

GEORGE S. DULIKRAVICH

INTERNATIONAL CENTRE FOR MECHANICAL SCIENCES

COURSES AND LECTURES - No. 366



NEW DESIGN CONCEPTS
FOR HIGH SPEED AIR TRANSPORT

EDITED BY

H. SOBIECZKY
DLR GERMAN AEROSPACE RESEARCH ESTABLISHMENT



SpringerWienNewYork

GEORGE S. DULIKRAVICH

INTERNATIONAL CENTRE FOR MECHANICAL SCIENCES

COURSES AND LECTURES - No. 366



NEW DESIGN CONCEPTS
FOR HIGH SPEED AIR TRANSPORT

EDITED BY

H. SOBIECZKY
DLR GERMAN AEROSPACE RESEARCH ESTABLISHMENT



SpringerWienNewYork

Chapter 10

Aerodynamic Shape Inverse Design Methods

George S. Dulikravich

10.1 Introduction

Aerodynamic problems are defined by the governing partial differential or integral equations, shapes and sizes of the flow domains, boundary and initial conditions, fluid properties, and by internal sources and external inputs of mass, momentum and energy. In the case of an analysis (direct problem) we are asked to predict the details of a flow-field if the shape(s) and size(s) of the object(s) are given. In the case of a design (inverse or indirect problem) we are asked to determine the shape(s) and size(s) of the aerodynamic configuration(s) that will satisfy the governing flow-field equation(s) subject to specified surface pressure or velocity boundary conditions and certain geometric constraints [130]-[138]. The entire design technology is driven by the increased industrial demand for reduction of the design cycle time and minimization of the need for the costly a posteriori design modifications.

Aerodynamic inverse design methodologies can be categorized as belonging to surface flow design and flow-field design. Surface flow design is based on specifying pressure, Mach number, etc. on the surface of the object, then finding the shape of the object that will generate these surface conditions. Flow-field design enforces certain global flow-field features (shock-free conditions, minimal entropy generation, etc.) at every point of the flow-field by determining the shape that will satisfy these constraints. An arbitrary distribution of the surface flow parameters or an arbitrary field distribution of the flow parameters could result in aerodynamic shapes that either cross over ("fish tail" shapes) or never meet ("open trailing edge" shapes). These problems can be avoided by appropriately constraining the surface distribution of the flow parameters [139].

It should be pointed out that inverse methods for aerodynamic shape design are capable of creating only point-designs, that is, the resulting shapes will have the desired aerodynamic characteristics only at the design conditions. If the angle of attack, free stream Mach number, etc. in actual flight situations are different from the values used in the design, the aerodynamic performance will deteriorate sometimes quite dramatically. For example, when designing transonic shock-free shapes with a surface flow design method, the resulting configuration could have a mildly concave part of its surface locally covered by the supersonic flow indicating the existence of a "hanging shock" or a "loose-foot" shock [139] even at the design conditions. At off-design, the hanging shock attaches itself to the aerodynamic surface causing a boundary layer separation. Consequently, it is more appropriate to design shapes that have a weak family of shocks [140] since such designs have been found not to increase the shock wave strengths appreciably at off-design conditions.

In this chapter, we will focus on briefly explaining only these aerodynamic shape inverse design concepts that are applicable to the design of three-dimensional (3-D) high speed configurations. An attempt will be made to focus on the techniques that have been found to be cost effective, reliable, easy to comprehend and implement, transportable to different computers, and accurate.

10.2 Surface Flow Data Specification

Once the global aerodynamic parameters (inlet and exit pressures, temperatures, and flow angles) have been specified, the next objective is to determine the best way to distribute aerodynamic quantities on the yet unknown configuration. Since the inverse shape design is based on the specified ("desired" or "target") surface pressure distribution, the common dilemma is the choice of the "best" surface target pressure. Specifically, it would be desirable to determine the best pressure distribution on the surface of the yet unknown configuration so that the aerodynamic efficiency is maximized by minimizing all possible contributions to the entropy generation in the entire flow-field. From the specified surface distribution of flow-field parameters it is possible to discern only certain aspects of the boundary layer. It is well known that the separated boundary layer significantly increases flow-field vorticity and, consequently, the viscous dissipation function, entropy generation and aerodynamic drag. To minimize these effects the desired surface pressure distribution can first be checked for possible flow separation before it is further used in the aerodynamic shape inverse design. A very fast method for detecting flow separation has been recently proposed [141]. It is based on the fact that the rate of change of flow kinetic energy reaches its minimum at the separation point. The kinetic energy can be calculated from the surface pressure distribution by assuming that pressure does not change across a boundary layer. The flow separation detection code is short and very simple since it involves algebraic and analytic expressions only. Thus, the surface pressure distribution, either specified by the designer or obtained while using an optimization process, can be quickly checked for possible flow separations (Figure 69) before it is actually enforced.

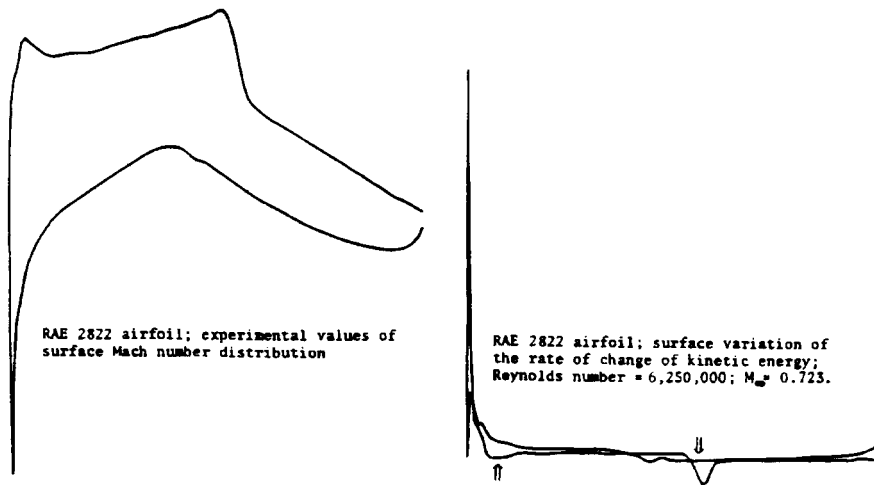


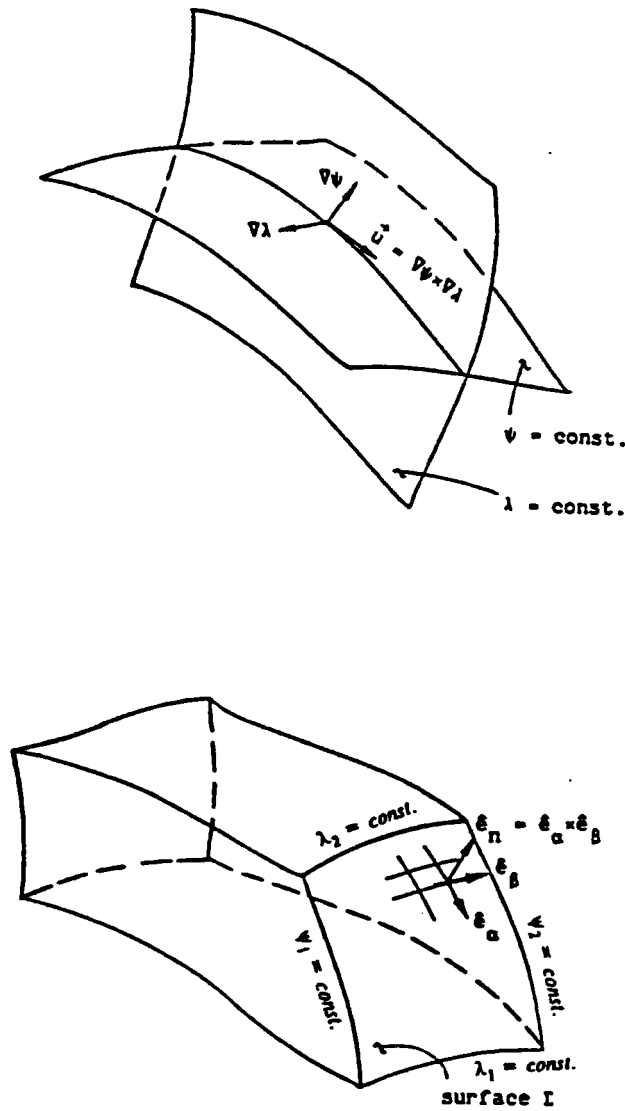
Figure 69 Detection of flow separation locations from the specified or measured surface pressure distribution [141]. Arrows point at experimentally found locations of the flow separation points. Our code predicts separation to occur at the minimum of the curve representing the local rate of change of surface flow kinetic energy.

This a priori checking can be automatically followed by the minor modifications of the local surface pressure distribution with the objective of moving the flow separation points further downstream. The updated surface pressure distribution can be re-checked for the locations of the flow separation points. This checking/modification procedure can be repeated until the separation points cannot move downstream any more. This procedure is extremely fast since it is accomplished without a single call to the flow-field analysis code.

10.3 Stream-Function-Coordinate (SFC) Concept

One of the fastest known inverse design techniques is based on a stream function formulation [142]. The inviscid, compressible, steady flow around a given 3-D configuration can be predicted by solving for two stream functions, $\Psi(x,y,z)$ and $\Lambda(x,y,z)$ [142]-[146] (Figure 70). For the purpose of inverse shape, design this formulation can be inverted. That is, two quasi-linear second order coupled partial differential equations of the mixed elliptic-hyperbolic type can be derived. These two equations treat the x -coordinate and the values of $\Psi(x,y,z)$ and $\Lambda(x,y,z)$ or their derivatives as known quantities on the, as yet, unknown solid surface of the 3-D configuration, while treating values of $y = y(x, \Psi, \Lambda)$ and $z = z(x, \Psi, \Lambda)$ for every 3-D streamline as the unknowns. Thus, the result of a numerical integration of this inverted system are the y -and- z coordinates of the 3-

D streamlines. Those streamlines that correspond to the specified surface values of $\Psi(x,y,z)$ and $\Lambda(x,y,z)$ are recognized as the desired 3-D aerodynamic configuration.



$$\dot{m} = \int_I \rho \vec{q} \cdot d\vec{A} = \int_I \nabla \Psi \times \nabla \Lambda \cdot d\vec{A} = \int_I |J| d\alpha d\beta = \int_I d\Psi d\Lambda = |(\Psi_2 - \Psi_1)(\Lambda_2 - \Lambda_1)|$$

Figure 70 Geometrical interpretation of mass flow rate through a stream tube bounded by two pairs of stream surfaces [144].

A computer code that implements this technique converges very fast because it implicitly satisfies mass at every iteration step thus avoiding the need for integrating the mass conservation equation. Moreover, such a code can be executed in an analysis mode when Dirichlet boundary conditions for $\Psi(x,y,z)$ and $\Lambda(x,y,z)$ are specified at every surface point, or in its inverse design mode when Neumann boundary conditions for $\Psi(x,y,z)$ and $\Lambda(x,y,z)$ are specified at every yet unknown surface point. Despite its remarkable speed of execution, robustness and the fact that the entire field of 3-D streamlines is obtained as a by-product of the computation, the SFC concept has its serious disadvantages. This inverse design method requires development of an entirely new code for the solution of the two Stream-Function-as-a-Coordinate (SFC) equations. The method is not applicable to viscous flow models, it suffers from the difficulties of the geometric multivaluedness of the stream functions, and is analytically singular at all of the points where the Jacobian of transformation $(\Psi, \Lambda, 0)/(x, y, z)$ become zero [142]. This occurs at every point where the x-component of the local velocity vector is zero which can happen at a number of points whose locations we do not know in advance. Consequently, the SFC method is recommended only for the inverse design of smooth 3-D configurations where preferably there are no stagnation points (3-D duct [146]) or where we are willing to neglect the designed shape in the vicinity of the leading and trailing edges.

10.4 Elastic Surface Motion Concept

There are many reliable and relatively fast 3-D flow-field analysis codes in existence. It would be highly economical to utilize these analysis codes in an inverse shape design process. This would require development of a short and simple design code that can utilize any of the existing flow-field analysis codes as a large, exchangeable subroutine. One very simple method for developing such a design code is based on utilizing an elastic membrane as a mathematical model. This concept treating those parts of the surface of the aerodynamic flight vehicle that are desired to have a specified surface pressure distribution as a membrane loaded with unsteady point-forces, ΔC_p , that are proportional to the local difference between the computed and the specified coefficient of surface pressure. Under such an unsteady load the membrane will iteratively deform until it assumes a steady position that experiences zero forcing function at each of the membrane points. Since this is a general non-physical concept for modeling the unsteady damped motion of the 3-D aerodynamic surface, any analytical expression governing damped motion of a continuous surface will suffice. Garabedian and McFadden [147] suggested a simple second-order linear partial differential equation as such a model where ΔC_p is proportional to the local surface slopes and curvatures.

$$\Delta n + \beta_1 \cdot \frac{\partial}{\partial x} \Delta n + \beta_2 \cdot \frac{\partial}{\partial y} \Delta n + \beta_3 \cdot \frac{\partial^2}{\partial x^2} \Delta n + \beta_4 \cdot \frac{\partial^2}{\partial y^2} \Delta n = \beta_5 \cdot \Delta C_p \quad (72)$$

Here, the unknowns are the local normal surface displacements, Δn . If the surface grid point in question is on the upper surface, $\beta_5 = 1.0$, if on the lower surface, $\beta_5 = -1.0$. The remain-

ing coefficients β_1 through β_4 are user-specified quantities that can accelerate the approach to a steady state. This partial differential equation can be discretized using finite difference representations for the partial derivatives. This leads to one penta-diagonal system or a sequence of two three-diagonal systems of algebraic equations that can readily be solved for the unknown normal surface modifications, Δn_i . This simple technique was successfully used to design isolated 3-D transonic supercritical wings [148] with a 3-D full potential code (Figure 71) as the flow analysis module. A simplified version of this concept (with Δz instead of Δn) was used to design engine nacelles and wing-body configurations [149]. The initial guess for the shape of a 3-D body does not have to be close to the final configuration for this method to work. Although requiring only 30-100 calls to a flow-field analysis code when using a panel code or a full potential equation code, the stiffness of the iterative matrix in the present formulation of the elastic membrane concept increases rapidly with the non-linearity of the flow-field analysis code used. This means that when using Euler or Navier-Stokes flow-field analysis code we will need between two and three orders of magnitude more calls to the flow-field analysis code than when using a simple linear panel code. For example, a two-dimensional airfoil shape inverse design with this method utilizing a Navier-Stokes flow analysis code may require over ten thousand calls to the Navier-Stokes code [150].

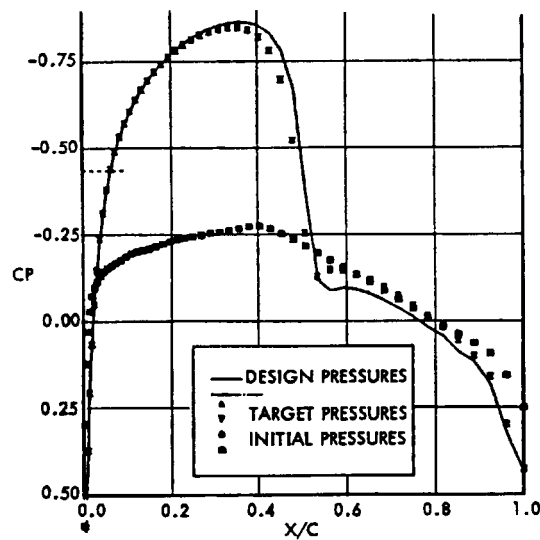


Figure 71 Comparison of baseline, target, and design surface pressures for a transonic purposely shocked airfoil using elastic surface motion concept [148].

10.5 Indirect Surface Transpiration Concept

This is one of the most common and oldest methods for aerodynamic shape inverse design. Like any iterative technique, it requires an initial guess for the aerodynamic shape. Then, using an inviscid flow-field analysis code with desired surface tangential velocity components enforced will result in non-zero values of the surface normal velocity component. The objective is then to find a configuration that has zero velocity components normal to the final body surface. The computed normal velocity components, v_n , are therefore used to modify the shape of the initially guessed configuration using a surface transpiration analogy. The new surface shape is predicted by treating the old surface as porous, hence fictitiously injecting the mass (ρv_n) normal to the original surface so that the new surface becomes an updated stream surface. The local surface displacements, Δn , can be obtained from mass conservation equations for the quasi two-dimensional sections (stream tubes) of the flow-field bounded by the two consecutive cross sections of the body surface, the original surface shape, and the updated surface shape displaced locally by Δn [149]–[153]. Starting from a stagnation line where $\Delta n_{i-1,j} = 0.0$ and $\Delta n_{i-1,j+1} = 0.0$, separate updating of the pressure surface and the suction surface can be readily performed by solving for $\Delta n_{i,j}$ and $\Delta n_{i,j+1}$ from a bi-diagonal system. With the classical transpiration concept, normal surface velocities can be computed using any potential flow solver including a highly economical surface panel flow [152], [153] analysis code Figure 72. The indirect surface transpiration method works quite satisfactory in conjunction with Euler and even Navier-Stokes equations barring any shock waves or flow separation. A drawback of this approach is that during the repetitive surface updating using this method, the updated surfaces develop a progressively increasing degree of oscillation. This can be eliminated by periodically smoothing the updated surfaces with a least-squares surface fitting algorithm.

10.6 Direct Surface Transpiration Concept

This is an equally simple method of utilizing the transpiration concept which is especially applicable to viscous flow solvers, but the manner of treating boundary conditions on surfaces is different. For example, in the flow analysis with Navier-Stokes equations, the no-slip condition is imposed at the solid surface by enforcing zero values of all three components of the contravariant velocity vector defined as $U = D\xi/Dt$, $V = D\eta/Dt$, $W = D\zeta/Dt$. The ξ, η, ζ curvilinear, non-orthogonal coordinate system follows the structured computational grid lines. For example, let η -grid lines (thus V contravariant velocity component) emanate from the surface of the 3-D object under design consideration. In this inverse shape design concept, the surface pressure distribution, which is obtained from the specified pressure coefficient distribution, is enforced iteratively together with $U = W = 0$. This can be done readily in any existing Navier-Stokes flow analysis code [154]. It will result in the contravariant velocity vector component, V , becoming non-zero at the surface. Hence, the surface will have to move with iterations or time steps until the convergence is reached, that is, until $V = 0$ is satisfied on the final surface configuration. This will require that the computational grid be regenerated with each update of the surface. Thus, the aerodynamic parameters need to be transferred between old and new grid points by an accurate 3-D interpolation.

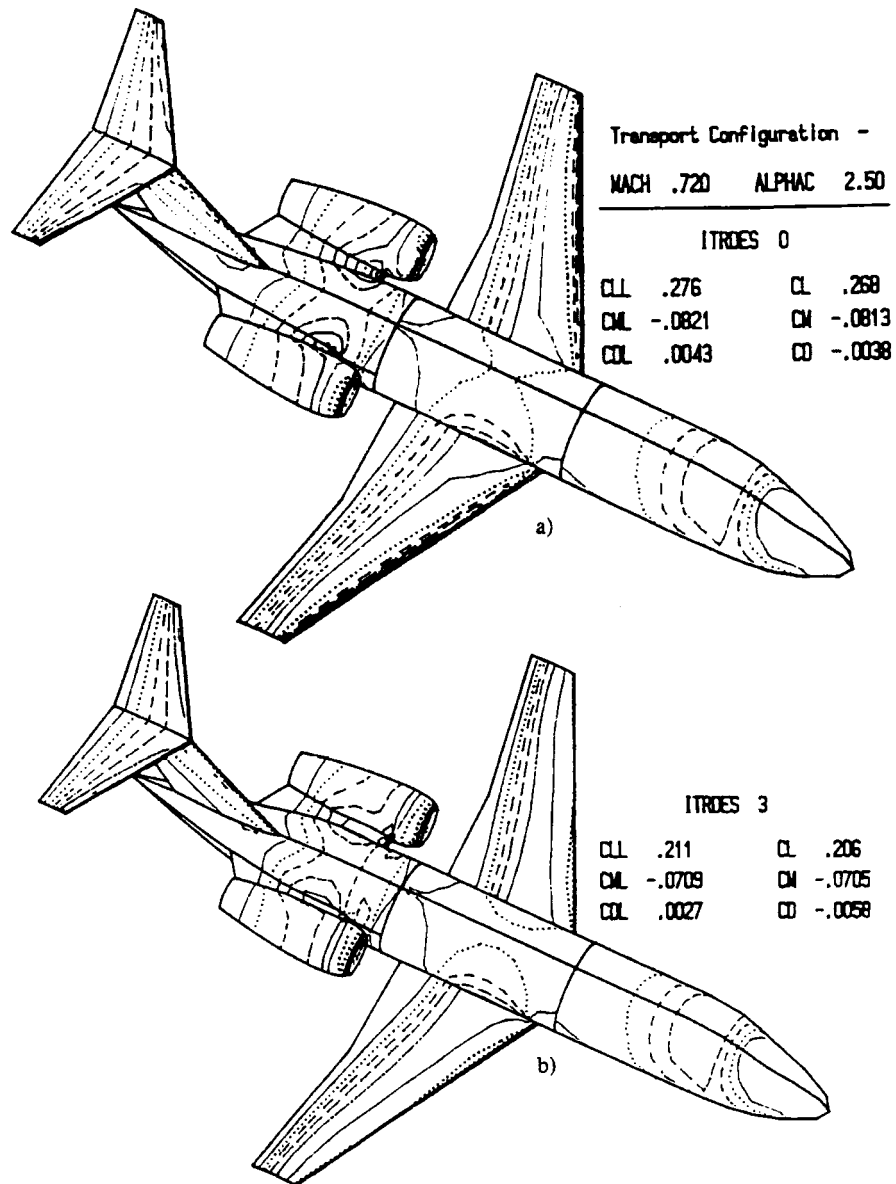


Figure 72 Inverse design of an entire business jet configuration using a high order surface panel code and indirect surface transpiration concept can be performed on a personal computer in less than one hour: a) before, and b) after three optimization cycles enforcing the desired surface pressures. Notice improvements in aerodynamic coefficients [153].

This design method will provide for a time-accurate motion of the solid boundaries if the code is executed in a time-accurate mode. For example, it is possible with this method to design a flexible 3-D shape that maintains an essentially steady surface pressure distribution in an otherwise unsteady flow. Specifically, it is possible to determine the correct instantaneous values of local swelling and contraction of a "smart" material coating on the surface thus creating the smart or continuously-adaptable aerodynamic shape design.

10.7 Characteristic Boundary Condition Concept

The correct number and types of boundary conditions for a system of partial differential equations can be determined by analyzing eigenvalues and the corresponding eigenvectors of the system in each coordinate direction separately. This approach to boundary condition treatment is called characteristic boundary conditions [155]-[157] since it suggests one-dimensional application of certain Riemann invariants at the boundaries that can be either solid or open boundaries. For example, in a 3-D duct flow, if the flow is locally subsonic at the exit, we will be able to compute all flow variables at the exit based on the information from the interior points except for one variable that we will have to specify at the exit. If the same characteristic boundary condition procedure is applied in the direction normal to the solid wall and if the desired pressure distribution is specified on the wall, this method of iteratively enforcing the boundary conditions at the wall will result in non-zero normal velocities at the wall which can be used to update the wall shape. The general concept follows.

