AIAA AVIATION Forum
5-9 June 2017, Denver, Colorado
18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference
10.2514/6.2017-3326

A Discrete Adjoint Framework for Low-Boom Supersonic Aircraft Shape Optimization

Brian C. Munguía*, Thomas D. Economon†, and Juan J. Alonso‡
Stanford University, Stanford, CA 94305, U.S.A.

In this paper, we present a framework for supersonic aerodynamic shape optimization which uses algorithmic differentiation (AD) to obtain design variable sensitivities. We couple the SU2 flow solver to a sonic boom propagation code based on the Thomas algorithm in order to predict the sonic boom signature at the ground. The entire codebase, including the primal solver, mesh movement routines, and boom propagation model are algorithmically differentiated to provide accurate adjoint information to the optimizer. The discrete adjoint framework is applied to sonic boom minimization problems with various constraints. The results indicate that significant improvements in the ground boom signal can be achieved even with the introduction of constraints to the design space.

I. Introduction

The Concorde had two major design flaws which prevented its success: its fuel consumption was significantly larger than competing subsonic options, and its loud boom, with a perceived loudness of 120 PLdB even when cruising at 50,000 ft, caused governments to ban it from flying over land. In order to reintroduce supersonic commercial transport, a large effort under NASA's Commercial Supersonic Technology Project has gone into developing multi-disciplinary analysis and optimization (MDAO) frameworks to design fuel-efficient low-boom aircraft. The goal of NASA's N+2 Supersonic Validations program, for example, is to have a supersonic vehicle which meets certain efficiency, environmental, and loudness constraints enter service in 2018-2020.[1]

Adjoint-based design methods[2] have the advantage of providing gradients with a computational cost independent of the number of design variables. Furthermore, the discrete adjoint method uses algorithmic differentiation (AD) to implement the adjoint equations, removing human error from hand derivation, making the implementation flexible to primal flow solver code updates, and allowing for adjoint solutions which are accurate to machine precision.[3] The discrete framework also allows for adjoint formulations of models that are not analytically differentiable since it only requires that they be algorithmically differentiable, which allows the introduction of arbitrary cost functions to the design problem.

In this paper, we present a AD-based discrete adjoint framework for supersonic aerodynamic shape optimization.

II. Methodology

A. Multi-Physics Flow Solver

The SU2 software suite[5–7] is an open-source collection of software tools written in C++ and Python for performing multi-physics simulation and design. It is built specifically for the analysis of partial differential equations (PDEs) and PDE-constrained optimization problems on unstructured meshes with state-of-the-art numerical methods, and it is particularly well suited for aerodynamic shape design. The initial applications of the suite were mostly in aerodynamics, but through the initiative of users and developers around the

*Ph.D. Candidate, Department of Aeronautics & Astronautics, AIAA Member.
†Postdoctoral Scholar, Department of Aeronautics & Astronautics, AIAA Senior Member.
‡Professor, Department of Aeronautics & Astronautics, AIAA Associate Fellow.

world, SU2 is now being used for a wide variety of problems beyond aeronautics, including automotive, naval, and renewable energy applications, to name a few.

1. Governing equations

In this work, We are concerned with compressible, invsicid fluid flows governed by the Euler equations, which can be expressed in differential form as

$$\begin{cases} \mathcal{R}(U) = \frac{\partial U}{\partial t} + \nabla \cdot \vec{F}^c = 0 & \text{in } \Omega \\ \vec{v} \cdot \vec{n} = \vec{0} & \text{on } S \\ (W)_+ = W_\infty & \text{on } \Gamma_\infty, \end{cases} \tag{1}$$

where the conservative variables are given by $U = \{\rho, \rho\vec{v}, \rho E\}^\mathsf{T}$ and the convective fluxes are

$$\vec{F}^c = \begin{Bmatrix} \rho\vec{v} \\ \rho\vec{v} \otimes \vec{v} + \bar{\bar{I}}p \\ \rho E\vec{v} + p\vec{v} \end{Bmatrix}, \tag{2}$$

where ρ is the fluid density, $\vec{v} = \{v_1, v_2, v_3\}^\mathsf{T} \in \mathbb{R}^3$ is the flow speed in a Cartesian system of reference, E is the total energy per unit mass, and p is the static pressure. Assuming a perfect gas with a ratio of specific heats γ and gas constant R, one can determine the pressure from $p = (\gamma - 1)\rho\left[E - 0.5(\vec{v} \cdot \vec{v})\right]$, the temperature is given by $T = p/(\rho R)$, and $c_p = \gamma R/(\gamma - 1)$.

In 3D, the first line of Eqn. (1) is a set of five coupled, nonlinear PDEs that are statements of mass (1), momentum (3), and energy (1) conservation in a fluid. The remaining lines of Eqn. (1) represent the flow-tangency boundary condition on aerodynamic surfaces S and a typical far-field boundary condition Γ_∞ that mimics the fluid behavior at infinity.[8]

2. Spatial integration

In SU2, both finite volume and finite element discretizations are available, but in this work, we focus on second-order finite volume schemes. The finite volume method (FVM)[4,9–11,14,15,18,21] is applied on the unstructured meshes in SU2 using a standard edge-based data structure on a dual grid with control volumes constructed using a median-dual, vertex-based scheme. Median-dual control volumes are formed by connecting the centroids, face, and edge midpoints of all primal cells sharing the particular vertex.

