DEPARTMENT OF AEROSPACE ENGINEERING
COLLEGE OF ENGINEERING AND TECHNOLOGY
OLD DOMINION UNIVERSITY
NORFOLK, VIRGINIA 23529-0247

EFFICIENT GRADIENT-BASED SHAPE OPTIMIZATION METHODOLOGY USING INVISCID/VISCOUS CFD

By
Dr. Oktay Baysal, Principal Investigator

Summary of Research Report
For the period of March 9, 1995 to March 8, 1997

Prepared for
National Aeronautics and Space Administration
Langley Research Center
Attn: Joseph Murray, Mail Stop 128
Hampton, VA 23681-0001

Grant# NCC-1-211
ODURF #152081

Submitted by the
Old Dominion University Research Foundation
P.O. Box 6369
Norfolk, VA 23508-0369

May 1997

DEPARTMENT OF AEROSPACE ENGINEERING
COLLEGE OF ENGINEERING AND TECHNOLOGY
OLD DOMINION UNIVERSITY
NORFOLK, VIRGINIA 23529-0247

EFFICIENT GRADIENT-BASED SHAPE OPTIMIZATION METHODOLOGY USING INVISCID/VISCOUS CFD

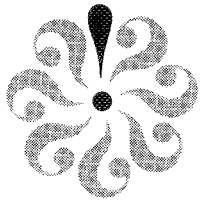
By
Dr. Oktay Baysal, Principal Investigator

Summary of Research Report
For the period of March 9, 1995 to March 8, 1997

Prepared for
National Aeronautics and Space Administration
Langley Research Center
Attn: Joseph Murray, Mail Stop 128
Hampton, VA 23681-0001

Grant# NCC-1-211
ODURF #152081

Submitted by the
Old Dominion University Research Foundation
P.O. Box 6369
Norfolk, VA 23508-0369



May 1997

Final report for NASA Langley Grant NCC-1-211
March 1995 - March 1997

Efficient Gradient-Based Shape Optimization Methodology Using Inviscid/Viscous CFD

Oktay Baysal
Old Dominion University
Aerospace Engineering Department
Norfolk, VA 23529-0247

§.1.0 Summary of Accomplishments

The formerly developed preconditioned-biconjugate-gradient (PbCG) solvers[1,2] for the analysis and the sensitivity equations had resulted in very large error reductions per iteration; quadratic convergence was achieved whenever the solution entered the domain of attraction to the root. Its memory requirement was also lower as compared to a direct inversion solver[3,4]. However, this memory requirement was high enough to preclude the realistic, high grid-density design of a practical 3D geometry. This limitation served as the impetus to the first-year activity (March 9, 1995 to March 8, 1996). Therefore, the major activity for this period was the development of the *low-memory* methodology for the discrete-sensitivity-based shape optimization[5]. This was accomplished by solving all the resulting sets of equations using an alternating-direction-implicit (ADI) approach.

The results indicated that shape optimization problems which required large numbers of grid points could be resolved with a gradient-based approach. Therefore, to better utilize the computational resources, it was recommended that a number of coarse grid cases, using the PbCG method, should initially be conducted to better define the optimization problem and the design space, and obtain an improved initial shape. Subsequently, a fine grid shape optimization, which necessitates using the ADI method, should be conducted to accurately obtain the final optimized shape.

The other activity during this period was the interaction with the members of the Aerodynamic and Aeroacoustic Methods Branch of Langley Research Center during one stage of their investigation to develop an adjoint-variable sensitivity method[6] using the viscous flow equations. This method had algorithmic similarities to the variational sensitivity[7] methods and the control-theory[8] approach. However, unlike the prior studies, it was considered for the three-dimensional, viscous flow equations.

The major accomplishment in the second period of this project (March 9,1996-March 8, 1997) was the extension of the shape optimization methodology for the Thin-Layer Navier-Stokes equations.[9] Both the Euler-based and the TLNS-based analyses compared with the analyses obtained using the CFL3D[10] code. The sensitivities, again from both levels of the flow equations, also compared very well with the finite-differenced sensitivities. A fairly large set of shape optimization cases were conducted to study a number of issues previously not well understood. The testbed for these cases was the shaping of an arrow wing in Mach 2.4 flow. All the final shapes, obtained either from a coarse-grid-based or a fine-grid-based optimization, using either a Euler-based or a TLNS-based analysis, were all re-analyzed using a fine-grid, TLNS solution for their function evaluations. This allowed for a more fair comparison of their relative merits. From

the aerodynamic performance standpoint, the fine-grid TLNS-based optimization produced the best shape, and the fine-grid Euler-based optimization produced the lowest cruise efficiency.[9]

Finally, the details of the findings from this project are reported in References 5, 9, 11 and 12. (See §.5.)

§.2.0 Identification and Features of Methodology

Based on the experiences gained thus far, certain recommendations can be made for the directions to be taken. Hence, the features of the present methodology are given in this section. Most are selected for their virtues, however, some are recommended for the ease of their implementation.

2.1 Flowfield analysis

- governing equations: thin-layer Navier-Stokes equations[19,13-16] (primarily for systems with multiple components in close proximity), with a switch for the choice of dropping the diffusion terms and employing the Euler equations (for isolated configurations), in conservation form and generalized curvilinear coordinates.

- CFD discretization:
  - at least second-order accurate, upwind-biased (van Leer flux-vector split), cell-centered, finite volume discretization in space;
  - either PbCG for the unfactored equations[1,2], or ADI[5,9] with pseudo-time (local time stepping) integration of the factored equations.

- van Albada limiter.

- domain decomposition for structured grids using multiblock[14-17] (MB) concepts (contiguous lines normal to the interfaces).

- vector processing of data; parallel processing with the number of grid blocks assigned to each processor determined by a load-balancing strategy.

2.2 Parameterization

- a general and easily differentiable surface parameterization[18] (in addition to surface grid points or shape-specific functions) for flexibility and to reduce the number of design variables;
  - for the definition and redefinition of the surfaces to be shape optimized, employ *N*th-degree Bezier-Bernstein polynomials.

- use the same approach, but with lower-order polynomials, to curve-fit the various shape schedules.[19]

- using a partial-differential-equation-based approach[20,12] may prove to be a viable alternative where the parameters used are more appealing to an aerodynamicist's intuition.

2.3 Optimization algorithm

- gradient-based and constrained optimization algorithm, such as, the feasible-directions method.[21]

- within the feasible region (no active constraints): deterministic search algorithms, such as, a univariate search strategy (sequential one-dimensional minimization in a multidimensional design space) with variable (using zeroth-order methods) alpha-step-size, or simply a constant-alpha-step-size (although less efficient for linear regions, better success rate for nonlinear regions).

- various options (user-changeable) of stopping criteria used for the optimization and for the flow analysis (either the convergence tolerances of different orders of magnitude, or the number of iterations, or both).

- for shape optimization, Bezier control points and other shape scales[18,19] (taper, thickness, twist, cant, etc.) as the design variables (as opposed to the relative slopes at the surface grid points) to reduce their numbers.

- allow for geometric as well as aerodynamic constraints[5,18].

2.4 Sensitivities

- to obtain the sensitivity coefficients (gradients) implement both the direct and adjoint-variable methods[3,4] as one may be advantageous over the other depending on the problem at hand (number of design variables versus the number of aerodynamic constraints).

- consistent differentiation[22,23] of the CFD-discretized equations to obtain all the necessary terms in the sensitivity equation and the equations rendering the gradients.

- writing the sensitivity equation for a multiblock grid with strong coupling of the sensitivities across the block interfaces; that is, employing the SADD[2,14,17] scheme.

- a desirable capability is the representation of the Reynolds stresses by an algebraic model in the flow analysis equations; however, consistent differentiation of the currently available models (empirical with tunable parameters) for the sensitivities needs further studying; an approximate approach may be the automatic differentiation[24,12] of the turbulence modeling terms.[23]

2.5 Solvers for algebraic systems

- to solve the large set of linear algebraic equations resulting from the sensitivity equation (or the adjoint-variable equations), have several inversion options in, both direct and iterative approaches as they all have their distinct niches depending on the problem in hand:
  - naive Gauss elimination with banded storage;[3,4]
  - sparse-matrix method for symbolic factorization;[3,4]
  - a first-order iterative method (such as, solving the delta-form of the equations by an ADI method,[5,9] also known as the incremental technique[23]);
  - a biconjugate-gradient-like method [such as, the restarting version of GMRES(k,m)][1,2] with several fill-in options for the preconditioners.

2.6 Programming

- optimizer as the outer `do-loop`.

- option for `out-of-core` storage.

- judiciously segregating the case-dependent routines, which should allow for ease in setting the code up for different application problems.

§.3.0 Utilization of Available Methodologies for Present Objectives

In order to achieve the goals of this investigation in a rather timely and cost-effective manner, the methodologies (and their computer codes) that were readily available and, in some cases, even reasonably validated, were utilized. This was done in a number of ways ranging from a "black-box" utilization to simply using as an illustrative example.

The following methodologies, given with their acronyms, were considered in this category: ADS,[21] ADOS,[14] AeSOP,[1,5,9,19] ADIFOR,[24] and RAPID.[20] It should be noted, however, although these methodologies may be somewhat general, some have their computer codes written for specific cases and they are intended to be "research codes."

ADOS. The three dimensional, aerodynamic shape optimization method, which include discrete "sensitivity analysis on decomposed computational domains" (SADD), was developed to handle complex configurations with multiple components. By *substructuring* the large Jacobian matrix ($\partial R/\partial Q$), which results from the sensitivity equation for large size problems, this method utilizes relatively less computer memory than the single-grid approaches. It employs the general purpose, multiblock CFD code CFL3D,[10] to perform the flow analyses.

AeSOP. This "third generation" methodology is rather efficient in computer resource usage. It utilizes a PbCG solver[1,19], but it has been reinvestigated for reductions in computer storage per grid point.[5,9] Some of the iterative methods being tried sought ways of solving the sensitivity equation without having to form the Jacobian matrix on the left-hand side, but rather on the explicit right-hand side, that is the ADI[5,9] approach. It solves both the Euler and TLNS equations.

ADS. This is a general purpose package of optimization methods with its genesis being for the structural mechanics.[21] Nonetheless, it may be modified to handle the aerodynamic problems. One of the main issues that often arise as a result of this switch in its application, is the limited suitability of its search strategies for the largely nonlinear design spaces encountered in solving the aerodynamic shape optimization problems. It has, however, all the capabilities desired in §2.3.

ADIFOR. In 1992, a mathematical utility, ADIFOR,[24] was developed and successfully demonstrated to automatically obtain the sensitivity coefficients from an existing CFD code. The output was also in the form of a computer code. One of the recent demonstrations used the code RAPID,[20] (interactive/batch version) which parameterized the geometry of a general aircraft configuration, generated its surface grid, and finally generated its grid sensitivities.[12] Some recent research has also shown promise in eliminating the extra differential terms obtained in the process. (The author and his graduate students had unsuccessfully tried to use symbolic differentiation available in the commercially available mathematical packages, such as, MATHEMATICA™ and MACSYMA™, to perform the differentiation needed to obtain the sensitivities. Some of the reasons for the failure are explained in Ref. 12.)

§.4.0 References

1. Burgreen, G.W., Baysal, O., "Aerodynamic Shape Optimization Using Preconditioned Conjugate Gradient Methods," *AIAA Journal*, Vol. 32, No. 11, Nov. 1994, pp. 2145-2152.

2. Eleshaky, M.E., and Baysal, O., "Preconditioned Domain Decomposition Scheme for 3-D Aerodynamic Sensitivity Analysis," *AIAA Journal* , Vol. 32, No. 12, Dec. 1994.

3. Baysal, O., and Eleshaky, M.E., "Aerodynamic Sensitivity Analysis Methods For the Compressible Euler Equations," *Journal of Fluids Engineering*, Vol. 113, No. 4, Dec. 1991, pp. 681-688.

4. Baysal, O., and Eleshaky, M.E., "Aerodynamic Design Optimization Using Sensitivity Analysis and Computational Fluid Dynamics," *AIAA Journal*, Vol. 30, No. 3, Mar. 1992, pp. 718–725.

5. Pandya, M.J., and Baysal, O., "Gradient-Based Aerodynamic Shape Optimization Using ADI Method for Large-Scale Problems," Paper No. 96-0091, AIAA 34th Aerospace Sciences Meeting, Reno, NV, Jan. 1996. To appear in *Journal of Aircraft*, May 1997. **(partial support under NCC-1-211)**

6. DeRise, George, "Viscous Aerodynamic Design using Adjoint Variable Approach - a Methodology," ASEE Summer Faculty Fellowship Lecture, Fluid Mechanics and Acoustics Division, NASA Langley Research Center, August 1995.

7. Ibrahim, A.H., and Baysal, O., "Design Optimization Using Variational Methods and CFD," AIAA Paper 94-0093, 32nd Aerospace Sciences Meeting, Reno, NV, Jan. 1994.

8. Reuther, J. "Practical Aspects of Aerodynamic Shape Optimization using Adjoint Based Sensitivity Derivatives," a Colloquium to The Aerodynamic and Aeroacoustic Methods Branch, NASA Langley Research Center, May 1995.

9. Pandya, M.J., and Baysal, O., "3D Viscous ADI Algorithm and Strategies for Shape Optimization," Paper No. 97-1853 CP, *Proceedings of AIAA 13th CFD Conference*, Snowmass, CO, June 1997. **(partial support under NCC-1-211)**

10. Thomas, J.L., Krist, S.T., and Anderson, W.K., "Navier-Stokes Computations of Vortical Flows Over Low Aspect Ratio Wings," *AIAA Journal*, Vol. 28, No. 2, Feb. 1990, pp. 205-215.

11. Baysal, O., Methods for Sensitivity Analysis in Gradient-based Shape Optimization: A Review," *Proceedings of ASME Fluids Engineering Conference*, Vancouver, BC, Canada, June 1997. **(partial support under NCC-1-211)**

12. Baysal, O., Cordero, Y., and Pandya, M.J., "Improving Discrete-Sensitivity-Based Approach for Practical Design Optimization," *Proceedings of JSME Fluids Engineering Conference*, Tokyo, Japan, July 1997. **(partial support under NCC-1-211)**

13. Eleshaky, M.E. and Baysal, O., "Airfoil Shape Optimization Using Sensitivity Analysis on Viscous Flow Equations," *Journal of Fluids Engineering*, Vol. 115, No. 1, Mar. 1993, pp. 75-84.

14. Eleshaky, M.E., and Baysal, O., "Design of 3-D Nacelle Near Flat-Plate Wing Using Multiblock Sensitivity Analysis (ADOS)," AIAA Paper 94-0160, 32nd Aerospace Sciences Meeting, Reno, NV, Jan. 1994. To appear in *Journal of Aircraft.*

15. Lacasse, J.M., and Baysal, O., "Shape Optimization of Single- and Two-Element Airfoils on Blocked Grids," AIAA Paper 94-4273, *Proceedings of 5th AIAA/USAF/NASA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Panama City, FL, Sept. 1994, pp. 108-116. To appear in *Journal of Inverse Problems in Engineering.*

16. Item, C.C., and Baysal, O., "Wing Section Optimization for Supersonic Viscous Flow," *CFD for Design and Optimization* (Ed. O. Baysal), ASME FED-Vol. 232, pp. 29-36, International Mechanical Engineering Conference and Exposition, November 12-17, 1995, San Francisco, CA. To appear in *Journal of Fluids Engineering.*

17. Eleshaky, M.E. and Baysal, O., "Aerodynamic Shape Optimization via Sensitivity Analysis on Decomposed Computational Domains," *Journal of Computers and Fluids,* Vol. 23, No. 4, May 1994, pp. 595-611.

18. Burgreen, G.W., Baysal, O., and Eleshaky, M.E., "Improving the Efficiency of Aerodynamic Shape Design Procedures," *AIAA Journal,* Vol. 32, No. 1, Jan. 1994, pp. 69-76.

19. Burgreen, G.W., and Baysal, O., "3-D Aerodynamic Shape Optimization of Wings Using Sensitivity Analysis," *AIAA Journal,* Vol. 34, No. 9, Sept. 1996, pp. 1761-1770.

20. Smith, R. E., Bloor, M.I.G., Wilson, M.J. and Thomas, A.M. ," Rapid Airplane Parametric Input Design (RAPID), ", *Proceedings of 12th AIAA Computational Fluid Dynamics Conference*, AIAA Paper 95-1687, June 1995.

21. Vanderplaats, G.N., "ADS - A FORTRAN Program for Automated Design Synthesis - Version 1.10," NASA Contractor Report 177985, Sept. 1985.

22. Baysal, O., Eleshaky, M.E., and Burgreen, G.W., "Aerodynamic Shape Optimization Using Sensitivity Analysis on Third-Order Euler Equations," *Journal of Aircraft*, Vol. 30, No. 6, Nov./Dec. 1993, pp. 953-961.

23 Korivi, V.M., Taylor, A.C. III, Newman, P.A., Hou, G.W., and Jones, H.E., "An Approximately Factored Incremental Strategy for Calculating Consistent Discrete CFD Sensitivity Derivatives," AIAA Paper 92-4746, Sept. 1992. Also, NASA-TM-104207, Feb. 1992.

24. Bischof, C., Corliss, G., Green, L., Griewank, A., Haigler, K., and Newman, P., "Automatic Differentiation of Advanced CFD Codes for Multidisciplinary Design," Symposium on High-Performance Computing for Flight Vehicles, Arlington, VA, Dec. 1992. To appear in *Journal on Computing Systems in Engineering.*

§.5.0 Publications

(left blank intentionally; please see the following pages)

FEDSM 97-3413

METHODS FOR SENSITIVITY ANALYSIS IN GRADIENT-BASED SHAPE OPTIMIZATION: A REVIEW

Oktay Baysal

Aerospace Engineering Department
Old Dominion University
Norfolk, Virginia 23529-0247

Preprint from the Proceedings

Fluids Engineering Division Summer Meeting
June 22-26 1997 / Vancouver BC, CANADA

The American Society of Mechanical Engineers

1997 ASME Fluids Engineering Division Summer Meeting
FEDSM'97
June 22 - 26, 1997

FEDSM97-3413

METHODS FOR SENSITIVITY ANALYSIS IN GRADIENT-BASED SHAPE OPTIMIZATION: A REVIEW

Oktay Baysal *
Old Dominion University
Norfolk, Virginia 23529-0247

INTRODUCTION

Near the beginning of this decade, a symposium on *Multidisciplinary Applications of Computational Fluid Dynamics* (Baysal, 1991) was organized. Its bound volume included a few papers reporting on the utilization of computational fluid dynamics (CFD) in a design environment. Since then, there have been a delightful increase in the interest on this topic for a clearly justified reason: CFD can be useful beyond just *simulating and analyzing* a fluid flow and be utilized for the purpose of design and optimization in order to cut down the cycle time for a new product design. This, however, may be a prohibitive proposition for the needed computational and human resources if, as often is the case, a large matrix of candidate designs or design variables are involved. Therefore, a few years later and with the motivation of providing a podium for beyond-cut-and-try approaches, where CFD would be a module of an automated methodology seeking the improved designs, the forum on *CFD for Design and Optimization* (Baysal, 1995) was organized. As it may be evidenced by the quality of the papers in its bound volume, some of the best in the field of CFD have directed their research to the various aspects of this topic.

Even a cursory glance through the emerging publications on this subject matter should attest to the attempts and successes on the topics which include: gradient-based numerical optimization methods, stochastic and genetic algorithms, shape optimization, direct and inverse methods, trade-off identification studies, multipoint designs, artificial-intelligence-based methods, pre- and post-optimization sensitivity analyses, adjoint methods, and discrete and continuous sensitivity methods. Today, there are government-funded initiatives to make even the *multidisciplinary* design optimization a reality, in which CFD plays one of the pivotal roles. Better yet, such efforts are not being perceived as just academic or esoteric exercises, but major corporations are investing in this exciting endeavor. It is the intention of the ASME Fluids Engineering Division to repeat these forums at a not-so-distant future, in its attempt to disseminate the further advances and applications on these topics. Consequently, a follow up forum on this topic, *Design Optimization Using CFD* (Johnson et al., 1997), has been organized, the proceedings of which is the present volume.

A clear message from these events is, that CFD is now being widely used in the design of mechanical and aerospace systems and their components. Improving the accuracy and efficiency of this process can reduce its cost and increase its effectiveness. Other motivating factors that push and pull CFD into design optimization are the following:
i) Need for design methodologies (single- and multi-disciplinary) to reduce cost and time in developing new fluid and aero systems, for fast response to market changes; ii) new design requirements outside existing database; iii) new CFD algorithms and growth in computing power; iv) need for increased product quality and reliability; (v) lack of test facilities; (vi) limitations and cost of testing techniques; vi) limitations of existing design practices; vii) prohibitive cost of and conflicting designs derived from isolated component analyses and design, which are repetitive in nature and omit mutual interference.

With all of this as impetus, new design methods, varying from inverse methods to numerical shape optimization methods, are being developed by the researchers. Some of the objectives that such efforts try to accomplish are: i) Automated design optimization with minimal need for man-in-loop in a given try; ii) Reduce the need for the design expertise and the prerequisite database; iii) Improve accuracy, efficiency and practicality; iv) Information on the most influencing parameters, which leads to a reduced CFD analysis matrix, requires sensitivities of objectives and constraints with respect to design variables; v) the ability to design with a variety of fluid dynamic and geometric constraints and geometric flexibility (type and number of design variables and efficient parameterization).

The current approaches to the engineering design may loosely be classified as follows.

* Professor and Eminent Scholar, Aerospace Engineering Department. Fellow, ASME.

1) *Non-automated design*: a) parametric analysis and cut-and-try approaches, b) database-matrix approaches, c) other intuitive approaches. These are highly data- and expertise-dependent, generally nonsystematic, and often inefficient practices, which provide discrete designs in the design space, with uncertainty in optimality.

2) *Inverse design* methods require a priori knowledge of the flowfield for the final design in order to prescribe it as a target, then perform indirect searches. They are usually weak on geometry control and constraints, resulting in poor off-design performance. However, when applicable, they are efficient, hence, widely used. They may be grouped as: a) surface flow design (e.g. prescribed $C_{p,surface}$ distribution), b) flowfield design (e.g. shock-free design).

