

A General Framework for the Optimal Control of Unsteady Flows with Applications

Markus P. Rumpfkeil* and David W. Zingg †

*University of Toronto Institute for Aerospace Studies
4925 Dufferin Street, Toronto, ON, M3H 5T6, Canada*

This paper presents a general framework to derive a discrete adjoint method for the optimal control of unsteady flows. First, we present the complete formulation of the time-dependent optimal design problem and outline how to derive the discrete set of adjoint equations in a general approach. After that we present results that demonstrate the application of the theory to one- and two-dimensional inverse pulseshape designs, the data assimilation problem in a shock-tube, the drag minimization of viscous flow around a rotating cylinder, and the remote inverse design of a turbulent flow around a NACA0012 airfoil at a high angle of attack.

Nomenclature

J	Objective function	Q	Flow variables
Y	Design variables	R	Flow residual
$\frac{dJ}{dY}$	Gradient of objective function	Q^*	Target flow variables
R^*	Unsteady flow residual	$(\nabla_Q R^*)^T$	Transpose of the unsteady flow Jacobian
\mathcal{L}	Lagrangian	ψ	Adjoint variables
Δt	Time discretization step	T	Final time
\bar{C}_D	Mean drag coefficient	Re	Reynolds number
u_∞	Free stream velocity	S_n	Strouhal number
\bar{S}	Vector of spline control points	λ	Steady state adjoint variable
p	Pressure	ρ	Density

I. Introduction and Motivation

The majority of work in aerodynamic shape optimization in the past has focused on the design of aerospace vehicles in a steady flow environment.^{1,2} Researchers have applied these advanced design algorithms, particularly the adjoint method, to numerous problems, ranging from the design of two-dimensional airfoils to full aircraft configurations to decrease drag, increase lift, and so on. These problems have been tackled using many different numerical schemes on both structured and unstructured grids. Unlike fixed-wing aircraft, helicopter rotors and turbomachinery blades operate in unsteady flows. Therefore, optimal control techniques for unsteady flows are needed to improve the performance of helicopter rotors and turbomachinery.

Similarly, understanding and reducing airframe-generated noise has not received much attention either, but with the significant quieting of modern engines, airframe noise now competes with engine noise.³ Thus airframe-generated noise is an important component of the total noise radiated from commercial aircraft, especially during aircraft approach and landing, when engines operate at reduced thrust, and airframe components (such as high-lift devices) are in the deployed state.⁴ Future Federal Aviation Administration

*Ph.D Candidate, Student Member AIAA, markus@oddjob.utias.utoronto.ca

†Professor, Canada Research Chair in Computational Aerodynamics, Associate Fellow AIAA, <http://goldfinger.utias.utoronto.ca/~dwz/>

noise regulations, the projected growth in air travel and the increase in population density near airports will require future civil aircraft to be substantially quieter than the current ones. Consequently, the study of airframe noise has become an important research topic.

In this paper, we develop a framework to calculate the gradient in a nonlinear unsteady flow environment via the discrete adjoint method. The optimal control of time-dependent problems is generally a computationally expensive task since one needs to solve the adjoint equations in reverse time from a final solution. Thus one has to store the entire flow history (potentially huge memory requirements) and then to integrate the adjoint equations backwards in time (equally huge processor requirements).

The goal of this paper is to present the general framework and to apply it to a variety of model problems to show its usefulness. Eventually, we want to be able to modify the shape of an airfoil to minimize the radiated noise while maintaining good performance. Once this is achieved, similar techniques can also be used to optimize blade shapes for helicopter rotors to minimize the average drag over a complete revolution, to optimize turbomachinery blades, and for many other inherently unsteady optimization problems.^{1,2,5}

II. Formulation of the Discrete Time-dependent Optimal Design Problem

In the following we assume that we want to control the unsteady flow in the time interval $[0, T]$ and that we start with an initial flow solution Q^0 . Furthermore, we use the implicit Euler time marching method to discretize the governing equations in time since it is less cumbersome to write down all the equations. This is not a restriction, since it is straightforward to modify the equations to use any other time marching method (e.g. for the second-order backwards difference method (BDF2) see Appendix A).

We introduce a cost function

$$J = \sum_{n=1}^N I^n(Q^n, Y) \quad (1)$$

where the function $I^n = I^n(Q^n, Y)$ depends on the time-dependent flow solution Q^n and design variables Y for $n = 1, \dots, N$. N is implicitly defined via $T = N\Delta t$, where Δt is the chosen time discretization step. We then assume that

$$R^{*n}(Q^n, Q^{n-1}, Y) := \frac{dQ^n}{dt} + R(Q^n, Y) = \frac{Q^n - Q^{n-1}}{\Delta t} + R(Q^n, Y) = 0 \quad (2)$$

defines implicitly the time-dependent flow solution Q^n for $n = 1, \dots, N$, where $R = R(Q^n, Y)$ represents a residue containing the convective and dissipative fluxes. This presented form naturally suggests the use of a dual time stepping scheme of the form

$$\frac{dQ^n}{d\tau} + R^{*n} = 0 \quad (3)$$

to drive the unsteady flow residual $R^{*n} = R^{*n}(Q^n, Q^{n-1}, Y)$ to zero using a time-marching method in the pseudo time τ . In our research we use the implicit Euler method to discretize the pseudo time derivative in Eq. (3).⁶ It does not matter how one solves Eq. (2) as long as $R^{*n} = 0$ for all n , since this is what we assume in the following.

The task of minimizing the cost function J can now be written as an unconstrained optimization problem of minimizing the Lagrangian function

$$\mathcal{L} = \sum_{n=1}^N [I^n(Q^n, Y) + (\psi^n)^T R^{*n}(Q^n, Q^{n-1}, Y)] \quad (4)$$

with respect to Q^0, \dots, Q^N and ψ^1, \dots, ψ^N , where ψ^1, \dots, ψ^N are the N vectors of the Lagrange multipliers. A necessary condition for an extremal is that the gradient of \mathcal{L} with respect to Q^0, \dots, Q^N and ψ^1, \dots, ψ^N should vanish. Since we start with Q^0 and calculate the states Q^1, \dots, Q^N using the constraints given by Eq. (2), we ensure that $\nabla_{\psi^n} \mathcal{L} = 0$ for $n = 1, \dots, N$ automatically.

The Lagrange multipliers ψ^n must now be chosen such that $\nabla_{Q^n} \mathcal{L} = 0$ for $n = 1, \dots, N$, which leads to

$$0 = \nabla_{Q^n} I^n + (\psi^n)^T \nabla_{Q^n} R^{*n} + (\psi^{n+1})^T \nabla_{Q^n} R^{*(n+1)} \quad \text{for } n = 1, \dots, N-1 \quad (5)$$

$$0 = \nabla_{Q^N} I^N + (\psi^N)^T \nabla_{Q^N} R^{*N} \quad (6)$$

This can be written equivalently as

$$\psi^N = -((\nabla_{Q^N} R^{*N})^T)^{-1} (\nabla_{Q^N} I^N)^T \quad (7)$$

$$\psi^n = -((\nabla_{Q^n} R^{*n})^T)^{-1} [(\nabla_{Q^n} I^n)^T + (\nabla_{Q^n} R^{*(n+1)})^T \psi^{n+1}] \quad \text{for } n = N-1, \dots, 1 \quad (8)$$

One can note that, since Q^1, \dots, Q^N have been calculated from the current guess of Y , that the vectors of Lagrange multipliers ψ^n can be calculated recursively backwards from the terminal boundary condition (7) using (8). The system of equations (7) and (8) is known as the system of adjoint equations for the model (2), or as the adjoint model. In this context, the Lagrange multipliers are also known as the adjoint variables.

Finally, one can evaluate the gradient of J with respect to the design variables Y , which can then be used in a gradient-based optimization algorithm such as BFGS⁷⁻¹⁰ to find the optimum:

$$\frac{dJ}{dY} = \frac{\partial \mathcal{L}}{\partial Y} = \sum_{n=1}^N [\nabla_Y I^n(Q^n, Y) + (\psi^n)^T \nabla_Y R(Q^n, Y)] \quad (9)$$

In summary, the gradient is determined by the solution of the adjoint equations in reverse time from the terminal boundary condition and the partial derivatives of the flow residual and objective function with respect to the design variables (while Q^n is held constant). One can also see that the computational costs of unsteady optimization problems are directly proportional to the desired number of time steps and (almost) independent of the number of design variables.

III. Results

We now present a number of simple examples to demonstrate how this framework can be applied in practice. In our 2D examples, we use the preconditioned Bi-CGSTAB algorithm¹¹ in order to invert $(\nabla_{Q^n} R^{*n})^T$ in the adjoint equations which we converge to an absolute tolerance of 10^{-12} . We find that Bi-CGSTAB is about fifty percent faster than the preconditioned generalized minimum residual (GMRES) method,¹² which we use for our steady flow solvers in conjunction with an inexact Newton strategy.¹³ The reason for this is most likely the fact that $(\nabla_{Q^n} R^{*n})^T$ is more diagonally dominant than the steady flow Jacobian $(\nabla_Q R)^T$ due to the extra terms on the diagonal, which makes this matrix more suited for the use of Bi-CGSTAB.

A. The Inverse Design of a Pulse in a Converging-Diverging Nozzle in 1D

As our first example, we consider the inverse shape design of a subsonic flow with a pulse in the static outflow pressure in a converging-diverging nozzle governed by the quasi-1D Euler equations. We use cubic spline interpolation with six control points to represent the nozzle shape, and, since we fix the in- and outflow control points to ensure a well-posed design problem, we have four shape design variables \bar{S} . The initial and target shapes of the nozzle together with the location of the control points are shown in Figure 1.