After integrating the governing equations over a control volume and applying the divergence theorem, one obtains the the semi-discretized, integral form:

$$\begin{aligned} 0 &= \int_{\Omega_i} \frac{\partial U}{\partial t}\, d\Omega + \sum_{j \in \mathcal{N}(i)} \tilde{F}^c_{ij}\, \Delta S_{ij} \\ &= \int_{\Omega_i} \frac{\partial U}{\partial t}\, d\Omega + R_i(U), \end{aligned} \tag{3}$$

where $R_i(U)$ is the numerical residual that represents the integration of all spatial terms for the control volume surrounding vertex i. $\tilde{F}^c_{ij}$ is the numerical approximation of the convective fluxes projected along an edge. ΔS_{ij} is the area of the face associated with the edge ij, $|\Omega_i|$ is the volume of the dual control volume, and $\mathcal{N}(i)$ is the set of neighboring vertices to vertex i.

The convective fluxes are evaluated at the midpoint of an edge. They can be discretized using centered or upwind schemes in SU2. Typical choices are the Jameson-Schmidt-Turkel (JST) scheme,[12] or the approximate Riemann solver of Roe,[13] which will be discussed in detail below. Second-order reconstruction for upwind methods is achieved via the MUSCL approach.[19] Slope limiting is applied to preserve monotonicity in the solution by limiting the gradients during higher-order reconstruction, and the Venkatakrishnan[20] limiter is a common choice for this. Source terms are approximated at each vertex using piece-wise constant reconstruction within each of the dual control volumes.

In practice, the numerical residual $R_i(U)$ at each vertex from Eqn. 3 is evaluated with each nonlinear iteration using a sequence of loops over the edges and vertices. This series of steps results in a value of $R_i(U)$ at each vertex at a particular instance in time, which is then be substituted into Eqn. 3 and integrated in time to arrive at either a steady state or a time-accurate solution for the state vector U.

3. Time integration

We now consider the techniques for time-marching the coupled system of ordinary differential equations (ODEs) for the flow problem presented in Eqn. 3, which can be rewritten as

$$\frac{d}{dt}\left(|\Omega_i|U_i\right) + R_i(U) = 0, \tag{4}$$

where $|\Omega_i| = \int_{\Omega_i(t)} d\Omega$. Here, we are assuming that there is no dynamic mesh motion, i.e., change in the control volumes, during time integration. By discretizing the time derivative term, one obtains a fully-discrete finite volume form of the governing equations.

For particularly stiff problems, implicit methods can be used to improve convergence due to their increased numerical stability. Here, we use the backward Euler scheme, where the residual is evaluated using the solution state at the new time level U^{n+1}. Applying this to Eqn. (4), one has

$$|\Omega_i|\frac{\Delta U_i}{\Delta t_i} = -R_i(U^{n+1}), \tag{5}$$

where time level n corresponds to the known solution in its current state, while time level $n+1$ represents the new solution state that is being sought after advancing one time step Δt where $\Delta t = t^{n+1} - t^n$ and $\Delta U_i = U_i^{n+1} - U_i^n$. However, the residuals at time level $n+1$ are now a function of the unknown solution state U^{n+1} and can not be directly computed. Therefore, a first-order linearization about time level n is performed, and we find that the following linear system should be solved to find the solution update (ΔU_i^n):

$$\left(\frac{|\Omega_i|}{\Delta t_i^n}\delta_{ij} + \frac{\partial R_i(U^n)}{\partial U_j}\right) \cdot \Delta U_j^n = -R_i(U^n). \tag{6}$$

Implicit methods enable the use of higher CFL conditions than with explicit methods, which translate to the specific values of Δt_i that are used to relax the problem. For steady problems, a constant time step for all cells is not required, and a local time-stepping technique can be used to accelerate convergence to a steady state. Allowable local time-step values can be calculated from an estimation of the convective spectral radii at every vertex in the mesh.[16] The resulting linear systems are typically solved using a preconditioned Generalized Minimal Residual (GMRES) method.[17]

B. Sonic Boom Propagation

Sonic boom propagation can be approached from an aerodynamic or acoustics point of view, both of which are illustrated in Figure 1. Due to the computational cost of extending a CFD domain to the ground, sonic boom loudness is typically obtained through propagation of a near-field pressure signature to the ground via acoustic ray tracing. The signature must be obtained sufficiently far away from the aircraft such that strong nonlinear effects have died down, but still close enough so that dissipation from the CFD solver does not artificially decrease the ground boom strength.[22] Several codes currently exist to propagate these signals, including the Thomas code,[23] PCBoom,[24] and sBOOM.[25]

This work utilizes a port of the SUBoom[27] code in C++, which is based on the Thomas algorithm and will be described below.

1. Acoustic ray tracing

In order to propagate the pressure signal to the ground, acoustic rays must first be computed from the signal source to the ground plane. Assuming that the wavelength of the pressure disturbance is much smaller than the propagation distance, geometrical acoustics can be used to propagate waves through a non-uniform medium.[26] The following system of ordinary differential equations can be solved to define the acoustic ray paths in a ground-fixed coordinate system:

$$\frac{dx}{dz} = \frac{cos(\phi)sin(\nu)}{sin(\phi)} - \frac{u_x}{a_\infty sin(\phi)} \tag{7}$$

$$\frac{dy}{dz} = \frac{cos(\phi)cos(\nu)}{sin(\phi)} - \frac{u_y}{a_\infty sin(\phi)} \tag{8}$$

