ENHANCEMENT OF A VISCOUS-INVISCID-INTERACTION AIRFOIL ANALYSIS CODE USING THE PARALLEL DIRECT SEARCH ALGORITHM

Andrew J. Meade, Jr.* and Michael Kokkolaras
Department of Mechanical Engineering
and Materials Science
MS 321
William Marsh Rice University
Houston, Texas, 77251-1892, USA
Phone: (713) 527-8101 ext. 3590
E-mail: meade@rice.edu

To appear in *Mathematical and Computer Modelling.*

---

* Recipient of correspondence, e-mail: meade@rice.edu
This work was supported under NASA grant number CRA2-35504.

ENHANCEMENT OF A VISCOUS-INVISCID-INTERACTION AIRFOIL ANALYSIS CODE USING THE PARALLEL DIRECT SEARCH ALGORITHM

Andrew J. Meade, Jr. and Michael Kokkolaras
Department of Mechanical Engineering
and Materials Science,
William Marsh Rice University
Houston, TX 77251-1892, USA

Key Words: boundary layer, viscous inviscid interaction, transpiration velocity, optimization, parallel processing, parallel direct search (PDS).

Abstract

The coupling mechanism for an existing viscous-inviscid-interaction (VII) code, developed for the analysis of two-dimensional, turbulent, attached flow around airfoils, is enhanced using the parallel direct search (PDS) optimization algorithm. It is demonstrated that this parallel processing implemented optimization scheme leads to faster convergence of the VII code, and therefore requires less computational time when the number of optimization (or design) variables is low, and a moderate number of processors are available. As the number of design variables increases more processors are required to maintain this advantage. Results are presented for the NACA-0012 and the RAE-2822 airfoils. The quality of the results obtained is satisfactory and confirms that the enhanced VII code can be an acceptable alternative to reduced Navier-Stokes solvers as an airfoil analysis tool.

ENHANCEMENT OF A VISCOUS-INVISCID-INTERACTION AIRFOIL ANALYSIS CODE USING THE PARALLEL DIRECT SEARCH ALGORITHM

Andrew J. Meade, Jr. * and Michael Kokkolaras †
Department of Mechanical Engineering
and Materials Science,
William Marsh Rice University
Houston, TX 77251-1892, USA

1 Introduction

The numerical simulation of a transonic flowfield for transonic airfoil design is considered a challenging task because it can involve the simultaneous modelling of subsonic and supersonic flow, shock waves, turbulence, turbulent transition, and flow separation. Two approaches exist in transonic airfoil flow analysis: one can either solve some form of the Navier-Stokes equations over the entire flow field or, alternatively, one can solve the boundary layer equations in the shear layer region of the flow, with either the potential or the Euler equations in the region of the flow approximated as inviscid. Both method have their advantages and disadvantages.

In the first approach, the direct numerical simulation (DNS) of the Navier-Stokes equations, which does not require a turbulence model, would be the ideal choice since it is arguably the most accurate existing method. Unfortunately, DNS solvers are in general impractical because of difficulties at high Reynolds numbers and large computational time and memory requirements [1]. Codes that solve high-fidelity solvable forms of the Navier-

*Associate Professor. Recipient of correspondence, e-mail: meade@rice.edu
†Research Assistant

Stokes equations (which include artificial turbulence models) have been developed and are continually improving in accuracy and speed at high Reynolds numbers. Navier-Stokes solver are especially advantageous when the flowfield experiences flow separation about the airfoil, which is likely to occur at off-design conditions. The second approach requires an interaction scheme between a viscous and inviscid solver so that smooth transition from one region of the flow to the other is guaranteed. These viscous-inviscid-interaction (VII) codes have the main advantage of providing accurate results for certain aerodynamic problems in less computational time than the high-fidelity Navier-Stokes methods [2], [3]. The primary disadvantage of VII codes is that they cannot model conditions with extensive flow separation. However, since on-design conditions usually include the reduction or elimination of flow separation, VII codes are especially useful in analyses near the final airfoil design.

