AIAA JOURNAL
Vol. 40, No. 4, April 2002

Technical Notes

TECHNICAL NOTES are short manuscripts describing new developments or important results of a preliminary nature. These Notes cannot exceed six manuscript pages and three figures; a page of text may be substituted for a figure and vice versa. After informal review by the editors, they may be published within a few months of the date of receipt. Style requirements are the same as for regular contributions (see inside back cover).

Inverse Transonic Airfoil Design Using Parallel Simulated Annealing and Computational Fluid Dynamics

X. Wang,* M. Damodaran,† and S. L. Lee‡
Nanyang Technological University,
Singapore 639798, Republic of Singapore

Introduction

PROBLEMS concerning the design of transonic airfoil shapes to satisfy desired aerodynamic behavior fall into two categories, namely, constrained and unconstrained design optimization problems. An example of an unconstrained design problem is the inverse problem of determining the airfoil shape that will support a specified (target) airfoil surface pressure distribution. This shape is determined by minimizing the discrepancy between the target and the evolving airfoil surface pressure distribution corresponding to the designed airfoil. Advances in computational fluid dynamics (CFD) techniques for solving the Navier–Stokes equations have strong implications for their use in evaluating objective functions.

One significant implication is the coupling of these techniques with appropriate numerical optimization algorithms to develop efficient and robust computational methods for the optimum design of fluid machinery and aerodynamic components as outlined by Labrujere and Slooff.[1]

The present study is focused on the application of parallel simulated annealing (PSA) for inverse aerodynamic design of transonic airfoil shapes in which a compressible Navier–Stokes flow solver is used to evaluate the objective function. Simulated annealing (SA) is a stochastic global optimization method that has been shown to be a useful tool for complex nonlinear optimization problems and that has been applied to a variety of problems. Aly et al.[2] have applied it for the design of an optimal aerodynamic shape of an axisymmetric forebody for minimum drag, where SA is used as the outer optimization loop and the flow solver is used to evaluate the objective function. SA has the advantage of yielding a global minimum and of overcoming the limitations of deterministic gradient-based search methods, such as that by Eyi et al.,[3] which have a tendency of getting trapped in local minima. A major drawback of SA is the extensive computational resources required to carry out the large number of evaluations of the objective function. Recently, Wang and Damodaran[4] have shown that a parallel version of the SA algorithm can significantly reduce the number of design iterations per processor and the wall-clock time. In this work, this aspect of PSA is explored for inverse design of transonic airfoil shapes.

Received 16 April 2001; revision received 17 September 2001; accepted for publication 12 December 2001.

*Research Fellow, Center for Advanced Numerical Engineering Simulations, Nanyang Avenue. Member AIAA.

†Associate Professor, School of Mechanical and Production Engineering and Singapore–Massachusetts Institute of Technology Alliance Faculty Fellow, Center for Advanced Numerical Engineering Simulations, Nanyang Avenue. Associate Fellow AIAA.

‡Research Associate, Center for Advanced Numerical Engineering Simulations; currently Military Officer, Ministry of Defence, Singapore 669638, Republic of Singapore.

Formulation of Design Optimization

The inverse design process is initiated by defining design variables controlling the airfoil shape. This is achieved by an approximate representation of the airfoil shape that will evolve during the design cycle. To maintain control over the size of the design space, a baseline airfoil is first chosen, and the steady flowfield around it for a specific Mach number and angle of attack is computed to start the design cycle iterations. The airfoil shape is updated by adding a smooth perturbation $\Delta y_k(x)$, defined as a linear combination of a family of smooth curves over the range $0 < x < 1$ as follows:

$$\Delta y_k(x) = \sum_{k=1}^{K} \delta_k f_k(x) \tag{1}$$

where x is the normalized chordwise position of the coordinates defining the airfoil contour and δ_k are the design variables that will change during the design iterations. K is the number of basis functions $f_k(x)$, which can either be patched polynomials or Hicks and Henne or Wagner functions, and so on, as outlined by Eyi et al.,[3] where the impact of the choice of these functions on the overall aerodynamic design effort has been assessed. The impact of the choice of these functions on inverse design using SA has been studied and reported by Lee.[5] Because the focus of this study is on the application of PSA for inverse design problems and not on the impact of basis functions, Wagner functions, which have been used in the sensitivity analysis from the point of view of gradient-based shape design by Eyi and Lee,[6] have been used for this study. Small side constraints to address the impact of nonlinearities of transonic flow, as well as to restrict the geometrical shape changes, are imposed so that the general shape that evolves during the design iteration will not lead to feasible shapes, which may cause the flow algorithm to fail. The leading edge and the trailing edges are fixed at the origin, a chord length away from the origin, respectively, along the x axis. This approach is taken to represent the airfoil shape and to restrict the design space, the size of which can be as large as the number of grid points defining the airfoil surface. For inverse design problems, a typical objective function $J(X)$ to be minimized is defined as follows:

