AIAA JOURNAL
Vol. 46, No. 10, October 2008

Control of a Simplified Unsteady Film-Cooling Flow Using Gradient-Based Optimization

Frank Muldoon*
Louisiana State University, Baton Rouge, Louisiana 70803

DOI: 10.2514/1.34120

Maximization of film-cooling effectiveness in an idealized film-cooling flow is accomplished using a gradient-based control strategy. The state equations are the two-dimensional unsteady incompressible Navier–Stokes equations and the temperature equation. The control is a spatially and temporally varying normal-velocity boundary condition subject to a constraint of zero mass and energy flux. The control that maximizes the objective function is found using a conjugate gradient method, in which the gradient of the objective function with respect to the control variables is obtained from solving a set of adjoint equations. The effectiveness of choices for the objective function are examined, along with the effect of the event horizon over which the control is obtained by the optimizer. Drastic improvements in the film-cooling effectiveness are obtained for certain choices of the objective function and event horizon. Although not a specific target of the optimizer, one result of the control is that the wall jet remains attached to the wall. The numerical issues involved with finding the control are discussed and the features of the resulting control are analyzed, with the goal of understanding the mechanisms that affect the film-cooling effectiveness.

Nomenclature

H	=	generic objective function
k	=	turbulent kinetic energy, $\frac{1}{2}(\overline{u'u' + v'v'})$
P_r	=	Prandtl number
p	=	pressure
Re	=	Reynolds number
$\mathbf{s}$	=	vector of all discretized state variables (velocity, pressure, and temperature)
T	=	temperature
t	=	time
t_{EH}	=	event-horizon length (the extent of the temporal dimension of the control-variable space)
U_0	=	maximum inlet crossflow velocity
$\mathbf{u}$	=	velocity vector
u	=	first component of the velocity vector, u_1
v	=	second component of the velocity vector, u_2
x	=	spatial coordinate in the downstream direction
y	=	spatial coordinate in the vertical direction
α	=	distance along line in the search direction
η	=	film-cooling effectiveness
$\boldsymbol{\lambda}$	=	adjoint variables
ϕ	=	control variables

I. Introduction

THE efficiency of turbine engines improves as the operating temperature of the working fluid in the turbine increases. However, higher fluid temperatures in turbines leads to turbine blade failure as a result of exceeding temperature limits for the turbine blade material. A common means of cooling turbine blades is film cooling, in which the goal is to protect the blade surface from the hot crossflow by a film of cooler fluid injected through holes in the blade surface. These film-cooling holes must be designed in such a way that the coolant jet covers and remains near the blade surface and does not excessively penetrate into the crossflow. The film-cooling jets consume a significant amount of process air, and therefore an important design goal is to maximize the cooling and minimize the mass flow through the jets.

Computational [1–3] and experimental studies [4–7] have shown that pulsing can greatly effect the structure and spreading of jets. In work specific to film cooling, recent experimental works [8–10] have investigated the effect of pulsing on film-cooling effectiveness. The results of these investigations have been mixed. No increase in film-cooling effectiveness from jet pulsation was found in [8,9]. However, in [10], increases in film-cooling effectiveness were observed with pulsing. These differences may be due to factors such as different jet inclination angles, Reynolds numbers, pulsing details, and the geometry of the film-cooling hole. Computational studies [11,12] have shown that pulsing has a strong effect on the behavior of the film-cooling jet and therefore a strong effect on the film-cooling effectiveness. Improvements in film-cooling effectiveness were found for some combinations of pulsing and duty cycle. It should be noted that the increases in film-cooling effectiveness in [10,11] were only found using duty cycles less than 1 (with a duty cycle of 1 meaning no pulsing). Because the duty cycle is equivalent to the time-averaged injection ratio, this implies that the proper baseline should be an unpulsed jet with the same time-averaged mass flow. A computational work [11] found no increase in film-cooling effectiveness over the unpulsed jet when comparing pulsed and unpulsed jets with the same time-averaged mass flow. However, for various reasons, the injection ratio of the unpulsed film-cooling jets cannot be lowered beyond a certain level. If pulsing with a duty cycle less than 1 increases the film-cooling effectiveness, then this is a win–win situation, because less process air is required for better film cooling.

The pulsing schemes examined in [11] were chosen primarily based on intuition; however, it is desired to find the pulsing that is optimum, and therefore some type of mathematical optimization procedure should be used. Because jet pulsation is an unsteady phenomena, this problem cannot be modeled using time-invariant models such as the Reynolds-averaged Navier–Stokes equations, but instead requires solution of the unsteady Navier–Stokes equations. This is unfortunate, because solutions to the unsteady Navier–Stokes equations are many orders of magnitude more computationally expensive than the Reynolds-averaged Navier–Stokes equations. For this reason, the coupling of optimization with the unsteady Navier–Stokes and temperature equations with the goal of determining the optimal pulsing for a film-cooling flow has, to date, not been accomplished. The aim of the present work is to study the numerical issues involved with determining the optimal pulsing (or control) and the features of the optimal pulsing for a simplified film-cooling flow, modeled by the unsteady Navier–Stokes and temperature equations.

Received 18 August 2007; revision received 26 January 2008; accepted for publication 8 February 2008.

*Center for Computation and Technology.

The primary purpose behind the mathematical modeling of fluid and heat transfer problems using computational fluid dynamics (CFD) is to aid in the design or control of physical systems. This implies optimization, because an *optimal* design or control is always desired (subject to certain constraints, of course). Although it still requires substantial computational resources, CFD has reached a stage in which solutions to many fluid and heat transfer problems may be reliably and quickly obtained. This has enabled CFD to replace expensive physical prototype models in the design process and has greatly reduced design cost and time. However, this use of CFD does not change the fundamentals of design processes, which are still guided primarily by intuition. A CFD model of a system provides much information that cannot be obtained from a physical model; proper use of this information can greatly aid the design process. For full advantage to be taken of CFD, it must be incorporated into the design process in a more fundamental manner than simply as a replacement for physical models. This may be done by combining CFD and optimization with the goal of optimizing or controlling some aspect of a physical system. The goal of the present work is the investigation of control strategies for improving film cooling of jet turbine blades. The present work concerns the incorporation of a CFD method into an optimization procedure and its application to a film-cooling flow. More specifically, the control entails varying (subject to certain constraints) the velocity boundary conditions in space and time with the goal of maximizing the film-cooling effectiveness in a film-cooling problem. The details of the resulting control are potentially very fine-scale, limited only by the spatial and temporal resolution of the discrete system model. Recent advances [13,14] in microelectromechanical systems have increased the practicality of performing fine-scale control of fluidic systems, with a goal of controlling heat transfer. A practical implementation of a controller using microelectromechanical systems will require detailed knowledge of the effect of the controller on the flow, along with the determination of effective control strategies.

Optimization methods can be divided into two general categories: those that use stochastic methods (examples are simulated annealing and genetic algorithms) and those that use the gradient of the objective function. If the gradient of the objective function cannot be obtained (for example, if the objective function is not continuous), then gradient-based methods cannot be used, although stochastic methods (which do not require the gradient) can be used. Stochastic methods generally require many more evaluations of the objective function than gradient-based methods. Because the objective function in the present work requires the solution of the unsteady Navier–Stokes and temperature equations, evaluation of the objective function is extremely computationally expensive. This gives a significant computational advantage to gradient-based methods. Primarily for this reason, the present work uses a gradient-based method (more specifically, a conjugate gradient algorithm) to find the minimum of the objective function. The gradient of the objective function is found by solving the adjoint equations, which are derived from the discretized Navier–Stokes and temperature equations. Solution of these equations requires approximately the same computational cost as solving the state equations, which, in the present work, are the two-dimensional unsteady incompressible Navier–Stokes and temperature equations.

A gradient-based optimizer, with the gradient obtained by solving the adjoint equations, has been applied to aerodynamic shape problems, beginning with [15]. In these and further works, the state equations were time-invariant and the system model was either the potential equations [16], steady Euler equations [17–19], or the Reynolds-averaged Navier–Stokes equations [17,19–22]. The solution of shape optimization problems using these equations is now commonplace in the aerospace industry and other industries. Because of the large computational expense occasioned by the temporal dimension, the unsteady Navier–Stokes equations have rarely been used for the system model; however, recent advances in computing power have made it possible to consider using these equations as the system model. Studies that used the two-dimensional unsteady incompressible Navier–Stokes equations for the state equations include [23–26]. In [23–25], the control was a source term added to the momentum equations throughout the computational domain. Although very good results can be obtained with this control, it is essentially impossible to physically implement such a controller. Control of vortex shedding behind a circular cylinder was accomplished in [26] using the more physically realizable control of injection and suction at the surface of the cylinder. The two-dimensional unsteady compressible Navier–Stokes equations were used for the state equations in [27], in which the choice of control space was investigated, and in [28], which focused on noise reduction in a mixing layer. The control in [28] was source terms added to the equations in the interior of the computational domain, although within a small subset of the computational domain. The issue of the proper solidification of metals in castings is of great industrial importance and has received attention in [29,30], in which control of natural convection and solidification was achieved using an adjoint-based conjugate gradient approach. In this work, the system model was the two-dimensional unsteady incompressible Navier–Stokes equations and the Boussinesq approximation. The two-dimensional unsteady incompressible Navier–Stokes and temperature equations were used as the system model in [31] concerning control of the temperature. Because the control in [31] was the temperature and the system model contained no influence of the temperature on the velocity and pressure, the velocity was solved independently of the temperature and only appeared as coefficients in the equation governing the temperature. Control of the temperature in a magnetoconvection problem was addressed in [32], in which the system model was the unsteady incompressible Navier–Stokes equations, along with the Boussinesq approximation and the Lorentz force arising from the magnetic field. Although most of the numerical problems solved in [32] were two-dimensional, one small $(10 \times 10 \times 10)$ three-dimensional problem was solved.

Because of the large computational expense involved, the three-dimensional unsteady incompressible Navier–Stokes equations have rarely been used as the state equations. Works in which the three-dimensional unsteady incompressible Navier–Stokes equations have been used as the state equations include [33–35]. The work by [33] concerned drag reduction of turbulent channel flow by means of varying the wall normal velocity. With proper choice of the objective function, significant reductions in drag were achieved, even to the extent of laminarizing the flow. Control of transition in the boundary layer and channel flows was studied in [35]. Recently, work [36,37] has also been done in the field of three-dimensional computer-based animations in which the goal is to generate physically realistic fluid flows that behave in a specific way (i.e., are controlled in a specific way). Although physical accuracy is not a prime concern in such animations, the desire for physically realistic flows causes the methods used to be very similar to those used in the physical and mathematical sciences.

II. Optimization Problem

In the present work, square brackets [ ] denote functional dependence, and parentheses ( ) denote multiplication; that is, $a[b + c]$ means that a is a function of $b + c$, and $a(b + c)$ means a times $b + c$; boldface type indicates vectors. Figure 1 is a schematic of the optimization problem with which the present work is concerned. The goal is to maximize the film-cooling effectiveness along the bottom wall of Fig. 1 by varying the normal component of the velocity at the bottom wall, subject to the constraint that the mass and energy flux through the bottom wall is zero. Mathematically, this problem is defined in a general form as follows.

Maximize an objective function $H[T[\phi[x, t]]]$ with respect to a control-variable function $\phi[x, t]$ defined on a boundary $\partial\Omega_\phi$, subject to the constraints of Eqs. (1–7).

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p - \frac{1}{Re} \nabla^2 \mathbf{u} = 0 \quad \text{in } \Omega \times (t_1, t_2) \tag{1}$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega \times (t_1, t_2) \tag{2}$$

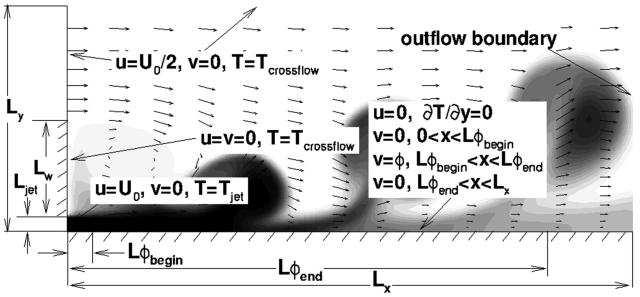

Fig. 1 Schematic of the physical domain for the minimization problem showing boundary conditions and dimensions.

$$\frac{\partial T}{\partial t} + \mathbf{u}\cdot\nabla T - \frac{1}{RePr}\nabla^2 T = 0 \quad \text{in } \Omega \times (t_1, t_2) \tag{3}$$

$$\Gamma[\mathbf{u}, \phi] = 0 \quad \text{on } \partial\Omega \times (t_1, t_2) \tag{4}$$

$$\Psi[T] = 0 \quad \text{on } \partial\Omega \times (t_1, t_2) \tag{5}$$

$$\Theta[\phi] = 0 \quad \text{on } \partial\Omega_\phi \times (t_1, t_2) \tag{6}$$

$$\Lambda[\phi, T] = 0 \quad \text{on } \partial\Omega_\phi \times (t_1, t_2) \tag{7}$$

The film-cooling effectiveness is defined by

$$\eta[x,t] = \frac{T_{\text{crossflow}} - T[x,0,t]}{T_{\text{crossflow}} - T_{\text{jet}}} \tag{8}$$

With the values of $T_{\text{crossflow}}$ and T_{jet} used in the present work (see Sec. V.A), this reduces to Eq. (9).

$$\eta[x,t] = 1 - T[x,0,t] \tag{9}$$

Note that the film-cooling effectiveness is bounded between zero and one and that maximizing the film-cooling effectiveness is equivalent to minimizing the wall temperature. This leads to the objective function defined by Eq. (10), in which $H[T[\phi[x,t]]]$ is the spatially and temporally averaged bottom-wall temperature squared.

$$H[T[\phi[x,t]]] = \frac{1}{(t_2 - t_1)}\frac{1}{L_x}\int_{t_1}^{t_2}\int_0^{L_x} (T[x,0,t])^2 \,\mathrm{d}x\,\mathrm{d}t \tag{10}$$

The interior of the spatial computational domain is Ω, $\partial\Omega$ is its boundary, $\partial\Omega_\phi$ is the part of $\partial\Omega$ in which control is applied, L_x is the length of the bottom wall, and t_1, t_2 are the beginning and the end of the time period over which the optimization problem is defined. The temporal length $t_2 - t_1$ should be long enough that the temporal statistics of the temperature are independent of $t_2 - t_1$. Equations (1–3), respectively, describe the conservation of momentum, mass, and energy. Constant physical properties are assumed throughout the present work; the physical properties are defined by the Reynolds number Re and the Prandtl number Pr. Equations (4–7) are, respectively, the velocity boundary conditions, the temperature boundary conditions, the zero-mass-flux constraint on the control variables, and the zero-energy-flux constraint on the control variables. Obtaining solutions to Eqs. (1–7) requires a numerical method that discretizes these equations and thereby transforms the continuous optimization problem into a discrete one.