3) *Direct numerical optimization* methods are systematic methodologies which extremize a chosen objective. They create or merely improve a design. They may produce the *best* design or just a *better* design, dependent on if the global or just a local minimum has been found. They can be a conduit to understand the design process, and the effects of physical phenomena associated with the evolving shapes or designs. It may be viewed as a science or an art. They are suitable for multipoint design and allow for trade-off studies. However, they can be rather computing intensive. Direct numerical optimization methods can be grouped based on several factors:

a) *Constrained optimization* is gradient-based ($\nabla F, \nabla G_j$), and relies on deterministic searches, ($D_i^{n+1} = D_i^n + \alpha^n S_i^n$). Its mathematical formulation consists of the extremization of an objective function $F\left(D_i, Q\left(D_i\right)\right)$, subject to: fluid dynamic or geometric *inequality constraints* $G_j\left(D_i, Q\left(D_i\right)\right) \le 0$, fluid dynamic or geometric *equality constraints* $H_k(D_i, Q(D_i)) = 0$, *analysis equations*, $R_m\left(D_i, Q(D_i)\right) = 0$, and the *side constraints*, $D_i^{lower} \le D_i \le D_i^{upper}$, where $i = 1, NDV$, $j = 1, NCON$, $k = 1, NCON_e$, $m = 1, NEQ$.

b) *Unconstrained optimization*, may also use deterministic searches, and the constraints may be by projected by some type of a Lagrangian approach ($\nabla F + \sum \lambda_j \bullet \nabla G_j = 0$).

c) Optimization by *stochastic* search methods, such as, *simulated annealing*, emulating cooling schedules, and *genetic algorithms*, emulating survival of the fittest concept. These have better chance to find the global minimum, however, they are computer-resource intensive, since the needed number of function evaluations is usually unknown; that is, there virtually is no stopping criterion.

d) *Knowledge-based and databased* algorithms use artificial intelligence with *if-then* rules, or the available databases on designs generated, for example, by the design of experiments. Response-surface methods or surrogate methods belong to this category.

One type of simulation-based and automated design method, utilizing a constrained optimization, is the shape optimization. This typically consists of the following components:

(i) formulation of the problem, i.e. identifying the objective and the constraint functions;
(ii) gradient-based and constrained optimization algorithm;
(iii) flowfield simulation and analysis, i.e. CFD;
(iv) gradients of the objective and the constraints with respect to the design variables, i.e. the sensitivity coefficients;
(v) parameterization and redefinition of shapes as they evolve, i.e. using computer-aided-design (CAD);
(vi) regeneration of surface and volume grids, and the grid sensitivities;
(vii) gradients of the optimized shape with respect to the design invariants, i.e. the sensitivity derivatives, used for trade-off studies and off-design conditions.

SENSITIVITY ANALYSIS METHODS

The accuracy of a gradient-based optimization method, and the efficiency with which it can accomplish this, are directly related to the accurate and efficient receipt of the gradient information on the objectives and the constraints. The sensitivity analysis provides: i) the gradients ($\nabla F, \nabla G_j$) for optimization and trade-off studies; ii) the post-optimization gradients for off-design conditions; iii) a first-order but inexpensive approximation to neighboring-point analysis via Taylor series expansion (Baysal et al., 1993). The gradient evaluation may be performed by the following approaches.

1) *Finite-difference* method is a brute-force approach, which is highly prone to inaccuracies and inefficiencies. A poor selection of the design variable increments used in the finite-differencing may lead to inaccurate sensitivities (Baysal and Eleshaky, 1991). Another drawback is the computational cost necessitated by the repeated flowfield analyses of which at least $NDV + 1$ are required when using a one-sided finite-difference technique, where NDV is the number of design variables. However, it is the easiest to formulate and is often readily available with off-the-shelf optimization packages.

2) *Implicit gradient evaluation* methods: The governing equations of fluid flow can be differentiated analytically either starting with their original differential form and using the variational concepts (*continuous* sensitivities) or after they have been discretized (*discrete* sensitivity analysis).

i) In *continuous adjoint formulation* (a.k.a. control theory), the adjoint (or *co-state*) equations and their boundary conditions are derived by the use of variational methods. These partial differential equations are of the same order and character as the flow equations. Hence, they can be discretized and solved for the Lagrange variables by the same CFD scheme used for the flow (state) equations. Hence, its computer storage requirement is no more than what the flow analysis requires. Nonetheless, its analytical development is highly *case-dependent*, often cumbersome, which may be characterized as front-end loaded.
ii) *Discrete sensitivity analysis* (a.k.a. quasi-analytical) is performed by differentiating the already CFD-discretized equations with respect to the design variables. It corresponds to a discrete solution of the continuous sensitivity function. Both hand-differentiation and automatic differentiation can be used to generate the sensitivity equations.

Continuous adjoint method

Variational design optimization blends the principles of optimal control theory and variational methods. The optimal control theory states the conditions under which the control variables, parameters, and functions, or their combinations, can be continuously altered so that the system is dictated to meet the desired criteria. When constraints and flow equations are involved, the systematic way of solving them is based on the classical adjoint functions technique, which transforms the constrained optimization to an unconstrained one. In a variety of constrained optimization problems, the converged solution of the continuous flowfield equations,

$$R(Q(D), X(D), D) = 0, \tag{1}$$

can be uniquely defined for the specific boundary conditions along the whole or part of the domain boundary. In other words, the constrained problem is transformed to find a boundary condition so that the unique solution of eq. (1) extremizes the objective function $F\left(D_i, Q(D_i)\right)$. Then, from the Lagrange variables approach, the adjoint functional is defined as

$$\begin{aligned} F_a(Q(D), D, \vec{\lambda}) &= F(Q(D), D) \\ &+ \iint \vec{\lambda}^t(X,t) \cdot R(X,t,Q(D),D)\, dX\, dt \end{aligned} \tag{2}$$

For the shape optimization, the optimum location of the boundary is iteratively computed such that the normal displacement produces a negative variation in the objective function (minimization). This is mathematically expressed as the scalar product between the gradient of the objective function and the variation of the design variables in the design space:

$$\delta F_a = < \nabla F_a, \delta D >_\Gamma \ , \text{ where } \ \nabla F_a \approx [\tfrac{\delta F}{\delta D}]\, \delta D \ . \tag{3}$$

Then, denoting the step size with α, the boundary is modified through,

$$\delta D \approx D^{m+1} - D^m = \begin{cases} -\alpha \nabla F_a \\ f(\nabla F_a) \end{cases} . \tag{4}$$

In eq. (4), the first choice is the steepest descent method. The second choice involves the more sophisticated search direction methods developed for the numerical optimization methods.

The first-order variation of eq. (2) with respect to the flow and design variables is derived, then simplified considering the fact that the variation $R \cdot \delta\lambda$ satisfies eq. (1):

$$\begin{aligned} \delta F_a(Q(D), D, \vec{\lambda}) &= \delta F(Q(D), D) \\ &+ \iint \vec{\lambda}^t(X,t) \cdot \delta R(X,t,Q(D),D) dX dt \end{aligned} \tag{5}$$

To extremize eq. (2), eq. (5) is set to zero. From the first term on the right hand side, the first integral provides the boundary (transversality) conditions and the second integral provides the initial conditions. From the second term on the right hand side, the first and the second integrals provide the adjoint equations and the corresponding sensitivity equation (functional derivative):

$$\begin{aligned} \delta F_a &= f(b.c.\ for\ \vec{\lambda})\Big|_0^X + f(i.c.\ for\ \vec{\lambda})\Big|_0^T \\ &+ \iint [adjoint\ eqs.\ for\ \vec{\lambda}]\, dX\, dt \end{aligned} \tag{6}$$

Adjoint equations are of the same order and character as the flow equations. Hence, they can be discretized and solved for the Lagrange variables by the same CFD scheme used for the flow (state) equations.

Discrete sensitivity analysis

The sensitivities needed by the optimization can be obtained by the chain rule of differentiation,

$$\begin{aligned} \nabla F &= \left[\left(\frac{\partial F}{\partial D_i} \right)_Q + \left(\frac{\partial F}{\partial Q} \right)_D^T \cdot Q_i' \right] \ , \\ \nabla G_j &= \left[\left(\frac{\partial G_j}{\partial D_i} \right)_Q + \left(\frac{\partial G_j}{\partial Q} \right)_D^T \cdot Q_i' \right] \ . \end{aligned} \tag{7}$$

where all, except the $Q_i' \equiv \frac{\partial Q}{\partial D_i}$ term can be obtained by direct differentiation. From the residual form of the *discretized* flowfield equations,

$$R(Q(D), M(X(D)), D) = \vartheta(tol) \ , \tag{8}$$

and by differentiation of eq. (8), an equation is obtained for Q_i',

$$R' = \left(\frac{\partial R}{\partial Q} \right)_D Q' + \left(\frac{\partial R}{\partial D} \right)_Q = 0 \ . \tag{9}$$

Then, the *direct method* is simply solving for Q' in eq. (9),

$$\left(\frac{\partial R}{\partial Q} \right)_D Q' = -\left(\frac{\partial R}{\partial D} \right)_Q \ , \tag{10}$$

which can also written in incremental (deficit-correction) form,

$$\left(\frac{\partial \hat{R}}{\partial Q} \right)_D^n \{\Delta Q'\}^n = -\left(\frac{\partial R}{\partial Q} \right)_D^n (Q')^n - \left(\frac{\partial R}{\partial D} \right)_Q \ . \tag{11}$$

Note that eqs. (10-11) scale with the number of design variables. An alternative, which scales with the number of inequality constraints, is the *adjoint method*. The adjoint sensitivities are obtained from eqs. (7, 10),

$$\nabla F = \left[\left(\frac{\partial F}{\partial D_i} \right)_Q - \lambda_F^T \frac{\partial R}{\partial D_i} \right], \nabla G_j = \left[\left(\frac{\partial G_j}{\partial D_i} \right)_Q - \lambda_{G_j}^T \frac{\partial R}{\partial D_i} \right] \tag{12}$$

The equations for the Lagrange variables are obtained again from eqs. (7, 10),

$$\left(\frac{\partial R}{\partial Q}\right)_D^T \lambda_F = \left(\frac{\partial F}{\partial Q}\right)_D , \left(\frac{\partial R}{\partial Q}\right)^T \lambda_{G_j} = \frac{\partial G_j}{\partial Q} \quad , \qquad (13)$$

which can also written in incremental (deficit-correction) form,

$$\left(\frac{\partial \hat{R}}{\partial Q}\right)^T \{\Delta \lambda_F\}^n = -\left(\frac{\partial R}{\partial Q}\right)_D^T \lambda_F^n - \left(\frac{\partial F}{\partial Q}\right)_Q . \qquad (14)$$

Therefore, to obtain the sensitivities, either eqs. (10 or 11) are solved for Q', or eqs. (13 or 14) are solved for λ_F, λ_G.

Automatic differentiation

In developing a discrete sensitivity method, one feature that requires intense effort is the differentiation of a number of functions with respect to the design variables. This has traditionally been done using the calculus definition of a derivative represented as a finite difference, then evaluating the function as many times as needed. A second method is the *hand-differentiation* of the equations or in their discretized form. This quasi-analytical approach is effort-intensive. A third option for the differentiation is the utilization of mathematical computer packages for this purpose. As most packages would include a symbolic differentiation module, this would be the natural approach to consider. Another approach is the *automatic differentiation* (e.g. Bischoff et al., 1992), which, given the computer code for a function, generates a computer code that can produce the sensitivities of that function. Rather than computing the full Jacobian, the product of the Jacobian and a seed matrix has been shown to be more efficient to compute. This method has recently been tried extensively by a number of researchers for different applications including those mentioned above (Baysal and Cordero, 1996).

Surface parameterization

A non-parametric geometry definition requires the surface mesh point coordinates. Although this approach utilizes the readily available data, it increases the number of design variables. On the other hand, a parametric geometry definition reduces the number of design variables, since the number of control points is significantly less than the surface mesh points. Current geometry definition techniques used in computer-aided design (CAD) are capable of parameterizing space curves and space surfaces with high fidelity. They do require some understanding in making the choice of parameterization technique, and extra computations and storage. As an example, the Bezier-Bernstein approach is considered to represent a surface defined by its grid points $X\{x_i, y_i, z_i\}$, $i = 1, 2, .., I$. Then, an (N,M)-th degree representation $(N << I, M << I)$ is obtained from a birectional spline frame. With $0 \le u, v \le 1$,

$$X_{j,k}(u,v) = \sum_{n=0}^{N} \sum_{m=0}^{M} P_{nm} B_n(u) B_m(v) \quad , \qquad (15)$$

where

$$B_n(u) = \binom{N}{n} u^n (1-u)^{N-n}, \; B_m(v) = \binom{M}{m} v^m (1-v)^{M-m} . \qquad (16)$$

After a single-indexed ordering $[i = 1, 2, .., (n \cdot m)]$ of the control points, the solution requires forming a tensor product of the 1D basis functions $B_n(u), B_m(v)$.

In parameterizing a configuration, which includes the CAD-parameterized design surfaces, parameters that are more intuitively recognizable to the specific discipline may be preferred. For example, a wing may be described by thickness and chord of the airfoil sections, taper distribution, sweep, span, spanwise bending, geometric twist, and global angle-of-attack. Once a new configuration and its design surfaces are defined, the next task is the re-generation of its volume grid. Hence, efficient approaches should be developed to analytically relate the surface parameters to their computational volume grid. Then the grid may be regenerated systematically as the vector of design variables is changed with the evolving shape. As the surface is changed during the shape optimization, the new volume grid may be found using the following relations:

$$X_i^{new} = X_i^{old} + [1 - v_j](X_b^{new} - X_b^{old}), \text{ where } v_j = \frac{s_j - s_2}{s_{j\,max} - s_2}, \qquad (17)$$

$$s_j = \sum_{i=2}^{j} \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2} \quad . \qquad (18)$$

SYNOPSIS OF SAMPLE RESULTS

In order to illustrate the concepts described above, a brief overview of the developments from the research by the author and his graduate students will be presented. For brevity, a comprehensive overview of parallel developments by other researchers (e.g. Huan and Modi, 1995, Reuther and Jameson, 1995, DeRise and Thomas, 1996, Knight, 1996, Pelz et al., 1997, Anderson and Venkatakrishnan, 1997), who are very much deserving, have not been included, but they should be followed in the pertinent literature.

An aerodynamic sensitivity analysis method was first presented for the 2D compressible Euler equations (Baysal and Eleshaky, 1991). This method was then placed in a design optimization methodology and demonstrated by applying it to the design of a scramjet-afterbody configuration for maximum axial thrust (Baysal and Eleshaky, 1992). The scramjet-afterbody was again used to validate an extension of the methodology to the higher-order discretization of the Euler equations, which was up to third-order accurate (Baysal et al., 1993). A first-order and inexpensive flowfield prediction method for a given design was also introduced, which used a CFD solution and its sensitivities at a nearby design. Effects of diffusion on design were later accounted for by deriving the sensitivities of the thin-layer Navier-Stokes equations and using them in optimizing a transonic airfoil (Eleshaky and Baysal, 1993). Then, the issue of computational-time efficiency was addressed in several ways, including the surface parameterization using Bezier splines and Bernstein polynomials, and solving the unfactored CFD equation by a quasi-Newton's method (Burgreen et al., 1994). The computer storage efficiency was later

addressed by examining the solution of the sensitivity equation as well as the CFD equation using a preconditioned biconjugate gradient (PbCG) method, known as the generalized minimum residual [GMRES(k,m)] method (restarting version), as compared to a direct inversion method (Burgreen and Baysal, 1994a).

Concurrently, the investigation was turned to the development of the *multiblock sensitivity analysis scheme* called SADD (Eleshaky and Baysal, 1994a). This scheme was motivated by the need to address the computer memory issues in the direct inversion of the sensitivity equation's coefficient matrix, which becomes particularly large. Another important benefit was its applicability to problems involving complex and multicomponent geometries, around which structured grids can only be generated by the use of domain decomposition techniques. The additional advantages of this approach were: (i) its amenability to parallel processing, (ii) demonstration (without the implementation) of a method for non-hierarchial, multidisciplinary sensitivity analysis, where the sensitivities of different disciplines would be coupled. This approach was then used in demonstrating the shape optimization of a multi-component airfoil for high lift, where the components were shaped simultaneously, that is, including the mutual interference into the design process (Lacasse and Baysal, 1994). The SADD scheme was later extended to three dimensions, where GMRES(k,m) was also incorporated (Eleshaky and Baysal, 1994b) and applied in the optimization of a nacelle near a wing (Eleshaky and Baysal, 1994c); another simultaneous shaping of components with mutual aerodynamic interference.

In the PbCG methods mentioned above, incomplete lower-upper (ILU) decompositions of the coefficient matrix with some fill-in ($ILU(n)$, $n \geq 0$) were used for the preconditioning. The 3D optimization methodology with an efficient wing parameterization was successfully demonstrated for a transport wing (Burgreen and Baysal, 1996), and later it was applied for shaping asymmetric delta wings and cranked delta wings (Burgreen and Baysal, 1994b). Another preconditioning option is to use the approximate factors of the coefficient matrix, thereby break the problem into a sequence of simpler problems. That is the premise of the *alternating-direction-implicit* (ADI) scheme, which has been used in CFD to solve the flow equations for more than two decades. The trade-off in the latter approach is a slower convergence for the much reduced computer memory storage.

The results from all the applications cited above indicate that, in general, the design methods obtain a final shape via an evolution of successively improved shapes. A practical use of fully implicit CFD methods (quasi-Newton method) within an optimization procedure allows the realization of high convergence rates and consequent reduction of CPU time. The memory requirement of such a procedure which uses a PbCG algorithm is low as compared to direct inversion solvers. However, this memory requirement is high enough to preclude the realistic, high-grid-density design of a practical 3D geometry, such as wings or wing-body combinations. This improvement has been achieved by the use of ADI-factored operators to serve as the preconditioning matrices for the CFD as well as the sensitivity equations (Pandya and Baysal, 1996). The method was then extended to the 3D thin-layer Navier Stokes equations and tested by shape optimizing cranked-arrow wings in supersonic flows (Pandya and Baysal, 1997).

To study the potential advantages of the automatic differentiation, a simple demonstration case has also been formulated based on this cranked-arrow wing geometry. It has been parameterized and the surface grid has been generated in two different ways: a method based on the partial differential equations and another based on the Bezier-Bernstein parameterization. Then, the grid sensitivities have been computed using an automatic differentiated code in the former method and using a hand-differentiated code in the latter method. It was observed that the results had comparable levels of accuracy, but the hand differentiated code required less number of arithmetic operations and utilized less storage. However, the advantage of the automatic differentiation was in the significantly reduced amount of effort for the development of the code that generated the sensitivities (Baysal and Cordero, 1996).

A design optimization method based on the variational methods has also been studied and a proof-of-concept study was conducted for a quasi-one-dimensional duct flow (Ibrahim and Baysal, 1994). This study indicated that the method required significantly less computer storage and processing time. Consequently, it was capable of handling large-scale optimization problems.

Recapitulating, the methods have been illustrated through a number of academic examples. They included viscous and inviscid designs, and isolated and mutually-interfering multi-component designs:

(i) National AeroSpace Plane (NASP) nozzle-afterbody shaping; 2D, supersonic/hypersonic, inviscid; Baysal and Eleshaky (1991, 1992), Baysal et al. (1993), Burgreen et al. (1994).
(ii) Nozzle; quasi-1D, inviscid; Ibrahim and Baysal (1994).
(iii) Airfoil shaping; 2D, transonic and supersonic, viscous and inviscid; Eleshaky and Baysal (1993), Eleshaky and Baysal (1994a), Burgreen and Baysal (1994a), Item and Baysal (1995).
(iv) Multi-element airfoil shaping; 2D, subsonic, viscous; Lacasse and Baysal (1994).
(v) Isolated nacelle and nacelle near a wing; 3D, supersonic, inviscid and viscous; Eleshaky and Baysal (1994b, 1994c).
(vi) Transport wings and delta wings; 3-D, transonic and low-supersonic, inviscid; Burgreen and Baysal (1994b, 1996),
(vii) Arrow Wings; 3D, supersonic, inviscid and viscous; Pandya and Baysal (1996, 1997), Baysal and Cordero (1996).

Moving from the single-discipline to a multi-discipline optimization may now be considered as a viable prospect. Towards this goal, there have been some suggested formulations to couple the disciplines at various levels. These vary regarding the type or types of optimizers per discipline, whether the analyses and optimizations are performed simultaneously or in a nested manner, and whether they are done at the system level or discipline level or both. Currently, there doesn't appear to be a unique combination that renders the designs in the most efficient way for all possible problems. Some a priori consideration of the problem at hand is deemed

productive, and may be, a hybrid approach would surface as the best. Whichever path a designer may choose to pursue, one certain thing ought to be to take advantage of the lessons learned from the single-discipline methodologies to shape optimization. A case may be made from a method developed for aerodynamic shape optimization for multi-component configuration design: it is suggested that the components be conceivably thought of as disciplines, hence deriving a tightly coupled multidisciplinary sensitivity equation. Although the mathematical procedure in solving the equivalent sensitivity equation of SADD (Eleshaky and Baysal, 1994a, 1994b) was somewhat similar to that of a single sensitivity equation, the sensitivities were obtained in a closely coupled manner. The extension of SADD to a multi-disciplinary problem would propose a single-level optimization with simultaneous analysis and design at the system level, but nested analysis and design at the discipline level (Baysal and Eleshaky, 1996).

ACKNOWLEDGMENTS

The author is grateful to his past and present graduate students who worked with him on design optimization: G.W. Burgreen, Y. Cordero, M.E. Eleshaky, A.H. Ibrahim, C.C. Item, J.M. Lacasse, and M.J. Pandya. Development of the methods were supported and the computer time was provided by NASA Langley Research Center. Technical monitors were D.S. Miller and J.L. Thomas.

REFERENCES

Anderson, W.K., and Venkatakrishnan, V., "Aerodynamic Design Optimization on Unstructured Grids with a Continuous Adjoint Formulation," Paper No. 97-0643, AIAA 35th Aerospace Sciences Meeting, Reno, NV.

Baysal, O., (Editor), 1991, Multidisciplinary Applications of Computational Fluid Dynamics, FED-Vol. 129, Winter Annual Meeting, Atlanta, GA; ASME, New York, NY.