The Euler equations for 3-D compressible unsteady flows expressed in non-conservative form and cast in a boundary-conforming, non-orthogonal, curvilinear (ξ, η, ζ) coordinate system can be transformed into

$$\frac{\partial Q}{\partial \tau} + B \cdot \frac{\partial Q}{\partial \eta} + D = 0 \quad (73)$$

where $Q = (\rho \ p \ u \ v \ w)$ is the transposed vector of the non-conservative primitive variables. Eigenvalues of B are

$$V - a \cdot \sqrt{(\eta_x^2 + \eta_y^2 + \eta_z^2)} \ V, V, V, V + a \cdot \sqrt{(\eta_x^2 + \eta_y^2 + \eta_z^2)} \quad (74)$$

where $V = \eta_t + \eta_x \cdot u + \eta_y \cdot v + \eta_z \cdot w$ and the local speed of sound is defined as $a = (\gamma p / \rho)^{1/2}$. If the η -grid lines are emanating from the 3-D aerodynamic configuration, we can have several situations. If $0 < V < a \sqrt{(\eta_x^2 + \eta_y^2 + \eta_z^2)}$, one eigenvalue is negative requiring a pressure boundary condition to be specified at that surface point.

Similarly, if $\sqrt{\eta_x^2 + \eta_y^2 + \eta_z^2} < V < 0$, four eigenvalues will be negative requiring pressure, velocity ratio $u/(u^2 + v^2 + w^2)^{1/2}$, total pressure and total temperature to be specified at that surface point. This method has been shown to converge quickly for transonic two-dimensional airfoil shape design when using compressible flow Euler equations [156],[157]. The method might be applicable to the inverse design of arbitrary 3-D configurations [155] although no such attempts have been reported yet. If the Euler code is executed in a time-accurate mode, the specified unsteady solid wall characteristic boundary conditions will provide for a time-accurate motion of the solid boundaries which is highly attractive for the design of "smart" aerodynamic configurations. This concept is not directly applicable to viscous flow codes since velocity components at the solid wall are zero.

10.8 Integro-Differential Equation Concept

An attractive property of integral equations is that the influence of the boundary conditions is transmitted throughout the flow-field instantaneously in the case of a linear flow problem. Even for non-linear flow problems the influence of the boundary conditions is transmitted throughout the flow-field extremely quickly as compared to the partial differential equation models where the finite difference or finite element discretization allows the influence of the boundary conditions to be transmitted at most one grid cell per during each iteration. A very fast and versatile 3-D aerodynamic shape inverse design algorithm was developed and is widely utilized in several countries [158]-[161]. It can accept any available 3-D flow-field analysis code as a large subroutine to analyze the flow around the intermediate 3-D configurations. The configurations are updated using a fast integro-differential formulation where a velocity potential perturbation $\phi(x,y,z)$ around an initial 3-D configuration $z_{+/-}(x,y)$ can be obtained from, for example, a Navier-Stokes code [161]. Here, the subscripts +/- refer to the upper and lower surfaces of the flight vehicle. Transonic 3-D small perturbation equation is

$$\frac{\partial^2}{\partial x^2}\Delta\phi + \frac{\partial^2}{\partial y^2}\Delta\phi + \frac{\partial^2}{\partial z^2}\Delta\phi = \frac{1}{V_\infty} \cdot \frac{\partial}{\partial x} \cdot \left(\frac{1}{2} \cdot \left(\frac{\partial\phi}{\partial x} + \frac{\partial}{\partial x}\Delta\phi \right)^2 - \frac{1}{2} \cdot \left(\frac{\partial\phi}{\partial x} \right)^2 \right) = \frac{1}{V_\infty} \cdot \frac{\partial\Gamma}{\partial x} \quad (75)$$

Here, differentially small potential perturbation is $\Delta\phi(x,y,z)$ and x,y,z coordinates have been scaled via Prandtl-Glauert transformation, V_∞ is the free stream magnitude, while

$$\frac{\partial\phi}{\partial x}(x, y, +/0) = -\frac{V_\infty}{2} \cdot C_p(x, y) \quad (76)$$

$$\frac{\partial}{\partial x} \Delta \phi(x, y, +/-0) = -\frac{(1 + \kappa) \cdot M \cdot a_{\infty}^2}{2 \cdot (1 - M \cdot a_{\infty}^2)} \cdot (C_{p+/-}^{spec} - C_{p+/-}^{calc}) \quad (77)$$

Here, M is the local Mach number and a_{∞} is the free stream speed of sound. The flow tangency condition is then

$$\frac{\partial}{\partial z} \Delta \phi(x, y, +/-0) = V_{\infty} \cdot \frac{\partial}{\partial x} \Delta z_{+/-}(x, y) \quad (78)$$

Since $\frac{\partial}{\partial z} \Delta \phi(x, y, +/-0)$ can be obtained from equation (75), the 3-D geometry is readily updated from

$$\Delta z_{+/-}(x, y) = \frac{1}{2} \cdot \int \frac{\partial}{\partial x} [\Delta z_{+}(x, y) + \Delta z_{-}(x, y)] dx \pm \frac{1}{2} \cdot \int \frac{\partial}{\partial x} [\Delta z_{+}(x, y) - \Delta z_{-}(x, y)] dx \quad (79)$$

Since equation (75) is linear, it can be reformulated using Green's theorem as an integro-differential equation. The Γ term on the right hand side of equation (75) would require volume integration which can be avoided if Γ is prescribed as smoothly decreasing away from the 3-D flight vehicle surface where it is known. Then, the problem can be very efficiently solved using the 3-D boundary element method. This inverse shape design concept has been successfully applied to a variety of planar wings [158]-[160] and wing-body configurations including the H-II Orbiting Plane with winglets [161], [162] where 3-D flow-field analysis codes were of the full potential, Euler and Navier-Stokes type. The method typically requires 10-30 flow analysis runs with an arbitrary flow solver and as many solutions of the linearized integro-differential equation.

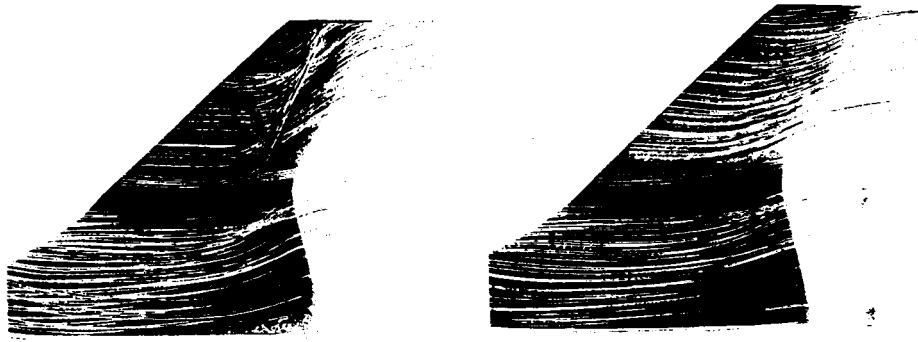


Figure 73 Winglets on a Japanese space plane were successfully redesigned using the integro-differential equation approach and a Navier-Stokes flow-field analysis code [161].

10.9 Conclusions

Several prominent and proven methods that are applicable to inverse design of 3-D aerodynamic shapes have been briefly surveyed. The design computer codes based on these methods can be readily developed by modifying solid boundary condition subroutines in most of the existing flow-field analysis codes. Thus, all of the design methods surveyed are computationally economical since they require typically only a few dozen calls to the 3-D flow-field analysis code. Although the inverse shape design methods generate only point-designs, it was pointed out that at least two of the methods are conceptually capable of inverse shape design for unsteady flow conditions.

10.10 References

- [130] **Dulikravich, G. S. (editor)**
Proceedings of the 1st International Conference on Inverse Design Concepts in Engineering Sciences (ICIDES-I), University of Texas, Dept. of Aero. Eng. & Eng. Mech., Austin, TX, October 17-18, 1984.
- [131] **Dulikravich, G. S. (editor)**
Proceedings of the 2nd International Conference on Inverse Design Concepts and Optimization in Engineering Sciences (ICIDES-II), Penn State Univ., October 26-28, 1987.
- [132] **Dulikravich, G. S. (editor)**
Proceedings of the 3rd International Conference on Inverse Design Concepts and Optimization in Engineering Sciences (ICIDES-III), Washington, D.C., October 23-25, 1991.
- [133] **Slooff, J. W.**
Computational Methods for Subsonic and Transonic Aerodynamic Design, in: Proc. of 1st Int. Conf. on Inverse Design Concepts and Optimiz. in Eng. Sci. (ICIDES-I), ed. G.S. Dulikravich, Dept. of Aero. Eng. and Eng. Mech., Univ. of Texas, Austin, TX, Oct. 17-18, 1984, pp. 1-68.
- [134] **Slooff, J. W. (editor)**
Proceedings of the AGARD Specialist's Meeting on Computational Methods for Aerodynamic Design (Inverse) and Optimization, AGARD CP-463, Loen, Norway, May 2-23, 1989.
- [135] **van dem Braembussche, R. (editor)**
Proceedings of a Special Course on Inverse Methods for Airfoil Design for Aeronautical and Turbomachinery Applications, AGARD Report No. 780, Rhode-St.-Genese, Belgium, May 1990.
- [136] **Dulikravich, G. S.**

-
- Aerodynamic Shape Design and Optimization: Status and Trends, AIAA Journal of Aircraft, Vol. 29, No. 5, Nov./Dec. 1992, pp. 1020-1026.
- [137] **Labrujere, T. E., Slooff, J. W.**
Computational Methods for the Aerodynamic Design of Aircraft Components, in: Annual Review of Fluid Mechanics, Vol. 25, 1993, pp. 183-214.
- [138] **Dulikravich, G. S.**
Shape Inverse Design and Optimization for Three-Dimensional Aerodynamics, AIAA invited paper 95-0695, AIAA Aerospace Sciences Meeting, Reno, NV, January 9-12, 1995; also to appear in AIAA Journal of Aircraft.
- [139] **Volpe, G.**
Inverse Design of Airfoil Contours: Constraints, Numerical Methods and Applications, AGARD CP-463, Loen, Norway, May 22-23, 1989, Ch. 4.
- [140] **Zhu, Z., Sobieczky, H.**
An Engineering Approach for Nearly Shock-Free Wing Design, Proc. of the Internat. Conf. on Fluid Mech., Beijing, China, July 1987.
- [141] **Dulikravich, G. S.**
A Criteria for Surface Pressure Specification in Aerodynamic Shape Design, AIAA paper 90-0124, Reno, NV, January 8-11, 1990.
- [142] **Huang, C.-Y., Dulikravich, G. S.**
Stream Function and Stream-Function-Coordinate (SFC) Formulation for Inviscid Flow Field Calculations, Computer Methods in Applied Mechanics and Engineering, Vol. 59, November 1986, pp. 155-177.
- [143] **Sherif, A., Hafez, M.**
Computation of Three Dimensional Transonic Flows Using Two Stream Functions, Internat. Journal for Numerical Methods in Fluids, Vol. 8, 1988, pp. 17-29.
- [144] **Stanitz, J. D.**
A Review of Certain Inverse Methods for the Design of Ducts With 2- or 3-Dimensional Potential Flow, Appl. Mech. Rev., Vol. 41, No. 6, June 1988, pp. 217-238.
- [145] **Zhang, S.**
Streamwise Transonic Computations, Ph. D. dissertation, Department of Mathematics & Statistics, University of Windsor, Windsor, Ontario, Canada, December 1993.
- [146] **Chen, N.-X., Zhang, F.-X., Dong, M.**
Stream-Function-Coordinate (SFC) Method for Solving 2D and 3D Aerodynamics Inverse Problems of Turbomachinery Flows, Inverse Problems in Engineering, Vol. 1, No. 3, March 1995.
- [147] **Garabedian, P., McFadden, G.**
Design of Supercritical Swept Wings, AIAA J., Vol. 20, No. 3, March 1982, pp. 289-291.
- [148] **Malone, J., Vadyak, J., Sankar, L. N.**

- A Technique for the Inverse Aerodynamic Design of Nacelles and Wing Configurations, *AIAA Journal of Aircraft*, Vol. 24, No. 1, January 1987, pp. 8-9.
- [149] **Hazarika, N.**
An Efficient Inverse Method for the Design of Blended Wing-Body Configurations, Ph.D. Thesis, Aerospace Eng. Dept., Georgia Institute of Technology, June 1988.
- [150] **Malone, J. B., Narramore, J. C., Sankar, L. N.**
An Efficient Airfoil Design Method Using the Navier-Stokes Equations, AGARD Specialists' Meeting on Computational Methods for Aerodynamic Design (Inverse) and Optimization, AGARD-CP-463, Loen, Norway, May 22-23, 1989.
- [151] **Bristow, D. R., Hawk, J. D.**
Subsonic 3-D Surface Panel Method for Rapid Analysis of Multiple Geometry Perturbations, AIAA paper 82-0993, St. Louis, MO, June 7-11, 1982.
- [152] **Fornasier, L.**
An Iterative Procedure for the Design of Pressure-Specified Three-Dimensional Configurations at Subsonic and Supersonic Speeds by Means of a Higher-Order Panel Method, AGARD-CP-463, Loen, Norway, May 22-23, 1989, Ch. 6.
- [153] **Kubrynski, K.**
Design of 3-Dimensional Complex Airplane Configurations with Specified Pressure Distribution via Optimization, Proc. of 3rd Int. Conf. on Inverse Design Concepts and Optimiz. in Eng. Sci. (ICIDES-III), editor: Dulikravich, G. S., Washington, D. C., October 23-25, 1991.
- [154] **Wang, Z., Dulikravich, G. S.**
Inverse Shape Design of Turbomachinery Airfoils Using Navier-Stokes Equations, AIAA paper 95-0304, Reno, NV, January 9-12, 1995.
- [155] **Zannetti, L., di Torino, P., Ayele, T. T.**
Time Dependent Computation of the Euler Equations for Designing Fully 3D Turbomachinery Blade Rows, Including the Case of Transonic Shock Free Design, AIAA paper 87-0007, Reno, NV, January 12-15, 1987.
- [156] **Leonard, O.**
Subsonic and Transonic Cascade Design, AGARD-CP-463, Loen, Norway, May 22-23, 1989, Ch. 7.
- [157] **Leonard, O., van den Braembussche, R. A.**
Design Method for Subsonic and Transonic Cascade With Prescribed Mach Number Distribution, *ASME J. of Turbomachinery*, Vol. 114, July 1992, pp. 553-560.
- [158] **Takanashi, S.**
Iterative Three-Dimensional Transonic Wing Design Using Integral Equations, *J. of Aircraft*, Vol. 22, No. 8, August 1985, pp. 655-660.
- [159] **Hua, J., Zhang, Z.-Y.**
Transonic Wing Design for Transport Aircraft, ICAS 90-3.7.4, Stockholm, Sweden, September 1990, pp. 1316-1322.

-
- [160] **Bartelheimer, W.**
An Improved Integral Equation Method for the Design of Transonic Airfoils and Wings, AIAA-95-1688-CP, San Diego, CA, June 1995.
- [161] **Kaiden, T., Ogino, J., Takanashi, S.**
Non-Planar Wing Design by Navier-Stokes Inverse Computation, AIAA paper 92-0285, Reno, NV, January 6-9, 1992.
- [162] **Fujii, K., Takanashi, S.**
Aerodynamic Design Methods for Aircraft and their Notable Applications - Survey of the Activity in Japan, Proc. of 3rd Int. Conf. on Inverse Design Concepts and Optimiz. in Eng. Sci. (ICIDES-III), editor: Dulikravich, G. S., Washington, D.C., October 23-25, 1991.

GEORGE S. DULIKRAVICH

INTERNATIONAL CENTRE FOR MECHANICAL SCIENCES

COURSES AND LECTURES - No. 366



NEW DESIGN CONCEPTS
FOR HIGH SPEED AIR TRANSPORT

EDITED BY

H. SOBIECZKY
DLR GERMAN AEROSPACE RESEARCH ESTABLISHMENT



SpringerWienNewYork

Chapter 11

Aerodynamic Shape Optimization Methods

George S. Dulikravich

11.1 Introduction

Although fast and accurate in creating aerodynamic shapes compatible with the specified surface pressure distribution, the inverse shape design methods create configurations that are not optimal even at the design operating conditions [163], [164]. At off design conditions, these configurations often perform quite poorly except when the specified surface pressure distribution, if available at all, would be provided by an extremely accomplished aerodynamicist. When using inverse shape design methods, it is physically unrealistic to generate a 3-D aerodynamic configuration that simultaneously satisfies the specified surface distribution of flow variables, manufacturing constraints (smooth variation of a lifting surface sweep and twist angles, smooth variation of its taper, etc.) and achieves the best global aerodynamic performance (overall total pressure loss minimized, lift/drag maximized, etc.). The designer should use an adequate global optimization algorithm that can utilize any available flow-field analysis code without changes and efficiently optimize the overall aerodynamic characteristics of the 3-D flight vehicle subject to the finite set of desired constraints. The constraints could be purely geometrical or they can be of the overall aerodynamic nature (minimize overall drag for the given values of flight speed, angle of attack and overall lift force, etc.). These objectives can only be met by performing an aerodynamic shape constrained optimization instead of an inverse shape design.

The size and shape of the mathematical space that contains all the design variables (for example, coordinates of all surface points) is very large and complex in a typical 3-D case. To find a global minimum of such a space requires a sophisticated numerical optimization algorithm that avoids local minima, honors the specified constraints and stays within the feasible design domain. The design variable space in a typical aerodynamic shape optimization has a

number of local minima. These minima are very hard to escape from even by switching the objective function formulation [165] or consecutive spline fitting and interpolation of the unidirectional search step parameter [166].

There are several fundamental concepts in creating an optimization algorithm. One family of optimization algorithms is based on reducing the objective or cost function (for example, aerodynamic drag) by evaluating the gradient of the cost function and then updating the design variables in the negative gradient direction [167]. Evolution search or genetic algorithms is another family of optimization algorithms that is based on a semi-random sampling through the design variable space and does not require any gradient evaluations [168], [169]. Since both families of optimization algorithms require flow-field analysis to be performed on every perturbed aerodynamic configuration, the optimization of 3-D aerodynamic shapes is a very computationally intensive task.

In a gradient-search optimization approach the flow analysis code must be called at least once for each design variable during each optimization cycle in order to compute the gradient of the objective function if one-sided finite differencing is used for the gradient evaluation. If a more appropriate central differencing is used for the gradient evaluation, the number of calls to the 3-D flow-field analysis code will immediately double. Despite this, the optimization algorithms are still often misused to minimize the difference between the specified and the computed surface flow data in inverse shape design - a task that is significantly more economical when accomplished with any of the standard inverse shape design algorithms.

The most serious drawback of the brute force application of the gradient search optimization in 3-D aerodynamics is that the computing costs increase nonlinearly with the growing number of design variables thus making these algorithms suitable for smaller optimization problems. On the other hand, the computing cost of using evolution search algorithms increases only moderately with the number of design variables (Figure 74) thus making these algorithms more suitable for large optimization problems. Only these optimization algorithms that require minimum number of calls to the flow-field analysis code will be realistic candidates for the 3-D aerodynamic shape optimization.

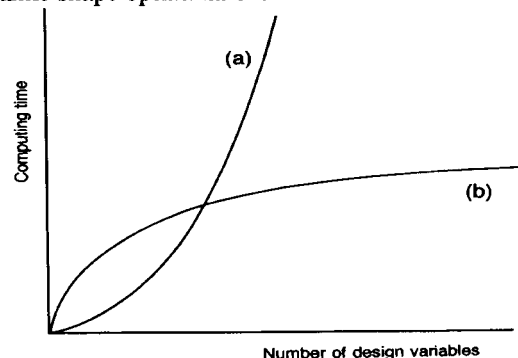


Figure 74 A sketch of the dependency of computational effort as a function of the number of design variables for the case of a classical gradient search (a) and genetic (b) optimization algorithms.

Since the actual 3-D flow-field analysis codes of Euler or Navier-Stokes type are very time consuming, the designer is forced to restrict the design space by working with a relatively small number of the design variables for parameterization (fitting polynomials) of either the 3-D surface geometry [170]-[172] or the 3-D surface pressure field [173], [174]. The optimization code then needs to identify the coefficients in these polynomials. The most plausible choices are cubic splines. Chebyshev and Fourier polynomials [175]-[177] are not advisable because they become excessively oscillatory with the increasing number of terms in the polynomial. Moreover, when perturbing any of the coefficients in such a polynomial, the entire 3-D shape will change. Since it is absolutely necessary to constrain and sometimes disallow motion of particular parts of the 3-D surface, the most promising choices for the 3-D parameterization appear to be different types of b-splines [178], [179], [171], [180], local analytical surface patches [181], and local polynomial basis functions [182]. Only when it is possible to use simple and very fast flow-field analysis codes could we afford an ideal optimization situation where each surface grid point on the 3-D optimized configuration is allowed to move independently.

Single-cycle optimization [183] offers one viable approach at reducing the computing costs. Here, the flow-field analysis code is run on each perturbed aerodynamic configuration for only a small number of iterations (instead to a full convergence) before an optimizer is used to determine the new geometry. An optimal aerodynamic shape is then found by optimally weighing each of the number of feasible configurations that can be obtained using inverse design methods. Hence, this optimization approach guarantees that the final configuration will be realistically shaped and manufacturable, although the range of geometric parameters to be optimized is limited by the geometry of the extreme members of the original family of configurations.

11.2 Optimization Using Sensitivity Derivatives

It is often desirable to have a capability to predict the behavior of the inputs to an arbitrary system by relating the outputs to the inputs via a sensitivity derivative matrix [184]-[188], while treating the system as a black box. The sensitivity derivative matrix can be used for the purpose of controlling the system outputs or to achieve an optimized constrained design that depends on the system outputs. The objective is to generate approximations of the infinite dimensional sensitivities and to transfer these approximate derivatives to the optimizer together with the approximate function evaluations. The control variables are then updated with the sensitivity derivatives which are the gradients of the cost function with respect to the control variables. The general concepts for the sensitivity analysis can be summarized as follows [185].

The system of governing flow-field governing equations after discretization results in a system of non-linear algebraic equations

$$R(Q(D), X(D), D) = 0 \quad (80)$$

where Q is the vector of the solution variables in the flow-field governing system, X is the computational grid and D is the vector of design variables (for example, coordinates of 3-D aerodynamic shape surface points). Hence

$$\frac{dR}{dD} = \frac{\partial R}{\partial Q} \frac{dQ}{dD} + \frac{\partial R}{\partial X} \frac{dX}{dD} + \frac{\partial R}{\partial D} = 0 \quad (81)$$

Similarly, aerodynamic output functions (lift, drag, lift/drag, moment, etc.) are defined as

$$F = F(Q(D), X(D), D) \quad (82)$$

Hence

$$\frac{dF}{dD} = \frac{\partial F}{\partial Q} \frac{dQ}{dD} + \frac{\partial F}{\partial X} \frac{dX}{dD} + \frac{\partial F}{\partial D} \quad (83)$$

System (81) is solved for the sensitivity derivatives of the field variables, dQ/dD , which are then substituted into the system (83) in order to obtain the sensitivity derivatives of the desired aerodynamic outputs, dF/dD . This approach is typically used if the dimension of F is greater than that of D which is seldom the case in a 3-D aerodynamic shape design.

