/j.jcp.2015.06.036](https://doi.org/10.1016/j.jcp.2015.06.036).
- [54] Ma, C., Su, X., and Yuan, X., “An efficient unsteady adjoint optimization system for multistage turbomachinery,” *Journal of Turbomachinery*, Vol. 139, No. 1, 2017, p. 011003.
- [55] Dhert, T., Ashuri, T., and Martins, J. R. R. A., “Aerodynamic Shape Optimization of Wind Turbine Blades Using a Reynolds-Averaged Navier–Stokes Model and an Adjoint Method,” *Wind Energy*, Vol. 20, No. 5, 2017, pp. 909–926. doi:[10.1002/we.2070](https://doi.org/10.1002/we.2070).
- [56] Madsen, M. H. A., Zahle, F., Sørensen, N. N., and Martins, J. R. R. A., “Multi-point high-fidelity CFD-based aerodynamic shape optimization of a 10 MW wind turbine,” *Wind Energy Science*, Vol. 4, 2019, pp. 163–192. doi:[10.5194/wes-4-163-2019](https://doi.org/10.5194/wes-4-163-2019).
- [57] Kröger, J., Kühn, N., and Rung, T., “Adjoint volume-of-fluid approaches for the hydrodynamic optimisation of ships,” *Ship Technology Research*, Vol. 65, No. 1, 2018, pp. 47–68.
- [58] He, P., Filip, G., Martins, J. R. R. A., and Maki, K. J., “Hull form hydrodynamic design using a discrete adjoint optimization method,” *13th International Marine Design Conference*, Helsinki, Finland, 2018.
- [59] Garg, N., Kenway, G. K. W., Lyu, Z., Martins, J. R. R. A., and Young, Y. L., “High-fidelity Hydrodynamic Shape Optimization of a 3-D Hydrofoil,” *Journal of Ship Research*, Vol. 59, No. 4, 2015, pp. 209–226. doi:[10.5957/JOSR.59.4.150046](https://doi.org/10.5957/JOSR.59.4.150046).

- [60] Liao, Y., Garg, N., Martins, J. R. R. A., and Young, Y. L., “Viscous Fluid Structure Interaction Response of Composite Hydrofoils,” *Composite Structures*, Vol. 212, 2019, pp. 571–585. doi:[10.1016/j.compstruct.2019.01.043](https://doi.org/10.1016/j.compstruct.2019.01.043).
- [61] Martins, J. R. R. A., and Lambe, A. B., “Multidisciplinary Design Optimization: A Survey of Architectures,” *AIAA Journal*, Vol. 51, No. 9, 2013, pp. 2049–2075. doi:[10.2514/1.J051895](https://doi.org/10.2514/1.J051895).
- [62] Martins, J. R. R. A., and Hwang, J. T., “Review and Unification of Methods for Computing Derivatives of Multidisciplinary Computational Models,” *AIAA Journal*, Vol. 51, No. 11, 2013, pp. 2582–2599. doi:[10.2514/1.J052184](https://doi.org/10.2514/1.J052184).
- [63] Kenway, G. K. W., Kennedy, G. J., and Martins, J. R. R. A., “Scalable Parallel Approach for High-Fidelity Steady-State Aeroelastic Analysis and Derivative Computations,” *AIAA Journal*, Vol. 52, No. 5, 2014, pp. 935–951. doi:[10.2514/1.J052255](https://doi.org/10.2514/1.J052255).
- [64] Martins, J. R. R. A., Kenway, G. K. W., and Brooks, T. R., “Multidisciplinary Design Optimization of Aircraft Configurations—Part 2: High-fidelity aerostructural optimization,” Lecture series, Von Karman Institute for Fluid Dynamics, Rode Saint Genèse, Belgium, May 2016. ISSN0377-8312.
- [65] Brooks, T. R., Kenway, G. K. W., and Martins, J. R. R. A., “Benchmark Aerostructural Models for the Study of Transonic Aircraft Wings,” *AIAA Journal*, Vol. 56, No. 7, 2018, pp. 2840–2855. doi:[10.2514/1.J056603](https://doi.org/10.2514/1.J056603).
- [66] Gray, J. S., Mader, C. A., Kenway, G. K. W., and Martins, J. R. R. A., “Modeling Boundary Layer Ingestion Using a Coupled Aeropropulsive Analysis,” *Journal of Aircraft*, Vol. 55, No. 3, 2018, pp. 1191–1199. doi:[10.2514/1.C034601](https://doi.org/10.2514/1.C034601).
- [67] Gray, J. S., Kenway, G. K. W., Mader, C. A., and Martins, J. R. R. A., “Aero-propulsive Design Optimization of a Turboelectric Boundary Layer Ingestion Propulsion System,” *2018 AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Atlanta, GA, 2018. AIAA 2018-3976.
- [68] Garg, N., Kenway, G. K. W., Martins, J. R. R. A., and Young, Y. L., “High-fidelity Multipoint Hydrostructural Optimization of a 3-D Hydrofoil,” *Journal of Fluids and Structures*, Vol. 71, 2017, pp. 15–39. doi:[10.1016/j.jfluidstructs.2017.02.001](https://doi.org/10.1016/j.jfluidstructs.2017.02.001).
- [69] Zhang, P., Lu, J., Song, L., and Feng, Z., “Study on continuous adjoint optimization with turbulence models for aerodynamic performance and heat transfer in turbomachinery cascades,” *International Journal of Heat and Mass Transfer*, Vol. 104, 2017, pp. 1069–1082.
- [70] Gkaragkounis, K., Papoutsis-Kiachagias, E., and Giannakoglou, K., “The continuous adjoint method for shape optimization in Conjugate Heat Transfer problems with turbulent incompressible flows,” *Applied Thermal Engineering*, Vol. 140, 2018, pp. 351–362.