$$\frac{dt}{dz} = \frac{1}{a_\infty sin(\phi)} \tag{9}$$

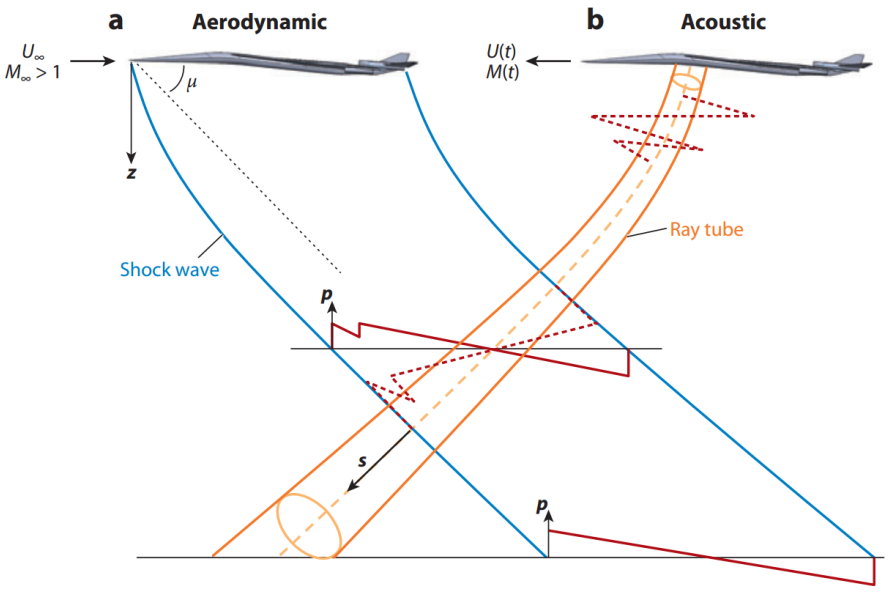


Figure 1: Sonic boom propagation from aerodynamic and acoustics point of view.[22]

where x, y, and z are the ray tube coordinates (with z defined positive downward), ϕ is the wave inclination angle, ν is the wave heading angle, u is the wind speed at z, and a_∞ is the freestream speed of sound at z.

The pressure signal is affected by both the ray tube paths and the changing ray tube area. Ray tube areas can be computed by integrating the paths from the corners of a small ray tube starting at the signal source then computing the area normal to the path.[27]

2. *Signal propagation*

After computing the rays, the near-field pressure signal can be propagated to the ground. Once again following the Thomas algorithm,[23] a pressure signal $p(T)$ can be defined as a series of linear segments parameterized by m_i, Δp_i, and λ_i, where m_i is the slope of segment i, Δp_i is the jump in pressure between segments i and $i-1$, and λ_i is the time duration of segment i, as illustrated in Figure 2.

The signal propagation is described by the following ODEs for each segment:

$$\frac{dm_i}{dt} = C_1 m_i^2 + C_2 m_i \tag{10}$$

$$\frac{d\Delta p_i}{dt} = \frac{C_1}{2}\Delta p_i(m_i + m_{i-1}) - C_2 \Delta p_i \tag{11}$$

$$\frac{d\lambda_i}{dt} = -\frac{C_1}{2}(\Delta p_i + \Delta p_{i-1}) - C_2 m_i \lambda_i \tag{12}$$

where

$$C_1 = \left(\frac{\gamma+1}{2\gamma}\right)\frac{a_\infty}{p_\infty c_n} \tag{13}$$

$$C_2 = \frac{1}{2}\left(\frac{3}{a_\infty}\frac{da_\infty}{dt} + \frac{1}{p_\infty}\frac{dp_\infty}{dt} - \frac{2}{c_n}\frac{dc_n}{dt} - \frac{1}{A}\frac{dA}{dt}\right) \tag{14}$$

where c_n is the acoustic propagation speed normal to the wavefront and A is the ray tube area. Note that $p(T)$ is a pressure disturbance and is taken to be relative to the freestream. Once a near-field pressure signal is obtained from CFD, it can be decomposed into N segments then propagated to any altitude by integrating the $3N$ ODEs.

Segments for which $\lambda_i \to 0$ must be removed to ensure numerical stability. Some segments will coalesce with others and we can simply add their pressure rises together once λ_i is below a certain tolerance. However,

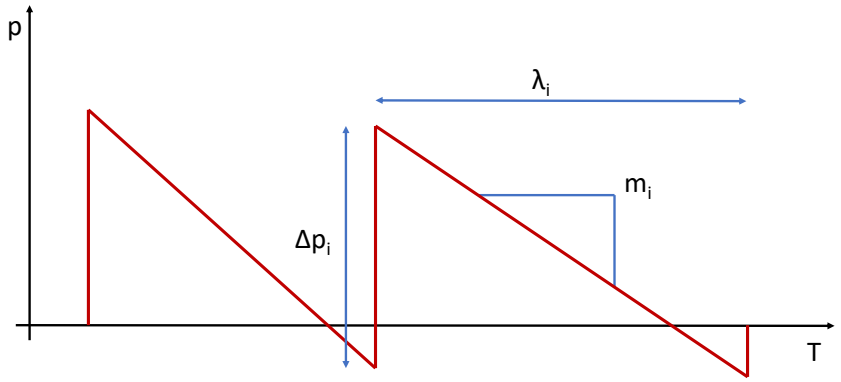


Figure 2: Piecewise linear pressure signal parameterization.

m_i can become unbounded for segments with a large, positive m_i and positive C_1 and C_2. Therefore, we must also capture new shocks when m_i is above a certain tolerance to ensure accurate results. Finally, we must take care of any expansion discontinuities present in the initial signal. A segment sith $m_i < 0$ will flatten over time, but a segment with $m_i \to -\infty$ will result in a a numerically unstable system. Therefore, we must relax any initial slopes which are too negative.

