ABSTRACT FOR 39TH AIAA AEROSPACE SCIENCES MEETING

Issues Related to Aerodynamic Design Optimization On Unstructured Meshes

Eric J. Nielsen* and W. Kyle Anderson†
NASA Langley Research Center, Hampton, VA 23681-2199

Abstract

Several limiting factors to large-scale aerodynamic design on unstructured meshes are addressed. A set of design codes based on a discrete adjoint method is extended to a multiprocessor environment using a shared memory approach. A nearly linear speedup is demonstrated, and the consistency of the linearizations is shown to remain valid. The full linearization of the residual is used to precondition the adjoint system, and a significantly improved convergence rate is obtained. A new mesh movement algorithm is implemented and several advantages over an existing technique are presented. A large-scale inviscid optimization is performed in which a substantial drag reduction is demonstrated. An inviscid design case using a multielement wing is also shown.

Introduction

With the advent of modern computer architectures, aerodynamic designers have sought to make use of high-fidelity computational fluid dynamics (CFD) codes in their everyday design efforts. While considerable progress has been made towards this goal, realistic use of such tools remains hindered by the extreme computational burden associated with such tasks.

A large focus has recently been placed on design algorithms. In the area of gradient-based optimization, research has focused on several methods for obtaining sensitivity information, and many of these approaches rely on an adjoint-variable formulation for efficiently computing sensitivity derivatives. The adjoint technique is particularly attractive for aerodynamic design problems in which there are a large number of design variables, yet relatively few constraints. Examples of both continuous and discrete approaches to this method can be found in Refs. 1-9.

In Refs. 1-3, a discrete adjoint technique has been implemented on unstructured grids for two- and three-dimensional flows. This work was primarily aimed at performing accurate linearizations of Reynolds-averaged Navier-Stokes solvers, using both compressible and incompressible formulations. Results indicated highly accurate sensitivity information for fully turbulent flows. However, the cost of such computations in a sequential-processing environment prevented large-scale design cases from being pursued. The preconditioning strategy used for the adjoint system in these references was based on a first-order linearization of the residual and often led to poor convergence rates. In addition, experience showed that the tension-spring analogy used for mesh movement was insufficient when large changes in the geometry were necessary.

In the current work, the linearizations developed in Refs. 1-3 are modified to run in a parallel processing environment. The domain decomposition and parallelization strategies are discussed, resulting speedups are demonstrated, and the linearizations are shown to remain consistent. A new preconditioning strategy for the adjoint solver is implemented and significantly improved convergence is demonstrated for turbulent flow. A new mesh movement strategy based on linear elasticity theory is also adopted, and several advantages over the tension-spring approach are presented. Several design cases are also shown.

*Research Scientist, Computational Modeling and Simulation Branch, Aerodynamics, Aerothermodynamics, and Acoustics Competency, Member AIAA.

†Formerly Senior Research Scientist, Computational Modeling and Simulation Branch, Aerodynamics, Aerothermodynamics, and Acoustics Competency. Currently CEO, Symantec Corp., Assoc. Fellow AIAA.

Nomenclature

Lift and drag coefficients
Chord
Vector of design variables
Cost function
Lagrangian function
Vector of dependent variables
Discretized residual vector
Nodal displacements
Vector of nodal displacements
Computational mesh
Vector of costate variables
Poisson's ratio

Design Methodology

The governing flow equations are the Reynolds-averaged Navier-Stokes equations,[10] coupled with the one-equation turbulence model of Spalart.[11] The solvers employed in the current work make use of a finite-volume formulation discretized on unstructured meshes and are described at length in Refs. 3, 12, and 13. The meshes used in this study have been generated using the software described in Refs. 14 and 15.

Given a steady-state flow solution in the form of , a Lagrangian function can be defined as

(1)

where represents a cost function to be minimized and represents a vector of Lagrange multipliers, or costate variables. Differentiating this expression yields the following:

(2)

Since the vector of costate variables is essentially arbitrary, the coefficient multiplying can be eliminated using the following equation:

(3)

Eq. 3 represents the discrete adjoint equation for the design problem. Once the solution for has been formed, the remaining terms in Eq. 2 can be evaluated to give the desired sensitivity information:

(4)

Details on the implementation and solution of these equations is presented in Refs. 1-4.