When the number of design variables D is larger than the number of the aerodynamic output functions F , it is more economical to avoid solving for dQ/dD . This can be accomplished by using an adjoint operator approach where a linear system

$$\left(\frac{\partial R}{\partial Q} \right)^T A + \left(\frac{\partial F}{\partial Q} \right)^T = 0 \quad (84)$$

must be solved first. Here, A is a discrete adjoint variable matrix associated with the aerodynamic output functions, F . Substituting equation (84) into equation (83), it follows from equation (81) that the aerodynamic output derivatives of interest can be computed from

$$\frac{dF}{dD} = A^T \cdot \left(\frac{\partial R}{\partial X} \frac{dX}{dD} + \frac{\partial R}{\partial D} \right) + \frac{\partial F}{\partial X} \frac{dX}{dD} + \frac{\partial F}{\partial D} \quad (85)$$

This quasi-analytical approach to computing sensitivity derivatives is more economical and accurate than when evaluating the derivatives using finite differencing. Nevertheless, sensitivity analysis is a very costly process requiring a large number of analysis runs.

In the gradient-search optimization approach the flow analysis code must be called at least once for each design variable in order to compute the gradient of the objective function

during each optimization cycle. Since each call to the analysis code is very expensive, such an approach to design is justified only if a small number of design variables is used. In the case of a 3-D design, this is hardly justifiable even if one uses 3-D surface geometry parametrization which severely constrains 3-D optimal configurations.

One of the most promising recent developments in the aerodynamic shape design optimization is a method that treats the entire system of partial differential equations governing the flow-field as constraints, while treating coordinates of all surface grid points as design variables [189]. This approach eliminates the need for geometry parameterization using shape functions to define changes in the geometry. Since fluid dynamic variables, Q , are treated here as the design variables, this method allows for rapid computation of partial derivatives of the objective function with respect to the design variables. This approach is straightforward to comprehend and efficient to implement in Newton-type direct flow analysis algorithms where solutions of the equations for dQ/dD or A amount to a simple back-substitution. The problem is that the classical Newton iteration algorithm is practically impossible to implement for 3-D aerodynamic analysis codes because of its excessive memory requirements when performing direct LU factorization of the coefficient matrix.

Instead of using an exact Newton algorithm in the flow-analysis code, it is more cost effective to use a quasi-Newton iterative formulation or an incremental iterative strategy [185], [190] given in the form

$$\frac{\partial \mathcal{R}}{\partial Q} \cdot D \left(\frac{dQ}{dD} \right) = \left(\frac{\partial R}{\partial D} \right)^n = \frac{\partial R}{\partial D} \cdot \left(\frac{dQ}{dD} \right)^n + \frac{\partial R}{\partial X} \frac{dX}{dD} + \frac{\partial R}{\partial D} \quad (86)$$

$$\left(\frac{dQ}{dD} \right)^{n+1} = \left(\frac{dQ}{dD} \right)^n + D \left(\frac{dQ}{dD} \right) \quad ; n=1,2,3, \dots \quad (87)$$

where $\frac{\partial \mathcal{R}}{\partial Q}$ could be any fully-converged numerical approximation of the exact Jacobian matrix.

11.3 Constrained Genetic Evolution Optimization

Besides a wide variety of the gradient-based optimization algorithms, truly remarkable results were obtained using an evolution type genetic algorithm or GA [168],[169],[191]. The GA methods simulate the mechanics of natural genetics for artificial systems based on operations which are the counter parts of the natural ones. They rely on the use of a random selection process which is guided by probabilistic decisions. In general, a GA is broken into three major steps: reproduction, crossover, and mutation. An initial population of complete design variable sets are analyzed

according to some cost function. Then, this population is merged using a crossover methodology to create a new population. This process continues until a global minimum is found. Generally, the design variable set that corresponds to the minimum point of the cost function will be representative as having the most "successful" features of previous "generations" of designs in the optimization process.

Although the standard genetic algorithm (GA) is computationally quite expensive since it requires a large number of calls to the flow-field analysis code, the robustness of this algorithm and the ease of its implementation have created a recently renewed interest in applying it for aerodynamic shape design [192]-[201]. Nevertheless, the examples presented in these publications involve aerodynamic shape optimization with a small number of design variables that form a relatively compact function space. Solutions of such optimization problems would be considerably more efficient when using more common gradient search algorithms. Moreover, none of the examples in these publications attempt to treat equality-type constrained optimization which represents the most difficult problem for a typical GA algorithm. The classical GA can handle constraints on the design variables, but it is not inherently capable of handling constraint functions [198]. Most of the recent publications involving the GA and aerodynamic shape optimization have involved problems posed in such a way as to eliminate constraint functions, or to penalize the cost function when a constraint is violated. These treatments of constraints reduce the chance of arriving at the global minimum.

Probably the most attractive feature of the GA is its remarkable robustness since it is not a gradient-based search method. The GA is exceptional at avoiding local minima, because it tests possible designs over a large design variable space. Hence, the GA is especially suitable for handling a large number of design variables that belong to widely different engineering disciplines, thus making it particularly suitable for true complex multidisciplinary optimization problems. The GA is especially suitable for the types of problems where the sensitivity derivatives might be discontinuous [173] which is sometimes the problem in 3-D aerodynamic optimization. The number of cost function evaluations per design iteration of a GA does not depend on the number of design variables. Rather, it depends on the size of the initial population.

There are some subtleties associated with the GA that, if treated properly, greatly increases the effectiveness and usefulness of the method. One such subtlety involves the crossover procedure. When two members of the population are chosen for a crossover, their design variable sets are generally encoded into strings called "chromosomes". After a crossover has taken place, the "child" variable set is recovered by decoding its newly created chromosome string. When the design variables are floating-point numbers, as is usually the case, this coding and decoding process can introduce a loss of precision arising from numerical truncation on a finite precision computer. This is particularly true when the chromosome string format is defined by a "bit-string" (a base 2 number), and the operating language is FORTRAN. A crossover method that preserves full precision in the design variables can be developed [202] by changing the format of the coded chromosome string and by the implementation of C and C++ as the operating language instead of FORTRAN. These languages allow bitwise shifting of floating point numerics and require no coding or decoding processes whatsoever. The C++ language further allows object-oriented programming techniques that provide a platform for the

truest genetic interaction of design variable sets. Equality and non-equality constraints can be incorporated by using Rosen's projection method [202],[180]. This improved GA could be used as a black box optimizer in aerodynamics, elasticity, heat transfer, etc, or in a multidisciplinary design optimization.

11.4 Hybrid GA/Gradient Search Constrained Optimization

The GA has proven itself to be an effective and robust optimization tool for large variable-set problems if the cost function evaluations are very cheap to perform. Nevertheless, in the field of 3-D aerodynamic shape optimization we are faced with the more difficult situation where each cost function evaluation is extremely costly and the number of the design variables is relatively large. Standard GA will require large memory if large number of design variables are used. The number of cost function evaluations per design iteration of a GA increases only mildly with the number of design variables, while increasing rapidly with the increased size of the initial population. Also, the classical GA can handle constraints on the design variables, but it is not inherently capable of handling constraint functions [198]. Thus, the brute force application of the standard GA to 3-D aerodynamic shape design optimization is economically unjustifiable.

Consequently, a hybrid optimization made of a combination of the GA and a gradient search optimization or the GA and an inverse design method has been shown to be an advisable way to proceed [180], [201]-[203]. Preliminary results obtained with different versions of a hybrid optimizer that uses a GA for the overall logic, a quasi-Newtonian gradient-search algorithm or a feasible directions method [167] to ensure monotonic cost function reduction, and a Nelder-Mead sequential simplex algorithm or a steepest descent methodology of the design variables into feasible regions from infeasible ones has proven to be effective at avoiding local minima. Since the classical GA does not ensure monotonic decrease in the cost function, the hybrid optimizer could store information gathered by the genetic searching and use it to determine the sensitivity derivatives of the cost function and all constraint functions [203], [180]. When enough information has been gathered and the sensitivity derivatives are known, the optimizer switches to the feasible directions method (with quadratic subproblem) for quickly proceeding to further improve on the best design.

One possible scenario for a hybrid genetic algorithm can be summarized as follows:

- Let the set of population members define a simplex like that used in the Nelder-Mead method.
- If the fitness evaluations for all of the population members does not yield a better solution, then define a search direction as described by the Nelder-Mead method.
- If there are active inequality constraints, compute their gradients and determine a new search direction by solving the quadratic subproblem.

- If there are active equality constraints, project this search direction onto the subspace tangent to the constraints.
- Perform line search.

Our recent work [203], [180] indicates that the hybrid GA can yield answers (Figure 75 and Figure 76) not obtainable by standard gradient methods at comparable convergence rates (Figure 77). This hybrid optimizer can also handle non-linear constraint functions, although the main computational cost will be incurred by enforcing the constraints since this task will involve evaluating the gradients of the constraint functions.

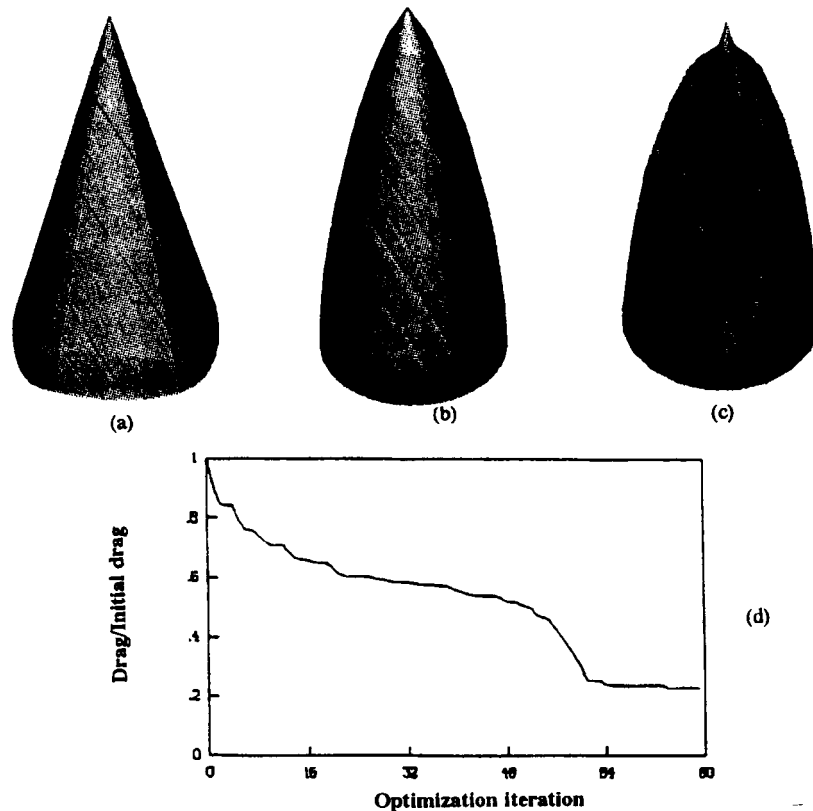


Figure 75 Our hybrid gradient search/genetic constrained optimizer keeps on improving the design even beyond the known analytic optimal solution (von Karman and Sears-Haack smooth ogives). It transformed an initially conical configuration (a) with 100% drag into a smooth ogive shape (b) with 56% drag and finally into a star-shaped spiked missile (c) with only 22% of the initial drag. Convergence history (d) shows monotonic convergence although 480 variables were optimized with a constrained volume and length of the missile [203], [180].

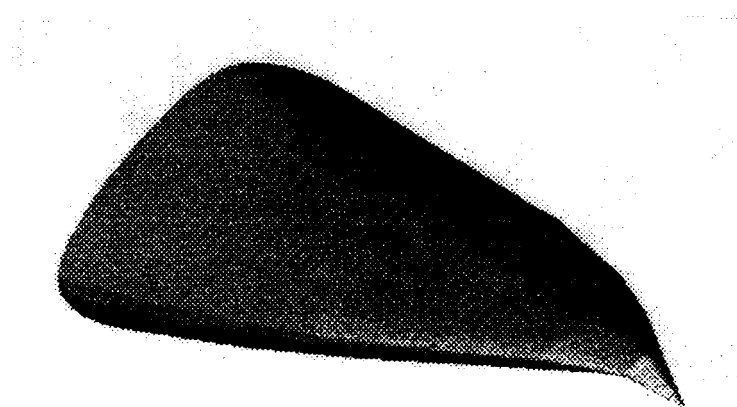


Figure 76 Starting from an initially conical body at zero angle of attack and a hypersonic oncoming flow, the hybrid GA/gradient search constrained algorithm arrived at this smooth, cambered 3-D lifting body with a maximized value of lift-to-drag while maintaining the initial volume and length [203], [180].

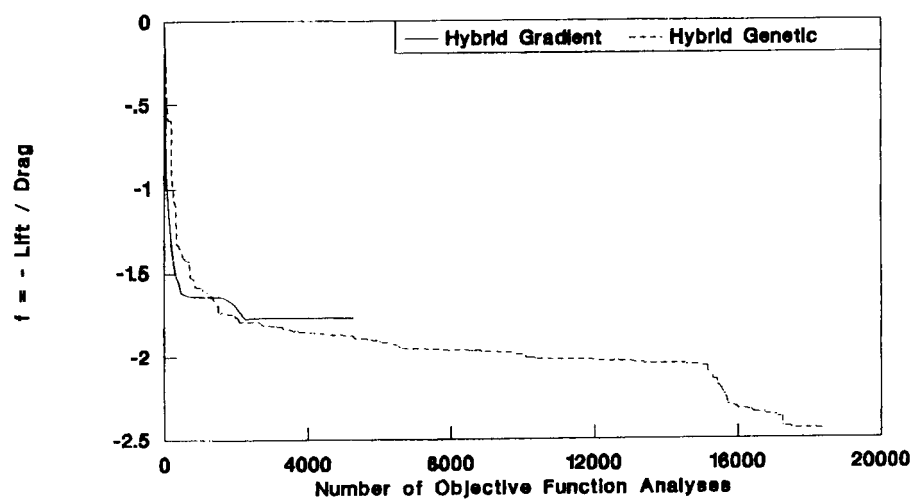


Figure 77 Comparison of convergence histories of a constrained gradient search and a constrained hybrid genetic/gradient search optimizer for example shown in Figure 76, [203], [180].

11.5 Conclusions

From the brief survey of positive and negative features of different optimization algorithms it can be concluded that:

- brute force application of gradient search sensitivity-based optimization methods is very computationally intensive and unreliable for large aerodynamic problems since they are prone to local minima,
- brute force application of genetic evolution optimizers is very computationally intensive for realistic 3-D problems that involve a number of constraints, and
- use of hybrid genetic/gradient search constrained optimization algorithms offers the most reliable performance and the best convergence.

11.6 References

- [163] **Dulikravich, G. S.**
Aerodynamic Shape Design and Optimization: Status and Trends, AIAA Journal of Aircraft, Vol. 29, No. 5, Nov./Dec. 1992, pp. 1020-1026.
- [164] **Dulikravich, G. S.**
Shape Inverse Design and Optimization for Three-Dimensional Aerodynamics, AIAA invited paper 95-0695, AIAA Aerospace Sciences Meeting, Reno, NV, January 9-12, 1995; also to appear in AIAA Journal of Aircraft.
- [165] **Dulikravich, G. S., Martin, T. J.**
Inverse Design of Super-Elliptic Cooling Passages in Coated Turbine Blade Airfoils, AIAA J. of Thermophysics and Heat Transfer, Vol. 8, No. 2, April-June, 1994, pp. 288-294.
- [166] **Dulikravich, G. S.**
Inverse Design and Active Control Concepts in Strong Unsteady Heat Conduction, Applied Mechanics Reviews, Vol. 41, No. 6, June 1988, pp. 270-277.
- [167] **Vanderplaats, G. N.**
Numerical Optimization Techniques for Engineering Design, McGraw-Hill, New York, 1984.
- [168] **Goldberg, D. E.**
Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, 1989.
- [169] **Davis, L.**
Handbook of Genetic Algorithms, Reinhold, 1990.

-
- [170] **Madabhushi, R.K., Levy, R., Pinkus, S. M.**
Design of Optimum Ducts Using an Efficient 3-D Viscous Computational Flow Analysis, Proc. of 2nd Int. Conf. on Inverse Design Concepts and Optimiz. in Eng. Sci. (ICIDES-II), ed. G.S. Dulikravich, Dept. of Aero. Eng., Penn State Univ., University Park, PA, Oct. 26-28, 1987, pp.147-166.
- [171] **Huddleston, D. H., Mastin, C. W.**
Optimization of Aerodynamic Designs Using Computational Fluid Dynamics, Proc. of 7th Int. Conf. on Finite Element Meth. in Flow Problems, ed. Chung, T.J. and Karr, G.R., Univ. of Alabama in Huntsville Press, April 3-7, 1989, pp. 899-907.
- [172] **Cosentino, G. B.**
Numerical Optimization Design of Advanced Transonic Wing Configurations, AIAA paper 85-0424, Reno, NV, Jan. 14-17, 1985.
- [173] **van Egmond, J. A.**
Numerical Optimization of Target Pressure Distributions for Subsonic and Transonic Airfoil Design, Computational Methods for Aerodynamic Design (Inverse) and Optimization, AGARD-CP-463, Loen, Norway, May 22-23, 1989, Ch. 17.
- [174] **van den Dam, R. F., van Egmond, J. A., Slooff, J. W.**
Optimization of Target Pressure Distributions, AGARD Report No. 780, Rhode-St.-Genese, Belgium, May 14-18, 1990.
- [175] **Dulikravich, G.S., Sheffer, S.**
Aerodynamic Shape Optimization of Arbitrary Hypersonic Vehicles, Proc. of 3rd Int. Conf. on Inverse Design Concepts and Optimiz. in Eng. Sci. (ICIDES-III), ed. G.S. Dulikravich, Washington, D.C., October 23-25, 1991.
- [176] **Sheffer, S. G., Dulikravich, G.S.**
Constrained Optimization of Three-Dimensional Hypersonic Vehicle Configurations, AIAA paper 93-0039, AIAA Aerospace Sciences Meeting, Reno, NV, Jan. 11-14, 1993.
- [177] **Dulikravich, G. S., Sheffer, S. G.**
Aerodynamic Shape Optimization of Hypersonic Configurations Including Viscous Effects, AIAA 92-2635, AIAA 10th Applied Aerodynamics Conference, Palo Alto, CA, June 22-24, 1992.
- [178] **Mortenson, M. E.**
Geometric Modeling, John Wiley & Sons, 1985, pp. 125-134.
- [179] **Farin, G.**
Curves and Surfaces for Computer Aided Geometric Design, Second Edition, Academic Press, Boston, 1990.
- [180] **Foster, N. F., Dulikravich, G. S., Bowles, J.**
Three-Dimensional Aerodynamic Shape Optimization Using Genetic Evolution and Gradient Search Algorithms, AIAA paper 96-0555, AIAA Aerospace Sciences Meeting, Reno, NV, January 15-19, 1996.
- [181] **Sobieczky, H., Stroeve, J. C.**

- Generic Supersonic and Hypersonic Configurations, AIAA paper 91-3301-CP, Proc. 9th AIAA Applied Aerodynamics Conference, Baltimore, MD, Sept. 22-24, 1991.
- [182] **Kubrynski, K.**
Design of 3-Dimensional Complex Airplane Configurations with Specified Pressure Distribution via Optimization, Proc. of 3rd Int. Conf. on Inverse Design Concepts and Optimiz. in Eng. Sci. (ICIDES-III), editor: Dulikravich, G. S., Washington, D. C., October 23-25, 1991.
- [183] **Rizk, M. H.**
Aerodynamic Optimization by Simultaneously Updating Flow Variables and Design Parameters, AGARD CP-463, Loen, Norway, May 22-23, 1989.
- [184] **Tortorelli, D. A., Michaleris, P.**
Design Sensitivity Analysis: Overview and Review, Inverse Problems in Engineering, Vol. 1, No. 1, 1994, pp. 71-105.
- [185] **Taylor, A. C., Newman, P. A., Hou, G. J.-W., Jones, H. E.**
Recent Advances in Steady Compressible Aerodynamic Sensitivity Analysis, IMA Workshop on Flow Control, Institute for Math. and Appl., Univ. of Minnesota, Minneapolis, MN, November 1992.
- [186] **Elbanna, H. M., Carlson, L. A.**
Determination of Aerodynamic Sensitivity Coefficients Based on the Three-Dimensional Full Potential Equation, AIAA paper 92-2670, June 1992.
- [187] **Baysal, O., Eleshaky, M. I.**
Aerodynamic Design Optimization Using Sensitivity Analysis and Computational Fluid Dynamics, AIAA paper 91-0471, Reno, NV, Jan. 7-10, 1991.
- [188] **Korivi, V. M., Taylor, A. C., Hou, G. W., Newman, P. A., Jones, H.**
Sensitivity Derivatives for Three-Dimensional Supersonic Euler Code Using Incremental Iterative Strategy, AIAA Journal, Vol. 32, No. 6, June 1994, pp. 1319-1321..
- [189] **Felker, F.**
Calculation of Optimum Airfoils Using Direct Solutions of the Navier-Stokes Equations, AIAA paper 93-3323-CP, Orlando, FL, July 1993.
- [190] **Huddleston, D. H., Soni, B. K., Zheng, X.**
Application of a Factored Newton-Relaxation Scheme to Calculation of Discrete Aerodynamic Sensitivity Derivatives, AIAA paper 94-1894, Colorado Springs, CO, June 1994.
- [191] **Hajela, P.**
Genetic Search - An Approach to the Nonconvex Optimization Problem, AIAA Journal, Vol. 28, No. 7, 1990, pp. 1205-1210.
- [192] **Misegades, K. P.**
Optimization of Multi-Element Airfoils, Von Karman Institute for Fluid Dynamics, Belgium, Project Report 1980-5, June 1980.