While there are various possible signature parameters that can be used as a cost function, it is known that outdoor noise and physical damage to infrastructure are primarily functions of impulse (integrated pressure) and overpressure (maximum pressure in the entire signal).[28] In this paper, we use the integrated squared pressure signal, or

$$J_B = \int_0^\tau p(t)^2 dt, \tag{15}$$

as our cost function. We use the squared pressure to take into account positive and negative pressure disturbances, giving us a metric for the total acoustical energy contained in the boom. However, our framework is agnostic to the choice of propagation algorithm and cost function since it relies on AD to compute sensitivities, so we could easily utilize the framework with a more complicated cost function.

C. AD-based Discrete Adjoint Solver

Thanks to the open-source community, the SU2 CFD code contains two distinctly different implementations of the adjoint methodology that have been completed by experts in the field: a continuous adjoint by Stanford University and a discrete adjoint via algorithmic differentiation (AD) of the entire solver by the Technical University of Kaiserslautern. This work focuses on the discrete adjoint framework.

The current implementation of the discrete adjoint solver in SU2 has been generated by algorithmically differentiating the entire codebase. To be more precise, this includes not only the flux routines, but also the complete nonlinear iteration structure of the solver. To improve efficiency, the group at the Technical University of Kaiserslautern has developed a custom set of AD tools specifically tuned for CFD applications. Currently, the discrete adjoint solver has been validated for 2D Euler (steady[30] and unsteady[3]), 2D RANS (including the SA or SST turbulence models for steady[30,31] as well as unsteady[3] flows), and 3D Euler[29] in combination with all of the existing design parameterizations available in SU2.

We are interested in the optimization problem

$$\begin{aligned} \underset{\alpha}{\text{minimize}} \quad & J(U(\alpha), X(\alpha)) \\ \text{subject to} \quad & R(U(\alpha), X(\alpha)) = 0 \end{aligned} \tag{16}$$

Consider a converged solution U^*. If we define our fixed point equation as

$$U^{n+1} = G(U^n) \tag{17}$$

then we can assume that G is stationary at the solution, i.e.

$$R(U^*) = 0 \Leftrightarrow U^* = G(U^*) \tag{18}$$

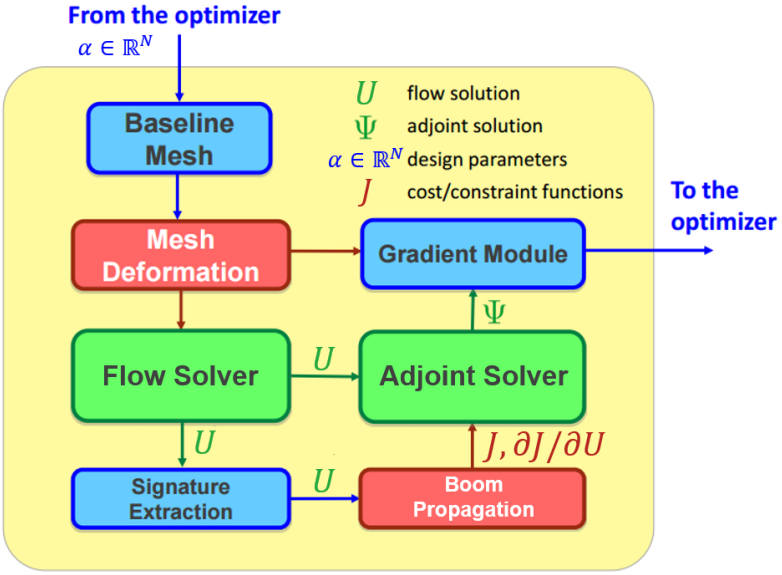


Figure 3: Low-boom supersonic aerodynamic shape optimization framework.

Following the derivation of Zhou et al. for aeroacoustic optimization[3] and omitting the influence of the mesh, our optimization problem can be rewritten as

$$\begin{aligned} \underset{\alpha}{\text{minimize}} \quad & J(U, \alpha) \\ \text{subject to} \quad & U(\alpha) = G(U, \alpha) \end{aligned} \tag{19}$$

Now consider the Lagrangian

$$L(U, \psi, \alpha) = J(U, \alpha) + \psi^T (G(U, \alpha) - U) \tag{20}$$

We differentiate L with respect to α to obtain our sensitivities. We can choose ψ such that we can eliminate the term $\frac{\partial U}{\partial \alpha}$, giving us

$$\psi = \left(\frac{\partial J}{\partial U}\right)^T + \left(\frac{\partial G}{\partial U}\right)^T \psi \tag{21}$$