$$J(X) = \left[\sum_{m=1}^{M} \left(C_{p_{t_m}} - C_{p_{b_m}} \right)^2 \Delta S_m \Big/ \sum_{m=1}^{M} \Delta S_m \right]^{\frac{1}{2}} \tag{2}$$

where $C_{p_{t_m}}$ is the pressure distribution of the target airfoil that is being sought, $C_{p_{b_m}}$ is the pressure distribution of the designed airfoil that evolves after each design iteration, ΔS_m is the length of the airfoil surface element, and the summation is done for M coordinate points defining the airfoil contour.

CFD Analysis for Evaluating Objective Function

The flow analysis module used to evaluate the objective function is based on the finite volume formulation of the unsteady Navier–Stokes equations for two-dimensional viscous flow. A cell-centered structured grid is used to discretize the physical domain into a large number of quadrilateral cells. The integral form of the conservation

laws is applied separately to each cell and results in a system of ordinary differential equations supplemented with adaptive artificial dissipationterms, which are a blend of second and fourth differences in the flow variables and explicitly added for shock-capturing and numerical stability purposes. The equations are advanced from a set of initial conditions to steady-state solutions for the desired flow conditions by a multistage time-stepping scheme. Several convergence acceleration strategies, such as local time stepping, implicit residual smoothing, and multigrid strategies, are used to accelerate the computation of steady-state solutions. Characteristic boundary conditions are imposed at the far-field boundaries, while a no-slip condition is imposed on the airfoil surface, which is also assumed to be adiabatic. A simple algebraic turbulence model is used to address the turbulence closure. Specific details of the flow algorithm used as the CFD analysis tool for computing the objective function can be found in work by Jameson et al.[7] and Martinelli et al.[8]

SA Algorithm and Parallel Implementation

SA is a concise optimization algorithm that is based on Monte Carlo techniques. It starts with a high temperature, which is lowered via temperature reduction factor, and a new sequence of moves is generated. The cycle is repeated until the design vector converges to the optimum value. The basic SA consists of several tuning parameters that can be adjusted to improve the performance of the optimization method. Additional tuning parameters can be manipulated to boost the efficiency of the SA as outlined by Lee and Damodaran.[9] The PSA is implemented by dividing a Markov main chain into subchains on multiple processors. The divisions algorithm of Aarts and Korst[10] uses a control parallelism strategy to implement PSA by dividing the effort of generating a Markov main chain used in sequential SA into p subchains with the same length, where p is the number of processors. Each processor works on the subchain and continues the generation of the subsequent subchain. Before updating the temperature at iteration level $k+1$ from iteration level k using an algebraic cooling schedule of the form $T^{k+1} = \gamma T^k$ (where γ is the annealing parameter that lies in the range from 0.05 to 0.3 in this study), a new subchain is obtained by choosing the best solution from the processors. The algorithm is further enhanced by incorporating the clustered method outlined by Bhandarkar and Machaka,[11] which functions efficiently in the lower temperature regions of the annealing process. When the acceptance ratio, that is, the ratio of the number of the accepted moves to the number of all of the search moves, is lower than a certain value (range of value chosen in this study is 0.4–0.6), the values of the objective function are gathered at the end of each search step, and the solution is chosen randomly from the accepted cost functions. Specific details of the implementation of PSA on an SGI Origin 2000 parallel computer can be found by Wang and Damodaran.[4]

Results and Discussion

Computational results are presented and discussed for the problem of inverse design of transonic airfoils. In this section, the flowfield and objective functions are computed on a body-conforming C grid around the airfoil consisting of 256×64 cells. The number of cells on the airfoil surface is five-eighths of the number of cells along the wraparound direction of the grid around the airfoil. The airfoil shape is parameterized using 14 basis functions, 7 of which define the upper surface of the airfoil and the remaining 7 define the lower surface of the airfoil, thereby creating a design space with 14 design variables for the parametric representation of the airfoil shape. The design variables are varied during the steps of the design cycle to estimate the perturbations in the design variables that will minimize the objective function and that enable the modification of the airfoil contour, as well as the computational grid. CFD analysis is used to calculate the airfoil surface pressure distribution for the baseline airfoil shape, as well as each subsequent shape of the designed airfoil. The perturbations on the airfoil surface as a result of the design iteration are used to alter the computational grid by shearing out the disturbances to the grid points between the airfoil surface and the far-field boundary. During each design cycle, the flow solution corresponding to the previous design flow analysis is used as the initial condition to reduce computational time for evaluating the objective function using CFD.