-
- [193] **Gregg, R. D., Misegades, K. P.**
Transonic Wing Optimization Using Evolution Theory, AIAA paper 87-0520, January 1987.
- [194] **Ghielmi, L., Marazzi, R., Baron, A.**
A Tool for Automatic Design of Airfoils in Different Operating Conditions, AGARD-CP-463, Loen, Norway, May 22-23, 1989.
- [195] **Rocchetto, A., Poloni, C.**
A Hybrid Numerical Optimization Technique Based on Genetic and Feasible Direction Algorithms for Multipoint Helicopter Rotor Blade Design, Twenty First European Rotorcraft Forum, St. Petersburg, Russia, 1995, Paper No II7,1-19.
- [196] **Gage, P., Kroo, I.**
A Role for Genetic Algorithms in a Preliminary Design Environment, AIAA paper 93-3933, August 1993.
- [197] **Mosetti, G., Poloni, C.**
Aerodynamic Shape Optimization by Means of a Genetic Algorithm, Proceedings of the 5th International Symposium on Computational Fluid Dynamics, (editor: H. Daiguji), Sendai, Japan, Japan Society for CFD, Vol. II, 1993, pp. 279-284.
- [198] **Crispin, Y.**
Aircraft Conceptual Optimization Using Simulated Evolution, AIAA paper 94-0092, Reno, NV, January 10-13, 1994.
- [199] **Quagliarella, D., Cioppa, A. D.**
Genetic Algorithms Applied to the Aerodynamic Design of Transonic Airfoils, AIAA paper 94-1896, Colorado Springs, CO, June 1994.
- [200] **Yamamoto, K., Inoue, O.**
Applications of Genetic Algorithm to Aerodynamic Shape Optimization, AIAA CFD Conference Proceedings, San Diego, CA, June 1995, AIAA-95-1650-CP, pp. 43-51.
- [201] **Dervieux, A., Male, J.-M., Marco, N., Periaux, J., Stoufflet, B., Chen, H. Q., Cefroui, M.**
Numerical vs. Non-Numerical Robust Optimisers for Aerodynamic Design Using Transonic Finite-Element Solvers, AIAA CFD Conference Proceedings, San Diego, CA, June 1995, AIAA-95-1761-CP, pp. 1339-1347.
- [202] **Poloni, C., Mosetti, G.**
Aerodynamic Shape Optimization by Means of Hybrid Genetic Algorithm, Proc. 3rd Internat. Congress on Industrial and Appl. Math., Hamburg, Germany, July 3-7, 1995.
- [203] **Foster, N. F.**
Shape Optimization Using Genetic Evolution and Gradient Search Constrained Algorithms, M. Sc. thesis, Dept. of Aerospace Eng., The Pennsylvania State University, University Park, PA, August 1995.

GEORGE S. DULIKRAVICH

INTERNATIONAL CENTRE FOR MECHANICAL SCIENCES

COURSES AND LECTURES - No. 366



NEW DESIGN CONCEPTS
FOR HIGH SPEED AIR TRANSPORT

EDITED BY

H. SOBIECZKY
DLR GERMAN AEROSPACE RESEARCH ESTABLISHMENT



SpringerWienNewYork

Chapter 12

Combined Optimization and Inverse Design of 3-D Aerodynamic Shapes

George S. Dulikravich

12.1 Introduction

The main drawback of using constrained optimization in 3-D aerodynamic shape design is that it requires anywhere from hundreds to tens of thousands of calls to a 3-D flow-field analysis code. Since certain general 3-D aerodynamic shape inverse design methodologies require only a few calls to a modified 3-D flow-field analysis code, it would be highly desirable to create a hybrid new design algorithm that would combine some of the best features of both approaches while requiring less computing time than a few dozen calls to the 3-D flow-field analysis code. We will discuss two such hybrid design formulations that have been proven to work and are distinctly different from each other.

12.2 Target Pressure Optimization Followed by an Inverse Design

The unique feature of this concept [204], [205] is that it offers the most economical approach to constrained aerodynamic shape optimization. It consists of two steps: surface pressure constrained optimization followed by an inverse shape design. This approach avoids most of the limitations of the inverse shape design while requiring considerably less computing time than the direct geometry optimization.

Surface pressure optimization phase of this design approach starts by parameterizing the initial surface pressure distribution using β -splines [206]. Typically, between five and ten control points in the β -spline representation are sufficient to describe the pressure distribution on the upper surface and the similarly small number of control points is sufficient for representation of the pressure distribution on the lower surface of a two-dimensional (2-D) section of the 3-D object. For example, if 7 control points are used to parameterize the upper surface pressure distribution and 7 control points are used to parameterize the lower surface pressure distribution (Figure 78), then 4 control points will have to be used to fix the locations of leading and trailing edge stagnation points. Constraints can be introduced at this stage by specifying the slopes of the pressure distribution at the leading edge. This will require fixing one additional control point on the upper and the lower surface pressure distribution [204]. The steeper the leading edge pressure distribution variation, the larger the leading edge radius of the resulting 2-D aerodynamic section will be. The optimization of the constrained surface pressure distribution can then be achieved [207], [208] by optimizing the 8 "floating" control points each defined by its x -coordinate and the corresponding value of pressure. The optimization is thus performed on surface pressure distribution, not on the actual aerodynamic shape.

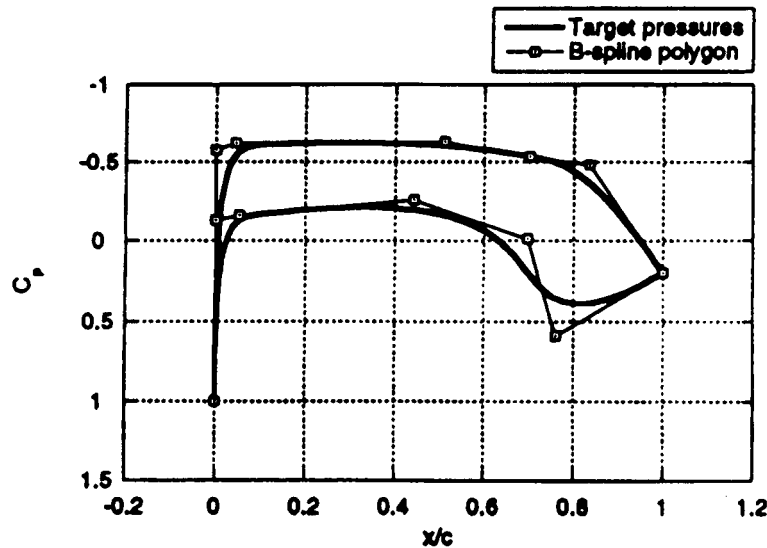


Figure 78 An example of surface pressure coefficient discretization using β -splines with enforced constraints on the slopes of the C_p curves at the leading edge stagnation point [204].

This means that the perturbations introduced in the surface pressure distribution during the optimization will have to be somehow related to the global objectives like aerodynamic lift, drag, etc. These global aerodynamic parameters depend on the geometric shape of the object and on the pressure distribution on its surface. It is therefore necessary to relate the geometry of the object and the surface pressure distribution. This is precisely what a typical flow-field analysis code should do except for the fact that geometry of the object is not known yet. A possibility would be to run an inverse shape design code for each surface pressure perturbation and then to integrate the surface pressure in order to evaluate the corresponding lift, drag, etc. This is

obviously an unacceptably, inefficient approach.

Instead, the guessed surface pressure distribution can be optimized without ever calling a flow-field analysis code or the inverse shape design code. To accomplish this, use is made of extremely efficient approximate relations [209], [210] relating the integrated surface pressure distribution and the object's maximum relative thickness [209], specified location of the flow transition points, and the aerodynamic drag with any of the classical boundary layer solutions [210]. Then, the optimization of the few coefficients, for example, of β -spline parameterization of the surface pressure distribution curves on each 2-D section can be performed reliably and economically using an optimization algorithm [211], [204], [212]. Optimization of the surface pressure distribution parameters, instead of directly optimizing the 3-D geometry parameters, has several advantages: the total number of optimization variables is reduced, the range of β -spline coefficient variations can be large, the design space does not have an excessive number of local minima, and the sensitivity derivatives do not have discontinuities. This procedure easily accepts constraints on surface pressure distribution such as the desired slopes of the curves at the leading and trailing edges, the maximum value of negative coefficient of pressure, a condition that the surface pressure distribution curves on the upper and the lower surface never cross each other thus avoiding locally negative lift force. These optimization tasks can be reliably achieved using a genetic evolution optimization algorithm [204], although an even more reliable and computationally faster approach would be to use a hybrid genetic/gradients search optimization algorithm [212], [213].

The second phase of this combined 3-D shape design procedure utilizes the optimized surface pressure distribution and any of the fast inverse shape design algorithms [214], [204], [215] to find the corresponding 3-D configuration (Figure 79).

The entire two-step design algorithm requires a minimum development time since β -spline discretization codes, constrained optimization codes, flow-field analysis codes, and inverse design codes are available to most aerodynamicists. If the optimized surface pressure distribution results in a 3-D geometry which violates some of the local geometric constraints, the computed pressure at the constrained points can be used instead of the optimized local pressure. Moreover, this approach gives the designer a partial control of the key elements of the design by asking him to specify a few key bounding features of the "good" surface pressure distribution and then letting the robust and fast optimizer find the details of the optimal pressure distribution. Using this approach to shape optimization the designer also has the ability to visually control and terminate or restart the design process if he decides that the values of some of his specified constraints could be improved.

The method offers the most economical and robust approach to 3-D constrained aerodynamic shape optimization (Figure 80) that consumes approximately 5 - 10 times the computing time needed by a single 3-D flow-field analysis [204]. This is much more cost-effective than the classical approach of simultaneously optimizing the surface pressure distribution and the corresponding 3-D geometry which costs an equivalent of hundreds and even thousands of flow field analysis runs.

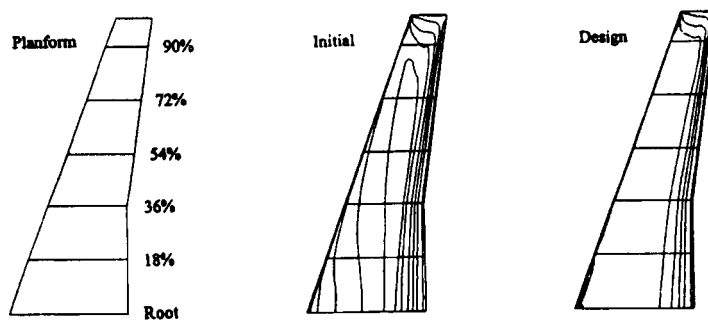


Figure 79 An example of an initial and an optimized surface pressure distribution on a 3-D transonic wing planform. Notice that the optimized pressure distribution is of the flat "roof-top" type [204].

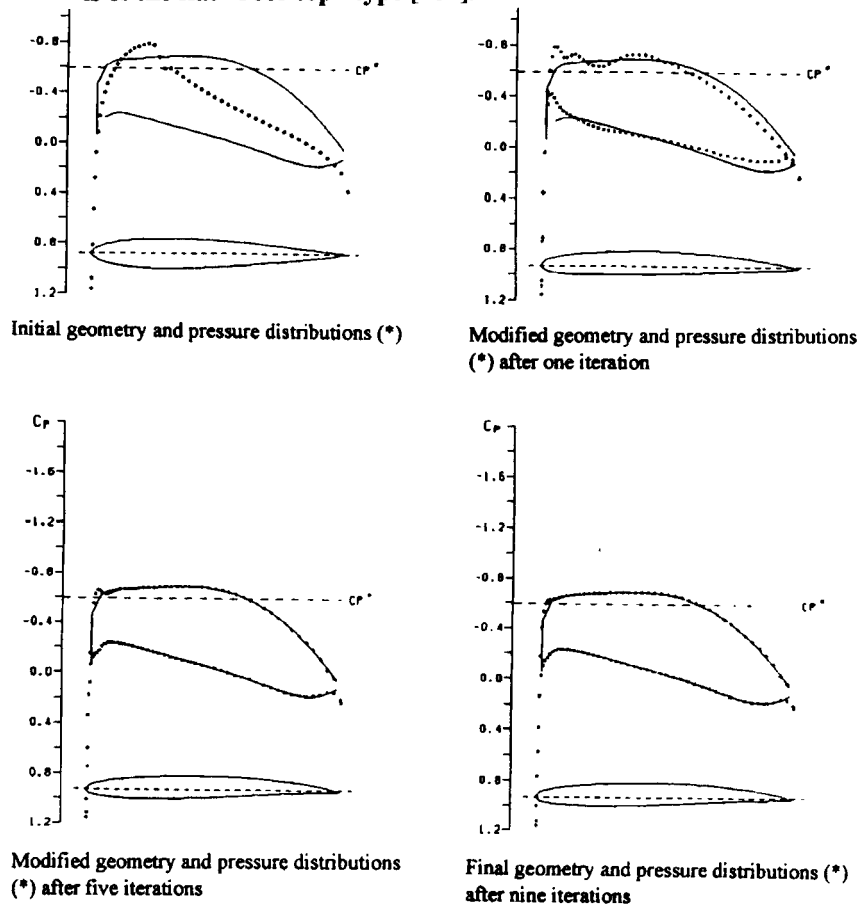


Figure 80 An example of a convergence history of target pressure optimization followed by transpiration inverse design [204].

12.3 Control Theory (Adjoint Operator) Approach

With the increasing number of variables that need to be optimized, the discrete function space containing the optimization variables tends towards a continuous function space for which it is possible to use a global analytical formulation (an adjoint system) instead of the local discretized formulation (gradient search and genetic evolution algorithms). The control theory (adjoint operator) concept is essentially a gradient search optimization approach where the gradient information is obtained by formulating and solving a set of adjoint partial differential equations rather than evaluating the derivatives using finite differencing. Like the classical optimization algorithms, the control theory (adjoint operator) formulation can be used either for optimizing a 3-D aerodynamic shape by maximizing its lift, minimizing drag, etc., or it can be used as a tool to enforce the desired surface pressure distribution in an inverse shape design process. If the governing system of partial differential equations is large and the solution space is relatively smooth, then the control theory (adjoint operator) approach is more appropriate than the classical optimization approaches [216]-[218]. The mathematical formulation is very involved and was only briefly sketched in the previous lecture.

There have been several approaches at creating an aerodynamics control theory (adjoint operator) formulation [216]-[233]. The early efforts [216]-[220] were highly mathematical, hard to understand and computationally intensive. Since then, two similar approaches have been followed by most researchers. They are associated with the research teams of Professor A. Jameson [221]-[225] who adopted and extended the previous efforts by the French researchers, and the research team of Professor V. Modi [226]-[228] who followed a more comprehensible mechanistic approach practised by researchers in the general fields of inverse shape design in elasticity and heat conduction.

To date, successful and impressive results have been obtained by both research groups where Jameson's group focused on transonic inviscid flow model and Modi's group focused on incompressible viscous laminar and turbulent flow models of the flow field. Published results of 3-D elbow diffuser and airfoil shape optimization using the adjoint operator approach and incompressible laminar and turbulent flow Navier-Stokes equations [226]-[228] suggest that a typical optimized design consumes the amount of computing time that is equivalent to between 20 - 40 flow-field analysis (Figure 81, Figure 82). Similar total effort (an equivalent of 30 - 60 flow-field analysis) was reported for 3-D transonic isolated wing design using Euler equations [21, 22]. This could be compared with several hundreds and even thousands of flow analysis runs when using a typical genetic algorithm or a typical gradient search optimization algorithm. These results dispel earlier reservations [30] that adjoint operator approach formulations might not be computationally efficient since they involve the solution of the governing flow-field equations, an additional set of adjoint equations, and several more interface partial differential equations.

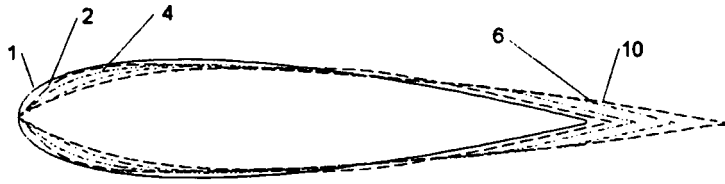


Figure 81 History of the iterative evolution of a minimum drag airfoil starting from a NACA0018 airfoil at $Re = 5000$. Labeling numbers correspond to the iterative cycles each consisting of one flow-field analysis and one solution of the adjoint system [228].

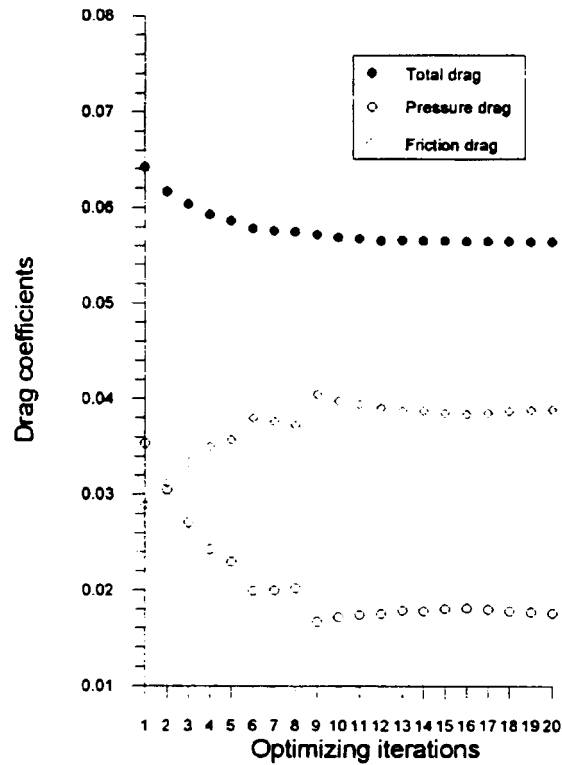


Figure 82 History of the iterative evolution of hydrodynamic drag coefficient during the shape optimization process with initial shape chosen as NACA 0018 and $Re = 5000$ [228].

Although very impressive and mathematically involved, the general control theory (adjoint operator) approach has certain problems. One drawback is that it does not always allow for flow separation. The method also suffers from tendency to converge to any of the numerous local minima like most of the gradient search optimizers. An additional drawback is that it requires the derivation of an entirely new system of partial differential equations in terms of some non-physical adjoint variables and specification of their boundary conditions. Choosing correct boundary conditions for the adjoint system is quite a challenge since the adjoint variables are not physical flow quantities [228]. There are many ways to derive the adjoint system and some additional partial differential equations coupling the original system and the adjoint system [221]-[226]. If the adjoint system of partial differential equations is different from the original system of the flow-field governing equations, a significant effort needs to be invested in separately coding the two systems. If the adjoint system is almost the same as the original governing system, the numerical algorithms for the two systems are practically the same and the entire approach could be implemented more readily using the existing CFD analysis software [228]. Actually, if the adjoint system has the same form as the flow-field governing system except that the sign of the convective term in the adjoint system is negative, the solution of the adjoint system will represent a flow in the reverse direction (Figure 83 and Figure 84).

The complexity of the entire adjoint operator formulation makes it difficult to comprehend, implement and modify. Moreover, the adjoint operator formulation is very specific and different formulation needs to be developed and coded for each flow field model (Euler, parabolized Navier-Stokes, Navier-Stokes, etc.). The control theory (adjoint operator) approach is very attractive if the designers want to use only one specific flow-field analysis code as the basis for a design code and if they want to perform design in only one discipline (for example, aerodynamic shape design only). Since the designers use a variety of progressively more sophisticated design codes during the design process and since the design objectives are inherently multidisciplinary, the control theory (adjoint operator) approach can hardly be justified in the context of the multidisciplinary objectives, funds typically available, and the time limits imposed on the designer.

In the case of a truly multidisciplinary problem, a new adjoint system would have to be derived and coded to solve several adjoint systems simultaneously. Since each disciplinary analysis and adjoint system usually has vastly different time scales, the combined multidisciplinary analysis and adjoint system would have a slow convergence rate and an overall marginal stability because of a very large number of local minimums. Reliability of such a design system might be questionable since it is known that sensitivity derivatives for highly non-linear systems might be discontinuous [207]. The adjoint system must be discretized for an approximation to the gradient to be found. This approach is less reliable than the implicit function theorem approach [231], [232]. Therefore, finding the gradients of the objective function using information from the discretized flow-field governing equations is more reliable [232] than if an analytic expression for the gradient is derived in terms of the exact flow-field solution and the solution to the adjoint system [223].

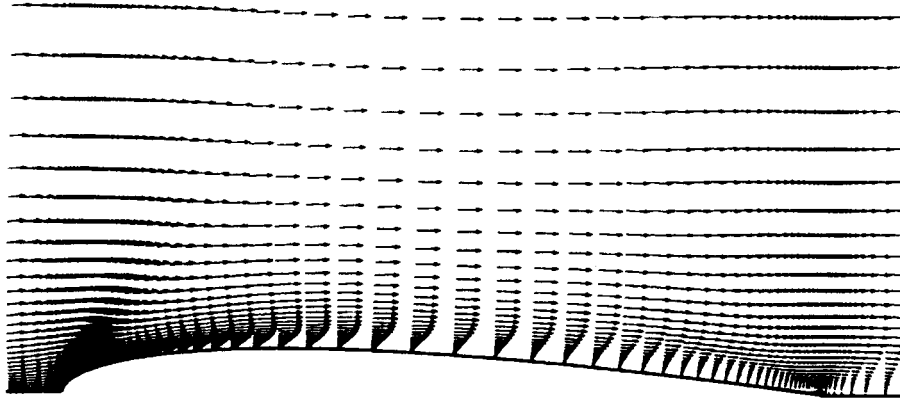


Figure 83 Horizontal components of the fluid velocity vector at different axial locations; the flow is from left to right [228].

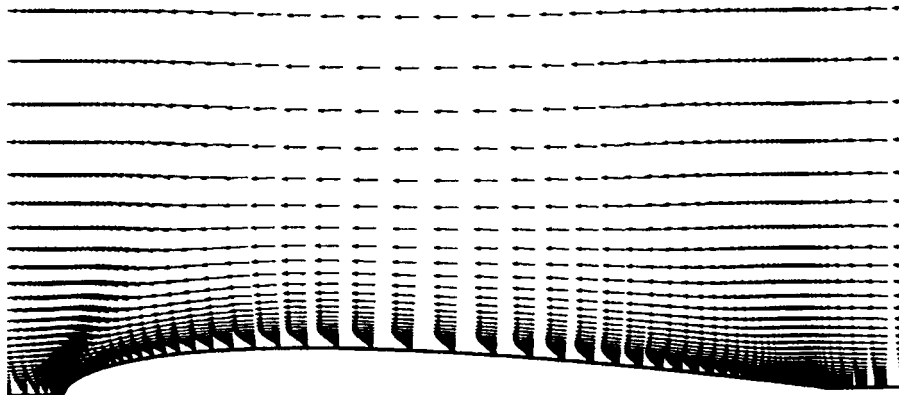


Figure 84 Horizontal components of an adjoint variable at different axial locations; the "flow" of the adjoint vector is in the opposite direction to the physical fluid flow if the adjoint system is made to resemble the flow-field governing system except for the change in sign of the convective term [228].

An attractive extension of the control theory (adjoint operator) approach represents the "one-shot" method [234]-[237]. It implicitly combines a multigrid solution technique with the classical control theory for systems of partial differential equations. Consequently, this approach is faster than the regular control theory (adjoint operator) approach where the multigrid technique was used only to accelerate the flow-field analysis code [225]. Nevertheless, the

"one-shot" formulation and implementation are even more complex than the control theory (adjoint operator) formulation, and it suffers from similar reliability issues due to the fact that it might be equally prone to the local minima.

12.4 Conclusions

A number of existing and emerging concepts and methodologies applicable to automatic inverse design and optimization of arbitrary realistic 3-D configurations have recently been surveyed and compared [235]. These attempts to classify the design methods and to expose their major advantages and disadvantages resulted in the following conclusions:

- control theory (adjoint operator) algorithms offer very economical shape design optimization although they are complex to understand and develop, hard to modify, too field-specific, and prone to local minima,
- one-shot method should be further researched as an even more economical possible successor to the adjoint operator formulations with all of its drawbacks, and
- combination of hybrid genetic/gradient search constrained optimization of surface pressure distribution followed by transpiration inverse design offers an attractive approach in the immediate future because of its unsurpassed robustness and acceptable computing cost.

