Math. Model. Nat. Phenom.
Vol. 6, No. 3, 2011, pp.141-165
DOI: 10.1051/mmnp/20116306

Integrated Design of an Active Flow Control System Using a Time-Dependent Adjoint Method

E.J. Nielsen* **and W.T. Jones**

Computational AeroSciences Branch, NASA Langley Research Center
Hampton, Virginia 23693 USA

Abstract. An exploratory study is performed to investigate the use of a time-dependent discrete adjoint methodology for design optimization of a high-lift wing configuration augmented with an active flow control system. The location and blowing parameters associated with a series of jet actuation orifices are used as design variables. In addition, a geometric parameterization scheme is developed to provide a compact set of design variables describing the wing shape. The scaling of the implementation is studied using several thousand processors and it is found that asynchronous file operations can greatly improve the overall performance of the approach in such massively parallel environments. Three design examples are presented which seek to maximize the mean value of the lift coefficient for the coupled system, and results demonstrate improvements as high as 27% relative to the lift obtained with non-optimized actuation. This lift gain is more than three times the incremental lift provided by the non-optimized actuation.

Key words: design, flow control, unsteady, adjoint, unstructured, Navier-Stokes
AMS subject classification: 30E99, 35Q30, 49M29, 68U99, 68W10, 76G25, 76M12, 76N25, 93C20

1. Introduction

The last two decades have seen tremendous growth in the use of adjoint-based algorithms for design optimization of aerodynamic concepts. The primary benefit of this approach is the ability to perform rigorous sensitivity analyses for complex flowfields at the nominal cost of a single

*Corresponding author. E-mail: Eric.J.Nielsen@nasa.gov

Article published by EDP Sciences and available at http://www.mmnp-journal.org or http://dx.doi.org/10.1051/mmnp/20116306

https://doi.org/10.1051/mmnp/20116306 Published online by Cambridge University Press

flowfield solution, regardless of the number of design variables. Numerous research groups have applied these methods to a wide range of aerospace applications. For extensive surveys of recent work in this area, the reader is referred to the excellent overviews given in Refs. [27] and [37].

However, to date, the use of adjoint techniques has been largely confined to the control of steady flows. The primary impediment associated with the unsteady form of the equations lies in the practical requirements of such an approach. Strictly speaking, adjoint methods require access to the entire time-dependent flow solution in order to perform sensitivity analyses for general unsteady phenomena. This is in stark contrast to the use of adjoint methods for steady problems, which merely linearize about the final steady-state solution. Furthermore, the cost associated with a single unsteady solution for a complex aerospace application can often be considerable. In such cases, the computational requirements necessary to perform a formal adjoint-based optimization for complex applications can be tremendous despite the algorithmic efficiencies. Nevertheless, adjoint-based methods are gradually making in-roads into the unsteady regime [8, 23, 25, 26, 28, 43].

Active flow control concepts would appear to be excellent candidates for the use of formal optimization methods. The use of synthetic jets and other actuation devices has become widespread across a broad range of configurations [5, 9, 11, 18, 19, 20, 36, 42, 47, 50]. However, the strategies used to establish the location and performance characteristics of such devices are often heuristically-based at best and the use of such devices is often an afterthought on an existing configuration. Instead, a simultaneous physics-based formal optimization of the combined vehicle configuration and actuation system could potentially yield unforeseen performance benefits for the coupled system. Examples of several previous efforts in this area are described in [4, 10, 12, 15, 24, 41, 51].

In [28], a discrete adjoint implementation for three-dimensional turbulent flows on dynamic unstructured grids was presented, and large-scale optimization examples were demonstrated for a tilt-rotor configuration and a fighter jet undergoing a simulated aeroelastic motion. The use of this design methodology is now examined in the context of a high-lift wing section augmented with an active flow control system. First, the formulation of the flow and adjoint equations is reviewed. The wing section and baseline flow conditions for the current study are introduced, as well as a geometric parameterization scheme describing the configuration. The parallel scaling of the implementation is evaluated using several thousand processors with special attention given to file I/O performance, which is critical to the overall efficiency of the approach. Finally, three optimization examples are presented for the test case considered.

2. Governing Equations

2.1. Flow Equations

The governing flowfield equations are the unsteady Reynolds-averaged Navier-Stokes equations, which may be written in the following form:

$$\frac{\partial}{\partial t}\int_V \mathbf{q} dV + \oint_{\partial V} (\mathbf{F}_i - \mathbf{F}_v)\cdot \hat{\mathbf{n}} dS = 0, \tag{2.1}$$

https://doi.org/10.1051/mmnp/20116306 Published online by Cambridge University Press

where V is the control volume bounded by the surface ∂V. The vector $\mathbf{q}$ represents the conserved variables for mass, momentum, and energy, the vectors $\mathbf{F}_i$ and $\mathbf{F}_v$ are the inviscid and viscous fluxes, respectively, and $\hat{\mathbf{n}}$ is an outward-pointing unit face normal.

By defining a volume-averaged quantity $\mathbf{Q}$ within each control volume,

$$\mathbf{Q} = \frac{\int_V \mathbf{q} dV}{V}, \tag{2.2}$$

the conservation equations take the form

$$\frac{\partial(\mathbf{Q}V)}{\partial t} + \oint_{\partial V} (\mathbf{F}_i - \mathbf{F}_v) \cdot \hat{\mathbf{n}} dS = 0, \tag{2.3}$$

where the conserved variables are defined as $\mathbf{Q} = [\rho, \rho u, \rho v, \rho w, E]^T$. Here, ρ is the density, u, v, and w are the Cartesian components of velocity, and E is the total energy per unit volume. The inviscid and viscous flux vectors, $\mathbf{F}_i$ and $\mathbf{F}_v$, are not explicitly shown here but can be found in [1]. The equations are closed with the perfect gas equation of state and an appropriate turbulence model for the eddy viscosity.

In the current study, a backward difference formula (BDF) known as BDF2opt resulting from a linear combination of the standard BDF2 and BDF3 schemes [35, 52] is employed, so that the equations can be evaluated at time level n as follows:

$$V^n \frac{a\mathbf{Q}^n + b\mathbf{Q}^{n-1} + c\mathbf{Q}^{n-2} + d\mathbf{Q}^{n-3}}{\Delta t} + \mathbf{R}^n = 0, \tag{2.4}$$

where

$$\mathbf{R}^n \equiv \oint_{\partial V} (\mathbf{F}_i^n - \mathbf{F}_v^n) \cdot \hat{\mathbf{n}} dS. \tag{2.5}$$

For $n \geq 3$, the temporal coefficients a, b, c, and d are taken as 1.66, -2.48, 0.98, and -0.16, respectively. Similar to the standard BDF2 scheme, BDF2opt is guaranteed to be stable and is formally second-order accurate; however, the BDF2opt scheme has a smaller coefficient on the leading term of the truncation error. For $n = 2$ and $n = 1$, the standard BDF2 ($a = 1.5$, $b = -2.0$, $c = 0.5$, $d = 0.0$) and BDF1 ($a = 1.0$, $b = -1.0$, $c = d = 0.0$) schemes are used, respectively. Initial solutions are specified at the initial time level $n = 0$.

References [1, 6, 13, 29] describe the flow solver used in the current work. The code can be used to perform aerodynamic simulations across the speed range and an extensive list of options and solution algorithms is available for spatial and temporal discretizations on general static or dynamic mixed-element unstructured meshes which may or may not contain overset grid topologies.

In the current study, the spatial discretization uses a finite-volume approach in which the dependent variables are stored at the vertices of tetrahedral meshes. Inviscid fluxes at cell interfaces are computed using the upwind scheme of Roe [40], and viscous fluxes are formed using an approach equivalent to a finite-element Galerkin procedure [1]. The eddy viscosity is modeled using the one-equation approach of Spalart and Allmaras [49]. Parallel scalability to thousands of processors is achieved through domain decomposition and message passing communication. Pre-

https://doi.org/10.1051/mmnp/20116306 Published online by Cambridge University Press

and postprocessing operations are also performed in parallel using distributed memory, avoiding the need for a single image of the mesh or solution at any time and ultimately yielding a highly efficient end-to-end simulation paradigm.

An approximate solution of the linear system of equations formed within each time step is obtained through several iterations of a multicolor Gauss-Seidel point-iterative scheme. The turbulence model is integrated all the way to the wall without the use of wall functions and is solved separately from the mean flow equations at each time step with a time integration and linear system solution scheme identical to that employed for the mean flow equations.

2.2. Grid Equations

The grid equations are posed in the general form $\mathbf{G}(\mathbf{X}, \mathbf{D}) = 0$, where $\mathbf{X}$ is the vector of mesh coordinates and $\mathbf{D}$ is the vector of design variables. In the current study, the grid is held static during the simulation and is subject to deformation only as a result of the shape optimization procedure. The mesh is modeled as an elastic medium that obeys the elasticity relations of solid mechanics, where an auxiliary system of linear partial differential equations (PDEs) is solved to determine the mesh coordinates after each shape update. Discretization of these PDEs yields a linear system of equations:

$$\mathbf{K}\mathbf{X} = \mathbf{X}_{surf}, \tag{2.6}$$

where $\mathbf{K}$ represents the elasticity coefficient matrix and $\mathbf{X}_{surf}$ is the vector of updated surface coordinates, complemented by zeros for all interior coordinates. Thus for the static grid formulation used here, the grid operator takes the specific form:

$$\mathbf{G}(\mathbf{X}, \mathbf{D}) \equiv \mathbf{X}_{surf} - \mathbf{K}\mathbf{X}. \tag{2.7}$$

The material properties of the system are chosen based on the local cell geometry and proximity to the surface, and the system is solved using a preconditioned generalized minimal residual algorithm. For further details on the approach, see [6, 30, 44].