III. Discretization of Navier–Stokes and Temperature Equations

A. Spatial Discretization

The state equations in the present work are the incompressible two-dimensional nonconservative unsteady Navier–Stokes and temperature equations (1–3). In the present work, the velocity vector is denoted by $\mathbf{u}$, the individual components of which are defined by $\mathbf{u} = \begin{pmatrix} u \\ v \end{pmatrix}$. Using finite differences, Eqs. (1–3) are discretized on a staggered grid, which is used to avoid the appearance of spurious modes in the pressure field. In two dimensions, each pressure grid point is surrounded by four velocity grid points. Pressure is not needed or defined on nonperiodic boundaries. For the convective terms, a three-point second-order centered scheme is used [3]. Because of the staggered grid, none of the velocity components are stored at the same location. They are, however, needed at the same location to compute the terms of $u_j(\partial u_i/\partial x_j)$, where $i \neq j$. They are obtained by fitting a two-dimensional Lagrange surface over the point in question. The fitting uses products of two-point Lagrange polynomials, resulting in a total of four points used to fit the surface, and is second-order-accurate. The number of points, and hence the accuracy, is retained as the boundary of the flow domain is approached. The diffusive terms are represented by three-point second-order-accurate central-difference schemes. The pressure-gradient term and the terms in the continuity equation are represented by two-point second-order-accurate centered stencils. Equation (3) is represented using the same order and type of scheme used for the momentum equations. Further details of the finite difference schemes can be found in [3].

B. Temporal Discretization

The spatially discretized convective and diffusive terms of the momentum and temperature equations are integrated in time using the explicit third-order Adams–Bashforth scheme [Eqs. (11) and (12)]. The time step for which the solution is sought is denoted by $n+1$.

$$\begin{aligned}\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} = {} & -\nabla p^{n+1} + \frac{23}{12}\left(\frac{1}{Re}\nabla^2\mathbf{u}^n - (\mathbf{u}\cdot\nabla\mathbf{u})^n\right) \\ & - \frac{16}{12}\left(\frac{1}{Re}\nabla^2\mathbf{u}^{n-1} - (\mathbf{u}\cdot\nabla\mathbf{u})^{n-1}\right) \\ & + \frac{5}{12}\left(\frac{1}{Re}\nabla^2\mathbf{u}^{n-2} - (\mathbf{u}\cdot\nabla\mathbf{u})^{n-2}\right)\end{aligned} \tag{11}$$

$$\begin{aligned}\frac{T^{n+1} - T^n}{\Delta t} = {} & \frac{23}{12}\left(\frac{1}{RePr}\nabla^2 T^n - (\mathbf{u}\cdot\nabla T)^n\right) \\ & - \frac{16}{12}\left(\frac{1}{RePr}\nabla^2 T^{n-1} - (\mathbf{u}\cdot\nabla T)^{n-1}\right) \\ & + \frac{5}{12}\left(\frac{1}{RePr}\nabla^2 T^{n-2} - (\mathbf{u}\cdot\nabla T)^{n-2}\right)\end{aligned} \tag{12}$$

C. Solver

Note that due to the explicit time integration of the convective terms, the spatially and temporally discretized counterparts of Eqs. (1) and (2) form a linear system of equations that must be solved at each time step. This linear system is solved using the direct sparse-matrix-solver routines of either the Intel Math Kernel Library or version 3.0 of the SuperLU matrix solver package [38]. This direct solution means that aside from round-off error, the errors in the numerical solution of Eqs. (1) and (2) are strictly the result of the spatial and temporal discretization, with no residual error in the solution of the system of equations resulting from the discretization. The spatially and temporally discretized counterpart of Eq. (3) does not require a solver, being fully explicit in T^{n+1}.

IV. Adjoint Equations and Gradient

A. Gradient of the Objective Function

In the present work, a conjugate gradient method is used to find the minimum of the objective function; therefore, the gradient of the objective function with respect to the control variables is needed. Note that the objective function (10) does not explicitly depend on the control-variable function $\phi[x, t]$, but does so only implicitly through the constraints defined by Eqs. (1–7); therefore, it is not a simple matter to obtain the gradient of the objective function with respect to the control variables. In the present work, the gradient is obtained by solving a set of adjoint equations, which are derived from the discretized versions of Eqs. (1–5).

B. Adjoint Equations

The goal is to determine the gradient of a scalar function given by Eq. (13) with respect to the discrete control variables $\phi_1, \ldots, \phi_M$,

$$H[u_1[\phi_1, \ldots, \phi_M], \ldots, u_N[\phi_1, \ldots, \phi_M], \phi_1, \ldots, \phi_M] \tag{13}$$

subject to the constraints given by Eq. (14), which are the discretized versions of Eqs. (1–5) (these constraints are also known as the state equations).

$$\begin{aligned}\mathbf{L}[\boldsymbol{\phi}, \mathbf{s}[\boldsymbol{\phi}]] &= \begin{pmatrix} L_1[u_1[\phi_1, \ldots, \phi_M], \ldots, u_N[\phi_1, \ldots, \phi_M], \phi_1, \ldots, \phi_M] \\ \cdots \\ L_N[u_1[\phi_1, \ldots, \phi_M], \ldots, u_N[\phi_1, \ldots, \phi_M], \phi_1, \ldots, \phi_M] \end{pmatrix} \\ &= \begin{pmatrix} 0 \\ \cdots \\ 0 \end{pmatrix}\end{aligned} \tag{14}$$

In Eqs. (13) and (14), $\boldsymbol{\phi}$ is the vector of control variables of dimension M; $\mathbf{s}[\boldsymbol{\phi}]$ is the vector of state variables (the discrete velocity, pressure, and temperature of the discretized Navier–Stokes and temperature equations) of dimension N; and $\mathbf{L}[\boldsymbol{\phi}, \mathbf{s}[\boldsymbol{\phi}]] = \mathbf{0}$ denotes the discretized Navier–Stokes and temperature equations of dimension N. Note that the state variables depend on the control variables (i.e., $u_1[\phi_1, \ldots, \phi_M], \ldots, u_N[\phi_1, \ldots, \phi_M]$); this dependence is described by $\mathbf{L}[\boldsymbol{\phi}, \mathbf{s}[\boldsymbol{\phi}]] = \mathbf{0}$. Writing a Taylor series expansion of the constraint equations around $\boldsymbol{\phi}$ yields Eq. (15).

$$\begin{aligned}\mathbf{L}[\boldsymbol{\phi} + \boldsymbol{\Delta}\phi, \mathbf{s}[\boldsymbol{\phi}] + \boldsymbol{\Delta}\mathbf{s}[\boldsymbol{\Delta}\boldsymbol{\phi}]] = \mathbf{L}[\boldsymbol{\phi}, \mathbf{s}[\boldsymbol{\phi}]] + \nabla_{\mathbf{s}}\mathbf{L}[\boldsymbol{\phi}, \mathbf{s}[\boldsymbol{\phi}]] \cdot \Delta\mathbf{s}[\Delta\boldsymbol{\phi}] \\ + \nabla_{\boldsymbol{\phi}}\mathbf{L}[\boldsymbol{\phi}, \mathbf{s}[\boldsymbol{\phi}]] \cdot \Delta\boldsymbol{\phi} + \mathcal{O}[\Delta\boldsymbol{\phi}^2]\end{aligned} \tag{15}$$

Assuming that the constraints given by Eq. (14) are satisfied yields $\mathbf{L}[\boldsymbol{\phi}, \mathbf{s}[\boldsymbol{\phi}]] = \mathbf{0}$. Assuming that the constraints are satisfied when moving $\boldsymbol{\Delta\phi}$ in control-variable space yields

$$\mathbf{L}\,[\boldsymbol{\phi} + \boldsymbol{\Delta\phi}, \mathbf{s}[\boldsymbol{\phi}] + \boldsymbol{\Delta}\mathbf{s}[\boldsymbol{\Delta\phi}]] = \mathbf{0}$$

Therefore, if the higher-order terms of the Taylor series expansion are neglected, then

$$\nabla_{\mathbf{s}}\mathbf{L}[\boldsymbol{\phi}, \mathbf{s}[\boldsymbol{\phi}]] \cdot \boldsymbol{\Delta}\mathbf{s}[\boldsymbol{\Delta\phi}] + \nabla_{\boldsymbol{\phi}}\mathbf{L}[\boldsymbol{\phi}, \mathbf{s}[\boldsymbol{\phi}]] \cdot \boldsymbol{\Delta\phi} = 0 \tag{16}$$

If Eq. (16) is satisfied, then for any vector $\boldsymbol{\lambda}$, Eq. (17), which is a scalar, holds.

$$(\nabla_{\mathbf{s}}\mathbf{L}[\boldsymbol{\phi}, \mathbf{s}[\boldsymbol{\phi}]] \cdot \boldsymbol{\Delta}\mathbf{s}[\boldsymbol{\Delta\phi}] + \nabla_{\boldsymbol{\phi}}\mathbf{L}[\boldsymbol{\phi}, \mathbf{s}[\boldsymbol{\phi}]] \cdot \boldsymbol{\Delta\phi}) \cdot \boldsymbol{\lambda} = 0 \tag{17}$$

The change in the objective function (ΔH, a scalar) for a small change in $\boldsymbol{\phi}$ is given by

$$\Delta H = \nabla_{\mathbf{s}}H \cdot \boldsymbol{\Delta}\mathbf{s}[\boldsymbol{\Delta\phi}] + \nabla_{\boldsymbol{\phi}}H \cdot \boldsymbol{\Delta\phi} + \mathcal{O}[\boldsymbol{\Delta\phi}^2] \tag{18}$$

Neglecting the higher-order terms of Eq. (18) and adding Eqs. (17) and (18), yields

$$\begin{aligned}\Delta H = \nabla_{\mathbf{s}}H \cdot \boldsymbol{\Delta}\mathbf{s}[\boldsymbol{\Delta\phi}] + \nabla_{\boldsymbol{\phi}}H \cdot \boldsymbol{\Delta\phi} \\ + (\nabla_{\mathbf{s}}\mathbf{L}[\boldsymbol{\phi}, \mathbf{s}[\boldsymbol{\phi}]] \cdot \boldsymbol{\Delta}\mathbf{s}[\boldsymbol{\Delta\phi}] + \nabla_{\boldsymbol{\phi}}\mathbf{L}[\boldsymbol{\phi}, \mathbf{s}[\boldsymbol{\phi}]] \cdot \boldsymbol{\Delta\phi}) \cdot \boldsymbol{\lambda}\end{aligned} \tag{19}$$

Rearranging Eq. (19) yields

$$\begin{aligned}\Delta H = \nabla_{\boldsymbol{\phi}}H \cdot \boldsymbol{\Delta\phi} + (\nabla_{\boldsymbol{\phi}}\mathbf{L}[\boldsymbol{\phi}, \mathbf{s}[\boldsymbol{\phi}]] \cdot \boldsymbol{\Delta\phi}) \cdot \boldsymbol{\lambda} + \nabla_{\mathbf{s}}H \cdot \boldsymbol{\Delta}\mathbf{s}[\boldsymbol{\Delta\phi}] \\ + (\nabla_{\mathbf{s}}\mathbf{L}[\boldsymbol{\phi}, \mathbf{s}[\boldsymbol{\phi}]] \cdot \boldsymbol{\Delta}\mathbf{s}[\boldsymbol{\Delta\phi}]) \cdot \boldsymbol{\lambda}\end{aligned} \tag{20}$$

The goal is to eliminate all derivatives with respect to $\mathbf{s}$ from the equation for the change in the objective function, leaving only derivatives with respect to the vector of control variables $\boldsymbol{\phi}$. This will be accomplished if

$$\nabla_{\mathbf{s}}H \cdot \boldsymbol{\Delta}\mathbf{s}[\boldsymbol{\Delta\phi}] + (\nabla_{\mathbf{s}}\mathbf{L}[\boldsymbol{\phi}, \mathbf{s}[\boldsymbol{\phi}]] \cdot \boldsymbol{\Delta}\mathbf{s}[\boldsymbol{\Delta\phi}]) \cdot \boldsymbol{\lambda} = 0 \tag{21}$$

Rewriting Eq. (21) yields

$$(\nabla_{\mathbf{s}}H + \nabla_{\mathbf{s}}\mathbf{L}[\boldsymbol{\phi}, \mathbf{s}[\boldsymbol{\phi}]] \cdot \boldsymbol{\lambda}) \cdot \boldsymbol{\Delta}\mathbf{s}[\boldsymbol{\Delta\phi}] = 0 \tag{22}$$

Now for the scalar Eq. (22) to be true for any vector $\boldsymbol{\Delta}\mathbf{s}[\boldsymbol{\Delta\phi}]$, the following vector equation must be satisfied.

$$\nabla_{\mathbf{s}}H + \nabla_{\mathbf{s}}\mathbf{L}[\boldsymbol{\phi}, \mathbf{s}[\boldsymbol{\phi}]] \cdot \boldsymbol{\lambda} = \mathbf{0} \tag{23}$$

After solving the system of Eqs. (23) (which are called the adjoint equations) for $\boldsymbol{\lambda}$ (the adjoint variables), Eq. (20) becomes

$$\Delta H = \nabla_{\boldsymbol{\phi}}H \cdot \boldsymbol{\Delta\phi} + (\nabla_{\boldsymbol{\phi}}\mathbf{L}[\boldsymbol{\phi}, \mathbf{s}[\boldsymbol{\phi}]] \cdot \boldsymbol{\Delta\phi}) \cdot \boldsymbol{\lambda} \tag{24}$$

Rearranging and simplifying yields

$$\Delta H = (\nabla_{\boldsymbol{\phi}}H + \nabla_{\boldsymbol{\phi}}\mathbf{L}[\boldsymbol{\phi}, \mathbf{s}[\boldsymbol{\phi}]] \cdot \boldsymbol{\lambda}) \cdot \Delta\boldsymbol{\phi} \tag{25}$$

where the term within ( ) is the final expression for the gradient of the objective function with respect to the vector of control variables $\boldsymbol{\phi}$.