The pulse in the static outflow pressure is given by

$$p(t) = p_s + A \cdot \sin(2\pi Ft) \quad (10)$$

where A and F are the given amplitude and frequency of the pulse, and p_s is a constant. We also enforce constant stagnation conditions, p_0 and T_0 , at the inlet and the remaining three boundary conditions are calculated through linear extrapolation as follows: At the inlet we use the Riemann invariant $R_1 = u - \frac{2a}{\gamma-1}$ and at the outlet $R_2 = u + \frac{2a}{\gamma-1}$ and $H = E + p/\rho$. Our flow solver is a one-dimensional implementation of ARC2D, a finite-difference code with blended first and third order artificial dissipation fluxes.¹⁴ We use the BDF2 time-marching method for a

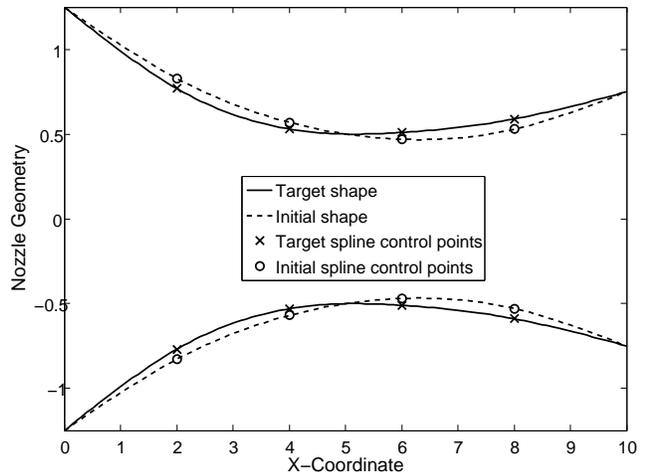


Figure 1: Initial and target nozzle shapes in 1D.

time-accurate flow solve and 100 nodes in space for all the presented cases. At $t = 0$ we initialize our unsteady flow solve with the steady state solution Q^0 of the quasi-1D Euler equations with $p(t = 0) = p_s$ (see Figure 2).

We choose $Y = (A, F, \bar{S})$ to be our control variables, and we consider two possible forms for the cost functional J :

- The observation is only obtained for the final time T

$$J_1 = \frac{1}{2} \sum_{j=1}^{100} (Q_j^N - Q_j^{*N})^2 \quad (11)$$

- The observation is distributed at assimilation times $0 \leq t \leq T$

$$J_2 = \frac{1}{2} \Delta t \sum_{n=1}^N \sum_{j=1}^{100} (Q_j^n - Q_j^{*n})^2 \quad (12)$$

Here, Q_j^{*n} are the target or desired observations at node j , which were obtained as solutions of the flow problem with the control vector $Y^* = (0.05, 1.5, \bar{S}^*)$, where \bar{S}^* are the four target shape design variables. We consider two different initial guesses $Y_1 = (0.04, 1.4, \bar{S})$ and $Y_2 = (0.08, 1.9, \bar{S})$ for this optimal control problem and we use $T = 1.0$ (in non-dimensional units) as the time horizon.

Matlab's command "fminunc" for unconstrained nonlinear optimization is used to solve this inverse design problem. We set the LargeScale option to "off" so that Matlab uses the BFGS Quasi-Newton method with a mixed quadratic and cubic line search procedure. We use the adjoint equations given in Appendix A to calculate the gradient with one slight modification: We have to account for the steady flow solution after we modify the shape of the nozzle due to new values of the design variables Y by adding $\lambda^T R(Q^0, Y)$ to the Lagrangian. This leads to one extra adjoint equation for λ that we have to solve for:

$$\lambda = - ((\nabla_{Q^0} R)^T)^{-1} [(\nabla_{Q^0} R^{*2})^T \psi^2 + \nabla_{Q^0} R^{*1})^T \psi^1] \quad (13)$$

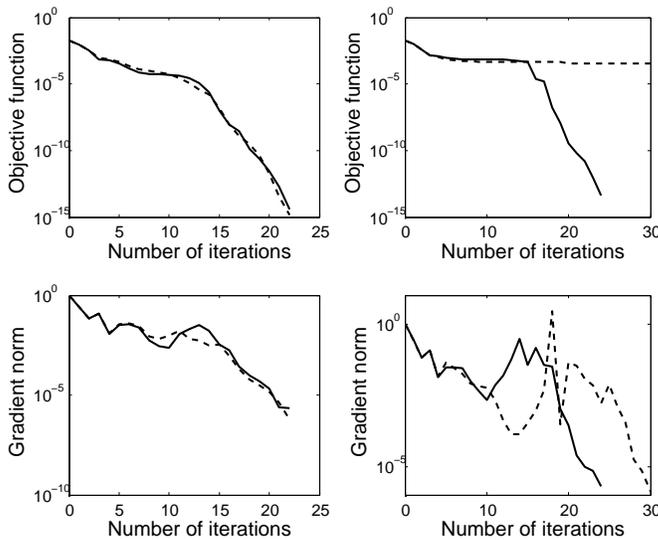


Figure 3: Convergence plots for J_1 (dashed) and J_2 (solid) using $T = 1.0$ for the first set of initial conditions in the left column and for the second set in the right column.

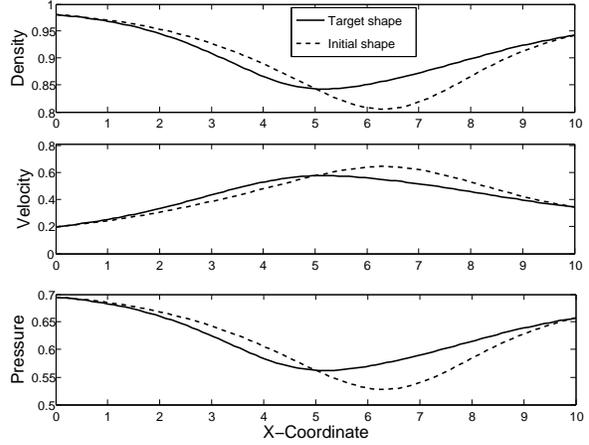


Figure 2: Initial and target pressure, velocity and density at $t = 0$.

Once we provide Matlab with the necessary routines to compute the cost function and gradient, it is able to drive both objective functions J_1 and J_2 , given by Eqs. (11) and (12), respectively, for the initial guess Y_1 , to machine zero. The same is true for J_2 with the initial guess Y_2 , but for J_1 (comparison only at the final time) and Y_2 , BFGS gets stuck in a different minimum (see Figure 3).

We also compare the resulting adjoint (*ad*) gradient of the cost functionals with respect to the design variables at the first iteration to the gradient computed via the complex-step (*cs*) method.¹⁵ The agreement is excellent with $\left(\left(\frac{dJ}{dY_k} \right)_{ad} - \left(\frac{dJ}{dY_k} \right)_{cs} \right) / \left(\frac{dJ}{dY_k} \right)_{cs} \leq 10^{-11}$ for all k and both J .

B. The Inverse Design of Flow in a Shock-tube

In this section we look at the inverse design of a flow in a 1D shock-tube. The presented framework is useful in data assimilation problems: Given a set of measurements of some actual flow on $[0, T]$, determine a “best” estimate for the initial state that leads to the observed flow behaviour. This problem has been explored before by Homescu and Navon,¹⁶ but unlike them we will use an analytic derivation of the adjoint model whereas they linearized the nonlinear forward model code line by line and viewed the resulting tangent linear model as the result of the multiplication of a number of operator matrices $O_1 O_2 \dots O_M$. They then derived the adjoint model as the product of adjoint subproblems $O_M^T O_{M-1}^T \dots O_1^T$.

The shock-tube problem can be described as follows: A tube that is filled with gas is initially divided by a membrane into two sections. The gas has a higher density and pressure in one half of the tube than in the other half, with zero velocity everywhere. At time $t = 0$, the membrane is suddenly removed and the gas is allowed to flow. We expect a net motion in the direction of lower pressure. Assuming uniform flow across the tube, there is variation in only one direction and the 1-D Euler equations apply.

We choose $Y = (p_L, p_R, \rho_L, \rho_R)$ to be the control variables, and we consider again the two forms of the cost functional J given by Eqs. (11) and (12), respectively. This time the target or desired observations Q_j^{*n} are obtained as solutions of the shock-tube problem for two different sets of initial conditions:

$$Y_1^* = (p_L = 1.1, p_R = 0.2, \rho_L = 1.1, \rho_R = 0.2)$$

$$Y_2^* = (p_L = 1.5, p_R = 0.6, \rho_L = 1.6, \rho_R = 0.4)$$

We use the Sod shock-tube values¹⁷

$$Y = (p_L = 1.0, p_R = 0.1, \rho_L = 1.0, \rho_R = 0.125)$$

as an initial guess, and our time horizon for the optimal control problem was $T = 0.21$ (in non-dimensional units). In Figure 4 we show different flow variables for the target observations at $t = 0.21$ obtained from the two different sets of initial conditions together with the observation of the initial guess at the same output time (using again 100 spatial nodes and the BDF2 time marching method).