3. Cost Functions and Adjoint Equations

The steady-state adjoint implementation described in [29]-[34] permits multiple objective functions and explicit constraints of the following form, each containing a summation of individual components:

$$f_i = \sum_{j=1}^{J_i} \omega_j (C_j - C_j^*)^{p_j}. \tag{3.1}$$

Here, ω_j represents a user-defined weighting factor; C_j is an aerodynamic coefficient such as the total drag or the pressure or viscous contributions to such quantities, the $*$ superscript indicates a user-defined target value of C_j, and p_j is a user-defined exponent chosen so that f_i is a convex functional. The user may specify computational boundaries to which each component function applies. The index i indicates a possibility to introduce several different cost functions or constraints,

https://doi.org/10.1051/mmnp/20116306 Published online by Cambridge University Press

which may be useful if the user desires separate sensitivities, e.g., for lift, drag, pitching moment, etc.

For the unsteady implementation described in [28], similar general cost functions f_i^n are defined at each time level n. The integrated cost function f_i is defined as a discrete time integral over a certain time interval $[t_i^1, t_i^2]$:

$$f_i = \sum_{n=N_i^1}^{N_i^2} f_i^n \Delta t, \tag{3.2}$$

where user-specified time levels N_i^1 and N_i^2 correspond to t_i^1 and t_i^2, respectively.

The discrete adjoint-based optimization methodology is based on the method of Lagrange multipliers, which is used to enforce the governing equations as constraints. For the sake of simplicity, a single cost function is assumed and therefore the index i is omitted. For the time-dependent equations, the Lagrangian functional is defined as follows:

$$\begin{aligned} L(\mathbf{D}, \mathbf{Q}, \mathbf{X}, \mathbf{\Lambda}_f, \mathbf{\Lambda}_g) &= \sum_{n=1}^{N} f^n \Delta t \\ &+ \sum_{n=1}^{N} [\mathbf{\Lambda}_f^n]^T \left(\frac{a\mathbf{Q}^n + b\mathbf{Q}^{n-1} + c\mathbf{Q}^{n-2} + d\mathbf{Q}^{n-3}}{\Delta t} V + \mathbf{R}^n \right) \Delta t \\ &+ (f^0 + [\mathbf{\Lambda}_0^f]^T \mathbf{R}^{in}) \Delta t + \mathbf{\Lambda}_g^T \mathbf{G} \Delta t \end{aligned} \tag{3.3}$$

where $f^n \equiv 0$ for $n < N^1$ and $n > N^2$; $\mathbf{G} = 0$ are the grid equations; $\mathbf{\Lambda}_f^n$ and $\mathbf{\Lambda}_g^n$ are vectors of Lagrange multipliers associated with the flow and grid equations, respectively; $\mathbf{D}$ is a vector of design variables; and $\mathbf{R}^{in} = 0$ is the initial condition for the flow equations.

The Lagrangian is differentiated with respect to $\mathbf{D}$, assuming that f^n depends on $\mathbf{Q}_n$, $\mathbf{X}$, and $\mathbf{D}$; $\mathbf{R}^{in}$ depends on $\mathbf{Q}^0$, $\mathbf{X}$, and $\mathbf{D}$; and $\mathbf{R}^n$ depends on $\mathbf{Q}^n$, $\mathbf{X}$, and $\mathbf{D}$. Regrouping terms to isolate the coefficients of $\partial \mathbf{Q}^n / \partial \mathbf{D}$ and equating the coefficients to zero yields the final form of the adjoint equations for the flowfield: for $3 \leq n \leq N$:

$$(a\mathbf{\Lambda}_f^n + b\mathbf{\Lambda}_f^{n+1} + c\mathbf{\Lambda}_f^{n+2} + d\mathbf{\Lambda}_f^{n+3}) \frac{V}{\Delta t} + \left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}^n} \right]^T \mathbf{\Lambda}_f^n = - \left[\frac{\partial f^n}{\partial \mathbf{Q}^n} \right]^T, \tag{3.4}$$

where $\mathbf{\Lambda}^{N+1} = \mathbf{\Lambda}^{N+2} = \mathbf{\Lambda}^{N+3} = 0$;
for $n = 2$:

$$(\frac{3}{2}\mathbf{\Lambda}_f^2 + b\mathbf{\Lambda}_f^3 + c\mathbf{\Lambda}_f^4 + d\mathbf{\Lambda}_f^5) \frac{V}{\Delta t} + \left[\frac{\partial \mathbf{R}^2}{\partial \mathbf{Q}^2} \right]^T \mathbf{\Lambda}_f^2 = - \left[\frac{\partial f^2}{\partial \mathbf{Q}^2} \right]^T; \tag{3.5}$$

for $n = 1$:

$$(\mathbf{\Lambda}_f^1 - 2\mathbf{\Lambda}_f^2 + c\mathbf{\Lambda}_f^3 + d\mathbf{\Lambda}_f^4) \frac{V}{\Delta t} + \left[\frac{\partial \mathbf{R}^1}{\partial \mathbf{Q}^1} \right]^T \mathbf{\Lambda}_f^1 = - \left[\frac{\partial f^1}{\partial \mathbf{Q}^1} \right]^T; \tag{3.6}$$

and for the initial conditions:

$$(-\mathbf{\Lambda}_f^1 + \frac{1}{2}\mathbf{\Lambda}_f^2 + d\mathbf{\Lambda}_f^3) \frac{V}{\Delta t} + \left[\frac{\partial \mathbf{R}^{in}}{\partial \mathbf{Q}^0} \right]^T \mathbf{\Lambda}_f^0 = - \left[\frac{\partial f^0}{\partial \mathbf{Q}^0} \right]^T. \tag{3.7}$$

https://doi.org/10.1051/mmnp/20116306 Published online by Cambridge University Press

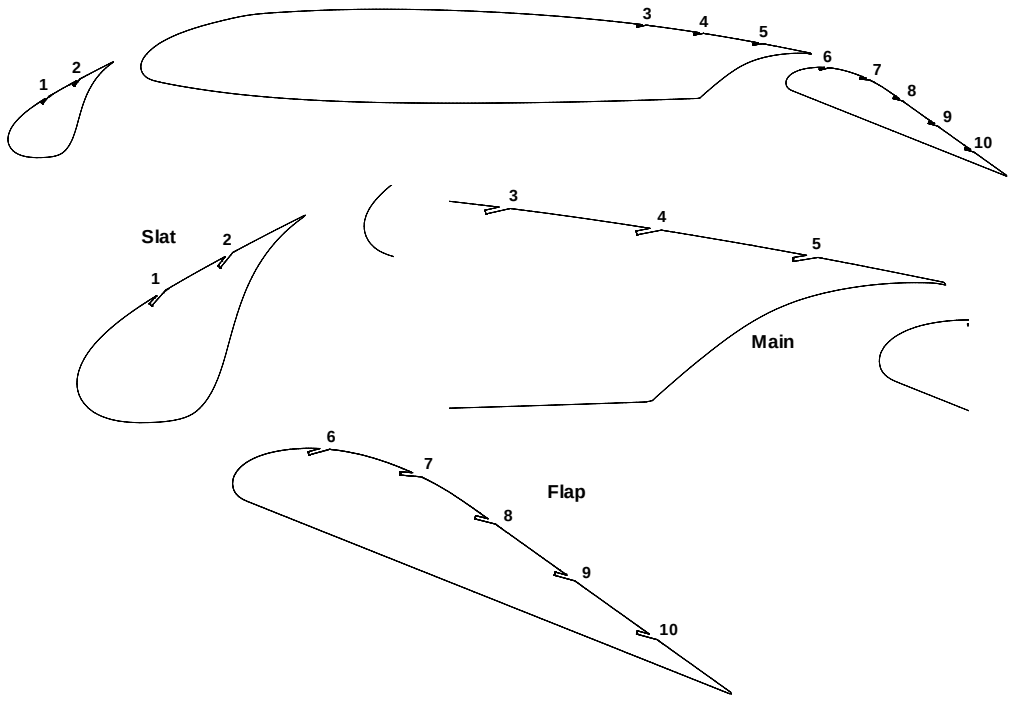


Figure 1: Baseline airfoil geometry from [48].