Baysal, O., 1992, "Flow Analysis and Design Optimization Methods for Nozzle-Afterbody of a Hypersonic Vehicle," Computational Methods in Hypersonic Aerodynamics (Ed.: T.K.S. Murthy), Ch. 10, pp. 341-386, Computational Mechanics, Kluwer Academic, Southampton, U.K.

Baysal, O., (Editor), 1995, CFD for Design and Optimization, FED Vol. 232, International Mechanical Engineering Conference and Exposition, San Francisco, CA; ASME, New York, NY.

Baysal, O., and Cordero, Y., 1996, "A Perspective on Automatic Differentiation for Gradient-Based Aerodynamic Shape Optimization," Minisymposium on Automatic Differentiation for Large-Scale Optimization, Proceedings of 3rd ECCOMAS Computational Fluid Dynamics Conference, Paris, France.

Baysal, O., and Eleshaky, M.E., 1991, "Aerodynamic Sensitivity Analysis Methods For the Compressible Euler Equations," *Journal of Fluids Engineering*, Vol. 113, No. 4, pp. 681-688.

Baysal, O., and Eleshaky, M.E., 1992, "Aerodynamic Design Optimization Using Sensitivity Analysis and Computational Fluid Dynamics," *AIAA Journal*, Vol. 30, No. 3, pp. 718–725.

Baysal, O., Eleshaky, M.E., and Burgreen, G.W., 1993, "Aerodynamic Shape Optimization Using Sensitivity Analysis on Third-Order Euler Equations," *Journal of Aircraft*, Vol. 30, No. 6, pp. 953-961.

Baysal, O., and Eleshaky, M.E., 1996, "Case for Single-Discipline Methodology in Multidisciplinary Design Optimization," *Proceedings of 33rd Annual Technical Meeting of Society of Engineering Science*, Tempe, AZ.

Bischof, C.H., Carle, A., Corliss, G.F., and Griewank, A., 1992, "ADIFOR: Generating Derivative Codes from Fortran Programs," *Scientific Programming*, Vol. 1, No. 1, pp. 1-29.

Burgreen, G.W., Baysal, O., 1994a, "Aerodynamic Shape Optimization Using Preconditioned Conjugate Gradient Methods," *AIAA Journal*, Vol. 32, No. 11, pp. 2145-2152.

Burgreen, G.W., and Baysal, O., 1994b, "Three-Dimensional Aerodynamic Shape Optimization of Delta Wings," Paper 94-4271 CP, Proceedings of 5th AIAA/USAF/NASA/ISSMO Multidisciplinary Analysis and Optimization Conference, Panama City, FL, pp. 87-97.

Burgreen, G.W., Baysal, O., and Eleshaky, M.E., 1994, "Improving the Efficiency of Aerodynamic Shape Design Procedures," *AIAA Journal*, Vol. 32, No. 1, pp. 69-76.

Burgreen, G.W., and Baysal, O., 1996, "3D Aerodynamic Shape Optimization Using Discrete Sensitivity Analysis," *AIAA Journal*, Vol. 34, No. 9, pp. 1761-1770.

DeRise, G., and Thomas, J.L., 1995, "Viscous Aerodynamic Design using Adjoint Variable Approach - a Methodology," ASEE Summer Faculty Fellowship Lecture, Fluid Mechanics and Acoustics Division, NASA Langley Research Center, Hampton, VA.

Eleshaky, M.E. and Baysal, O., 1993, "Airfoil Shape Optimization Using Sensitivity Analysis on Viscous Flow Equations," *Journal of Fluids Engineering*, Vol. 115, No. 1, Mar. 1993, pp. 75-84.

Eleshaky, M.E. and Baysal, O., 1994a, "Aerodynamic Shape Optimization via Sensitivity Analysis on Decomposed Computational Domains," *Journal of Computers and Fluids*, Vol. 23, No. 4, pp. 595-611.

Eleshaky, M.E., and Baysal, O., 1994b, "Preconditioned Domain Decomposition Scheme for 3-D Aerodynamic

Sensitivity Analysis," *AIAA Journal*, Vol. 32, No. 12, pp. 2489-2491.

Eleshaky, M.E., and Baysal, O., 1994c, "Design of 3D Nacelle Near Flat-Plate Wing Using Multiblock Sensitivity Analysis (ADOS)," Paper 94-0160, AIAA 32nd Aerospace Sciences Meeting, Reno, NV. To appear in *Journal of Aircraft.*

Huan, J., and Modi, V., 1995, "Design of Minimum Drag Bodies in Incompressible Laminar Flow," CFD for Design and Optimization (Ed. O. Baysal), FED-Vol. 232, pp. 37-44, International Mechanical Engineering Conference and Exposition, San Francisco, CA; ASME , New York, NY.

Ibrahim, A.H., and Baysal, O., 1994, "Design Optimization Using Variational Methods and CFD," Paper 94-0093, AIAA 32nd Aerospace Sciences Meeting, Reno, NV.

Item, C.C., and Baysal, O., 1995, "Wing-Section Optimization for Supersonic Viscous Flow," CFD for Design and Optimization (Ed. O. Baysal), FED Vol. 232, International Mechanical Engineering Conference and Exposition, San Francisco, CA; ASME , New York, NY.

Johnson, G., Baysal, O., and Lee, Y-T. (Editors), 1997, "Design Optimization Using CFD," Proceedings of Fluids Engineering Division Summer Meeting, FED Vol. xxx, Vancouver, Canada; ASME , New York, NY.

Knight, D.D., 1996, "Automated Optimal Design Using CFD and High Performance Computing," 2nd International Meeting on Vector and Parallel Processing, Porto, Portugal.

Lacasse, J.M., and Baysal, O., 1994, "Shape Optimization of Single- and Two-Element Airfoils on Blocked Grids," Paper 94-4273, Proceedings of 5th AIAA/USAF/NASA/ISSMO Multidisciplinary Analysis and Optimization Conference, Panama City, FL, pp. 108-116. To appear in *Inverse Problems in Engineering.*

Pandya, M.J., and Baysal, O., 1996, "Gradient-Based Aerodynamic Shape Optimization Using ADI Method for Large-Scale Problems," Paper No. 96-0091, AIAA 34th Aerospace Sciences Meeting, Reno, NV. To appear in *Journal of Aircraft.*

Pandya, M.J., and Baysal, O., 1997, "3D Viscous ADI Algorithm and Strategies for Shape Optimization," Paper No. 97-1853 CP, *Proceedings of AIAA 13th CFD Conference*, Snowmass, CO.

Pelz, R.B., Ogot, M., Aly, S., Cantelmi, F., Burke, B., 1997, "Global Stochastic Methods in MDO/CFD," Paper No. 97-0352, AIAA 35th Aerospace Sciences Meeting, Reno, NV.

Reuther, J., and Jameson, A., 1995, "Supersonic Wing and Wing-Body Shape Optimization using an Adjoint Formulation," CFD for Design and Optimization (Ed. O. Baysal), FED-Vol. 232, pp. 45-52, International Mechanical Engineering Conference and Exposition, San Francisco, CA; ASME , New York, NY.

AIAA 96-0091
C-8728

Gradient-Based Aerodynamic Shape Optimization Using ADI Method for Large-Scale Problems

Mohagna J. Pandya and Oktay Baysal

Aerospace Engineering Department
Old Dominion University
Norfolk, Virginia 23529-0247

Preprint from

Journal of Aircraft
Volume 34, No. 3, May-June 1997
The American Institute of Aeronautics and Astronautics

Journal of Aircraft
Volume 34, No. 3, May-June 1997

Gradient-Based Aerodynamic Shape Optimization Using ADI Method for Large-Scale Problems

Mohagna J. Pandya and Oktay Baysal

Aerospace Engineering
Old Dominion University
Norfolk, Virginia

A gradient-based shape optimization methodology, that is intended for practical three-dimensional aerodynamic applications, has been developed. It is based on the quasi-analytical sensitivities. The flow analysis is rendered by a fully implicit, finite volume formulation of the Euler equations. The aerodynamic sensitivity equation is solved using the alternating-direction-implicit (ADI) algorithm for memory efficiency. A flexible wing geometry model, that is based on surface parameterization and planform schedules, is utilized. The present methodology and its components have been tested via several comparisons. Initially, the flow analysis for a wing is compared with those obtained using an unfactored, preconditioned conjugate gradient approach (PCG), and an extensively validated CFD code. Then, the sensitivities computed with the present method have been compared with those obtained using the finite-difference and the PCG approaches. Effects of grid refinement and convergence tolerance on the analyses and the shape optimization have been explored. Finally, the new procedure has been demonstrated in the design of a cranked arrow wing at Mach 2.4. Despite the expected increase in the computational time, the results indicate that shape optimization problems, which require large numbers of grid points can be resolved with a gradient-based approach.

Nomenclature

A	area of airfoil section
aoa	angle of attack
C_D	coefficient of drag
C_L	coefficient of lift
C_L/C_D	lift-to-drag ratio
c, chdscal	chord and its distribution
C_P	coefficient of pressure
D	design variables
F	objective function
$\hat{F},\hat{G},\hat{H}$	fluxes in curvilinear coordinates
G	constraints
M	vector of metric terms
MW	million 64-bit words
NDV	number of design variables
Q	vector of conserved flow variables
Q'	sensitivity of Q to design variable
R	residual
s	arc length
spn	half-span length
t, thkscal	thickness and its distribution
tranx, tranz	coordinates of wing leading edge
twst	geometric twist
V	wing volume
X	vector of volume-grid coordinates
∇F	gradient of objective function
ΔQ	correction to flow variables
ε	tolerance
λ	adjoint-variable vector
Λ	sweep angle
ω	relaxation parameter
θ	included angle at trailing edge

Subscripts and superscripts

b	body surface
n	time level
crank	mid half-span location
LE	wing leading edge
TE	wing trailing edge

Introduction

Developing an effective three-dimensional design optimization procedure, that is also automated and practical to use, has been a topic of intense research since its introduction two decades ago.[1] Despite the formidable strides made in recent years, the challenge does not appear to have been met.

The accuracy of a gradient-based optimization method, and the efficiency with which it can accomplish this, are directly related to the accurate and efficient receipt of the gradient information on the objectives and the constraints. These sensitivity coefficients can be delivered to the optimizer via *sensitivity analysis*, which often refers to the quasi-analytically obtained sensitivities, rather than those obtained by the traditional finite-differences approach. To

this end, the governing equations of fluid flow can be differentiated analytically either starting with their original differential form and using the variational concepts[2,3] (*variational* sensitivity analysis; also known as continuous method, or adjoint formulation, or control theory approach), or after they have been discretized[4] (*discrete* sensitivity analysis).

In the variational sensitivity analysis approach, the *co-state* equations (or the *adjoint* equations) are partial differential equations, which are of the same order and character as the flow equations. Hence, they can be discretized and solved for the perturbations of the dependent variables by the same CFD technique, resulting in its computer memory requirement being no more than what the flow analysis requires.[5] Nonetheless, this approach is not without a drawback: its analytical development is highly *case-dependent*, rather cumbersome, and may be characterized as front-end loaded.

Since the early 90's, there have been promising developments in aerodynamic design optimization based on discrete sensitivity analysis.[6] Its computational-time efficiency has been addressed in several ways,[7] including the surface parameterization using Bezier splines and Bernstein polynomials, and solving the unfactored CFD equations by a quasi-Newton method. The computer storage efficiency was later addressed, first by examining the solution of all the equations using a *preconditioned conjugate gradient* (PCG) method,[8] then by the *multiblock sensitivity* analysis scheme.[9] An important benefit of the latter was its applicability to a complex and multicomponent geometry, around which structured grids could only be generated by the use of domain decomposition techniques. The multiblock sensitivity analysis was extended for 3D shapes, where a PCG method was also incorporated[10] and applied in the optimization of a multiple-component configuration.[11]

In the PCG methods mentioned above,[8,10] incomplete lower-upper (ILU) decompositions of the coefficient matrix with some fill-in (ILU (n), n>0) were used for the preconditioning. Another preconditioning option is to use the approximate factors of the coefficient matrix, then break the problem into a sequence of simpler problems; that is the premise of the ADI (*alternating-direction-implicit*) schemes, which have been used to solve the CFD equation for more than two decades.[12] The trade-off in the latter approach is a slower convergence for the much reduced computer memory storage.[13]

The results from all the applications cited above indicate that, in general, the design methods obtain a final shape via an evolution of successively improved shapes. A practical use of fully implicit CFD methods (quasi-Newton method) within an optimization procedure allows the realization of high convergence rates and consequent reduction of CPU time. The memory requirement of such a procedure which uses a PCG algorithm is low as compared to direct inversion solvers. However, this memory requirement is high enough to preclude the realistic, high-grid-density design of a practical 3D geometry such as a wing or wing-body combination.

This particular limitation served as the impetus to improve upon the recently developed 3D optimization methodology, which was demonstrated for a transport wing,[14] and later it was applied for shaping asymmetric delta wings and cranked delta wings.[15] This improvement has been achieved by the use of ADI-factored operators to serve as preconditioning matrices for the CFD as well as the sensitivity equations.

Synopsis of Methodology

Aerodynamic design optimization aims at the extremization of an objective function $F(D,Q(D))$ subject to constraints $G(D,Q(D))$. Both F and G may be nonlinear or non-smooth[16] functions of D and Q. In the present formulation, F and G are obtained from the governing equations of three-dimensional, compressible, inviscid flow, which are written in the steady-state *residual* form

$$R = R\{Q(D), M[X(D)]\} = \vartheta(\varepsilon) \tag{1}$$

Their fully implicit discretization written in delta, or deficit correction, form is,

$$\left(\frac{\partial R}{\partial Q}\right)^n \Delta Q^n = -R\left(Q^n, M\right) \tag{2}$$

Equation (2) was discretized in space using a cell centered, control volume formulation. The flux vectors and the Jacobian matrix $\partial R/\partial Q$ were evaluated using the flux-vector-splitting technique of van Leer. The cell interface values of Q were determined using a spatially third-order accurate, upwind-biased, MUSCL interpolation formula with van Albada flux limiter.

A strict application of Newton method, including consistent treatment of the boundary conditions, generally results in very large error reductions per iteration; quadratic convergence is achieved when the solution enters the domain of attraction to the root. However, because of the large memory requirement of the method, practical application of the same is thwarted. Since at convergence, the deficit correction is within $\vartheta(\varepsilon)$, the coefficient matrix in Eq. (2) is simply a preconditioner for an iterative method. Even then, the memory storage required for a large-size problem is rather prohibitive.[8,10] In the present work, this is circumvented by resorting to a first-degree iterative scheme, where the true Jacobian matrix is split into more memory-manageable parts using the *approximate factorization* followed by the ADI scheme.[12]

The employed optimization algorithm is based on the *method of feasible directions* which is coded in ADS.[17] This method requires the first-order sensitivity gradients of both the objective function and the constraints. In the present formulation, these gradients are evaluated analytically. The analytic gradient of, for example, the objective function (similarly, it can be written for the constraints) is expressed by

$$\nabla F = \frac{\partial F(D,Q)}{\partial D_i}\hat{e}_i = \left[\left(\frac{\partial F}{\partial D_i}\right)_Q + \left(\frac{\partial F}{\partial Q}\right)_D^T Q_i'\right]\hat{e}_i, \quad i \in \{1,\ldots,NDV\} \tag{3}$$

where $Q_i' \equiv \partial Q / \partial D_i$.

For the discrete sensitivity analysis, either the direct method or the adjoint-variable method,[4-7] given below, respectively, is used.

$$\left(\frac{\partial R}{\partial Q}\right)_D Q' = -\left(\frac{\partial R}{\partial D}\right)_Q \tag{4}$$

$$\left(\frac{\partial R}{\partial Q}\right)_D^T \lambda_F = \left(\frac{\partial F}{\partial Q}\right)_D \tag{5}$$

The direct formulation renders Q' explicitly, which allows the use of approximate flow analyses,[6] but the number of equations to be solved scales with the number of design variables. The number of equations to be solved in the adjoint formulation, however, scales with the number of aerodynamic constraints plus one for the objective.

Following the same solution options considered for the analysis equation (1), the sensitivity equation, (4) or (5), is written in the delta form as shown below, respectively,

$$\left(\frac{\partial \hat{R}}{\partial Q}\right)_D^n \{\Delta Q'\}^n = -\left(\frac{\partial R}{\partial Q}\right)_D^n Q'^n - \left(\frac{\partial R}{\partial D}\right)_Q \tag{6}$$

$$\left(\frac{\partial \hat{R}}{\partial Q}\right)^T \{\Delta\lambda_F\}^n = -\left(\frac{\partial R}{\partial Q}\right)_D^T \{\lambda_F\}^n - \left(\frac{\partial F}{\partial Q}\right)_D \tag{7}$$

Here, $\partial \hat{R} / \partial Q$ is an approximation to the true Jacobian that is used as a preconditioner. Note that the delta form of the adjoint equation for the constraints can be written similar to Eq. (7). Then, these equations are either solved using the PCG approach, or they are approximately factored, then solved by the ADI scheme. With the addition of a relaxation term for diagonal dominance, the ADI scheme is written as

$$\left(\frac{I}{\omega} + \frac{\partial \hat{F}}{\partial Q}\right)^n \Delta Q'^* = -\left(\frac{\partial R}{\partial Q}\right)_D^n \Delta Q'^n - \left(\frac{\partial R}{\partial D}\right)_Q ,$$

$$\left(\frac{I}{\omega} + \frac{\partial \hat{G}}{\partial Q}\right)^n \Delta Q'^{**} = \left(\frac{I}{\omega}\right)\Delta Q'^* ,$$

$$\left(\frac{I}{\omega} + \frac{\partial \hat{H}}{\partial Q}\right)^n \{\Delta Q'\}^n = \left(\frac{I}{\omega}\right)\Delta Q'^{**} . \tag{8}$$

The relaxation parameter, ω, may be set to $\Delta\tau$, where τ is a time-like variable, or other conventional options may be used.[14] For simplicity, the coefficient matrices in Eq. (8) are constructed using the first-order-accurate upwind scheme, and the consistent linearization of the boundary conditions is neglected.

Wing Shape Parameterization

The wing geometry used for demonstration herein is generated using the parameterization adopted from Burgreen and Baysal.[14] This wing is generated from the geometrically simple wing which is unswept, untwisted, uncambered and rectangular, with both its chord and span equal to unity; hence, this wing is referred to as the *unit wing*. To generate a variety of shapes, the geometric parameterization of a wing should allow flexibility in its sections, taper distribution, sweep, span, spanwise bending, geometric twist, and global angle-of-attack. In the present parameterization (fig. 1), each feature is implemented as an independent geometric operation in a sequential manner. Its details can be found from Burgreen and Baysal.[14]

Once a new wing shape and its surface grid are defined, the next task is the regeneration of its volume grid. This process of regridding enroute to the optimized shape is completed using the *flexible grid* approach,[14] where the volume grid is written as a function of the body surface grid (X_b). As the surface is changed during the shape optimization, the x-coordinates, for example, of the new volume grid are found using the following relations:

$$x_i^{new} = x_i^{old} + [1 - v_j](x_b^{new} - x_b^{old}), \tag{9}$$

where $v_j = \frac{s_j - s_2}{s_{j\max} - s_2}$,

$$s_j = \sum_{i=2}^{j} \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2} . \tag{10}$$

Results

Validation of ADI approach

To test the presently developed ADI methodology, an *arrow wing* (Table 1) was considered to be at 3-deg aoa to an oncoming Mach 2.4 flow. This wing was generated from the unit wing by linear distributions of `chdscal` and `thkscal` along the span as schematically shown in fig. 1. The entire wing was behind the Mach cone. The coarse grid was used in order to accommodate the memory-intensive PCG solutions within reasonable computer resources, as well as for the grid refinement studies.

The flowfield was computed on the coarse grid using the present ADI and PCG methods as well as CFL3D[18] (Version 3.0). The results of these analyses were successfully compared (fig. 2) via their chordwise pressure coefficient distributions at three spanwise sections. The drag values of CFL3D and ADI (7.6e-3 vs 7.8e-3) matched to the third digit and, for the ADI and PCG solutions, the lift and drag values matched up to five significant digits (Table 2).

As it is necessary for quasi-Newton methods, the PCG method was initialized with the solution to a $M_\infty = 2.35$ flow, obtained using the ADI method. Then, the convergence to the M_∞=2.4 solution was timed for both of the methods (Table 2). As expected, by using ADI, the storage was reduced by a factor of about six compared to the PCG, even for this coarse grid. The initialization, apparently, was not in the domain of attraction to the root, hence, the convergence time of PCG was also unexpectedly higher.

In fig. 3, the chordwise pressure coefficient distributions, obtained on the coarse and fine grids with the ADI method, are presented for three sections. Although the coarse-grid flowfield appeared plausible with its salient features, improvement in the solution due to grid refinement was clearly observed: with coarse-grid computations, the pressure peaks were overestimated in the root section, yet they were underestimated in the mid and tip sections. The surface pressures obtained on the fine grid (fig. 4) were deemed satisfactory based on the general flowfield features, such as, the crisper definition of the upper surface shock. A more conclusive evidence to an almost grid-independent solution, however, would have required another level of finer grid solution which would differ only insignificantly, if at all, from the current fine-grid solution.

Next, the sensitivities were computed by the PCG and ADI methods using the direct differentiation (eqs. 2.4 and 2.8) on the coarse grid. The primary reason for the computations on the coarse grid was naturally the significant cost savings. Note that the point of this exercise was a fair comparison of the sensitivities from different methods, and not their absolute accuracy. Most of the values matched to the fourth significant digit (Table 3). As for the efficiencies, this comparison essentially epitomized the characteristics of both the methods. The unfactored solution procedure based on PCG method, when in the domain of attraction to its root, was very efficient from CPU time standpoint. Whereas, the factored solution procedure (ADI) could only achieve the linear convergence, but it was very efficient from the memory viewpoint. The results shown in Table 3 were obtained using a convergence criterion of five orders of magnitude reduction in the L_2-norm residual. As this much reduction may not always be necessary during an optimization, the effect of the order of magnitude on the efficiency and accuracy was investigated (Table 4). Each two orders of reduction increased the CPU time by a factor of about 1.6. Generally, the change in the values, when the residual was dropped from five orders to seven orders, was in the fifth or higher significant digit. Finally, ADI based sensitivities, obtained for the convergence criterion of three order of reduction, were compared with those by the finite difference method[4,6] and they were found to be in excellent agreement (Table 5).