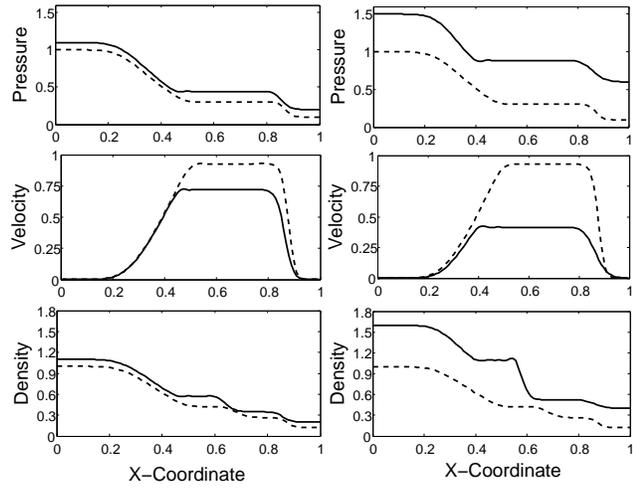


Figure 4: Pressure, velocity and density: Initial guess (dashed line) and target observation (solid line) at $t = T = 0.21$ using Y_1^* in the left column and Y_2^* in the right column.

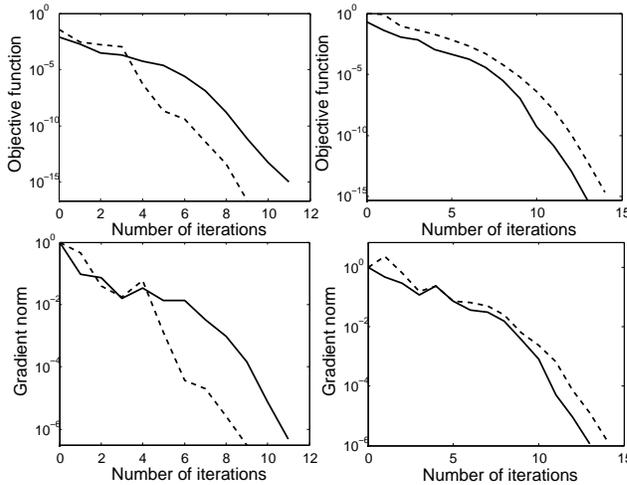


Figure 5: Convergence Plots for J_1 (dashed) and J_2 (solid) using $T = 0.21$ for the first set of initial conditions in the left column and for the second set in the right column.

thus showing that the presented framework has no problems

dealing with discontinuities such as shocks. In order to provide Matlab’s BFGS algorithm with the necessary gradient we use the adjoint equations given in Appendix A, but since this is a data assimilation problem it is advantageous to write the gradient of J with respect to the design variables Y as follows:

$$\begin{aligned} \frac{dJ}{dY} &= \frac{\partial \mathcal{L}}{\partial Y} = \frac{\partial \mathcal{L}}{\partial Q^0} \frac{\partial Q^0}{\partial Y} \\ &= [(\psi^2)^T \nabla_{Q^0} R^{*2} + (\psi^1)^T \nabla_{Q^0} R^{*1}] \frac{\partial Q^0}{\partial Y} \end{aligned}$$

Matlab is able to drive both objective functions J_1 and J_2 , given by Eqs. (11) and (12), respectively, for the two target observations Y_1^* and Y_2^* , to machine zero (see Figure 5).

We also compared the resulting adjoint gradient of the cost functionals with respect to the design variables at the first design iteration to the ones computed via the complex-step method.¹⁵ The agreement is again excellent,

C. The Inverse Design of a Pulse in a Converging-Diverging Nozzle in 2D

We use our 2D multi-block structured solver, Tornado,¹⁸ which is based on the well known ARC2D flow solver,¹⁴ in Euler mode to solve the full 2D version of the pulse problem presented in Section A. The pulse in the static outflow pressure is again given by equation (10), and we specify constant stagnation conditions at the inlet. We use cubic spline interpolation, but this time with only five control points to represent the nozzle shape, and since we fix the in- and outflow control points to ensure a well-posed design problem, we have three shape design variables \bar{S} . The x -locations of the control points are 2.5, 5.0 and 7.5, respectively, and the initial and target shapes of the nozzle are shown in Figure 6.

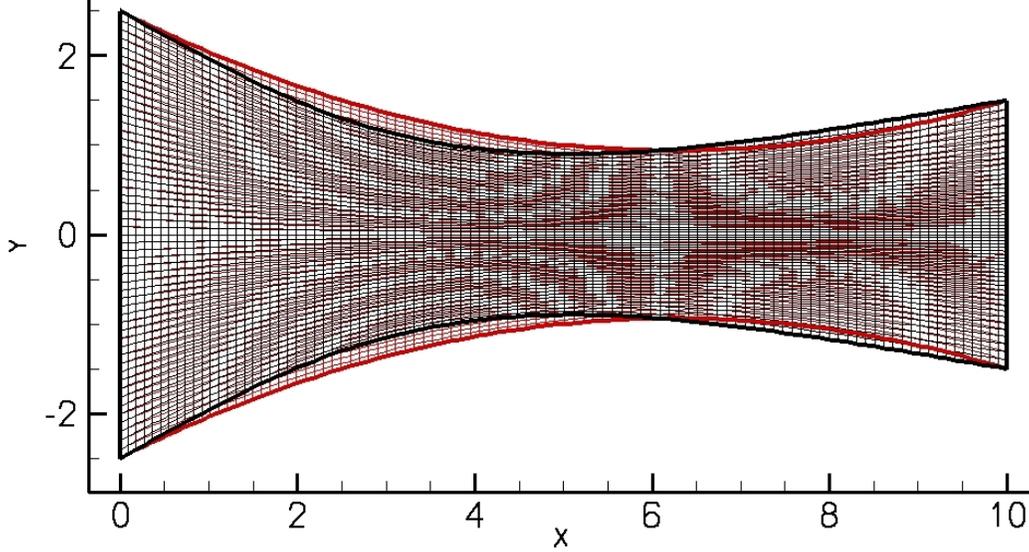


Figure 6: Initial (red) and target (black) converging-diverging nozzle shapes in 2D.

The control vector is again given by $Y = (A, F, \bar{S})$ and our objective function is as follows:

$$J = \frac{1}{2} \Delta t \sum_{n=1}^N \sum_{i=1}^{Ie} \sum_{j=1}^{Je} \sum_{k=1}^4 (\tilde{Q}_{i,j,k}^n - \tilde{Q}_{i,j,k}^{*n})^2$$

where $\tilde{Q}_{i,j,k}^n$ are the conservative flow variables at node (i, j) in the computational domain (the map of the curvilinear grid into a uniform and equally spaced grid). We choose $N = 200$, $Ie = 99$ and $Je = 55$ with a constant time step $\Delta t = 0.1$, which yields $T = 20.0$ as our final time. We also scale the objective function J by a factor such that the initial value is unity. The starting point Q^0 is the steady-state solution of the 2D Euler equations with $p(t=0) = p_s = 92 \text{ kPa}$, and we show the nondimensionalized target pressure at $t = 0$ in Figure 7.

The target flow variables $\tilde{Q}_{i,j,k}^{*n}$ are obtained as the solution of the flow problem with the control vector $Y^* = (1000, 0.15, \bar{S}^*)$. The gradient of J with respect to the control variables Y is calculated using the adjoint equations given in Appendix A. Again we have to account for the steady flow solution after modifying the shape of the nozzle by solving one additional adjoint equation for λ given by Eq. (13).

In order to save computational time and storage, we only save the flow field every five time steps (and thus compare

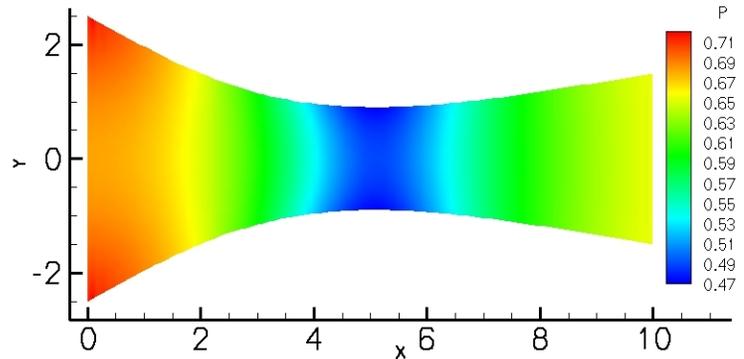


Figure 7: Nondimensionalized target pressure at $t = 0$ in 2D Nozzle.

the flowfield with the target flowfield only every five time steps). This means, in particular, that we have to invert the transpose of the Jacobian only 40 times as opposed to 200 times in order to calculate the complete gradient. This of course has an influence on the accuracy of the gradient, as we will demonstrate with the following example. If we use $Y = (2500, 0.21, \bar{S})$ as an initial guess for the control vector and we use second-order central finite-differences (*fd*) with a stepsize of $h = 10^{-7}$ to calculate the gradient at the first iteration (comparing the flowfield at every time step) we get

$$\left(\frac{dJ}{dY}\right)_{fd} = (-0.0519, 1.9142, -3.7971, 4.0757, 5.5990)$$

$$\left(\frac{dJ}{dY}\right)_{ad} = (-0.0519, 1.9142, -3.7661, 4.0911, 5.6017)$$

whereas the adjoint method with jumping over five time steps (*ad5*) yields

$$\left(\frac{dJ}{dY}\right)_{ad5} = (0.2219, 1.0391, -4.1309, 3.7740, 5.6894)$$

Nonetheless, one can see in the convergence plot for different initial guesses for A and F in Figure 8 that the objective function can be driven to machine zero in all the presented cases (not all initial guesses converge). We have to impose constraints on the shape design variables ($\bar{S} \in [0.7, 1.7]$) to ensure that the flow through the nozzle always stays entirely subsonic in the inverse design process. We also constrain the amplitude and frequency of the pulse ($A \in [-14000, 14000]$ and $F \in [-0.5, 0.5]$) to prevent pulses with excessively large amplitudes or frequencies.