Collecting the coefficients of $\partial\mathbf{X}/\partial\mathbf{D}$ and equating them to zero leads to similar adjoint equations for the grid. Assuming that the grid operator $\mathbf{G}$ depends on $\mathbf{X}$ and $\mathbf{D}$, the grid adjoint equations are defined as

$$-\left[\frac{\partial \mathbf{G}}{\partial \mathbf{X}}\right]^T \mathbf{\Lambda}_g = \sum_{n=3}^{N} \left\{ \left[\frac{\partial f^n}{\partial \mathbf{X}}\right]^T + \left[\frac{a\mathbf{Q}^n + b\mathbf{Q}^{n-1} + c\mathbf{Q}^{n-2} + d\mathbf{Q}^{n-3}}{\Delta t}\frac{\partial V}{\partial \mathbf{X}} + \frac{\partial \mathbf{R}^n}{\partial \mathbf{X}}\right]^T \mathbf{\Lambda}_f^n \right\}$$
$$+ \left[\frac{\partial f^2}{\partial \mathbf{X}}\right]^T + \left[\frac{1}{2}\frac{3\mathbf{Q}^2 - 4\mathbf{Q}^1 + \mathbf{Q}^0}{\Delta t}\frac{\partial V}{\partial \mathbf{X}} + \frac{\partial \mathbf{R}^2}{\partial \mathbf{X}}\right]^T \mathbf{\Lambda}_f^2 \tag{3.8}$$
$$+ \left[\frac{\partial f^1}{\partial \mathbf{X}}\right]^T + \left[\frac{\mathbf{Q}^1 - \mathbf{Q}^2}{\Delta t}\frac{\partial V}{\partial \mathbf{X}} + \frac{\partial \mathbf{R}^1}{\partial \mathbf{X}}\right]^T \mathbf{\Lambda}_f^1 + \left[\frac{\partial f^0}{\partial \mathbf{X}}\right]^T + \left[\frac{\partial \mathbf{R}^{in}}{\partial \mathbf{X}}\right]^T \mathbf{\Lambda}_f^0.$$

With the adjoint coefficients satisfying the flowfield and grid adjoint equations, the sensitivity derivatives are calculated as follows:

$$\frac{dL}{d\mathbf{D}} = \sum_{n=1}^{N} \left(\frac{\partial f^n}{\partial \mathbf{D}} + \left[\mathbf{\Lambda}_f^n\right]^T \frac{\partial \mathbf{R}^n}{\partial \mathbf{D}} \right) \Delta t + \left(\frac{\partial f^0}{\partial \mathbf{D}} + \left[\mathbf{\Lambda}_0^f\right]^T \frac{\partial \mathbf{R}^{in}}{\partial \mathbf{D}} + \mathbf{\Lambda}_g^T \frac{\partial \mathbf{G}}{\partial \mathbf{D}} \right) \Delta t. \tag{3.9}$$

The implementation of the time-dependent adjoint and sensitivity equations has been previously described and verified in [28]. The general methodology for solving the adjoint equations is similar in nature and reverse in order to that used for the primal equations. In practice, the primary

https://doi.org/10.1051/mmnp/20116306 Published online by Cambridge University Press

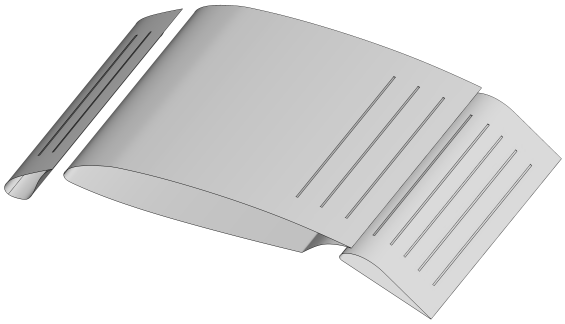


Figure 2: Baseline wing geometry.

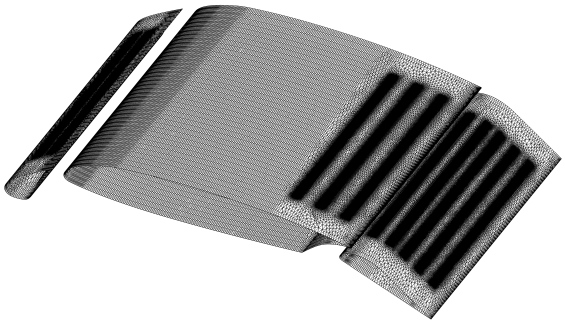


Figure 3: Surface grid for baseline wing.

concern in solving the adjoint equations is the need to have $\mathbf{Q}^n$ available at all time levels during the reverse time integration for $\mathbf{\Lambda}_f^n$. For three-dimensional flows with a one-equation turbulence model, this requires 48 bytes of disk storage per grid point per physical time step. For realistic problems, this can easily yield disk storage requirements on the order of many terabytes. In the current implementation, the values of $\mathbf{Q}^n$ are written to disk independently by each processor for each local grid partition at the completion of each time step of the forward problem. During the solution of the adjoint equations, $\mathbf{Q}^n$ is loaded as needed from disk in the same fashion. The practical implications of this strategy will be discussed in a subsequent section. Finally, any third-party geometric parameterization tool providing $\mathbf{X}_{surf}$ and $\partial\mathbf{X}_{surf}/\partial\mathbf{D}$ may be used with the current implementation.

4. Test Case

4.1. Geometry and Computational Grid

The geometry for the current study is based on a three-element airfoil section taken from [48]. The airfoil consists of a main element with a leading edge slat and trailing edge flap as shown in Figure 1. A series of ten rectangular jet orifices are located along the chord c of the airfoil. Each orifice has a $0.002c$ cross-sectional width with its centerline oriented 20° to the local surface tangent. For discussion purposes, the orifices are numbered as shown: two orifices are located on the slat, three are located on the main element, and the remaining five orifices are located on the flap.

To create the three-dimensional wing section used in the current study, the two-dimensional airfoil section has been extruded one chord length at a 0° sweep angle in the spanwise direction. Rectangular jet slots with cross sections and chordwise locations identical to the nominal airfoil geometry are centered in the spanwise direction and measure $0.75c$ in length. The resulting three-dimensional geometry is shown in Figure 2. The computational grid consists of 50,870,813 tetrahedral elements and 8,823,297 grid points and has been generated using the methodologies described in [16, 39]. This grid resolution has been chosen as a compromise between adequately

https://doi.org/10.1051/mmnp/20116306 Published online by Cambridge University Press

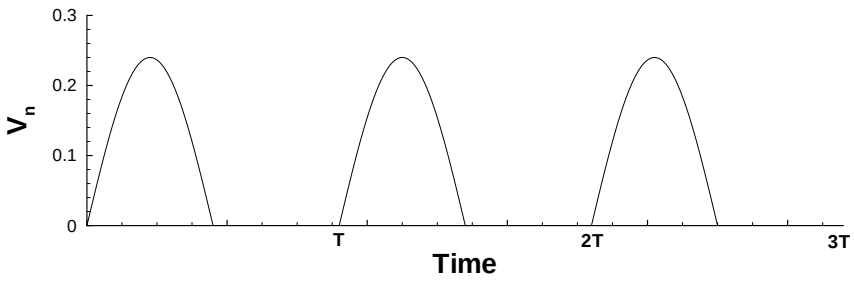


Figure 4: Specified normal velocity profile for each jet orifice; positive values indicate blowing.

resolving the pertinent flow physics and maintaining a reasonable overall computational cost. The planar sidewalls are modeled using tangency boundary conditions and are deformed in-plane in response to changes in the wing surface geometry described below. A view of the surface grid for the wing section is shown in Figure 3. The farfield boundary conditions utilize a locally one-dimensional system of Riemann invariants [1].

4.2. Baseline Conditions and Aerodynamic Performance

The external freestream conditions are a Mach number of 0.1, a 19° angle of attack, and a Reynolds number of 3 million based on the chord of the wing. As reported in [48], these conditions correspond to a near maximum lift condition for the baseline airfoil section with no actuation applied. Although the implementation supports the use of dynamic grids which could be used to mimic actual piston diaphragm motion, a time-dependent blowing condition is instead imposed at the base of each rectangular slot, resulting in periodic jet flow emanating from each orifice along the chord of the wing section. The actuation is achieved through a prescribed sinusoidally-varying velocity V_{n} normal to the base of each slot, clipped at a minimum value of zero:

$$V_{\text{n}} = \max[0, \bar{V}\sin(\omega t + \phi)] \tag{4.1}$$

The clipping is applied to simplify the numerical boundary condition for the purposes of the current investigation. Similar to the approach taken in [48], the value of $\bar{V}$ is 0.24, with the resulting blowing profile shown in Figure 4 where T represents the period of the blowing cycle. The density is also specified at the base of each slot and is taken as the freestream value, yielding a maximum nominal local Mach number of 0.24. The reduced frequency $F^+ \equiv f \cdot c/V_\infty$ for each jet is also taken from [48] as 1.52, and the boundary conditions are applied such that each jet is 180° out of phase ϕ with the immediately adjacent jets. The sidewalls of each slot are modeled as no-slip surfaces.