Equation 21 is a fixed-point equation in ψ and once a numerical solution U^* is obtained, can be solved similar to the primal problem with the iteration

$$\psi^{n+1} = \frac{\partial L}{\partial U}(U^*, \psi^n) \tag{22}$$

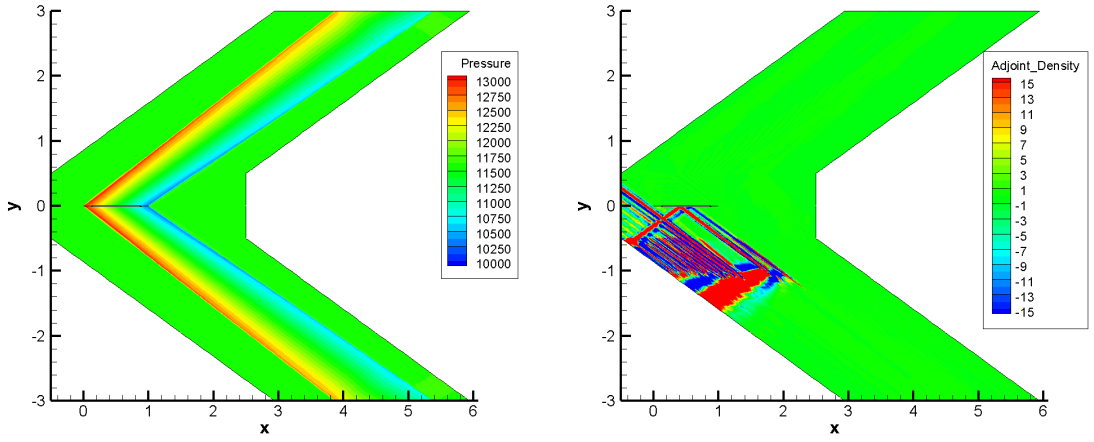
As shown by Albring et al.,[32] this fixed-point equation inherits the convergence properties of the primal solver. Once an adjoint solution is obtained, the sensitivity can be computed from

$$\frac{\partial L}{\partial \alpha} = \frac{\partial J}{\partial \alpha} + \psi^T \frac{\partial G}{\partial \alpha} \tag{23}$$

In this work, we build on the existing tools present in SU2 by adding a discrete adjoint framework for sonic boom loudness, depicted in Figure 3. After obtaining a CFD solution, a pressure signature is extracted from a plane below the aircraft and propagated to the ground. The sonic boom post-processing routine is differentiated using the AD tools in SU2 to obtain values of $\frac{\partial J}{\partial U}$ at the extraction plane which are treated as a source term for the discrete adjoint solver.

III. Results

Here we present results for a flow over a biparabolic airfoil with thickness-to-chord ratio of 0.02 at $M = 1.7$ and an altitude of 15240 m. Primal and adjoint results for the baseline airfoil geometry are presented in



(a) Pressure contours **(b)** Boom adjoint density

Figure 4: Biparabolic airfoil flow and adjoint solutions

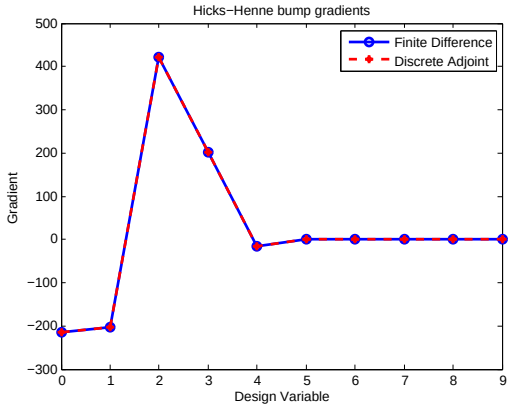


Figure 5: Gradient comparison for finite difference and reverse-mode AD

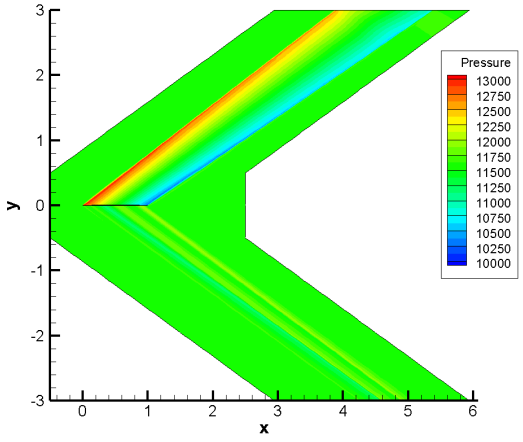
Figure 4. The airfoil has a drag coefficient of $C_d = 0.001552$. In this test case, the near-field pressure signature was extracted 1 m below the airfoil, which can be seen in the adjoint density contours where the extraction plane is acting as a source. The adjoint density contours show that the solution is dependent on perturbations at the forward facing Mach cone. Since the airfoil is thin, the solution we obtain is close to linear supersonic theory, so we would expect the adjoint solution to propagate along upstream-running characteristics.

The gradients obtained from the discrete adjoint results were verified against finite difference results for a biparabolic airfoil parameterized by Hicks-Henne bump functions located at 0.15c, 0.3c, 0.5c, 0.7c, and 0.85c on both lower and upper surfaces. Figure 5 shows that the gradients match well. In the verification test case, design variables 0-4 defined the lower surface while variables 5-9 defined the upper surface. As we would expect, the ground boom strength is significantly more sensitive to the shape of the lower surface than the shape of the upper surface.

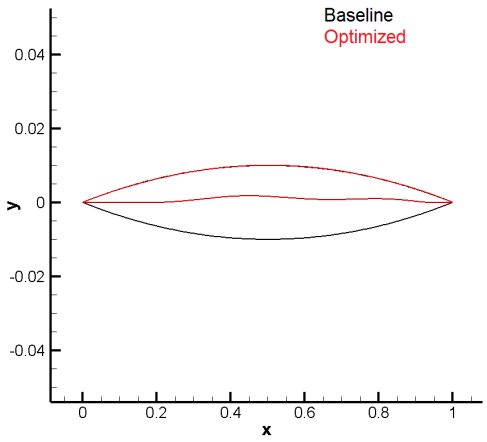
In the following subsections, we present various optimization results.