To demonstrate the application of stochastic optimization methods for inverse design of transonic airfoils, the target pressure distribution that is the desirable goal of the design is chosen to correspond with the pressure distribution around a Royal Aircraft Establishment RAE 2822 (target airfoil) immersed in transonic flow at a Mach number 0.730, with an angle of attack of 2.78 deg and Reynolds number $6.5E+06$. To initiate the design process to seek the airfoil shape that will correspond to the target pressure distribution, an initial guess for the shape of the baseline airfoil corresponding to a NACA-0012

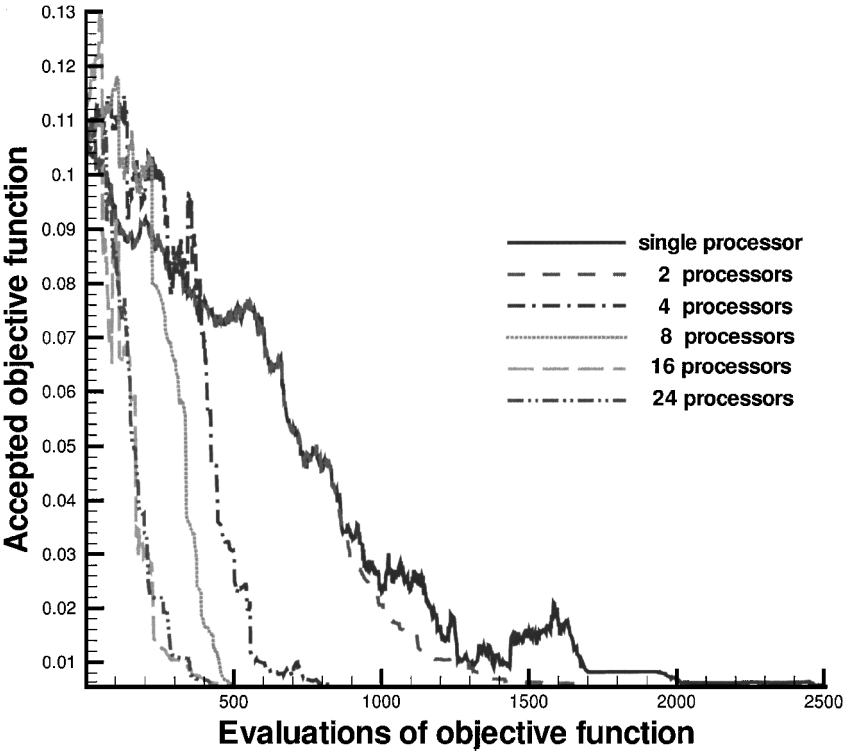


Fig. 1 Convergence of objective function vs number of design iterations incurred on each processor for shape design using PSA implemented on various combinations of multiple processors.