All simulations are performed using the BDF2opt scheme with 10 subiterations. This value has been chosen based on numerical experience and represents a compromise between temporal accuracy and computational cost. Ideally, an error-based subiterative controller such as that described in [6] would be used; such an approach is available for the solution of (2.4), but has not been implemented for (3.4). The effect of time step size was investigated and it was found that

https://doi.org/10.1051/mmnp/20116306 Published online by Cambridge University Press

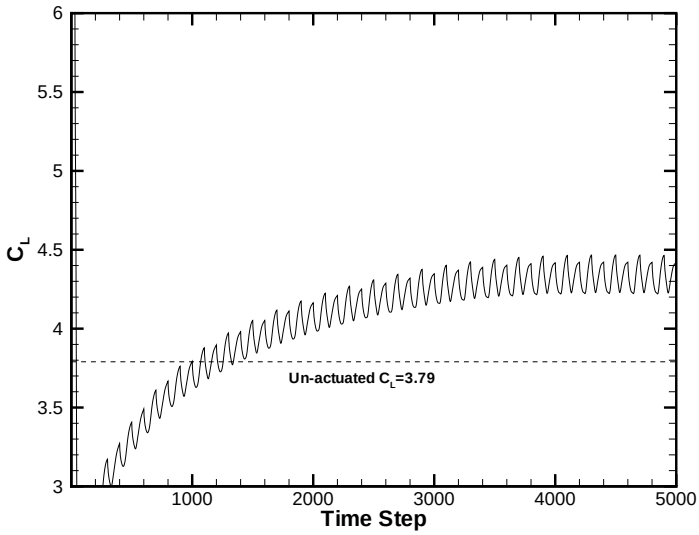


Figure 5: Unsteady lift profile for the baseline geometry and blowing conditions.

the asymptotic mean value of the lift coefficient did not change significantly using more than 100 steps per blowing cycle. However, a time step corresponding to 200 steps per cycle was chosen in order to resolve potentially higher blowing frequencies to be discussed in a subsequent section.

The baseline lift profile for the nominal flow conditions is shown in Figure 5. The lift coefficient history shows the effects of initial flowfield transients, after which a periodic behavior is established following 5,000 physical time steps or 25 jet blowing cycles. Averaging the lift coefficient over the final 1,000 time steps yields a mean value of 4.33. For reference, the steady value of the lift coefficient corresponding to no jet actuation is 3.79. This value has been determined by performing a simulation with the blowing velocity set to zero at the base of each jet. A similar value of the lift is obtained by using a grid with no jet orifices present.

5. Design Variables

5.1. Jet Parameters

For each of the ten jets located along the chord of the wing, the frequency, velocity magnitude, and phase of the blowing actuation applied at the jet base may be used as design variables. The reduced frequency of each jet is allowed to vary from 0.76 to 3.04, such that the minimum number of time steps per cycle would be 100, corresponding to the highest allowable frequency. The normal velocity at the base of each jet is allowed to range from zero to 0.48, while the phase is allowed to vary between 0° and 180°.

https://doi.org/10.1051/mmnp/20116306 Published online by Cambridge University Press

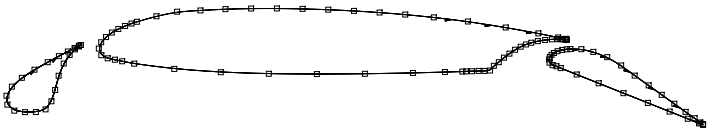


Figure 6: B-spline control point locations for shape parameterization.

5.2. Geometric Parameterization

In order to vary the wing shape and chordwise location and incidence angle of each jet orifice during the course of an optimization, a specialized geometry tool has been developed to parameterize wing sections such as the model used in the current study. This tool assumes a nominally two-dimensional geometry in the spanwise direction, with the exception of the finite span of the jet actuation slots. The application is not intended to handle general three-dimensional configurations but rather to facilitate the current exploratory study. General tools such as the free-form deformation packages outlined in [45] and [46] could be modified to accommodate geometric features arbitrarily translating along general morphing surfaces as required by the current investigation; however, such development work is beyond the scope of this effort.

5.2.1. Geometry Preprocessing

A preprocessing step is first performed in which the surface grid is loaded for the body of interest, along with topological information related to the boundary definition and any actuation slots that may be present. Based on this information, a two-dimensional, cubic B-spline curve similar to that used in [2, 3] is fit to the existing mesh coordinates, establishing an analytic representation of the nominally two-dimensional external wing surface. The least squares global approximation technique described in [38] is used in the curve fitting process. The fit is constructed given the desired polynomial degree of the curve, number of control points, and the type of curve parameterization; an arc length parameterization is used here. In addition to the parametric location and incidence angle of each jet, the displacements of each B-spline control point normal to the surface can also be used as design variables. Finally, the horizontal, vertical, and rotational displacement of each body may also be specified as design parameters.

Since the number of control points contributes to the total number of design variables, multiple B-spline segments of common degree are defined to reduce the number of control points needed to accurately fit the data. The segments and the corresponding number of control points for each are manually defined as part of the preprocessing step. To facilitate the design process, the individual B-spline segments are then concatenated together to form a single composite curve fit of the data. By definition, the segments are compatible as they have a common degree. The concatenation is straightforward, whereby the curves are linked sequentially and the parameterization of each segment is adjusted such that the starting parameter of the segment is equal to the ending parameter of the preceding segment. The knot vector of the composite curve has multiplicity equal to the common degree at the junction points of the segments. Knot removal is then used to recover continuity at the segment junctions. The control points for the final B-spline definition are shown

https://doi.org/10.1051/mmnp/20116306 Published online by Cambridge University Press

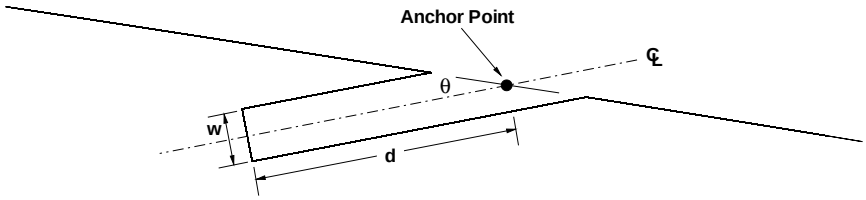


Figure 7: Jet notation.

in Figure 6.

The baseline locations and incidence angles θ of each jet orifice are also established, including an anchor point for each jet as shown in Figure 7. The coordinates of the anchor point represent the intersection of the jet centerline with the B-spline defining the wing surface. The corresponding B-spline parameter value of the anchor point is determined by treating the jet centerline as an implicit line and finding roots by evaluating the B-spline and its derivatives at a series of Newton steps. The normal distance from the anchor point to the jet base is taken as the depth d of the jet. The baseline surface grid and spline and jet information are then stored for use during the actual optimization.

5.2.2. Surface Grid Deformation

An additional tool has been developed to deform the surface mesh for each body during the course of the design. This tool loads the baseline information established in the preprocessing step as well as the values of the design variables for the current body. Each jet orifice is rotated about its anchor point and translated along the spline curve as specified, maintaining the baseline $0.002c$ cross-sectional width. The resulting displacement of the anchor point is determined and the interior of the jet is explicitly relocated since it is not contained within the analytic B-spline definition. The parametric intersections of the sidewalls of each jet are determined and the length of each jet sidewall is then extended or reduced to maintain the baseline jet depth. The net displacements are distributed in a linearly decaying fashion for each grid point along the sidewall edges as shown in the side views in Figure 8. The locations of the surface grid points interior to each sidewall boundary are then smoothed based on the updated locations of the grid points along the sidewall edges.

The parametric displacement of the intersections between the external wing surface and jet slots are then determined. These parametric displacements are linearly decayed across the remaining grid points on the external wing surface as a final step. Since in general the width of a slot opening may change as the jet is altered, special care is taken to reposition the points on the wing surface at chordwise locations between the upstream and downstream sides of each jet accordingly as shown in the top views in Figure 8.

5.2.3. Surface Mesh Jacobians

Rather than develop an analytic form of the Jacobian matrix $\partial \mathbf{X}_{surf}/\partial \mathbf{D}$ required by the discrete adjoint formulation, a complex-variable approach as outlined in [22] is used to evaluate these lin-

https://doi.org/10.1051/mmnp/20116306 Published online by Cambridge University Press

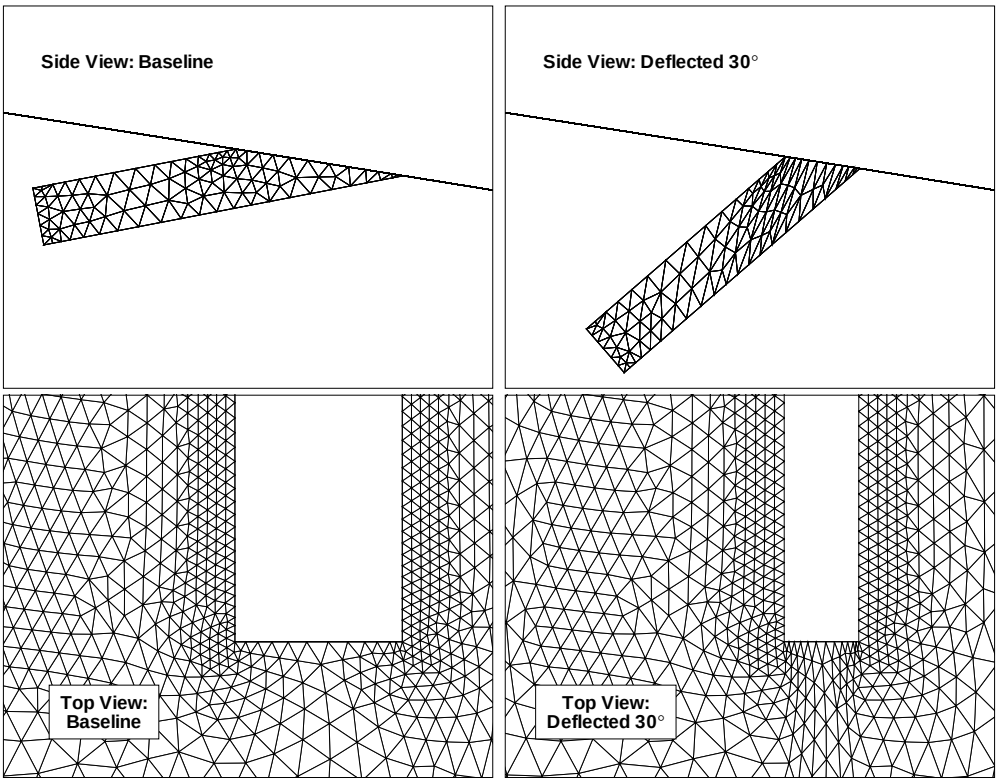


Figure 8: Illustration of jet rotation. Surface grid for interior of jet has been omitted from top views for clarity.

earizations numerically. In this approach, an expression for the derivative of a real-valued function $f(x)$ may be found by expanding the function in a complex-valued Taylor series, using an imaginary perturbation $i\varepsilon$:

$$\frac{\partial f}{\partial x} = \frac{Im[f(x+i\varepsilon)]}{\varepsilon} + O(\varepsilon^2) \tag{5.1}$$

The primary advantage of this method is that true second-order accuracy may be obtained by selecting step sizes without concern for subtractive cancellation error typically present in real-valued divided differences. Moreover, the implementation merely requires the real-valued variables in the application to be declared of complex type with execution carried out as normal, once the imaginary perturbation has been applied to the parameter of interest.