A. Unconstrained Boom Minimization

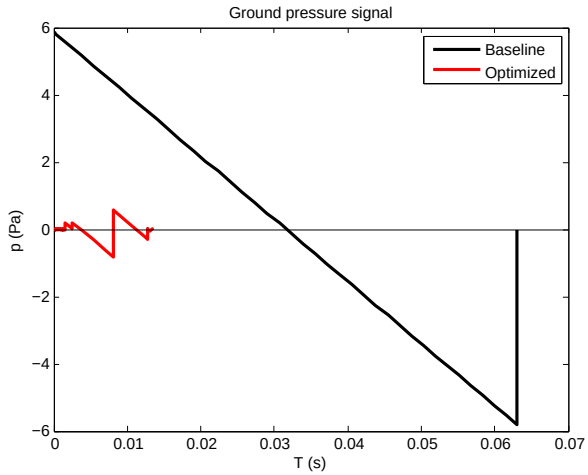
In this test case, we consider unconstrained boom minimization of a biparabolic airfoil with thickness-to-chord ratio of 0.02 at $M = 1.7$. Here we use 2 free-form deformation (FFD) boxes, one for the upper and



(a) Pressure contours for optimized airfoil



(b) Airfoil comparison



(c) Ground pressure signal comparison

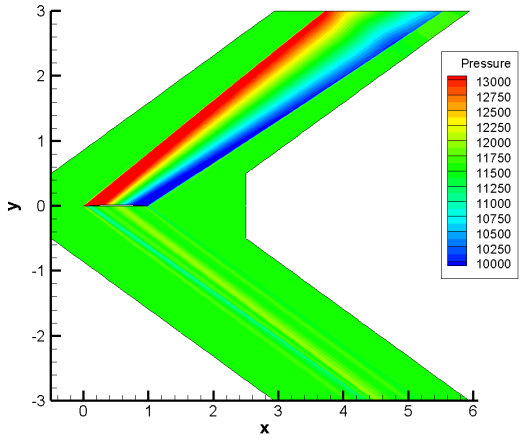
Figure 6: Unconstrained boom optimization results

one for the lower surface, as our design variables. Each FFD box has 20 equally spaced control points in the x-direction and 2 control points in the y-direction. Results for the optimization can be found in Figure 6. The optimized airfoil was obtained after 191 CFD evaluations, including line searches.

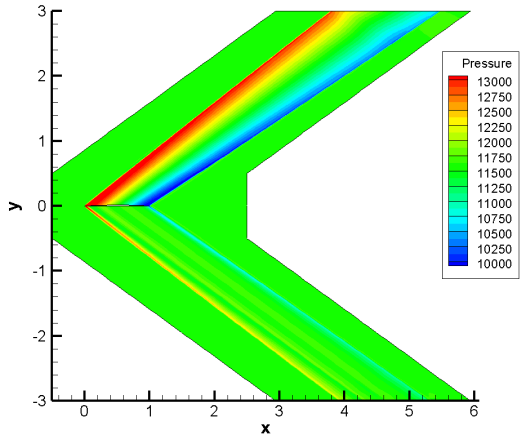
From the pressure contours and airfoil shape comparison, it is clear that the optimizer only modified the lower surface which is what we would expect. We can also see that the modified airfoil shape has significantly altered the ground signature. The new signal now has several shocks, all of which are weaker than the shocks from the baseline signal. The weaker shocks result in less stretching of the signal as it propagates to the ground, resulting in a shorter time duration of the ground signal.

B. Constrained Boom Minimization

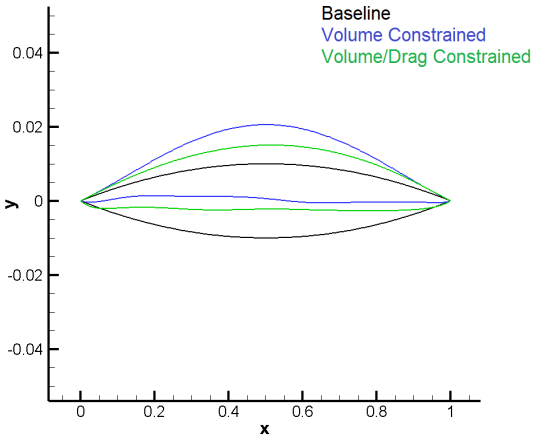
With the new framework giving good results for unconstrained boom optimization, we now focus our attention on constrained optimization problems. For the constrained optimization test cases, we use the same FFD box to define our airfoil shape.