Equation (23) can also be obtained by considering Eqs. (1–5) and (10) as a constrained optimization problem in which the goal is to minimize Eq. (10) subject to the constraints of Eqs. (1–5), and addressing it using Lagrange multipliers, the resulting Lagrange multipliers can be shown to be the adjoint variables.

C. Discussion of Adjoint Equations

At this point, it is appropriate to mention some qualities of the adjoint equations (23). Derived from the discretized Navier–Stokes and temperature equations, the adjoint equations have a form very similar to them. In fact, the adjoint equations arising from considering only the linear parts of the discretized Navier–Stokes and temperature equations (i.e., mass conservation and diffusion) have a coefficient matrix multiplying the adjoint variables that is the transpose of the coefficient matrix multiplying the linear part of the discretized Navier–Stokes and temperature equations. This similarity to the discretized Navier–Stokes equations means that solution methods used for the discretized Navier–Stokes and temperature equations can be used with little modification to solve the adjoint equations. Although the present work uses the same sparse direct solver to solve the linear systems arising from the discretized Navier–Stokes and temperature equations and the adjoint equations, iterative methods are also commonly used. The convergence rate of iterative methods is highly dependent on the eigenvalues of the matrix of the linear system and, of great importance for solution of the adjoint equations, the eigenvalues of a matrix are not changed by taking its transpose. Another important property of the adjoint equations is that they are linear in the adjoint variables, and therefore their solution should be easier than those for the nonlinear Navier–Stokes and temperature equations. Although the temporal integration of the spatially discretized Navier–Stokes and temperature equations in the present work has resulted in a linear system at each time step, implicit temporal integration of the convective terms would result in a nonlinear system of equations at each time step. However, regardless of the temporal integration of the spatially discretized Navier–Stokes and temperature equations, the system of equations defining the adjoint variables is always linear.

An impediment to the solution of the adjoint equations is that they depend on the state variables (i.e., the discrete velocity, pressure, and temperature). When the state equations are the unsteady Navier–Stokes and temperature equations as in the present work, this means that in addition to being stored at every spatial location, the discrete velocity, pressure, and temperature must also be stored at every temporal location in the time segment over which the gradient is desired. This requirement presents a severe computational restriction caused by the storage requirement of the temporal dimension. However, aside from this storage requirement, the adjoint equations provide a way to obtain the gradient of the objective function with respect to the control variables, the computational cost of which is independent of the number of control variables.

There are two methods of obtaining the adjoint equations: In the first, described in Sec. IV.B and used in the present work, the adjoint equations are derived from the discretized state equations (1–5). In the second, they are derived from the continuous state equations (1–5) and then discretized. The two methods are not identical on any finite grid spacing. Deriving the adjoint equations from the discretized equations instead of deriving them from the continuous equations and then discretizing the continuous adjoint equations has benefits: the primary one is that the gradient thereby computed is exact and can be verified on any grid using a finite difference method. As the grid spacing is reduced, the two methods become the same. However, the methods are not the same on practical grid sizes; one difference is that the gradient computed from the second method may be significantly different from the exact gradient and may thereby cause the search direction in the conjugate gradient scheme to not be a descent direction. However, both methods have been successfully used for problems in which the physical accuracy of the simulation is important, which implies that the grid spacing is small enough that it is not a significant factor in the numerical model, and therefore the two methods would be expected to be relatively close. When only qualitative agreement with the physics is needed (and therefore the grid spacing is large, such as in computer animations [36,37]), the first method is invariably used. Comparisons between the two methods are given in [39,40].

It is possible to derive the adjoint equations automatically from the discrete state equations using an automatic-differentiation software package, which takes a computer program that implements the state equations as input and returns software code for the adjoint equations [41,42]. This method is not used in the present work for two reasons: the first is that the resulting software code for the adjoint equations is generally inefficient and the second is that automatic-differentiation software has difficulties if the computer program that implements the state equations is a parallel program using a message-passing library such as the Message-Passing Interface (MPI). Because efficient code is of great importance in the present work and the computer code used in the present work is parallelized using domain decomposition and the MPI, the software code for the adjoint equations was derived by hand. Because this is an extremely tedious and error-prone process, care was taken to verify that the gradient computed by the hand-coded adjoint equations is correct by comparing it with the gradient computed using a finite difference method.

In a finite difference method, the gradient is obtained from

$$\frac{\partial H}{\partial \phi_k} = \frac{H[T[\phi_1, \phi_k + \varepsilon, \ldots, \phi_K]] - H[T[\phi_1, \phi_k, \ldots, \phi_K]]}{\varepsilon}$$

where ε is a small positive number; each component of the gradient is obtained by perturbing each respective control variable in turn. Computationally, this is very demanding, because the objective function must be evaluated for each perturbed control variable. In the present work, there are up to 2,019,840 control variables (see Sec. VIII.E.3), each of which would have to be perturbed, meaning that the computationally demanding Navier–Stokes and temperature equations would have to be solved 2,019,840 times to determine the gradient. Because the gradient will need to be computed potentially thousands of times during the optimization process, this approach is computationally prohibitive. In addition, tests must be performed to determine an appropriate value for ε, because the expression for the gradient is very sensitive to round-off error.

The finite difference schemes described in Sec. III and the adjoint equations derived from them were implemented in a Fortran 95 computer program called Tetra, developed by the author. All results in the present work were obtained from this computer program. Additional details concerning Tetra can be found in [3].

V. Physical Problem and Grid Independence

A. Physical Problem

Figure 1 describes the physical problem, dimensions, and boundary conditions of the present work. The numerical values of the dimensions are $L_{\text{jet}} = 0.3$, $L_w = 2.25$, $L_y = 8$, $L_x = 20$, $L\phi_{\text{begin}} = 0.5$, and $L\phi_{\text{end}} = 15$. At $x = 0$ $(0 \le y \le L_{\text{jet}})$, the boundary conditions are $u = U_0$, $v = 0$, and $T = 0$. The boundary $x = 0$ $(L_{\text{jet}} \le y \le L_w)$ is a no-slip wall, where $u = 0$, $v = 0$, and $T = 1$. The boundary $x = 0$ $(L_w \le y \le L_y)$ and $y = L_y$ $(0 \le x \le L_x)$ has boundary conditions of $u = U_0/2$, $v = 0$, and $T = 1$. At $x = L_x$, a convective outflow boundary condition is used for both velocity components and temperature. The bottom-wall boundary $y = 0$, $0 \le x \le L_x$ has a temperature boundary condition of $\partial T/\partial y = 0$. On this same boundary (where $L\phi_{\text{begin}} \le x \le L\phi_{\text{end}}$), the velocity boundary conditions are $u = 0$ and $v = \phi$; otherwise, they are $u = 0$ and $v = 0$. No control is applied to the wall jet $(0 \le y \le L_{\text{jet}})$. The problem is nondimensionalized by U_0, $(1/0.3)L_{\text{jet}}$ and $T_{\text{crossflow}}$. The Reynolds number based on U_0 and $(1/0.3)L_{\text{jet}}$ is 300; the Prandtl number is unity. Figure 2 shows an instance in time of the base (uncontrolled) flow. The unstable shear layer and the resultant vortex shedding can be clearly seen. Just past the jet exit there are three layers in the vertical direction of vorticity of alternating sign, with the top layer due to a strong recirculation region behind the vertical wall. At the end of the top layer of vorticity at $x \approx 3$, vortex induction between the bottom two layers of vorticity causes the wall jet to lift off of the wall. This liftoff is the cause of the failure of the film-cooling jet to protect the surface from the hot fluid.

B. Grid Independence

The outcome of a finite difference numerical approximation of a partial differential equation depends on the spatial and temporal step sizes and on the residual of the solution method used to solve the discretized equations. The residual is not an issue in the present work because a direct method is used, which solves the discretized equations such that the L_1 norm of their residual is close to machine precision. The effect of the spatial and temporal resolution is investigated using three grids (170×52, 337×103, and 673×205)

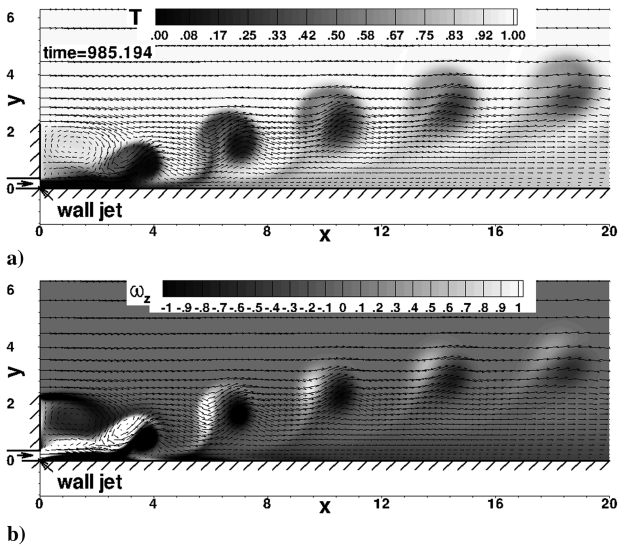

Fig. 2 Instantaneous view of flow without control: a) temperature and b) vorticity; 337 × 103 grid; every other fourth vector is shown.

Table 1 Grid dependence of Strouhal number and spatially and temporally averaged film-cooling effectiveness and wall shear (percentage is relative to the results from the finest grid)

Grid	Strouhal number	$\overline{T}_{\text{wall}}$ [Eq. (26)]	$\overline{\tau}_{\text{wall}}$ [Eq. (27)]
170×52	0.096 (11.1%)	0.399 (4.1%)	1.252 (21.1%)
337×103	0.106 (1.9%)	0.413 (.7%)	1.450 (8.6%)
673×205	0.108	0.416	1.586

in which the grid spacing is decreased by almost exactly one-half when moving to the next-finest grid. The time step used in Eqs. (11) and (12) is 0.012 on the 170×52 grid. It is halved on the 337×103 grid and, due to the viscous stability limit, is reduced by one-eighth on the 673×205 grid. On the 170×52 grid, the maximum Courant–Freidrichs–Levy number in the x direction is ~ 0.12; it is ~ 0.07 in the y direction. The grid spacing is smaller in the x direction near the origin, to increase resolution in the recirculation region (the ratio of the maximum grid spacing of the grid on which u is defined to the minimum in the x direction is 1.8 on the 337×103 grid); in the y direction, based on the grid on which v is defined, it is decreased near the wall to better resolve the boundary layer (the ratio of the maximum grid spacing to the minimum is 6.1 on the 337×103 grid). The difference in the primary quantity of interest in the present work, the wall temperature, defined by Eq. (26), along with the Strouhal number based on U_0 and the wall shear, defined by Eq. (27), are given in Table 1. These results demonstrate that for this Reynolds number, a solution independent of the grid and time step can be obtained on the 337×103 grid.

$$\overline{T}_{\text{wall}} = \frac{1}{L_x(960-480)} \int_{480}^{960} \int_0^{L_x} T[x,0,t]\,\mathrm{d}x\,\mathrm{d}t \tag{26}$$

$$\overline{\tau}_{\text{wall}} = \frac{1}{L_x(960-480)} \int_{480}^{960} \int_0^{L_x} \frac{\partial u}{\partial y}[x,0,t]\,\mathrm{d}x\,\mathrm{d}t \tag{27}$$

VI. Objective Functions and Control Variables

A. Control Variables

The control variables are functions of space and time. Ideally, it is desired to find the control over the long period of time given by t_1 and t_2 in Eq. (10). Because computational limitations prevent this, the time period is broken up into equally sized segments; the length of these segments is called the event horizon. Control is then found only within each event horizon; the total control is then obtained by piecing together (without any breaks or overlaps) the control found within each event horizon. An illustration of this temporal piecing together of the control is given in Fig. 3.

The control variables are the normal velocity along the bottom wall, as described in Sec. V.A and shown in Fig. 4. Therefore, the velocity at the grid points in which control is applied is $v = \phi$. There are 131 and 263 control variables at each time step for the 170×52 and 337×103 grids, respectively. It is necessary to impose bounds on the control variables such that the velocity at the grid points at which control is applied does not exceed physically realistic values; these velocities are restricted to be $\geq -0.4U_0$ and $\leq 0.4U_0$. This also ensures that the maximum Courant–Freidrichs–Levy number of the flow with control is not significantly different from that without control, which is necessary for numerical stability. This is accomplished by imposing an inequality bound of the form $|\phi| \leq 0.4U_0$ on each control variable.

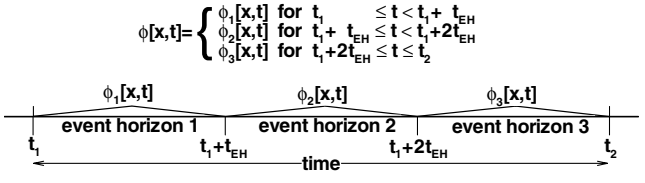

Fig. 3 Illustration of construction of the total control by the piecing together of the controls found within each event horizon.

B. Objective Functions

Because the goal of the present work is to minimize Eq. (10), the objective function used by the minimizer ideally should be defined by Eq. (10). However, due to finite computer resources, this is impractical because the time period $[t_1, t_2]$ in Eq. (10) is quite large, and therefore evaluation of Eq. (10) [especially the storage requirements of the adjoint equations (23)] is too computationally expensive. Therefore, the objective function can be defined only over a much smaller time period. This finite period of time (commonly called the event horizon and abbreviated in the present work as t_{EH}) is the distance forward in time over which the flow is evolved and in which new control variables are generated to minimize the objective function. It is therefore the length of time over which the gradient of the objective function with respect to the control variables must be computed. Clearly, the longer the event horizon, the more control exists over the objective function. However, this must be balanced against the increased computational expense of longer event horizons. An important computational limitation is that the velocity and (if the objective function depends on them) the pressure and temperature must be stored over the entire event horizon for use in solving the adjoint equations given by Eq. (23) (the solution of which is needed to determine the gradient).

