



AIAA Abstract

Aerodynamic Design of Complex Configurations Using Cartesian Methods and CAD Geometry

Marian Nemec, Michael J. Aftosmis,
and Thomas H. Pulliam

*NASA Ames Research Center
MS T27B-1, Moffett Field, CA 94035*

42nd AIAA Aerospace Sciences Meeting
Jan. 5–8, 2004/Reno, NV

For permission to copy or to republish, contact the copyright owner named on the first page.
For AIAA-held copyright, write to AIAA Permissions Department,
1801 Alexander Bell Drive, Suite 500, Reston, VA, 20191-4344.

Aerodynamic Design of Complex Configurations Using Cartesian Methods and CAD Geometry

Marian Nemec,* Michael J. Aftosmis,[†]
and Thomas H. Pulliam[‡]

NASA Ames Research Center
MS T27B-1, Moffett Field, CA 94035

Introduction

AERODYNAMIC design is inherently a multi-disciplinary problem that involves complex surface geometry, competing objectives, multiple operating conditions, and strict design constraints. Consequently, important considerations for an effective optimization framework include: 1) optimization techniques and the degree of coupling with the underlying solvers, 2) geometry parameterization, 3) objective function and constraint specification, and 4) mesh-generation and mesh-perturbation methods. One approach for constructing such frameworks is to view the parameterized geometry as the center of the optimization procedure. In modern engineering design environments, the surface geometry is generally produced from a parametric CAD solid representation. Ideally, it is this representation, accessible in its native environment, that should serve as the basis of the optimization procedure.

Recently, a promising approach has been developed using a standardized application programming interface that allows direct access to the native CAD solid representation. This approach is *vendor neutral*, i.e. independent of a specific CAD system, and is based on the Computational Analysis and PRogramming Interface (CAPRI).¹⁻³ In addition to providing an effective tool for surface discretization, CAPRI exposes the master-model feature tree of the CAD solid representation and allows direct modifications of the parameters within the tree. Hence, the design variables and geometric constraints can be intrinsic to the CAD solid representation. Although this approach is conceptually very appealing, its implementation presents significant challenges and potential pitfalls, for example:

- The efficiency of the geometry updates and the surface discretization depend on the efficiency of the proprietary CAD geometry kernel.
- The use of “legacy” geometry, or geometry with no parametric CAD representation, requires special consideration.
- Practical issues such as the number of available CAD licenses, and the requirement for a communication protocol between CAD workstations and dedicated compute platforms may significantly reduce the efficiency of a parallel optimization procedure.

Assuming that a CFD-ready surface discretization can be obtained, we turn our attention to mesh-generation and mesh-perturbation schemes. Since structured and unstructured mesh generation algorithms can be computationally expensive and usually require some user supervision, mesh-perturbation schemes⁴⁻⁶ are used during the optimization process to modify a given baseline mesh. These schemes preserve the initial mesh topology, which permits the use of simpler and faster solution-transfer algorithms and helps maintain a smoother design landscape. For sufficiently large geometry changes, however, the mesh-perturbation schemes breakdown and a new mesh must be created.

Cartesian methods⁶⁻⁹ offer a promising alternative, since the mesh generation is fast, robust, and perhaps the closest to being truly automatic. Due to the decoupling of the surface discretization from the volume mesh, Cartesian mesh generation is virtually insensitive to the complexity of the input geometry. A difficulty associated with this approach arises from the arbitrary intersection of the surface geometry with the mesh, resulting in a layer of cut-cells. As the surface geometry evolves during the optimization, the surface resolution as well as the number of cells within the mesh may change, becoming a potential source of noise in the design landscape.