For the current application, an outer loop is placed around the series of function calls required to generate the deformed surface grid. The loop is executed once for each design variable associated with each jet. Within the loop, the imaginary part of the current design variable is perturbed by $\varepsilon = 10^{-50}$, the complex-valued surface grid is evaluated, and the elements of the Jacobian matrix $\partial \mathbf{X}_{surf}/\partial \mathbf{D}$ are extracted as the imaginary parts of the result. The cost of this procedure scales linearly with the number of design variables; however, the expense associated with evaluating the parameterized surface grid as outlined here is negligible, resulting in an acceptable overall cost incurred by the complex-variable approach.

https://doi.org/10.1051/mmnp/20116306 Published online by Cambridge University Press

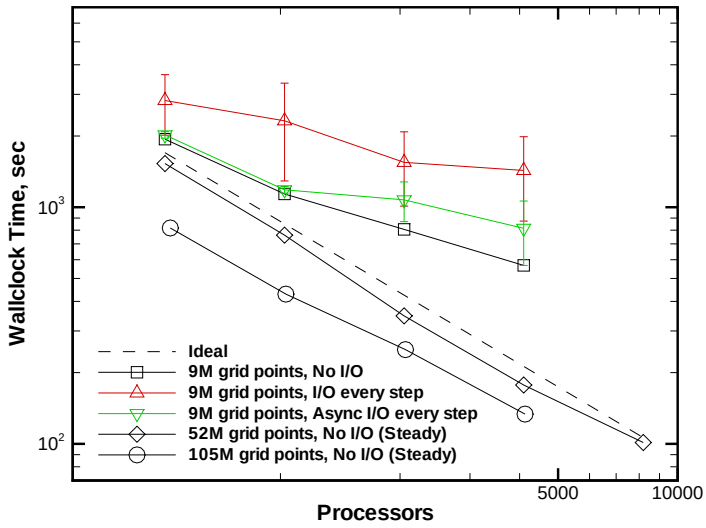


Figure 9: Parallel scaling for flow solver.

6. Parallel Scaling and I/O Performance

In order for routine optimization of large-scale unsteady problems to become tractable, the implementation must utilize massively parallel computing environments as efficiently as possible. Ideally, the time required for the application to execute would scale inversely with the number of processors used. For traditional solvers in which execution time is dominated by floating-point operations, this is generally straightforward to achieve when sufficiently large grids are distributed across dozens, hundreds, or even a few thousand processors. For unsteady flow simulations for which a large number of time steps may be required, the use of many thousands of processors becomes attractive in addressing potentially very long runtimes. However, partitioning the spatial mesh for such environments can result in exceedingly small amounts of computational work per time step on each processor. Furthermore, the overhead cost associated with inter-processor communication generally grows with the number of partitions. The net result is that a typical solver may experience linear (or perhaps even superlinear) speedup over an initial range of processors for a given grid. However, as the computational work per processor is reduced and communication costs increase, solvers generally experience a point at which diminishing returns are observed, and may even exhibit an inverse behavior where the use of more processors actually causes the application to exhibit longer run times.

In the current study, the issue of parallel efficiency is further complicated by the need to perform frequent disk I/O involving large amounts of data. At each time step of the flow solution procedure, the solution $\mathbf{Q}^n$ must be stored to disk; these back planes of data must then be loaded successively in reverse order as the adjoint solution proceeds. Great care must be taken to perform these operations in a scalable manner as will be discussed below. It should also be noted that any

https://doi.org/10.1051/mmnp/20116306 Published online by Cambridge University Press

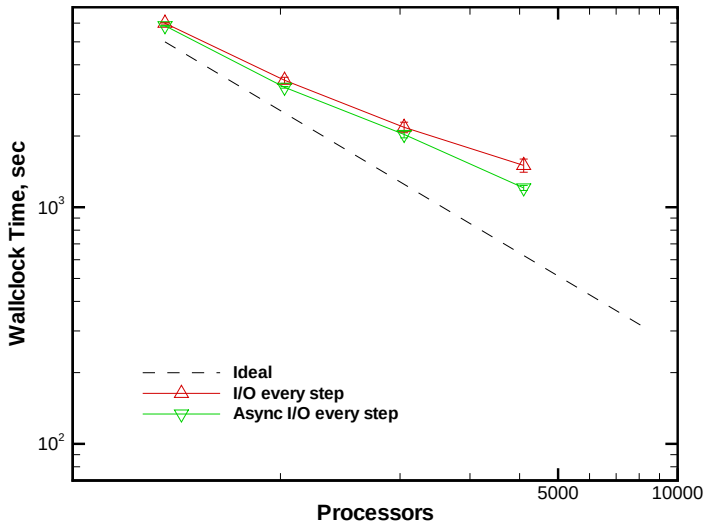


Figure 10: Parallel scaling for unsteady adjoint solver.

issues related to disk I/O revealed in the current study would likely be magnified in the dynamic grid context, where the mesh coordinates and velocities are also stored at each time step [28].

Whether a single plane of steady data or several back planes of an unsteady solution are required for restart purposes, the solvers traditionally use a strategy in which every processor writes to a single file. This is achieved through the use of the MPI-IO layer present in the MPI-2 specification, or alternatively, through a native implementation based on the stream I/O capability introduced in the Fortran 2003 standard. In general, both approaches perform well and a convenient file containing all of the necessary restart information for a problem involving 20-30 million grid points on O(500-1000) processors can be produced in approximately two to three minutes on commodity hardware.

The traditional approach described above is acceptable when solution files are only periodically required, as the cost associated with the I/O process is amortized over the entire solution procedure. However, this approach becomes untenable for adjoint-based design, where solution data must be stored or loaded at every time step. As described in [28], the baseline strategy for meeting this more stringent requirement relies on each processor simultaneously accessing its own direct-access file, where each record of the file contains all of the data relevant to that partition for time level n. In this manner, the file pointer can be placed immediately at the location of interest and I/O can proceed. This approach worked well for the cases shown in [28] using 1,024 processors. Those simulations utilized a Lustre-based parallel file system [14]; all results described here rely on identical hardware.

Recent experiments have indicated the existing strategy may not be sufficient as the number of processors increases further. To illustrate this trend, 200 time steps of a flow solution are performed on the current high-lift wing test case using a varying number of processors. Each computation is

https://doi.org/10.1051/mmnp/20116306 Published online by Cambridge University Press

performed in a fully-dense fashion (all processing cores within a node are used) on a set of dual-socket, quad-core nodes with 3.0 GHz Intel Xeon E5472 processors and is executed several times to establish repeatability and the effects of varying file system load.

The timing results based solely on computational work are shown as the square symbols in Figure 9; no disk I/O is performed for this result. As expected, the time is reduced with increasing processor count. However, the performance falls short of the ideal profile indicated by the dashed line. This is not surprising, as the nominal number of local grid points drops to approximately 4,000 and 2,000 on 2,048 and 4,096 processors, respectively. Grid partitions of this size represent very little computational work per processor. To demonstrate that the implementation scales well when sufficient work is allocated to each processor, results are also shown from a recent study [21] using grids containing approximately 52 and 105 million points (300 and 600 million elements), respectively. These computations represent steady runs using a different number of time steps, so that the absolute levels of the curves cannot be compared to one another nor to those of the current data. It should also be noted that the 105 million grid point result was performed in a fully-dense fashion on dual-socket, hex-core nodes with Xeon X5670 processors. Nonetheless, the slopes are useful in confirming the desired scaling behavior provided there is sufficient computational work. Finally, note that all results for which disk I/O is not required tend to be highly repeatable; execution times are very consistent across multiple runs.

The previous test is now repeated with $\mathbf{Q}^n$ written to disk at the completion of each time step n using the existing direct-access I/O strategy as previously outlined in [28], with the result included as the triangle symbols in Figure 9. The scaling is generally poor, especially at the higher processor counts. Moreover, the repeatability of each result, denoted by the bars plotted at each data point, is erratic and the potential for an inverse scaling behavior appears to exist.