Domain Decomposition Methodology

In the current work, the mesh partitioner MeTiS[16] is used to divide the original mesh into subdomains suitable for a parallel environment. Given the connectivities associated with each node in the mesh and the number of partitions desired, MeTiS returns an array that designates a partition number for each node in the mesh. The user is then responsible for extracting the data structures required by the specific application.

Due to the gradient terms used in the reconstruction procedure, achieving second-order accuracy in the flow solver requires information from the neighbors of each mesh point as well as their neighbors. In the present implementation, the gradients of the dependent variables are computed on each mesh partition, then the results are scattered onto neighboring partitions. This approach dictates that a single level of "ghost" nodes be stored on each processor. These ghost nodes that are connected to mesh points on the current partition are referred to as "level-1" nodes. Similarly, the neighbors of level-1 nodes that do not lie on the current partition are designated "level-2" nodes. This terminology is illustrated graphically in Figure 1.

Partition
Boundary

Node on the current partition

Level-1 ghost node

Level-2 ghost node

Figure 1. Information required beyond partition boundaries.

The adjoint solver requires similar information; however, unlike the flow solver, residual contributions must be written into off-processor memory locations associated with level-2 mesh points. This implies that a second level of ghost information must be retained along partition boundaries.

Software has been developed to extract the required information from a pre-existing mesh based on the partitioning array provided by MeTiS. This domain decomposition operation is done prior to performing any computations. The user is also able to read in existing subdomains and their corresponding solution files and repartition as necessary. This capability is useful in the event that additional processors become available or processors currently being employed must be surrendered to other users. In addition, software has been developed that reassembles partition information into global files and aids in post-processing the solutions.

Parallelization Strategy

Each of the codes has been modified to run in a multiprocessor environment using a shared memory implementation. This approach has been chosen because the primary hardware to be utilized is a Silicon Graphics Origin 2000 system. In the current implementation, ghost information is exchanged across partition boundaries by loading data into global shared arrays which are accessible from each processor. Simple compiler directives specific to the Origin 2000 system are used to spawn child processes for each partition in the mesh. This approach is also readily extendable to a message-passing implementation, in the event that distributed memory systems become more readily available.

The speedup obtained by parallelizing the flow and adjoint solvers is demonstrated in Figs. 2 and 3. For this test, turbulent flow over the ONERA M6 wing shown in Fig. 4 is computed. The mesh contains 359,536 nodes with a wall spacing of of the mean aerodynamic chord (MAC). The surface mesh consists of 9,129 nodes. The freestream Mach number is 0.3, the angle of attack is , and the Reynolds number is based on the MAC. It can be seen that a nearly linear speedup is obtained. Similar behavior has been observed with the mesh movement and gradient evaluation codes.

Linear
Actual

Figure 2. Parallel speedup obtained for the flow solver.

Linear
Actual

Figure 3. Parallel speedup obtained for the adjoint solver.

Figure 4. Surface mesh for viscous ONERA M6 wing.

To verify that the linearizations performed in Refs. 1-3 have remained consistent through the port to the parallel environment, sensitivity derivatives obtained using the parallel solvers on eight processors are compared with centered finite differences. Here, turbulent flow over an ONERA M6 wing is computed using a freestream Mach number of 0.3, an angle of attack of , and a Reynolds number of based on the MAC. The mesh used for this case consists of 16,391 nodes. All results have been converged to machine accuracy, and a step size of has been used for the finite-difference computations. For this case, the cost function is a linear combination of lift and drag, and the design variables generated using the software described in Ref. 17 are depicted in Figure 21. It can be seen from Table 1 that the derivatives are highly consistent.