The selection of an optimization technique is a critical factor in attaining an accurate, robust, and efficient

*NRC Research Associate, nemec@nas.nasa.gov

[†]Research Scientist, Senior Member AIAA

[‡]Associate Fellow AIAA, tpulliam@mail.arc.nasa.gov

Copyright © 2004 by the American Institute of Aeronautics and Astronautics, Inc. No copyright is asserted in the United States under Title 17, U.S. Code. The U.S. Government has a royalty-free license to exercise all rights under the copyright claimed herein for Governmental Purposes. All other rights are reserved by the copyright owner.

optimization procedure. Our primary requirements include: 1) scalability of the optimization technique in a parallel computing environment, 2) degree of coupling among the modules and the high-fidelity solvers within the framework, and 3) insensitivity to the presence of noise. In addition to the potential mesh-related noise, the use of steady-state CFD simulations for complex three-dimensional configurations may introduce a level of uncertainty in the evaluation of aerodynamic performance. This is mostly due to factors such as local flow unsteadiness and aspects associated with the numerical method, which may hinder deep convergence of the flow solution and influence the behavior of the optimization technique.

Hence, it is desirable to construct a sufficiently flexible framework to serve as a test-bed for various optimization techniques. For the problems under consideration here, the most promising techniques range from autonomous approaches such as evolutionary¹⁰⁻¹² and finite-difference gradient-based algorithms to methods requiring greater coupling such as the adjoint approach¹³⁻¹⁶ for gradient computations. Furthermore, the use of these techniques in conjunction with approximation methods, for example local response surface and surrogate models,^{17,18} can help deal with noisy design landscapes and reduce the computational cost of the optimization.

The objective for this paper is to present the development of an optimization capability for the Cartesian inviscid-flow analysis package of Aftosmis *et al.*^{8,19} We evaluate and characterize the following modules within the new optimization framework:

- A component-based geometry parameterization approach using a CAD solid representation and the CAPRI interface.
- The use of Cartesian methods in the development of automated optimization tools.
- Optimization techniques using a genetic algorithm and a gradient-based algorithm.

The discussion and investigations focus on several real-world problems of the optimization process. We examine the architectural issues associated with the deployment of a CAD-based design approach in a heterogeneous parallel computing environment that contains both CAD workstations and dedicated compute nodes. In addition, we study the influence of noise on the performance of optimization techniques, and the overall efficiency of the optimization process for aerodynamic design of complex three-dimensional configurations.

Problem Formulation

The aerodynamic optimization problem consists of determining values of design variables X , such that the objective function \mathcal{J} is minimized

$$\min_X \mathcal{J}(X, Q) \quad (1)$$

subject to constraint equations C_j :

$$C_j(X, Q) \leq 0 \quad j = 1, \dots, N_c \quad (2)$$

where the vector Q denotes the conservative flowfield variables and N_c denotes the number of constraint equations. The flowfield variables are forced to satisfy the governing flowfield equations, \mathcal{F} , within a feasible region of the design space Ω :

$$\mathcal{F}(X, Q) = 0 \quad \forall X \in \Omega \quad (3)$$

which implicitly defines $Q = f(X)$. The governing flow equations are the three-dimensional Euler equations of a perfect gas, where the vector $Q = [\rho, \rho u, \rho v, \rho w, \rho E]^T$.

For the examples under consideration here, the objective function is given by

$$\mathcal{J} = \begin{cases} \omega_L \left(1 - \frac{C_L}{C_L^*}\right)^2 + \omega_D \left(1 - \frac{C_D}{C_D^*}\right)^2 & \text{if } C_D > C_D^* \\ \omega_L \left(1 - \frac{C_L}{C_L^*}\right)^2 & \text{otherwise} \end{cases} \quad (4)$$

where C_D^* and C_L^* represent the target drag and lift coefficients, respectively. The weights ω_D and ω_L are user specified constants. This objective can be used for both lift-enhancement and lift-constrained drag minimization problems, and additional terms such as the moment coefficient can be readily included.

Two separate approaches are used for geometry parameterization and the definition of design variables. The first is an in-house CAD system based on the work of Charlton.²⁰ Although not as general as a commercial CAD system, it provides a parametric geometry definition for configurations such as a business jet and incurs relatively small computational cost. We use this approach to evaluate and benchmark our geometry communication architecture and test the individual modules of the optimization framework. The second approach is based on the master-model interface of CAPRI,³ which allows access to the CAD solid representation of most commercial CAD systems. Design variables are associated directly with values exposed in the feature tree. Geometry constraints are expressed parametrically within the CAD solid representation to avoid a-physical configurations, for example wings that detach from the fuselage.