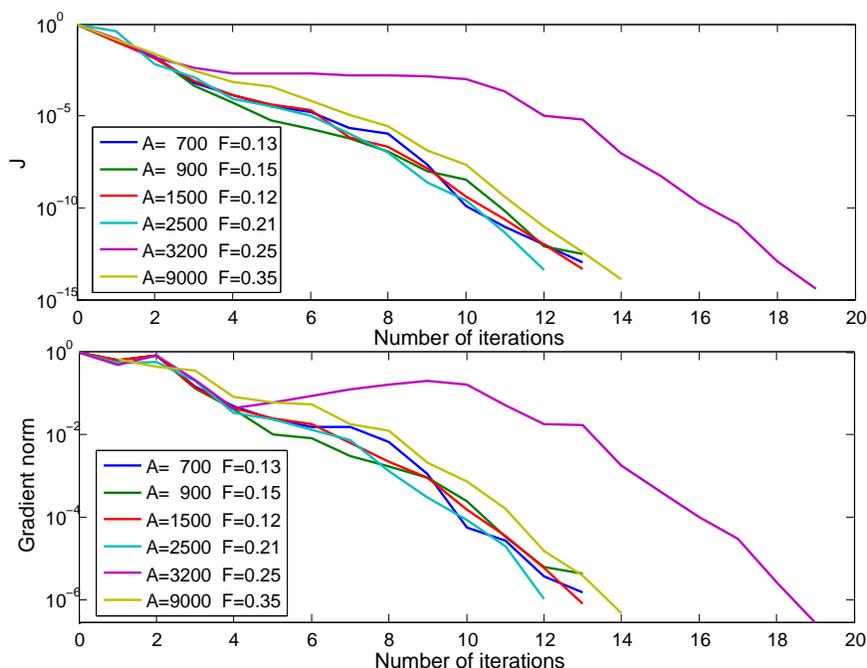


Figure 8: Convergence plots for the inverse design of a pulse in a 2D converging-diverging nozzle.

D. Drag Minimization for Viscous Flow around a Rotating Cylinder

The viscous flow past a circular cylinder has been comprehensively studied due to its simple geometry and its representative behavior of general bluff body wakes. The various flow regimes are identified by the character of the flow in the wake and boundary layer of the cylinder, which is highly dependent on the Reynolds number (Re).¹⁹ The first flow regime, $Re < 1$, is characterized by purely laminar, attached, steady, two-dimensional flow. In the second flow regime, $3 - 5 < Re < 30 - 40$, flow separates from the upper and lower

surfaces of the cylinder and two steady recirculation zones appear, which are two-dimensional, laminar and attached vortices. As the Reynolds number increases to $Re > 40$, these vortices become unstable and are shed alternately from the upper and lower surfaces of the cylinder. The flow field exhibits an unsteady train of two-dimensional laminar vortices travelling in the wake, the so-called Karman vortex shedding (which has been extensively studied for the last century, starting with the pioneering work of von Karman²⁰). The next major flow regime occurs at a Reynolds number of around 180 where the flowfield becomes three-dimensional. This flow regime also marks the beginning of the transition from laminar to turbulent flow in the wake and regular vortex shedding at a Strouhal number of about 0.2 is observed over a range of Reynolds numbers from roughly 200 to 100,000.

Using the Magnus Effect (also known as the Robbins Effect), which can be observed for rotating spheres as well as cylinders, one can try to minimize the drag and to suppress the Karman vortex shedding by controlling the angular velocity of the rotating body. The idea is that a deep understanding of the control strategies necessary to control flows past rotating bluff bodies could be applied in areas like drag reduction, lift enhancement, vibration control and last but not least, our particular interest, noise control.

In order to solve the underlying 2D unsteady Navier-Stokes equations, we use our 2D single-block structured solver, PROBE,¹³ which is a Newton-GMRES solver loosely based on ARC2D.¹⁴ We implement the rotational boundary conditions by requiring the normal velocity on the surface of the cylinder to be zero and the tangential velocity to be equal to $\Omega \cdot r$, where Ω is the angular velocity and $r = 0.5$ the radius of the cylinder. Several researchers^{21,22} have considered two control cases: The constant rotation case, $\Omega(t) = \Omega$, and the time harmonic rotary oscillation case, $\Omega(t) = A \cdot \sin(2\pi Ft)$. Thus, they use as design variables $Y = \Omega$ or $Y = (A, F)$, respectively.

Table 1: A comparison of the mean drag coefficients and Strouhal numbers.

Re	\bar{C}_D		S_n	
	100	1000	100	1000
Present work	1.45	1.53	0.179	0.252
Homescu <i>et al.</i> ²¹	1.42	1.68	-	-
He <i>et al.</i> ²²	1.35	1.52	0.167	0.239
Williamson ²³	-	-	0.164	-
Henderson ²⁴	1.35	1.51	0.166	0.237

There is a reasonable agreement for both Reynolds numbers. We could probably increase the accuracy of our flow solver by using a smaller time step, a finer grid, or a higher-order time-marching method, but since we are only using this as a model problem we are satisfied with the results. Because the primary focus of this paper is on the optimal control of unsteady flows, grid density and grid extent studies are not performed. We recognize that the 140×90 grid is relatively coarse and that the computed unsteady flow is not grid converged for this case. Also, the use of the thin-layer Navier-Stokes equations which are implemented in PROBE is questionable for a bluff-body flow and will deviate to some degree from a full Navier-Stokes solution.

The experimental work of Tokumaru and Dimotakis²⁵ motivates the attempt to find an optimal angular velocity in order to minimize the drag. We show the vortex shedding for $Re = 1000$ at the time point where we will start our rotary control in Figure 9. The effect of different values of Ω on the drag coefficient for the constant rotation case can be seen in Figure 10. The rotation starts impulsively at zero time steps, and one can see that after a transition period of about 1500 time steps the mean drag coefficient \bar{C}_D is decreased for all presented cases.

For comparison purposes, it is convenient to introduce the Strouhal number

$$S_n = d \cdot f_n / u_\infty$$

where d is the diameter of the cylinder, f_n is the Karman vortex shedding frequency, and u_∞ is the free stream velocity. Using an O-mesh with 140×90 grid nodes, and the BDF2 time-marching method with a time step of $\Delta t = 0.1$, we compare our results for the mean value of the drag coefficient \bar{C}_D and the Strouhal number S_n with experimentally and computationally obtained values by various authors²¹⁻²⁴ in Table 1.

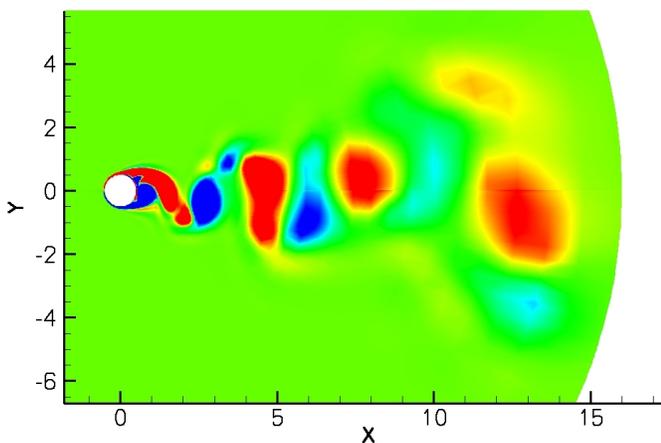


Figure 9: Vortex shedding behind a cylinder with unit diameter, $M_\infty = 0.2$ and $S_n = 0.252$ (we show the vorticity).

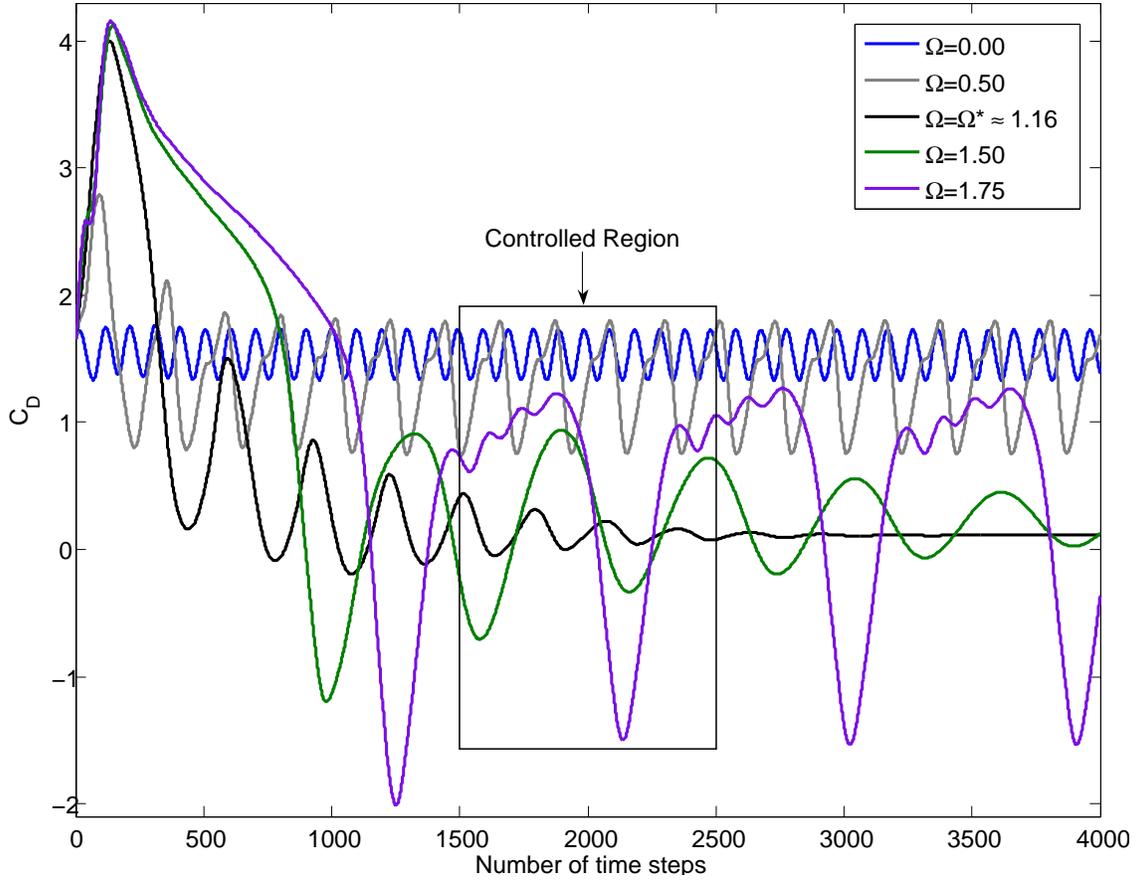


Figure 10: Drag coefficients for the constant rotation case for different values of Ω ($\Delta t = 0.1$).