Because it is computationally impractical to use the desired objective function, the suitability of other objective functions can be explored. Another objective function used in the present work is based on the turbulent kinetic energy. The rationale behind considering an objective function based on the turbulent kinetic energy is that it will target the unsteady vortex shedding, which increases the wall temperature by convecting the cold-wall jet away from the wall while simultaneously breaking it up.

C. Spatial Measures

The spatial measures used in defining the objective functions are given by Eq. (28), which describes the temporally varying spatially averaged film-cooling effectiveness, and by Eq. (29), which describes the temporally varying spatially averaged turbulent kinetic energy.

$$\eta[t] = \frac{1}{L_x} \int_0^{L_x} (1 - T[x,0,t])\,\mathrm{d}x \tag{28}$$

$$\begin{aligned} k[t,\gamma] &= \frac{1}{L_y(\beta_2-\beta_1)} \int_{\beta_1}^{\beta_2} \left(\int_{y_2}^{y_{j\max}} \left(\frac{1}{t_{\text{EH}}} \int_{(\gamma-1)t_{\text{EH}}}^{\gamma t_{\text{EH}}} u[x,y,t]\,\mathrm{d}t - u[x,y,t] \right)^2 \right. \\ &\quad \left. + \int_{yv_2}^{yv_{j\max-1}} \left(\frac{1}{t_{EH}} \int_{(\gamma-1)t_{\text{EH}}}^{\gamma t_{\text{EH}}} v[x,y,t]\,\mathrm{d}t - v[x,y,t] \right)^2 \right) \mathrm{d}y\,\mathrm{d}x \end{aligned} \tag{29}$$

where $\gamma = 1, 2, \ldots$ indicates the discrete event horizons.

Note that the spatial average in Eq. (29) is not over the entire range in the x direction with $\beta_1 = .5$ and $\beta_2 = 19$. Note also that this measure is specific to each event horizon, because the temporal mean

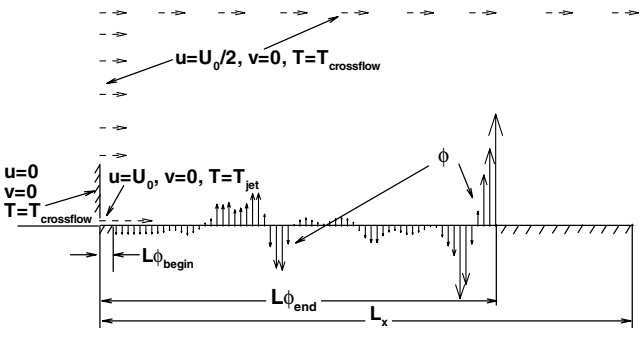

Fig. 4 Location of control variables, represented by solid vectors.

is carried out over the time interval defined by the event horizon; the integral

$$\int_{(\gamma-1)t_{\mathrm{EH}}}^{\gamma t_{\mathrm{EH}}}$$

is to be understood to be carried out over the unique event horizon to which t belongs. The integrals in Eq. (29) are only over the interior of the domain; therefore, the control on the wall is not included in Eq. (29). In Eq. (29), y_2 and $y_{j\max}$ are the y locations of the first main grid point off of the wall and the last grid point immediately off of the freestream boundary; yv_2 and $yv_{j\max-1}$ are the same for the locations at which the vertical component of velocity is stored.

D. Temporal Weighting

The spatial measures of interest must have the temporal dimension removed to define an objective function within each event horizon. There are many ways of doing this; in the present work, two approaches are investigated. The first is an average over the event horizon, and the second is to only consider the measure at the final (terminal) time step of the event horizon. Applying the two temporal weightings to the two spatial measures yields the four objective functions (30) studied in the present work. The desired objective functions

$$H_d = \left\{ \begin{array}{c} \bar{\eta} = \frac{1}{t_{\mathrm{EH}}} \int_{(\gamma-1)t_{\mathrm{EH}}}^{\gamma t_{\mathrm{EH}}} (1-\eta[t])^2 \,\mathrm{d}x\,\mathrm{d}t \\ \bar{k} = \frac{1}{t_{\mathrm{EH}}} \int_{(\gamma-1)t_{\mathrm{EH}}}^{\gamma t_{\mathrm{EH}}} k[t,\gamma]\,\mathrm{d}t \\ \eta_{\mathrm{terminal}} = (1-\eta[t_b+\gamma t_{\mathrm{EH}}])^2 \\ k_{\mathrm{terminal}} = k[t_b+\gamma t_{\mathrm{EH}},\gamma] \end{array} \right\} \tag{30}$$

where $\eta[\,]$ is given by Eq. (28), $k[\,]$ is given by Eq. (29), and t_b is the time that the optimization begins. Note that the terminal objective functions η_{terminal} and k_{terminal} do not penalize the spatial measures at any point in time other than the final one of each event horizon. Therefore, the optimizer is free to increase the spatial measures at any time other than the final time step of the event horizon, if doing so results in a decrease in the measure at the final time step of the event horizon. If the objective function depends only on temperature, then there is no influence of the velocity at the end of the event horizon on the objective function, due to the explicit time integration of the convective terms in the temperature equation. This is addressed by removing the last time step from the control-variable temporal dimension; in this case, the control is set to zero for the last time step.

E. Zero-Mass- and Energy-Flux Constraints

For the optimization problem to be physically realistic, mass- and energy-flux constraints must be imposed on the control variables. These mass- and energy-flux constraints are defined by Eqs. (31) and (32), respectively, which are the discrete counterparts of Eqs. (6) and (7). The mass-flux constraint is always linear, whereas the energy-flux constraint is linear if the temperature is assumed to not depend on the control variables.

The discrete mass-flux constraint is that for each time step n,

$$\sum_{i=i_b}^{i_e} \phi_i^n \Delta x_i = 0 \tag{31}$$

The discrete energy-flux constraint is that for each time step n,

$$\sum_{i=i_b}^{i_e} \phi_i^n T_{i,1}^n \Delta x_i = 0 \tag{32}$$

where i_b–i_e is the range of grid indices at which the control-variable indices are defined.

In the optimization process, new values of the control variables (at the $k+1$ iteration) are found by performing a line search by varying α along a search direction $\mathbf{S}$, where $\mathbf{S}$ is determined by a conjugate gradient algorithm (i.e., $\boldsymbol{\phi}^{k+1} = \boldsymbol{\phi}^k + \alpha\mathbf{S}$), where the control variables at the kth iteration are assumed to satisfy the zero-mass- and energy-flux constraints. These new control variables must satisfy the zero-mass- and energy-flux constraints at each time step. In general, the control variables $\boldsymbol{\phi}^{k+1}_{\mathrm{unconstrained}}$ along this line will not satisfy these constraints. New control variables $\boldsymbol{\phi}^{k+1}_{\mathrm{constrained}}$ that satisfy the constraints are sought that are as close as possible to those that do not satisfy the constraints. This can be accomplished by adding a perturbation vector $\boldsymbol{\varepsilon}$ to the control variables and minimizing the L_2 norm of this vector, subject to the constraint that the resulting control variables must satisfy the zero-mass- and energy-flux constraints. This defines the following optimization problem.

Minimize:

$$\begin{aligned} G &= (\boldsymbol{\phi}^{k+1}_{\mathrm{constrained}} - \boldsymbol{\phi}^{k+1}_{\mathrm{unconstrained}})^2 = (\boldsymbol{\phi}^k + \alpha\mathbf{S} + \alpha\boldsymbol{\varepsilon} - (\boldsymbol{\phi}^k + \alpha\mathbf{S}))^2 \\ &= \alpha^2\boldsymbol{\varepsilon}^2 \end{aligned} \tag{33}$$

Subject to the constraints given by Eqs. (31) and (32).

This optimization problem can be solved using Lagrange multipliers, leading to a linear system with dimensions of the number of control variables plus the number of constraints. Because there are two constraints at each time step, the number of constraints is twice the number of time steps. This linear system can be broken up into a series of smaller linear systems for the mass- and energy-flux constraints at each time step. A solution to each of these smaller linear systems can be obtained using a direct method, the computational cost of which scales linearly with the number of control variables. At each nth time step, the Lagrange multipliers are given by

$$\begin{aligned} \begin{pmatrix} \lambda_1^n \\ \lambda_2^n \end{pmatrix} &= \begin{pmatrix} -\frac{1}{2}\sum_{i=i_b}^{i_e} \Delta x_i \Delta x_i & -\frac{1}{2}\sum_{i=i_b}^{i_e} \Delta x_i \Delta x_i T_{i,1}^n \\ -\frac{1}{2}\sum_{i=i_b}^{i_e} \Delta x_i \Delta x_i T_{i,1}^n & -\frac{1}{2}\sum_{i=i_b}^{i_e} \Delta x_i \Delta x_i T_{i,1}^n T_{i,1}^n \end{pmatrix}^{-1} \\ &\times \begin{pmatrix} -\sum_{i=i_b}^{i_e} \Delta x_i S_i^n \\ -\sum_{i=i_b}^{i_e} \Delta x_i T_{i,1}^n S_i^n \end{pmatrix} \end{aligned} \tag{34}$$

After solving for $\boldsymbol{\lambda}$ at each time step, the solution for $\boldsymbol{\varepsilon}$ is given by

$$\varepsilon_i^n = -\tfrac{1}{2}(\Delta x_i \lambda_1^n + \Delta x_i T_{i,1}^n \lambda_2^n) \qquad i = i_b, \ldots, i_e \tag{35}$$

The result of solving this optimization problem is that the search direction $\mathbf{S}$ is projected into a subspace that satisfies the zero-mass- and energy-flux constraints, yielding a new search direction given by $\mathbf{S} + \boldsymbol{\varepsilon}$. This will result in satisfying both constraints for a specified spatial and temporal distribution of the temperature. However, as one moves in control-variable space in the direction defined by this new projected vector, the spatial and temporal distribution of the temperature will change and the constraints will likely not be satisfied. The fundamental problem is that the coefficients of the energy-flux constraint (32) are themselves functions of the temperature and therefore, through the solution of the Navier–Stokes and temperature equations, functions of the control variables. The results of this assumption are addressed in Sec. VIII.D. Note that the mass-flux constraint is always satisfied because the coefficients of this constraint do not change.

F. Penalty Term

The objective function seen by the conjugate gradient minimizer differs slightly from the desired objective function (30) by the inclusion of a penalty term in the objective function used by the optimizer. As discussed in Sec. VI.A, an inequality bound is imposed on the control variables of the form $|\phi| \le 0.4U_0$ to ensure that the velocity does not exceed physically realistic values. This is accomplished by using the penalty function given by Eq. (36) with $M = 10^5$.

$$H_p = \sum_{n=1}^{N} \left(\sum_{i=i_b}^{i_e} \left(\frac{\phi_i^n}{0.4U_0} \right)^{2M} \right) \tag{36}$$

This function has the desired property of being very close to zero when all $|\phi| < 0.4U_0$ and increasing extremely rapidly if any

$|\phi| > 0.4U_0$. The outer summation on the time index n is present because the penalty is present at each time step of the event horizon. The objective function used by the conjugate gradient minimizer H is the sum of the desired objective function H_d (30) and the penalty term H_p (36). In all results in the present work, the size of the penalty term is negligible, compared with the desired objective function.

G. Sensitivity of Objective Functions

An important consideration is the influence of the control variables on the objective function (i.e., the sensitivity of the objective function to the control variables), which is related to the gradient of the objective function. Because the objective function, and hence the gradient, may be scaled by an arbitrary amount, comparisons of the gradients of different objective functions with the goal of comparing the sensitivities should normalize the gradients by the objective functions. The control variables depend on both space and time. The spatial distribution of the control variables is fixed, however, the temporal distribution is not, because it depends on the event horizon. Therefore, the temporal history is of interest and the spatial distribution of the control variables can be addressed by a spatial norm.

Figure 5 shows the spatial L_∞ norm of the gradients of the objective functions normalized by the objective function. The figure should be understood as looking backward from time $t = 979.2$ (hence, the origin is reset to $t = 0$) and observing the sensitivity of control variables in the past. The gradient shown is before the gradient vector is projected into the linear subspace that satisfies the mass- and energy-flux constraints. Comparison of $\bar{\eta}$ to η_{terminal} shows that they have very different sensitivities, with a peak sensitivity for η_{terminal}. This peak sensitivity occurs $\sim$33 units of time from the end of the event horizon; this time scale is slightly more than that required by a particle to convect through the entire x dimension [the spatial integral in Eq. (28) is over the entire x dimension] at the average velocity of the wall jet. This not surprising, because $1/RePr \ll 1$ and the dominant means of transporting heat is convection. Past this peak, the sensitivity decays toward zero as time recedes. This is expected, because the influence of an event (i.e., a control variable) on the solution at a point in time "washes out" as the temporal distance between the event and the solution at the point in time goes to infinity. In comparison, the sensitivity of $\bar{\eta}$ shows no peak as it appears to asymptote past the peak sensitivities of η_{terminal}, which is expected, because the average over time always includes the peak sensitivity.

The sensitivity of k_{terminal} is different from that of η_{terminal} in that its maximum value is four times greater than that of η_{terminal} and it has a different shape. In particular, it does not decay away toward zero as does η_{terminal}. This behavior is the result of the temporal integral in Eq. (29), which defines the mean velocity within each event horizon and which equally weights the velocity at all time steps within the event horizon. Therefore, even though the influence of a temporally distant (from the end of the event horizon) control variable on the velocity at the end of the event horizon would be expected to be small, this control variable will still have some influence on the objective function because of the temporal integral in Eq. (29). As a result, once far enough away in time from the end of the event horizon, all control variables would be expected to reach a common sensitivity defined by the equal weighting of the temporal integral in Eq. (29).