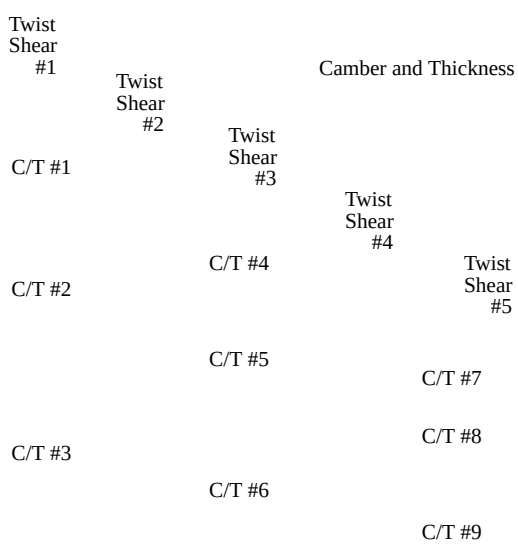

Figure 5. Location of design variables for ONERA M6 wing.

Table 1. Sensitivity derivatives for ONERA M6 wing.

Design Variable	Finite Difference	Adjoint	Percent Error
Camber #3	2.7762	2.7763	0.004%
Thickness #4	-0.03970	-0.03971	0.025%
Twist #4	0.00747	0.00747	0.000%
Shear #1	0.62023	0.62050	0.044%

Adjoint Preconditioning Scheme

In Refs. 1-4, a preconditioned GMRES algorithm has been used to solve Eq. 3. In these references, an incomplete LU-factorization with no fill-in allowed [ILU(0)] is employed as the preconditioner. The factorization is based on the first-order linearization of the residual, thereby avoiding excessive storage penalties associated with the higher-order stencil for the inviscid fluxes. It has been shown in Ref. 3 that the GMRES algorithm may stall and a converged adjoint solution may frequently be difficult to obtain using this preconditioner, particularly for viscous flows. This has been found to be the case for both two- and three-dimensional problems.

In an effort to develop a more robust adjoint solver, an improved ILU(0) preconditioning technique based on the complete linearization of the residual is employed in the current work. As shown in Ref. 18, the additional memory required for storing the complete linearization is roughly four times that of the first-order matrix for three-dimensional problems. This requirement can be somewhat alleviated by utilizing half-precision storage for these terms. As described in Refs. 2 and 3, the linearizations required for the matrix-vector products in the GMRES algorithm are stored for the nearest-neighbor terms; these linearizations are also stored in half-precision in the current work. Experiments have shown that this strategy yields a total memory requirement of about 50% more than the previous version of the solver.

To demonstrate the improved performance using the higher-order preconditioner, adjoint solutions are computed in parallel for turbulent flow over the ONERA M6 wing shown in Fig. 4 using eight processors. The freestream Mach number is , the angle of attack is , and the Reynolds number is based on the MAC. For this case, 10 GMRES cycles are used with 10 search directions and 5 restarts. The CFL number is set to . Results for the first- and second-order preconditioning strategies are shown in Fig. 6. It can be seen that the solver based on the first-order preconditioner fails to converge the solution, whereas the method employing the complete linearization steadily reduces the residual by nearly five orders of magnitude. Fig. 7 shows the same computations performed using full- and half-precision storage for the sec-

ond-order preconditioner and matrix-vector products. It can be seen that the convergence histories are identical.

Figure 6. Convergence of the adjoint solution for different preconditioners.

Figure 7. Convergence of the adjoint solution for full- and half-precision matrix storage schemes.

Mesh Movement Strategy

As stated in Refs. 1-4, a tension-spring analogy has previously been employed as a means for modifying volume meshes as the geometric shape is changed throughout the design process. It has been found that this algorithm lacks the robustness necessary for the design environment, particularly for large surface deformations, highly-stretched meshes, and essentially all three-dimensional geometries. For this reason, a new approach based on linear elasticity theory has been implemented to overcome the shortcomings of the previously used method.