12.5 References

- [204] **Obayashi, S., Takanashi, S.**
Genetic Optimization of Target Pressure Distributions for Inverse Design Methods, AIAA Paper 95-1649, San Diego, CA, June 19-22, 1995.
- [205] **Goel, S., Lamson, S.**
Automating the Design Process for 3D Turbine Blades, ASME WAM'95, Nov. 14-17, 1995, in Symp. for Design & Optimization (editor: O. Baysal), ASME Vol. 232, pp. 85-95.
- [206] **Rogers, D. F., Adams, J. A.**
Mathematical Elements for Computer Graphics, Second Edition, McGraw-Hill, Inc., 1990.
- [207] **van Egmond, J. A.**
Numerical Optimization of Target Pressure Distributions for Subsonic and Transonic Airfoil Design, AGARD-CP-463, Loen, Norway, May 22-23, 1989, Ch. 17.
- [208] **van den Dam, R. F., van Egmond, J. A., Slooff, J. W.**
Optimization of Target Pressure Distributions, AGARD Report No. 780, Rhode-St.-Gen-

- ese, Belgium, May 14-18, 1990.
- [209] **Inger, G. R.**
Application of Oswatitsch's Theorem to Supercritical Airfoil Drag Calculation, *J. of Aircraft*, Vol. 30, No. 3, May-June 1993, pp. 415-416.
- [210] **Young, A. D.**
Boundary Layers, AIAA Education Series, 1989.
- [211] **Davis, L.**
Handbook of Genetic Algorithms, Reinhold, 1990.
- [212] **Foster, N. F., Dulikravich, G. S., Bowles, J.**
Three-Dimensional Aerodynamic Shape Optimization Using Genetic Evolution and Gradient Search Algorithms, AIAA paper 96-0555, AIAA Aerospace Sciences Meeting, Reno, NV, January 15-19, 1996.
- [213] **Poloni, C., Mosetti, G.**
Aerodynamic Shape Optimization by Means of Hybrid Genetic Algorithm, *Proc. 3rd Internat. Congress on Industrial and Appl. Math.*, Hamburg, Germany, July 3-7, 1995.
- [214] **Takanashi, S.**
Iterative Three-Dimensional Transonic Wing Design Using Integral Equations, *J. of Aircraft*, Vol. 22, No. 8, August 1985, pp. 655-660.
- [215] **Wang, Z., Dulikravich, G. S.**
Inverse Shape Design of Turbomachinery Airfoils Using Navier-Stokes Equations, AIAA paper 95-0304, Reno, NV, January 9-12, 1995.
- [216] **Lions, J. L.**
Optimal Control of Systems Governed by Partial Differential Equations, Springer-Verlag, New York, 1971. Translated by S. K. Mitter.
- [217] **Angrand, F.**
Optimum Design for Potential Flows, *International Journal for Numerical Methods in Fluids*, Vol. 3, 1983, pp. 265-282.
- [218] **Abergel, F., Temam, R.**
On Some Control Problems in Fluid Mechanics, *Theoretical and Comput. Fluid Dynamics*, Vol. 1, 1990, pp. 303-325.
- [219] **Gu, C.-G., Miao, Y.-M.**
Blade Design of Axial-Flow Compressors by the Method of Optimal Control Theory--Physical Model and Mathematical Expression, ASME paper 86-GT-183 presented at the 31st Internat. Gas Turbine Conf. and Exhibit, Dusseldorf, Germany, June 1986.
- [220] **Gu, C.-G., Miao, Y.-M.**
Blade Design of Axial-Flow Compressors by the Method of Optimal Control Theory--Application of Pontryagin's Maximum Principles, a Sample Calculation and Its Results, ASME paper 86-GT-182 presented at the 31st Internat. Gas Turbine Conf. and Exhibit, Dusseldorf, Germany, June 1986.

-
- [221] **Jameson, A.**
Aerodynamic Design Via Control Theory, J. of Scientific Computing, Vol. 3, 1988, pp. 233-260.
- [222] **Jameson, A.**
Optimum Aerodynamic Design via Boundary Control, AGARD-FDP-VKI Special Course, VKI, Rhode St.-Genese, Belgium, April 25-29, 1994.
- [223] **Reuther, J., Jameson, A.**
Control Theory Based Airfoil Design for Potential Flow and a Finite Volume Discretization, AIAA paper 94-0499, Reno, NV, January 1994.
- [224] **Reuther, J., Jameson, A.**
Control Based Airfoil Design Using the Euler Equations, AIAA paper 94-4272-CP, 1994.
- [225] **Jameson, A.**
Optimum Aerodynamic Design Using CFD and Control Theory, AIAA CFD Conference Proceedings, San Diego, CA, June 1995, AIAA-95-1729-CP, pp. 926-949.
- [226] **Cabuk, H., Modi, V.**
Optimum Design of Oblique Flow Headers, ASHRAE Winter Meeting, New York, NY, January 19-23, 1991.
- [227] **Cabuk, H., Sung, C.-H., Modi, V.**
Adjoint Operator Approach to Shape Design for Internal Incompressible Flows, Proc. of the 3rd Internat. Conf. on Inverse Design Concepts and Opt. in Eng. Sci. (ICIDES-III), Ed.: G.S. Dulikravich, Washington, D.C., Oct. 23-25, 1991, pp. 391-404.
- [228] **Huan, J., Modi, V.**
Optimum Design of Minimum Drag Bodies in Incompressible Laminar Flow Using a Control Theory Approach, Inverse Problems in Engineering, Vol. 1, No. 1, 1994, pp. 1-25.
- [229] **Zhang, J., Chu, C. K., Modi, V.**
Design of Plane Diffusers in Turbulent Flow, Inverse Problems in Engineering, Vol. 2, No. 2, 1995, pp. 85-102.
- [230] **Huan, J., Modi, V.**
Design of Minimum Drag Bodies in Incompressible Laminar Flow, Inverse Problems in Engineering, in press 1996.
- [231] **Frank, P. D., Shubin, G. R.**
A Comparison of Optimization-Based Approaches for a Model Computational Aerodynamics Design Problem, Journal of Computational Physics, Vol. 98, 1992, pp. 74-89.
- [232] **Dixon, A. E., Fletcher, C. A. J.**
Optimization Applied to Aerofoil Design, Proc. 5th Int. Symp. on CFD, Sendai, Japan, 1993.

-
- [233] **Dulikravich, G. S.**
Aerodynamic Shape Design and Optimization: Status and Trends, AIAA Journal of Aircraft, Vol. 29, No. 5, Nov./Dec. 1992, pp. 1020-1026.
- [234] **Dulikravich, G. S.**
Shape Inverse Design and Optimization for Three-Dimensional Aerodynamics, AIAA invited paper 95-0695, AIAA Aerospace Sciences Meeting, Reno, NV, January 9-12, 1995; also to appear in AIAA Journal of Aircraft.
- [235] **Ta'asan, S.**
One Shot Methods for Optimal Control of Distributed Parameter Systems I: Finite Dimensional Control, ICASE Report No. 91-2, January 1991.
- [236] **Arian, E., Ta'asan, S.**
Multigrid One Shot Methods for Optimal Control Problems: Infinite Dimensional Control, ICASE Report No. 94-52, July 1994.
- [237] **Ta'asan, S.**
Trends in Aerodynamics Design and Optimization: A Mathematical Viewpoint, AIAA CFD Conference Proceedings, San Diego, CA, June 1995, AIAA-95-1731-CP, pp. 961-970.

GEORGE S. DULIKRACH

INTERNATIONAL CENTRE FOR MECHANICAL SCIENCES

COURSES AND LECTURES - No. 366



NEW DESIGN CONCEPTS
FOR HIGH SPEED AIR TRANSPORT

EDITED BY

H. SOBIECZKY
DLR GERMAN AEROSPACE RESEARCH ESTABLISHMENT



SpringerWienNewYork

Chapter 13

Thermal Inverse Design and Optimization

George S. Dulikravich

13.1 Introduction

During the design of high speed flight vehicles the designer should take into account the aerodynamic heating due to surface friction and the high temperature air behind the strong shock waves. The allowable exterior surface temperatures are limited by the material properties of the skin material of the flight vehicle. In addition, the amount of heat that penetrates the skin structure should be minimized since it will have to be absorbed by the fuel and not allowed to enter the passenger cabin. A typical remedy is to cool the structure by pumping a cooling fluid (typically the fuel) through numerous passages manufactured inside the outer structure of the flight vehicle. A design optimization method should therefore provide the designer with a tool to guide the development of innovative designs of internally cooled, thermally coated or non-coated structures that will cost less to manufacture, have a longer life span, be easier to repair, and sustain higher surface temperatures.

13.2 Determination of Number, Sizes, Locations, and Shapes of Coolant Flow Passages

During the past dozen years, the author's research team has been developing a unique inverse shape design methodology and accompanying software which allows a thermal system designer to determine the minimum number and correct sizes, shapes, and locations of coolant passages in arbitrarily-shaped internally-cooled configurations [238]-[255]. The designer needs to specify

both the desired temperatures and heat fluxes on the hot surface, and either temperatures or convective heat coefficients on the guessed coolant passage walls. The designer must also provide an initial guess of the total number, sizes, shapes, and locations of the coolant flow passages. Afterwards, the design process uses a constrained optimization algorithm to minimize the difference between the specified and computed hot surface heat fluxes by automatically relocating, resizing, reshaping and reorienting the initially-guessed coolant passages. All unnecessary coolant flow passages are automatically reduced to a very small size and eliminated while honoring the specified minimum distances between the neighboring passages and between any passage and the thermal barrier coating if such exists.

This type of computer code is highly economical, reliable and geometrically flexible if it utilizes the boundary element method (BEM) instead of finite element or finite difference method for the thermal field analysis. The BEM does not require generation of the interior grid and it is non-iterative [238], [239]. Thus the method is computationally efficient and robust. The resulting shapes of coolant passages are smooth, and easily manufacturable.

The methodology has been successfully demonstrated on 2-D coated and non-coated turbine blade airfoils [240]-[249], scramjet combustor struts [252], and 3-D coolant passages in the walls of 3-D rocket engine combustion chambers [253] and 3-D turbine blades [254], [255].

Nonlinear BEM algorithms are the best choice for the thermal analysis because of their computational speed, reliability (due to their non-iterative nature) and accuracy with elliptic type problems. A simple method for escaping local minima has been implemented and involves switching the objective function when a stationary point is achieved [245]. An accurate method, based on exponential spline fitting and interpolation of the cost function values, has been developed for finding the value of optimal search step parameter during gradient search optimization [244]. It is also possible to develop a version of the 3-D inverse shape design code that will allow for multiple realistically shaped coolant flow passages with an arbitrary number of fins or ribs in each of the passages [252] and prespecified locations of struts [242]. In addition, this version of the 3-D inverse design code could allow for a variable thickness, segmented and non-segmented thermal barrier coatings with temperature-dependent thermal conductivities.

13.3 Determination of Steady Thermal Boundary Conditions

Inverse determination of unknown steady thermal boundary conditions when temperature and heat flux data are not available on certain boundaries is an ill-posed problem [256]-[264] (Figure 85). In this case, additional overspecified measurement data involving both temperature and heat flux are required on some other, more accessible boundaries or at a finite number of points within the domain. For example, when using a BEM algorithm, if at all four vertices designated with subscripts 1, 2, 3, and 4 of a quadrilateral computational grid cell the heat sources p_i are known, at two vertices both temperature $\Theta = \bar{\Theta}$ and heat flux $q = \bar{q}$ are known, while at the re-

maining two vertices neither Q or q is known, the boundary integral equation becomes

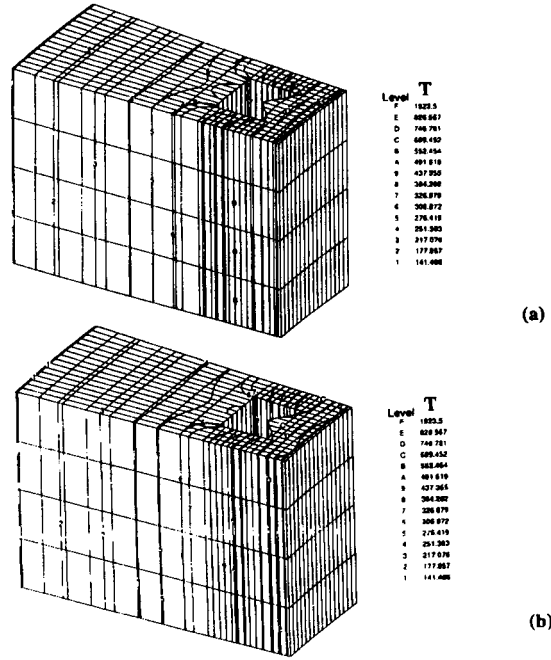


Figure 85 Surface isotherms predicted on a circumferentially-periodic three-dimensional segment of a rocket chamber wall with a cooling channel made of four different materials: a) direct problem, b) inverse boundary condition problem [261].

$$\begin{bmatrix} \tilde{h}_{11} & \tilde{h}_{12} & \tilde{h}_{13} & \tilde{h}_{14} \\ \tilde{h}_{21} & \tilde{h}_{22} & \tilde{h}_{23} & \tilde{h}_{24} \\ \tilde{h}_{31} & \tilde{h}_{32} & \tilde{h}_{33} & \tilde{h}_{34} \\ \tilde{h}_{41} & \tilde{h}_{42} & \tilde{h}_{43} & \tilde{h}_{44} \end{bmatrix} \begin{bmatrix} \bar{\Theta}_1 \\ \bar{\Theta}_2 \\ \bar{\Theta}_3 \\ \bar{\Theta}_4 \end{bmatrix} = \begin{bmatrix} g_{11} & g_{12} & g_{13} & g_{14} \\ g_{21} & g_{22} & g_{23} & g_{24} \\ g_{31} & g_{32} & g_{33} & g_{34} \\ g_{41} & g_{42} & g_{43} & g_{44} \end{bmatrix} \begin{bmatrix} \bar{q}_1 \\ \bar{q}_2 \\ \bar{q}_3 \\ \bar{q}_4 \end{bmatrix} + \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix} \quad (88)$$

Here, coefficients of $[h]$ and $[g]$ matrices are all known because they depend on geometric relations and the configuration is known. In order to solve this set, all of the unknowns will be collected on the right-hand side, while all of the knowns are assembled on the left. A simple algebraic manipulation yields the following set:

$$\begin{bmatrix} \tilde{h}_{11} & -g_{12} & \tilde{h}_{13} & -g_{14} \\ \tilde{h}_{21} & -g_{22} & \tilde{h}_{23} & -g_{24} \\ \tilde{h}_{31} & -g_{32} & \tilde{h}_{33} & -g_{34} \\ \tilde{h}_{41} & -g_{42} & \tilde{h}_{43} & -g_{44} \end{bmatrix} \begin{bmatrix} \bar{\Theta}_2 \\ q_2 \\ \Theta_4 \\ q_4 \end{bmatrix} = \begin{bmatrix} -\tilde{h}_{11} & g_{11} & -\tilde{h}_{13} & g_{13} \\ -\tilde{h}_{21} & g_{21} & -\tilde{h}_{23} & g_{23} \\ -\tilde{h}_{31} & g_{31} & -\tilde{h}_{33} & g_{33} \\ -\tilde{h}_{41} & g_{41} & -\tilde{h}_{43} & g_{43} \end{bmatrix} \begin{bmatrix} \bar{\Theta}_1 \\ \bar{q}_1 \\ \bar{\Theta}_3 \\ \bar{q}_3 \end{bmatrix} + \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix} \quad (89)$$

Since the entire right-hand side is known, it may be reformulated as a vector of knowns, $\{\mathbf{F}\}$. The left-hand side remains in the form $[\mathbf{A}]\{\mathbf{X}\}$. Also, additional equations may be added to the equation set if, for example, temperature or heat flux measurements are known at certain locations within the domain. In general, the geometric coefficient matrix $[\mathbf{A}]$ will be non-square and highly ill-conditioned. Most matrix solvers will not work well enough to produce a correct solution. There exists a very powerful technique for dealing with sets of equations that are either singular or very close to singular. These techniques, known as Singular Value Decomposition (SVD) methods [265], are widely used in solving most linear least squares problems. Thus, using an SVD algorithm it is possible to solve for the unknown surface temperatures and heat fluxes very accurately and non-iteratively. If the general formulation of the problem is to be done with finite elements or any other numerical technique instead of BEM, the inverse boundary condition determination problem would become iterative and would require special regularization methods to keep it stable.

13.4 General Design Objectives

The objective of a conjugate heat transfer design and optimization program should be to provide industry with a modular, reliable and proven design optimization tool that will take into account the interaction of 3-D exterior hot gas aerodynamics, heat convection at the hot exterior surface, heat conduction throughout the structure material, and heat convection on the walls of the interior coolant passages. Most of these concepts have already been individually developed, proven and published by the author and other researchers, thus providing strong assurance that the overall integration can be successfully accomplished.

The entire program should be conveniently broken into individual self-sustaining modular tasks. To remain within the desired time frame and budget, the designers should try to utilize as much as possible the existing analysis, inverse design and constrained optimization computer codes. They should also try to implement methods for the acceleration of iterative algorithms in arbitrary systems of partial differential equations on highly clustered grids [266]. To make the entire design optimization software package as flexible as possible and responsive to the user's needs, it should be implemented on a cluster of disparate microcomputers, workstations, a vector multiprocessor, and a massively parallel processor.

Such a design tool should feature the combination of fully 3-D (not 2-D or quasi 3-D) aero and thermal capabilities dedicated to address user specified design objectives and improvements, and should provide:

- optimization of surface heat fluxes and temperatures for minimum coolant flow rate,
- minimization of the number of coolant flow passages,
- optimization of thicknesses of walls and interior struts, and
- determination of convective heat transfer coefficients on surfaces of 3-D coolant passages.

These design objectives could be accomplished using a number of conceptually different approaches. One specific set of scenarios that has been developed by the author can be summarized as follows.

13.5 Optimization of Surface Heat Fluxes and Temperatures for Minimum Coolant Flow Rate

As a by-product of the aerodynamic inverse design using Navier-Stokes computation of the hot gas flow field (with the specified temperatures as thermal boundary conditions on the hot surface) the corresponding hot surface heat flux distribution can be obtained. Since one of the global objectives is to minimize the coolant mass flow rate, the total integrated hot surface heat flux must be minimized. This can be achieved efficiently by utilizing a hybrid genetic evolution/gradient search constrained optimizer [267] and a reliable thermal boundary layer code. Input to such a code can be the hot surface temperature distribution. This temperature distribution can be discretized using β -splines [268] so that the locations of β -spline vertices (control points) can serve as the design variables. Each perturbation to the location of the β -spline vertices will create different hot surface temperature distribution. Wherever the computed hot surface local temperatures are larger than the maximum allowable temperature specified by the designer, they can be explicitly locally reduced to the maximum allowable temperature.

This temperature distribution and an already optimized hot surface pressure distribution can be used as inputs to the thermal boundary layer code. The hot surface heat flux distribution predicted by the thermal boundary layer code will be integrated to obtain the net heat input to the 3-D structure. After the hot surface temperatures have converged to their values that are compatible with the minimum net heat input to the structure, the aerodynamic shape inverse design Navier-Stokes code can be run again subject to these optimized hot surface temperatures. The resulting 3-D aerodynamic external shape will be slightly different than after the first inverse shape design and if necessary, the entire hot surface thermal optimization will be repeated with the redesigned external shape.

This repetitive simultaneous minimization of the integrated hot surface heat flux and the truncation and maximization of the hot surface local temperatures will converge to the final

aerodynamically optimized 3-D external shape that satisfies optimized surface pressure distribution, maximum allowable surface temperature, and compatible hot surface thermal boundary conditions. The minimized integrated hot surface heat fluxes imply a minimized coolant flow rate requirement. Notice that this entire process does not require knowledge of the thermal field and the coolant flow passage configuration inside the internally cooled structure.

13.6 Minimization of the Number of Coolant Flow Passages

Thermal boundary condition inverse determination BEM codes can then use the hot surface temperatures and heat fluxes and non-iteratively predict the distribution of temperatures and heat fluxes on walls of the coolant passages. Convective heat transfer coefficients can then be computed on the walls of the coolant passages. These coefficients might locally exceed the realistic values attainable with the coolant flow in smooth passages. The locally varying heat convection coefficients should, therefore, be limited to their maximum allowable values. The corresponding modified heat fluxes on the walls of the coolant passages can then be determined. These "cold" heat fluxes and temperatures will then be submitted to the BEM inverse code that will determine the corresponding "hot" temperatures and heat fluxes. The resulting computed hot surface temperatures and heat fluxes will be different from the optimized hot surface values. The difference between the computed and the previously optimized hot surface temperatures and heat fluxes will then serve as the forcing function in a constrained optimization code. It will drive the sizes of unnecessary coolant passages to zero, while relocating, resizing, and reshaping the minimum necessary number of the passages until the differences in the computed and the optimized hot surface heat fluxes and temperatures are negligible. Minimum allowable distances among the coolant passages or from the thermal barrier coating interface will serve as constraints (Figure 86 and Figure 87).

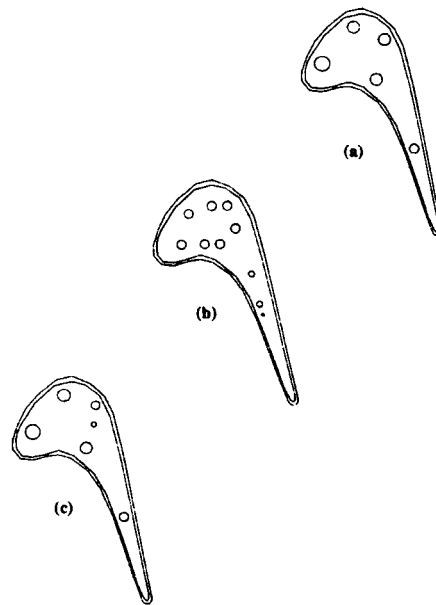


Figure 86 Minimization of the number of circular cross-section coolant passages inside a ceramically coated turbine blade airfoil: a) target geometry; b) initial guess, and c) an almost converged final result of the inverse shape design [245].

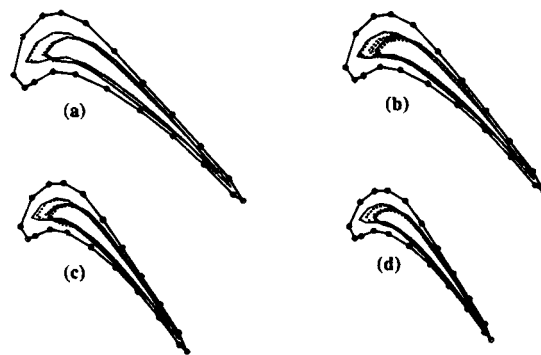


Figure 87 Geometric history of the optimization of a single coolant flow passage in a three-dimensional turbine blade showing sections at (a) $r = 0$, (b) $r = 0.25$, (c) $r = 0.75$, and (d) $r = 1.0$ subject to the specified temperatures and heat fluxes on the blade hot surface and temperatures on the coolant passage wall [253], [254].