We want to “jump” over the adjusting or transition period as quickly as possible, thus taking a bigger time step $\Delta T = 0.5$ for $N^* = 300$ time steps. Once we reach the domain where we actually want to control the problem, we use a smaller time step $\Delta t = 0.2$ for another 500 time steps (the control time window is indicated by the box in Figure 10), which yields a total of $N = 800$ time steps for each flow solve. The corresponding adjoint equations for the described situation are given in Appendix B.

Our choice for the objective function for the constant and harmonic rotating cases is a time average (mean) drag minimization problem:

$$J = \bar{C}_D = \frac{1}{N - N^*} \sum_{n=N^*+1}^N C_D^n \quad (14)$$

where C_D^n is the drag coefficient at time step n . Using BFGS^{26,27} and constraining Ω to values between 0 and 1.9 we are able to minimize the drag with gradient norms of 10^{-8} at the minima. We show the resulting design space in Figure 11 with the gradients at the different design points represented as straight lines. One can see that there are several local minima in this design space, with the global minimum in the given interval at $\Omega = \Omega^* \approx 1.16$ leading to $\bar{C}_D \approx 0.11$. This optimum value minimizes the mean drag value far beyond the extent of the control time window, as can be seen in Figure 10, and this behaviour was also observed by other researchers.^{21,22}

Once again we try to save computational time and storage by saving the flowfield in the control time window only every other time step leading to only $300 + 500/2 = 550$ matrix inversions for the adjoint as compared to 800 in the original case. The result is also shown in Figure 11, and the gradients and objective function values are in reasonable agreement with each other thus leading to a similar convergence history except that in this case the local minima are slightly shifted (about 0.25 percent off) and the gradient norms only reduce to 10^{-3} at these minima.

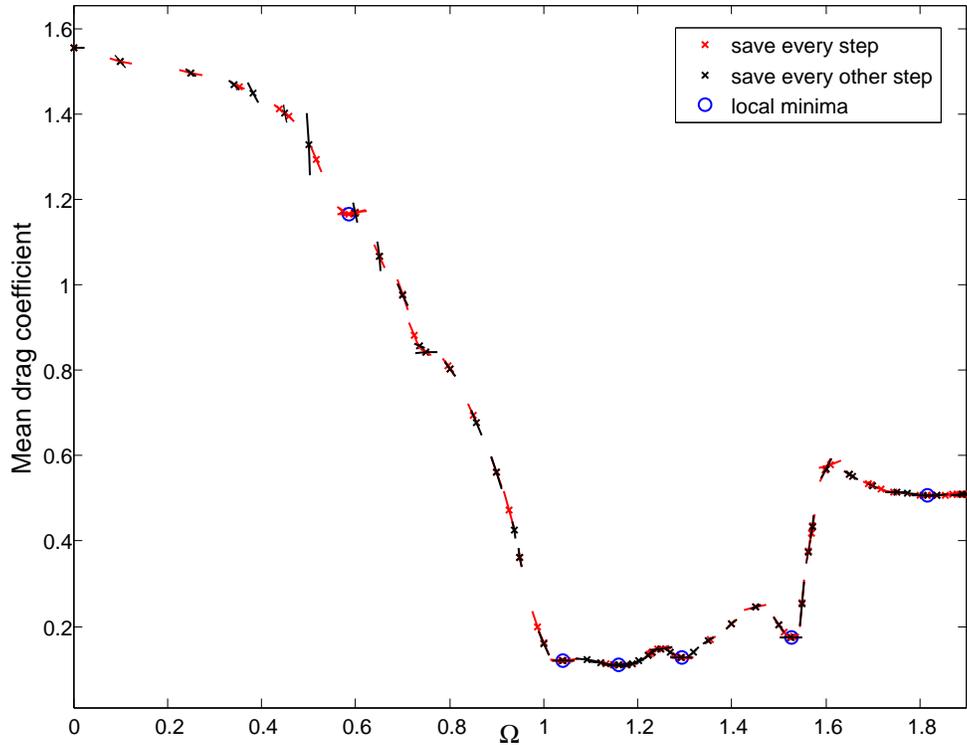


Figure 11: The design space of the constant rotating cylinder $\Omega(t) = \Omega$.

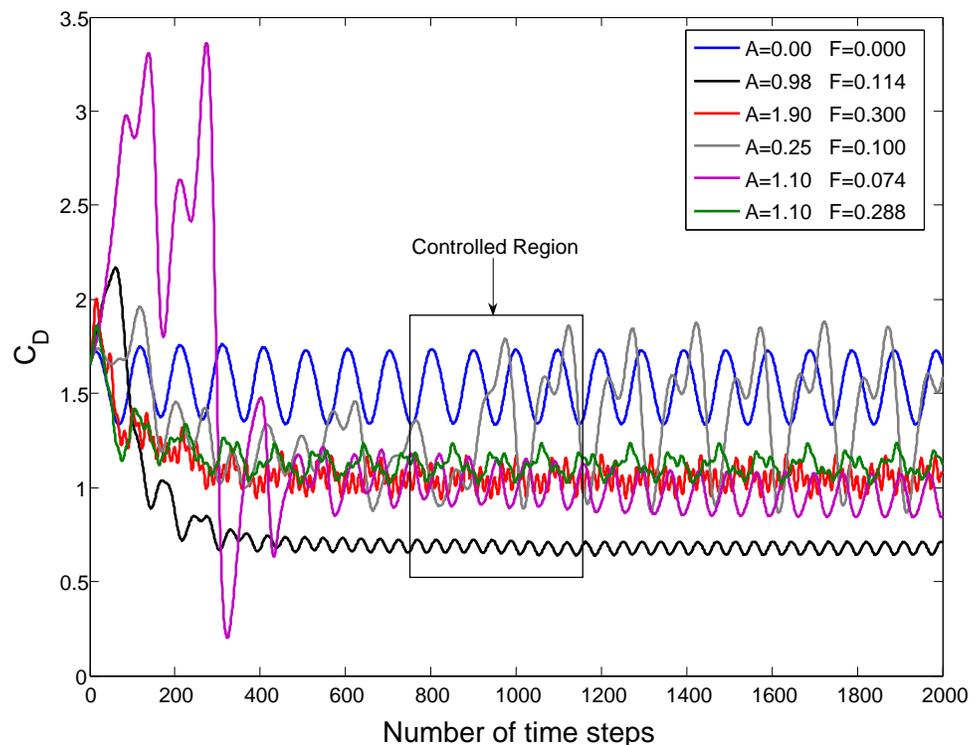


Figure 12: Drag coefficients for the harmonic rotation case for different values of A and F ($\Delta t = 0.1$).

In Figure 12 we show the effect of different values of A and F on the drag coefficient for the harmonic rotation case. The rotation starts smoothly at zero time steps, and one can see that after a transition period of about 750 time steps the mean drag coefficient \bar{C}_D is again decreased for all presented cases.

This time we “jump” over the transition period with a time step of $\Delta T = 0.2$ for $N^* = 375$ time steps and then switch to a smaller time step $\Delta t = 0.1$ in the actual control window for another 400 time steps, giving a total of $N = 775$ time steps for each flow solve. Using the same objective function as for the constant rotating cylinder given by Eq. (14) and constraining the amplitude A to $[0, 1.9]$ and the frequency F to $[0, 0.3]$, we can minimize the drag with gradient norms of about 10^{-4} at the minima. The resulting design space can be seen in Figure 13 with the gradients in the different design points represented by arrows and the objective function values given by a colour scale with red representing the highest and blue the lowest values. Once again several local minima can be seen, with a global minimum for $Y = Y^* \approx (0.98, 0.114)$ leading to $\bar{C}_D \approx 0.6832$, which again leads to a minimized mean drag value far beyond the extent of the control time window (see Figure 12).