The central idea behind both approaches is to use a component-based geometry paradigm. Since CAD solid representations typically rely on the use of parametric B-splines, the computation of component intersections is costly within the CAD system. In the present approach, the components are intersected after the surface discretization. This operation is performed efficiently as a part of the mesh generation process using the approach of Aftosmis *et al.*⁸ We tag components that experience only translations and rotations in order to avoid unnecessary re-triangulations.

Numerical Method

The details of the numerical method will be provided in the complete paper. Only a brief summary is given here.

We cast the optimization problem as an unconstrained problem by lifting the side constraints, Eq. 2, into the objective function \mathcal{J} using a penalty method. The constraint imposed by the flowfield equations, Eq. 3, is satisfied at every point within the feasible design space, and consequently these equations do not explicitly appear in the formulation of the optimization problem. This formulation allows a direct comparison between various optimization techniques and serves as a good starting point for the development of an hybrid approach. The primary optimization techniques are the genetic algorithm of Holst and Pulliam,¹² and an unconstrained BFGS quasi-Newton algorithm coupled with a backtracking line search.¹⁶ The flow equations are solved using the parallel steady-state Cartesian solver of Aftosmis *et al.*¹⁹

We exploit coarse-grained parallelization in the implementation of each module within the optimization framework. For both the genetic algorithm and the gradient-based method, this involves constructing the geometry, surface discretization, and the computational mesh in parallel and prior to any flow solutions. Note that the individual algorithms used for these tasks are serial algorithms. This is in contrast to flow solutions, where we leverage the parallel efficiency of the flow solver. A subset of the available processors is used for each flow solution, such that the degradation in performance due to scalar I/O is minimized. For example, for the cases presented in the next section, we typically use a total of 64 processors and select a subset of 16 processors for each flow solution, which allows us to run four flow solutions in parallel.

The objective function gradient is evaluated using central-differences. Following a base-state flow solution, which is computed using all available processors, we "warm-start" the finite-difference gradient computations. The solution-transfer algorithm is described by Murman *et al.*²¹ and an example is provided in the following section.

Results and Discussion

For this abstract, a relatively simple design example is considered that demonstrates the capability of the present optimization framework and introduces factors which will be fully discussed in the final paper.

The design example considered is the optimization of a business jet wing-body configuration. The freestream Mach number, M_∞ , is set to 0.84 and the initial angle of attack, α , is 4° . The design variables are the sweep angle of the wing and the angle of attack. The sweep angle for the initial configuration is 16.0° . The following two computational meshes for the

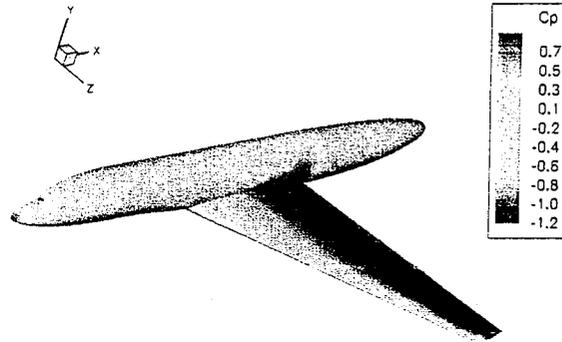


Fig. 1 Surface pressure coefficient for initial wing-body configuration, Sweep = 16° , $\alpha = 4^\circ$

symmetric half-body configuration are used to demonstrate the results:

Mesh A Coarse mesh consisting of 237,912 cells with 11,073 cut-cells on the body.

Mesh B Medium mesh consisting of 519,062 cells with 41,298 cut-cells on the body.

Note that mesh B has a roughly four times better geometry resolution when compared with mesh A.