- [71] Verstraete, T., Müller, L., and Müller, J.-D., “Adjoint-Based Design Optimisation of an Internal Cooling Channel U-Bend for Minimised Pressure Losses,” *International Journal of Turbomachinery, Propulsion and Power*, Vol. 2, No. 2, 2017, p. 10.
- [72] Hayek, M. E., Wang, Q., and Laskowski, G. M., “Adjoint-based optimization of RANS eddy viscosity model for U-bend channel flow,” *2018 AIAA Aerospace Sciences Meeting*, 2018, p. 2091.
- [73] Albring, T., Sagebaum, M., and Gauger, N. R., *New Results in Numerical and Experimental Fluid Mechanics X: Contributions to the 19th STAB/DGLR Symposium Munich, Germany, 2014*, Springer International Publishing, Cham, 2016, Chaps. A Consistent and Robust Discrete Adjoint Solver for the SU2 Framework—Validation and Application, pp. 77–86. doi:[10.1007/978-3-319-27279-5_7](https://doi.org/10.1007/978-3-319-27279-5_7).
- [74] Xu, S., and Timme, S., “Robust and efficient adjoint solver for complex flow conditions,” *Computers & Fluids*, Vol. 148, 2017, pp. 26–38. doi:[10.1016/j.compfluid.2017.02.012](https://doi.org/10.1016/j.compfluid.2017.02.012).
- [75] He, P., Mader, C. A., Martins, J. R. R. A., and Maki, K. J., “Aerothermal optimization of internal cooling passages using a discrete adjoint method,” *AIAA/ASME Joint Thermophysics and Heat Transfer Conference*, Atlanta, GA, 2018.
- [76] Verstraete, T., “The VKI U-Bend Optimization Test Case,” Tech. rep., 2016. URL <http://aboutflow.sems.qmul.ac.uk/events/munich2016/benchmark/testcase1/>.
- [77] He, P., Mader, C. A., Martins, J. R. R. A., and Maki, K. J., “An object-oriented framework for rapid discrete adjoint development using OpenFOAM,” *AIAA Science and Technology Forum and Exposition*, 2019. doi:[10.2514/6.2019-1210](https://doi.org/10.2514/6.2019-1210).
- [78] Lambe, A. B., and Martins, J. R. R. A., “Extensions to the Design Structure Matrix for the Description of Multidisciplinary Design, Analysis, and Optimization Processes,” *Structural and Multidisciplinary Optimization*, Vol. 46, 2012, pp. 273–284. doi:[10.1007/s00158-012-0763-y](https://doi.org/10.1007/s00158-012-0763-y).
- [79] Othmer, C., “A continuous adjoint formulation for the computation of topological and surface sensitivities of ducted flows,” *International Journal for Numerical Methods in Fluids*, Vol. 58, No. 8, 2008, pp. 861–877. doi:[10.1002/fld.1770](https://doi.org/10.1002/fld.1770).
- [80] He, S., Jonsson, E., Mader, C. A., and Martins, J. R., “Aerodynamic Shape Optimization with Time Spectral Flutter Adjoint,” *2019 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, American Institute of Aeronautics and Astronautics, San Diego, CA, 2019. doi:[10.2514/6.2019-0697](https://doi.org/10.2514/6.2019-0697).
- [81] Kenway, G. K., Kennedy, G. J., and Martins, J. R. R. A., “A CAD-Free Approach to High-Fidelity Aerostructural Optimization,” *Proceedings of the 13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference*, Fort Worth, TX, 2010. doi:[10.2514/6.2010-9231](https://doi.org/10.2514/6.2010-9231).