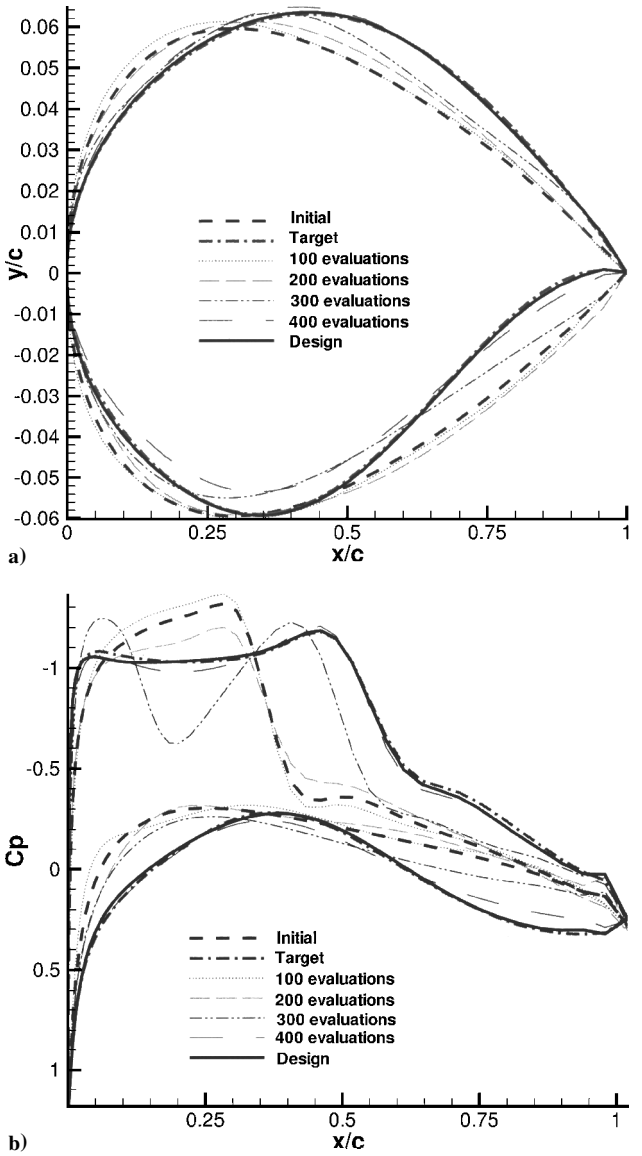


Fig. 2 Convergence of a) optimal design shape of airfoil from initial shape to target airfoil shape and b) optimal airfoil surface pressure from initial to target airfoil surface pressure vs design iterations incurred on each processor for shape design using PSA implemented on various combinations of multiple processors.

airfoil at the same flow conditions is made. The convergence criterion for the PSA optimization is set at a fixed value, and as the objective function reduces to 0.006, the process is terminated, and the final design variables are taken as the optimized results for the inverse design. The value of the objective function based on the baseline airfoil is 0.11420. The objective function reduced to a value of 0.006 in about 2500 iterations on a single processor. PSA is invoked on an SGI Origin 2000 system using a message passing interface (MPI) library to further reduce the number of objective function evaluations (or wall-clock time) on the multiple processors. Figure 1 shows the variation of the objective function with the number of function evaluations on different combinations of processors (only results from processor number zero are given). Figure 2a shows the evolution of the airfoil shape from the initial guess to the final design (corresponding to the case of implementing PSA on eight processors) by comparing the evolving airfoil shapes at the end of 100, 200, 300, and 400 design iterations with the target (designed) and initial airfoil shape. The corresponding evolution of the airfoil surface pressure coefficient C_p distribution at the specified design iteration levels is shown in Fig. 2b. Note that the number of function evaluations is reduced from 2521 to 1681 when 2 processors are used, to 824 when 4 processors are used, to 561 when 8 processors are used, to 448 when 16 processors are used, and to 421 when 24 processors are used. The corresponding speedups based on this number of function evaluations are 1.5, 3.0, 4.5, 5.6, and 6.0, respectively, and this is almost linear.

Grid-dependence studies can be quite prohibitive for this class of problems, and a limited attempt has been made by Lee,[5] who used a coarse grid of 128×48 cells. Finer grids resolve the computed flow features such as the shock structure better than coarser grids. The convergence of the objective function follows identical trends shown in Fig. 1 except that the cost of the computational analysis per design iteration on the finer grid is increased to about five times more than that on the coarse grid on a single processor. This can be significantly reduced by using a high-resolution CFD solver to preserve the accuracy of the numerical results even on coarse grids and by parallelizing the CFD solver. Details of the convergence characteristics of PSA and numerical accuracy have also been addressed in related aerodynamic shape design problems by Wang and Damodaran.[4] The application of a gradient-based method based on the least-squares minimization as outlined in Eyi and Lee[12] for the same problem was implemented and took about 150 function evaluations to reduce the objective function to the same level of reduction.