While the continual progress and rapid growth of supercomputer speed, as well as the development of parallel processing technology, motivate research on full Navier-Stokes methods [4], [5], there is no indication that these same technological advances could not be implemented in an existing VII code. In this paper it will be shown that rather than directly applying parallel processing methods to the boundary layer and Euler codes, a parallel optimization technique known as the parallel direct search (PDS) algorithm [6] can be used to enhance the functionality of the interaction scheme. The speed up of this enhanced code combination can be controlled by the number of processors and design variables without adversely affecting the accuracy of the results.

2 VII Code

The original VII code consisted of:

- a two-dimensional Euler code, known as GAUSS2, that utilized a floating shock fitting technique, combined with an implicit upwind numerical scheme that allows accurate calculations on nonadaptive grids [7], [8], and

- a code for the solution of the two-dimensional, steady, compressible turbulent boundary-layer equations that utilizes the semi-discrete Galerkin method [9] - [11].

The classical interaction approach is to 1) obtain a solution to the inviscid flow approximation, 2) extract velocity and pressure from the inviscid solution and use them as external conditions in the viscous layer approximation, 3) extract displacement thickness from the viscous solution, and 4) use the displacement thickness to modify the original body geometry and obtain another estimate of the inviscid flow. These iterative cycles, depicted in Fig. 1, continue until the inviscid and viscous solutions are converged and compatible.

An alternative to adding the displacement thickness distribution to the original body thickness is to impose a transpiration velocity boundary condition at the body surface, also shown in Fig. 1. The transpiration velocity is an inviscid normal-velocity boundary condition which is imposed at the body surface to simulate the displacement of the inviscid flow by the viscous flow momentum defect [12]. Specifically, the transpiration velocity is a scalar quantity that is considered positive if injecting fluid into the airfoil flow or negative if removing fluid. An expression for the transpiration velocity can be obtained by integrating the difference between the inviscid and viscous continuity equations across the boundary layer while applying the Prandtl boundary-layer and uniform inviscid-flow assumptions [13]. The transpiration velocity distribution should result in an inviscid streamline that is coincident with the height of the effective body obtained by the displacement thickness approach. The advantages of the transpiration velocity approach are that the inviscid grid need not be regenerated after each viscous iteration and that the interaction always allows a smooth transition to separation [14].

3 The Coupling Mechanism

Clearly, the functionality of the coupling mechanism is very important to the overall performance of the code, i.e. its efficiency, accuracy and stability [3], [13]. As mentioned previously, the coupling mechanism between the viscous and the inviscid regions of the flow is based on the introduction of the transpiration velocity concept. The interested reader will find various VII combinations and/or different interaction mechanisms in references [15] and [16].

Though the transpiration velocity is a scalar quantity, its spatial distribution along the surface of the airfoil and wake can be formatted as a single column matrix or numerical vector. For the remainder of this paper the numerical vector constructed from the spatial distribution will be known as the transpiration distribution vector or transpiration vector. For our code the transpiration distribution vector, whose components are equal to the values of the transpiration velocity at the grid points on the airfoil and its wake, imposes a boundary condition on the Euler solver and is assumed to be initially zero. The first transpiration distribution vector with nonzero entries is obtained from the first execution of the boundary layer code. In the original version of the VII code, all subsequent transpiration distribution vectors undergo a strong under-relaxation (typically $\omega = 0.1$) according to the formula

$$\mathrm{v}_{\mathrm{t,rel}} = \omega {\mathrm{v}_{\mathrm{t}}}^{k} + (1-\omega){\mathrm{v}_{\mathrm{t}}}^{k-1} \tag{1}$$

where v_{t} and $\mathrm{v}_{\mathrm{t,rel}}$ represent the transpiration velocity and relaxed transpiration velocity, respectively. The variable ω is the relaxation parameter, and the superscripts indicate k-th and $(k-1)$-th VII global iteration [‡], respectively. Relaxation [17] is necessary, since in our case the direct use of the calculated transpiration velocity is a source of numerical instability that can prevent the convergence of the VII code.

Since the transpiration velocity in nonadaptive grid methods is analogous to the effective body in adaptive grid methods, convergence of the transpiration velocity corresponds to convergence of the effective body to its final geometric shape, and therefore yields smooth transition between the two different regions of the flow. Knowing that the coupling mechanism strongly affects the number of global VII iterations that are necessary for convergence,

[‡] For the sake of simplicity, the serial execution of the Euler and the boundary layer codes will be referred as a "VII global iteration" or a "black-box run".

our objective is the fast convergence of the transpiration velocity. The obvious aim is therefore to minimize the difference between the transpiration velocity of two consecutive VII global iterations, and to avoid relaxation.