The surface pressure coefficient for the initial configuration is shown in Fig. 1 using mesh B. The initial values of C_L and C_D are 0.3 and 0.0340, respectively. The objective function is given by Eq. 4, where we set $C_L^* = 0.2$ and $C_D^* = 0.006$. The weights ω_L and ω_D are set to 1.0. The total wetted-surface area of the configuration is constrained to equal the initial area using an additional quadratic term in Eq. 4.

We use mesh A to examine the convergence of warm-started flow solutions for central-difference gradient computations. Figure 2 shows an example convergence history for a warm-started flow solution. First, 250 multigrid cycles are performed for the baseline solution using a 3-level full multigrid. To prevent limiter fluctuations from contaminating the convergence of the flow solver, the flow solver is run in first-order mode. Next, we perturb the sweep design variable using a relative stepsize of 2×10^{-3} . This stepsize is chosen on the basis of numerical experiments that will be summarized in the final paper. The perturbed mesh has 124 additional cells. The baseline solution is transferred to the new mesh and 100 additional multigrid cycles are performed. As shown in Fig. 2, the warm-started flow solution convergences to the same level as the baseline solution in roughly 65 additional multigrid cycles. Perturbations of the angle of attack design variable usually converge within just 50 multigrid cycles, resulting in a saving of 2/3 in computational effort.

We perform the optimization of the configuration using mesh B. The surface resolution of mesh A is not sufficient to ensure a smooth design landscape.

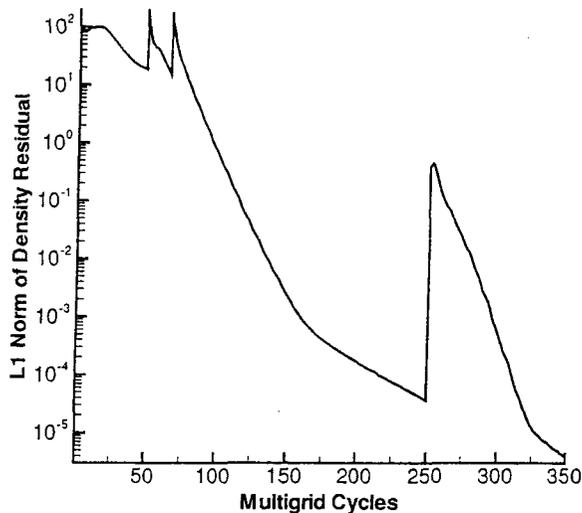


Fig. 2 Convergence history for baseline and warm-started flow solutions (sweep design variable)

Table 1 Wallclock time for individual optimization modules (400 MHz SGI Origin 3000, 32 processors)

Module	Time (s)
Geometry Generation ^a	32.0
Mesh Generation ^b	47.0
Flow Solution	435.0
Mesh Solution Transfer	7.0
Warm-Start Flow Solution	216.6

^a Includes surface creation and triangulation

^b Includes component intersection, mesh generation, domain decomposition, and coarse-mesh generation

The cost of the main modules within the optimization framework is summarized in Table 1. The timings are obtained using 32 processors on a 400 MHz SGI Origin 3000 system and are averaged over ten design iterations.

Figure 3 reveals the design landscape and compares the convergence of the genetic and quasi-Newton algorithms. The contour map is based on a coarse sampling of the design space, which causes the apparent non-smoothness. The number of chromosomes (population size at each generation) is set to 16 for the genetic algorithm. We assign each chromosome in the first generation the same initial condition, which is a sweep of 16° and an angle of attack of 4°, in order to make a more meaningful comparison of the optimization algorithms.