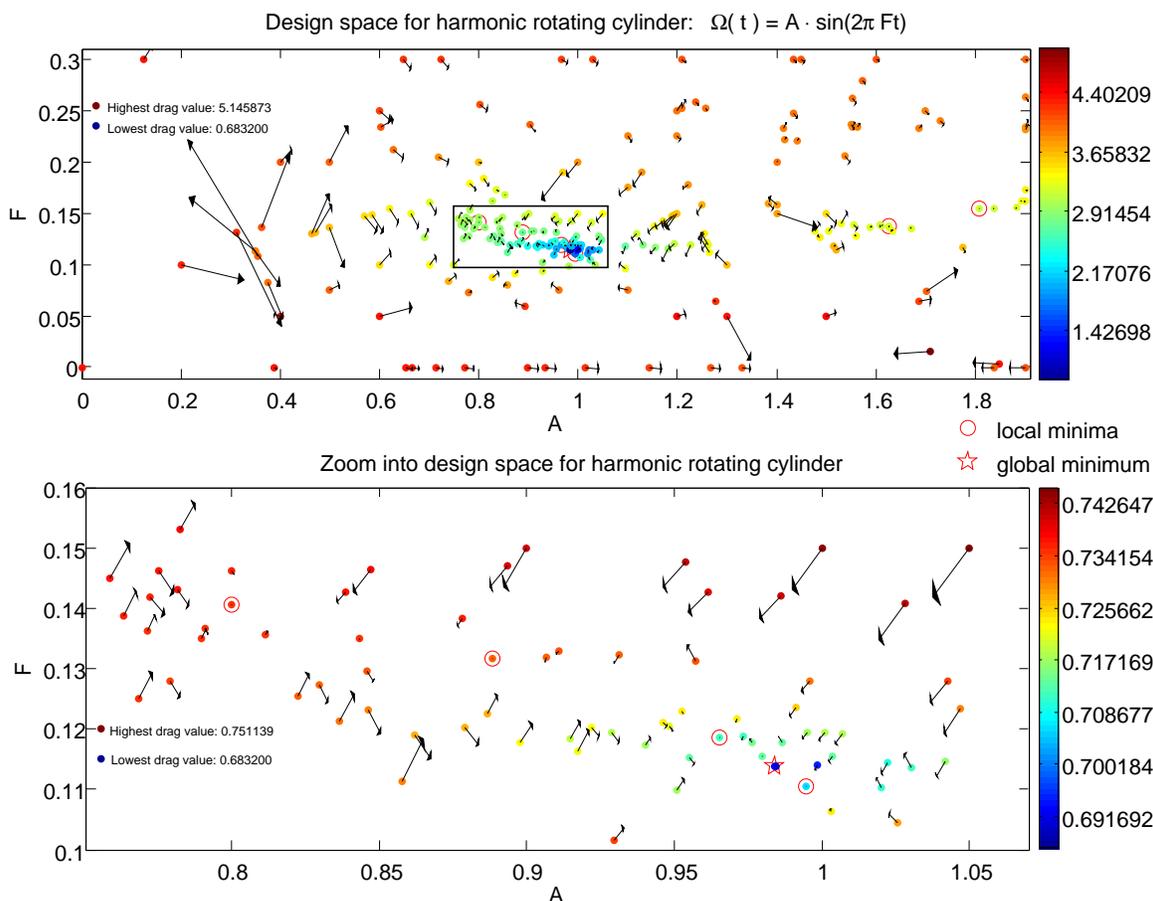


Figure 13: The design space of the harmonic rotating cylinder and a zoom into the most interesting region.

E. The Remote Inverse Shape Design of an Airfoil in Unsteady Turbulent 2D Flow

Traditional adjoint implementations are aimed at minimizing a cost function computed from flow variables on the surface, for example of an airfoil, that is being modified. However, for many problems, such as noise reduction, we want to minimize an objective function using flow quantities that are not collocated at the points where the surface is being modified. The goal is to use our set of discrete adjoint equations to quantify the influence of geometry modifications on the flow variables at an arbitrary location (e.g. a near-field plane) within the domain of interest. This type of sensitivity calculation has been successfully used before by Nadarajah *et al.*^{28,29} for the steady case of sonic boom minimization and will be necessary for a variety of problems including inlet design, turbomachinery design, and airfoil-generated noise.

A typical approach to tackle the high-lift noise reduction problem is to represent the CFD solution on a reasonable computational mesh that does not extend too far from the airfoil (see Figure 14). At a pre-specified distance below the aircraft but still within the CFD mesh, one can specify the location of a near-field plane. This plane is the effective interface between the CFD solution and a wave propagation program based on principles of geometrical acoustics and nonlinear wave propagation³⁰ which calculates the pressure fluctuations at the ground plane. This solution can then be used to determine any of a variety of measures of the airframe-generated noise, for example overpressures. This paper only focuses on controlling the near-field pressures, which are one of the inputs to a wave propagation program.

Our test case is a remote inverse shape design problem which involves turbulent flow over an airfoil. The free stream Mach number is 0.2 with a Reynolds number of 4×10^6 , and the angle of attack is 20° . At these conditions the airfoil experiences vortex shedding. We use PROBE¹³ with the one-equation Spalart-Allmaras turbulence model³¹ to solve this unsteady turbulent flow problem.

The geometry of the airfoil is described with cubic B-spline curves,¹⁸ which means that some of the y -coordinates of the B-spline control points can easily be used as shape design variables. We use four shape design variables to keep the problem simple and to be able to compare our adjoint gradient with a finite-differenced one. Our initial airfoil shape is the NACA0012 and we perturb the four shape design variables slightly to get a target airfoil shape (see Figure 15). Our discrete cost function J is given by

$$J = \frac{1}{2} \Delta t \sum_{n=N^*+1}^N \sum_{NF} (p^n - p^{*n})^2$$

where p^n is the near-field pressure obtained from the current airfoil shape, and p^{*n} is the near-field pressure obtained from the target airfoil shape (both at time step n). The sum over NF implies a sum over all the grid nodes that define the near-field plane; our particular choice is shown in Figure 16.

In Figure 17 we show the drag coefficients for the initial and target airfoil shapes over time. Both flow solves are warmstarted from a NACA0012 periodic steady state solution, thus one can see an adjustment period for the target airfoil. We want to “jump” over this unphysical adjusting period after a shape modification has taken place as quickly as possible. Therefore, we take a bigger time step $\Delta T = 0.1$ for the first $N^* = 300$ time steps, and once we reach our desired control window, we use a smaller time step $\Delta t = 0.05$ for another 200 time steps, for a total of $N = 500$ time steps for each flow solve. The corresponding adjoint equations for this situation are again given in Appendix B.

The convergence history of this remote inverse design problem with the adjoint approach in comparison to a second-order central finite-difference approach with a step size of $h = 10^{-7}$ is shown in Figure 18. The objective function J is again always scaled by a factor such that its initial value is unity.

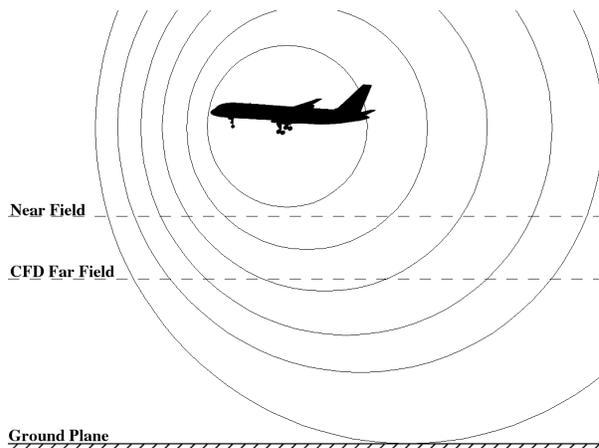


Figure 14: Schematic of the propagation of the aircraft pressure signature.

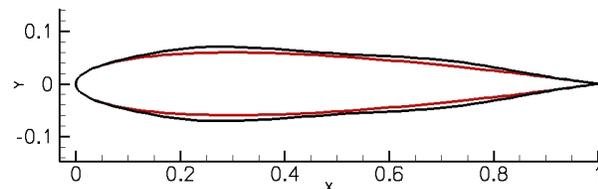


Figure 15: The initial (red) and target (black) airfoil shapes.

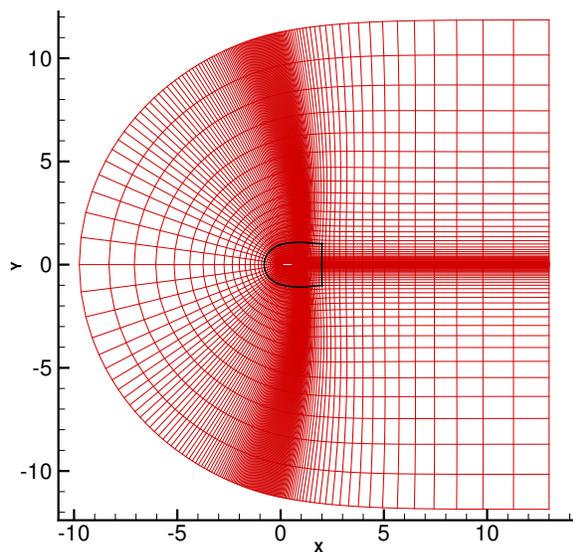


Figure 16: Our mesh where the near-field plane is shown in black.