- [82] Luke, E., Collins, E., and Blades, E., “A Fast Mesh Deformation Method Using Explicit Interpolation,” *Journal of Computational Physics*, Vol. 231, No. 2, 2012, pp. 586–601. doi:[10.1016/j.jcp.2011.09.021](https://doi.org/10.1016/j.jcp.2011.09.021).
- [83] Perez, R. E., Jansen, P. W., and Martins, J. R. R. A., “pyOpt: A Python-Based Object-Oriented Framework for Nonlinear Constrained Optimization,” *Structural and Multidisciplinary Optimization*, Vol. 45, No. 1, 2012, pp. 101–118. doi:[10.1007/s00158-011-0666-3](https://doi.org/10.1007/s00158-011-0666-3).
- [84] Gill, P. E., Murray, W., and Saunders, M. A., “SNOPT: An SQP algorithm for large-scale constrained optimization,” *SIAM Journal of Optimization*, Vol. 12, No. 4, 2002, pp. 979–1006. doi:[10.1137/S1052623499350013](https://doi.org/10.1137/S1052623499350013).
- [85] Spalart, P., and Allmaras, S., “A One-Equation Turbulence Model for Aerodynamic Flows,” *30th Aerospace Sciences Meeting and Exhibit*, 1992. doi:[10.2514/6.1992-439](https://doi.org/10.2514/6.1992-439).
- [86] Balay, S., Gropp, W. D., McInnes, L. C., and Smith, B. F., “Efficient Management of Parallelism in Object Oriented Numerical Software Libraries,” *Modern Software Tools in Scientific Computing*, edited by E. Arge, A. M. Bruaset, and H. P. Langtangen, Birkhäuser Press, 1997, pp. 163–202.
- [87] Balay, S., Buschelman, K., Gropp, W. D., Kaushik, D., Knepley, M. G., McInnes, L. C., Smith, B. F., and Zhang, H., “PETSc Web page,” , 2009. [Http://www.mcs.anl.gov/petsc](http://www.mcs.anl.gov/petsc).
- [88] Balay, S., Abhyankar, S., Adams, M. F., Brown, J., Brune, P., Buschelman, K., Dalcin, L., Dener, A., Eijkhout, V., Gropp, W. D., Kaushik, D., Knepley, M. G., May, D. A., McInnes, L. C., Mills, R. T., Munson, T., Rupp, K., Sanan, P., Smith, B. F., Zampini, S., Zhang, H., and Zhang, H., “PETSc Users Manual,” Tech. Rep. ANL-95/11 - Revision 3.10, Argonne National Laboratory, 2018. URL <http://www.mcs.anl.gov/petsc>.
- [89] Coletti, F., Verstraete, T., Bulle, J., Van der Wielen, T., Van den Berge, N., and Arts, T., “Optimization of a U-Bend for Minimal Pressure Loss in Internal Cooling Channels–Part II: Experimental Validation,” *Journal of Turbomachinery*, Vol. 135, No. 5, 2013, p. 051016.
- [90] Verstraete, T., Coletti, F., Bulle, J., Vanderwielen, T., and Arts, T., “Optimization of a U-Bend for Minimal Pressure Loss in Internal Cooling Channels–Part I: Numerical Method,” *Journal of Turbomachinery*, Vol. 135, No. 5, 2013, p. 051015.
- [91] He, P., Sun, Z., Guo, B., Chen, H., and Tan, C., “Aerothermal investigation of back-face clearance flow in deeply scalloped radial turbines,” *Journal of Turbomachinery*, Vol. 135, No. 2, 2013, p. 021002.
- [92] Kenway, G. K. W., Mader, C. A., He, P., and Martins, J. R. R. A., “Effective Adjoint Approaches for Computational Fluid Dynamics,” *Progress in Aerospace Sciences*, 2019. (Submitted).