Each square symbol in Fig. 3 denotes the best individual within a generation. Approximately ten generations (160 flow solutions) are required to reach the optimal solution. The delta symbols denote the best

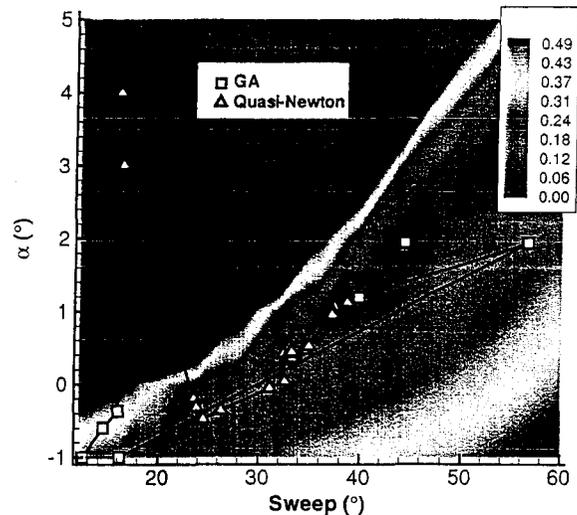


Fig. 3 Design landscape and convergence of genetic algorithm and quasi-Newton method

design of the quasi-Newton algorithm after each successful line search. The gradient method stalled after 48 flow solutions and 96 warm-started flow solutions, which are used for gradient computations. Attempts to restart the quasi-Newton algorithm by resetting the approximate Hessian matrix did not improve the design. Although the quasi-Newton algorithm descended into the “valley” of the design landscape within 2 iterations, its progress in this relatively shallow and curving valley is slow. This is not only due to the poor scaling of the objective function, but also due to the presence of noise in the evaluation of the objective function. We will examine this issue in the final paper.

Comparison of the initial and final designs is shown in Fig. 4. The genetic algorithm converged to a design with $C_L = 0.16$ and $C_D = 0.00636$. For the quasi-Newton algorithm, the aerodynamic coefficients of the final design are $C_L = 0.154$ and $C_D = 0.00634$. Hence, in terms of performance the two designs are quite similar.

Acknowledgments

This work was performed while the first author held a National Research Council Research Associateship Award at the NASA Ames Research Center.

References

- ¹Haines, R. and Follen, G., “Computational Analysis Programming Interface,” *Proceedings of the 6th International Conference on Numerical Grid Generation in Computational Field Simulations*, edited by Cross, Eiseman, Hauser, Soni, and Thompson, University of Greenwich, 1998.
- ²Aftosmis, M. J., Delanaye, M., and Haines, R., “Automatic Generation of CFD-Ready Surface Triangulations from CAD Geometry,” AIAA Paper 99-0776, Reno, NV, Jan. 1999.
- ³Haines, R. and Aftosmis, M. J., “On Generating High Quality “Water-tight” Triangulations Directly from CAD,” Tech. rep., Meeting of the International Society for Grid Generation, (ISGG), Honolulu, HI, June 2002.

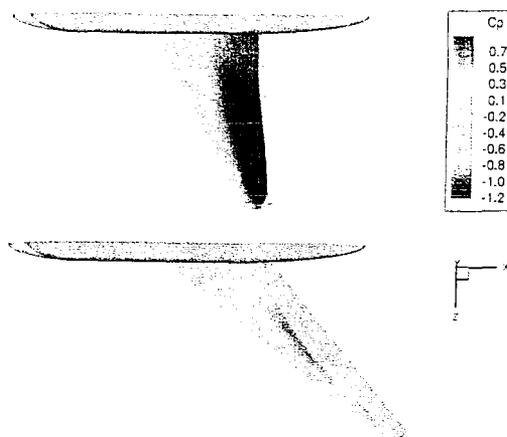


Fig. 4 Comparison of initial and final designs

⁴Nielsen, E. J. and Anderson, W. K., "Recent Improvements in Aerodynamic Design Optimization on Unstructured Meshes," *AIAA Journal*, Vol. 40, No. 6, 2002, pp. 1155-1163.

⁵Reuther, J. J., Alonso, J. J., Rimlinger, M. J., and Jameson, A., "Aerodynamic Shape Optimization of Supersonic Aircraft Configurations via an Adjoint Formulation on Distributed Memory Parallel Computers," *Computers & Fluids*, Vol. 28, 1999, pp. 675-700.