Both objective functions based on the turbulent kinetic energy have greater sensitivity and, for the terminal objective function, have a peak sensitivity a shorter distance backward in time than the objective functions based on the temperature. This suggests that the k_{terminal} and $\bar{k}$ objective functions will be easier to minimize for short event horizons than the η_{terminal} and $\bar{\eta}$ objective functions. For all objective functions, the distinct frequency appearing in Fig. 5 is exactly that of the natural frequency (see Table 1) of the problem. For the terminal objective functions, the shape of the plots of the sensitivities with their distinct peak sensitivities suggests that as the length of the event horizon increases, the control effectiveness will

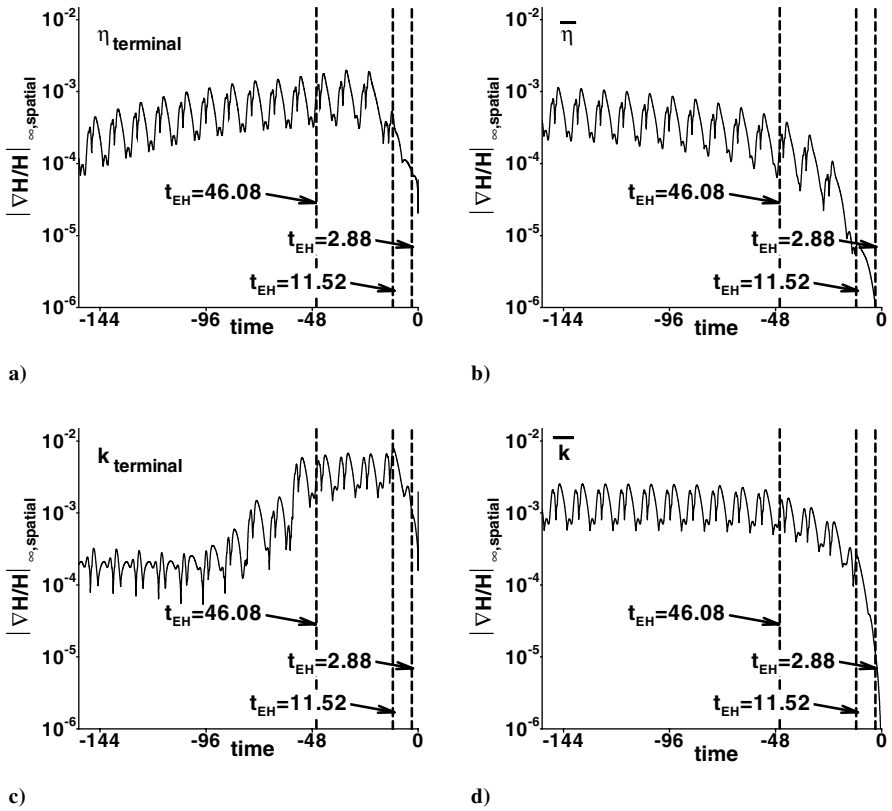

Fig. 5 Temporal history of the L_∞ spatial norm of the gradient of the control variables normalized by the value of the objective function at $t = 979.2$ before control is applied; dashed lines indicate the extent of the temporal domain of the control variables included by the different event horizons; no penalty term is used; 170 × 52 grid.

increase until the peak sensitivity lies within the event horizon, with further increases in the length of the event horizon unlikely to improve the control.

VII. Minimization Method

The optimization strategy in the present work uses the Fletcher–Reeves conjugate gradient method [43] with a one-dimensional line search. This method was chosen after tests showed it to be superior to the Polak–Ribiere and Hestenes–Stiefel conjugate gradient methods [43] for a related optimization problem. Results from using different conjugate gradient methods are discussed in Sec. VIII.E.2. The results and efficiency of a conjugate gradient method applied to a nonlinear problem depend heavily on the termination criteria and the line-search method; therefore, these are described in detail.

The optimization is terminated if either of two criteria is met. The first criterion is that the ratio of the objective function to the objective function at the beginning of the simulation before the optimization has begun (at $t = 979.2$) is less than 1×10^{-2}. Although this termination criterion is applied to all objective functions, it never actually comes into play for those based on the film-cooling effectiveness, because with a starting (before control is applied) film-cooling effectiveness of $\sim$0.55 and a maximum film-cooling effectiveness of 1, this termination criterion is never reached for the objective functions based on the film-cooling effectiveness. A test was performed for the $\bar{k}$ objective function with an event-horizon length of 2.88 using criteria of 1×10^{-1}, 1×10^{-2}, and 1×10^{-3}. The maximum difference in the film-cooling effectiveness (using the criterion of 1×10^{-3} as the base) was $\sim$1.1% for the criterion of 1×10^{-1} and $\sim$0.1% for the criterion of 1×10^{-2}. The second is a stalling criterion, which is met if the absolute value of the ratio of the maximum difference in the last five objective function values to the maximum of the last five objective function values is less than 10^{-3}. At the beginning of each new event horizon, the initial guess (needed by the optimizer) for the control variables is zero, which satisfies the zero-mass- and energy-flux constraints.

The golden-section line search is used for the one-dimensional search in control-variable space, terminating when the absolute value of the difference between successive values divided by the maximum of the absolute value of those values is less than 10^{-1}. If the line search initially fails to find a reduction in the objective function using the maximum-possible line-search line length, which is limited by the constraints on the control variables (Sec. VI.A), this length is reduced by one-eighth and a new search is begun. This process is repeated up to twice more; if all three fail, the search direction is then reset to the gradient direction and the objective function is computed a small (10^{-7}) distance along this line. If this objective function value is smaller than at the beginning of the line, then a new search is begun along this line. If not, or if the search fails, then the optimization is terminated for that event horizon.

VIII. Results and Discussion

A. Objective Functions η_{terminal}, $\bar{\eta}$, k_{terminal}, and $\bar{k}$

The flow is first evolved 979.2 units in time [i.e., $t_b = 979.2$ in Eq. (30)], after which time control is initiated. The control is continued until the flow has evolved a further 138.24 units in time. During this time, optimal values are computed for the control variables without any breaks (i.e., the control is on for the entire 138.24 units of time). Results are presented for four different event-horizon lengths of 0.72, 2.88, 11.52, and 46.08, along with the four different objective functions, described in Sec. VI.B. These event horizons are chosen so that they are integer multiples of each other and of the total length of time over which control is computed. The maximum length is chosen to be 46.02, because this length is sufficient to capture the peak sensitivities of all the objective functions.

Figure 6 shows that for all objective functions, at all event horizons, the film-cooling effectiveness is increased. In general, when the objective function is based on the film-cooling effectiveness, there is less increase in the film-cooling effectiveness than when the objective function is based on the turbulent kinetic energy. The author believes that the primary reason that the objective function based on the turbulent kinetic energy performs better is that it directly targets the primary reason for the breakdown of the film-cooling flow, which is the unsteady vortex shedding that transports the cold-wall jet away from the wall and transports hot fluid toward the wall. Because objective function $\bar{k}$ results in the greatest increase

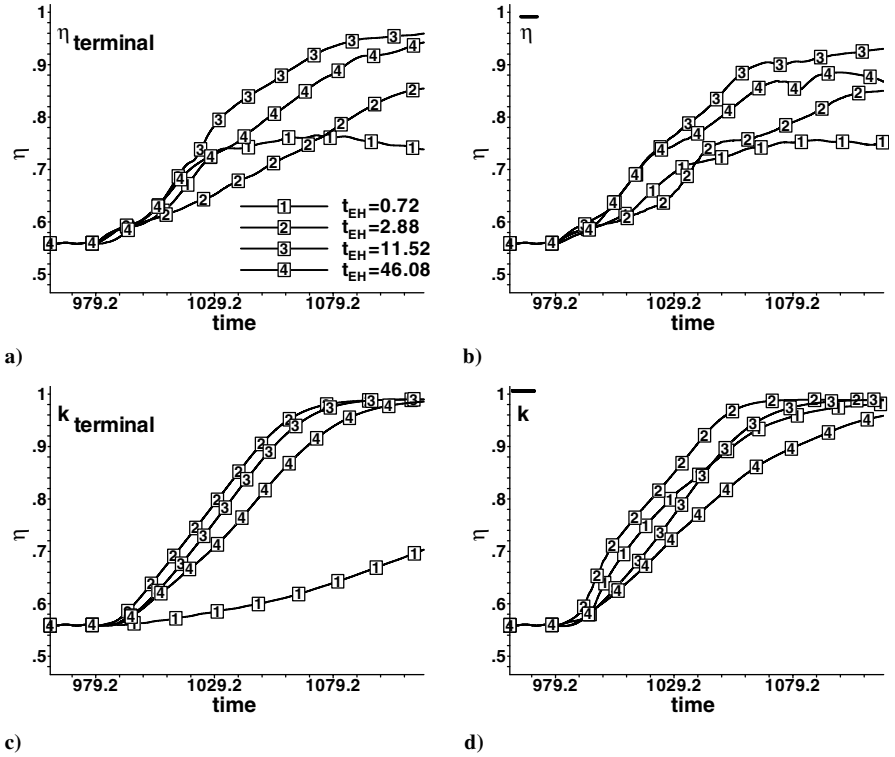

Fig. 6 Instantaneous film-cooling effectiveness for all objective functions at all event horizons.

in film-cooling effectiveness, the present work focuses attention on this objective function.

The data in Fig. 6 may appear counterintuitive, because better results are not always obtained from longer event horizons. For example, four event horizons of 11.52 can also be found within an event horizon of 46.08, because the control-variable space spanned by the event horizon of 46.08 includes all of the control-variable spaces spanned by four event horizons of 11.52. However, this plot concerns the film-cooling effectiveness, and two of the objective functions (Figs. 6c and 6d) are different measures of the turbulent kinetic energy. For these objective functions, the film-cooling effectiveness, indeed the entire temperature field, has no effect on the optimizer. For the two objective functions based on measures of the film-cooling effectiveness, however, this is counterintuitive. The reason why a larger event horizon will not always result in a lower value of the objective function is a result of the landscape of the objective function in control-variable space in conjunction with the type of optimizer used. The objective function is unlikely to have one local minimum, because the objective function depends on the nonlinear Navier–Stokes equations. A gradient-based method, as used in the present work, will find a local minimum quickly (which minimum it will find will depend on the starting guess and specifics of the line search). Stochastic methods do not use the gradient of the objective function and, as a result, are less likely to become trapped in a local minimum, but they also generally require many more evaluations of the objective function than do gradient-based methods. Because the evaluation of the objective function in the present work is extremely expensive, these methods are extremely inefficient. A potential solution to this problem is the use of stochastic methods to change the initial guess for the start of the gradient-based method.

B. Objective Function $\bar{k}$

Note that the time between successive optimization windows (i.e., the length of the event horizon) has frequencies of $1/0.72 = 1.338$, $1/2.88 = 0.347$, $1/11.52 = 0.087$, and $1/46.08 = 0.022$ for event-horizon lengths of 0.72, 2.88, 11.52, and 46.08, respectively. Because the natural frequency of the system is 0.106, only the event-horizon lengths of 11.52 and 46.08 are long enough to contain control with components at this frequency. Figure 7 shows the L_∞ spatial norm of the control in frequency space for the four event horizons. This norm is a good metric for energetic injection or sucking near the wall. The longest event horizon shows a distinct peak at the natural frequency, indicating that the optimizer is targeting the vortex shedding. By contrast, none of the other event horizons show any especially large component at the natural frequency. Note that the dominant component for all event horizons is the frequency with which the event horizons are repeated (although the frequency range shown does not include that for $t_{\mathrm{EH}} = 0.72$; this is also true for this event horizon). Figure 8 presents the temporal history of the L_∞ spatial norm of the control. Note that for the longest event horizon, the control decreases by three orders of magnitude, whereas for the shortest, the magnitude decreases only slightly. For event-horizon lengths of 11.52 and 46.08, the magnitude of the control always drops off sharply toward the end of the event horizon. This behavior can be explained by the sharply decreasing sensitivity of the objective function with respect to the control variables (see Fig. 5). This decrease at the end of the event horizon also occurs for event horizon lengths of 0.72 and 2.88, but not for every event horizon. This can be explained by the fact that the shorter the event horizon, the smaller the range of sensitivity within the event horizon. Although not shown, this decrease at the end of the event horizon is even more pronounced for the $\bar{\eta}$ objective function. The discontinuity near $t = 1079.2$ in Fig. 8c is a result of the control being turned off when the objective function is reduced beyond a specified level (see Sec. VII for specifics).

Because the control can consist of both injection and suction, a comparison between them is of interest. Equation (37) provides a measure of the magnitude of injection ($v > 0$) to suction ($v < 0$). If $v_+ + v_-$ is positive, then the spatially averaged injection velocity is larger than the spatially averaged suction velocity. Because of the zero-mass-flux constraint, this also means that when Eq. (37) is positive, the area over which injection occurs is smaller (but the average velocity in this area is greater) than the area over which suction occurs.

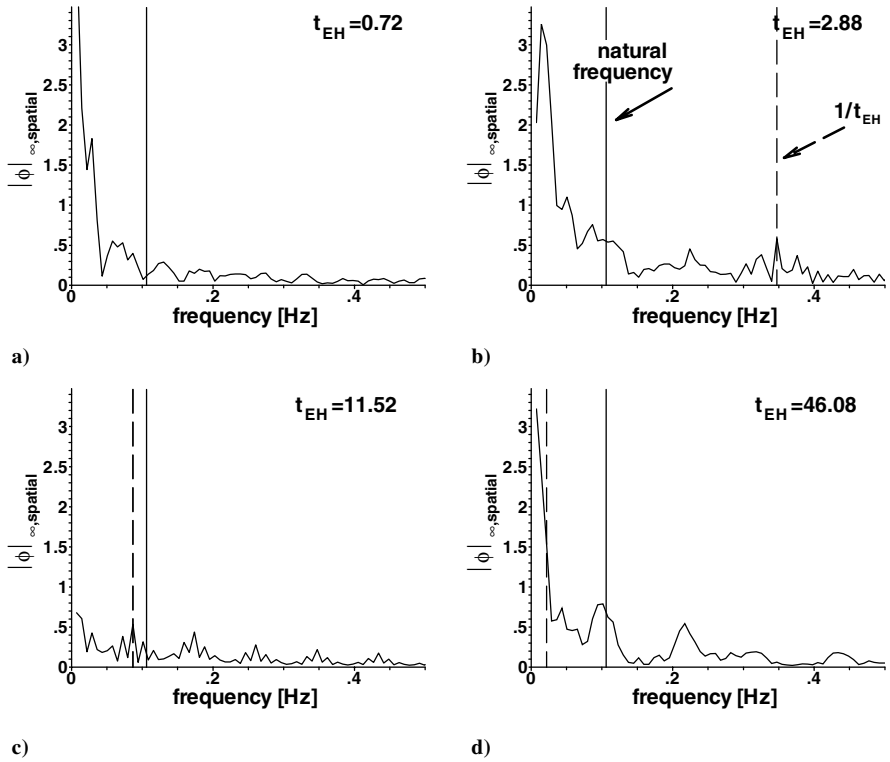

Fig. 7 L_∞ spatial norm of the control in frequency space for the $\bar{k}$ objective function at four event horizons; frequency $1/t_{\mathrm{EH}}$ with which the event horizons are repeated (dashed line) and natural frequency of 0.106 (solid line).