Shape Optimization

The primary thrust of the present study was to apply the gradient-based optimization technique for 3D shapes, which often require large size grids. The challenge was to perform such a task using only a feasible amount of computer memory, despite the fact that higher grid densities than what has been used previously,[14,15] were needed. As a demonstration, the present method was implemented to optimize a supersonic arrow wing configuration (Table 1 and fig. 4), first on a coarse grid then on a fine grid.

The present problem formulation had nine design variables (`spn`, and at mid and tip sections: `thkscal`, `chdscal`, `tranx` and `twst`). The objective function was selected to be the maximization of C_L/C_D, subject to fourteen geometric and two aerodynamic inequality constraints:

$$C_L \geq 0.11,\ C_D \leq 0.008,\ V_{wing} \geq 0.9V_{wing}^{initial}$$

$$A_{midspan} \geq 0.6A_{midspan}^{initial}. \qquad (10)$$

At the root, mid, and tip sections:

$$2^0 \leq \theta_{0.90chord} \leq 20^0, \qquad 2^0 \leq \theta_{0.90chord} \leq 20^0 \qquad (11)$$

The volume constraint prevented the wing from becoming too thin whereas the θ constraint ensures that the trailing edge does not become too sharp or too blunt. Also, side (equality) constraints were imposed on the design variables; for example, `spn` was tightly bounded to avoid the optimizer seeking higher lift by simply increasing the wing span. The selected values of the design variables allowed for the formation of a planform break in the chord distribution. Also, the thickness, sweep and geometrical twist were allowed to have linear spanwise distributions. The airfoil section at the wing root was elected to remain unchanged. For both of the cases, C_L, θ_{tip} at 90% chord, and V constraints were violated in the final design. Additionally, for the fine grid optimization, $A_{midspan}$ constraint was also violated.

The optimization resulted in a *cranked* arrow wing (figs. 5 and 6). The primary change was observed in the enlarged tip chord, the planform break, reduced thickness and the increased geometric twist. A summary of the optimized wing's geometry and aerodynamics is given in Table 6 (for initial wing, see Tables 1 and 2). The surface pressures of the optimized wing are presented in fig. 7, which may be contrasted with fig. 4. The objective function histories during the evolution to the optimized shape are presented in fig. 8. The trends appear similar for the coarse and the fine grid cases, but the values are distinctly different and the coarse grid shape converged in fewer iterations. Finally, Table 7 presents some of the demographics for these optimization cases. It should be noted that the PCG method, had it been used for this fine-grid shape optimization, would require 164 MW memory.

Conclusions

A gradient-based shape optimization methodology, that is intended for reasonably practical 3D aerodynamic applications, has been developed. To accommodate the inherent larger size of such problems, the fluid dynamic and the aerodynamic sensitivity equations were solved using the ADI algorithm for memory efficiency.

The flow analysis for an arrow wing at Mach 2.4 compared well with those obtained using an unfactored approach (PCG), and an extensively used CFD code. The sensitivities of the present method have also compared well with those obtained using the PCG approach and finite difference approach. As expected the ADI method reduced the

storage memory but increased the computing time as compared to the PCG method. It should be mentioned that, although initial flowfield analysis for the PCG method appeared to be more time intensive as compared to ADI method (Table 1), in a typical optimization cycle, two successive designs are only incrementally different and this feature would be very much amenable for the efficacy of PCG to render CPU-time efficient designs. The new procedure has been demonstrated in the design of a cranked arrow wing. Effects of grid refinement and convergence tolerance on the analyses as well as the shape optimization results have been explored.

The results indicate that shape optimization problems which require large numbers of grid points can be resolved with a gradient-based approach. Therefore, to better utilize the computational resources, it is recommended that a number of coarse grid cases, using the PCG method, should initially be conducted to better define the optimization problem and the design space, and obtain an improved initial shape. Subsequently, a fine grid shape optimization should be conducted, using the ADI method, to accurately obtain the final optimized shape.

References

1. Hicks, R. M., and Henne, P.A. "Wing Design by Numerical Optimization," AIAA Paper 77-1247, AIAA Aircraft Systems & Technology Meeting, Seattle, WA, Aug. 1977.

2. Ibrahim, A.H., and Baysal, O., "Design Optimization Using Variational Methods and CFD," AIAA Paper 94-0093, AIAA 32nd Aerospace Sciences Meeting, Reno, NV, Jan. 1994.

3. Reuther, J., and Jameson, A., "Supersonic Wing and Wing-Body Shape Optimization using an Adjoint Formulation," *CFD for Design and Optimization* (Ed. O. Baysal), ASME FED-Vol. 232, pp. 45-52, International Mechanical Engineering Conference and Exposition, November 1995.

4. Baysal, O., and Eleshaky, M.E., "Aerodynamic Sensitivity Analysis Methods For the Compressible Euler Equations," *Journal of Fluids Engineering*, Vol. 113, No. 4, Dec. 1991, pp. 681-688.

5. Huan, J., and Modi, V., "Design of Minimum Drag Bodies in Incompressible Laminar Flow," *CFD for Design and Optimization* (Ed. O. Baysal), ASME FED-Vol. 232, pp. 37-44, International Mechanical Engineering Conference and Exposition, November 1995.

6. Baysal, O., and Eleshaky, M.E., "Aerodynamic Design Optimization Using Sensitivity Analysis and Computational Fluid Dynamics," *AIAA Journal*, Vol. 30, No. 3, Mar. 1992, pp. 718–725.

7. Burgreen, G.W., Baysal, O., and Eleshaky, M.E., "Improving the Efficiency of Aerodynamic Shape Design Procedures," *AIAA Journal,* Vol. 32, No. 1, Jan. 1994, pp. 69-76.

8. Burgreen, G.W., Baysal, O., "Aerodynamic Shape Optimization Using Preconditioned Conjugate Gradient Methods," *AIAA Journal*, Vol. 32, No. 11, Nov. 1994, pp. 2145-2152.

9. Eleshaky, M.E. and Baysal, O., "Aerodynamic Shape Optimization via Sensitivity Analysis on Decomposed Computational Domains," *Journal of Computers and Fluids,* Vol. 23, No. 4, May 1994, pp. 595-611.

10. Eleshaky, M.E., and Baysal, O., "Preconditioned Domain Decomposition Scheme for 3-D Aerodynamic Sensitivity Analysis," *AIAA Journal*, Vol. 32, No. 12, Dec. 1994, pp. 2489-2491.

11. Eleshaky, M.E., and Baysal, O., "Design of 3-D Nacelle Near Flat-Plate Wing Using Multiblock Sensitivity Analysis (ADOS)," AIAA Paper 94-0160, AIAA 32nd Aerospace Sciences Meeting, Reno, NV, Jan. 1994. Also, to appear in *Journal of Aircraft.*

12. Beam, R.M. and Warming, R.F., "An Implicit Factored Scheme for the Compressible Navier-Stokes Equations," *AIAA Journal*, Vol. 16, April 1978, pp. 393-402.

13. Korivi, V. M., Taylor, A. C., III, Hou, G. W., Newman, P. A., and Jones, H. E., "Sensitivity Derivatives for Three-Dimensional Supersonic Euler Code Using Incremental Iterative Strategy," *AIAA Journal*, Vol. 32, No. 6, June 1994, pp. 1319-1321.

14. Burgreen, G.W., and Baysal, O., "3-D Aerodynamic Shape Optimization Using Discrete Sensitivity Analysis," *AIAA Journal,* Vol. 34, No. 9, Sept. 1996, pp. 1761-1770.

15. Burgreen, G.W., and Baysal, O., "Three-Dimensional Aerodynamic Shape Optimization of Delta Wings," Paper 94-4271 CP, *Proceedings of 5th AIAA/USAF/NASA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Panama City, FL, Sept. 1994, pp. 87-97.

16. Narducci, R., Grossman, B., Valorani, M., Dadone, A., and Haftka, R.T., "Optimization Methods for Non-Smooth or Noisy Objective Functions in Fluid Design Problems," Paper No. 95-1648 CP, *Proceedings of 12th AIAA Computational Fluid Dynamics Conference*, pp. 21-32, San Diego, CA, June 1995.

17. Vanderplaats, G.N., *ADS - A FORTRAN Program for Automated Design Synthesis - Version 1.10*, NASA Contractor Report 177985, Sept. 1985.

18. Thomas, J.L., Krist, S.T., and Anderson, W.K., "Navier-Stokes Computations of Low Aspect Ratio Wings," *AIAA Journal,* Vol. 28, No. 2, Feb. 1990, pp. 205-215.

Acknowledgment

This research was supported by NASA Langley Research Center under Grant No. NCC-1-211. The technical monitor was Dr. James L. Thomas.

Tables

Table 1. Geometry of initial arrow wing

Λ_{LE} and Λ_{TE} (deg)	Aspect ratio	$(t/c)_{root,mid,tip}$ (%)	θ_{tip} (deg)	Half-span/c_{root}
72 - 56.3	1.01	4.80, 4.55, 4.80	6.80	0.57

Table 2. Efficiency and accuracy comparisons for CFD analyses*

	PCG	ADI	
	Coarse Grid (43x15x9)#	Coarse Grid (43x15x9)	Fine Grid (127x43x25)
Iterations to convergence	100	241	327
CPU time (sec)	640**	283	8138
Memory (MW)	10.61	2.25	42.11
C_L	8.4815e-02	8.4814e-02	8.7349e-02
C_D	7.8270e-03	7.8270e-03	7.6629e-03
(C_L/C_D)	10.8362	10.8362	11.40

*Computed flowfield for M_∞=2.4;
**Initialized with converged ADI solution of M_∞=2.35 flow
#In circumferential, normal, and spanwise directions, respectively

Table 3. Comparison of quasi analytical sensitivities*

	ADI	PCG
Iterations to convergence	112	9
CPU time (sec)	344	45
Memory (MW)	2.60	10.61
$\partial C_L/\partial(chd_{tip})$	0.0217	0.0217
$\partial C_L/\partial(tranx_{mid})$**	-0.0142	-0.0142
$\partial C_L/\partial(spn)$	0.0439	0.0439
$\partial(C_L/C_D)/\partial(tranx_{mid})$	2.5124	2.5120
$\partial(C_L/C_D)/\partial(tranx_{tip})$	2.7549	2.7549
$\partial(C_L/C_D)/\partial(spn)$	-10.4890	-10.4881

*Computed flowfield for M_∞=2.4 on 43x15x9 C-H grid; direct method (eqs. 4 and 8) with $\vartheta(5)$ convergence.
**Parameters as defined in fig. 1.

Table 4. Effect of convergence tolerance on efficiency and accuracy of sensitivities for ADI method*

	$\vartheta(3)$	$\vartheta(5)$	$\vartheta(7)$
Iterations to convergence	65	112	163
CPU time (sec)	204	344	498
$\partial C_L/\partial(chd_{tip})$	0.0217	0.0217	0.0217
$\partial C_L/\partial(tranx_{mid})$	-0.0142	-0.0142	-0.0142
$\partial C_L/\partial(spn)$	0.0439	0.0439	0.0439
$\partial(C_L/C_D)/\partial(chd_{mid})$	7.0183	7.0186	7.0186
$\partial(C_L/C_D)/\partial(tranx_{mid})$	2.5128	2.5124	2.5124
$\partial(C_L/C_D)/\partial(tranx_{tip})$	2.7547	2.7549	2.7549
$\partial(C_L/C_D)/\partial(spn)$	-10.4889	-10.4890	-10.4890

*Computed flowfield for M_∞=2.4 on 43x15x9 C-H grid; direct method (eq. 8).

Table 5. Comparison of ADI (quasi analytical) and finite difference sensitivities*

	Quasi Analytical**	Finite Difference	$\frac{\text{Quasi Analytical}}{\text{FiniteDifference}}$
$\partial C_L/\partial(chd_{tip})$	0.0217	0.0216	1.0055
$\partial C_L/\partial(twst_{crank})$	0.0124	0.0124	0.9993
$\partial C_D/\partial(thk_{tip})$	0.0007	0.0007	0.9956
$\partial C_D/\partial(tranx_{crank})$	-0.0031	-0.0031	1.0006
$\partial C_D/\partial(twst_{crank})$	0.0014	0.0014	1.0005
$\partial(C_L/C_D)/\partial(chd_{tip})$	2.3543	2.3571	0.9988
$\partial(C_L/C_D)/\partial(twst_{crank})$	-0.4108	-0.4091	1.0065
$\partial(C_L/C_D)/\partial(spn)$	-10.4889	-10.4829	1.0006

*Computed flowfield for M_∞=2.4 on 43x15x9 C-H grid.
**Direct method (eq. 8) with $\vartheta(3)$ convergence.

Table 6. Aerodynamics and geometry of optimized wing

	Coarse Grid (43x15x9)	Fine Grid (127x43x25)
C_L	9.5675e-02	9.9113e-02
C_D	7.8516e-03	7.4940e-03
C_L/C_D	12.1855	13.2257
Λ_{LE} (deg)	71.9 - 67.0	71.9 - 67.6
Aspect ratio	0.8701	0.9214
$(t/c)_{root,mid,tip}$ (%)	4.71, 3.07, 1.44	4.71, 2.32, 1.51
$(twst)_{root,mid,tip}$ (deg)	0.00, 0.50, 0.12	0.00, 0.50, 0.24
$\%A_{mid}^{initial}$	68.7	52.0
$\%V_{wing}^{initial}$	88.6	80.4

Table 7. Demographics from optimization cases*

	Coarse Grid (43x15x9)	Fine Grid (127x43x25)
Lift change (%)	12.8	13.5
Drag change (%)	-1.26	-2.22
C_L/C_D change (%)	16.1	12.5
1-D searches	69	88
Gradient evaluations	5	9
Memory (MW)	2.93	48.37
CPU time (h)	0.53	38.1

*Direct sensitivity method (eq. 8) with $\vartheta(3)$ convergence.

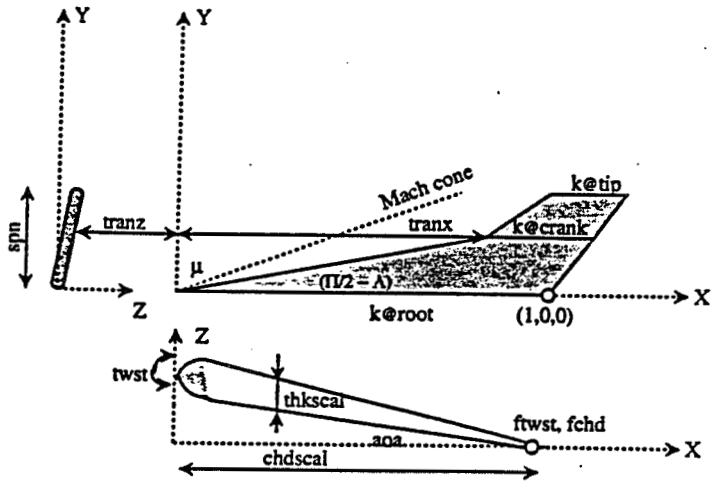
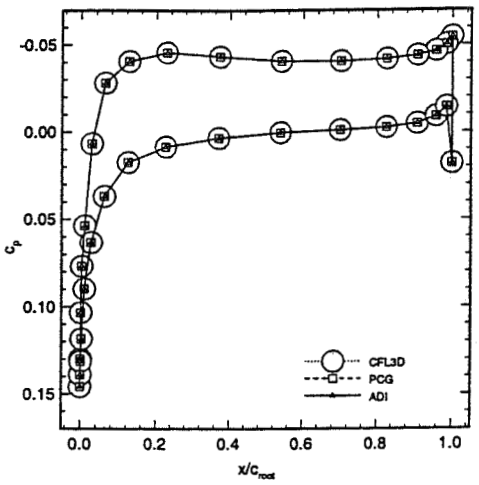
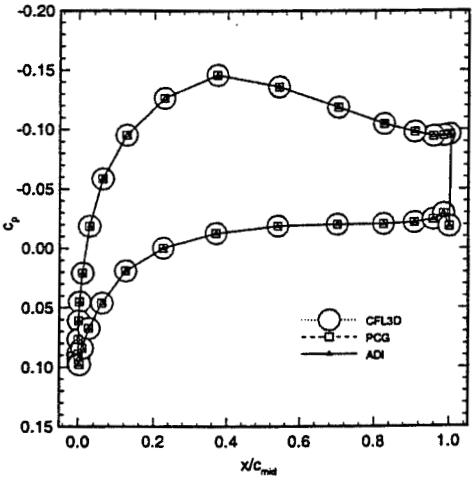


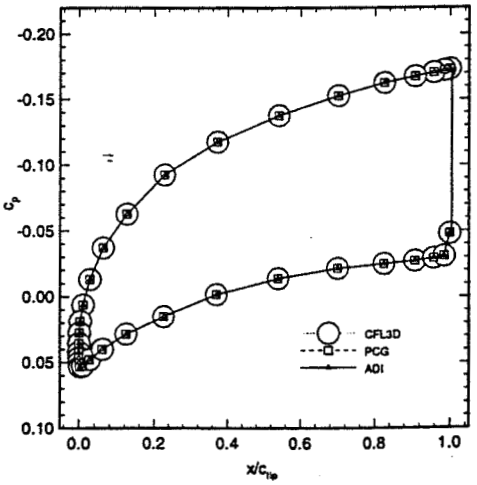
Figure 1: Parameterization of a wing.



Root Section

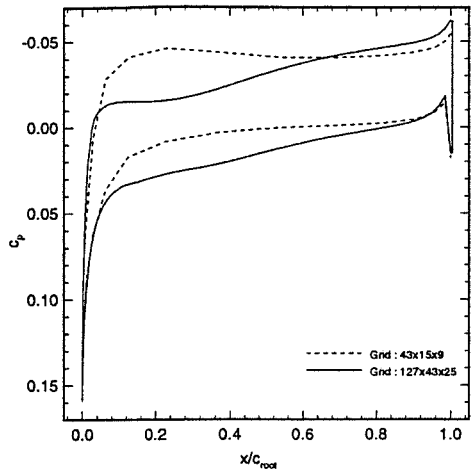


Mid Section

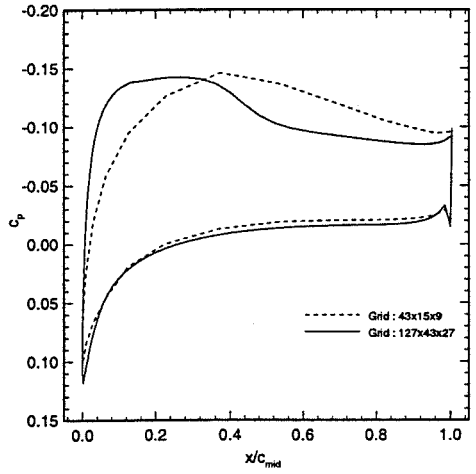


Tip Section

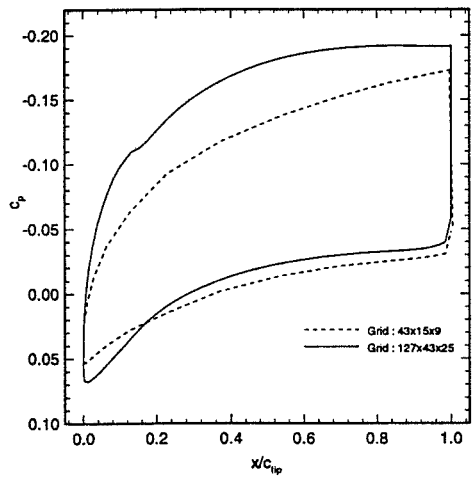
Figure 2: Comparison of chordwise pressure coefficients on coarse grid (43x15x9). M=2.4, aoa=3-deg.



Root Section



Mid Section



Tip Section

Figure 3: Comparison of chordwise pressure coefficient distributions by ADI method on fine and coarse grids. M=2.4, aoa=3-deg.

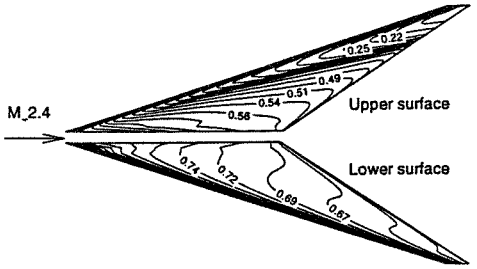
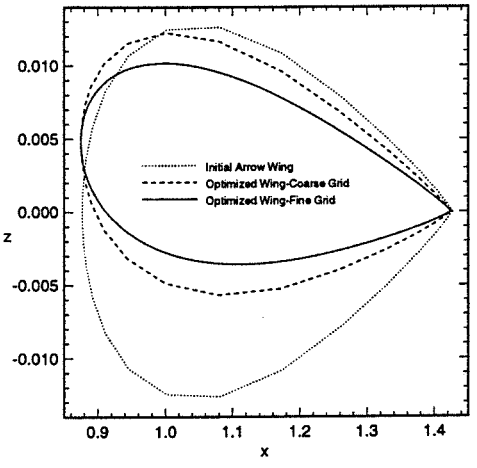
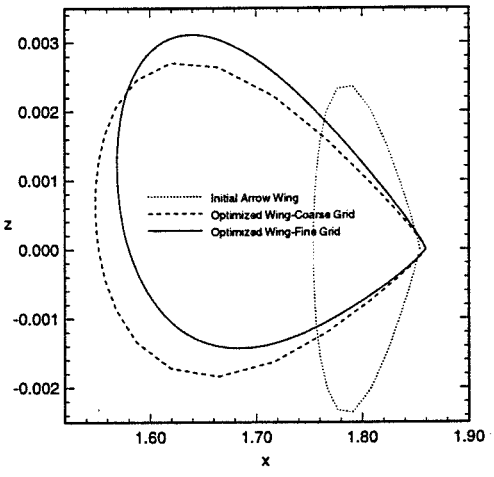


Figure 4: Normalized pressure contours by ADI method on fine grid (127x43x25). M=2.4, aoa=3-deg.



Mid Section



Tip Section

Figure 5: Evolution of wing sections from initial to coarse- and fine-grid optimized shapes.