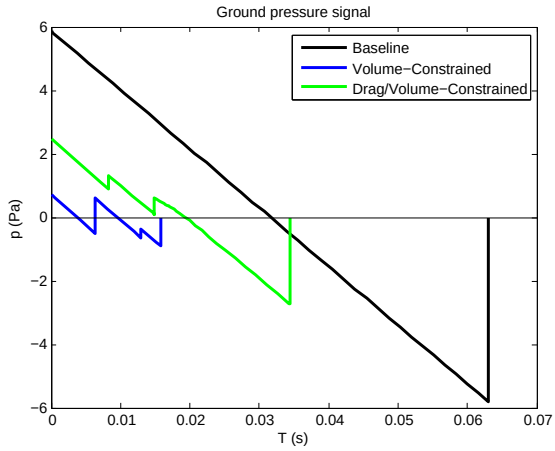
(a) Pressure contours for volume-constrained optimized airfoil



(b) Pressure contours for drag- and volume- constrained optimized airfoil



(c) Airfoil comparison



(d) Ground pressure signal comparison

Figure 7: Constrained boom optimization results

First, we impose a volume constraint on the airfoil. The optimization problem is given by

$$\begin{aligned} \underset{\alpha}{\text{minimize}} \quad & J_B(U(\alpha), X(\alpha)) \\ \text{subject to} \quad & A > 0.012 \end{aligned} \tag{24}$$

where J_B is our boom cost function as described previously. The initial area of the biparabolic airfoil was 0.0133 so the optimized design should maintain roughly 90% of the initial area. The optimizer achieved the design shown in Figure 7. The design is once again tailored to reduce the shock strength of the lower surface. However, in order to meet the volume constraint, the upper surface must also be modified in a way which results in stronger shocks. This causes the drag on the airfoil to increase by a factor of 1.93.

In order to control the drag on the airfoil, another constraint is imposed and the optimization problem becomes

$$\begin{aligned} \underset{\alpha}{\text{minimize}} \quad & J_B(U(\alpha), X(\alpha)) \\ \text{subject to} \quad & A > 0.012 \\ & C_d < 0.002 \end{aligned} \tag{25}$$

Table 1: Optimization cost function comparison

Design	Drag	Boom
Baseline	0.001552	0.7151
Unconstrained	0.000823	0.001636
Volume-Constrained	0.002998	0.003457
Drag/Volume-Constrained	0.001980	0.067651

The resulting airfoil is shown in Figure 7. For this new design problem, the optimizer is less aggressive with the lower surface, so the upper surface is no longer forced to accommodate as large of an area change.

Comparing the ground signatures of both new designs, we see that in a similar manner to the unconstrained boom optimized airfoil, the new pressure signatures have more, weaker shocks than the baseline airfoil and a shorter duration. The drag coefficient and boom cost function for the baseline and the constrained designs are presented in Table 1. We can see that even with the addition of a drag constraint, the new ground signature is significantly improved compared to the baseline biparabolic airfoil. Therefore, we can use this design framework to shape the boom without seriously degrading performance.

IV. Conclusions and Future Work

In this paper we presented a discrete adjoint framework for low-boom supersonic aerodynamic shape optimization. A sonic boom propagation code based on the Thomas algorithm was coupled with the SU2 software suite, and a cost function based on the integrated ground pressure signal was introduced. The SU2 codebase, including the new boom propagation routine, was algorithmically differentiated to obtain numerically exact adjoint information.

The new framework was applied to three test cases: an unconstrained boom optimization, a volume-constrained boom optimization, and a volume- and drag- constrained boom optimization. All three optimization problems yielded intuitive results, and it was shown that the ground signature can be greatly improved without serious loss to performance.

This framework is in the process of being extended to full 3D aircraft optimization problems in the near future. For 3D problems, the entire boom carpet must be accounted for in the cost function, so the sonic boom propagation routine will be modified to propagate signals at various azimuths. The framework will then be built upon to allow for error estimation for goal-oriented mesh adaptation. Goal-oriented anisotropic mesh adaptation has already been implemented in SU2 using Inria's AMG.[33–35] These capabilities will be extended to mesh adaptation in the loop of optimization

V. Acknowledgements

The authors would like to thank the SU2 development team for providing code they could use as a foundation, with special thanks to Professor Nicolas Gauger, Tim Albring, and Becket Zhou for their work on the AD tools. Brian Munguía would like to acknowledge the support of the Department of Defense (DoD) through the National Defense Science & Engineering Graduate Fellowship (NDSEG) Program.

References

[1]Morgenstern, J. , et al., "Advanced Concept Studies for Supersonic Commercial Transports Entering Service in the 2018-2020 Period Phase 2," *NASA NTRS*, NASA/CR-2015-218719, July 2015.

[2]A. Jameson. "Aerodynamic design via control theory," *Journal of Scientific Computing*, Vol. 3, No. 3, pp. 233260, 1988.

[3]B.Y. Zhou, T. Albring, N.R. Gauger, T. D. Economon, F. Palacios, and J. J. Alonso, "A discrete adjoint framework for unsteady aerodynamic and aeroacoustic optimization,".*AIAA Aviation Forum 2015*, Dallas, TX, June 2015.

[4]T. J. Barth. Aspect of unstructured grids and finite-volume solvers for the Euler and Navier-Stokes equations. In *Lecture Notes Presented at the VKI Lecture Series*, 1994 - 05, Rhode Saint Genese Begium, 2 1995. Von karman Institute for fluid dynamics.

[5]Francisco Palacios, Michael R Colonno, Aniket C Aranake, Alejandro Campos, Sean R Copeland, Thomas D Economon,

Amrita K Lonkar, Trent W Lukaczyk, Thomas WR Taylor, and Juan J Alonso. Stanford university unstructured (su2): An open-source integrated computational environment for multi-physics simulation and design. *AIAA Paper*, 287:2013, 2013.

[6]Francisco Palacios, Thomas D Economon, Aniket C Aranake, Sean R Copeland, Amrita K Lonkar, Trent W Lukaczyk, David E Manosalvas, Kedar R Naik, A Santiago Padrón, Brendan Tracey, et al. Stanford university unstructured (su2): Open-source analysis and design technology for turbulent flows. *AIAA paper*, 243:13–17, 2014.