- [93] Willeke, S., and Verstraete, T., “Adjoint optimization of an internal cooling channel U-bend,” *ASME Turbo Expo 2015: Turbine Technical Conference and Exposition*, American Society of Mechanical Engineers, 2015, pp. V05AT11A029–V05AT11A029.

A Dependence of simulation and optimization results on configuration of computational fluid dynamics

As mentioned in the introduction, flow separation is commonly present in turbine internal cooling channels. The complex flow conditions require that we evaluate how various CFD configurations affect the simulation and optimization results. In this appendix, we compare the simulation results obtained from three turbulence models and three mesh densities. We also run fine-mesh simulations of the optimized geometries to verify the optimization results.

We first compare the simulated velocity fields by using three turbulence models (Spalart–Allmaras, Launder–Sharma low Reynolds number $k-\varepsilon$, and $k-\omega$ SST), as shown in Figs. 19 and 20. We use the same mesh (409 600 cells) for these three simulations. In terms of averaged quantities, the Spalart–Allmaras model gives the best agreement, although all three turbulence models overestimate the total pressure loss. For flow details, the Spalart–Allmaras model most accurately captures the velocity magnitude and separation bubble.

Figures 21 and 22 compare the distribution of Nusselt number for the three turbulence models. The Launder–Sharma $k-\varepsilon$ model overestimates the average Nusselt number, whereas the Spalart–Allmaras and $k-\omega$ SST models underestimate it. All three turbulence models fail to quantitatively capture the distribution of the Nusselt number, although the Spalart–Allmaras model gives the best agreement. Overall, we conclude that the Spalart–Allmaras model is the best option for this case.

Table 4 shows how mesh density affects the simulated pressure loss and Nusselt number when using the Spalart–Allmaras turbulence model. Refining the meshes from medium to fine results in a 4.5% increase in C_{PL} for the rib-free case, whereas the variation is less than 2.0% for all other cases. This amount of variation is expected, especially given the large separation in the flow. As mentioned in Sec. 3.1, we use the medium mesh for optimizations to save computation time.

To further confirm the validity of the improved results obtained from the optimizations with the medium-mesh, we ran fine-mesh simulations for the optimized geometries and compare the corresponding Pareto fronts in Fig. 23. For both rib-free and ribbed cases, the Pareto front obtained when using a medium mesh is similar to that obtained when using using a fine mesh. This result indicates that the derivatives computed by using the medium mesh follow the same trend as those computed by using the fine mesh. Therefore, the medium mesh is adequate for design optimization studies.

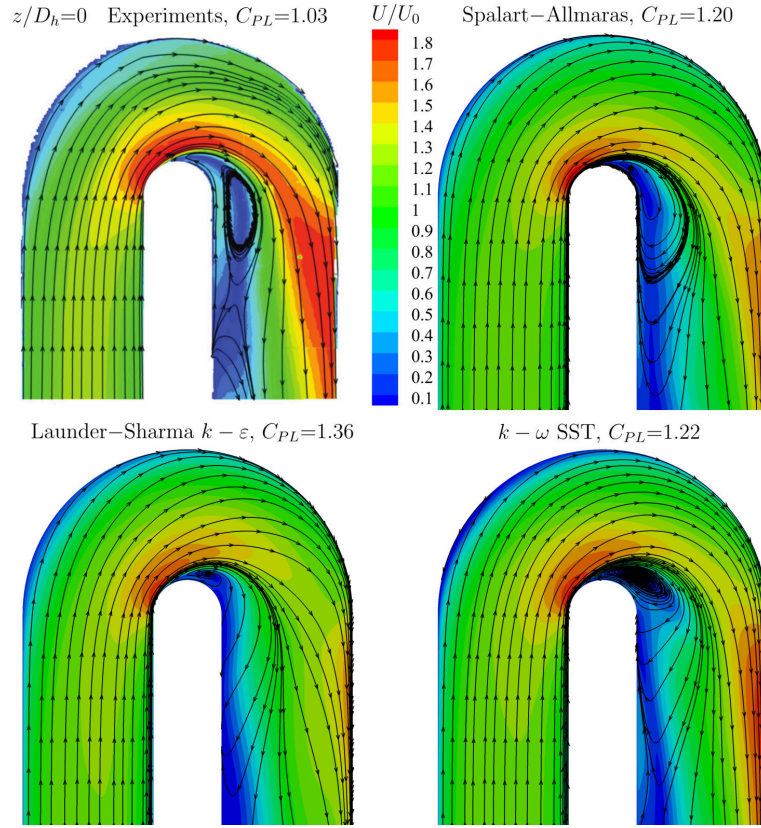


Figure 19: The velocity fields ($z/D_h = 0$) obtained by using the Spalart-Allmaras turbulence model are the closest to the experimental data acquired by Coletti et al. [89].