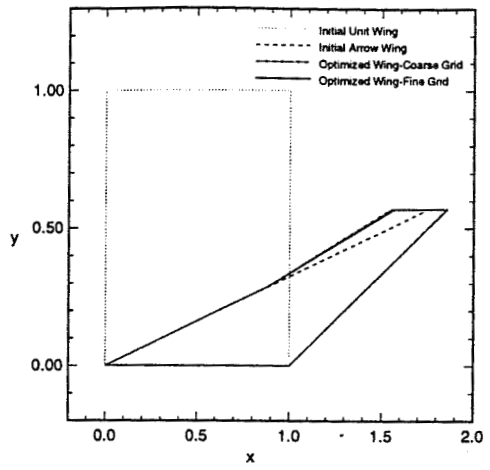


Figure 6: Evolution of wing planform shape.

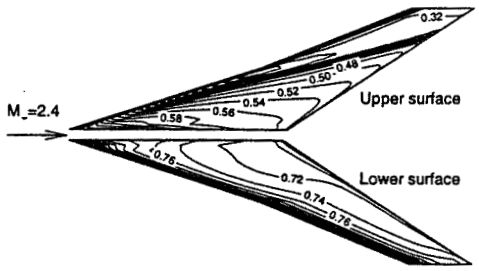


Figure 7: Normalized pressure contours on optimized arrow wing obtained by ADI method on fine grid (127x43x25).

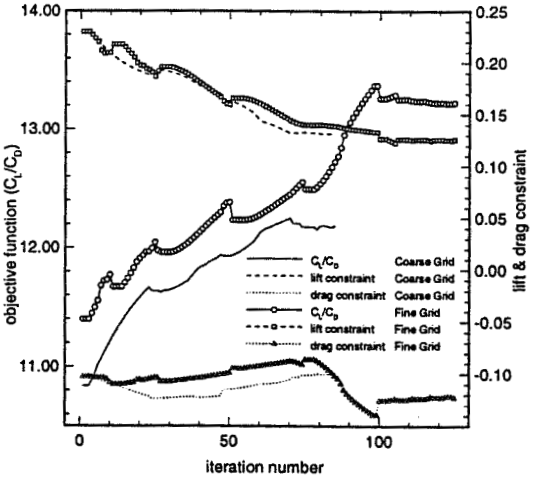


Figure 8: Histories of objective function and aerodynamic constraints.

The Japan Society of Mechanical Engineers
JSME Centennial Grand Congress

IMPROVING DISCRETE-SENSITIVITY-BASED APPROACH FOR PRACTICAL DESIGN OPTIMIZATION

Oktay Baysal, Yvette Cordero, and Mohagna J. Pandya

Aerospace Engineering Department
Old Dominion University
Norfolk, Virginia 23529-0247 USA

Preprint from the Proceedings

International Conference on Fluid Engineering
July 13-16 1997 / Tokyo, JAPAN

IMPROVING DISCRETE-SENSITIVITY-BASED APPROACH FOR PRACTICAL DESIGN OPTIMIZATION

Oktay Baysal,* Yvette Cordero** and Mohagna J. Pandya**

Aerospace Engineering Department
Old Dominion University
Norfolk, Virginia 23529-0247 USA

ABSTRACT

In developing the automated methodologies for simulation-based optimal shape designs, their accuracy, efficiency and practicality are the defining factors to their success. To that end, four recent improvements to the building blocks of such a methodology, intended for more practical design optimization, have been reported. First, in addition to a polynomial-based parameterization, a partial differential equation (PDE) based parameterization was shown to be a practical tool for a number of reasons. Second, an alternative has been incorporated to one of the tedious phases of developing such a methodology, namely, the automatic differentiation of the computer code for the flow analysis in order to generate the sensitivities. Third, by extending the methodology for the thin-layer Navier-Stokes (TLNS) based flow simulations, the more accurate flow physics was made available. However, the computer storage requirement for a shape optimization of a practical configuration with the high-fidelity simulations (TLNS and dense-grid based simulations), required substantial computational resources. Therefore, the final improvement reported herein responded to this point by including the alternating-direction-implicit (ADI) based system solver as an alternative to the preconditioned biconjugate gradient (PbCG) and other direct solvers.

INTRODUCTION

When the symposium on *Multidisciplinary Applications of Computational Fluid Dynamics* [1] was organized, it included only a few papers reporting on the utilization of computational fluid dynamics (CFD) in a design environment. Since then, however, there has been an alluring eruption of interest on this topic for a justifiable reason. CFD can now be utilized for the purpose of design and optimization in order to cut down the cycle time for a new product design. This, however, may be a prohibitive proposition for the necessary computational and human resources if a large matrix of candidate designs or design variables are involved.

Therefore, a few years later and with the motivation of providing a podium for beyond the cut-and-try approaches, where CFD would be a module of an automated methodology in search of improved designs, the forum on *CFD for Design and Optimization* [2] was organized. This forum attested to the fact, that the researchers were now developing new methods, varying from inverse to the direct numerical optimization methods. Some of the objectives that such efforts were trying to accomplish were: **i)** Automated design optimization with minimal need for the man-in-loop in a given try; **ii)** Reduce the need for the design expertise and the prerequisite database; **iii)** Information on the most influencing parameters, which leads to a reduced CFD analysis matrix, requires sensitivities of the objectives and constraints; **iv)** Ability to design with a variety of fluid dynamic and geometric constraints, and with geometric flexibility; finally, **v)** Improve accuracy, efficiency and practicality, which also constitutes the motivation for the present paper.

The approaches to the engineering design may be categorized as: **1)** Non-automated design, **2)** Inverse design, and **3)** Direct numerical optimization, methods. As the present paper reports on a methodology which belongs to the last category, it will be the only one briefly introduced next.

Direct numerical optimization [3] is a systematic methodology which extremizes a chosen objective. It may produce the *best* design or just a *better* design, dependent on if the global or just a local minimum has been found. However, it can be rather cumbersome to develop and its usage may be computing intensive. A *constrained optimization* is gradient based ($\nabla F, \nabla G_j$), and it relies on the deterministic searches, $(D_i^{n+1} = D_i^n + \alpha^n S_i^n)$, $i = 1, NDV$. Its mathematical formulation consists of the extremization of an objective function $F\left(D_i, Q(D_i)\right)$, subject to satisfying the analysis equations, $R_m\left(D_i, Q(D_i)\right)=0$, $m = 1, NEQ$, fluid dynamic or geometric inequality constraints, $G_j\left(D_i, Q(D_i)\right) \le 0$, $j = 1, NCON$, fluid dynamic or geometric equality constraints $H_k(D_i, Q(D_i)) = 0$, $k = 1, NCON_e$, and finally the side constraints, $D_i^{lower} \le D_i \le D_i^{upper}$.

One type of a simulation-based and automated design methodology, which uses a constrained optimization, is the fluid dynamic shape optimization. This typically consists of the following components [3,17]: **(i)** formulation of the problem, i.e. identifying the objective and the constraint functions; **(ii)** gradient-based and constrained optimization algorithm; **(iii)** flowfield simulation, i.e. CFD; **(iv)** parameterization and redefinition of shapes as they evolve, i.e. using computer-aided-design (CAD); **(v)** regeneration of the surface and volume grids, and the grid sensitivities; **(vi)** gradients of the objective and the constraints with respect to the design variables, i.e. the sensitivity coefficients; (vii) gradients of the optimized shape with respect to the design invariants, i.e. the post-optimization sensitivity derivatives, used for trade-off studies and off-design conditions.

The present paper will focus on the four improvements to this methodology that have recently been investigated in order to attain a more practical design optimization methodology: **1)** PDE based surface parameterization and generation, **2)** automatic differentiation (AD) to generate the code for the sensitivities, **3)** ADI scheme for the sensitivity and analysis equations, and **4)** shape optimization based on the TLNS simulations. Consequently, of the above mentioned methodology components, only **(iv)** through **(vi)** will be explained in the following sections. However, for the sake of

* Professor and Eminent Scholar. Fellow, ASME.
** Graduate Research Assistant.

completeness, three important building blocks will be summarized next.

Flowfield simulation by CFD

The three-dimensional, compressible, thin-layer Navier-Stokes equations (TLNS), with an option to drop the diffusion terms when the Euler equations would suffice, are written in the conservation form and generalized curvilinear coordinates [14,18,20,23]. They are discretized by a cell-centered, finite volume method, where the flux vectors and the Jacobian matrices are evaluated using the van Leer flux-vector splitting, with the third-order MUSCL extrapolations to the cell interfaces and the van Albada limiter [22,23].

Optimization algorithm

The feasible-directions method [25], which is a gradient-based and constrained optimization algorithm, is used as the driver of the overall methodology [5,6]. Within the feasible region, deterministic search algorithms, such as, a univariate search strategy (a sequential one-dimensional minimization in a multidimensional design space) with a variable step size using zeroth-order methods, or simply a constant step size, are employed. Various stopping criteria are used for the optimization and for the flow analysis, where either the convergence tolerances of different orders of magnitude, or the number of iterations, or both, are specified.

Algebraic systems

For the large systems of linear algebraic equations, resulting from the sensitivity equation or the flowfield equations, there are several inversion options in, both the direct and the iterative approaches, as they all have their distinct niches depending on the problem at hand. These are: **(i)** naive Gauss elimination with banded storage [5,6], **(ii)** sparse-matrix method with symbolic factorization [5,6], **(iii)** a first-order iterative method, such as, solving the delta-form of the equations by an ADI method [22], **(iv)** a preconditioned biconjugate gradient method (PbCG), such as, the restarting version of GMRES(k, m), with several fill-in options as the preconditioners [10,16].

SURFACE PARAMETERIZATION

A non-parametric geometry definition requires the surface mesh point coordinates. Although this approach utilizes data readily available for CFD, it increases the number of design variables. On the other hand, a parametric geometry definition reduces the number of design variables, since the number of control points is significantly less than the surface mesh points. Current geometry definition techniques used in computer-aided design (CAD) are capable of parameterizing space curves and space surfaces with high fidelity. They do require extra computations, storage, and some expertise. Among the desirable properties in choosing a particular parameterization technique, are their generality and differentiability.

As an example, the Bezier-Bernstein approach [11] is considered to represent a surface given by its grid points $X\{x_i,y_i,z_i\}, i=1,2,..,I$. Then, an (N,M)-th degree representation $(N<<I, M<<I)$ is obtained from a bi-directional spline frame:

$$X_{j,k}(u,v)=\sum_{n=0}^{N}\sum_{m=0}^{M} P_{nm}\,B_n(u)B_m(v) \quad , \qquad (1)$$

where $0\le u,v\le 1$ and

$$B_n(u)=\binom{N}{n}u^n(1-u)^{N-n} \quad , \quad B_m(v)=\binom{M}{m}v^m(1-v)^{M-m} \qquad (2)$$

After a single-indexed ordering $[i=1,2,..,(n\cdot m)]$ of the control points, the solution requires forming a tensor product of the 1-D basis functions, $B_n(u), B_m(v)$.

In defining a configuration which includes the CAD-parameterized design surfaces, parameters that are more intuitively recognizable to the specific discipline may be preferred. For example, a wing may be described in part by its sweep, span, spanwise bending, geometric twist, spanwise distributions of taper and section chords and thicknesses [12]. Once a new configuration and its design surfaces are defined, the next task is the re-generation of its surface and volume grids. Hence, efficient approaches are developed to analytically relate the surface parameters to the computational volume grid. Then, the grid is re-generated systematically using these relations, as the vector of design variables is changed with the evolving shape. For example [12],

$$X_i^{new}=X_i^{old}+[1-v_j](X_b^{new}-X_b^{old}) \quad , \quad v_j=\frac{s_j-s_2}{s_{j\max}-s_2} \quad , \qquad (3)$$

$$s_j=\sum_{i=2}^{j}\sqrt{(x_i-x_{i-1})^2+(y_i-y_{i-1})^2+(z_i-z_{i-1})^2} \quad . \qquad (4)$$

An alternative is to build on the experiences with the PDE approaches that have long been used for the grid generation. Drawing from the smoothing properties of the Poisson equation, $\nabla^2\phi=Q$, the method is very effective in creating smooth surfaces that act as the transition blocks between neighboring surfaces, i.e. the blend generation. One such approach [9] has recently been incorporated into the airplane design research [24] as well as into the present methodology. The mapping from the physical to the computational space,

$$X(\xi,\eta)=X[x(\xi,\eta),y(\xi,\eta),z(\xi,\eta)] \quad , \qquad (5)$$

is performed by a fourth-order equation with a weighting parameter,

$$[a^2\frac{\partial^2}{\partial\xi^2}+\frac{\partial^2}{\partial\eta^2}]^2\,X(\xi,\eta)=0 \quad . \qquad (6)$$

However, rather than solving the equations by the usual iterative techniques, the solution is expressed as an approximate Fourier series representation, with the coefficients obtained from the Dirichlet (D) and Neumann (N) boundary conditions. For example, at a $\eta=[0,1]$ boundary,

$$X(\xi,0)=D_0(\xi) \, , \; X(\xi,1)=D_1(\xi) \quad , \qquad (7)$$

$$\frac{\partial X}{\partial\eta}(\xi,0)=N_0(\xi) \, , \; \frac{\partial X}{\partial\eta}(\xi,1)=N_1(\xi) \quad . \qquad (8)$$

D_0, D_1 and N_0, N_1 are then specified as in eq. (5), but with case-specific relations, $\bar{x}_i(\xi^j)$ and $\frac{\partial\bar{x}_i}{\partial\xi^j}$, respectively, where $i=1,2,3$ and $j=1,2$. These relations are determined with the parameters of the particular geometry to be generated. By selecting the appropriate boundary conditions, the desired surfaces, such as, a wing, a fuselage, or a nacelle, can be generated; then, they are blended via, e.g. fillets or pylons, to obtain a wing-body, or a conceptual aircraft configuration. Further, the output is a discrete set of points arranged as a surface mesh $X(i,j)$. The grid spacing parameters are also

included in the usual way through the non-homogeneous part of the Poisson equation and the boundary conditions.

SENSITIVITY ANALYSIS

The accuracy of a gradient-based optimization method, and the efficiency with which it can accomplish this, are directly related to the accurate and efficient receipt of the gradient information on the objectives and the constraints. The sensitivity analysis provides: **i)** the gradients ($\nabla F, \nabla G_j$) for the optimization and trade-off studies; **ii)** the post-optimization gradients for off-design conditions; **iii)** a first-order but inexpensive approximation to neighboring-point analysis via Taylor series expansion [7]. The gradient evaluation may be performed by either a finite-difference method [5], or a continuous adjoint method [3], or a discrete sensitivity method [5,6]. The discrete sensitivity analysis (a.k.a. quasi-analytical) is performed by differentiating the already CFD-discretized equations with respect to the design variables [15]. It corresponds to a discrete solution of the continuous sensitivity function. Since this approach is used in the present methodology, it will be further discussed next.

Discrete sensitivity analysis

The sensitivities needed by the optimization can be obtained by the chain rule of differentiation,

$$\nabla F = \left[\left(\frac{\partial F}{\partial D_i} \right)_Q + \left(\frac{\partial F}{\partial Q} \right)_D^T \cdot Q_i' \right] , \tag{9}$$

$$\nabla G_j = \left[\left(\frac{\partial G_j}{\partial D_i} \right)_Q + \left(\frac{\partial G_j}{\partial Q} \right)_D^T \cdot Q_i' \right] . \tag{10}$$

where all, except the $Q_i' \equiv \frac{\partial Q}{\partial D_i}$ term, can be obtained analytically by a direct differentiation. Deriving an equation for Q_i' starts from the residual form of the *discretized* flowfield equations,

$$R(Q(D), M(X(D)), D) = \vartheta(tol) , \tag{11}$$

and by differentiating eq. (11), the equation for Q_i' is obtained,

$$R' = \left(\frac{\partial R}{\partial Q} \right)_D Q' + \left(\frac{\partial R}{\partial D} \right)_Q = 0 . \tag{12}$$

After eq. (12) is written in the delta form, the *direct method* is simply solving for the deficit-correction (incremental) form of the unknown $\Delta Q'$,

$$\left(\frac{\partial \hat{R}}{\partial Q} \right)_D^n \{\Delta Q'\}^n = - \left(\frac{\partial R}{\partial Q} \right)_D^n (Q')^n - \left(\frac{\partial R}{\partial D} \right)_Q . \tag{13}$$

Note that eq. (13) scales with the number of design variables. An alternative, which scales with the number of inequality constraints, is the *adjoint method.* The adjoint sensitivities are obtained from eqs. (9, 10, 12) to be as follows,

$$\nabla F = \left[\left(\frac{\partial F}{\partial D_i} \right)_Q - \lambda_F^T \frac{\partial R}{\partial D_i} \right], \nabla G_j = \left[\left(\frac{\partial G_j}{\partial D_i} \right)_Q - \lambda_{G_j}^T \frac{\partial R}{\partial D_i} \right] \tag{14}$$

The equations for the adjoint (Lagrange) variables are defined, from eqs. (9, 10, 12) and written in the delta form,

$$\left(\frac{\partial \hat{R}}{\partial Q} \right)^T \left\{ \Delta \lambda_F \right\}^n = - \left(\frac{\partial R}{\partial Q} \right)_D^T \lambda_F^n - \left(\frac{\partial F}{\partial Q} \right)_Q , \tag{15}$$

$$\left(\frac{\partial \hat{R}}{\partial Q} \right)^T \left\{ \Delta \lambda_{G_j} \right\}^n = - \left(\frac{\partial R}{\partial Q} \right)_D^T \lambda_{G_j}^n - \left(\frac{\partial F}{\partial Q} \right)_Q . \tag{16}$$

Therefore, to obtain the sensitivities, either eq. (13) is solved for $\Delta Q'$, or eqs. (15, 16) are solved for λ_F, λ_{G_j}.

Automatic differentiation (AD)

In developing a discrete sensitivity method, one feature that requires intense effort is the differentiation of a number of functions with respect to the design variables. This has traditionally been done using the calculus definition of a derivative represented as a *finite difference*, then evaluating the function as many times as needed. A second method is the *hand-differentiation* of the equations in their discretized form, relying on the chain rule of differentiation based on the functional dependence. This *quasi-analytical* approach produces the most efficient code, but it can be tedious and may not always be possible for non-continuous functions. A third option for the differentiation is the *symbolic differentiation* available in the mathematical computer packages. They are not, in general, capable of handling constructs, such as, subroutines, loops and branches in a computer code. Furthermore, in their basic form, they are not capable of handling the combinatorial explosion effect, which arise from the fact that a derivative expression for, say, a multiplication doubles the string.

An alternative approach is AD, which produces the derivative by applying the chain rule of differentiation on any type of function, as many times as needed. This is possible irrespective of the complexity of the function, since the computer execution of that function is always a sequence of elementary operations and elementary functions. There are various approaches, and a survey of these AD tools is given in [19]. Therefore, it is possible to develop the computer codes, such as, ODYSEE [21] or ADIFOR [8], which, given a computer code for a function evaluation, breaks the code into elementary operations and functions to simplify the problem, then generates a computer code that can produce the sensitivities of that function.

The two traditional methods for the chaining are the *forward mode* and the *reverse mode.* Consider computing certain partial derivatives of a dependent variable vector *F(1:m)* with respect to all or certain elements of an independent variable vector *D(1:n)*, that is, $\frac{\partial F(i)}{\partial D(1:n)}$. From,

$$\begin{pmatrix} \nabla F(1) \\ \cdot \\ \cdot \\ \nabla F(m) \end{pmatrix} = \begin{pmatrix} \frac{\partial F(1)}{\partial D(1)} & \cdot & \cdot & \frac{\partial F(1)}{\partial D(n)} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \frac{\partial F(m)}{\partial D(1)} & \cdot & \cdot & \frac{\partial F(m)}{\partial D(n)} \end{pmatrix} * \begin{pmatrix} \nabla D(1) \\ \cdot \\ \cdot \\ \nabla D(n) \end{pmatrix} , \tag{17}$$

where the Jacobian is $J = \frac{\partial F(1:n)}{\partial D(1:m)}$ and ∇ denotes a derivative object, by initializing $\nabla D(i)$ to the i-th canonical unit vector of length n, the output $\nabla F(i)$ contains the desired derivatives. Hence, the *forward mode* computes arbitrary linear combinations of the *columns* of the Jacobian,

$$\nabla F(i) = [J * \nabla D(1{:}n)^T]^T \quad . \tag{18}$$

A forward-mode code is relatively easier to generate and it preserves the parallelizability of the original code structure. To compute n derivatives, the forward mode requires n times as much computer time and storage as the original code, that is, the required resources scale with the number of the independent variables. In the *reverse mode*, the derivatives with respect to some intermediate variables, denoted by overbars, are computed,

$$\left(\overline{\nabla D(1)} \quad . \quad . \quad \overline{\nabla D(n)}\right) = \left(\overline{\nabla F(1)} \quad . \quad . \quad \overline{\nabla F(m)}\right) * J \quad . \tag{19}$$

By initializing $\overline{\nabla F(i)}$ to the i-th canonical unit vector of length m, the output $\overline{\nabla D(i)}$ contains $\frac{\partial D(i)}{\partial F(1:m)}$. Hence, the reverse mode computes arbitrary linear combinations of the *rows* of the Jacobian. A reverse-mode code is much more complex. To compute n derivatives, the reverse mode requires m times as much computer time as the original code, that is, the required resources scale with the number of the dependent variables, which, however, is usually less than the number of independent variables n.

The approach of the code ADIFOR [8], which is used herein, is a hybrid of the forward and reverse modes. It applies the reverse mode first to compute the derivatives of the result with respect to the design variables and then employs the forward mode to propagate the overall derivatives. By defining a seed vector as $S \equiv \nabla D(1{:}n)^T$, eq. (18) is rewritten as

$$g\_F = [\tfrac{\partial F}{\partial D} * (g\_D)^T] = [J * S]^T \quad . \tag{20}$$

Since S is initialized as a vector, rather than an identity matrix, only the product of the Jacobian and the seed vector is computed (instead of computing the full Jacobian). The output is a code which not only computes the function but also the sensitivities. This code is usually two to three times longer than the original function analysis code.

RESULTS

Each of the four improvements proposed for a shape optimization methodology, as stated in the *Introduction* section, were developed and demonstrated through a series of test cases. Included in the present paper are only a few representative results from this investigation.