Conclusions

The design methodology developed in this work demonstrates that coupling a stochastic optimization method such as SA, especially its parallel implementation, with a compressible flow solver for the Navier–Stokes equations is a feasible approach and can be effectively applied for the design of transonic airfoil shapes. Future work will explore the application of this approach to constrained aerodynamic design problems in which optimal aerodynamic shapes are sought for minimizing drag or maximizing lift-to-drag ratio or multipoint airfoil design. It will also be applied to inverse shape design problems with a more arbitrarily chosen target pressure distribution (instead of the one used in this study), which in general may not be physically realizable. In this way, the ability of the PSA algorithm to home in on the global minima where multiple local minima are present can be investigated, and the advantages of PSA over gradient-based methods, which may get trapped in local minima, can be fully demonstrated. Because PSA is a stochastic algorithm whose efficiency depends on the various tuning parameters and the problem, future work will also focus on studying the reliability of PSA by extracting the statistics from the execution of PSA over a number of times and then assessing the percentage that gave the minima. The computation of inverse design using PSA demonstrates that it can be applied to airfoil design optimization efficiently on coarse-grained multiprocessor computer. PSA can also be easily implemented from its serial SA code on parallel computers using an MPI library. This work sets the foundation for applying more efficient flow analysis and high-performance parallel computing to reduce the computational time for developing efficient global optimization algorithms. These algorithms can be coupled with nonlinear CFD solvers for optimizing more complex design optimization problems.

References

[1] Labrujere, T. E., and Slooff, J. W., "Computational Methods for the Aerodynamic Design of Aircraft Components," *Annual Review of Fluid Mechanics*, Vol. 25, 1993, pp. 183–214.

[2] Aly, S., Ogot, M., and Pelz, R., "Stochastic Approach to Optimal Aerodynamic Shape Design," *Journal of Aircraft*, Vol. 33, No. 5, 1996, pp. 956–961.

[3] Eyi, S., Hager, J. O., and Lee, K. D., "Airfoil Design Optimization Using the Navier–Stokes Equations," *Journal of Optimization Theory and Applications*, Vol. 83, No. 3, 1994, pp. 447–461.

[4] Wang, X., and Damodaran, M., "Aerodynamic Shape Optimization Using Computational Fluid Dynamics and Parallel Simulated Annealing Algorithms," *AIAA Journal*, Vol. 39, No. 8, 2001, pp. 1500–1509.

[5] Lee, S. L., "Aerodynamic Shape Design of Transonic Airfoils Using Stochastic Optimisation Methods and Computational Fluid Dynamics," Ph.D. Dissertation, School of Mechanical and Production Engineering, Nanyang Technological Univ., Singapore, June 2000.

[6] Eyi, S., and Lee, K. D., "Effects of Sensitivity Derivatives on Aerodynamic Design Optimization," *Inverse Problems in Engineering*, Vol. 3, No. 2, 1996, pp. 213–225.

[7] Jameson, A., Schmidt, W., and Turkel, E., "Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge–Kutta Time Stepping Schemes," AIAA Paper 81-1259, June 1981.

[8]Martinelli, L., Jameson, A., and Grasso, F., "A Multigrid Method for Navier–Stokes Equations," AIAA Paper 86-0208, Jan. 1986.

[9]Lee, S. L., and Damodaran, M., "Aerodynamic Design of Transonic Airfoils Using Simulated Annealing and Navier–Stokes Equations," AIAA Paper 2000-0782, Jan. 2000.

[10]Aarts, E., and Korst, J., *Simulated Annealing and Boltzmann Machines, A Stochastic Approach to Combinatorial Optimization and Neural Computing*, Wiley, New York, 1989, p. 12.

[11]Bhandarkar, S. M., and Machaka, S., "Chromosome Reconstruction from Physical Maps Using a Cluster of Workstations," *Journal of Supercomputing*, Vol. 11, No. 1, 1997, pp. 61–87.

[12]Eyi, S., and Lee, K. D., "Aerodynamic Design via Optimization," *Journal of Aircraft*, Vol. 29, No. 6, 1992, pp. 1012–1019.

P. Givi
Associate Editor

Improved Formulation for Boundary-Layer-Type Flows

H. Dumitrescu* and V. Cardoş†
Romanian Academy, RO 70700 Bucharest, Romania

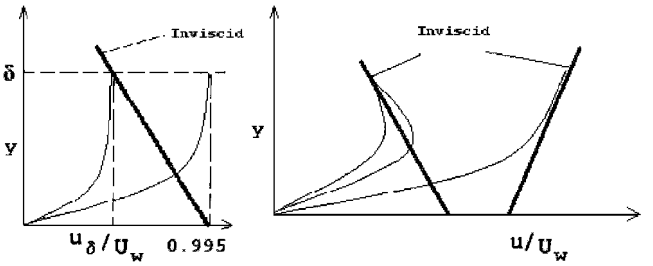
I. Introduction

THE matching of a viscous boundary-layer calculation with a inviscid flowfield solution is not without its difficulties. It is well known that when the pressure gradient is prescribed, the wall shear stress approaches zero in a singular fashion at separation, which invariably gives rise to problems on numerical convergence there.[1] This is particularly true for an inviscid shear layer near the edge of the boundary layer. Traditional boundary-layer methods neglect this layer, and even when it is considered, it is assumed that the flowfield properties approach the inviscid values at the boundary-layer edge with a zero slope. This assumption is particularly poor for lower Reynolds numbers and at separation approaching where the boundary layer is thickening. If the boundary-layer solution does not match the inviscid solution well, then surface properties such as skin-friction and heat transfer rates are not calculated accurately. Therefore, this Note develops a new method for better matching inviscid–boundary-layer flowfields.