A modified implementation using the asynchronous I/O feature introduced in Fortran 2003 has been developed to address the poor scaling performance associated with frequent disk I/O. In this approach, the data associated with a read or write statement is buffered, while execution continues for subsequent lines of code. In this manner, floating-point operations may proceed while file operations are performed simultaneously in the background. However, care must be taken not to alter the relevant data before the desired I/O has occurred.

The results from the asynchronous approach are also included in Figure 9 as the inverted triangle symbols. At the lower processor counts, the I/O appears to have been successfully masked behind the floating-point operations as indicated by the close proximity to the results associated with the computations involving no disk I/O. The behavior at these data points is also considerably more repeatable, as shown by the very small vertical bars at these points. The asynchronous I/O also yields a benefit at the higher processor counts, although some overhead for file I/O still appears to exist for these data points. The scaling behavior also appears to be marginally improved over the previous result.

Similar scaling tests are performed for the adjoint solver. In this case, data is read from disk rather than written. Since the adjoint solution procedure requires the values of $\mathbf{Q}^n$, plotting a result based solely on computational work in the absence of any I/O is not practical in this context. The scaling behavior of the baseline implementation using synchronous I/O is shown as the triangle symbols in Figure 10. In general, the wall-clock time is reduced with increasing processor counts,

https://doi.org/10.1051/mmnp/20116306 Published online by Cambridge University Press

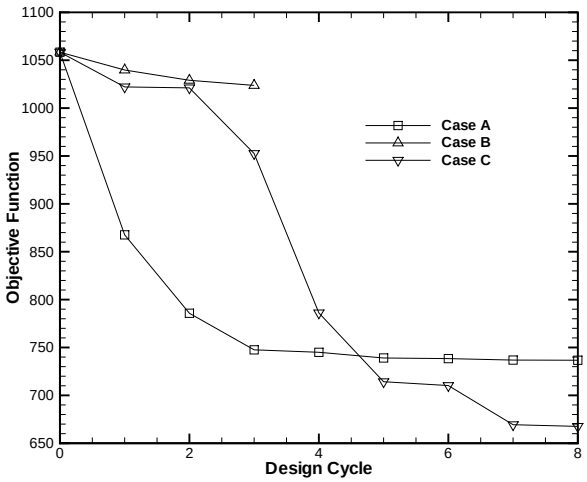


Figure 11: Convergence histories for objective function.

with very little spread in the timings at a given data point. The asynchronous result is also shown as the inverted triangle symbols, with the largest improvements evident at the 4,096-processor data point. The scaling for both strategies in the adjoint solver is better than the corresponding approach for the flow solution. One possible reason for this behavior is the considerable amount of additional computational work the adjoint solver must perform at each time step. The adjoint solution procedure requires exact linearizations of the nonlinear residual, and the linear residual computation performed during each relaxation sweep requires matrix-vector products over the full spatial stencil, which is typically four to five times the size of the nearest-neighbor stencil used for the relaxation operator. The exact linearizations with respect to the grid must also be evaluated at each time step of the adjoint solution. Furthermore, the flow solver has been extensively optimized for computational performance [21], while the adjoint implementation has not been optimized to the same extent. Finally, file-locking necessary during write operations may also contribute to the performance differences observed, as no such locking is required during read operations.

7. Design Results

Based on the results presented above, all subsequent simulations have been performed using 256 dual-socket, quad-core nodes with 3.0 GHz Intel Xeon E5472 processors in a fully dense fashion, for a total of 2,048 processors. The nominal wall-clock time required to perform and store an unsteady flow solution to disk for the current test case using this strategy is approximately 7.5 hours. The corresponding adjoint solution and sensitivity analysis requires approximately 14 hours. Approximately 2.1 terabytes of total disk space are necessary to store a flowfield solution for all time levels.

The objective function for all test cases is to maximize the lift coefficient C_L over the final

https://doi.org/10.1051/mmnp/20116306 Published online by Cambridge University Press

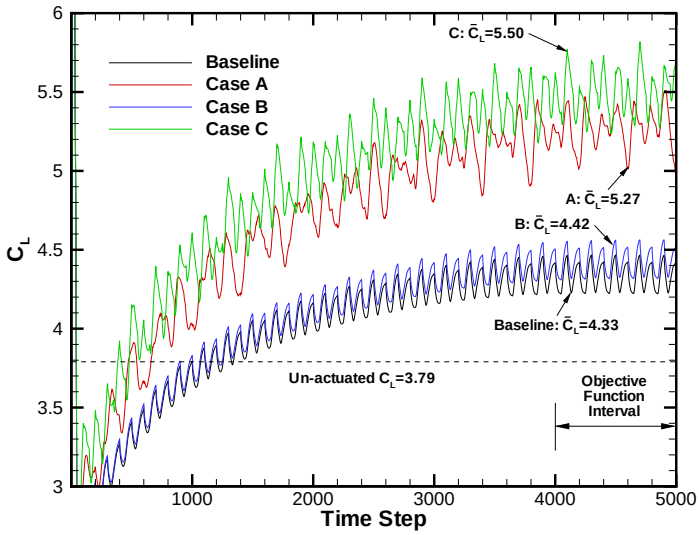


Figure 12: Unsteady lift profiles for the baseline and design results.

1,000 time steps of the simulation:

$$f = \sum_{n=4,001}^{5,000} (C_L - 10.0)^2 \Delta t. \tag{7.1}$$

Here, the target lift coefficient of 10.0 has been chosen to sufficiently exceed the baseline lift profile shown in Figure 5. The optimizations are performed using a trust region method from the package described in [17], and occasionally require restarting due to queuing limitations or sporadic system failures.

Three test cases are used to evaluate the optimization methodology. The first example seeks to minimize the objective function given in (7.1) by holding the external wing shape fixed while altering the jet locations, incidence angles, and blowing parameters. The remaining two test cases allow the wing shape to change in addition to the jet-related variables and also demonstrate the impact of design variable scaling.

7.1. Case A: Variable Jet Parameters, Fixed Wing Geometry

For the first test case, the external shape of the wing is held fixed while the jet arrangement and blowing parameters are used as the design variables. The parametric location, incidence angle, and blowing frequency, velocity, and phase shift are allowed to change for each individual jet, yielding a total of 50 active design variables.

The convergence history for the objective function is shown in Figure 11. After eight design cycles, the value of the objective function has been reduced from its baseline value of 1058.4 to a

https://doi.org/10.1051/mmnp/20116306 Published online by Cambridge University Press

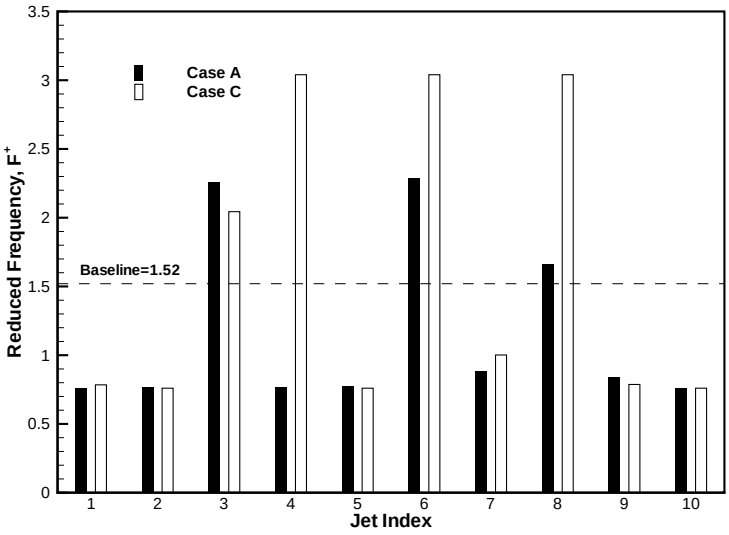


Figure 13: Final frequency values for each jet.

final value of 736.7, although the majority of the improvements have been achieved within the first 3 design cycles. The resulting unsteady lift profile is shown in Figure 12, with the baseline profile also included for reference. The mean value of the lift coefficient over the final 1,000 time steps has been increased 22% from the baseline value of 4.33 to 5.27. The jet velocity amplitude has been increased to its maximum value of 0.48 for all jets, and the final jet frequencies are plotted in Figure 13. All of the frequencies are at or near the lower bound of 0.76, with the exception of jets 3, 6, and 8, whose values have increased above the baseline value of 1.52. The final values of the phase shift for each jet were within 0.1° of the initial values and are not given here. The final jet arrangement is shown in Figure 14. The optimization has moved all of the jets located on the slat and main elements downstream. The jets on the flap have all been moved upstream, with the exception of jet 8 which has been shifted downstream. Minor changes in the jet incidence angles are not visible; however, all jets have been rotated slightly closer to the surface tangent direction with the largest deflection of 0.14° taking place at jet 6. During the course of the optimization, a total of 14 flow solutions and 8 adjoint solutions have been performed. Approximately 444,000 CPU hours were required, or a total of 9 days of wall-clock time.