13.7 Optimization of Thicknesses of Walls and Interior Struts

Next, the designer could run a 3-D Navier-Stokes coolant flow analysis code and compare its predicted coolant passage wall temperatures with the temperatures obtained in the previous task. The objective could be to minimize the difference in coolant passage surface temperatures obtained from the Navier-Stokes coolant flow analysis code and the heat conduction BEM code. This will be achieved by further altering the shapes of the 3-D coolant passages. This iterative modification of the 3-D coolant passages is required since all thermal boundary conditions on the external and internal surfaces must be compatible with each other and with the corresponding flow fields.

In addition, the converged configuration of the 3-D coolant passages might not be desirable from the manufacturing point of view and might not be structurally sound. Thus, the designer should use the converged configuration of the 3-D coolant passages only as an intelligent initial guess to help in the structural design. For this purpose the designer might wish to specify the locations of mid-planes of the walls separating the converged shapes of 3-D coolant passages and treat them now as mid-planes of the future interior struts. Other design constraints and requirements may be considered and incorporated. This automatically defines the initial thickness distributions of each strut and the wall. These thicknesses can be parameterized with a small number of β -spline coefficients which can then be the design variables in the optimization process.

13.8 Determination of Convective Heat Transfer Coefficients on Surfaces of Coolant Passages

Since the local values for the convective heat transfer coefficient, h_{conv} , on the surfaces of the 3-D coolant flow passages have been specified thus far, it is necessary to compute their actual values. This can be done using a reliable 3-D Navier-Stokes code with the best turbulence model available at the time. With the specified inlet coolant temperature and the already determined temperature distribution on the walls of the coolant flow passages, a single run with the Navier-Stokes code should predict a detailed distribution of h_{conv} on the 3-D surfaces of the coolant flow passages.

If the predicted values of h_{conv} are significantly different from those used in the previous tasks, the last two tasks should be repeated until convergence.

13.9 Conclusions

It is possible for thermal design and optimization to perform relatively efficiently, even in a fully 3-D case, if BEM codes are used for thermal field analysis and inverse determination of unknown boundary conditions and optimization is performed using a hybrid genetic evolution/gradient search constrained optimization algorithm. One specific conjugate heat transfer design and optimization scenario has been suggested for which all of the individual tasks have been proven to work by the author's research team. The main uncertainty of the entire conjugate heat transfer design process still rests with the issue of reliability of turbulence models in typical 3-D Navier-Stokes flow-field analysis codes to predict surface temperatures and heat fluxes.

13.10 References

- [238] **Brebbia, C. A.**
The Boundary Element Method for Engineers, John Wiley & Sons, New York, 1978.
- [239] **Brebbia, C. A., Dominguez, J.**
Boundary Elements, An Introductory Course, McGraw-Hill Book Company, New York, 1989.
- [240] **Kennon, S. R., Dulikravich, G. S.**
The Inverse Design of Internally Cooled Turbine Blades, ASME Journal of Engineering for Gas Turbines and Power, Vol. 107, pp. 123-126, January 1985.
- [241] **Kennon, S. R., Dulikravich, G. S.**
Inverse Design of Multiholed Internally Cooled Turbine Blades, International Journal of Numerical Methods in Engineering, Vol. 22, No. 2, pp. 363-375, 1986.
- [242] **Kennon, S. R., Dulikravich, G. S.**
Inverse Design of Coolant Flow Passages Shapes With Partially Fixed Internal Geometries, International Journal of Turbo & Jet Engines, Vol. 3, No. 1, pp. 13-20, 1986.
- [243] **Chiang, T. L., Dulikravich, G. S.**
Inverse Design of Composite Turbine Blade Circular Coolant Flow Passages, ASME Journal of Turbomachinery, Vol. 108, pp. 275-282, Oct. 1986.
- [244] **Dulikravich, G. S.**
Inverse Design and Active Control Concepts in Strong Unsteady Heat Conduction, Applied Mechanics Reviews, Vol. 41, No. 6, June 1988, pp. 270-277.
- [245] **Dulikravich, G. S., Kosovic, B.**
Minimization of the Number of Cooling Holes in Internally Cooled Turbine Blades, ASME paper 91-GT-103, ASME Gas Turbine Conf., Orlando, Florida, June 2-6, 1991; also in Internat. Journal of Turbo & Jet Engines, Vol. 9, No. 4, pp. 277-283, 1992.

-
- [246] **Dulikravich, G. S.**
Inverse Design of Proper Number, Shapes, Sizes and Locations of Coolant Flow Passages, in Proceedings of the 10th Annual CFD Workshop (ed. R. Williams), NASA MSFC, Huntsville, AL, April 28-30, 1992, NASA CP-3163, Part 1, pp. 467-486, 1992.
- [247] **Dulikravich, G. S., Martin, T. J.**
Determination of Void Shapes, Sizes and Locations Inside an Object With Known Surface Temperatures and Heat Fluxes, in Proceedings of the IUTAM Symposium on Inverse Problems in Engineering Mechanics (editors: M. Tanaka and H.D. Bui), Tokyo, Japan, May 11-15, 1992; also in Springer-Verlag, pp. 489-496, 1993.
- [248] **Dulikravich, G. S., Martin, T. J.**
Determination of the Proper Number, Locations, Sizes and Shapes of Superelliptic Coolant Flow Passages in Turbine Blades, in Proceedings of the International Symposium on Heat and Mass Transfer in Turbomachinery (ICHMT) (ed. R.J. Goldstein, A. Leontiev, and D. Metzger), Athens, Greece, August 24-28, 1992.
- [249] **Dulikravich, G. S., Martin, T. J.**
Design of Proper Super-Elliptic Coolant Passages in Coated Turbine Blades With Specified Temperatures and Heat Fluxes, AIAA paper 92-4714, 4th AIAA/AHS/ASCE Symposium on Multidisciplinary Analysis & Optimization, Cleveland, Ohio, Sept. 21-23, 1992; also in AIAA Journal of Thermophysics and Heat Transfer, Vol. 8, No. 2, pp. 288-294, April - June 1994.
- [250] **Dulikravich, G. S., Martin, T. J.**
Inverse Design of Super-Elliptic Cooling Passages in Coated Turbine Blade Airfoils, AIAA Journal of Thermophysics and Heat Transfer, Vol. 8, No. 2, April-June, 1994, pp. 288-294.
- [251] **Dulikravich, G. S., Chiang, T. L., Hayes, L. J.**
Inverse Design of Coolant Flow Passages in Ceramically Coated Scram-Jet Combustor Struts, ASME WAM'86, Anaheim, CA, December 1986, in Proceedings of Symposium on Numerical Methods in Heat Transfer (editors: M. Chen and K. Vafai), ASME HTD-Vol. 62, pp. 1-6, 1986.
- [252] **Martin, T. J., Dulikravich, G. S.**
Inverse Design of Threedimensional Shapes With Overspecified Thermal Boundary Conditions, Monograph on Inverse Problems in Mechanics, (editor: S. Kubo), Atlanta Technology Publications, Atlanta, GA, September 1993, pp. 128-140.
- [253] **Dulikravich, G. S., Martin, T. J.**
Three-Dimensional Coolant Passage Design for Specified Temperatures and Heat Fluxes, AIAA paper 94-0348, AIAA Aerospace Sciences Meeting, Reno, NV, January 10-13, 1994.
- [254] **Dulikravich, G. S., Martin, T. J.**
Geometrical Inverse Problems in Three-Dimensional Non-Linear Steady Heat Conduction, Engineering Analysis with Boundary Elements, Vol. 15, 1995, pp. 161-169.
- [255] **Dulikravich, G. S., Martin, T. J.**

- Inverse Shape and Boundary Condition Problems and Optimization in Heat Conduction, Chapter 10 in *Advances in Numerical Heat Transfer*, Vol. 1, (editors: W. J. Minkowycz and E. M. Sparrow), Taylor and Francis, 1996.
- [256] **Martin, T. J., Dulikravich, G. S.**
A Direct Approach to Finding Unknown Boundary Conditions in Steady Heat Conduction, Proc. of 5th Annual Thermal and Fluids Workshop, NASA CP-10122, NASA LeRC, Ohio, Aug. 16-20, 1993, pp. 137-149.
- [257] **Martin, T. J., Dulikravich, G. S.**
Inverse Determination of Temperatures and Heat Fluxes on Inaccessible Surfaces, Boundary Element Technology IX, Computational Mechanics Publications, Southampton (editors: C. A. Brebbia and A. Kassab), pp. 69-76, 1994.
- [258] **Dulikravich, G. S., Martin, T. J.**
Inverse Problems and Design in Heat Conduction, in *Proceedings of 2nd IUTAM International Symposium on Inverse Problems in Engineering Mechanics* (editors: H.D. Bui, M. Tanaka, M. Bonnet, H. Maigre, E. Luzzato, and M. Reynier), Paris, France, November 2-4, 1994, A. A. Balkema, Rotterdam, pp. 13-20, 1994.
- [259] **Martin, T. J., Dulikravich, G. S.**
Finding Unknown Surface Temperatures and Heat Fluxes in Steady Heat Conduction, in *Proceedings of 4th Intersociety Conference on Thermal Phenomena in Electronic Systems* (editors: A. Ortega and D. Agonafer), Washington, D. C., May 4-7, 1994, pp. 214-221; also in *IEEE Transactions on Components, Packaging and Manufacturing Technology (CPMT) - Part A*, Vol. 18, No. 3, Sept. 1995, pp. 540-545.
- [260] **Martin, T. J., Dulikravich, G. S.**
Inverse Determination of Boundary Conditions in Steady Heat Conduction With Heat Generation, in *Symposiums on Conjugate Heat Transfer, Inverse Problems, and Optimization, and Inverse Problems in Heat Transfer*, (editors: W. J. Bryan and J. V. Beck), ASME National Heat Transfer Conf., Portland, OR, Aug. 6-8, 1995, ASME HTD-Vol. 312, pp. 39-46.
- [261] **Martin, T. J., Dulikravich, G. S.**
Finding Temperatures and Heat Fluxes on Inaccessible Surfaces in 3-D Coated Rocket Nozzles, in *Proceedings of 1995 JANNAF (Joint Army-Navy-NASA-Air Force) Propulsion and Subcommittee Joint Meeting*, Tampa, FL, December 4-8, 1995.
- [262] **Martin, T. J., Dulikravich, G. S.**
Inverse Determination of Boundary Conditions and Sources in Steady Heat Conduction With Heat Generation, *ASME Journal of Heat Transfer*, Vol. 110, No. 3, Aug. 1996, pp. 546-554.
- [263] **Martin, T. J., Dulikravich, G. S.**
Determination of Temperatures and Heat Fluxes on Surfaces of Multidomain Three-Dimensional Electronic Components (with T. J. Martin), *Symposium on Application of CAE/CAD to Electronic Systems*, 1996 ASME Int. Mechanical Eng. Congress and Expo., Atlanta, GA, November 17-22, 1996.

-
- [264] **Martin, T. J., Dulikravich, G. S.**
Inverse Determination of Temperatures and Heat Fluxes on Surfaces of 3-D Objects, PanAmerican Congress of Applied Mechanics (PACAM-V), San Juan, Puerto Rico, Jan. 2-4, 1997.
- [265] **Press, W. H., Teukolsky, S. A., Vetterling, W. T., Flannery, B. P.**
Numerical Recipes in FORTRAN: The Art of Scientific Computing, Second Edition, Cambridge University Press, 1992.
- [266] **Choi, K. Y., Dulikravich, G. S.**
Acceleration of Iterative Algorithms on Highly Clustered Grids, AIAA Journal, Vol. 34, No. 4, April 1996, pp. 691-699.
- [267] **Foster, N. F., Dulikravich, G. S., Bowles, J.**
Three-Dimensional Aerodynamic Shape Optimization Using Genetic Evolution and Gradient Search Algorithms, AIAA paper 96-0555, AIAA Aerospace Sciences Meeting, Reno, NV, January 15-19, 1996.
- [268] **Farin, G.**
Curves and Surfaces for Computer Aided Geometric Design, Second Edition, Academic Press, Boston, 1990.

GEORGE S. DULIKRAVICH

INTERNATIONAL CENTRE FOR MECHANICAL SCIENCES

COURSES AND LECTURES - No. 366



NEW DESIGN CONCEPTS
FOR HIGH SPEED AIR TRANSPORT

EDITED BY

H. SOBIECZKY
DLR GERMAN AEROSPACE RESEARCH ESTABLISHMENT



SpringerWienNewYork

Chapter 14

Structural Inverse Design and Optimization

George S. Dulikravich

14.1 Introduction

Design of structural components for specified aerodynamic and dynamic loads can be achieved with an extensive use of a reliable and versatile stress-deformation prediction code and a constrained optimization code. The finite element method (FEM) is the favorite method for structural analysis because of its adaptability to complex 3-D structural configurations and the ability to easily account for a point-by-point variation of physical properties of the material. With the recent improvements in the sparse matrix solver algorithms, the finite element techniques are also becoming competitive with the finite difference techniques in terms of the computer memory and computing time requirements. For relatively smaller problems in elasticity, it is even more advantageous to use boundary element method (BEM) because it is faster and more reliable since it is non-iterative and it requires only surface discretization.

14.2 Optimization in Elasticity

The field of structural optimization [269]-[273] has a considerably longer history than aerodynamic shape optimization. Almost every textbook on optimization involves examples from linear elastostatics since these types of problems usually result in smooth convex function spaces that have continuous and finite sensitivity derivatives, making them ideal for gradient search optimization. As the geometric complexity and the diversity of materials involved in aerospace structures has increased, so has the demand for a more robust non-gradient search optimization

algorithms. Consequently, the majority of the present-day structural optimization is performed using different variations of genetic evolution search strategy and a hybrid gradient/genetic approach [274], [275]. This is quite evident when researching the literature dealing with structures made of composite materials, ceramically coated structures, and smart structures. In the case of smart structures, their "smart" attribute comes from the ability of such materials to respond with a desired degree of deformation to the applied pressure, thermal, electric or magnetic field. This automatic response can be very fast and can be used for active control of the structural shape in the regions exposed to the air flow. This influences the flow field, surface heat transfer and the aerodynamic forces acting on the structure.

Structural optimization is routinely used for the purpose of achieving aeroelastic tailoring. Since this requires specifications of a large number of constraints in the form of desired local deformations, this application can also be qualified as a *de facto* inverse structural shape design. The objective is to find the appropriate shape and orientation of each layer of a composite material to be formed so that the final object (for example, a helicopter rotor blade, an airplane wing, etc.) will have the most uniform stress distribution (thus minimum weight) and, when loaded, will deform into a desired form.

14.3 Inverse Problems in Elastostatics

An elastostatic problem is well-posed when the geometry of the general 3-D multiply-connected object is known and either displacement vectors, \mathbf{u} , or surface traction vectors, \mathbf{p} , are specified everywhere on the surface of the object. The elastostatic problem becomes ill-posed when either: a) a part of the object's geometry is not known, or b) when both \mathbf{u} and \mathbf{p} are unknown on certain parts of the surface. Both types of inverse problems can be solved only if additional information is provided. This information should be in the form of over-specified boundary conditions where both \mathbf{u} and \mathbf{p} are simultaneously provided at least on certain surfaces of the 3-D body.

The inverse determination of locations, sizes and shapes of unknown interior voids subject to overspecified stress-strain outer surface field is a common inverse design problem in elasticity [276]-[279]. The general approach is to formulate a cost function that measures a sum of least squares differences in the surface values of given and computed stresses or deformations for a guessed configuration of voids. This cost function is then minimized using any of the standard optimization algorithms by perturbing the number, sizes, shapes and locations of the guessed voids. Thus, the process is identical to the already described inverse design of coolant flow passages subject to over-specified surface thermal conditions [280]. It should be pointed out that this approach to inverse design of interior cavities and voids can generate interior configurations that are potentially non-unique.

14.4 Inverse Determination of Elastostatic Boundary Conditions

Another type of inverse problem in elastostatics is to deduce displacements and tractions on surfaces where such information is unknown or inaccessible, although the geometry of the entire 3-D configuration is given. It is often difficult and even impossible to place strain gauges and take measurements on a particular surface of a solid body either due to its small size or geometric inaccessibility or because of the severity of the environment on that surface. With our inverse method [281] these unknown elastostatics boundary values can be deduced from additional displacement and surface traction measurements made at a finite number of points within the solid or on some other surfaces of the solid. This approach is robust and fast since it is non-iterative. A similar inverse boundary value formulation has been shown [282] to compute meaningful and accurate thermal fields during a single analysis using a straight-forward modification to the BEM non-linear heat conduction analysis code. An example of the concept follows using the BEM.

In general elastostatics, we can write for any discretized boundary point "i" and for each direction "l" a boundary integral equation [283]

$$c_1^i \cdot u_1^i + \int_{\Gamma} u_k p_{1k}^* d\Gamma = \int_{\Gamma} p_k u_{1k}^* d\Gamma \quad (90)$$

where the asterisk designates fundamental solutions and the term c_1 is obtained with some special treatment of the surface integral on the left hand side [283]. Explicit calculation of this value can be obtained by augmenting the surface integral over the singularity that occurs when the integral includes the point "i". Fortunately, explicit calculation is not necessary as it can be obtained using the rigid body motions. The boundary G is discretized into N_{sp} surface panels connected between N nodes. The functions \mathbf{u} and \mathbf{p} are quadratically distributed over each panel with adjacent panels sharing nodes such that there will be twice as many boundary nodes as there are surface panels. A transformation from the global (x,y) coordinate system to a localized boundary fitted (ξ,η) coordinate system is required in order to numerically integrate each surface integral using Gaussian quadrature. The displacements and tractions are defined in terms of three nodal values and three quadratic interpolation functions. The whole set of boundary integral equations can be written in matrix form (omitting the body forces for simplicity) as

$$[H]\{U\} = [G]\{P\} \quad (91)$$

where the vectors $\{U\}$ and $\{P\}$ contain the nodal values of the displacement and traction vectors. Each entry in the $[H]$ and $[G]$ matrices is developed by properly summing the contributions from each numerically integrated surface integral. The surface tractions were allowed to be discontinuous between each neighboring surface panel to allow for proper corner treatment. The set of boundary integral equations will contain a total of $2N$ equations and $6N$ nodal values of displacements and surface tractions.

For a well-posed boundary value problem, at least one of the functions, \mathbf{u} or \mathbf{p} , will be known at each boundary node (either Dirichlet or von Neumann boundary condition) so that the equation set will be composed of $2N$ unknowns and $2N$ equations. Since there are two distinct traction vectors at corner nodes, the boundary conditions applied there should include either two tractions or one displacement and one traction. If only displacements are specified across a corner node, special treatment is required [283].

For an ill-posed boundary value problem, both \mathbf{u} and \mathbf{p} should be enforced simultaneously at certain boundary nodes, while either \mathbf{u} or \mathbf{p} should be enforced at some of the other boundary nodes, and nothing enforced at the remaining boundary nodes. For the simple example of a quadrilateral plate with four nodes, if at two boundary nodes both $\mathbf{u} = \mathbf{U}$ and $\mathbf{p} = \mathbf{P}$ are known, but at the other two nodes neither \mathbf{u} nor \mathbf{p} is known, the BEM equation set before any rearrangement appears as

$$\begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \\ h_{31} & h_{32} & h_{33} & h_{34} \\ h_{41} & h_{42} & h_{43} & h_{44} \end{bmatrix} \cdot \begin{bmatrix} U_1 \\ u_2 \\ U_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} g_{11} & g_{12} & g_{13} & g_{14} \\ g_{21} & g_{22} & g_{23} & g_{24} \\ g_{31} & g_{32} & g_{33} & g_{34} \\ g_{41} & g_{42} & g_{43} & g_{44} \end{bmatrix} \cdot \begin{bmatrix} P_1 \\ p_2 \\ P_3 \\ p_4 \end{bmatrix} \quad (92)$$

where each of the entries in the $[\mathbf{H}]$ and $[\mathbf{G}]$ matrices is a 3×3 submatrix in 3-D elastostatics. Straight-forward algebraic manipulation yields the following set

$$\begin{bmatrix} h_{11} & -g_{12} & h_{14} & -g_{14} \\ h_{22} & -g_{22} & h_{24} & -g_{24} \\ h_{32} & -g_{32} & h_{34} & -g_{34} \\ h_{42} & -g_{42} & h_{44} & -g_{44} \end{bmatrix} \cdot \begin{bmatrix} u_2 \\ p_2 \\ u_4 \\ p_4 \end{bmatrix} = \begin{bmatrix} -h_{11} & g_{11} & -h_{13} & g_{13} \\ -h_{21} & g_{21} & -h_{23} & g_{23} \\ -h_{31} & g_{31} & -h_{33} & g_{33} \\ -h_{41} & g_{41} & -h_{43} & g_{43} \end{bmatrix} \cdot \begin{bmatrix} U_1 \\ P_1 \\ P_3 \\ P_3 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} \quad (93)$$

The right-hand side vector $\{\mathbf{F}\}$ is known and the left-hand side remains in the form $[\mathbf{A}]\{\mathbf{X}\}$. Once the matrix $[\mathbf{A}]$ is solved, the entire \mathbf{u} and \mathbf{p} fields within the solid can be easily deduced from the integral formulation. The equation set $[\mathbf{A}]\{\mathbf{X}\} = \{\mathbf{F}\}$ resulting from our inverse boundary value formulation is highly singular and most standard matrix solvers will produce an incorrect solution. Singular Value Decomposition (SVD) methods [284] can be used to solve such problems accurately. The number of unknowns in the equation set need not be the same as the number of equations [284], so that virtually any combination of boundary conditions will yield at least some solution. Additional equations may be added to the equation set if \mathbf{u} measurements are known at locations within the solid in order to enhance the accuracy of the inverse steady boundary condition algorithm. A proper physical solution will be obtained if the number of equations equals or exceeds the number of unknowns. If the number of equations is less than the number of unknowns, the SVD method will find one solution, although it does not

necessarily have to be the proper solution from the physical point of view [285]. Thus, the more overspecified data is made available, the more accurate and unique the predicted boundary values will be.

The accuracy of the inverse boundary condition code was verified [281] on simple 2-D geometry consisting of an infinitely long thick-walled pipe subject to an internal gauge pressure. The shear modulus for this problem was $G = 8.0 \times 10^4 \text{ N/mm}^2$ and Poisson's ratio was $\nu = 0.25$. The radius of the inner surface of the pipe was 10 mm and the outer radius was 25 mm. The inner and outer boundaries were discretized with 12 quadratic panels each. The internal gauge pressure was specified to be $P_r = 100 \text{ N/mm}^2$, while the outer boundary was specified with a zero surface traction. Figure 88 depicts a contour plot of constant values of stress tensor components σ_{yy} that was computed using the analysis version of our second-order accurate BEM elastostatic code. The numerical results of this well-posed boundary value problem were then used as boundary conditions applied to the following two ill-posed problems.