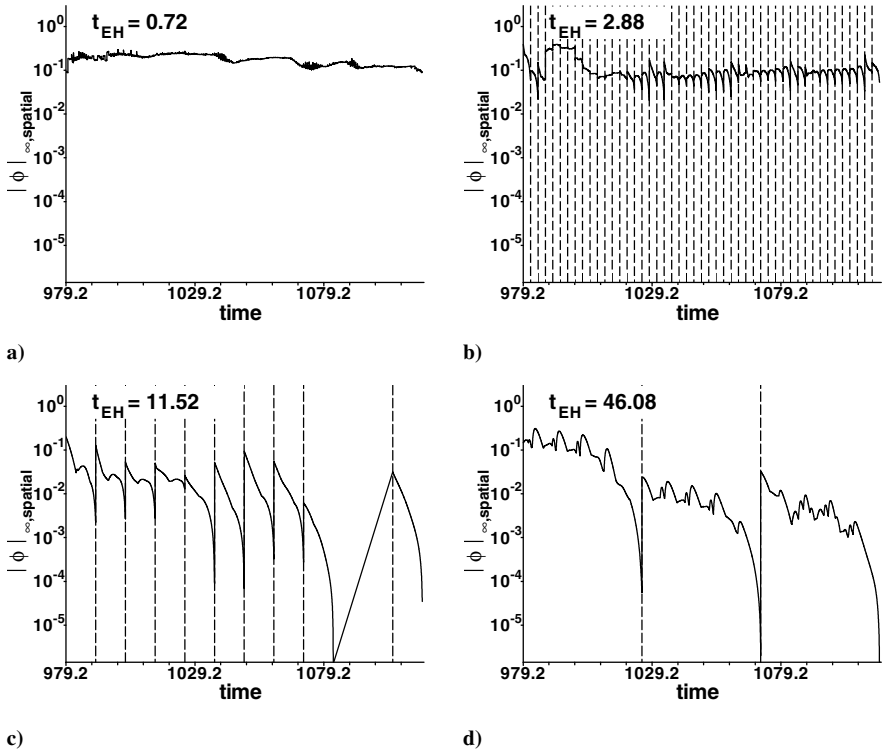

Fig. 8 L_∞ **spatial norm of control as a function of time for the** $\bar{k}$ **objective function at four event horizons; for** $t_{\mathrm{EH}} = 11.52$**, the discontinuity at** $t \approx 1079.2$ **is the result of the control being turned off because the objective function had been reduced to the desired level (see Sec. VII for details).**

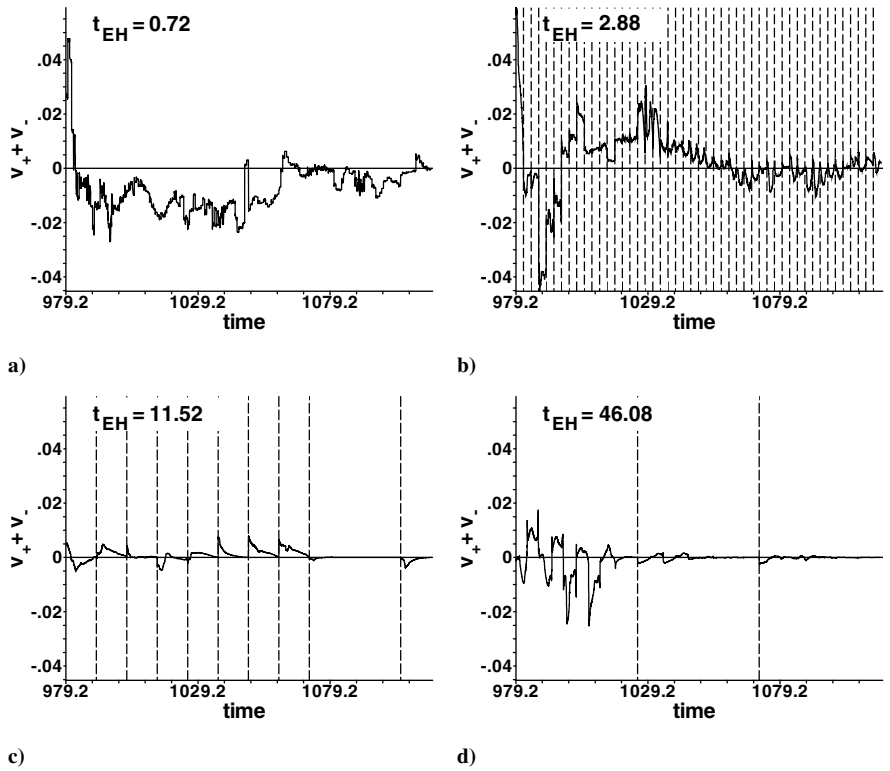

Fig. 9 Plot of $v_+ + v_-$ **given by Eq. (37); dashed lines indicate boundaries between event horizons. These boundaries are not shown for** $t_{\mathrm{EH}} = 0.72$**, because the large number of them would obscure the figure. For** $t_{\mathrm{EH}} = 11.52$**, the missing boundaries are the result of the control being turned off, because the objective function had been reduced to the desired level (see Sec. VII for details).**

$$v_+ + v_- = \frac{1}{\partial\Omega_+}\int^{\partial\Omega_+} v[x,0,t]\,\mathrm{d}x + \frac{1}{\partial\Omega_-}\int^{\partial\Omega_-} v[x,0,t]\,\mathrm{d}x \quad (37)$$

where $\partial\Omega_+(\partial\Omega_-)$ is that part of $\partial\Omega\phi$ where $v > (<)0$.

Figure 9 shows $v_+ + v_-$ in the temporal domain for the four event horizons. The average of $v_+ + v_-$ over the time interval is -6.4×10^{-3}, 2.2×10^{-3}, 5.0×10^{-4}, and -5.6×10^{-4} for event horizons of 0.72, 2.88, 11.52, and 46.08, respectively. This shows

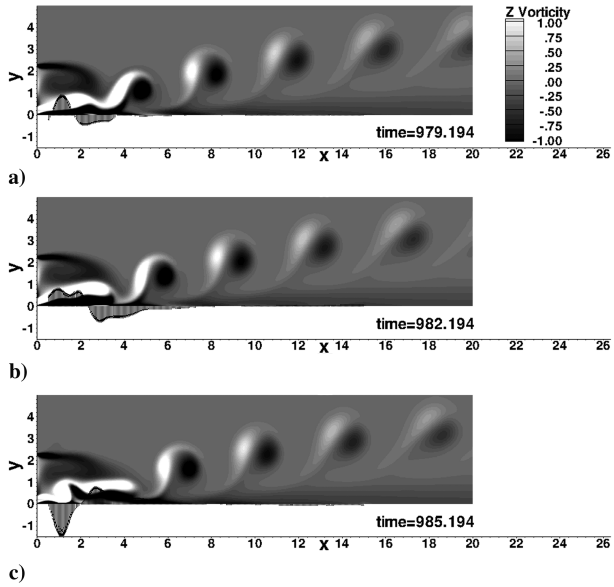

Fig. 10 Contours of vorticity with the control superimposed at three different times for the $\bar{k}$ objective function; vectors are scaled by 100; $t_{\mathrm{EH}} = 46.08$.

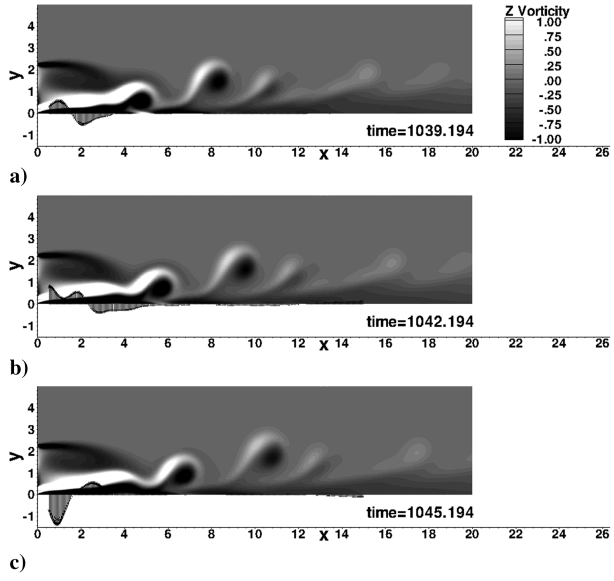

Fig. 11 Contours of vorticity with the control superimposed at three different times for the $\bar{k}$ objective function; vectors are scaled by 100; $t_{\mathrm{EH}} = 46.08$.

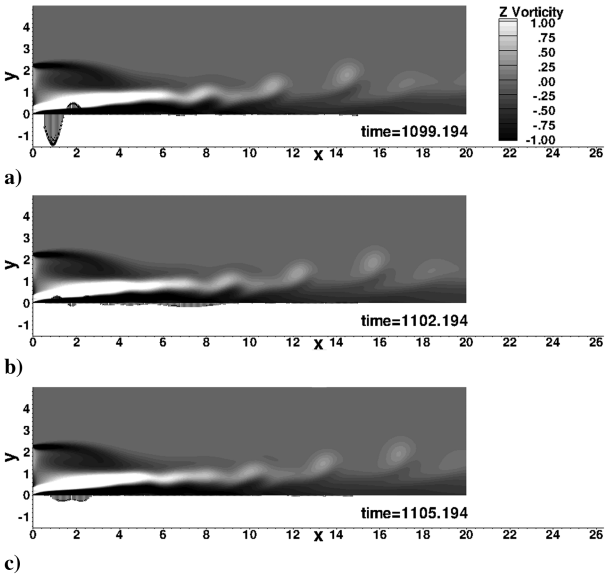

Fig. 12 Contours of vorticity with the control superimposed at three different times. The vectors are scaled by 275. $\bar{k}$ objective function, $t_{\mathrm{EH}} = 46.08$.

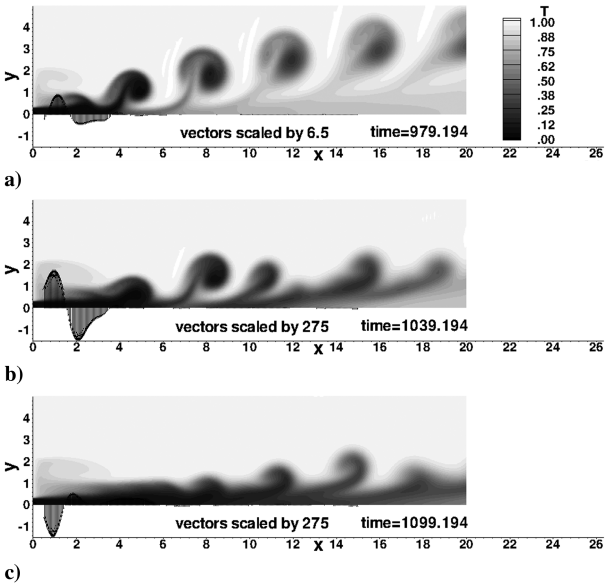

Fig. 13 Contours of temperature with the control superimposed at three different times for the $\bar{k}$ objective function; vector scaling is not the same at each time step, but is indicated on the plot; $t_{\mathrm{EH}} = 46.08$.

that, on average, the velocity magnitude of injection is essentially the same as suction. However, Fig. 9 shows that instantaneously there are significant differences in which $v_+ + v_-$ is an order of magnitude larger than its average. For the longest event horizon, the frequency of the sign change of $v_+ + v_-$ is that of the natural frequency. This indicates that the control for this event horizon, which is approximately four times longer than the period of the natural frequency, is targeting components of the objective function at the natural frequency. This is not seen for the other event horizons, in which the primary frequency component is that at which the event horizon is repeated.

Figures 10–12 show the control and the vorticity in a sequence of images at different time steps, showing the reattachment of the jet to the wall. Note that the scaling of the vectors representing the control in Fig. 12 is 2.75 times greater than that in Figs. 10 and 11. Almost all of the control is exerted close to the inflow boundary in which the inlet of the wall jet is located. This is where the recirculation region exists and the vortices form. Once the vortices are suppressed by this control near the inlet; very little control is needed further along the x axis to keep the jet attached to the wall. The length over which the control contiguously has the same sign is approximately equal to the length scale of the vortices. The control is evidently suppressing the vortex induction between the bottom two layers of vorticity of alternating sign. This vortex induction is what causes the jet to lift off of the wall and to break up. Once the jet is reattached, much less control energy is needed to keep it reattached (see Fig. 8). Figure 13 shows the temperature and the control at different time steps. The almost complete reattachment of the jet to the wall by the optimal control is clearly seen in this figure. Note that the vectors representing the control in Fig. 13 are scaled by $6.5/275$, relative to those in Figs. 13b and 13c.