3.1 Parallel Direct Search

PDS (Parallel Direct Search) is an algorithm for the solution of nonlinear optimization problems using direct search methods [18]. Direct search methods have the advantage of using only information from the C_0-continuous objective function and not requiring derivative calculations. The objective function in our case is given by

$$f = (\mathrm{v}^{\mathrm{output}} - \mathrm{v}^{\mathrm{input}})^{\mathrm{T}}(\mathrm{v}^{\mathrm{output}} - \mathrm{v}^{\mathrm{input}}) \tag{2}$$

where v is the transpiration distribution vector, consisting of P components, and the superscripts indicate the input and output transpiration vectors in a VII global iteration. The complexity of the transpiration velocity, which depends on a number of dependent and independent variables, encourages the use of PDS as a derivative-free optimization method. Furthermore, and most importantly, the required function evaluations are calculated in parallel with an almost linear speed-up [18], which means that the decrease of execution time is linearly proportional to the number of processors added. The numerical tests presented in this paper were executed on the Rice-CRPC-IBM/SP2, where PDS is executed in parallel by means of the Message Passing Interface (MPI) environment [19].

The strategy of applying PDS to the coupling mechanism is as follows: as with the original VII code, a zero transpiration vector is assumed in the first execution of the Euler and the boundary layer codes. The nonzero transpiration distribution vector obtained is relaxed with a coefficient of $\omega = 0.1$, and this relaxed transpiration vector forms the initial guess for the optimization problem to be solved by PDS. The PDS then creates a number of perturbed transpiration vectors [18], which in turn form the Euler code boundary conditions. The black-box code is executed for all of the resulting transpiration distribution vectors. The objective function is then evaluated, i.e. the difference between every input transpiration vector and its corresponding output vector. Finally, the input transpiration vector which provided the smallest difference is chosen as the starting vector for the next PDS iteration. Figure 2 illustrates the PDS iteration scheme; on the left side of each PDS iteration box is a transpiration distribution vector, in the box are m objective function evaluations for the perturbed transpiration distribution vectors, and on the right side of each box is the chosen transpiration vector, which now is the starting vector for the next PDS iteration.

There were 147 airfoil and wake grid points in the numerical tests presented in this paper resulting in a 147 component transpiration velocity distribution vector. Although the PDS algorithm performs best for a small number of design variables (less than 10), preliminary numerical tests showed the potential of the method, even for the 147 components. However, PDS requires at least $2N$ function evaluations per iteration, where N is the number of design variables. It was concluded that even advanced parallel machines with a large number of available processors would not be capable of achieving convergence in less CPU time than the original code. This conclusion holds true unless the number of components are reduced to a reasonable quantity without the loss of overall accuracy. As a result, the number of design variables were reduced by function approximation.

3.2 Reduction of Variables By Function Approximation

A one-dimensional, finite-element subroutine based on the least-squares method was developed, in order to represent the function defined by the components of the transpiration distribution vector.

$$\mathrm{v}(x) = \sum_{j=1}^{N} \Phi_j(x) c_j \tag{3}$$

Here, $\mathrm{v}(x)$ is the original transpiration distribution vector of length P, N is the number of coefficients used to approximate the transpiration distribution vector, $\Phi_j(x)$ are quadratic interpolation functions, and c_j are the respective transpiration velocity coefficients. Clearly we would like $N < P$. The number of coefficients used to approximate the transpiration velocity is chosen by the user.

For both the NACA-0012 and the RAE-2822 airfoils, there are 74 transpiration velocity grid points per airfoil side; 66 along the airfoil ($0 < x \leq 1$) and 7 within the wake ($1 < x \leq 2.2$). There is one common grid point for both sides at $x = 0$. Numerical tests were performed with 5 and 9 transpiration velocity coefficients per airfoil side, respectively.