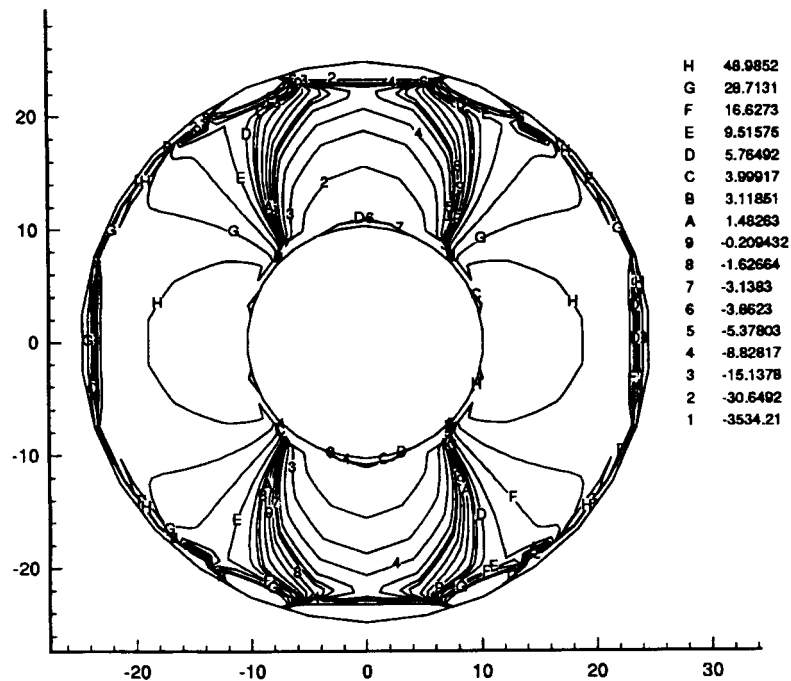


Figure 88 Contours of constant stress, σ_{yy} , from the wellposed analysis of an annular pressurized disk.

First, the displacement vectors computed on the inner circular boundary were applied as over-specified boundary conditions in addition to the surface tractions already enforced there. At the same time nothing was specified on the outer circular boundary. Figure 89 represents the

contour plot of σ_{yy} that was obtained with our inverse boundary value BEM code. This stress averaged a much larger error, about 3.0%, with some asymmetry in the stress field, when compared with the analysis results.

Next, the displacement vectors computed on the outer circular boundary by the well-posed numerical analysis were used to over-specify the outer circular boundary. At the same time nothing was specified on the inner circular boundary. Figure 90 the contour plot of σ_{yy} as computed by the inverse BEM technique. The predicted inner surface deformations were in error by less than 0.1%, while the predicted inner surface stresses averaged less than a 1.0% error as compared to the analysis results.

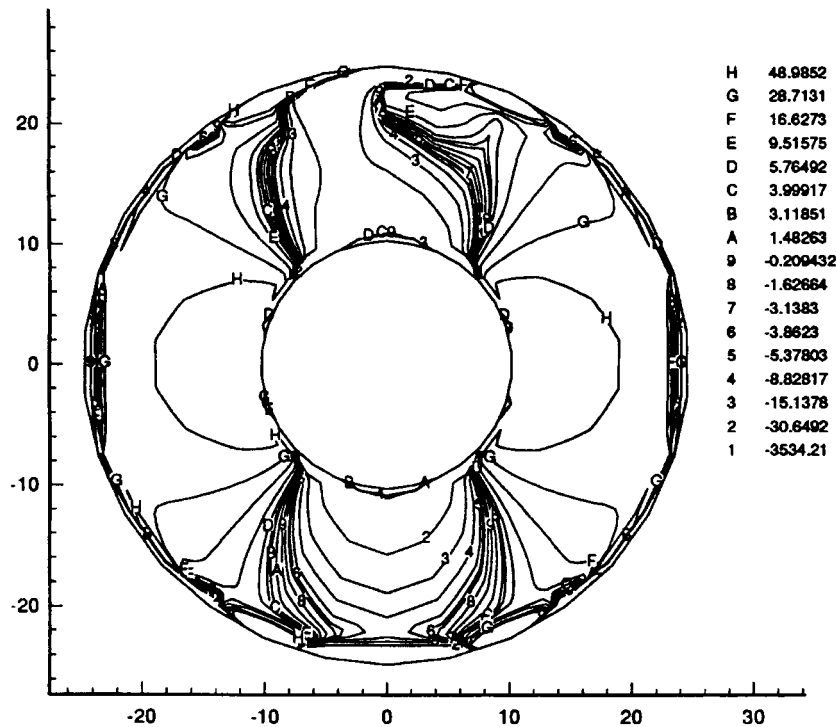


Figure 89 Contours of constant stress, σ_{yy} , from the ill-posed analysis of an annular pressurized disk: inner boundary over-specified.

It seems that an over-specified outer boundary produces a more accurate solution than one having an over-specified inner boundary. It was also shown that as the over-specified boundary area or the resolution in the applied boundary conditions was decreased, the amount of over-specified data also decreases, and thus the accuracy of the inverse boundary value technique deteriorates.

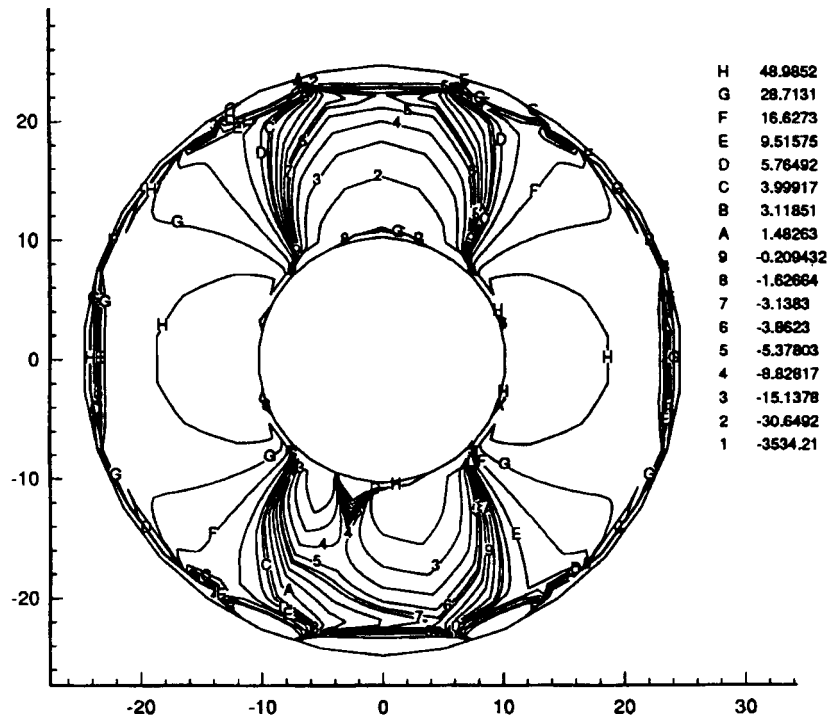


Figure 90 Contours of constant stress, σ_{yy} , from the ill-posed analysis of an annular pressurized disk: outer boundary over-specified.

14.5 Conclusions

Structural design of complex 3-D composite structures is possible only by using reliable optimization codes. For simpler configurations with smaller numbers of variables it is also possible to use inverse shape design methodologies based on over-specified surface tractions and deformations. Inverse shape design and inverse boundary condition determination can be performed using both BEM and finite element techniques. Hybrid optimization algorithms [286] that are based on genetic evolution, simulated annealing, fuzzy logic and neural networks and that involve manufacturing constraints [287]-[288] will be the candidates for active elastodynamic/aeroelastic control of smart structures.

14.6 References

- [269] **Zabinsky, Z. B.**
Global Optimization for Composite Structural Design, Proceedings of 35th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Material Conference, 1994, pp 1-7.
- [270] **Swanson, G. D., Ilcewicz, L. B., Walker, T. H., Graesser, D. L., Tuttle, M. E., Zabinsky, Z. B.**
Local Design Optimization for Composite Transport Fuselage Crown Panels, Proceedings of the Ninth DOD/NASA/FAA Conference on Fibrous Composites in Structural Design, November 1991, pp 1-20.
- [271] **Tuttle, M. E., Zabinsky, Z. B.**
Methodologies for Optimal Design of Composite Fuselage Crown Panels Proceedings of 35th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Material Conference, 1994, pp 1-12.
- [272] **Zabinsky, Z. B., Graesser, D., Tuttle, M., Kim, G.-I.**
Global Optimization of Composite Laminates Using Improved Hit-and-Run, Recent Advances in Global Optimization, Editors: Floudas & Pardalos, Princeton University Press, 1992, pp. 344-367.
- [273] **Graesser, D. L., Zabinsky, Z. B., Tuttle, M. E., Kim, G.-I.**
Optimal Design of a Composite Structure, Composite Structures, Vol. 24, 1993, pp. 273-281.
- [274] **Graesser, D. L., Zabinsky, Z. B., Tuttle, M. E., Kim, G.-I.**
Designing Laminated Composites Using Random Search Techniques, Composite Structures, Vol. 18, 1991, pp. 311-325.
- [275] **Sudipto, N., Zabinsky, Z. B., Tuttle, M. E.**
Optimal Design of Composites Using Mixed Discrete and Continuous Variables, Processing, Design and Performance of Composite Materials, ASME MD-Vol. 52, 1994, pp. 91-107.
- [276] **Bezzera, L. M., Saigal, S.**
A boundary element formulation for the inverse elastostatics problem (IESP) of flaw detection, International Journal for Numerical Methods in Engineering, Vol. 36, 1993, pp. 2189 -2202.
- [277] **Kassab, A. J., Moslehy, F. A., Daryapurkar, A.**
Detection of cavities by inverse elastostatics boundary element method: experimental results, in Boundary Element Technology IX, Computational Mechanics Publications, Southampton, Editors: C. A. Brebbia and A. Kassab, 1994, pp. 85-92.

-
- [278] **Tanaka, M., Yamagiwa, K.**
A Boundary Element Method for Some Inverse Problems in Elastodynamics, *Appl. Math. Modell.*, Vol. 13, 1989, pp. 307-312.
- [279] **Zabararas, N., Morellas, V., Schnur, D.**
A Spatially Regularized Solution of Inverse Elasticity Problems Using the Boundary Element Method, *Communications in Appl. Numer. Meth.*, Vol. 5, 1989, pp. 547-553.
- [280] **Dulikravich, G. S., Martin, T. J.**
Inverse design of super-elliptic cooling passages in coated turbine blade airfoils, *AIAA Journal of Thermophysics and Heat Transfer*, 8, No. 2, 288-294, April-June, 1994.
- [281] **Martin, T. J., Halderman, J. D., Dulikravich, G. S.**
An Inverse Method for Finding Unknown Surface Traction and Deformations in Elastostatics, *Computers and Structures*, Vol. 56, No. 5, Sept. 1995, pp. 825-836.
- [282] **Martin, T. J., Dulikravich, G. S.**
Inverse Determination of Boundary Conditions in Steady Heat Conduction with Heat Generation, *ASME Journal of Heat Transfer*, Vol. 118, No. 3, August 1996, pp. 546-554.
- [283] **Brebbia, C. A.**
The Boundary Element Method for Engineers, John Wiley & Sons, New York, 1978.
- [284] **Press, W. H., Teukolsky, S. A., Vetterling, W. T., Flannery, B. P.**
Numerical Recipes in FORTRAN, Second Edition, Cambridge University Press, 1992.
- [285] **Okuma, M., Kukil, S.**
Correction of Finite Element Models Using Experimental Modal Data for Vibration Analysis, Monograph on Inverse Problems in Mechanics, Editor: S. Kubo, Atlanta Technology Publications, Atlanta, GA, 204-211, September, 1993.
- [286] **Zabinsky, Z. B., Smith, R., McDonald, J., Romeijn, H., Kaufman, D.**
Improving Hit-and-Run for Global Optimization, *Journal of Global Optimization*, Vol. 3, 1993, pp. 171-192.
- [287] **Kristinsdottir, B., Zabinsky, Z. B., Tuttle, M. E., Csendes, T.**
Incorporating Manufacturing Tolerances in Near-Optimal Design of Composite Structures, *Engineering Optimization*, 1996, Vol. 26, pp. 1-23.
- [288] **Kristinsdottir, B. P., Zabinsky, Z. B.**
Including Manufacturing Tolerances in Composite Design, Proceedings of 35th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Material Conference, 1994, pp 1-10.

Chapter 15

Multidisciplinary Inverse Design and Optimization (MIDO)

George S. Dulikravich

15.1 Introduction

Designing for improved performance and life expectancy of high speed transport configurations is traditionally conducted by performing a repetitive sequence of uncoupled, single-discipline analyses involving flow field, temperature field, stress-strain field, structural dynamics, manufacturability tradeoffs and a large amount of personal designer's experience and intuition [289]-[295]. Since the entire aircraft system is seemingly highly coupled, it would be plausible that both analysis and design should be performed using an entirely new generation of computer codes that solve a huge system of partial differential equations governing aerodynamics, elastodynamics, heat transfer inside the structure, dynamics, manufacturing cost estimates, etc. simultaneously. This approach offers very stable computation since all boundary and interfacing conditions are incorporated implicitly. On the other hand, this approach might not be the most computationally economical since different subsystems (Navier-Stokes equations, elastodynamic equations, heat conduction equation, Maxwell's equations, etc.) that form such a complex mathematical system have vastly different eigenvalues and consequently converge at significantly different rates to a steady state solution. In addition, a rigorous analysis can show that even seemingly highly coupled systems are only loosely coupled and can be analyzed semi-sequentially [296]. Such a semi-sequential approach is presently used by most researchers and the industry since it can utilize most of the existing analysis and inverse design and optimization software as ready and interchangeable modules with minimum time invested in their modifications. Nevertheless, this approach is much more prone to global instability because of the often unknown and inadequately treated boundary and interface conditions.

In the remaining part of this article the focus will be on the computational grid, acceleration of iterative algorithms and parallelization and networking issues that are pertinent to the MIDO efforts.

15.2 Computational Grid Generation

It is well understood that automatic discretization (boundary-conforming computational grid generation) is the main bottleneck of the entire computational aerodynamics. Typically, it takes more time to generate an acceptable new grid for a new realistic 3-D aerodynamic configuration than it takes to predict the 3-D flow field around it. Since any shape inverse design and optimization effort implies repetitive generation of the 3-D grid, it is quite clear that developing a user-friendly, fast and reliable 3-D computational grid generation code is an extremely important issue. The existing automatic 3-D grid generation codes accept surface grid coordinates as an input and then generate the coordinates of field grid points. This approach is not reliable in the sense that there is no guarantee that some of the generated grid cells will not fold-over, or become needle-like, or that excessively large cells will not neighbor excessively-small grid cells, or that the grid will not be sufficiently clustered in the regions of interest, or that the grid will not become excessively non-orthogonal. Such problems cause significantly slower convergence of the analysis iterative algorithms, they create significantly larger computational errors, cause numerical instability, and often lead to outright divergence of the iterative process.

Any grid generation algorithm must be computationally efficient in terms of storage and execution time and able to accept even incomplete initial grids, that is, the grids that have only surface grid points resulting from a solid modeling software package (for example, CATIA). One technique capable of generating reliable 3-D grids in *an a posteriori* fashion is the grid optimization method which has been applied to 2-D [297] and 3-D [298] structured and block-structured [299] and to 2-D triangular unstructured grids including automatic solution-adaptive clustering [300]. This fully 3-D robust computational grid post-processing algorithm checks the entire original grid for possible negative Jacobians (existence of fold-over grid cells), untangles the grid lines, and optimizes the grid in a sense that it becomes maximally locally orthogonal and smoothly clustered in the regions of interest.

Notice that this grid generation procedure is especially suitable to the tightly coupled approached mentioned above. In this case, the grid from the flow field smoothly blends with the grid in the solid structure and the interior of the aircraft while clustering at the interface surfaces.

15.3 Convergence Acceleration and Reliability Enhancement

At present, it is possible to solve an impressive array of complex fluid dynamic problems by numerical techniques, but in many instances a minor change in the problem being computed can require an unwarranted amount of operator intervention, and/or an unexpectedly large increase in computer time. For example, a minor change in geometry or grid spacing sometimes has a major effect on convergence and can even lead to divergence, while coaxing a new problem through to a successfully converged solution can take many hours for even an experienced numerical analyst. Difficulties of this nature frequently dominate the cost of completing numerical solutions and can prevent the effective implementation of numerical techniques in design procedures.

Reduction of total computing time required by iterative algorithms for numerical integration of Navier-Stokes equations for 3-D, compressible, turbulent flows with heat transfer is an important aspect of making the existing and future analysis and inverse design codes widely acceptable as main components of optimization design tools. Reliability of analysis and design codes is an equally important item especially when varying input parameters over a wide range of values. Although a variety of methods have been tried, it remains one of the most challenging tasks to develop and extensively verify new concepts that will guarantee substantial reduction of computing time over a wide range of grid qualities (clustering, skewness, etc.), flow field parameters (Mach numbers, Reynolds numbers, etc.), types and sizes of systems of partial differential equations (elliptic, parabolic, hyperbolic, etc.). While a number of methods are capable of reducing the total number of iterations required to reach the converged solution, they require more time per iteration so that the effective reduction in the total computing time is often negligible.

The existing techniques for convergence acceleration are known to have certain drawbacks. Specifically, residual smoothing [301], although simple to implement, is a highly unreliable method, because it can offer either substantial reduction of number of iterations or it can abruptly diverge due to a poor choice of smoothing parameters [302]. Enthalpy damping [301] assumes constant total enthalpy which is incompatible with viscous flows including heat transfer. Multigridding in 3-D space is only marginally stable [303] when applied to non-smooth and non-orthogonal grids. GMRES method [304] based on conjugate gradients requires a large number of solutions to be stored per each cycle which is intractable in 3-D viscous flow computations. Power method [305] which is practically identical to the GNLMR method [306] works well with a multigrid code. Without multigridding, it is highly questionable if the power method would offer any acceleration when applied to a system of nonlinear partial differential equations. Despite the multigrid method's superior capability of effectively reducing low and high frequency errors yielding impressive convergence rates, its efficiency is significantly reduced on a highly-clustered grid [302], [303] and it is difficult to implement reliably. The preconditioning methods [307], although very powerful in alleviating the slow convergence associated with a stiff system for solving the low Mach number compressible flow equations, have not been shown to perform universally well on highly- clustered grids. Distributed Minimal Residual

(DMR) method [308], [309] in its present form has the same problem in transonic range. Numerous other methods have been published that are considerably more complex, while less reliable and effective.

The main commonality to all of these methods is that they all experience loss of their ability to reduce the computing time on highly-clustered non-orthogonal grids [307], [310] that are unavoidable for 3-D aerodynamic configurations and high Reynolds number turbulent flows.

Existing iterative algorithms are based on evaluating a correction, ΔQ^t , to each of the variables and then adding a certain fraction, $\omega \Delta Q^t$, of the correction to the present value of the vector of variables, Q^t , thus forming the next iterative estimates as

$$Q^{t+1} = Q^t + \omega \Delta Q^t \quad (94)$$

The optimal value of the relaxation factor, ω , can be determined from the condition that the future residual is minimized with respect to ω .

Instead, corrections from N consecutive iterations could be saved and added to the present value of the variable in a weighted fashion, where each of the consecutive iterative corrections is weighted by its own relaxation factor. This is the General Nonlinear Minimal Residual (GNLMR) acceleration method [306], summarized as

$$Q^{t+1} = Q^t + \omega^t \Delta Q^t + \omega^{t-1} \Delta Q^{t-1} + \dots + \omega^{t-N-1} \Delta Q^{t-N-1} \quad (95)$$

The optimal values of the relaxation factors, ω , can be determined from the condition that the future residual is minimized simultaneously with respect to each of the N values of ω 's.

Now, let us consider an arbitrary system of M partial differential equations. If the GNLMR method is applied to each of the M equations so that each equation has its own sequence of N relaxation factors premultiplying its own N consecutive corrections, this defines the Distributed Minimal Residual (DMR) acceleration method [308], [309] formulated as

$$Q_1^{t+1} = Q_1^t + \omega_1^t \Delta Q_1^t + \omega_1^{t-1} \Delta Q_1^{t-1} + \dots + \omega_1^{t-N-1} \Delta Q_1^{t-N-1} \quad (96)$$

$$Q_M^{t+1} = Q_M^t + \omega_M^t \Delta Q_M^t + \omega_M^{t-1} \Delta Q_M^{t-1} + \dots + \omega_M^{t-N-1} \Delta Q_M^{t-N-1} \quad (97)$$

The DMR is applied periodically where the number of iterations performed with the basic non-accelerated algorithm between two consecutive applications of the DMR is an input parameter. Here, ω 's are the iterative relaxation parameters (weight factors) to be calculated and optimized, ΔQ 's are the corrections computed with the non-accelerated iteration scheme, N denotes the total number of consecutive iteration steps combined when evaluating the optimum ω 's, and M stands for the total number of equations in the system that is being iteratively solved.

The DMR method calculates optimum ω 's to minimize the L-2 norm of the future residual of the system integrated over the entire domain, D . The present formulation of the DMR uses the same values of the $N \times M$ optimized relaxation parameters at every grid point, although different parts of the flow field converge at different rates.

An arbitrary system of partial differential equations governing an unsteady process can be written as $\mathbf{R} = -\frac{\partial \mathbf{Q}}{\partial t} = \mathbf{L}(\mathbf{Q})$, where \mathbf{Q} is the vector of solution variables, t is the physical time, \mathbf{L} is the differential operator and \mathbf{R} is the residual vector.

Sensitivity-Based Minimum Residual (SBMR) method [311], [312] uses the fact that the future residual at a grid point depends upon the changes in \mathbf{Q} at the neighboring grid points used in the local finite difference approximation. The sensitivities are determined by taking partial derivatives of the finite difference approximation of R_r ($r=1, \dots, r_{\max}$ where r_{\max} is the number of equations in the system) with respect to each component of Q_{ms} ($m = 1, \dots, M$ where M is the number of unknowns; $s = 1, \dots, S$ where S is the number of surrounding grid points directly involved in the local discretization scheme). This information is then utilized to effectively extrapolate \mathbf{Q} so as to minimize the future residual, \mathbf{R} . Nine grid points located at $(i-1, j-1; i-1, j; i-1, j+1; i, j-1; i, j; i, j+1; i+1, j-1; i+1, j; i+1, j+1)$ are used to formulate the global SBMR method for a two-dimensional problem when using central differencing compared to nineteen grid points for a three-dimensional case when using central differencing. This approach is different from the DMR [308], [309] method where the analytical form of \mathbf{R} was differentiated.