⁶Cliff, S. E., Thomas, S. D., Baker, T. J., and Jameson, A., "Aerodynamic Shape Optimization Using Unstructured Grid Methods," AIAA Paper 2002-5550, Atlanta, GA, Sept. 2002.

⁷Aftosmis, M. J., "Solution Adaptive Cartesian Grid Methods for Aerodynamic Flows with Complex Geometries," Lecture notes, von Karman Institute for Fluid Dynamics, Series: 1997-02, Brussels, Belgium, March 1997.

⁸Aftosmis, M. J., Berger, M. J., and Melton, J. E., "Robust and Efficient Cartesian Mesh Generation for Component-Based Geometry," *AIAA Journal*, Vol. 36, No. 6, 1998, pp. 952-960.

⁹Rodriguez, D. L., "Response Surface Based Optimization With a Cartesian CFD Method," AIAA Paper 2003-0465, Jan. 2003.

¹⁰Obayashi, S., "Aerodynamic Optimization with Evolutionary Algorithms," *Inverse Design and Optimization Methods, Lecture Series 1997-05*, edited by R. A. Van den Braembussche and M. Manna, von Karman Institute for Fluid Dynamics, Brussels, Belgium, 1997.

¹¹Marco, N., Désidéri, J.-A., and Lanteri, S., "Multi-Objective Optimization in CFD by Genetic Algorithms," Tech. Rep. 3686, Institut National De Recherche En Informatique Et En Automatique (INRIA), France, April 1999, Also see www.lania.mx/~ccoello/EMOO/EMOObib.html.

¹²Holst, T. L. and Pulliam, T. H., "Aerodynamic Shape Optimization Using a Real-Number-Encoded Genetic Algorithm," AIAA Paper 2001-2473, Anaheim, CA, June 2001.

¹³Jameson, A., "Aerodynamic Shape Optimization Using the Adjoint Method," Lecture notes, von Karman Institute for Fluid Dynamics, Brussels, Belgium, Feb. 2003.

¹⁴Nielsen, E. J. and Anderson, W. K., "Aerodynamic Design Optimization on Unstructured Meshes Using the Navier Stokes Equations," *AIAA Journal*, Vol. 37, No. 11, 1999, pp. 1411-1419.

¹⁵Elliott, J. and Peraire, J., "Constrained, Multipoint Shape Optimisation for Complex 3D Configurations," *Aeronautical Journal*, Vol. 102, No. 1017, 1998, pp. 365-376.

¹⁶Nemec, M. and Zingg, D. W., "Newton-Krylov Algorithm for Aerodynamic Design Using the Navier-Stokes Equations," *AIAA Journal*, Vol. 40, No. 6, 2002, pp. 1146-1154.

¹⁷Alexandrov, N. M., Nielsen, E. J., Lewis, R. M., and Anderson, W. K., "First-Order Model Management with Variable-

Fidelity Physics Applied to Multi-Element Airfoil Optimization," AIAA Paper 2000-4886, September 2000.

¹⁸Ong, Y. S., Nair, P. B., and Keane, A. J., "Evolutionary Optimization of Computationally Expensive Problems via Surrogate Modeling," *AIAA Journal*, Vol. 41, No. 4, 2003, pp. 687-696.

¹⁹Aftosmis, M. J., Berger, M. J., and Adomavicius, G., "A Parallel Multilevel Method for Adaptively Refined Cartesian Grids with Embedded Boundaries," AIAA Paper 2000-0808, Reno, NV, Jan. 2000.

²⁰Charlton, E. F., *An Octree Solution to Conservation-laws over Arbitrary Regions (OSCAR) with Applications to Aircraft Aerodynamics*, Ph.D. thesis, University of Michigan, 1997.

²¹Murman, S. M., Aftosmis, M. J., and Berger, M. J., "Implicit Approaches for Moving Boundaries in a 3-D Cartesian Method," AIAA Paper 2003-1119, Reno, NV, Jan. 2003.