The 5 coefficient version approximated the transpiration velocity distribution in the range $0 \leq x \leq 1$, and the 9 coefficient version those in the range $0 \leq x \leq 2.2$. The results obtained from the two different approximation orders are discussed in the following section. Having 10 (5 for each side) or 18 (9 for each side) design variables resulted in at least 20 and 36 function evaluations, respectively, per PDS iteration. Note that parallel machines with a number of available processors equal to or greater than the number of the required function evaluations, achieve convergence in a computational time much shorter than the one needed by the original serial code.

3.3 Convergence Criterion

Another important issue considered was the selection of the convergence criterion. In the original code, the global convergence criterion was the change in value of the aerodynamic coefficients before and after a global iteration, i.e. a black-box code execution. However, the convergence criterion of the PDS algorithm is based on the relative change of the design variables in every PDS iteration, not the aerodynamic coefficients. In addition, numerical instabilities in the Euler code were encountered during various numerical tests which primarily affected the values of the lift coefficient and the pressure contribution of the drag coefficient[§]. Fortunately, the viscous contribution to the drag coefficient (obtained from the boundary layer code) was found to be consistently stable. As a result, two convergence criteria were used for each test case: the evolution of the viscous contribution to the drag coefficient and/or the change of the objective function used by PDS.

[§]This phenomenon was also observed in the original version of the code, where it caused the values of the aerodynamic coefficients to oscillate and slowed convergence.

4 Results and Discussion

Two-dimensional, transonic, turbulent, and attached flows about a NACA-0012 airfoil and a supercritical RAE-2822 airfoil were simulated. Both airfoils have been previously studied numerically and experimentally. In each of the cases, a 161 $\times$ 33 C-grid was used by the Euler code. The grids for the NACA and RAE airfoils are illustrated in Fig. 3. The test cases were selected for comparison with results from references [20] and [21] and are presented in Table 1. The symbol α_{num} represents the numerical angle of attack, α_{exp} the corresponding experimental angle of attack, M_∞ the freestream Mach number, Re the chord based Reynolds number, x/c the location of the numerically tripped transition to turbulence (activation of numerical turbulence modelling), and ω represents the relaxation parameter in the original version of the code. The turbulence model utilized by the boundary layer code is the same described in reference [22] and applied in references [2] and [9]. The aerodynamic coefficients were calculated by means of the pressure coefficient distribution obtained from the Euler-solver and the friction coefficient distribution obtained from the boundary-layer solver.

4.1 NACA-0012 Airfoil

At first, 18 coefficients were used to approximate and reconstruct the 147 component transpiration distribution vector in a range containing both the airfoil and its wake. Evolution of the aerodynamic coefficients and objective function values is presented in Table 2. It can be seen that convergence based on the change of the objective function is achieved after 4 PDS iterations, satisfying a tolerance of 1×10^{-4} in absolute difference. The pressure contribution to the drag coefficient C_{D_p}, the friction contribution to the drag coefficient C_{D_v}, and their sum the total drag coefficient C_D are given in counts, where 1 count equals 1×10^{-4}. The aerodynamic coefficients strongly oscillate but converge after 10 PDS iterations. Due to the repeatedly observed numerical instabilities in the Euler code, and their influence on the aerodynamic coefficients for this case, we relied on the PDS objective function variation as the convergence criterion.

For the sake of comparison, and in order to draw conclusions about the effect of the wake transpiration velocity on the coupling mechanism, 10 coefficients were also used to approximate and reconstruct the transpiration distribution vector in a range containing only the airfoil. Convergence, again based on the change of the objective function, was achieved after 5 PDS iterations. The aerodynamic coefficients and objective function values after 5 and 10 PDS iterations are tabulated in Table 3. An overestimation of the lift coefficient is observed. However, the drag coefficient is closer to the experimental value, and the objective function values are an order of magnitude smaller, than those from the approximation using 18 coefficients.

In Fig. 4, the surface C_p distribution using 18 and 10 coefficients, respectively, are compared to those obtained from the original code and experimental data. The match is satisfactory, and the small scatter near the trailing edge of the airfoil when using 10 coefficients to approximate the transpiration velocity function has also been observed in reference [2]. One possible explanation for this phenomenon is the displacement action of the boundary layer [23]. Figure 5 illustrates the Mach isolines about the airfoil obtained from the original code (left) and the enhanced code when using 18 coefficients (right). This match is also satisfactory.