Suppose that we are performing an iterative solution of an arbitrary evolutionary system using an arbitrary iteration algorithm. Suppose we know the solution vectors \mathbf{Q}^t and \mathbf{Q}^{t+n} at iteration levels t and $t+n$, respectively. Here, n is the number of regular iterations performed by the original non-accelerated algorithm. Then, $\Delta \mathbf{Q}$ between the two iteration levels is defined as $\mathbf{Q}^{t+n} = \mathbf{Q}^t + \Delta \mathbf{Q}$ if no acceleration algorithm is used. Using the first two terms of a Taylor series expansion in the artificial (iterating) time direction, the residual for each of the equations in the system after n iterations is

$$R_r^{t+n} = R_r^t + \sum_m^M \sum_s^S \frac{\partial R_r^t}{\partial Q_{m,s}} \Delta Q_{m,s} \quad (98)$$

Notice that the total number of equations in the system is the same as the total number of unknown components of \mathbf{Q} , that is, $r_{\max} = M$. If we introduce convergence rate acceleration coefficients $\alpha_1, \alpha_2, \dots, \alpha_M$ multiplying corrections respectively, then each component, $Q_{m,s}^{(t+n)+1}$, of the future solution vector at the grid point, s , can be extrapolated as

$$Q_{m,s}^{(t+n)+1} = Q_{m,s}^t + \alpha_m \Delta Q_{m,s} \quad (99)$$

This can be applied at every grid point in the domain, but it results in a huge system of $i_{\max} \times j_{\max} \times k_{\max} \times M$ unknown acceleration coefficients, α . If each of the α 's is assumed to have the same value over the entire domain, D , the number of unknown α 's is reduced to M . This pragmatic approach is called the global SBMR method [311], [312]. It requires solution of a much smaller $M \times M$ matrix. The future residual at the iteration level $(t+n)+1$ can, therefore, be approximated by

$$R_r^{(t+n)+1} = R_r^t + \sum_m^M \left[\alpha_m \cdot \sum_s^S \frac{\partial R_r^t}{\partial Q_{m,s}} \Delta Q_{m,s} \right] \quad (100)$$

Subtracting (94) from (96) yields

$$R_r^{(t+n)+1} = R_r^t + \sum_m^M (\alpha - 1) \cdot a_{rm} \quad (101)$$

where

$$a_{rm} = \sum_s^S \frac{\partial R_r^t}{\partial Q_{m,s}} \Delta Q_{m,s} \quad (102)$$

The optimum α 's are determined such that the sum of the L-2 norm of the future residuals over the entire domain, D , will be minimized.

$$\sum_D \sum_r^{r_{\max}} \frac{\partial (R_r^{(t+n)+1})^2}{\partial \alpha_m} = 2 \sum_D \sum_r^{r_{\max}} \frac{\partial R_r^{(t+n)+1}}{\partial \alpha_m} R_r^{(t+n)+1} = 0 \quad (103)$$

for $m = 1, \dots, M$. With the help of (101) and (102), the system (103) becomes

$$\begin{aligned} \sum_D \sum_r^{r_{\max}} \left\{ R_r^{t+n} + \sum_m^M a_{rm} \cdot (\alpha - 1) a_{r1} \right\} &= 0 \\ \sum_D \sum_r^{r_{\max}} \left\{ R_r^{t+n} + \sum_m^M a_{rm} \cdot (\alpha - 1) a_{r2} \right\} &= 0 \\ &\dots \\ \sum_D \sum_r^{r_{\max}} \left\{ R_r^{t+n} + \sum_m^M a_{rm} \cdot (\alpha - 1) a_{rM} \right\} &= 0 \end{aligned} \quad (104)$$

In equation (101), the R's and a's are known from the preceding iteration levels. Since each α_m is assumed to have the same value over the entire computational domain, equation (104) gives a tractable system of M simultaneous algebraic equations for M optimum $\alpha_1, \alpha_2, \dots, \alpha_M$.

$$\begin{aligned} & \left[\sum_D \left(\sum_r^{rmax} a_{r1} a_{r1} \right) \right] \cdot (\alpha_1 - 1) + \dots + \left[\sum_D \left(\sum_r^{rmax} a_{rM} a_{r1} \right) \right] \cdot (\alpha_M - 1) = - \sum_D \left(\sum_r^{rmax} R_r^{i+n} a_{r1} \right) \\ & \left[\sum_D \left(\sum_r^{rmax} a_{r1} a_{r2} \right) \right] \cdot (\alpha_1 - 1) + \dots + \left[\sum_D \left(\sum_r^{rmax} a_{rM} a_{r2} \right) \right] \cdot (\alpha_M - 1) = - \sum_D \left(\sum_r^{rmax} R_r^{i+n} a_{r2} \right) \quad (105) \\ & \dots \\ & \left[\sum_D \left(\sum_r^{rmax} a_{r1} a_{rM} \right) \right] \cdot (\alpha_1 - 1) + \dots + \left[\sum_D \left(\sum_r^{rmax} a_{rM} a_{rM} \right) \right] \cdot (\alpha_M - 1) = - \sum_D \left(\sum_r^{rmax} R_r^{i+n} a_{rM} \right) \end{aligned}$$

For the general case of a system composed of M partial differential equations with M unknowns, the system (105) will become a full M x M symmetric matrix for M unknown optimum α 's.

It is plausible that for non-uniform computational grids and rapidly varying dependent variables, optimum α 's should not necessarily be the same over the whole computational domain. A modification of the SBMR method called Line Sensitivity-Based Minimal Residual (LSBMR) method was developed [311], [312] to allow α 's to have different values from one grid line to another. The resulting system has, for example, jmax x M unknown α 's, which is quite tractable, although it is more complex to implement than the SBMR method.

The performance of the SBMR and the LSBMR methods [311], [312] depends on how frequently these methods are applied during the basic iteration process and on the number of iterations performed with the basic iterative algorithm that are involved in the evaluation of the change of the solution vector. In the case of two-dimensional incompressible viscous flows without severe pressure gradient, the SBMR and LSBMR methods significantly accelerate the convergence of iterative procedure on clustered grids with the LSBMR method becoming more efficient as grids are becoming highly clustered. The SBMR and the LSBMR methods tested for a two-dimensional incompressible laminar flow maintain the fast convergence for highly non-orthogonal grids and for flows with closed and open flow separation. The SBMR method is capable of accelerating the convergence of inviscid, low Mach number, compressible flows where the system is very stiff. An Alternating Plane Sensitivity-Based Minimal Residual (APSBMR) method [312], a three-dimensional analogy of the LSBMR method, has been shown to successfully reduce the computational effort by 50% when solving a three-dimensional, laminar flow through a straight duct without flow separation.

The general formulation of these acceleration methods is applicable to any iteration scheme (explicit or implicit) as the basic iteration algorithm. Hence, it should be possible to apply these convergence acceleration methods in conjunction with the other iteration algorithms

and with other acceleration methods (preconditioning [307], multigriding [310], etc.) to explore the possibilities for a cumulative convergence acceleration effect.

In the case of optimization methods based on sensitivity analysis, the main difficulty is to make the evaluation of the derivatives less difficult and more reliable. A worthwhile effort is to research further on use of the automatic differentiation (ADIFOR) based on a chain-rule for evaluating the derivatives of the functions defined by flow analysis code with respect to its input variables [313]-[317]. This approach still needs to be made more computationally efficient. One obvious possibility is to utilize parallel computer architecture in the optimization process not only with the gradient search algorithms [318], [319], but especially with genetic evolution algorithms [320] where massively parallel approach should offer impressive reductions in computing time. The use of parallel computing will become unavoidable as the scope of optimization becomes multiobjective [321]-[325].

An equally worthwhile effort is to further research the "one shot" method that carefully combines the adjoint operator approach and the multigrid method [326]. This algorithm optimizes the control variables on coarse grids, thus eliminating costly repetitive flow analysis on fine grids during each optimization cycle. The entire shape optimization should be ideally accomplished in one application of the full multigrid flow solver [326].

Finally, it should be pointed out that although the exploratory efforts in applications of neural networks [327], [328] and virtual reality are still computationally inefficient, their time is coming inevitably as we attempt to optimize the complete real aircraft systems. Recent pioneering efforts [329] in coupling wind tunnel experimental measurements and a multiobjective hybrid genetic optimization algorithm indicate that the decades old dream of effectively and harmoniously utilizing both resources is soon to become a reality.

15.4 Parallelization and Networking Issues

The treatment of complex geometries has led us to adopt a multi-block grid made of several structures with overlapping or patched domains. The method of distributing domains over processors is important in that it leads to typical load balancing problems and synchronization of waiting time. Issues that need to be addressed are: flexibility in load balancing, reading and generating block data, reading and generating interface data, and updating block and interface data during communication. The advantage in using the multi-block approach is that one can have more than one block solver (for example, Euler, Navier-Stokes, etc.) in different blocks depending on the complexity of the flow field in a given block, thereby improving the overall efficiency of the algorithm. The most important parallelization issue for CFD applications is the way in which the computational domain is partitioned among a cluster of processors. Even a highly efficient parallel algorithm can give poor results for a poorly implemented domain decomposition. The domain decomposition technique has been successfully implemented on both SIMD and distributed memory MIMD computers. Algorithms like the "Masked Multi-block Algorithm" allow

for dynamically partitioning the domain depending on the distribution of load among processors. This eliminates the possibility of distributing each domain on a processor since in most cases domains will have irregular sizes. Other algorithms distribute separated planes of a 3-D computational domain between processors to synchronize time waiting. The measure of the efficiency of a parallel algorithm is given by the ratio of the computation time to the communication time for a particular application. High performance message passing can be achieved by "overlapping communication", performing assembly-coded gather-scatter operations. A machine like the CM-2 is a SIMD type machine where most of the parallelization is carried out by the compiler which is responsible for data layout. The Cray's parallel processing capabilities can be exploited by the use of "auto-tasking" wherein the user indicates points of potential parallelism in the implementation by the use of directives which instruct a pre-processor to reconfigure the source program in such a way so as to enable maximum speedup to be obtained.

An alternative to parallel hardware architecture is the poor man's machine or PVM (Parallel Virtual Machine) that can simulate a parallel machine across a host of serial machines. The programming model supported by PVM is distributed memory multi-processing with low level message passing. A PVM application essentially uses routines in the PVM to do message passing, process control and automatic data conversion. A special process runs on each node (each machine) of the virtual machine and provides communication support and process control. However since message passing is carried out on the Ethernet it is considerably slower than the Intel Interprocessor network. There is also an overhead on account differences between speeds of different machines that create load balancing problems.

15.5 Suggested General Structure of the MIDO Algorithms

Any MIDO algorithm stresses high transportability as well as extensive graphical user interaction and support. The product is a full featured, self-configuring MIDO package capable of utilizing a variable number of diverse CPU's and input/output devices that is valuable to research and industrial practical use. Massively parallel machines are the latest significant step in this technology. This expansion from a single CPU has greatly increased computational power; however, it has also increased the problem of transportability of the software. The rationale behind the advent of FORTRAN 90 was that users needed a standard method for the utilization of a variable number of CPU's. The CPU job tasking, while referred to by the programmer, is essentially handled by the FORTRAN 90 compiler. While FORTRAN 90 delivers a standard to the user of parallel machines, it in no way increases the user's interface with the code itself. The researcher or design engineer who simply provides a source code to the FORTRAN 90 compiler has no standard method for producing runtime graphics, real-time flow-field visualization, or other graphical user interactions. The only option for the user is to make calls to an external graphics package that is loaded on the particular system in use. By doing this, the generality of the code is lost. This problem is also true of other input/output devices.

This problem can be solved by giving the user real-time graphical visualization as an integral part of the MIDO software package that will allow researchers and design engineers to watch as they analyze and optimize their designs. Furthermore, criteria and boundary conditions are dynamic, allowing users to alter or update them as execution progresses. The problem of transportability is overcome because the package does not require a compiler since such MIDO package is pre-compiled and pre-linked. It is a stand-alone executable that detects and utilizes the host system. It is a self-contained, auto-configuring package that is to aerodynamic shape design and thermal design what I-DEAS and GENESIS are to structural design. An executable such as this MIDO package, can also be developed in lower level languages such as C/C++ or Assembler rather than less efficient high level languages such as FORTRAN. As well as being more efficient, low level languages operate in a regime that is close to the native language of the machine, allowing direct manipulation of the host system hardware. This direct access programming allows the final product to be highly modular and adaptable. These advantages could be utilized in the MIDO package with the end result that the user can choose the desired aspects of the package, and the algorithm will conform itself.

Because of the complex algorithms and high degree of programming skills required for this package, the development process can be divided in two stages.

During the first stage of such a MIDO package development, it can be written in FORTRAN 77 so that its legitimacy can be checked. Numerical solutions produced by this package can be checked against available analytical and experimental data to ensure that true physical phenomena are captured. Then, the MIDO package can be rewritten into low level languages as modules to a central-switching logic that is designed for a single CPU system. A full-featured interactive graphical environment can be developed for this system.

The second stage of the suggested MIDO package development can be centered on the expansion of the package to include support for a variable CPU system. Package development can culminate in its high modularity, full user interface environment (for a PC with MS-Windows-LINUX and workstations with UNIX and X-Windows), ability to change flow solvers in the environment, change boundary conditions, switch between analysis mode, inverse design mode, and optimization (based on user defined criteria) mode in the environment while developing a desired design.

Although most aerodynamic, heat transfer and elasticity analysis codes are written in FORTRAN while genetic algorithms and graphical interfaces are more efficient when coded in C++ language, the use of the PVM (Parallel Virtual Machine) or the new generation MPL (Message Passing Libraries) can make their simultaneous use possible. With the PVM or MPL, the hybrid GA optimizer can be compiled from a C++ source, while an analysis code can be compiled from a FORTRAN source. The message-passing between the two running codes is not extensive, yet it is independent of the language of the source code.

The efforts should be focused in developing a commercial quality MIDO package for research and direct industrial use. Algorithms should be modified to run in FORTRAN 90 on parallel CM-2, CM-5, IBM SP-2, CRAY-90, etc. or clusters of arbitrary PC's, workstations and

mainframes connected through a router thus stressing transportability to an arbitrary number of CPU's.

15.6 Conclusions

The Multidisciplinary Inverse Design and Optimization (MIDO) approach to a complete aircraft system design is already a reality in some of the leading companies. It has been taking two distinct paths where either a semi-sequential disciplinary optimization is performed or a simultaneous analysis and optimization of the entire system is performed. Both approaches have their own difficulties with computing time requirements and numerical stability. These issues will progressively be resolved by a better use of massively and distributed parallel computation and by the development of faster iterative algorithms for analysis and constrained optimization.

15.7 References

- [289] **Sobieski, J. S.**
Multidisciplinary Optimization for Engineering Systems: Achievements and Potential, NASA TM 101566, March 1989.
- [290] **Voigt, R. G.**
Requirements for Multidisciplinary Design of Aerospace Vehicles on High Performance Computers, ICASE Report No. 89-70, Sept. 1989.
- [291] **Dovi, A. R., Wrenn, G. A.**
Aircraft Design for Mission Performance Using Nonlinear Multiobjective Optimization Methods, J. of Aircraft, Vol. 27, No. 12, 1990, pp. 1043-1049.
- [292] **Savu, G., Trifu, O.**
On A Global Aerodynamic Optimization of a Civil Transport Aircraft", Proc. of 3rd Int. Conf. on Inverse Design Concepts and Optimiz. in Eng. Sci. (ICIDES-III), editor: G.S. Dulikravich, Washington, D.C., October 23-25, 1991.
- [293] **Fornasier, L.**
Numerical Optimization in Germany: A Non-Exhaustive Survey on Current Developments with Emphasis on Aeronautics, Proc. of 3rd Int. Conf. on Inverse Design Concepts and Optimiz. in Eng. Sci. (ICIDES-III), editor: Dulikravich, G. S., Washington, D. C., October 23-25, 1991.
- [294] **Sobieszczanski-Sobieski, J., Haftka, R. T.**
Multidisciplinary Aerospace Design Optimization: Survey of Recent Developments, AIAA paper 96-0711, Reno, NV, January 1995.

-
- [295] **Appa, K., Argyris, J.**
Nonlinear Multidisciplinary Design Optimization Using System Identification and Optimal Control Theory, AIAA-95-1481, New Orleans, LA, April 1995.
- [296] **Arian, E.**
Analysis of the Hessian for Aeroelastic Optimization, ICASE Report No. 95-84, Dec. 1995.
- [297] **Kennon, S. R., Dulikravich, G. S.**
A Posteriori Optimization of Computational Grids, AIAA Paper 85-0483, Reno, NV, 1985; also AIAA Journal, Vol. 24, No. 7, July 1986, pp. 1069-1073.
- [298] **Carcaillet, R., Kennon, S. R., Dulikravich, G. S.**
Optimization of Three-Dimensional Computational Grids, AIAA Paper 85-4087, Colorado Springs, CO, October 1985; also Journal of Aircraft, Vol. 23, No. 5, May 1986, pp. 415-421.
- [299] **Kennon, S. R., Dulikravich, G. S.**
Composite Computational Grid Generation Using Optimization, Proceedings of the First International Conf. on Numerical Grid Generation Comp. Fluid Dynamics, editor J. Haeuser, Landshut, W. Germany, July 14-17, 1986.
- [300] **Carcaillet, R., Dulikravich, G. S., Kennon, S. R.**
Generation of Solution Adaptive Computational Grids Using Optimization, Computer Meth. in Appl. Mech. and Eng., Vol. 57, Sept. 1986, pp. 279-295.
- [301] **Jameson, A., Schmidt, W., Turkel, E.**
Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes, AIAA Paper 81-1259, June 1981.
- [302] **Martinelli, L., Jameson, A., Grasso, F.**
A Multigrid Method for the Navier-Stokes Equations, AIAA Paper 86-0208, January 1986.
- [303] **Chima, R. V. and Turkel, E., Schaffer, S.**
Comparison of Three Explicit Multigrid Methods for the Euler and Navier-Stokes Equations, AIAA Paper 87-0602, January 1987.
- [304] **Saad, Y., Schultz, M.**
Conjugate Gradient-Like Algorithms for Solving Non-symmetric Linear Systems, Mathematics of Computation, Vol. 44, No. 170, 1985, pp. 417-424.
- [305] **Hafez, M., Parlette, E., Salas, M. D.**
Convergence Acceleration of Iterative Solutions of Euler Equations for Transonic Flow Computations, AIAA Paper 85-1641, July 1985.
- [306] **Huang, C. Y., Dulikravich, G. S.**
Fast Iterative Algorithms Based on Optimized Explicit Time-Stepping, Computer Methods in Applied Mechanics and Engineering, Vol. 63, August 1987, pp. 15-36.

-
- [307] **Turkel, E.**
Review of Preconditioning Methods for Fluid Dynamics, ICASE Report No. 92-47, NASA Langley Research Center, Hampton, VA, Sept. 1992.
- [308] **Lee, S., Dulikravich, G. S.**
Distributed Minimal Residual (DMR) Method for Acceleration of Iterative Algorithms, Computer Methods in Applied Mechanics and Engineering, Vol. 86, 1991, pp. 245-262.
- [309] **Lee, S., Dulikravich, G. S.**
Accelerated Computation of Viscous Flow with Heat Transfer, Numerical Heat Transfer: Fundamentals, Part B, Vol. 19, June 1991, pp. 223-241.
- [310] **Brandt, A.**
Multigrid Technique, Guide with Applications to Fluid Dynamics, GMD Studien 85, GMD-AIW, Postfach 1240, D-5205 St. Augustine, Germany, 1984.
- [311] **Choi, K. Y., Dulikravich, G. S.**
Sensitivity-Based Methods for Convergence Acceleration of Iterative Algorithms, Computer Methods in Applied Mechanics and Engineering, Vol. 123, Nos. 1-4, June 1995, pp. 161-172.
- [312] **Choi, K. Y., Dulikravich, G. S.**
Acceleration of Iterative Algorithms on Highly Clustered Grids, AIAA Journal, Vol. 34, No. 4, April 1996, pp. 691-699.
- [313] **Griewank, A., Corliss, G. F. (editors)**
Automatic Differentiation of Algorithms: Theory, Implementation and Application, SIAM, Philadelphia, PA, 1991.
- [314] **Bischof, C. H., Carle, A., Corliss, G., Griewank, A., Hovland, P.**
ADIFOR - Generating Derivative Code from Fortran Programs, Scient. Programming, Vol. 1, No. 1, 1992, pp. 1-29.
- [315] **Green, L. L., Newman, P. A., Haigler, K. J.**
Sensitivity Derivatives for Advanced CFD Algorithm and Viscous Modelling Parameters via Automatic Differentiation, AIAA paper 93-3321-CP, Orlando, FL, July 6-9, 1993.
- [316] **Hou, G. J.-W., Maroju, W., Taylor, A. C., Korivi, V., Newman, P. A.**
Transonic Turbulent Airfoil Design Optimization With Automatic Differentiation in Incremental Iterative Forms, AIAA CFD Conference Proceedings, San Diego, CA, June 1995, AIAA-95-1692-CP, pp. 512-526.
- [317] **Hovland, P., Altus, S., Kroo, I., Bischof, C.**
Using Automatic Differentiation With the Quasi-Procedural Method for Multidisciplinary Design Optimization, AIAA paper 96-0090, Reno, NV, January 1996.
- [318] **Cheung, S.**
Aerodynamic Design Parallel CFD and Optimization Routines, AIAA CFD Conference Proceedings, San Diego, CA, June 1995, AIAA-95-1748-CP, pp. 1180-1187.

-
- [319] **Jameson, A., Alonso, J. J.**
Automatic Aerodynamic Optimization on Distributed Memory Architectures, AIAA paper 96-0409, Reno, NV, January 1996.
- [320] **Poloni, C., Fearon, M., Ng, D.**
Parallelisation of Genetic Algorithm for Aerodynamic Design Optimisation, Proceedings of 2nd Conf. on Adaptive Computing in Engineering Design and Controls (ACEDC), Plymouth, UK, March 23-27, 1996.
- [321] **Osyczka, A.**
Multicriterion Optimization in Engineering, Ellis Hoewood, Ltd., England, 1984.
- [322] **Eschenauer, H., Koski, J., Osyczka, A.**
Multicriteria Design Optimization Procedures and Applications, Springer-Verlag, Berlin, 1990.
- [323] **Azarm, S., Sobieszczanski-Sobieski, J.**
Reduction Method With System Analysis for Multiobjective Optimization-Based Design, ICASE Report No. 93-22, April 1993.
- [324] **Poloni, C., Mosetti, G., Contessi, S.**
Multi-Objective Optimization by GAs: Application to System and Component Design, invited STS lecture at ECCOMAS 96 (European Community in Computational Methods in Applied Sciences), Paris, France, Sept. 9-13, 1996.
- [325] **Poloni, C.**
Hybrid Genetic Algorithm for Multiobjective Aerodynamic Optimisation, in: Genetic Algorithms in Engineering and Computer Science, pp. 397-415, John Wiley & Sons, England, December 1995, ISBN 0-471-95859-X, Contributed book edited by: G. Winter, J. Periaux, M. Galan and P. Cuesta.
- [326] **Ta'asan, S.**
Trends in Aerodynamics Design and Optimization: A Mathematical Viewpoint, AIAA CFD Conference Proceedings, San Diego, CA, June 1995, AIAA-95-1731-CP, pp. 961-970.
- [327] **Prasanth, K., Markin, R.E., Whitaker, K. W.**
Design of Thrust Vectoring Exhaust Nozzles for Real-Time Applications Using Neural Networks, Proc. of 3rd Int. Conf. on Inverse Design Concepts and Optimization in Engineering Sciences (ICIDES-III), editor: G. S. Dulikravich, Washington, D. C., October 23-25, 1991.
- [328] **Huang, S. Y., Miller, L. S., Steck, J. E.**
An Exploratory Application of Neural Networks to Airfoil Design, AIAA paper 94-0501, Reno, NV, January 1994.
- [329] **Poloni, C., Mosetti, G.**
Private Conversation, April 1996.