Table 4: Impact of mesh size on pressure loss and Nusselt number when using the Spalart-Allmaras turbulence model.

Case	Mesh size	Mesh cells	C_{PL}	\overline{Nu}/Nu_0
Rib-free	Coarse	51 200	1.196	1.489
	Medium	409 600	1.202	1.432
	Fine	3 276 800	1.257	1.404
Ribbed	Coarse	167 400	2.018	1.468
	Medium	1 339 200	1.984	1.422
	Fine	10 713 600	2.020	1.409

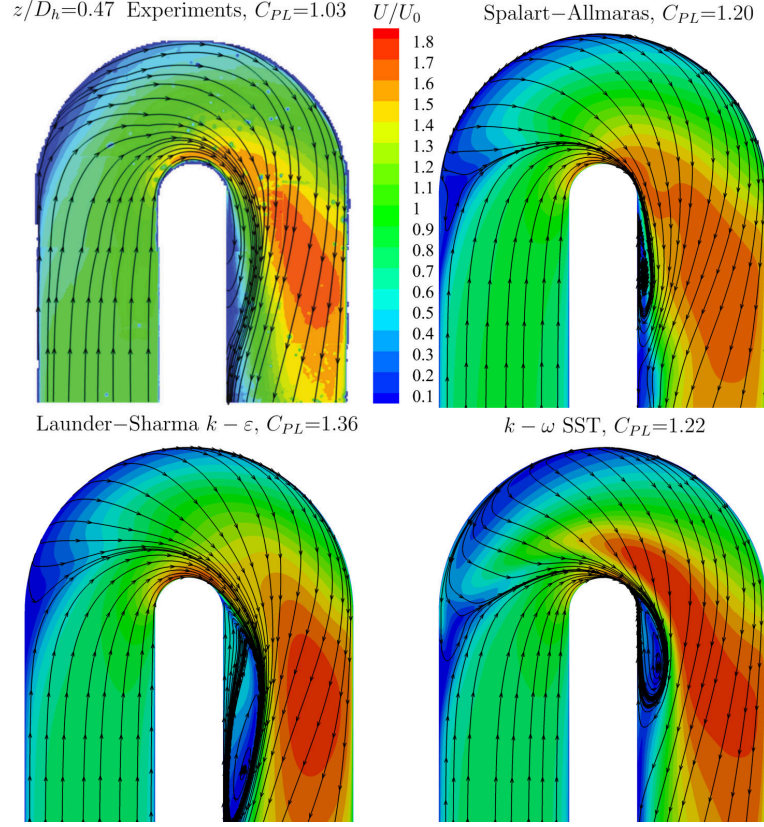


Figure 20: The velocity fields ($z/D_h = 0.47$) obtained by using the Spalart-Allmaras turbulence model are the closest to the experimental data acquired by Coletti et al. [89].