As shown, for the NACA-0012 airfoil test case convergence is achieved after 4 and 5 PDS iterations when using 10 and 18 coefficients, respectively. Keeping in mind that one PDS iteration using N coefficients will require at least $2N$ function evaluations, the number of processors available will determine whether or not the enhanced code will be faster than the original. The original code with the relaxation scheme required 10 global iterations for convergence, it is concluded then that a parallel machine, where the number of processors is at least as high as the number of the required function evaluations, will execute the enhanced code in half the time needed by the original code. Table 4 summarizes the above conclusion for all of the test cases and confirms the success of the enhanced code. The formula used to calculate the maximum speed-up is

$$S = \frac{i_{OR}}{\mathcal{I}(\frac{2N}{p}) i_{PDS}} \tag{4}$$

where S is the speed-up, function $\mathcal{I}$ truncates the argument to the higher integer value ($\mathcal{I}(2N/p) > 0$), i_{OR} is the number of global VII iterations required for convergence by the original code, i_{PDS} is the number of PDS iterations required for convergence by the enhanced code, N is the number of coefficients in the transpiration velocity approximation, and p is the number of processors available. It is clear that the ratio $2N/p$ is of primary importance in the speed-up.

4.2 RAE-2822 Airfoil

The RAE-2822 airfoil is a supercritical airfoil with a moderate amount of aft camber which can pose difficulties in achieving VII convergence. In addition, the test cases (summarized in Table 1) simulate critical transonic flow conditions. Therefore, these cases constitute a challenging validation criterion for the enhanced code.

For Case 2 of Table 1 convergence of the aerodynamic coefficients (with a tolerance of 0.001 in absolute difference) has been achieved after 5 PDS iterations using 18 coefficients and after 6 PDS iterations using 10 transpiration coefficients (once again neglecting wake effects), while the original code required 41 global iterations. As can be seen in Table 5, the aerodynamic coefficients are in satisfactory agreement with experimental values using 18 coefficients, while they diverge from the experimental values when using 10 coefficients. The conclusion drawn from this observation, is that the transpiration velocity values in the wake are nonnegligible for the supercritical airfoil in transonic flow conditions. In addition, it should be noted that C_{D_p}, especially when using 10 coefficients, is the primary contributor to errors in the aerodynamic coefficients.

It is interesting to observe, as Fig. 6 indicates, that while the aerodynamic coefficients do not match the experimental values exactly, the C_p distributions for 18 transpiration coefficients are in satisfactory agreement. The reason for the aerodynamic coefficient mismatch probably lies in the method utilized in their calculation; as mentioned previously, aerodynamic coefficients are calculated based on the pressure and friction coefficient distributions without taking full account of the wake effects. The effect of the wake is clearly nonnegligible for this test case, as illustrated in Fig. 6 for the calculation with 10 transpiration coefficients, which completely ignores the existence of the wake. The accuracy of this run is insufficient, especially around the suction peak.

The original code required 42 global iterations to achieve convergence for Case 3 of Table 1. The enhanced code required only 5 PDS iterations when using 18 coefficients and 6 PDS iterations when using 10 coefficients to approximate the transpiration velocity function. Table 6 summarizes the evolution of aerodynamic coefficient and objective function values to illustrate the previous statement.

In Fig. 7, the surface pressure coefficient distribution obtained from the enhanced code are compared to experimental values and those obtained from the original code (left) for Case 3. The match is satisfactory with the exception of the prediction on shock wave location. Figure 7 (right) presents results [2] taken at an angle of attack ($\alpha_{num} = 2.92$) slightly different from Case 3. The results of reference [2] are included to illustrate that in general, VII codes tend to underpredict the location of the shock wave while Navier-Stokes solvers tend to overpredict its location.