In the approach taken in the current work, it is assumed that the computational mesh obeys the isotropic linear elasticity relations which take the following form in two dimensions:[19]

(5)

(6)

where is Poisson's ratio and the nodal displacement vector is given by . Despite the assumption of an isotropic material, a spatially-varying value of Poisson's ratio is used in order to maintain the physical integrity of highly skewed cells. This value has been chosen based on *ad hoc* numerical experiments and is set so that the coefficient is equal to the aspect ratio of the local cell. In this manner, low aspect ratio cells mimic compressible materials such as cork, while high aspect ratio cells tend to behave in an incompressible fashion, much like rubber. This mesh movement scheme has also been utilized in Ref. 20, with a similar modification for distorted cells.

The linear elasticity approach to mesh movement has demonstrated several advantages over the tension-spring analogy. For this test, the flap on a multielement airfoil has been deflected and each of the mesh movement strategies have been applied. Figure 8 shows a near-field view of the baseline mesh in the region between the main element and flap. Figures 9 and 10 show the meshes resulting from the spring and elasticity methods, respectively. It can be seen that the spring analogy allows gaps to form in the mesh, whereas the elasticity approach pulls in nearby material to fill the voids. This behavior has been observed in several cases.

Figure 8. Near-field view of baseline mesh.

Figure 9. Near-field view of mesh after applying spring analogy.

Figure 10. Near-field view of mesh after applying linear elasticity method.

It has been found that the elasticity approach also allows for significantly larger geometric deformations. In a similar test, the flap of a multielement airfoil has been deflected from its baseline position shown in Figure 11. As can be seen from Figs. 12 and 13, the spring approach has yielded an invalid mesh, while the elasticity formulation has handled the deformation in an acceptable manner. Similarly, when a series of flap translations and rotations is applied to the geometry shown in Figure 14, the meshes resulting from the elasticity technique maintain a high degree of quality as shown in Figure 15.

Figure 11. Near-field view of mesh with flap in baseline position.

Figure 12. Near-field view of mesh with flap rotated using the spring analogy.

Figure 13. Near-field view of mesh with flap rotated using linear elasticity method.

Figure 14. Mesh with flap in baseline position.

Figure 15. Mesh with flap translated and rotated .

To further quantify the differences between the two mesh movement schemes, derivatives of lift and drag due to horizontal translations of a main element and flap are examined. In theory, the derivative due to a translation of the flap should be equal and opposite in sign to a derivative due to an equal and opposite translation of the main element. In practice however, this relationship is limited by the finite topology of the mesh and the manner in which it varies during a shape modification.

To demonstrate this behavior, derivatives of lift and drag due to equal and opposite horizontal translations of the main element and flap on the two-element airfoil shown in Figure 1 have been computed for a turbulent flow. For this case, the freestream Mach number is 0.25, the Reynolds number is , and the angle of attack is . Table 2 shows the lift and drag derivatives due to translations of the main element and flap using the spring analogy, and the last column is the sum of these two derivatives which ideally would be zero. However, it can be seen from the table that the derivatives are not at all equal in magnitude, and the drag derivatives are not even of opposite sign. This inconsistency would surely have an adverse effect on an optimization procedure. Table 3 shows the same derivatives obtained using the linear elasticity formulation. Although these derivatives do not sum to zero, they do exhibit opposite signs and are much closer in magnitude. This tendency has been observed in several cases and seems to indicate that the linear elasticity formulation maintains the mesh topology in a more consistent fashion.

Figure 16. Geometry used for translation derivatives.

Table 2. Derivatives of lift and drag due to flap and main element translation using the spring analogy.

Derivative			
	-1.4785	2.4033	0.9248
	0.0183	0.0277	0.0460

Table 3. Derivatives of lift and drag due to flap and main element translation using linear elasticity.

Derivative			
	-3.8064	3.8671	0.0607
	0.1722	-0.1615	0.0107

Design Cases

Recovery of Experimental Flap Configuration

The first test case is a two-dimensional turbulent flow problem for which a target pressure distribution is sought. In Ref. x, an experimental study of the multielement airfoil geometry shown in Fig. 17 has been performed. It can be seen from Fig. 18 that computations using the baseline geometry are in slight disagreement with the experimental results. It has been found that the model used in the experiment had a non-uniform gap and overlap, and that the flap deflected at high dynamic pressures. The goal of the current work is to determine the position of the flap based on the pressure distribution obtained in the experiment. The improved mesh movement capability described earlier allows for the flap adjustment required by such a problem.