7.2. Case B: Variable Jet Parameters and Wing Geometry

The next test case is more general in that the external shape of the wing is allowed to vary in addition to the jet parameters used in the previous test. The control points in the vicinity of the trailing edge of each element are held fixed, while a total of 105 control point locations are allowed to vary. In addition, the rotations and horizontal and vertical locations of the slat and flap relative to the main element are also active design variables. Approximate quarter-chord locations are used as the centers of rotation for the slat and flap elements. The combination of these shape-related variables with the 50 jet parameters used in the previous example yields a total of 161 active design

https://doi.org/10.1051/mmnp/20116306 Published online by Cambridge University Press

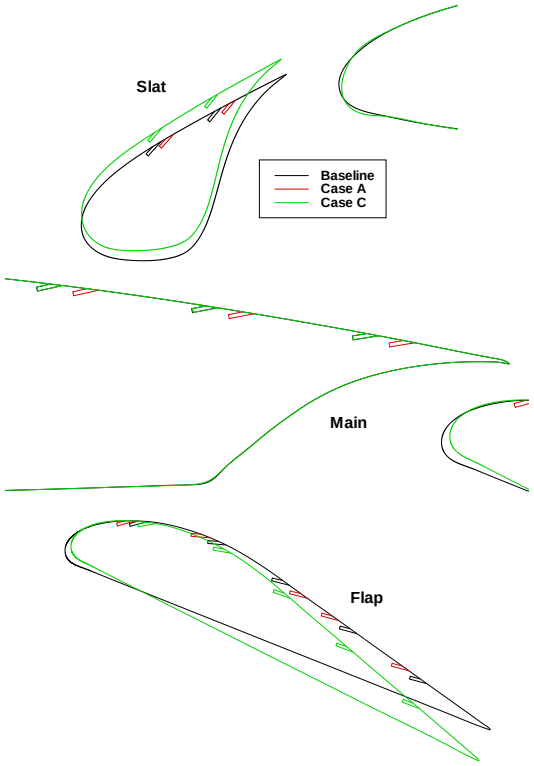


Figure 14: Cross-section of baseline and final wing geometries.

variables for the current test case.

The convergence history for the objective function is included in Figure 11. The optimization progresses quite slowly and has difficulty reducing the objective function by a substantial amount. The lift profile is shown in Figure 12 and reflects the modest reduction in the objective function. The mean value of the lift has been increased to 4.42, a 2% improvement over the baseline value. Changes in the jet parameters and airfoil shape are very minor and are not included here.

Closer inspection of this test case showed that the optimization procedure repeatedly attempted to alter many of the shape-related variables during the first few design cycles, while ignoring the parameters associated with the jet locations and actuation which yielded considerable improvement in the previous example. This is likely explained by the fact that the sensitivity derivatives for the shape variables are generally 1-3 orders of magnitude larger than those for the jet-related parameters. Given the large number of design variables in this example, it is possible the optimization procedure may eventually explore changes in the jet-related parameters and achieve improvements similar to the first test case. However, execution was terminated due to the slow convergence observed over the first three design cycles, which consumed just over 4 days of wall-clock time or approximately 209,000 hours of CPU time. A total of 8 flow solutions and 3 adjoint solutions were performed.

https://doi.org/10.1051/mmnp/20116306 Published online by Cambridge University Press

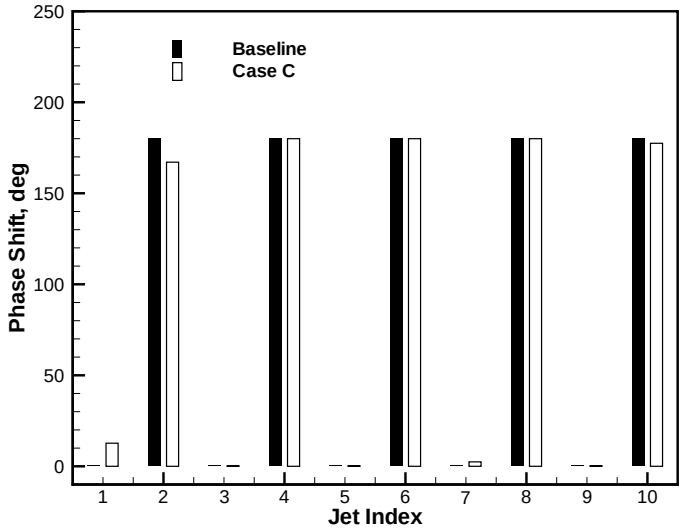


Figure 15: Final phase values for each jet.

7.3. Case C: Variable Jet Parameters and Wing Geometry, with Scaling

As in the previous example, the same general set of 161 design variables describing the wing shape and jet actuation is used. However, the variables and their associated sensitivity derivatives are now normalized using a simple diagonal scaling procedure to improve the numerical conditioning of the optimization procedure.

After eight design cycles, the value of the objective function has been reduced to 667.6 as shown in Figure 11. The time-averaged lift shown in Figure 12 is 5.50, which represents a 27% net increase over the baseline actuated value of 4.33. The velocities imposed at the base of each jet have increased to 0.48 with the exception of jet 1, which has been reduced to a zero velocity condition. The final jet frequencies shown in Figure 13 are similar to those found in the first example; however, the values for jets 4, 6, and 8 have been reached significantly higher values. The geometry for the final wing is included in Figure 14. The jet incidence angles have changed very little; all deflections are closer to the local surface tangent with a maximum deflection of 0.7° for jet 2. The chordwise positions of the jet orifices have changed very little. The orifices located on the slat and main elements have been moved slightly downstream, while all of the jet orifices on the flap have been moved upstream, with the exception of jet 6. The slat has been translated upwards and rotated 1.8° nose-down with slight modifications to its shape. The leading edge of the main element has been blunted and slightly thickened. The flap has been deflected nose-up by the maximum allowed value of five degrees and has also been translated slightly upwards. The final values of the phase angles are shown in Figure 15. Jets 1 and 2 show the largest changes over the baseline values. For this example, the optimizer requested 10 flow solutions and 8 adjoint solutions, requiring approximately 8 days of wall-clock time or 383,000 CPU hours.

https://doi.org/10.1051/mmnp/20116306 Published online by Cambridge University Press

8. Summary and Concluding Remarks

The use of a time-dependent discrete adjoint technique for optimization of a high-lift wing section with an active flow control system has been evaluated. A three-dimensional wing section containing ten jet actuation slots has been created using a spanwise extrusion of an existing high-lift airfoil section. In addition to the jet configuration and blowing parameters, a geometric parameterization based on the use of B-spline curves has been used to provide a compact set of design variables describing the external wing surface. The linearizations of the surface deformation scheme have been evaluated numerically using a complex-variable formulation.

Since the unsteady adjoint formulation requires substantial file I/O, the parallel scaling of the implementation has been studied extensively using several thousand processors. The introduction of an asynchronous I/O paradigm at each time step was helpful in alleviating bottlenecks associated with massively parallel file operations.

Three optimization examples have been performed. The mean value of the lift coefficient has been dramatically increased through optimization of the jet locations and blowing parameters. Simultaneous optimization of the external wing shape and jet parameters required a simple scaling technique to improve the numerical conditioning of the optimization procedure and resulted in a net gain of 27% relative to the lift obtained with non-optimized actuation. Optimization of the wing shape subject to the baseline actuation would likely yield improved performance as well; however, this has not been performed here.

The results shown demonstrate that time-dependent adjoint techniques can be successfully applied to the optimization of flow control applications. However, a number of related research efforts hold considerable potential in making such methods more practical for routine use. Local-in-time techniques [53] avoid the need to store the entire forward solution, thereby drastically reducing the amount of disk space required. Higher-order temporal schemes provide an opportunity to use considerably larger physical time steps, ultimately reducing the number of steps required to simulate a given unsteady phenomenon.

A number of enabling strategies are also being pursued from a computer science perspective. Tremendous potential exists for computational improvement within the adjoint implementation, such as more efficient use of memory cache and register re-use. Judicious use of general purpose graphics processing units is also being actively investigated for massive thread parallelism, and ongoing efforts continue to explore the efficient use of massively parallel architectures consisting of $10^4 - 10^5$ processing cores. Finally, whether implemented at the application or system level, fault-tolerant algorithms will become critical as the odds of experiencing a system failure grow with increased levels of parallelism and longer execution times [7].

Acknowledgements

The authors would like to thank Arvin Shmilovich of The Boeing Company for providing the baseline airfoil section. Natalia Alexandrov and Chris Rumsey of the NASA Langley Research Center and Boris Diskin of the National Institute of Aerospace are also acknowledged for their many helpful comments and suggestions.

https://doi.org/10.1051/mmnp/20116306 Published online by Cambridge University Press

References

[1] W.K. Anderson, D.L. Bonhaus. *An implicit upwind algorithm for computing turbulent flows on unstructured grids*. Comp. and Fluids, 23 (1994), No. 1, 1–21.

[2] W.K. Anderson, V. Venkatakrishnan. *Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation*. Comp. and Fluids, 28 (1999), No. 4, 443–480.

[3] W.K. Anderson, D.L. Bonhaus. *Airfoil design on unstructured grids for turbulent flows*. AIAA J., 37 (1999), No. 2, 185–191.

[4] O. Baysal, M. Koklu, N. Erbas. *Design optimization of micro synthetic jet actuator for flow separation control*. J. Fluids Eng., 128 (2006), No. 5, 1053–1062.

[5] T.R. Bewley. *Flow control: new challenges for a new renaissance*. Prog. in Aero. Sci., 37 (2001), No. 1, 21–58.

[6] R.T. Biedron, J.L. Thomas. *Recent enhancements to the FUN3D Flow solver for moving mesh applications*. AIAA 2009-1360 (2009).