Since the present design optimization methodology is automated, the parameterization technique to be used needs to be performed "on-the-fly," hence, be batch processed. Then, the parameters used in the boundary condition specifications of the PDE surfaces become the design variables; they are incrementally changed, depending on the step size dictated by the optimizer, for the shapes to evolve. However, a prerequisite step prior to the shape optimization is the *conceptual design* phase, where the designer intervenes in the decision making process. Also, the intervention is desirable in the construction of the adequate design variables and their side constraints. With this impetus, an interactive processing version [13] of the method was developed to provide a graphics visualization package that could directly implement the design changes to the model in near-real-time. Continuous input by the user allowed the model to dynamically evolve into a new configuration creating a morphing effect. Presented in Figure 1 are two sample configurations and their surface meshes generated by this method.

Often the utility of porting a parameterization technique is desired to a shape optimization methodology, which was originally developed for another parameterization technique. Also, the shape or configuration to be optimized is usually an existing design. Then, the parameterization code is required to replicate the shape, easily and with high-fidelity, so that the optimization could subsequently be performed. These inverse problems were studied with the present PDE method (designated as BWB) and the Bezier-Bernstein flexible wing model [12,22] (designated as FWM). As the methods differed in their parameterizations, rather than a point-to-point comparison, the regenerated surface contours were compared. Presented in Figure 2 are the redefinitions from these two methods (case 1) of a cranked-arrow wing section. By and large, the redefinitions were successful for the section shown as well as the rest of the wing sections.

To study the potential advantages of the automatic differentiation, a simple demonstration case was also formulated based on the same cranked-arrow wing geometry [22,23]. Subsequent to the parameterizations in two different ways, their surface meshes were automatically generated by their respective extensions, as explained in the *Surface Parameterization* section. Then, the grid sensitivities were computed using an automatic differentiated code in the former method and using a quasi-analytical hand-differentiated code in the latter method. From the results [13], as sampled in Figure 3 (case 2) for the sensitivity of the normal coordinate to the thickness of the airfoil section at the crank ($\frac{\partial z}{\partial thk_{crank}}$), it was observed that the methods had comparable levels of accuracy; but, the hand-differentiated code required less number of arithmetic operations and utilized less storage. Nonetheless, the advantage of the automatic differentiation was in the significantly reduced amount of effort for the development of the code that generated the grid sensitivities [4].

The second phase of the study involves the shape optimization cases 3-6. The present wing is considered to be at a 4.5-deg angle-of-attack to an oncoming freestream flow of Mach 2.4. To retain the validity of the laminar flow assumption for the current viscous optimization, the freestream Reynolds number is assumed to be the relatively low value 2.0e5. To assess the effect of the grid density, a coarse grid (43x15x9) and a fine grid (85x29x17) are used. The clustering of the fine grid in the direction normal to the planform is done such that the boundary layer is resolved with at least ten points. The objective is to maximize the lift-to-drag ratio subject to 14 geometric constraints and 2 aerodynamic constraints. The geometric constraints include bounds on the wing volume, section areas, and the trailing edge angles. On the other hand, the aerodynamic constraints ensure that the lift is not below and the drag is not above their tolerable values [22,23]. Cases 3-6 were selected to illustrate the effects of viscosity (Euler-based Vs. TLNS-based designs) and the grid size (coarse-grid-based Vs. fine-grid-based designs). The Euler-based design on the coarse-grid is case 3 and on the fine grid is case 5. Correspondingly, the TLNS-based design on the coarse-grid is case 4 and on the fine grid is case 6.

The optimized shape from the fine-grid TLNS-based computations was substantially different, as compared to those from the fine-grid Euler- and the coarse-grid TLNS-based computations (case 6 Vs cases 5 and 4), and produced the best lift-to-drag ratio. Their respective CPU times were 18.52 hr, 1.10 hr, and 6.78 hr. These optimized wing designs were then re-analyzed via the fine-grid TLNS-based flowfield computations. These high-fidelity, post-optimization

flowfield analyses provided a common measure of merit and helped in illustrating the trade-off between their computational efficiencies and performance gains. It was observed, that the optimized wing of the coarse-grid TLNS-based computations (case 4) resulted in an aerodynamic performance close but not equal to that of the optimized wing from the fine-grid TLNS-based computations (case 6). However, the fine-grid Euler-based optimized wing design (case 5) yielded the lowest aerodynamic performance.

CONCLUDING REMARKS

The recent advances in the numerical design methods have manifested themselves in more automation, much better accuracy, improved computational efficiency, but only moderately more practical for the real application problems. With this motivation, four recent improvements to the building blocks of such a methodology have been reported. First, in addition to a polynomial-based parameterization, a PDE based parameterization was shown to be a practical tool for a number of reasons. Second, an alternative has been incorporated to one of the tedious phases of developing such a methodology, namely, the automatic differentiation (AD) of the computer code for the flow analysis in order to generate the sensitivities. Third, by extending the methodology for the TLNS based flow simulations, the more accurate flow physics was made available. However, the computer storage requirement for a shape optimization of a practical configuration with the high-fidelity simulations, required substantial computational resources. Therefore, the final improvement reported herein responded to this point by including the ADI based system solver as an alternative to the PbCG an other direct solvers.

ACKNOWLEDGMENTS

The author is grateful to his past and present graduate students who worked with him on design optimization. Development of the methods were supported and the computer time was provided by NASA Langley Research Center. Technical monitors were D.S. Miller, J.L. Thomas and R.E. Smith.

REFERENCES

1) Baysal, O., (Editor), *Multidisciplinary Applications of Computational Fluid Dynamics*, FED-Vol. 129, Winter Annual Meeting, 1991, Atlanta, GA; ASME, New York, NY.

2) Baysal, O., (Editor), *CFD for Design and Optimization*, FED Vol. 232, International Mechanical Engineering Conference and Exposition, 1995, San Francisco, CA; ASME, New York, NY.

3) Baysal, O., "Methods for Sensitivity Analysis in Gradient-based Shape Optimization: A Review," *Proceedings of ASME Fluids Engineering Conference*, 1997, Vancouver, BC, Canada.

4) Baysal, O., and Cordero, Y., "A Perspective on Automatic Differentiation for Gradient-Based Aerodynamic Shape Optimization," Minisymposium on Automatic Differentiation for Large-Scale Optimization, *Proceedings of 3rd ECCOMAS Computational Fluid Dynamics Conference*, 1996, Paris, France.

5) Baysal, O., and Eleshaky, M.E., "Aerodynamic Sensitivity Analysis Methods For the Compressible Euler Equations," *Journal of Fluids Engineering*, Vol. 113, No. 4, 1991, pp. 681-688.

6) Baysal, O., and Eleshaky, M.E., "Aerodynamic Design Optimization Using Sensitivity Analysis and Computational Fluid Dynamics," *AIAA Journal*, Vol. 30, No. 3, 1992, pp. 718–725.

7) Baysal, O., Eleshaky, M.E., and Burgreen, G.W., "Aerodynamic Shape Optimization Using Sensitivity Analysis on Third-Order Euler Equations," *Journal of Aircraft*, Vol. 30, No. 6, 1993, pp. 953-961.

8) Bischof, C.H., Carle, A., Corliss, G.F., and Griewank, A., "ADIFOR: Generating Derivative Codes from Fortran Programs," *Scientific Programming*, Vol. 1, No. 1, 1992, pp. 1-29.

9) Bloor, M.I.G., and Wilson, M.J., "Generating Blend Surfaces Using Partial Differential Equations," *Computer Aided Design*, Vol. 21, No. 3, 1989, pp. 165-171.

10) Burgreen, G.W., Baysal, O., "Aerodynamic Shape Optimization Using Preconditioned Conjugate Gradient Methods," *AIAA Journal*, Vol. 32, No. 11, 1994, pp. 2145-2152.

11) Burgreen, G.W., Baysal, O., and Eleshaky, M.E., "Improving the Efficiency of Aerodynamic Shape Design Procedures," *AIAA Journal*, Vol. 32, No. 1, 1994, pp. 69-76.

12) Burgreen, G.W., and Baysal, O., "3D Aerodynamic Shape Optimization Using Discrete Sensitivity Analysis," *AIAA Journal*, Vol. 34, No. 9, 1996, pp. 1761-1770.

13) Cordero, Y., "Interactive PDE Approach for Geometry Definition and Its Automatic Differentiated Sensitivities," Master's Thesis, 1997, Aerospace Engineering Department, Old Dominion University, Norfolk, VA (in preparation).

14) Eleshaky, M.E. and Baysal, O., "Airfoil Shape Optimization Using Sensitivity Analysis on Viscous Flow Equations," *Journal of Fluids Engineering*, Vol. 115, No. 1, 1993, pp. 75-84.

15) Eleshaky, M.E. and Baysal, O., "Aerodynamic Shape Optimization via Sensitivity Analysis on Decomposed Computational Domains," *Journal of Computers and Fluids*, Vol. 23, No. 4, 1994, pp. 595-611.

16) Eleshaky, M.E., and Baysal, O., "Preconditioned Domain Decomposition Scheme for 3-D Aerodynamic Sensitivity Analysis," *AIAA Journal*, Vol. 32, No. 12, 1994, pp. 2489-2491.

17) Eleshaky, M.E., and Baysal, O., "Design of 3D Nacelle Near Flat-Plate Wing Using Multiblock Sensitivity Analysis (ADOS)," Paper 94-0160, AIAA 32nd Aerospace Sciences Meeting, 1994, Reno, NV. To appear in *Journal of Aircraft*.

18) Item, C.C., and Baysal, O., "Wing-Section Optimization for Supersonic Viscous Flow," *CFD for Design and Optimization* (Ed. O. Baysal), FED Vol. 232, International Mechanical Engineering Conference and Exposition, 1995, San Francisco, CA; ASME, New York, NY.

19) Juedes, D., "A Taxonomy of Automatic Differentiation Tools,", *Proceedings of the Workshop on Automatic*

Differentiation Algorithms: Theory, Implementation and Applications, (Eds. Griewank and Corliss), 1991, Society of Industrial and Applied Mathematics, Philadelphia, PA, pp. 315-330.

20) Lacasse, J.M., and Baysal, O., "Shape Optimization of Single- and Two-Element Airfoils on Blocked Grids," Paper 94-4273, *Proceedings of 5th AIAA/USAF/NASA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 1994, Panama City, FL, pp. 108-116. To appear in *Inverse Problems in Engineering*.

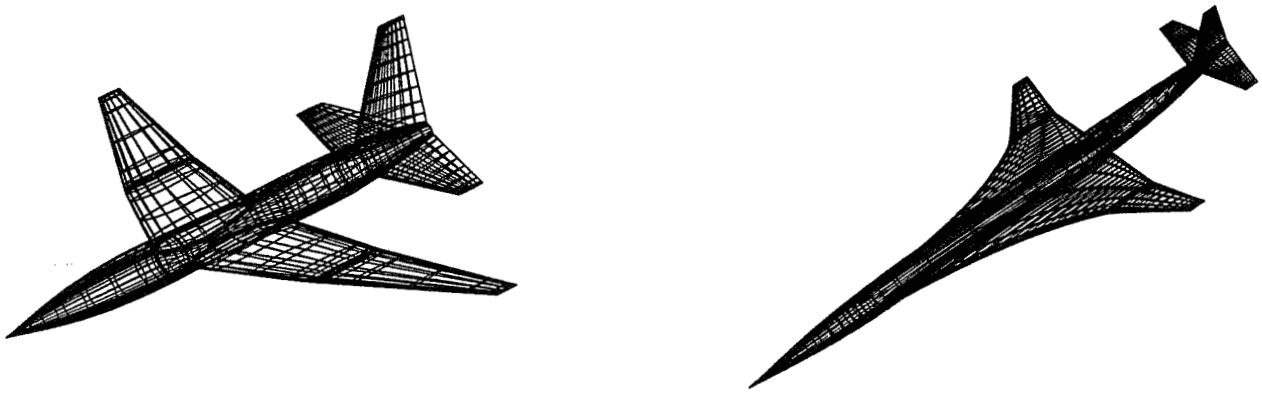
21) Rostaing, N, Dalmas, S., and Galligo, A., "Automatic Differentiation in ODYSSEE," *Tellus*, Vol. 45a, No. 5, 1994, pp. 558-568.

22) Pandya, M.J., and Baysal, O., "Gradient-Based Aerodynamic Shape Optimization Using ADI Method for Large-Scale Problems," Paper No. 96-0091, AIAA 34th Aerospace Sciences Meeting, 1996, Reno, NV. Also, *Journal of Aircraft*, Vol. 34, No. 3, 1997.

23) Pandya, M.J., and Baysal, O., "3D Viscous ADI Algorithm and Strategies for Shape Optimization," Paper No. 97-1853 CP, *Proceedings of AIAA 13th CFD Conference*, 1997, Snowmass, CO.

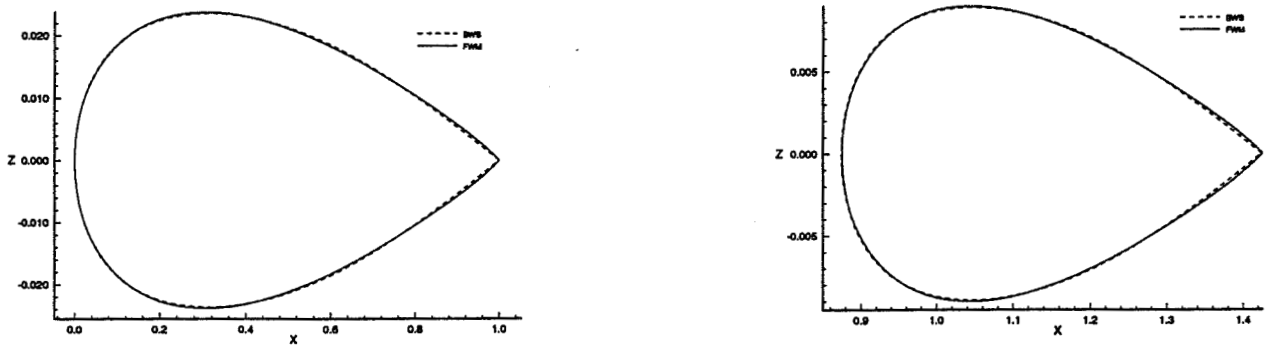
24) Smith, R.E., Mastin, C.W., and Cordero, Y., "Conceptual Airplane Design with Automatic Surface Generation," SAE Paper 96-5517, First World Aviation Congress and Exposition, 1996, Los Angeles, CA.

25) Vanderplaats, G.N., "ADS - A FORTRAN Program for Automated Design Synthesis - Version 1.10," NASA Contractor Report 177985, 1985, Washington, D.C.



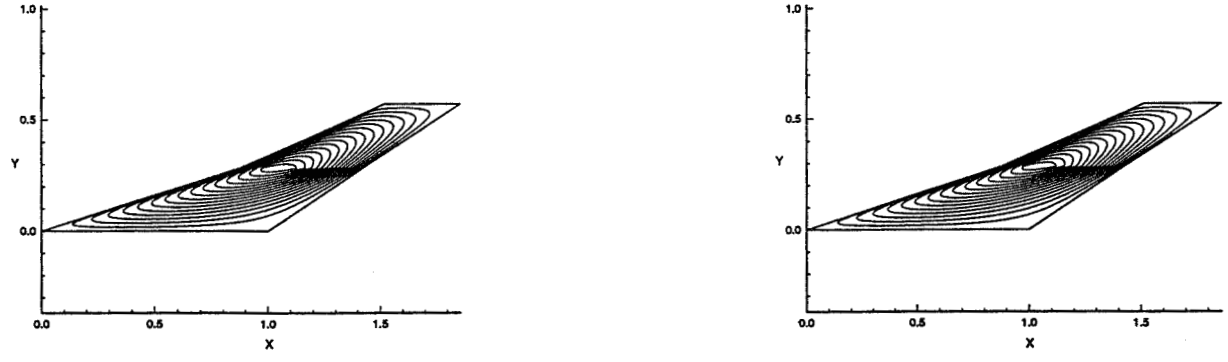
subsonic transport supersonic transport

Figure 1: Sample configurations and their surface meshes generated by the PDE method.



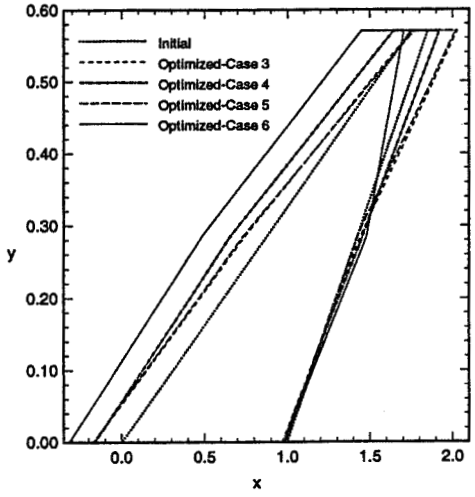
root airfoil crank airfoil

Figure 2: Comparison of wing sections (case I) parametrized using the Bezier–Bernstein polynomials (FWM) and the PDE method (BWB).

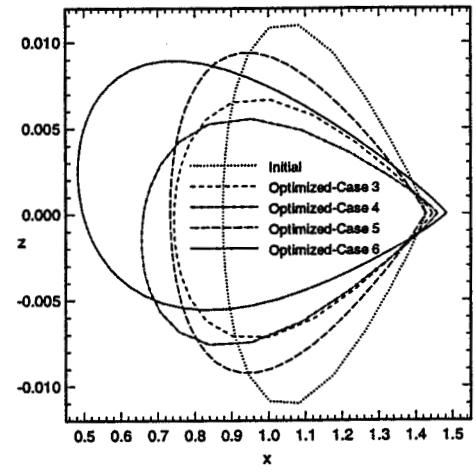


automatic differentiation analytical hand differentiation

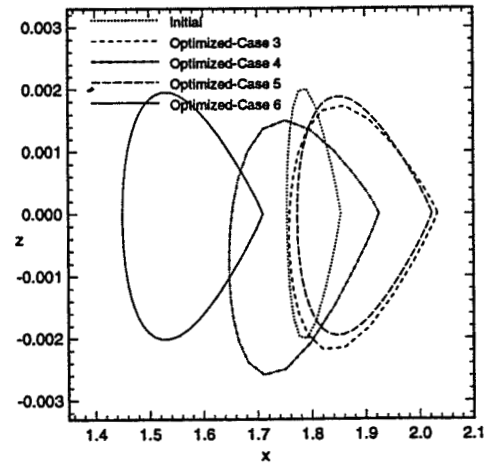
Figure 3: Sensitivity contours of the normal coordinate to the thickness of the crank airfoil (case 2).



Planform

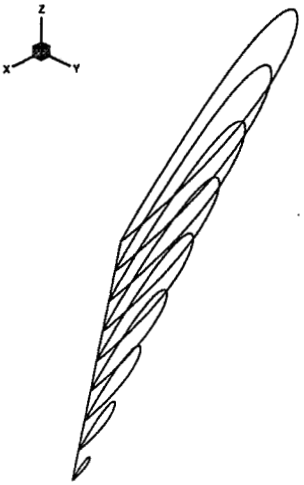


Mid Section

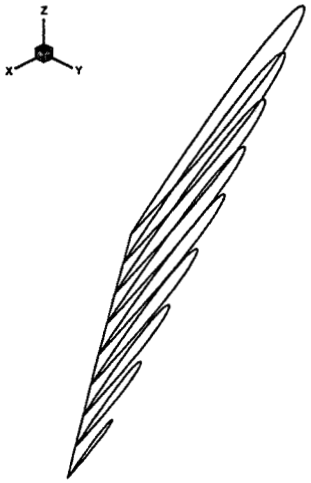


Tip Section

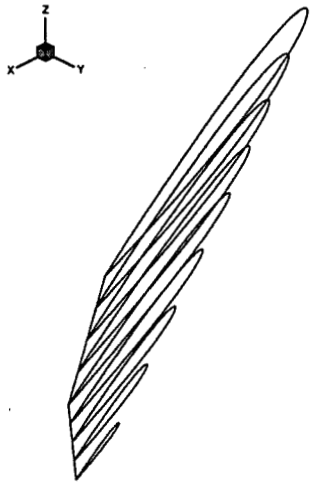
Figure 4: Comparison of optimized shapes from cases 3–6



Initial arrow wing



Optimized arrow wing (Case 4)



Optimized arrow wing (Case 6)

Figure 5: 3D view of initial and optimized wing

The American Institute of Aeronautics and Astronautics

AIAA 97-1853 CP

Three-Dimensional Viscous Alternating Direction Implicit Algorithm and Strategies for Shape Optimization

Mohagna J. Pandya and Oktay Baysal

Aerospace Engineering Department
Old Dominion University
Norfolk, Virginia 23529-0247

Preprint from the Proceedings

13th AIAA Computational Fluid Dynamics Conference
June 29 - July 2, 1997 /Snowmass, Colorado

AIAA 97-1853-CP

Three-Dimensional Viscous ADI Algorithm and Strategies for Shape Optimization

Mohagna J. Pandya[1] and Oktay Baysal[2]

Aerospace Engineering Department
Old Dominion University
Norfolk, VA 23529-0247

ABSTRACT

A gradient-based shape optimization methodology based on quasi-analytical sensitivities has been extended for practical three-dimensional aerodynamic applications. The flow analysis has been rendered by a fully implicit, finite-volume formulation of the Euler and Thin-Layer Navier-Stokes (TLNS) equations. Initially, the viscous laminar flow analysis for a wing has been compared with an independent computational fluid dynamics (CFD) code which has been extensively validated. The new procedure has been demonstrated in the design of a cranked arrow wing at Mach 2.4, with coarse- and fine-grid-based computations performed with Euler and TLNS equations. The influence of the initial constraints on the geometry and aerodynamics of the optimized shape has been explored. Various final shapes generated for an identical initial problem formulation but with different optimization path options (coarse or fine grid, Euler or TLNS), have been aerodynamically evaluated via a common fine-grid TLNS-based analysis.