The freestream Mach number is 0.7, the angle of attack is , and the Reynolds number is . For this case, the design variables are the rotation and x- and y-translations of the flap. After x design cycles, the flap has been repositioned as shown in Fig. 17. It

can be seen from Fig. 18 that the resulting agreement with the experimental results is significantly improved.

Baseline
Modified

Figure 17. Baseline and modified geometries for multielement airfoil problem.

Baseline
Modified
Experiment

Figure 18. Pressure distributions for multielement airfoil problem.

Inviscid Wing Design

An inviscid wing optimization is performed using an ONERA M6 wing geometry. The mesh consists of 357,900 nodes and is shown in Figure 19. The freestream Mach number is 0.84 and the angle of attack is . For these conditions, the baseline geometry exhibits a swept shock extending from the root leading edge and a strong normal shock further aft as shown in Figure 20. The objective for this example is to reduce drag while maintaining a specified lift. The 28 shape design variables are shown in Figure 21, and the the angle of attack is also allowed to vary.

Cross-sections of the initial and final geometries can be seen in Figure 22. After 5 design cycles, the drag has been decreased from xxx to xxx, a 27% reduction. Pressure distributions at several locations across the span of the wing are shown in Figure 23, and density contours for the final geometry are shown in Figure 24.

Figure 19. Surface mesh for inviscid ONERA M6 wing.

Figure 20. Density contours for the baseline geometry.

Camber
Twist
Camber and Thickness
Twist
Twist
Twist

Figure 21. Location of design variables for ONERA M6 wing.

Figure 22. Cross-sections of the initial and final wing geometries.

Figure 23. Pressure distributions for the initial and final wing geometries.

Figure 24. Density contours for the final geometry.

Multielement Wing

In order to handle an arbitrary number of three-dimensional elements parameterized by the package described in Ref. 17, software has been developed to combine multiple bodies which employ independent parameterizations. To evaluate this capability, a simple inviscid three-dimensional test case is examined. For this problem, the airfoil section depicted in Fig. 25 is extruded in a spanwise direction to create a swept wing as shown in Fig. 26. The surface grid shown contains 48,077 nodes and the volume mesh consists of 249,037 nodes and 1,335,153 tetrahedra. Accurate analysis of such a geometry would require a finer mesh and incorporation of viscous

effects, however the problem is merely designed to validate the capability of handling multiple elements parameterized independently.

For this case, the main element and flap are parameterized separately using the 38 design variables shown in Fig. 27. The angle of attack is also allowed to vary. The objective is to reduce the drag while maintaining a specified lift. The freestream Mach number is 0.75 and the baseline angle of attack is .

After 5 design cycles, the drag has been reduced from 0.0185 to 0.0166, while the lift has maintained its original value. Cross-sections of the baseline and modified geometries can be seen in Fig. 28, while pressure distributions are shown in Fig. 29.

Figure 25. Airfoil section used for inviscid multielement wing.

Figure 26. Surface mesh for inviscid multielement wing.

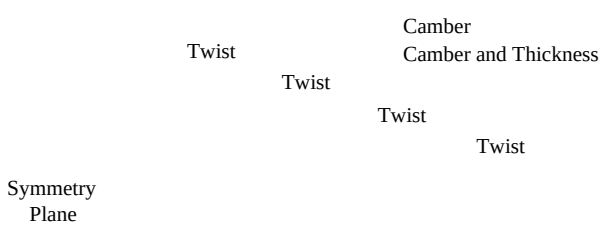

Figure 27. Location of design variables for multielement wing.

Figure 28. Cross-sections of the initial and final wing geometries.

Figure 29. Pressure distributions for the initial and final wing geometries.

Final Paper

For the final version of the paper, we plan to show more design examples, including a large-scale three-dimensional viscous test case. Comparisons with complex-variable derivatives may also be incorporated.

Acknowledgments

The authors would like to thank Jamshid Samareh for continued assistance with his geometric parameterization package.