[7]Thomas D. Economon, Francisco Palacios, Sean R. Copeland, Trent W. Lukaczyk, and Juan J. Alonso. Su2: An open-source suite for multiphysics simulation and design. *AIAA Journal*, 54(3):828–846, 2015.

[8]C. Hirsch. *Numerical Computation of Internal and External Flows*. Wiley, New York, 1984.

[9]A. Jameson. Analysis and design of numerical schemes for gas dynamics 1 artificial diffusion, upwind biasing, limiters and their effect on accuracy and multigrid convergence. *RIACS Technical Report 94.15, International Journal of Computational Fluid Dynamics*, 4:171–218, 1995.

[10]A. Jameson. Analysis and design of numerical schemes for gas dynamics 2 artificial diffusion and discrete shock structure. *RIACS Report No. 94.16, International Journal of Computational Fluid Dynamics*, 5:1–38, 1995.

[11]A. Jameson. A perspective on computational algorithms for aerodynamic analysis and design. *Progress in Aerospace Sciences*, 37:197–243, 2001.

[12]A. Jameson, W. Schmidt, and E. Turkel. Numerical solution of the Euler equations by finite volume methods using Runge-Kutta time stepping schemes. *AIAA Paper 1981-1259*, 1981.

[13]Roe, Philip L., "Approximate Riemann solvers, parameter vectors, and difference schemes," *Journal of Computational Physics*, Vol. 43, No. 2, Elsevier, 1981, pp. 357–372.

[14]R. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge Univesity Press, 2002.

[15]A. Quarteroni and A. Valli. *Numerical approximation of partial differential equations*, volume 23 of *Springer series in computational mathematics*. Springer-Verlag Berlin Heidelberg New York, 1997.

[16]P. Eliasson. Edge, a Navier-Stokes solver for unstructured grids. Technical Report FOI-R-0298-SE, FOI Scientific Report, 2002.

[17]Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7:856–869, 1986.

[18]E. F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics: a Practical Introduction*. Springer-Verlag, 1999.

[19]B. van Leer. Towards the ultimate conservative difference scheme V. a second-order sequel to Godunov's method. *Journal of Computational Physics*, 32(1):101–136, July 1979.

[20]V. Venkatakrishnan. On the accuracy of limiters and convergence to steady state solutions. *AIAA Paper 1993-0880*, 1993.

[21]P. Wesseling. *Principles of Computational Fluid Dynamics*, volume 29 of *Springer Series in Computational Mathematics*. Springer-Verlag Berlin Heidelberg New York, 2000.

[22]Alonso, J. J. and Colonno, M. , "Mutidisciplinary Optimization with Applications to Sonic-Boom Minimization," *Annual Review of Fluid Mechanics*, Vol. 44, 2012, pp. 505–526.

[23]Thomas, C. L. , "Extrapolation of Sonic Boom Pressure Signatures by the Waveform Parameter Method," NASA TN D-6832, June 1972.

[24]Plotkin, K. J. , "PCBoom3 Sonic Boom Prediction Model-Version 1.0e", Wyle Research Report WR 95-22E, Oct. 1998.

[25]Rallabhandi, S. K. , "Advanced Sonic Boom Prediction Using Augmented Burger's Equation," *Journal of Aircraft*, Vol. 48, No. 4, 2011, pp. 1245–1253.

[26]D. Blokhintzev, "The Propagation of Sound in an Inhomogeneous and Moving Medium I," *The Journal of the Acoustical Society of America*, Vol. 18, No. 2, 1946, pp. 322-328.

[27]M.R. Colonno and J.J. Alonso, "Sonic Boom Minimization Revisited: The Robustness of Optimal Low-Boom Designs," *13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference*, Fort Worth, TX, Sept. 2010.

[28]Seebass, R. , "Sonic Boom Theory," *Journal of Aircraft*, Vol. 6, 1969, pp. 177–184.

[29]T. Albring, M. Sagebaum, and N.R. Gauger. "A consistent and robust discrete adjoint solver for the Stanford University Unstructured (SU2) framework, validation and application," *Notes on Numerical Fluid Mechanics*, 2015.

[30]T. Albring, B.Y. Zhou, N.R. Gauger, and M. Sagebaum, "An aerodynamic design framework based on algorithmic differentiation," *Ercoftac Bulletin*, 102 pp. 10-16, 2015.

[31]T. Albring, M. Sagebaum, and N.R. Gauger, "Development of a consistent discrete adjoint solver in an evolving aerodynamic design framework," *AIAA Aviation Forum 2015*, Dallas, TX, June 2015.

[32]T. Albring, M. Sagebaum, and N.R. Gauger, "Efficient Aerodynamic Design using the Discrete Adjoint Method in SU2," *AIAA Aviation Forum 2016*, Washington, D.C., June 2016.

[33]Adrien Loseille and Rainald Löhner, "Boundary Layer Mesh Generation and Adaptivity," *AIAA Paper*, No. AIAA-2011–894, 2011.

[34]Adrien Loseille and Victorien Menier, "Serial and Parallel Mesh Modification Through a Unique Cavity-Based Primitive," *22nd International Meshing Roundtable*, Sandia National Lab, 2013.

[35]Adrien Loseille, Victorien Menier, and Frédéric Alauzet, "Parallel generation of large-size adapted meshes," *24th International Meshing Roundtable*, Sandia National Lab, 2015.