The initial constraint conditions show significant bearing on the optimization results. Also, the results demonstrate that to produce an aerodynamically efficient design, it is imperative to include the viscous physics in the optimization procedure with the proper resolution. Based upon the present results, to better utilize the scarce computational resources, it is recommended that, a number of viscous coarse grid cases using either a preconditioned bi-conjugate gradient (PbCG) or an alternating-direction-implicit (ADI) method, should initially be employed to improve the optimization problem definition, the design space and the initial shape. Optimized shapes should subsequently be analyzed using a high fidelity (viscous with fine-grid resolution) flow analysis to evaluate their true performance potential. Finally, a viscous fine-grid-based shape optimization should be conducted, using an ADI method, to accurately obtain the final optimized shape.

INTRODUCTION

Aerodynamic design by the numerical optimization method has witnessed a spectacular rise in the research interest and limited success in handling realistic problems. In this pursuit, several inviscid-flow-based optimization results have been presented in the past. The motivation of the present study is to report an extension of such an approach for the viscous flow equations and present certain algorithmic strategies for formulating and solving a practical problem. For this purpose, a gradient-based shape optimization methodology has been developed and used.

Since the early 90's, there have been promising developments in the area of aerodynamic design optimization based on discrete sensitivity analysis [1]. The issue of computational-time efficiency was addressed in several ways [2], including the surface parameterization using Bezier splines and Bernstein polynomials, and solving the unfactored CFD equations by a quasi-Newton method. The computer storage efficiency was later addressed, first by examining the solution of all the equations using a preconditioned biconjugate gradient (PbCG) method [3], then by the multiblock sensitivity analysis scheme [4]. Another benefit of the latter was its applicability to a complex and multicomponent geometry, around which structured grids could only be generated by the use of domain decomposition techniques. The multiblock sensitivity analysis also incorporated [5] and applied in the optimization of a multiple-component configuration [6].

In the PbCG methods [3,5], incomplete lower-upper (ILU) decompositions of the coefficient matrix with some fill-in (ILU (n), $n \geq 0$) were used for the preconditioning. Another preconditioning option is to use the approximate factors of the coefficient matrix, then break the problem into a sequence of simpler problems; that is the premise of the ADI schemes, which have been used to solve the flow equations for more than two decades [7]. The trade-off in the latter approach is a slower convergence rate for the much reduced computer memory storage [8].

The results from all the applications cited above indicate that, in general, the design methods obtain a final shape via an evolution of successively improved shapes, although sometimes at a rather high computational cost due to the large number of flow analyses involved. A practical use of the fully implicit CFD methods (quasi-Newton method) within an optimization procedure allows the realization of high convergence rates and consequent reduction of the CPU time. The memory requirement of such a procedure which uses a PbCG algorithm is low as compared to direct inversion solvers. However, this memory requirement is high enough to preclude the realistic, high grid density design of a practical 3D geometry such as a wing or wing-body combination. This limitation served as the impetus to improve upon the formerly developed 3D optimization methodology, which was demonstrated for a transport wing [9], an asymmetric delta wing, and a cranked delta wing [10]. This improvement has been achieved by using ADI-factored operators to serve as the preconditioning matrices for the CFD as well as the sensitivity equations [11]. The results indicated that the shape optimization problems which required large numbers of grid points could be resolved with a gradient-based approach. The method, however, was developed and demonstrated only for the Euler equations.

Since there are numerous real design cases where the viscous effects could not be neglected, the method needed to be extended for the viscous flow equations. The vis-

[1]Graduate Research Assistant, Student Member, AIAA.

[2]Professor and Eminent Scholar, Associate Fellow, AIAA.

cous design has previously been performed, for example, for single and multielement airfoils [12,13] and to demonstrate the leading-edge thrust phenomenon [14,15]. Therefore, this was the first objective of the present study. The second objective was inspired from the work of Refs. 9 and 11. In Ref. 11, it was recommended that, to better utilize the computational resources, a number of coarse grid cases, preferably using the PbCG method, should initially be conducted to better define the optimization problem and the design space, and obtain an improved initial shape; subsequently, a fine-grid shape optimization should be conducted, using the ADI method, to accurately obtain the final optimized shape. Also, a comment was made in Ref. 9, that the minor changes in a problem formulation could result in radically different designs. Indeed, this is also true for the computational resources needed for each design. Hence, an investigation of some pertinent strategies was deemed necessary and conducted subsequently to be presented herein.

SYNOPSIS OF METHODOLOGY

Aerodynamic design optimization extremizes an objective function $F(D, Q(D))$ subject to constraints $G(D, Q(D))$, where D is a vector of related design variables [1]. The employed optimization algorithm is the method of feasible directions as coded in ADS [16]. To evaluate the necessary objective function and aerodynamic constraints, the present study employs CFD analysis [11] to solve either the Euler or the thin-layer Navier-Stokes equations. The flux vectors and the Jacobian matrices are evaluated using the flux-vector-splitting technique of van Leer. The cell interface values of the flow variables are determined using a spatially third-order accurate, upwind-biased, MUSCL interpolation formula with van Albada flux limiter. The viscous fluxes are centrally differenced. The sensitivity derivatives of the objective function and aerodynamic constraints are obtained analytically [1] from the steady-state residual form of the flow equations. The direct differentiation method [1] is used for obtaining the resulting sensitivity equation. The flowfield and the sensitivity equations are then solved by employing the approximate factorization followed by the ADI scheme of Beam and Warming [7]. This specific choice was made to reduce the computer memory requirement of the optimization algorithm and thereby facilitate a relatively higher grid density design [11]. The wing parameterization and the regridding model are adapted from Burgreen and Baysal [9]. The wing parameterization is schematically represented in Figure 1.

RESULTS

To evaluate the viability of the present extension of the gradient-based optimization methodology, and to explore the ways to deploy it wittingly, several problems related to wing optimization have been formulated and solved. Also, for a better posed optimization problem, some studies are aimed at investgating the algorithmic strategies with regard to the initial constraint specification. Results of these studies are presented next.

Effect of Viscosity on Flow and Sensitivity Analyses

The initial arrow wing used for this purpose has thickness-to-chord ratio of 4%, leading- and trailing-edge angles (Λ_{LE}, Λ_{TE}) of 72-deg and 56.3-deg, respectively, and an aspect ratio of 1.01. The present wing is considered to be at a 4.5-deg angle-of-attack (aoa) to an oncoming freestream flow of Mach 2.4. To retain the validity of the laminar flow assumption for the current viscous optimization, the freestream Reynolds number is assumed to be the relatively low value of 2.0×10^5. As the computational grids, a coarse grid $(43\times15\times9)$[3] and a fine grid $(85\times29\times17)$ are used. The clustering of the fine grid in the direction normal to the planform is done such that the boundary layer is resolved with at least ten points.

The flowfield on the initial wing is computed by the fine-grid analyses performed using the present viscous code and CFL3D-version 3.0 [17]. Results of these analyses show excellent agreement as manifested in the comparison of the chordwise pressure coefficient distributions at three spanwise sections the of initial wing (Figure 2). For this case, the present code predicts the lift coefficient to be 0.0189 and lift-to-drag ratio to be 6.068, whereas, the corresponding values predicted by CFL3D are 0.0187 and 6.055, respectively.

Subsequently, the present code is used to compute the flowfield on the initial wing corresponding to the Euler- and TLNS-based flow analyses on the coarse and fine grids (Figure 3 and Table 1). From the surface pressure coefficient distributions, contrasted for three sections of this wing (Figure 3), it is evident that the fine-grid computations render a crisper definition of the suction peaks and slightly increased levels of the lower surface pressures. In addition, the upper surface pressure variations are significantly different for the outboard portion of the wing. Also the fine-grid computations reveal a marked difference in the pressure distributions between the Euler and TLNS cases. Hence, it is confirmed that the fine-grid computations have the adequate resolution of the boundary layer. The predicted aerodynamic coefficients from these cases along with their computational resource requirements are summarized in Table 1. As expected, the fine-grid viscous analysis requires the largest resources but yields significantly more accurate aerodynamic coefficients.

A comparative database, in regards to the the grid refinement and viscous effects, is also developed for the sensitivities starting with the above mentioned flow analyses (Table 2). The direct differentiation method [1] is used with a convergence criterion of 2.5 orders of magnitude ($\vartheta(2.5)$) reduction in the residual. This choice has been deemed adequate, after studying the effects of stricter convergence criteria on the gradients, and considering the importance of the accurate and efficient computational procedure on the overall performance of the optimization methodology. As it can be observed, viscous fine-grid sensitivities require a larger number of iterations to converge, thence, more CPU time. Also, they are quite different from the corresponding sensitivities of the viscous coarse-grid, Euler coarse-grid, and Euler fine-grid cases.

Shape Optimization

The present problem formulation has thirteen design variables (spn; and at the wing root, mid and tip sections: `thkscal`, `chdscal`, `tranx` and `twst`). Each section of this wing is initially a NACA-0004 airfoil defined in the $x-z$ plane. The objective function is selected to be the maximization of C_L/C_D, subject to 14 geometric and two aerodynamic inequality constraints:

$$C_L \geq C_{L_{\min}}\,, \qquad C_D \leq C_{D_{\max}}\,, \tag{1}$$

$$V_{\text{wing}} \geq 0.8V_{\text{wing}}^{\text{initial}}\,, \quad A_{\text{midspan}} \geq 0.6A_{\text{midspan}}^{\text{initial}}\,, \tag{2}$$

and at the root, mid, and tip sections:

$$2^\circ \leq \theta_{0.90\text{chord}} \leq 20^\circ\,, \quad 2^\circ \leq \theta_{0.98\text{chord}} \leq 20^\circ\,, \tag{3}$$

where, $C_{L_{\min}}$ and $C_{D_{\max}}$ are the threshold values of the lift and drag coefficients, respectively. V denotes the wing

[3] Dimensions in streamwise, normal and spanwise directions, respectively.

volume, A denotes wing section area and θ is the trailing-edge included angle. The current problem formulation also imposes side (equality) constraints on the design variables. The utility of imposing such volume and angle constraints has been discussed in Ref. 11.

A reliable and comprehensive wing design methodology should be able to address issues, such as, the effects of grid refinement and viscosity on the optimized shape. Thus, the primary objective of this study is to quantify these effects and somewhat elevate the present wing design methodology from the realm of conjectures on these issues. Note that for all the analysis cases studied above for the identical initial shape (Figure 3 and Table 1), the flowfields themselves are different in their character, consequently, they predict different values of lift and drag coefficients. These deviations would generate, even for the exact same values of $C_{L_{\min}}$ and $C_{D_{\max}}$, different initial aerodynamic constraints and would lead the optimizer to compute different search directions and optimized shapes. To neutralize this initial influence on the optimizer and facilitate judicious evaluation of the effect of grid refinement and viscous computations enroute to the optimized shape, $C_{L_{\min}}$ and $C_{D_{\max}}$ values are altered to obtain identical initial constraints. Therefore, a test matrix of six cases is reached, as given with their rationale in Table 3. In these optimization studies, full CFD analyses are used for the flowfield computations, prior to the search direction evaluations as well as for the one-dimensional searches. During the one-dimensional searches, the wing geometry is perturbed by varying the design variables in 1% increments ($\Delta\alpha = 0.01$).

The primary objectives for cases 1-4 (Table 3) are to study the effects of initial constraints on the optimized shapes and to compare the optimized shapes generated by the Euler- and TLNS-based computations for the identical values of initial aerodynamic and geometric constraints. Due to the consideration of the available computational resources, only the coarse-grid computations are performed. To make the comparisons rather broad-based, two sets of initial aerodynamic constraint conditions are selected. For the first condition, lift constraint is violated and drag constraint is inactive (lift-driven design, cases 1 and 2), whereas for the second condition, lift as well as drag constraints are violated, with the drag violation being more severe than the lift violation (drag- and lift-driven design, cases 3 and 4).

Presented in Table 4 is a summary of aerodynamic features, geometric changes, and statistics for cases 1-4. The optimized wing shapes for these cases are contrasted against the initial shape in Figure 4 and the corresponding evolutions of the objective function and the aerodynamic coefficients are given in Figure 5. For both the Euler and the TLNS cases, the results are distinctly different for the two sets of initial constraint conditions (In Table 4 and Figures 4 and 5: case 1 vs 3, and case 2 vs 4). As explained before, these two constraint conditions represent lift-driven, and drag- and lift-driven designs, respectively. This difference leads the optimizer to derive the pertinent aerodynamic performance improvements along different search directions in the design space and, hence, produces very distinct designs as illustrated further below. For the cases with initially violated lift constraint and inactive drag constraint (cases 1 and 2), search directions are evaluated with the emphasis on the lift improvement, even incurring a drag penalty. This is reflected in the optimized shapes having reduced leading-edge sweep of inboard wing and positive geometric twists of the sections. The first search direction also results into substantial drag increase with the concomitant lift improvement. This increase in drag violates the drag constraint and, in the later stages of the optimization, is sought to be satisfied via a reduction of the wave drag due to the volume. The effect of viscosity appears to be of secondary importance for these cases (see, e.g. cases 1 and 2 in Figure 4).

For the cases that initially violated lift and drag constraints, with the drag violation being more severe (cases 3 and 4), the optimizer generates the search directions by focusing on drag reduction, leading to the optimized shapes with increased leading-edge sweep, reduced thickness-to-chord ratios, and substantially trimmed wings.

Another remarkable feature of the optimized design is that outboard portions of the optimized wings have drooped leading edges. TLNS-based optimized wing has an outboard portion with more pronounced leading edge-drooping as compared to that of the Euler-based optimized wing (Figure 4). This trend can be linked to the realization of the leading-edge thrust potential, which has been reported in the literature [14,15]. The main idea is to reduce the vulnerability to separation, by aligning the wing's subsonic leading edges with the freestream, which inturn, reduces the suction peaks and the entailing adverse pressure gradients due to the pressure recovery. Although the evolution trends (Figure 5) of respective objective functions and aerodynamic constraints are similar, large differences exist in their values. This is a natural outcome of excluding the friction drag from the Euler computations, and its minor variations, as compared to the inviscid drag, during the TLNS-based optimization. Again, it is noted that the Euler- and the TLNS-based optimized shapes and their aerodynamics are quite different for the identical values of $C_{L_{\min}}$ and $C_{D_{\max}}$ (cases 1 vs 4, Table 4).

The second phase of the study involves the fine-grid-based optimization cases using the Euler and TLNS analyses (cases 5 and 6, respectively, in Table 3). For these cases, only one set of initial aerodynamic constraints, violated lift and drag constraints, is chosen (comparable to cases 3 and 4). A summary of the results is presented in Table 4. The optimized shapes are compared with each other and the initial shape in Figures 6 and 7. The evolution of the objective function and the aerodynamic coefficients are recorded and compared in Figure 8.

The optimized shape from the fine-grid TLNS-based computations is substantially different, as compared to those from the fine-grid Euler- and coarse-grid TLNS-based computations. The leading-edge drooping, as observed in the outboard portion of the optimized wing from the coarse-grid TLNS-based computations, is absent from that of the fine-grid TLNS-based computations (case 4 vs case 6, Figures 6 and 7). Although, the optimized shapes are not quite similar in these cases, their aerodynamic performance improvements towards the respective optimized shapes, display similar trends (Figure 8). However, as seen in Table 4, the fine-grid TLNS-based optimization requires much more CPU time as compared to cases 4 and 5 (18.52 hr vs. 1.10 hr and 6.78 hr).

Finally, various optimized wing designs generated with different computational strategies (cases 3–6), are reanalyzed via the fine-grid TLNS-based flowfield computations, and denoted as cases 3'-6'. These high-fidelity, post-optimization flowfield analyses provide a common measure of merit for various optimization cases and illustrate the trade-off between computational efficiency and performance gain. A summary of these results is presented in Table 5. Note that C_L/C_D values now compare differently for these cases (3–6 in Table 4 vs 3'-6' in Table 5). It is observed from Table 5, that the optimized wing of coarse-grid TLNS-based computations (case 4') yields aerodynamic performance comparable to that of the optimized wing of the fine-grid TLNS-based computations (case 6'), while retaining its higher computational efficiency. However, the fine-grid Euler-based optimized wing design (case 5') yields the lowest aerodynamic performance.

SUMMARY AND CONCLUSIONS

A gradient-based optimization procedure has been demonstrated for the viscous flow conditions, by optimizing an

arrow wing in supersonic freestream. Initially, the viscous flow analyses have been successfully compared with those from a highly validated CFD code. The viscous effects, when properly resolved, dictate differences not only in the flow analyses but also in the sensitvity analyses and the shape optimization. The present studies also demonstrate the strong influence of the changes in the initial constraints, on the optimization path and the overall aerodynamic performance improvement. The drag- and lift-driven optimization renders better performance, as compared to merely a lift-driven optimization. However, the relative merits of each optimization strategy would not be understood, hence properly compared, unless a uniform and a reliable method of evaluating their final designs is devised and implemented. For the added value to the aerodynamic performance, the viscous fine-grid design strategy is the most effective, the Euler fine-grid design is the least effective, and the viscous coarse-grid design ranks somewhere in between. As for the CPU-time requirement of these three cases, the viscous fine-grid design requires the highest, the viscous coarse-grid design requires the lowest, whereas, the Euler fine-grid design ranks between these extremes.

From these results, it may be concluded, that to produce an aerodynamically efficient design, it is imperative to include the viscous physics in the optimization procedure. This would be advantageous, even if the viscous effects are only approximately resolved, as it may be necessary due to the conceivable CPU-time constraints. Actually, to better utilize the computational resources, it is recommended that a number of viscous coarse grid cases are initially explored, so to improve the optimization problem definition and the design space for the initial shape. Subsequently, the optimized shapes should be analyzed using high-fidelity (viscous, fine-grid resolution) flow analyses, to evaluate their true performance potential. Finally, a fine-grid shape optimization should be conducted, using the ADI method, to accurately obtain the final optimized shape.

REFERENCES

1. Baysal, O., and Eleshaky, M.E., "Aerodynamic Design Optimization Using Sensitivity Analysis and Computational Fluid Dynamics," *AIAA Journal*, Vol. 30, No. 3, March 1992, pp. 718-725.

2. Burgreen, G.W., Baysal, O., and Eleshaky, M.E., "Improving the Efficiency of Aerodynamic Shape Design Procedures," *AIAA Journal*, Vol. 32, No. 1, January 1994, pp. 69-76.

3. Burgreen, G.W., Baysal, O., "Aerodynamic Shape Optimization Using Preconditioned Conjugate Gradient Methods," *AIAA Journal*, Vol. 32, No. 11, November 1994, pp. 2145-2152.

4. Eleshaky, M.E. and Baysal, O., "Aerodynamic Shape Optimization via Sensitivity Analysis on Decomposed Computational Domains," *Journal of Computers and Fluids*, Vol. 23, No. 4, May 1994, pp. 595-611.

5. Eleshaky, M.E., and Baysal, O., "Preconditioned Domain Decomposition Scheme for 3-D Aerodynamic Sensitivity Analysis," *AIAA Journal*, Vol. 32, No. 12, December. 1994, pp. 2489-2491.

6. Eleshaky, M.E., and Baysal, O., "Design of 3-D Nacelle Near Flat-Plate Wing Using Multiblock Sensitivity Analysis (ADOS)," AIAA Paper 94-0160, AIAA 32nd Aerospace Sciences Meeting, Reno, NV, January 1994. Also, to appear in *Journal of Aircraft.*

7. Beam, R.M., and Warming, R.F., "An Implicit Factored Scheme for the Compressible Navier-Stokes Equations," *AIAA Journal*, Vol. 16, April 1978, pp. 393-402.

8. Korivi, V.M., Taylor, A.C., III, Hou, G.W., Newman, P.A., and Jones, H.E., "Sensitivity Derivatives for Three-Dimensional Supersonic Euler Code Using Incremental Iterative Strategy," *AIAA Journal*, Vol. 32, No. 6, June 1994, pp. 1319-1321.

9. Burgreen, G.W., and Baysal, O., "3-D Aerodynamic Shape Optimization Using Discrete Sensitivity Analysis," *AIAA Journal*, Vol. 34, No. 9, September 1996, pp. 1761-1770.

10. Burgreen, G.W., and Baysal, O., "Three-Dimensional Aerodynamic Shape Optimization of Delta Wings," Paper 94-4271 CP, Proceedings of 5th AIAA/USAF/NASA/ISSMO Multidisciplinary Analysis and Optimization Conference, Panama City, FL, September 1994, pp. 87-97.

11. Pandya, M.J., and Baysal, O., "Gradient-Based Aerodynamic Shape Optimization Using ADI Method for Large-Scale Problems," AIAA Paper No. 96-0091, 34th Aerospace Sciences Meeting, January 15-18, 1996, Reno, NV. To appear in *Journal of Aircraft*, Vol. 34, No. 3, May-June 1997.

12. Eleshaky, M.E., and Baysal, O., "Airfoil Shape Optimization Using Sensitivity Analysis on Viscous Flow Equations," [ASME] *Journal of Fluids Engineering*, Vol. 115, No. 1, March 1993, pp. 75-84.

13. Lacasse, J.M., and Baysal, O., "Shape Optimization of Single- and Two-Element Airfoils on Blocked Grids," AIAA Paper 94-4273, Proceedings of 5th AIAA/USAF/NASA/ISSMO Multidisciplinary Analysis and Optimization Conference, Panama City, FL, September 1994, pp. 108-116. To appear in *Journal of Inverse Problems in Engineering.*

14. Item, C.C., and Baysal, O., "Wing Section Optimization for Supersonic Viscous Flow," CFD for Design and Optimization (Ed. O. Baysal), ASME FED-Vol. 232, pp. 29-36, International Mechanical Engineering Conference and Exposition, November 12-17, 1995, San Francisco, CA. To appear in *Journal of Fluids Engineering.*

15. Carlson H.W., and Mann, M.J., "Survey and Analysis of Research on Supersonic Drag Due-to-Lift Minimization with Recommendations for Wing Design," NASA Technical Paper, No. 3202, September 1992.

16. Vanderplaats, G.N., ADS - A FORTRAN Program for Automated Design Synthesis - Version 1.10, NASA Contractor Report 177985, September 1985.

17. Thomas, J.L., Krist, S.T., and Anderson, W.K., "Navier-Stokes Computations of Low Aspect Ratio Wings," *AIAA Journal*, Vol. 28, No. 2, February 1990, pp. 205-215.

ACKNOWLEDGMENT

This research was partially supported by NASA Langley Research Center under Grant NCC-1-211. The technical monitor was Dr. James L. Thomas.