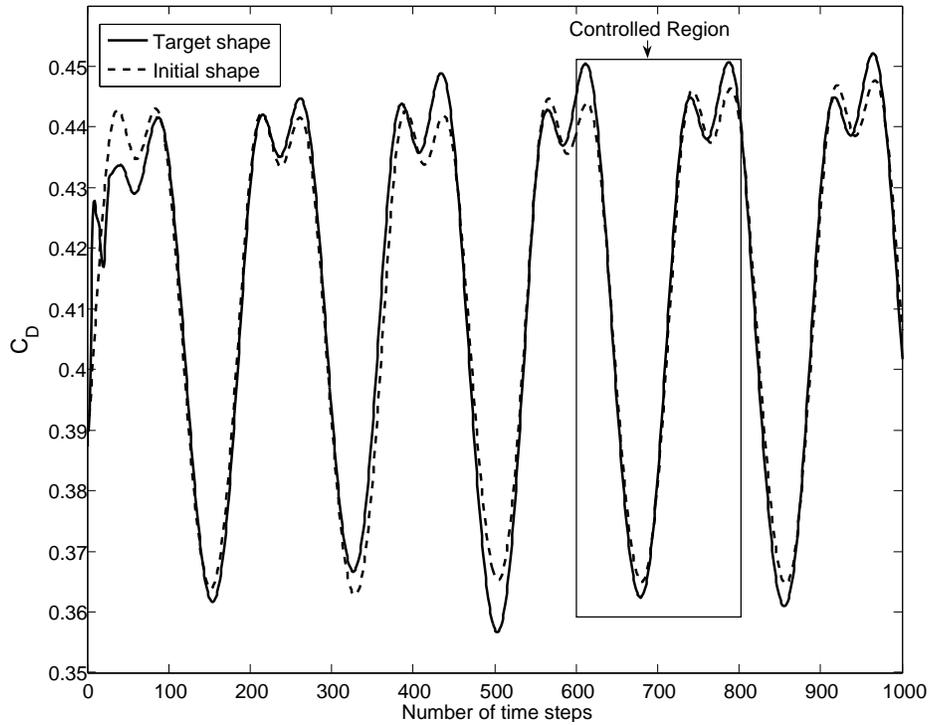


Figure 17: Drag coefficient for the initial and target airfoil shape ($\Delta t = 0.05$).

One can see that the objective function is driven to a small value in about twenty-six design iterations and that the two approaches show a reasonable agreement, which means that our adjoint approach for the gradient calculation is accurate. We also try to save computational time and storage by saving the flowfield in the adjusting period and in the control window only every fourth time step, leading to only $500/4 = 125$ matrix inversions for the solution of the adjoint equations. The result is also shown in Figure 18, and the gradients and objective function values are in reasonable agreement with the original adjoint and finite-difference approach, thus leading to a similar convergence history while saving 75 percent computational resources.

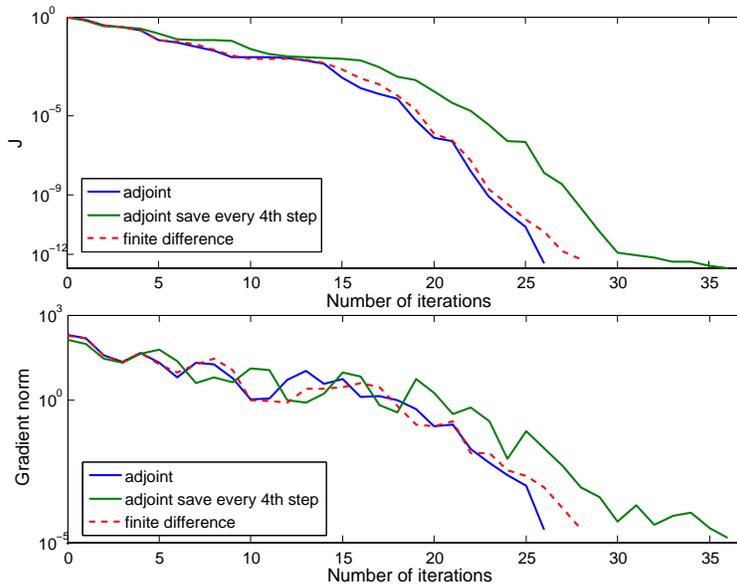


Figure 18: Convergence history of the remote inverse design problem with four design variables.

IV. Conclusion

We have presented a general framework to derive a discrete adjoint method for the optimal control of unsteady flows. We applied this framework to several applications of interest and thus demonstrated the generality of our approach. We also showed that marching with a bigger time step over transition or unphysical adjusting periods as well as skipping time steps (e.g. every other) while recording the flow solution works well in practice, thus resulting in significant savings in both memory and computational time for unsteady optimization problems. Our future work will focus on the ability to modify the shape of an airfoil to minimize the radiated noise while maintaining good performance. Therefore, we will investigate the presented remote inverse design problem further by using more complex geometries (slats and flaps) as well as more design variables.

Appendix

A. Adjoint Equations for BDF2

In this appendix, we derive the discrete adjoint equations resulting from discretizing the time derivative in Eq. (2) with the second-order implicit backward difference (BDF2) time marching method. Since this method is not self-starting, we use the implicit Euler method for the first time step. The time-dependent flow solution Q^n is then implicitly defined through the following unsteady residuals:

$$\begin{aligned} R^{*1}(Q^1, Q^0, Y) &:= \frac{Q^1 - Q^0}{\Delta t} + R(Q^1, Y) = 0 \\ R^{*n}(Q^n, Q^{n-1}, Q^{n-2}, Y) &:= \frac{3Q^n - 4Q^{n-1} + Q^{n-2}}{2\Delta t} + R(Q^n, Y) = 0 \quad \text{for } n = 2, \dots, N \end{aligned}$$

The problem of minimizing the discrete objective function given by $J = \sum_{n=1}^N I^n(Q^n, Y)$ is equivalent to the unconstrained optimization problem of extremizing the Lagrangian function

$$\mathcal{L} = \sum_{n=2}^N [I^n(Q^n, Y) + (\psi^n)^T R^{*n}(Q^n, Q^{n-1}, Q^{n-2}, Y)] + I^1(Q^1, Y) + (\psi^1)^T R^{*1}(Q^1, Q^0, Y)$$

with respect to Q^0, \dots, Q^N and ψ^1, \dots, ψ^N . The Lagrange multipliers ψ^n must now be chosen such that $\nabla_{Q^n} \mathcal{L} = 0$ for $n = 1, \dots, N$, which leads to

$$\begin{aligned} 0 &= \nabla_{Q^n} I^n + (\psi^n)^T \nabla_{Q^n} R^{*n} + (\psi^{n+1})^T \nabla_{Q^n} R^{*n+1} + (\psi^{n+2})^T \nabla_{Q^n} R^{*n+2} \quad \text{for } n = 1, \dots, N-2 \\ 0 &= \nabla_{Q^{N-1}} I^{N-1} + (\psi^N)^T \nabla_{Q^{N-1}} R^{*N} + (\psi^{N-1})^T \nabla_{Q^{N-1}} R^{*N-1} \\ 0 &= \nabla_{Q^N} I^N + (\psi^N)^T \nabla_{Q^N} R^{*N} \end{aligned}$$

This can be written equivalently as

$$\begin{aligned} \psi^N &= -((\nabla_{Q^N} R^{*N})^T)^{-1} (\nabla_{Q^N} I^N)^T \\ \psi^{N-1} &= -((\nabla_{Q^{N-1}} R^{*N-1})^T)^{-1} [(\nabla_{Q^{N-1}} R^{*N})^T \psi^N + (\nabla_{Q^{N-1}} I^{N-1})^T] \\ \psi^n &= -((\nabla_{Q^n} R^{*n})^T)^{-1} [(\nabla_{Q^n} R^{*n+2})^T \psi^{n+2} + (\nabla_{Q^n} R^{*n+1})^T \psi^{n+1} + (\nabla_{Q^n} I^n)^T] \quad \text{for } n = N-2, \dots, 1 \end{aligned}$$

Finally, the gradient of J with respect to the design variables Y is again given by

$$\frac{dJ}{dY} = \frac{\partial \mathcal{L}}{\partial Y} = \sum_{n=1}^N [\nabla_Y I^n(Q^n, Y) + (\psi^n)^T \nabla_Y R(Q^n, Y)]$$

B. Adjoint Equations for Warmstarted BDF2 with a Time Step Size Change

In this appendix, we derive the discrete adjoint equations in the form in which we use them to present our rotating cylinder and remote inverse design results. We usually warmstart our flow solve at some time point which means that we know Q^0 and Q^{-1} . We also want to “jump” over the adjusting or transition period as

quickly as possible thus taking a bigger time step ΔT for N^* time steps. Once we reach the domain where we actually want to control the problem we use a smaller time step Δt for $N - N^*$ time steps. Thus we have a total of N time steps and to keep the second-order time accuracy, the time-dependent flow solution Q^n is implicitly defined through the following unsteady residuals:

$$\begin{aligned} R^{*n}(Q^n, Q^{n-1}, Q^{n-2}, Y) &:= \frac{3Q^n - 4Q^{n-1} + Q^{n-2}}{2\Delta T} + R(Q^n, Y) = 0 \quad \text{for } n = 1, \dots, N^* \\ R^{*N^*+1}(Q^{N^*+1}, Q^{N^*}, Q^{N^*-1}, Y) &:= \frac{(2\Delta t\Delta T + \Delta T^2)Q^{N^*+1} - (\Delta t + \Delta T)^2Q^{N^*} + \Delta t^2Q^{N^*-1}}{\Delta t\Delta T(\Delta t + \Delta T)} + R(Q^{N^*+1}, Y) = 0 \\ R^{*n}(Q^n, Q^{n-1}, Q^{n-2}, Y) &:= \frac{3Q^n - 4Q^{n-1} + Q^{n-2}}{2\Delta t} + R(Q^n, Y) = 0 \quad \text{for } n = N^* + 2, \dots, N \end{aligned}$$