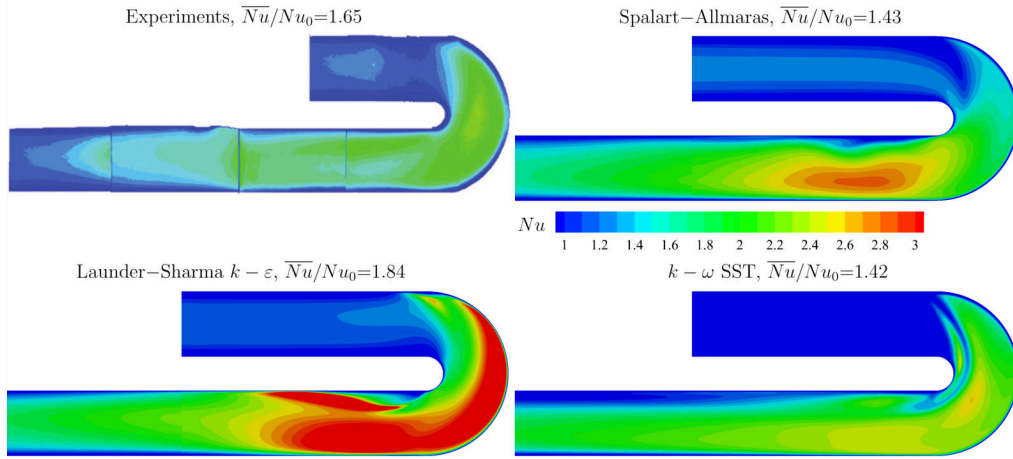


Figure 21: The Nusselt numbers obtained by using the Spalart-Allmaras turbulence model are the closest to the experimental data acquired by Coletti et al. [89].

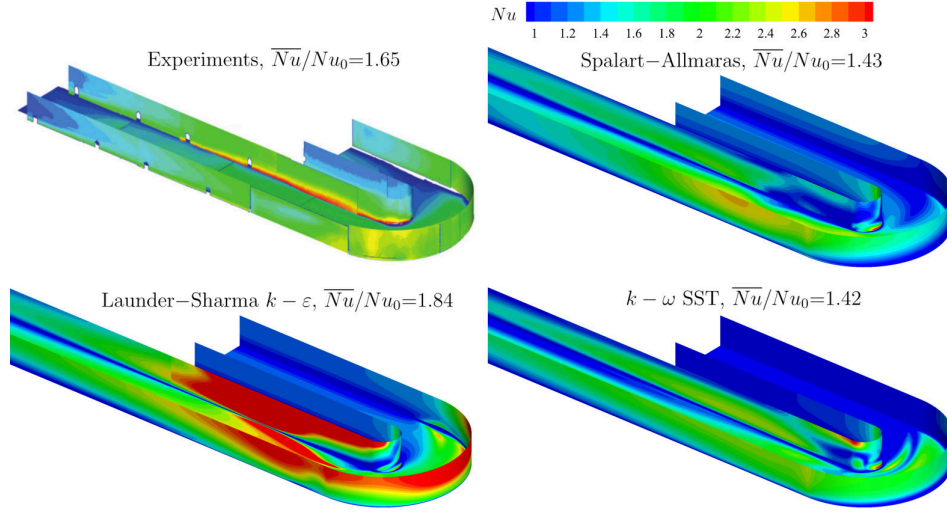


Figure 22: The Nusselt numbers obtained by using the Spalart–Allmaras turbulence model are the closest to the experimental data acquired by Coletti et al. [89].

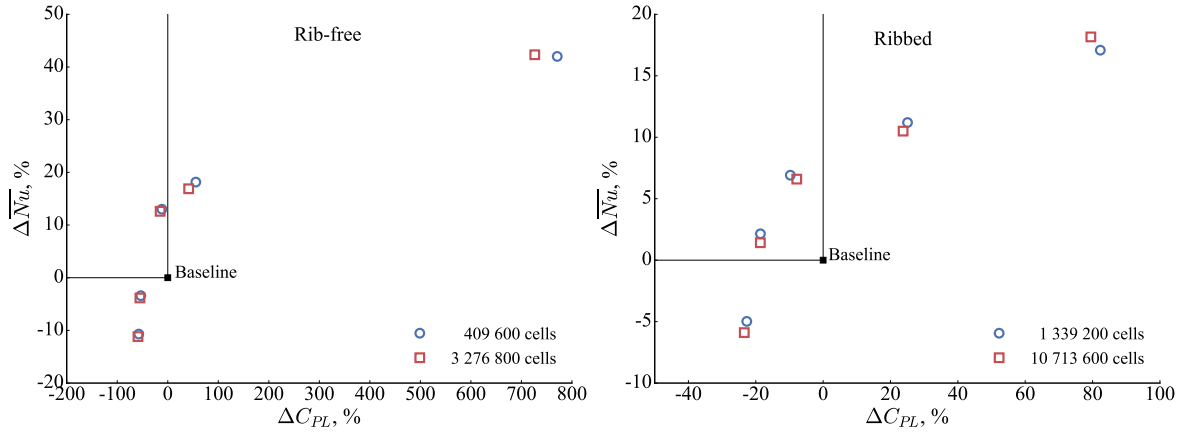


Figure 23: When verifying the performance of optimized geometries obtained by using fine meshes, we obtain Pareto fronts similar to those obtained when using medium meshes. Therefore, medium meshes are adequate for optimizations.