Table 1: Efficiency and accuracy comparisons of various cases for flowfield analysis*

	Euler		TLNS	
	Coarse Grid (43×15×9)	Fine Grid (85×29×17)	Coarse Grid (43×15×9)	Fine Grid (85×29×17)
Iterations	237	401	236	686
CPU time (sec)	278.7	3202.1	281.5	5646.7
Memory (MW)	2.25	18.2	2.25	18.2
C_L	0.12762	0.13298	0.12515	0.11488
C_D	0.01248	0.01200	0.01629	0.01893
(C_L/C_D)	10.226	11.079	7.681	6.068

*Computed flowfield for $M_\infty = 2.4$ on initial arrow wing.

Table 2: Efficiency and accuracy comparison of various cases for sensitivity analysis*

	Euler		TLNS	
	Coarse Grid (43×15×9)	Fine Grid (85×29×17)	Coarse Grid (43×15×9)	Fine Grid (85×29×17)
Iterations	50	115	49	182
CPU time (sec)	174.3	2893.9	173.6	4580.0
Memory (MW)	3.58	20.99	3.60	21.01
$\partial C_D/\partial(\texttt{chdscal}_{\texttt{mid}})$**	-0.1061	-0.0708	-0.1193	-0.3310
$\partial C_D/\partial(\texttt{thkscal}_{\texttt{root}})$	0.1059	0.1106	0.1125	0.0550
$\partial C_D/\partial(\texttt{twst}_{\texttt{mid}})$	0.1662	0.1729	0.1627	0.1163
$\partial(C_L/C_D)/\partial(\texttt{chdscal}_{\texttt{mid}})$	2.8407	2.1976	1.9114	3.3196
$\partial(C_L/C_D)/\partial(\texttt{thkscal}_{\texttt{root}})$	-1.5735	-1.7550	-1.0308	-0.6720
$\partial(C_L/C_D)/\partial(\texttt{twst}_{\texttt{mid}})$	-0.8053	-0.9446	-0.2808	0.0044

*Computed flowfield for $M_\infty = 2.4$ on initial arrow wing; direct sensitivity method with $\vartheta(2.5)$ convergence.
**Parameters as defined in Figure 1.

Table 3: Summary and rationale of optimization cases

Case no.	Equations	Grid	$C_{L\min}$	$C_{D\max}$	Initial Constraints¶	
					Lift	Drag
1	Euler	Coarse#	0.15000	0.01285	0.14923 V†	-0.02879 I
2	TLNS	Coarse	0.14710	0.01678	0.14925 V	-0.02901 I
3	Euler	Coarse	0.15297	0.00984	0.16575 V	0.26830 V
4	TLNS	Coarse	0.15000	0.01285	0.16570 V	0.26800 V
5	Euler	Fine*	0.15940	0.00946	0.16575 V	0.26832 V
6	TLNS	Fine**	0.13698	0.01479	0.16569 V	0.26794 V

#Coarse: 43×15×9 points; *Fine: 85×29×17 points; **Fine: 85×29×9 points.
¶Lift $= 1 - C_L/C_{L\min}$, Drag $= C_D/C_{D\max} - 1$; †V: violated, I: inactive.

Rationale	Effect of Constraints	Effect of Viscosity	Effect of Grid
Cases	Euler 1 Vs 3 TLNS 2 Vs 4	Coarse 1 Vs 2 || 3 Vs 4 Fine 5 Vs 6	Euler 3 Vs 5 TLNS 4 Vs 6

Table 4: Optimization results for cases 1–6

	Case 1*	Case 2	Case 3	Case 4	Case 5#	Case 6
C_L	0.13790	0.13541	0.13156	0.12839	0.13425	0.12453
C_D	0.01290	0.01680	0.01075	0.01438	0.01073	0.01673
C_L/C_D	10.69	8.06	12.24	8.93	12.52	7.44
Lift constraint	0.0807	0.0794	0.1399	0.1441	0.1578	0.0909
Drag constraint	0.0037	0.0012	0.0923	0.1188	0.1333	0.1317
Aspect ratio	0.862	0.877	0.775	0.725	0.786	0.607
$\%A_{mid}^{initial}$	101	99	80	86	107	119.6
$\%V_{wing}^{initial}$	95	92	81	80	106	107
Λ_{LE} (deg)	69.4-72.4	68.9-71.9	72.6-74.3	70.8-73.9	72.5-74.6	70.4-73.5
	3.7	3.8	2.29	2.1	3.0	2.2
$(t/c)_{root,mid,tip}$ (%)	3.1	3.0	1.97	1.62	2.7	1.4
	1.5	1.4	1.41	1.44	1.5	1.54
	0.10	0.11	0.13	0.23	0.05	0.25
$(\texttt{twst})_{root,mid,tip}$ (deg)	0.08	0.09	−0.03	−0.11	0.01	0.14
	0.01	0.01	−0.07	−0.16	−0.01	−0.01
Lift change (%)	8.05	8.19	3.09	2.59	0.96	8.97
Drag change (%)	3.35	3.13	-13.86	-11.76	-10.58	-11.62
1-D searches	36	37	57	73	31	82
Gradient evaluations	4	4	11	10	3	8
Memory (MW)	3.58	3.60	3.58	3.60	20.99	12.37
CPU time (hr)	0.63	0.58	1.15	1.10	6.78	18.52

*Cases 1–4 with coarse grid: 43×15×9.
#Case 5 with fine grid: 85×29×17, Case 6 with fine grid: 85×29×9.

Table 5: Aerodynamic performance and efficiency comparisons of optimized shapes from various cases*:

	Case 3′	Case 4′	Case 5′	Case 6′
C_L	0.11915	0.12323	0.11558	0.12453
$C_{D\,inviscid}$	0.00966	0.00991	0.00974	0.01037
$C_{D\,viscous}$	0.00701	0.00680	0.00708	0.00636
$C_{D\,total}$	0.01667	0.01671	0.01682	0.01673
C_L/C_D	7.15	7.38	6.87	7.44
CPU# time (hr)	1.97	1.92	7.63	18.52

*TLNS flowfield analysis for $M_\infty = 2.4$, 85×29×9 C-H grid.
#Includes respective optimization and current flowfield analysis

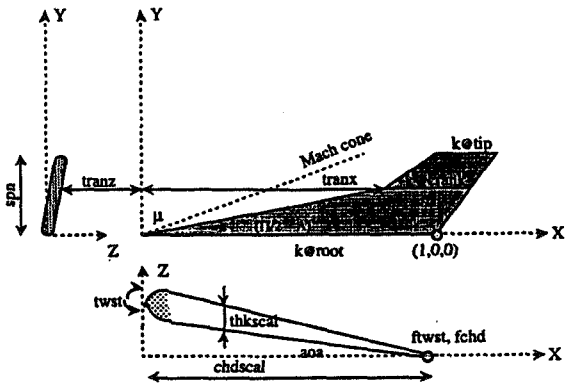
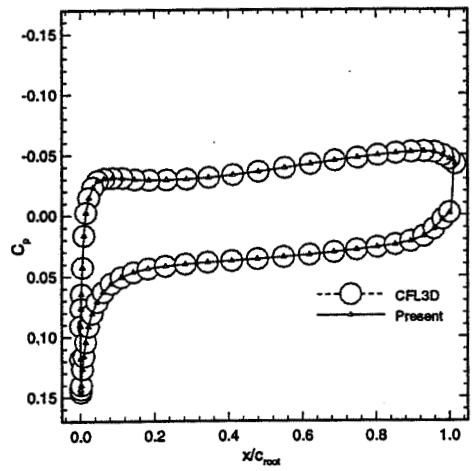
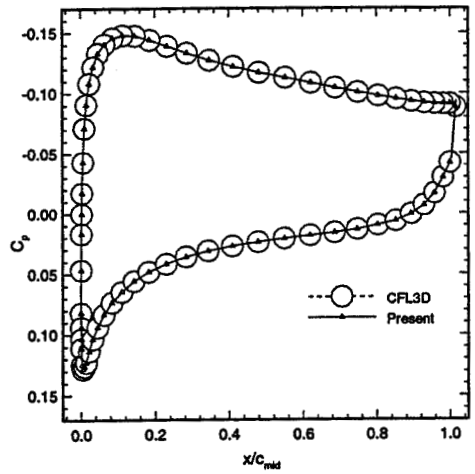


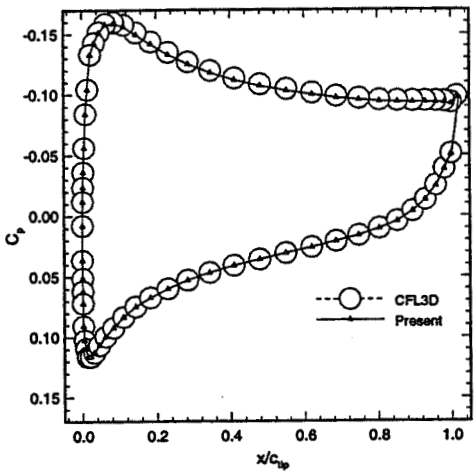
Figure 1: Wing Parameterization.



Root Section

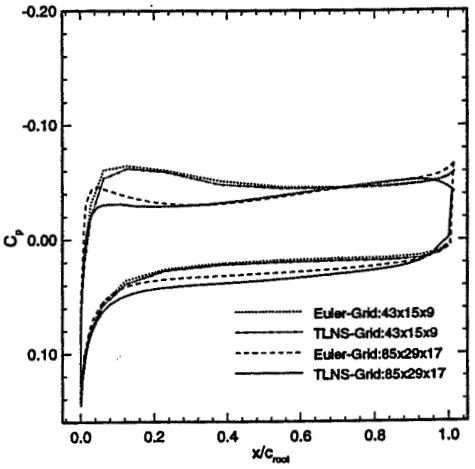


Mid Section

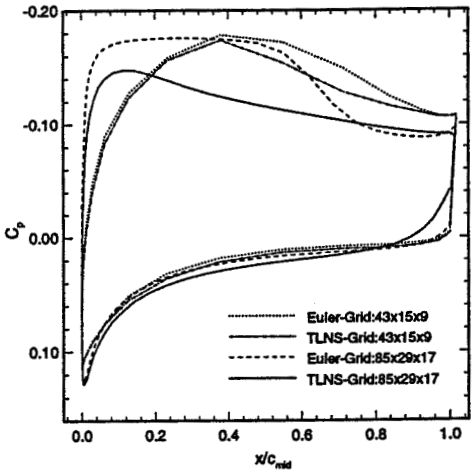


Tip Section

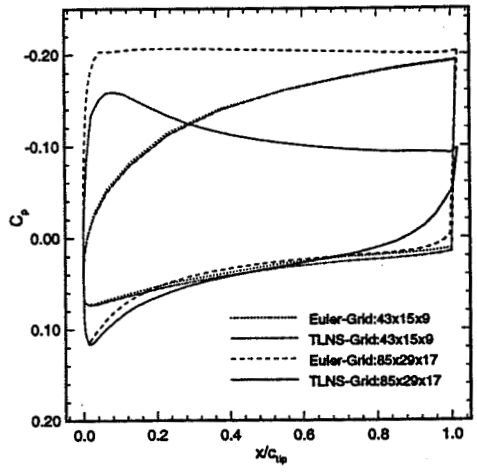
Figure 2: Comparison of chordwise pressure coefficients on fine grid (85x29x17). M=2.4, aoa=4.5-deg.



Root Section

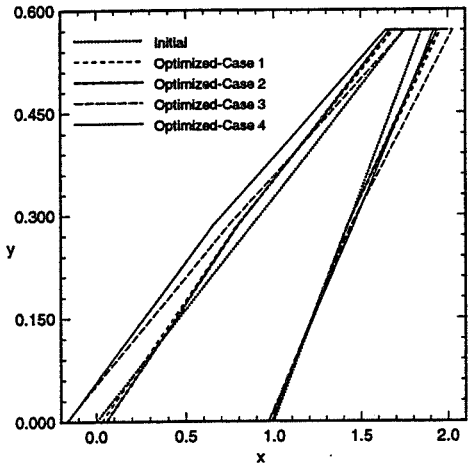


Mid Section

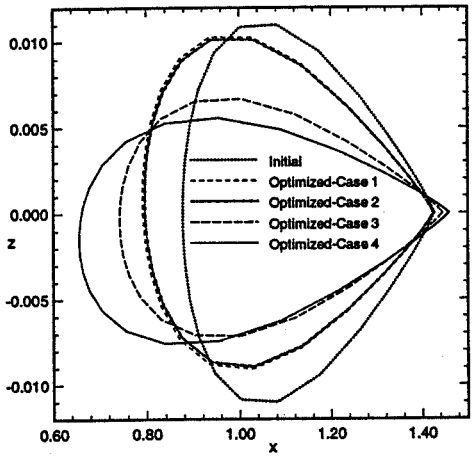


Tip Section

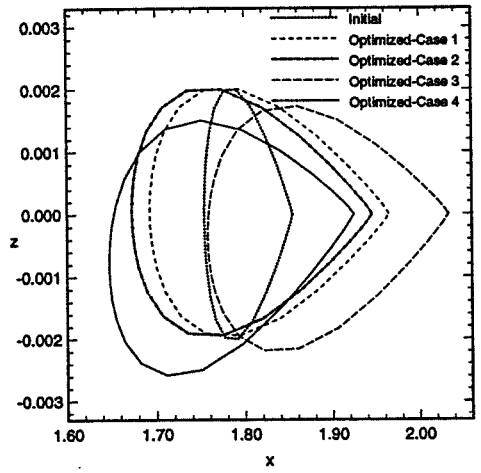
Figure 3: Effects of grid refinement and viscosity on the chordwise pressure coefficient distributions. M=2.4, aoa=4.5-deg.



Planform

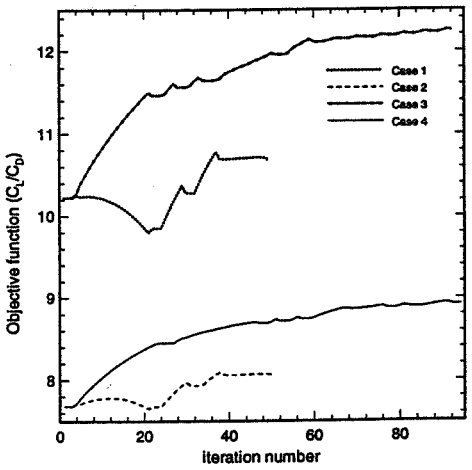


Mid Section

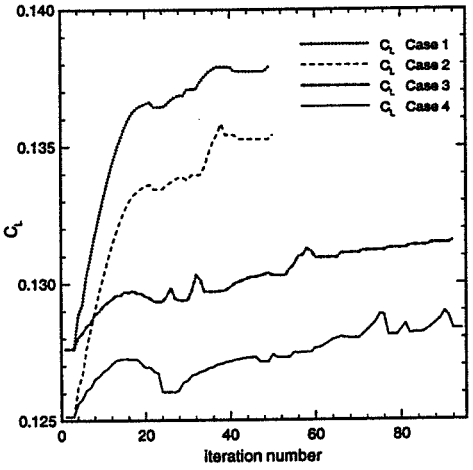


Tip Section

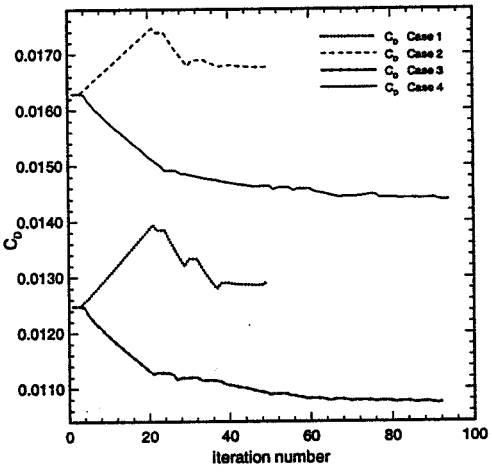
Figure 4: Comparison of optimized shapes from cases 1–4



Objective Function

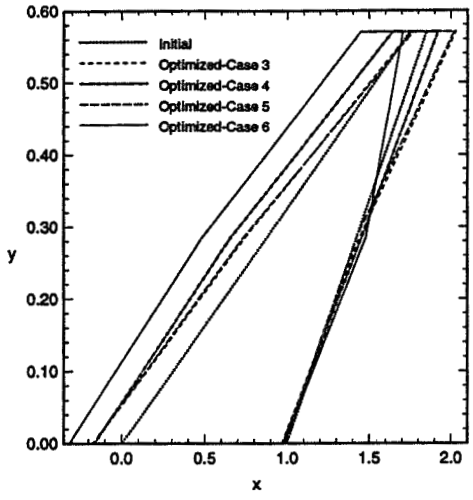


Lift Coefficient

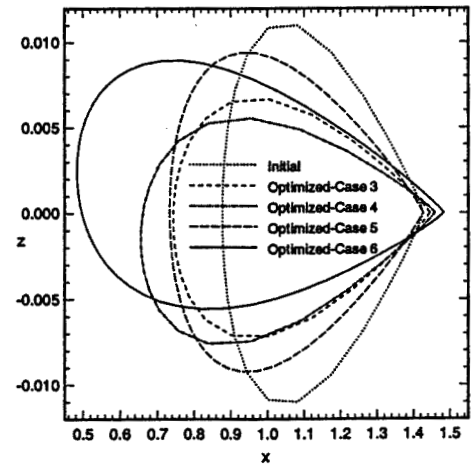


Drag Coefficient

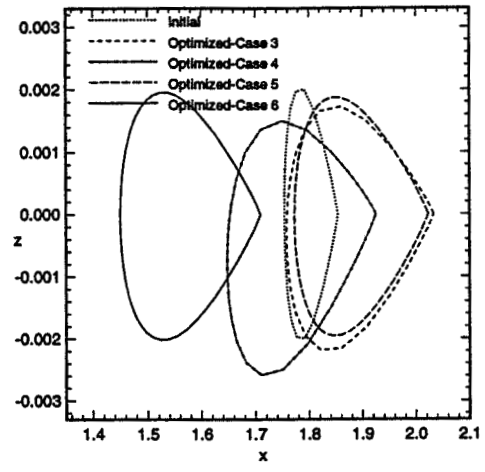
Figure 5: Evolution of objective function and aerodynamic coefficients from optimization cases 1–4



Planform

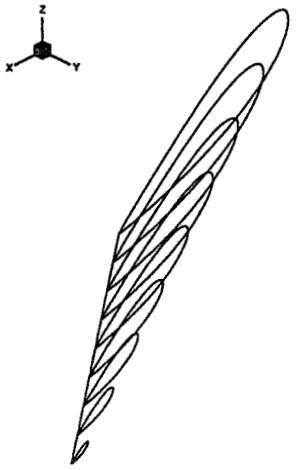


Mid Section

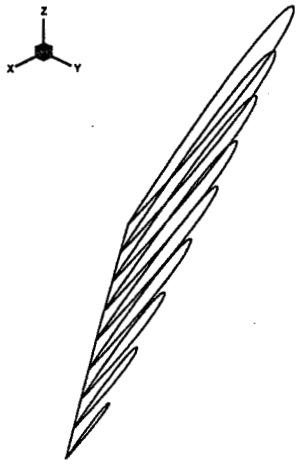


Tip Section

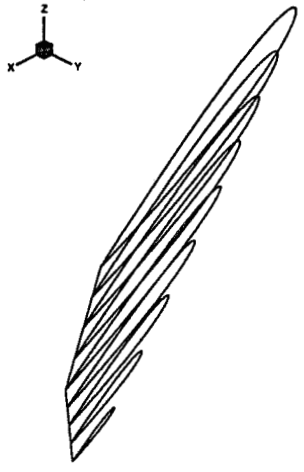
Figure 6: Comparison of optimized shapes from cases 3–6



Initial arrow wing

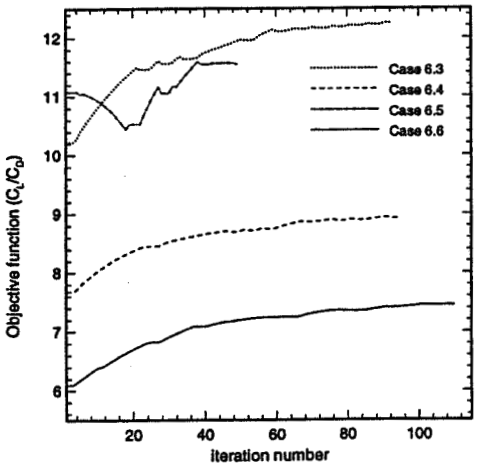


Optimized arrow wing (Case 4)

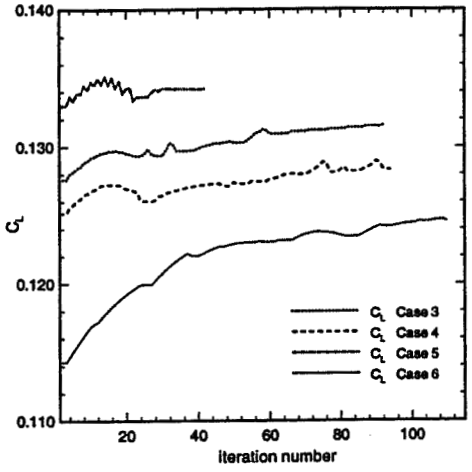


Optimized arrow wing (Case 6)

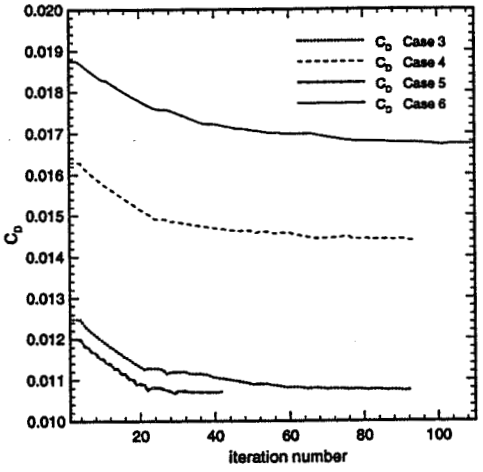
Figure 7: Perspective view of initial and optimized shapes from cases 6.4 and 6.6.



Objective Function



Lift Coefficient



Drag Coefficient

Figure 8: Evolution of objective function and aerodynamic coefficients from optimization cases 3–6