II. New Boundary-Layer Model

The traditional method of calculating the boundary-layer properties assumes a distinct hierarchy between the viscous and inviscid flowfields. That is, an inviscid flowfield is first calculated neglecting the boundary layer, and the inviscid properties on the body surface are subsequently used as edge conditions for the boundary-layer solutions. However, for the boundary layers in adverse pressure gradients, a shear layer exists near the surface in the inviscid solution, and the inviscid properties at the edge of the boundary layer can be significantly different from those on the surface. For these flowfields, the boundary-layer solution will yield more accurate results if the inviscid properties at a distance from the surface are used as edge conditions. In this case, although some boundary-layer methods, take into account the inviscid shear layer, they assume the flowfield properties approach the inviscid values at the boundary-layer edge with a zero slope (Fig. 1a).

The method presented here uses a new form of the boundary-layer equations that matches exactly all of the properties except the normal velocity component (Fig. 1b). The normal velocity can also be matched by means of a usual iterative process.[2]



a) Classical methods **b) Present method**

Fig. 1 Inviscid and boundary-layer solution matching.

To illustrate the operation of this procedure more clearly, we consider the specific case of steady, two-dimensional, incompressible, laminar flow. The inviscid equations for this case are as follows.

The continuity equation:

$$\frac{\partial U}{\partial x} + \frac{\partial V}{\partial x} = 0 \tag{1}$$

The x-momentum equation:

$$U\frac{\partial U}{\partial x} + V\frac{\partial U}{\partial y} = -\frac{1}{\rho}\frac{\partial P}{\partial x} \tag{2}$$

The y-momentum equation:

$$U\frac{\partial V}{\partial x} + V\frac{\partial V}{\partial y} = -\frac{1}{\rho}\frac{\partial P}{\partial y} \tag{3}$$

They are subjected to the boundary conditions

$$y = 0, \qquad V(x, 0) = 0 \tag{4}$$

$$y \to \infty, \qquad U^2 + V^2 = U_\infty^2 + V_\infty^2 \tag{5}$$

Also, the viscous flowfield is described by the following classical boundary-layer equations.

The continuity equation:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial x} = 0 \tag{6}$$

The x-momentum equation:

$$u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} - \nu\frac{\partial^2 u}{\partial y^2} = -\frac{1}{\rho}\frac{\partial P}{\partial x} \tag{7}$$

The y-momentum equation:

$$0 = -\frac{1}{\rho}\frac{\partial P}{\partial y} \tag{8}$$

These have the boundary conditions

$$y = 0, \qquad u(x, 0) = v(x, 0) = 0 \tag{9}$$

$$y \to \infty, \qquad u = U(x, 0) = U_w \tag{10}$$

The mathematical difficulties of the problem are associated primarily with the mixed elliptic and parabolic types of the equations and the presence of strong interactions at separation. In this context, note that if the boundary-layer solution u cannot match exactly the inviscid solution U (that is, both in magnitude and slope) at the outer boundary, then a smooth solution is not possible at separation. An appropriate solution to separation requires the proper matching of the boundary-layer solution with the inviscid solution, that is,

$$u = U(x, y), \qquad v = V(x, y) \qquad \text{for} \qquad y \geq \delta \tag{11}$$

The boundary conditions given by Eq. (11) will match exactly the boundary-layer flowfield with the inviscid flowfield, as shown in Fig. 1b. Now, because $\partial v/\partial y$ is the highest y derivative on v, the boundary conditions imposed on v given by Eqs. (9) and (11) are overspecified, and this enables the solution of an additional unknown. That unknown introduced here is the inviscid transpiration

Received 19 April 2001; revision received 1 September 2001; accepted for publication 27 September 2001.

*Professor and Senior Researcher, Caius Iacob Institute of Applied Mathematics, P.O. Box 1-24.

†Senior Researcher, Caius Iacob Institute of Applied Mathematics, P.O. Box 1-24.