The problem of minimizing the discrete objective function given by $J = \sum_{n=N^*+1}^N I^n(Q^n, Y)$ is then equivalent to the unconstrained optimization problem of extremizing the Lagrangian function

$$\mathcal{L} = \sum_{n=N^*+1}^N I^n(Q^n, Y) + \sum_{n=1}^N (\psi^n)^T R^{*n}(Q^n, Q^{n-1}, Q^{n-2}, Y)$$

with respect to Q^0, \dots, Q^N and ψ^1, \dots, ψ^N . This leads to the following equations for ψ^n :

$$\begin{aligned} 0 &= (\psi^n)^T \nabla_{Q^n} R^{*n} + (\psi^{n+1})^T \nabla_{Q^n} R^{*(n+1)} + (\psi^{n+2})^T \nabla_{Q^n} R^{*(n+2)} && \text{for } n = 1, \dots, N^* \\ 0 &= \nabla_{Q^n} I^n + (\psi^n)^T \nabla_{Q^n} R^{*n} + (\psi^{n+1})^T \nabla_{Q^n} R^{*(n+1)} + (\psi^{n+2})^T \nabla_{Q^n} R^{*(n+2)} && \text{for } n = N^*+1, \dots, N-2 \\ 0 &= \nabla_{Q^{N-1}} I^{N-1} + (\psi^N)^T \nabla_{Q^{N-1}} R^{*N} + (\psi^{N-1})^T \nabla_{Q^{N-1}} R^{*(N-1)} \\ 0 &= \nabla_{Q^N} I^N + (\psi^N)^T \nabla_{Q^N} R^{*N}. \end{aligned}$$

which can be written equivalently as

$$\psi^n = \begin{cases} -((\nabla_{Q^n} R^{*n})^T)^{-1} [(\nabla_{Q^n} I^n)^T] & \text{for } n = N \\ -((\nabla_{Q^n} R^{*n})^T)^{-1} [(\nabla_{Q^n} I^n)^T + (\nabla_{Q^n} R^{*(n+1)})^T \psi^{n+1}] & \text{for } n = N-1 \\ -((\nabla_{Q^n} R^{*n})^T)^{-1} [(\nabla_{Q^n} I^n)^T + (\nabla_{Q^n} R^{*(n+1)})^T \psi^{n+1} + (\nabla_{Q^n} R^{*(n+2)})^T \psi^{n+2}] & \text{for } n = N-2, \dots, N^*+1 \\ -((\nabla_{Q^n} R^{*n})^T)^{-1} [(\nabla_{Q^n} R^{*(n+1)})^T \psi^{n+1} + (\nabla_{Q^n} R^{*(n+2)})^T \psi^{n+2}] & \text{for } n = N^*, \dots, 1 \end{cases}$$

A little care must be taken in calculating derivatives of R^{*N^*+1} with respect to Q^n since the factors in front of Q^{N^*+1} , Q^{N^*} and Q^{N^*-1} are slightly different. The gradient of J with respect to the design variables Y is then given by

$$\frac{dJ}{dY} = \frac{\partial \mathcal{L}}{\partial Y} = \sum_{n=N^*+1}^N \nabla_Y I^n(Q^n, Y) + \sum_{n=1}^N (\psi^n)^T \nabla_Y R(Q^n, Y)$$

Acknowledgments

The funding of the second author by the Natural Sciences and Engineering Research Council of Canada and the Canada Research Chairs program is gratefully acknowledged.

References

- ¹S. Nadarajah, M. McMullen, and A. Jameson, "Optimum Shape Design for Unsteady Flows using Time Accurate and Non-linear Frequency Domain Methods," AIAA, 2003-3875, 2003.
- ²S. Nadarajah and A. Jameson, "Optimal Control of Unsteady Flows using a Time Accurate Method," AIAA, 2002-5436, 2002.
- ³B.A. Singer, K.S. Brentner, and D.P. Lockard, "Computational Aeroacoustic Analysis of Slat Trailing-Edge Flow," *AIAA Journal*, Vol. 38 no.9, 2000, pp. 1558–1564.
- ⁴M.R. Khorrami, M.E. Berkman, and M. Choudhari, "Unsteady Flow Computations of a Slat with a Blunt Trailing Edge," *AIAA Journal*, Vol. 38 no.11, 2000, pp. 2050–2058.
- ⁵M.C. Duta, M.B. Giles, and M.S. Campobasso, "The Harmonic Adjoint Approach to Unsteady Turbo Machinery Design," *International Journal for Numerical Methods in Fluids*, Vol. 40, 2002, pp. 323–332.

- ⁶S. Venkateswaran and C.L. Merkle, "Dual Time Stepping and Preconditioning for Unsteady Computations," AIAA, 95-0078, 1995.
- ⁷C.G. Broyden, "The Convergence of a Class of Double-Rank Minimization Algorithms," *Journal Inst. Math. Applic.*, Vol. 6, 1970, pp. 76–90.
- ⁸R. Fletcher, "A New Approach to Variable Metric Algorithms," *Computer Journal*, Vol. 13, 1970, pp. 317–322.
- ⁹D. Goldfarb, "A Family of Variable Metric Updates Derived by Variational Means," *Mathematics of Computing*, Vol. 24, 1970, pp. 23–26.
- ¹⁰D.F. Shanno, "Conditioning of Quasi-Newton Methods for Function Minimization," *Mathematics of Computing*, Vol. 24, 1970, pp. 647–656.
- ¹¹H. van der Vorst, "Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems," *SIAM Journal on Scientific and Statistical Computing*, Vol. 13, 1992, pp. 631–644.
- ¹²Y. Saad and M.H. Schultz, "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems," *SIAM Journal on Scientific and Statistical Computing*, Vol. 7, No.3, 1986, pp. 856–869.
- ¹³A. Pueyo and D.W. Zingg, "An Efficient Newton-GMRES Solver for Aerodynamic Computations," AIAA, 97-1955, 1997.
- ¹⁴T. H. Pulliam, *Efficient Solution Methods for the Navier-Stokes Equations*, Lecture Notes for the Von Karman Institute For Fluid Dynamics Lecture Series, 1986.
- ¹⁵W.Squire and G. Trapp, "Using Complex Variables to Estimate Derivatives of Real Functions," *SIAM Review*, Vol. 40(1), 1998, pp. 110–112.
- ¹⁶C. Homescu and I.M. Navon, "Optimal Control of Flow with Discontinuities," *Journal of Computational Physics*, Vol. 187, 2003, pp. 660–682.
- ¹⁷G.A. Sod, "A Survey of Finite-difference Methods for Systems of Nonlinear Conservation Laws," *J. Comput. Phys.*, Vol. 27 (1), 1978.
- ¹⁸M. Nemeć, *Optimal Shape Design of Aerodynamic Congurations: A Newton-Krylov Approach*, Ph.D. thesis, University of Toronto, 2003.
- ¹⁹J.S. Cox, K.S. Brentner, and C.L. Rumsey, "Computation of Vortex Shedding and Radiated Sound for a Circular Cylinder: Subcritical to Transcritical Reynolds Numbers," *Theoretical and Computational Fluid Dynamics*, Vol. 12, 1998, pp. 233–253.
- ²⁰T. von Karman, "Ueber den Mechanismus des Widerstandes den ein bewegter Koerper in einer Fluessigkeit erzeugt," *Nachr. Ges. Wiss. Goettingen*, Vol. Math. Phys. Klasse, 1911, pp. 509–517.
- ²¹C. Homescu, I.M. Navon and Z. Li, "Suppression of Vortex Shedding for Flow around a Circular Cylinder Using Optimal Control," *Int. Journal for Num. Meth. Fluids*, Vol. 38, 2002, pp. 43–69.
- ²²J.-W. He, R. Glowinski, R. Metcalfe, A. Nordlander, and J. Periaux, "Active Control and Drag Optimization for Flow past a Circular Cylinder," *Journal of Computational Physics*, Vol. 163, 2000, pp. 83–117.
- ²³C.H.K. Williamson, "Oblique and Parallel Modes of Vortex Shedding in the Wake of a Circular Cylinder at Low Reynolds Numbers," *Journal of Fluid Mechanics*, Vol. 206, 1989, pp. 579.
- ²⁴R.D. Henderson, "Nonlinear Dynamics and Patterns in Turbulent Wake Transition," *Journal of Fluid Mechanics*, Vol. 352, 1997, pp. 65.
- ²⁵P.T. Tokumar and P.E. Dimotakis, "Rotary Oscillation Control of a Cylinder Wake," *Journal of Fluid Mechanics*, Vol. 224, 1991, pp. 77.
- ²⁶R. H. Byrd, P. Lu, J. Nocedal and C. Zhu, "A Limited Memory Algorithm for Bound Constrained Optimization," *SIAM J. Scientific Computing* 16, Vol. 5, 1995, pp. 1190–1208.
- ²⁷C. Zhu, R.H. Byrd, P. Lu, J. Nocedal, "L-BFGS-B: A Limited Memory FORTRAN Code for Solving Bound Constrained Optimization Problems," Tech. Rep. NAM-11, EECS Department, Northwestern University, 1994.
- ²⁸S. Nadarajah, A. Jameson, and J. Alonso, "Adjoint-based Sonic Boom Reduction for Wing-body Configurations in Supersonic Flow," *50th CASI Annual General Meeting and Conference, Montreal*, 2003.
- ²⁹S. Nadarajah, A. Jameson, and J. Alonso, "An Adjoint Method for the Calculation of Remote Sensitivities in Supersonic Flow," AIAA, 2002-0261, 2002.
- ³⁰K.S. Brentner and F. Farassat, "Analytical Comparison of the Acoustic Analogy and Kirchhoff Formulation for Moving Surfaces," *AIAA Journal*, Vol. 36 no.8, 1998, pp. 1379–1386.
- ³¹P. R. Spalart and S. R. Allmaras, "A One-Equation Turbulence Model for Aerodynamic Flows," AIAA, 92-0439, 1992.