References

[1]Anderson, W.K., and Bonhaus, D.L., “Aerodynamic Design on Unstructured Grids for Turbulent Flows,” NASA TM 112867, June 1997.

[2]Nielsen, E.J., and Anderson, W.K., “Aerodynamic Design Optimization on Unstructured Meshes Using the Navier-Stokes Equations,” AIAA Paper 98-4809, September 1998.

[3]Nielsen, E.J., “Aerodynamic Design Sensitivities on an Unstructured Mesh Using the Navier-Stokes Equations and a Discrete Adjoint Formulation,” Ph.D. Thesis, Virginia Polytechnic Institute and State University, 1998.

[4]Anderson, W.K., and Venkatakrishnan, V., “Aerodynamic Design Optimization on Unstructured Grids with a Continuous Adjoint Formulation,” AIAA Paper 97-0643, January 1997.

[5]Jameson, A., Pierce, N.A., and Martinelli, L., “Optimum Aerodynamic Design Using the Navier-Stokes Equations,” AIAA Paper 97-0101, January 1997.

[6]Reuther, J., Alonso, J.J., Martins, J.R.R.A., and Smith, S.C., “A Coupled Aero-Structural Optimization Method for Complete Aircraft Configurations,” AIAA Paper 99-0187, January 1999.

[7]Elliott, J., and Peraire, J., “Aerodynamic Optimization on Unstructured Meshes with Viscous Effects,” AIAA Paper 97-1849, June 1997.

[8]Soemarwoto, B., “Multipoint Aerodynamic Design by Optimization,” Ph.D. Thesis, Delft University of Technology, 1996.

[9]Mohammadi, B., “Optimal Shape Design, Reverse Mode of Automatic Differentiation and Turbulence,” AIAA Paper 97-0099, January 1997.

[10]White, F.M., Viscous Fluid Flow, McGraw-Hill, New York, 1974.

[11]Spalart, P.R., and Allmaras, S.R., “A One-Equation Turbulence Model for Aerodynamic Flows,” AIAA Paper 92-0439, January 1992.

[12]Anderson, W.K., and Bonhaus, D.L., “An Implicit Upwind Algorithm for Computing Turbulent Flows on Unstructured Grids,” Computers and Fluids, Vol. 23, No.1, 1994, pp. 1-21.

[13]Anderson, W.K., Bonhaus, D.L., McGhee, R., and Walker, B., “Navier-Stokes Computations and Experimental Comparisons for Multielement Airfoil Configurations,” AIAA Journal of Aircraft, Vol. 32, No. 6, 1995, pp. 1246-1253.

[14]Marcum, D.L. "Generation of Unstructured Grids for Viscous Flow Applications," AIAA Paper 95-0212, Jan. 1995.

[15]Pirzadeh, S., "Viscous Unstructured Three-Dimensional Grids by the Advancing-Layers Method," AIAA Paper 94-0417, Jan. 1994.

[16]Karypis, G., and Kumar, V., “A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs,” SIAM Journal of Scientific Computing, Vol. 20, No. 1, 1998, pp. 359-392.

[17]Samareh, J., “Geometry Modeling and Grid Generation for Design and Optimization,” ICASE/LaRC/NSF/ARO Workshop on Computational Aerosciences in the 21st Century, April 22-24, 1998.

[18]Barth, T. "Parallel CFD Algorithms on Unstructured Meshes," Paper presented in an AGARD-FDP-VKI special course on "Parallel Computing in CFD," held at the Von Karman Institute, Rhode-Saint-Genese, Belgium, from 15-19 May 1995 and 16-20 October 1995 at NASA Ames, United States and published in R-807.

[19]Pilkey, W.D., and Wunderlich, W., Mechanics of Structures: Variational and Computational Methods, CRC Press, Florida, 1994.

[20]Johnson, A.A., and Tezduyar, T.E., "Simulation of Multiple Spheres Falling in a Liquid-Filled Tube," Comput. Methods Appl. Mech. Engrg., Vol. 134, 1996, pp. 351-373.

[21]2D experiment reference