C. Behavior of the $\bar{k}$ Objective Function

The behavior of the objective function (i.e., how it changes when the control variables change) is of interest, because it can indicate

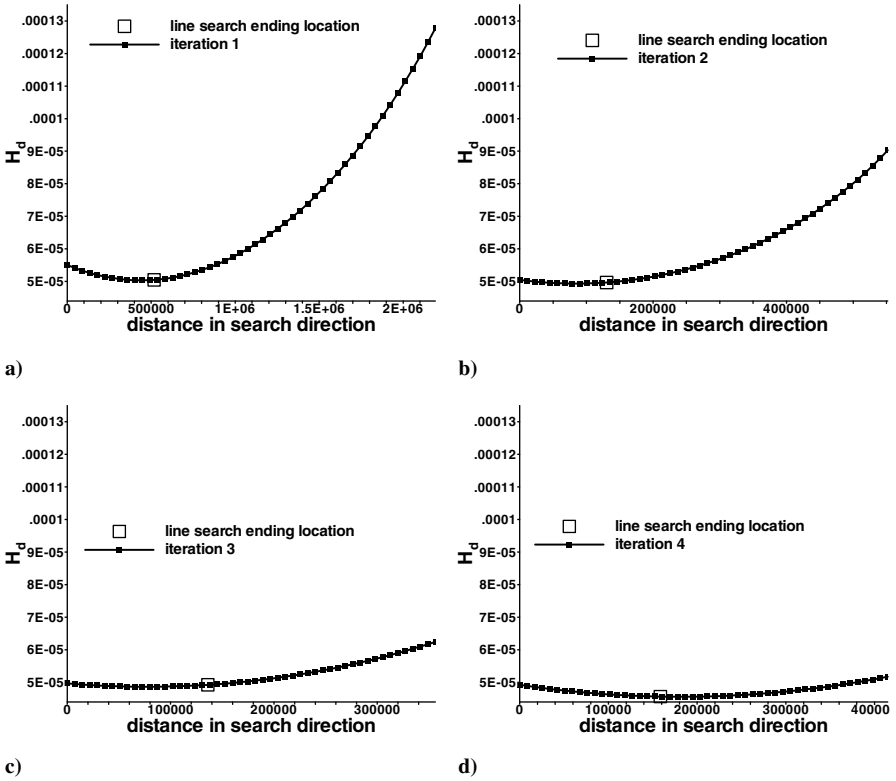

Fig. 14 Objective function sampled along the search direction. Solid squares indicate the sample points; large hollow square indicates termination location of the line section search. Event horizon of 0.72, search directions of the first four iterations of the conjugate gradient algorithm of the 28th event horizon.

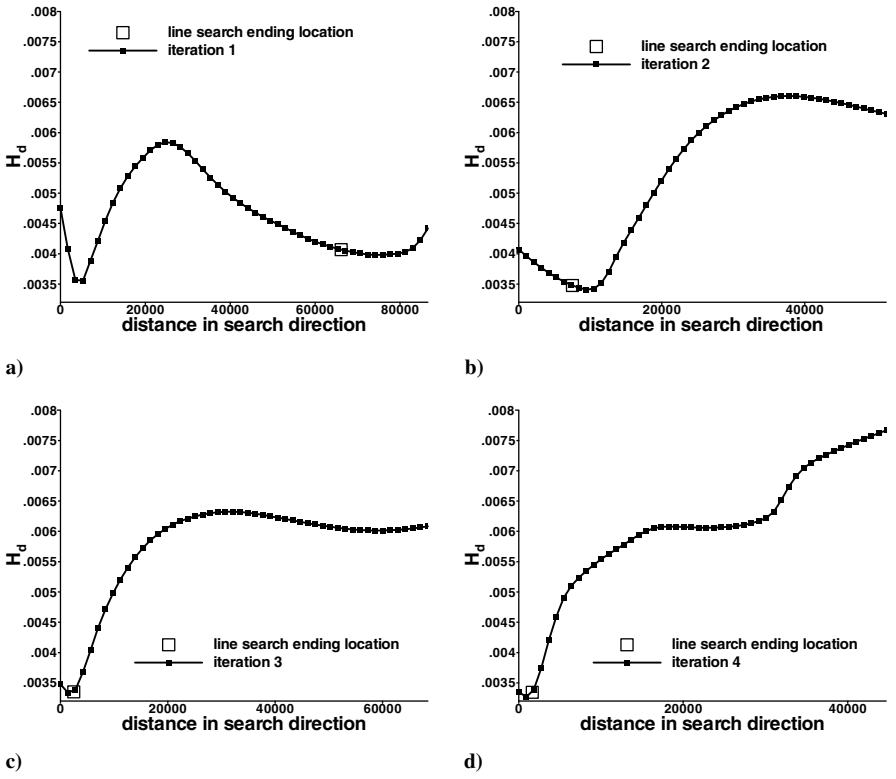

Fig. 15 Objective function sampled along the search direction. Solid squares indicate the sample points; large hollow square indicates termination location of the line section search. Event horizon of 46.08, search directions of the first four iterations of the conjugate gradient algorithm of the first event horizon.

whether the optimization algorithm is appropriate. In particular, the behavior of the objective function along the search direction is of great interest. Figures 14 and 15 show the change in the objective function when moving along the line in control-variable space defined by the search direction of the conjugate gradient algorithm for the $\bar{k}$ objective function at event-horizon lengths of 0.72 and 46.08. The line along which the objective function is sampled in Figs. 14 and 15 begins at the location in control-variable space in

Table 2 Energy flux through the part of the lower wall in which the control exists; 337 × 103 grid

Mass-flux constraint only	Mass- and energy-flux constraint
$t_{EH} = 0.72 \rightarrow 2.2 \times 10^{-1}$	$t_{EH} = 0.72 \rightarrow 8.9 \times 10^{-5}$
$t_{EH} = 2.88 \rightarrow 2.0 \times 10^{-1}$	$t_{EH} = 2.88 \rightarrow 1.5 \times 10^{-3}$
$t_{EH} = 11.52 \rightarrow 3.3 \times 10^{-3}$	$t_{EH} = 11.52 \rightarrow 9.8 \times 10^{-5}$
$t_{EH} = 46.08 \rightarrow 3.3 \times 10^{-3}$	$t_{EH} = 46.08 \rightarrow 5.9 \times 10^{-5}$

which the search direction is defined and continues in the search direction until the inequality constraints on the control variables are exceeded. There are 50 evenly spaced sample points, beginning at the point in control-variable space in which the search direction is defined and terminating when further movement in the search direction exceeds the inequality constraints on the control variables. Note that the desired objective function (30), which does not include the penalty term (36), is plotted. Because of the large exponent in Eq. (36), the difference between the desired objective function and the objective function plus the penalty is miniscule wherever the inequality bounds on the control variables are satisfied. Plotting the desired objective function avoids the irrelevant sharp spike in the penalty term in Eq. (36), which has a value greater than 1 at the termination of the line, where at least one of the control variables equals the value of the inequality bound on the control variables. For the event-horizon length of 0.72, the objective function is very well-behaved along the search direction with only one minimum, whereas at an event-horizon length of 46.08, the objective function behaves quite differently. For the event-horizon length of 46.08, the objective function is not unimodal. Figure 15b shows the objective function with one local minimum and Figs. 15a, 15c, and 15d show the objective function with two local minimums. This behavior can cause the line search to fail to find the global minimum; this occurs in Fig. 15a, in which the line search finds the higher local minimum.

The objective function along the line has a differing shape, depending on the specific event horizon and iteration; however, the basic characteristic of increasing complexity and nonlinearity holds as the event horizon is increased. Because of the relatively parabolic shape of the objective function along the search direction for the event-horizon length of 0.72, the golden-section line search could be replaced with a more efficient method involving finding the minimum of a parabola. Because the value of the objective function at the starting point of the parabola and the derivative at the starting point are known, only one additional evaluation of the objective function is needed to define a parabola. Although not shown, at event-horizons lengths of 2.88 and 11.52, the behavior of the objective function is very similar to that at an event-horizon length of 0.72, with the only multiple local minimum observed at an event-horizon length of 46.08. This behavior of the objective function at the event-horizon length of 46.08, in conjunction with the gradient-based optimization method, constitutes the explanation mentioned in Sec. VIII.A for why this event-horizon length does not result in higher values for the film-cooling effectiveness, compared with the event-horizon length of 11.52.

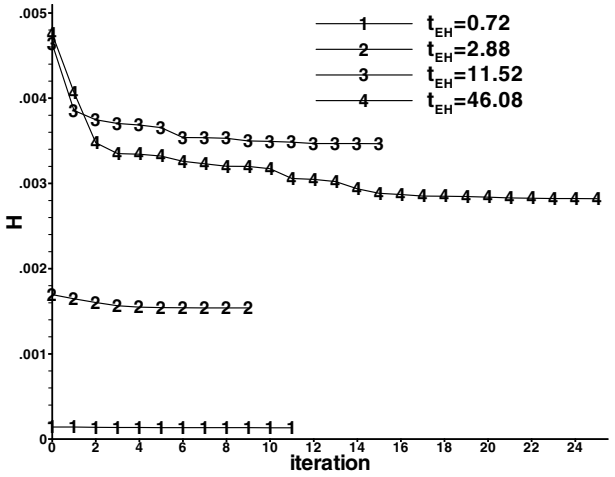

Fig. 16 Convergence history of the first event horizon for the $\bar{k}$ objective function.

Table 3 Total number of objective function evaluations needed for different event horizons and objective functions

	$t_{EH} = 0.72$	$t_{EH} = 2.88$	$t_{EH} = 11.52$	$t_{EH} = 46.08$
$k_{terminal}$	8375	1057	823	621
k	22522	1720	774	468

D. Energy-Flux Constraint

As mentioned in Sec. VI.E, the energy-flux constraint is not exactly satisfied because it is assumed to be a linear constraint, but in actuality, it depends on the control variables through the solution of the Navier–Stokes and temperature equations. Table 2 presents temporal norms of the spatially integrated energy flux

$$\left| \frac{1}{\partial\Omega_\phi} \int^{\partial\Omega_\phi} \phi[x] T[x, 0] \, \mathrm{d}x \right|$$

using the mass- and energy-flux constraints versus using only the mass-flux constraint. It can be seen that the use of the mass- and energy-flux constraints reduces the energy flux greatly. If only the mass-flux constraint is used, then the energy-flux constraint is violated to a much greater degree, by approximately two orders of magnitude. For comparison, if the wall jet had a temperature of 1, then the energy flux into the domain due to it would be 0.3.

E. Performance of the Minimizer

The convergence history of the $\bar{k}$ objective function for the first event horizon is shown in Fig. 16. The most noticeable feature is the increasing rate of decrease in the objective function as the length of the event horizon increases. This behavior is not surprising, because the longer the event horizon, the more control exists over the objective function. For all event horizons except the shortest, the objective function sharply decreases at first, before leveling out and eventually terminating the optimization, due to the stalling criteria described in Sec. VII. Upon termination of the optimization for the first event horizon, the L_∞ norm of the gradients is 1×10^{-7}, 2×10^{-7}, 8×10^{-7}, and 6×10^{-6} for event-horizons lengths of 0.72, 2.88, 11.52, and 46.08, respectively.

1. *Effect of Line-Search Termination Criteria*

In nonlinear optimization problems, a significant amount of computational time is commonly spent determining the minimum along a search direction. Table 3 presents the number of objective function evaluations needed for different event horizons and objective functions. The count of these objective function evaluations includes those at the origin of the search direction, which are computed when computing the gradient used to determine the search direction, plus the additional evaluations along the search direction needed to bracket the minimum during the line search. Because each evaluation requires a solution of the discretized versions of Eqs. (1–5) over the event horizon, this involves a substantial amount of computational work. The reason that the number of objective function evaluations does not decay linearly with the increase in the event-horizon length is because of the greater potential to reduce the objective function for longer event horizons (see Sec. VIII.E), and therefore more iterations required, before the stalling criteria (see Sec. VII) terminates the optimization for that event horizon.

In the present work, the line-search termination criterion given in Sec. VII is used. Analysis of the effect of the line-search termination criterion found that it has very little influence on the results. As an example, for the $\bar{k}$ objective function using the 170×52 grid and

event-horizon lengths of 0.72 and 46.08, there is a maximum difference of ∼4.2% in the film-cooling effectiveness during the time period in which the control is on, using the line-search criterion of 10^{-n}, where $n = 0, \ldots, 6$.

2. *Effect of the Conjugate Gradient Method*

Comparisons are made between the Fletcher–Reeves, the Hestenes–Stiefel, the Polak–Ribiere, and the steepest-descent methods [43] for the $\bar{\eta}$ and $\bar{k}$ objective functions at event-horizon lengths of 0.72 and 11.52. Taking the Fletcher–Reeves method as the base, the greatest difference in the film-cooling effectiveness over the entire time that the optimization is on is 13% for the steepest-descent methods for the $\bar{k}$ objective function at an event-horizon length of 0.72. For this case, the Fletcher–Reeves method performs best early in time, but all methods converge toward the end of the time period. For objective function $\bar{\eta}$ at an event-horizon length of 11.52, the steepest-descent method performs best (by ∼4%) at the end of the time period, with the Fletcher–Reeves method performing worst and the Hestenes–Stiefel and Polak–Ribiere methods lying in between. For objective function $\bar{k}$ at an event horizon of 11.52 and objective function $\bar{\eta}$ at an event-horizon length of 0.72, there is very little difference between the methods.

3. *Computational Requirements*

The nonlinear optimization problems solved within each event horizon in the present work are computationally challenging in their own right without considering the necessity of solving them for numerous event horizons to determine the control variables for the entire temporal domain of interest. The control-variable space in which the minimizer operates in the present work has a very large dimension: for the 337×103 grid using the longest event horizon ($t_{\mathrm{EH}} = 46.08$), there are 263 points in the spatial dimension times 7680 points in the temporal dimension, for a total of 2,019,840 control variables in each event horizon. The calculations in the present work were carried out using either single 2.4-GHz 64-bit Opteron processors or 32-bit Intel P4/Xeon processors; simulations requiring more than 2 GB of memory were run on the Opteron processors. The greatest CPU time required for all the simulations in Fig. 6 was 76.2 h of CPU time. For the $\bar{k}$ objective function, the CPU time required was 46.5, 20.9, 76.2, and 62.6 h for event-horizon lengths of 0.72, 2.88, 11.52, and 46.08, respectively. The memory requirements (due to the necessity of storing the velocity and temperature for use in solving the adjoint equations) scale linearly with the length of the event horizon. The largest event horizon used in the present work required 6.3 gigabytes of memory. All simulations were carried out in double-precision.