[7] F. Cappello, A. Geist, W. Gropp, S. Kale, B. Kramer, M. Snir. *Toward exascale resilience*. Int. J. High Perf. Comp. App., 23 (2009), No. 4, 374–388.

[8] S. Choi, M. Potsdam, K. Lee, G. Iaccarino, J.J. Alonso. *Helicopter rotor design using a time-spectral and adjoint-based method*. AIAA 2008-5810 (2008).

[9] S.S. Collis, R.D. Joslin, A. Seifert, V. Theofilis. *Issues in active flow control: theory, control, simulation, and experiment*. Prog. in Aero. Sci., 40 (2004), No. 4, 237–289.

[10] R. Duvigneau, M. Visonneau. *Optimization of a synthetic jet actuator for aerodynamic stall control*. Comp. and Fluids, 35 (2006), No. 6, 624–638.

[11] D. Greenblatt, I.J. Wygnanski. *The control of flow separation by periodic excitation*. Prog. in Aero. Sci., 36 (2000), No. 7, 487–545.

[12] Z.-H. Han, K.-S. Zhang, W.-P. Song, Z.-D. Qiao. *Optimization of active flow control over an airfoil using a surrogate-management framework*. AIAA J. Aircraft, 47 (2010), No. 2, 603–612.

[13] http://fun3d.larc.nasa.gov, last accessed December 1, 2010.

[14] http://wiki.lustre.org/index.php/Main_Page, last accessed December 1, 2010.

[15] L. Huang, G. Huang, R. LeBeau. *Optimization of airfoil flow control using a genetic algorithm with diversity control*. AIAA J. Aircraft, 44 (2007), No. 4, 1337–1349.

[16] W.T. Jones. *GridEx – an integrated grid generation package for CFD*. AIAA 2003-4129 (2003).

https://doi.org/10.1051/mmnp/20116306 Published online by Cambridge University Press

[17] L. Kaufman, D. Gay. *PORT Library: optimization and mathematical programming – user's manual*. Bell Laboratories, 1997.

[18] W.R. Lanser, L.A. Meyn. *Forebody flow control on a full-scale F/A-18 aircraft*. AIAA J. Aircraft, 31 (1994), No. 6, 1365–1371.

[19] C. Leclerc, E. Levallois, P. Gillieron, A. Kourta. *Aerodynamic drag reduction by synthetic jet: a 2D numerical study around a simplified car*. AIAA 2006-3337 (2006).

[20] E.M. Lee-Rausch, V.N. Vatsa, C.L. Rumsey. *Computational analysis of dual radius circulation control airfoils*. AIAA 2006-3012 (2006).

[21] E.M. Lee-Rausch, D.P. Hammond, E.J. Nielsen, S.Z. Pirzadeh, C.L. Rumsey. *Application of the FUN3D unstructured-grid Navier-Stokes solver to the 4th AIAA Drag Prediction Workshop cases*. AIAA 2010-4551 (2010).

[22] J.N. Lyness. *Numerical algorithms based on the theory of complex variables*. Proc. ACM 22nd Nat. Conf., Thomas Book Co., Washington, D.C. (1967), 124–134.

[23] D.J. Mavriplis. *Solution of the unsteady discrete adjoint for three- dimensional problems on dynamically deforming unstructured meshes*. AIAA 2008-727 (2008).

[24] M. Meunier. *Simulation and optimization of flow control strategies for novel high-lift configurations*. AIAA J., 47 (2009), No. 5, 1145–1157.

[25] F. Muldoon. *Control of a Simplified Unsteady film-cooling flow using gradient-based optimization*. AIAA J., 46 (2008), No. 10, 2443–2458.

[26] S. Nadarajah, A. Jameson. *Optimal control of unsteady flows using time accurate and nonlinear frequency domain methods*. AIAA 2002-5436 (2002).

[27] J.C. Newman III, A.C. Taylor III, R.W. Barnwell, P.A. Newman, G.J.-W. Hou. *Overview of sensitivity analysis and shape optimization for complex aerodynamic configurations*. AIAA J. Aircraft, 36 (1999), No. 1, 87–96.

[28] E.J. Nielsen, B. Diskin, N.K. Yamaleev. *Discrete adjoint-based design optimization of unsteady turbulent flows on dynamic unstructured grids*. AIAA J., 48 (2010), No. 6, 1195–1206.

[29] E.J. Nielsen. *Aerodynamic design sensitivities on an unstructured mesh using the Navier-Stokes equations and a discrete adjoint formulation*. Ph.D. Dissertation, Dept. of Aero. and Ocean Eng., Virg. Poly. Inst. and St. Univ. (1998).

[30] E.J. Nielsen, W.K. Anderson. *Recent improvements in aerodynamic design optimization on unstructured meshes*. AIAA J., 40 (2002), No. 6, 1155–1163.

[31] E.J. Nielsen, W.K. Anderson. *Aerodynamic design optimization on unstructured meshes using the Navier-Stokes equations*. AIAA J., 37 (1999), No. 11, 1411–1419.

https://doi.org/10.1051/mmnp/20116306 Published online by Cambridge University Press

[32] E.J. Nielsen, J. Lu, M.A. Park, D.L. Darmofal. *An Implicit, exact dual adjoint solution method for turbulent flows on unstructured grids*. Comp. and Fluids, 33 (2004), No. 9, 1131–1155.

[33] E.J. Nielsen, W.L. Kleb. *Efficient construction of discrete adjoint operators on unstructured grids by using complex variables*. AIAA J., 44 (2006), No. 4, 827–836.

[34] E.J. Nielsen, M.A. Park. *Using an adjoint approach to eliminate mesh sensitivities in computational design*. AIAA J., 44 (2006), No. 5, 948–953.

[35] M. Nyukhtikov, N. Smelova, B.E. Mitchell, D.G. Holmes. *Optimized dual-time stepping technique for time-accurate Navier-Stokes calculation*. Proceedings of the 10th Int. Sym. on Unst. Aero., Aeroac., and Aeroelas. of Turbomach. (2003).

[36] O.J. Ohanian III, E.D. Karni, W.K. Londenberg, P.A. Gelhausen. *Ducted-fan force and moment control via steady and synthetic jets*. AIAA 2009-3622 (2009).

[37] J.E.V. Peter, R.P. Dwight. *Numerical sensitivity analysis for aerodynamic optimization: A survey of approaches*. Comp. and Fluids, 39 (2010), No. 3, 373–391.

[38] L. Piegl, W. Tiller. The NURBS book (2nd ed.). Springer-Verlag New York, New York, 1997.

[39] S. Pirzadeh. *Three-dimensional unstructured viscous grids by the advancing front method*. AIAA J., 34 (1996), No. 1, 43–49.

[40] P.L. Roe. *Approximate Riemann solvers, parameter vectors, and difference schemes*. J. Comp. Phys., 43 (1981), No. 2, 357–372.

[41] J.L. Rogers. *A parallel approach to optimum actuator selection with a genetic algorithm*. AIAA 2000-4484 (2000).

[42] J.M. Rullan, P.P. Vlachos, D.P. Telionis, M.D. Zeiger. *Post-stall flow control of sharp-edged wings via unsteady blowing*. AIAA J. Aircraft, 43 (2006), No. 6, 1738–1746.

[43] M.P. Rumpfkeil, D.W. Zingg. *A general framework for the optimal control of unsteady flows with applications*. AIAA 2007-1128 (2007).

[44] Y. Saad, M.H. Schultz. *GMRES: A generalized minimal residual algorithm for solving Non-symmetric linear systems*. SIAM J. Sci. and Stat. Comp., 7 (1986), No. 3, 856–869.

[45] J.A. Samareh. *A Novel shape parameterization approach*. NASA TM-1999-209116 (1999).

[46] J.A. Samareh. *Aerodynamic shape optimization based on free-form deformation*. AIAA 2004-4630 (2004).

[47] A. Seifert, S. David, I. Fono, O. Stalnov, I. Dayan. *Roll control via active flow control: From concept to flight*. AIAA J. Aircraft, 47 (2010), No. 3, 864–874.

https://doi.org/10.1051/mmnp/20116306 Published online by Cambridge University Press

[48] A. Shmilovich, Y. Yadlin. *Active flow control for practical high-lift systems*. AIAA J. Aircraft, 46 (2009), No. 4, 1354–1364.

[49] P.R. Spalart, S.R. Allmaras. *A one-equation turbulence model for aerodynamic flows*. La Recherche Aerospatiale, 1 (1994), 5–21.

[50] E. Stanewsky. *Adaptive wing and flow control technology*. Prog. in Aero. Sci., 37 (2001), No. 7, 583–667.

[51] M. Tadjouddine, S.A. Forth, N. Qin. *Automatic differentiation of a time-dependent CFD solver for optimisation of a synthetic jet*. Presented at the Int. Conf. of Num. Anal. and App. Math., Rhodes, Greece (2005).

[52] V.N. Vatsa, M.H. Carpenter, D.P. Lockard. *Re-evaluation of an optimized second order backward difference (BDF2OPT) scheme for unsteady flow applications*. AIAA 2010-0122 (2010).

[53] N. Yamaleev, B. Diskin, E. Nielsen. *Local-in-time adjoint-based method for design optimization of unsteady flows*. J. Comp. Phys., 229 (2010), No. 14, 5394–5407.

https://doi.org/10.1051/mmnp/20116306 Published online by Cambridge University Press