IX. Conclusions

The present work demonstrates the feasibility of numerical control of the unsteady Navier–Stokes and temperature equations for a simplified film-cooling problem using an optimization algorithm. The optimization algorithm is gradient-based, with the gradient obtained by solving a set of adjoint equations. The optimization was accomplished over a very-high-dimensional space, with up to 2,019,840 dimensions (i.e., control variables). The present work provides guidance concerning the numerical issues involved with applying an adjoint-based optimization scheme and information about physics of the resulting control for a film-cooling flow. The ultimate goal behind the present work is to develop a practical means to predict and control a film-cooling flow to increase film-cooling effectiveness and/or to reduce the process air required to do so.

Although all objective functions studied in the present work increased film-cooling effectiveness, best results were obtained using an objective function based on the turbulent kinetic energy. Three of the four objective functions were relatively sensitive to the length of the event horizon. However, the $\bar{k}$ objective function, which was the most effective objective function and on which attention was focused, was relatively insensitive to the length of the event horizon. Numerically, this is a great advantage, because the memory requirements scale linearly with the length of the event horizon. Although a large increase in the film-cooling effectiveness was obtained with the $\bar{k}$ objective function at all event horizons studied, the control energy required depends strongly on the length of the event horizon. For the shortest event horizon, the control remained near the inequality bounds placed on the control, whereas for the longest event horizon, the control decreased by more than two orders of magnitude. Although this is not necessarily a concern computationally, for a physical system, this may be an important advantage. For the longest event horizon, the $\bar{k}$ objective function was found to have multiple local minima in the search direction. This suggests that an improvement in the results could be obtained by adding a stochastic component to the line-search algorithm to allow escape from one local minimum to another, potentially lower, local minimum. In addition, future work could consider escape from local minima within the entire control-variable space (not just the direction defined by the line-search vector), which could potentially be accomplished by using a stochastic method to change the initial guess for the start of the gradient-based method.

Acknowledgments

This work was supported by the Center for Computation and Technology at Louisiana State University and the U.S. Air Force Office of Scientific Research under grant number FA9550-06-1-0335. The simulations were run on workstations at the Center for Computation and Technology at Louisiana State University. The author is indebted to Jason Hugenroth for suggesting helpful clarifications and for a careful proofread.

References

[1] Danaila, I., and Boersma, B. J., "Direct Numerical Simulation of Bifurcating Jets," *Physics of Fluids*, Vol. 12, May 2000, pp. 1255–1257.
doi:10.1063/1.870377

[2] Hilgers, A., "Control and Optimization of Turbulent Jet Mixing," *Annual Research Briefs: 2000*, Center for Turbulence Research, Stanford Univ., Stanford, CA, 2000, pp. 45–54.

[3] Muldoon, F., "Numerical Methods for the Unsteady Incompressible Navier–Stokes Equations and Their Application to the Direct Numerical Simulation of Turbulent Flows," Ph.D. Thesis, Louisiana State Univ., Baton Rouge, LA, 2004.

[4] Johari, H., Pacheco-Tougas, M., and Hermanson, J. C., "Penetration and Mixing of Fully Modulated Turbulent Jets in Crossflow," *AIAA Journal*, Vol. 37, No. 7, 1999, pp. 842–850.

[5] M'Closkey, R. T., King, J. M., Cortelezzi, L., and Karagozian, A. R., "The Actively Controlled Jet in Crossflow," *Journal of Fluid Mechanics*, Vol. 452, Feb. 2002, pp. 325–335.
doi:10.1017/S0022112001006589

[6] Narayanan, S., Barooah, P., and Cohen, J. M., "The Dynamics and Control of an Isolated Jet in Cross Flow," *AIAA Journal*, Vol. 41, No. 12, 2003, pp. 2316–2330.

[7] Reynolds, W. C., Parekh, D. E., Juvet, P. J. D., and Lee, M. J. D., "Bifurcating and Blooming Jets," *Annual Review of Fluid Mechanics*, Vol. 35, 2003, pp. 295–315.
doi:10.1146/annurev.fluid.35.101101.161128

[8] Coulthard, S. M., Volino, R. J., and Flack, K. A., "Effect of Jet Pulsing on Film Cooling, Part 1: Effectiveness and Flowfield Temperature Results," ASME Turbo Expo 2006: Power for Land, Sea and Air, Barcelona, Spain, American Society of Mechanical Engineers Paper GT2006-91273, May 2006.

[9] Coulthard, S. M., Volino, R. J., and Flack, K. A., "Effect of Jet Pulsing on Film Cooling, Part 2: Heat Transfer Results," ASME Turbo Expo 2006: Power for Land, Sea and Air, Barcelona, Spain, American Society of Mechanical Engineers Paper GT2006-91274, May 2006.

[10] Ou, S., and Rivir, R. B., "Shaped-Hole Film Cooling with Pulsed Secondary Flow," ASME Turbo Expo 2006: Power for Land, Sea and Air, Barcelona, Spain, American Society of Mechanical Engineers Paper GT2006-90272, May 2006.

[11] Muldoon, F., and Acharya, S., "Computations of Pulsed Film-Cooling," ASME Turbo Expo 2007: Power for Land, Sea and Air, Montreal, Canada, American Society of Mechanical Engineers Paper GT2006-98156, May 2007.

[12] Nikitopoulos, D., Acharya, S., Oertling, J., and Muldoon, F., "On Active Control of Film-Cooling Flows, ASME Turbo Expo 2006: Power for Land, Sea and Air. Barcelona, Spain, American Society of Mechanical Engineers Paper GT2006-90051, 2006.

[13] Collier, J., Wroblewski, D., and Bifano, T., "Development of a Rapid-Response Flow-Control System Using MEMS Microvalve Arrays," *Journal of Microelectromechanical Systems*, Vol. 13, No. 6, Dec. 2004, pp. 912–922. doi:10.1109/JMEMS.2004.838392

[14] Vandelli, N., Wroblewski, D., Velonis, M., and Bifano, T., "Development of a MEMS Microvalve Array for Fluid Flow Control," *Journal of Microelectromechanical Systems*, Vol. 7, No. 4, Dec. 1998, pp. 395–403. doi:10.1109/84.735347

[15] Jameson, A., "Aerodynamic Design via Control Theory," *Journal of Scientific Computing*, Vol. 3, No. 3, 1988, pp. 233–260. doi:10.1007/BF01061285

[16] Kuruvila, G., Ta'asan, S., and Salas, M. D., "Airfoil Optimization by the One-Shot Method," *Optimum Design Methods for Aerodynamics*, AGARD Rept. AGARD-R-803, Neuilly-sur-Seine, France, 1994.

[17] Jameson, A., "Aerodynamic Shape Optimization Using the Adjoint Method," von Kármán Inst. for Fluid Dynamics, Brussels, 2003, http://aero-comlab.stanford.edu/Papers/jameson.vki03.pdf [retrieved 5 Apr. 2008].

[18] Li, W., Huyse, L., and Padula, S., "Robust Airfoil Optimization to Achieve Drag Reduction over a Range of Mach Numbers," *Structural and Multidisciplinary Optimization*, Vol. 24, No. 1, Aug. 2002, pp. 38–50. doi:10.1007/s00158-002-0212-4

[19] Soemarwoto, B. I., and Labrujère, Th. E., "Airfoil Design and Optimization Methods: Recent Progress at NLR," *International Journal for Numerical Methods in Fluids*, Vol. 30, No. 2, 1999, pp. 217–228. doi:10.1002/(SICI)1097-0363(19990530)30:2<217::AID-FLD830>3.0.CO;2-6

[20] Giles, M. B., Duta, M. C., Muller, J.-D., and Pierce, N. A., "Algorithm Developments for Discrete Adjoint Methods," *AIAA Journal*, Vol. 41, No. 2, 2003, pp. 198–205.

[21] Giles, M. B., and Pierce, N. A., "An Introduction to the Adjoint Approach to Design," *Flow, Turbulence and Combustion*, Vol. 65, Nos. 3–4, Dec. 2000, pp. 393–415. doi:10.1023/A:1011430410075

[22] Jameson, A., Martinelli, L., and Pierce, N. A., "Optimum Aerodynamic Design Using the Navier–Stokes Equations," *Theoretical and Computational Fluid Dynamics*, Vol. 10, 1998, pp. 213–237. doi:10.1007/s001620050060

[23] Gunzburger, M. D., and Manservisi, S., "The Velocity Tracking Problem for Navier–Stokes Flows with Bounded Distributed Control," *SIAM Journal on Control and Optimization*, Vol. 37, No. 6, 1999, pp. 1913–1945. doi:10.1137/S0363012998337400

[24] Hinze, M., and Kunisch, K., "Second Order Methods for Optimal Control of Time-Dependent Fluid Flow," *SIAM Journal on Control and Optimization*, Vol. 40, No. 3, 2001, pp. 925–946. doi:10.1137/S0363012999361810

[25] Hou, L. S., and Yan, Y., "Dynamics And Approximations of a Velocity, Tracking Problem for the Navier Stokes Flows, with Piecewise Distributed Controls," *SIAM Journal on Control and Optimization*, Vol. 35, No. 6, 1997, pp. 1847–1885. doi:10.1137/S036301299529286X

[26] Min, C., and Choi, H., "Suboptimal Feedback Control of Vortex Shedding at Low Reynolds Numbers," *Journal of Fluid Mechanics*, Vol. 401, 1999, pp. 123–156. doi:10.1017/S002211209900659X

[27] Collis, S. S., Ghayour, K., Heinkenschloss, M., Ulbrich, M., and Ulbrich, S., "Optimal Control of Unsteady Compressible Viscous Flows," *International Journal for Numerical Methods in Fluids*, Vol. 40, No. 11, 2002, pp. 1401–1429. doi:10.1002/fld.420

[28] Wei, M., and Freund, J. B., "A Noise-Controlled Free Shear Flow," *Journal of Fluid Mechanics*, Vol. 546, 2006, pp. 123–152. doi:10.1017/S0022112005007093

[29] Yang, Z., "The Adjoint Method for the Inverse Design of Solidification Processes with Convection," Ph.D. Thesis, Cornell Univ., Ithaca, NY, 1997.

[30] Zabaras, N., and Yang, G. Z., "A Functional Optimization Formulation and Implementation of an Inverse Natural Convection Problem," *Computer Methods in Applied Mechanics and Engineering*, Vol. 144, Nos. 3–4, May 1997, pp. 245–274. doi:10.1016/S0045-7825(96)01184-X

[31] Gunzburger, M. D., and Lee, H. C., "Analysis, Approximation, and Computation of a Coupled Solid/Fluid Temperature Control Problem," *Computer Methods in Applied Mechanics and Engineering*, Vol. 118, No. 1, 1994, pp. 133–152. doi:10.1016/0045-7825(94)00022-0

[32] Sampath, R., and Zabaras, N., "A Functional Optimization Approach to an Inverse Magnetoconvection Problem," *Computer Methods in Applied Mechanics and Engineering*, Vol. 190, Nos. 15–17, Jan. 2001, pp. 2063–2097. doi:10.1016/S0045-7825(00)00222-X

[33] Bewley, T. R.,Moin, P., and Temam, R., "DNS-Based Predictive Control of Turbulence: An Optimal Benchmark for Feedback Algorithms," *Journal of Fluid Mechanics*, Vol. 447, 2001, pp. 179–225.

[34] Chevalier, M., "Adjoint Based Control and Optimization of Aerodynamic Flows," Royal Inst. of Technology, Stockholm, Sweden, 2002, http://www.diva-portal.org/diva/getDocument?urn_nbn_se_kth_diva-1435-2__fulltext.pdf.

[35] Chevalier, M., "Feedback and Adjoint Based Control of Boundary Layer Flows," Ph.D. Thesis, Royal Inst. of Technology, Stockholm, Sweden, 2004.

[36] McNamara, A., Treuille, A., Popovic, Z., and Stam, J., "Fluid Control Using the Adjoint Method," *ACM Transactions on Graphics (ACM SIGGRAPH 2004)*, http://grail.cs.washington.edu/projects/control/fluidAdjoint.pdf, 2004.

[37] Treuille, A., McNamara, A., Popovic, Z., and Stam, J., "Keyframe Control of Smoke Simulations,"*ACM Transactions on Graphics*, Vol. 22, No. 3, 2003, pp. 716–723. doi:10.1145/882262.882337

[38] Demmel, J. W., Eisenstat, S. C., Gilbert, J. R., Li, X. S., and Liu, J. W. H., "A Supernodal Approach to Sparse Partial Pivoting," *SIAM Journal on Matrix Analysis and Applications*, Vol. 20, No. 3, 1999, pp. 720–755. doi:10.1137/S0895479895291765

[39] Nadarajah, S., and Jameson, A., "A Comparison of the Continuous and Discrete Adjoint Approach to Automatic Aerodynamic Optimization," AIAA 38th Aerospace Sciences Meeting and Exhibit, AIAA Paper 2000-0667, 2000.

[40] Nadarajah, S., and Jameson, A., "Studies of the Continuous and Discrete Adjoint Approaches to Viscous Automatic Aerodynamic Shape Optimization," 15th AIAA Computational Fluid Dynamics Conference, Anaheim, CA, AIAA Paper 2001-2530, 2001.

[41] Masserey, A., and Poirier, J.-R., "Optimal Control of an Induction Heating Process Using Automatic Differentiation," *International Journal for Numerical Methods in Engineering*, Vol. 62, No. 12, 2005, pp. 1721–1736. doi:10.1002/nme.1251

[42] Müller, J.-D., and Cusdin, P., "On the Performance of Discrete Adjoint CFD Codes Using Automatic Differentiation," *International Journal for Numerical Methods in Fluids*, Vol. 47, Nos. 8–9, 2005, pp. 939–945. doi:10.1002/fld.885

[43] Nocedal, J., and Wright, S. J., *Numerical Optimization*, Springer–Verlag, New York, 1999.

Z. Wang
Associate Editor