5 Conclusions

An existing VII code has been enhanced by optimizing the functionality of its coupling mechanism. The VII coupling mechanism, which is based on the concept of transpiration velocity, was optimized using the PDS algorithm, which offers the considerable advantage of parallel function evaluations without requiring information on derivatives. It has been shown that the enhanced code achieves convergence in less computational time than the original code. However, a strong dependency on the number of available processors exists; the ratio of function evaluations required to number of processors available has to be as small as possible for maximal speed-up. The quality of the results obtained from the enhanced code was maintained at the same level with the quality of the results from the original code. It was observed that for the NACA-0012 airfoil the speed-up was moderate compared to that of the two RAE-2822 airfoil test cases. Computational time has been kept low by reducing the design variables of the optimization problem by means of function approximation. It must be noted that the effects of the wake are not taken into account in the calculation of aerodynamic coefficients resulting in mismatches with the experimental values. However, considering the fidelity of the VII fluid model and the simplicity of the enhanced optimization algorithm, the results obtained can be characterized as highly satisfactory.

Acknowledgments

The authors would like to thank the Transonic Aerodynamics Branch of the NASA Langley Research Center and the Center for Research in Parallel Computations at Rice University for the use of their respective computing facilities. The authors would also like to thank Dr. Peter Hartwich and David Serafini for their assistance with the Euler solver and the MPI version of PDS, respectively. This work was supported under NASA grant number CRA2-35504.

REFERENCES

[1] M. M. Rai and P. Moin, "Direct Numerical Simulation of Transition and Turbulence in a Spatially Evolving Boundary Layer," AIAA paper 91-1607-CP, *Proceedings of the AIAA 10th Computational Fluid Dynamics Conference*, pp. 890-914, 1991.

[2] T. L. Holst, "Viscous Transonic Airfoil Workshop Compendium of Results," *Journal of Aircraft*, **25**, pp. 1073-1087, December 1988.

[3] R. C. Lock and B. R. Williams, "Viscous-Inviscid Interactions in External Aerodynamics," *Progress in Aerospace Sciences*, **24**, pp. 51-171, 1987.

[4] H. Li and C. J. Chang, "A Fast Scheme to Analyze 3D Disk Airflow on a Parallel Computer," *Parallel Computational Fluid Dynamics, Implementations and Results*, H. D. Simon, editor, MIT Press, 1992.

[5] D. Drikakis and E. Schreck, "Development of Parallel Implicit Navier-Stokes Solvers on MMD Multi-Processor Systems," AIAA 93-0062, presented at the 31st AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, January 1993.

[6] J. E. Dennis, Jr. and V. Torczon, "Direct Search Methods on Parallel Machines," *SIAM Journal of Optimization*, **1**, No. 4, pp.448-474, November 1991.

[7] P. M. Hartwich, "Fresh Look at Floating Shock Fitting," AIAA-90-0108, presented at the 28th AIAA Aerospace Sciences Meeting, Reno, Nevada, January 1990.

[8] P. M. Hartwich, "Split Coefficient Matrix (SCM) Method with Floating Shock Fitting For Transonic Airfoils," presented at the 12th International Conference on Numerical Methods in Fluid Dynamics, Oxford, U.K., July 1990

[9] B. A. Day, "Semi-Discrete Galerkin Solution of the Compressible Boundary-Layer Equations with Viscous-Inviscid Interaction," M.S.-thesis, Rice University, Houston, Texas, December 1992.

[10] A.J. Meade Jr., "Semi-Discrete Galerkin Modelling of Compressible Viscous Flow Past a Circular Cone at Incidence," Ph.D Thesis, University of California, Berkeley, 1989.

[11] A. J. Meade, Jr., "The Semi-Discrete Galerkin Finite Element Modelling of Compressible Viscous Flow Past an Airfoil," Final Technical Report, NAG-1-1196, December 1992.

[12] M.J. Lighthill, "On Displacement Thickness," *Journal of Fluid Mechanics*, **4**, pp. 383-392, 1958.

[13] W. R. Dalsem, J. L. Steger and K. V. Rao, "Some Experiences with the Viscous-Inviscid Interaction Approach," NASA TM 100015, pp. 1-44, 1987.

[14] J.C. Le Balleur, "Numerical Flow Calculation and Viscous-Inviscid Interaction Techniques," *Computation Methods in Viscous Flows Vol. 3*, pp. 419-450, Pineridge Press, 1984.

[15] U. R. Mueller and H. Henke, "Computation of Subsonic Viscous and Transonic Viscous-Inviscid Unsteady Flow," *Computers and Fluids*, **22**, pp. 649-661, 1993.

[16] I. H. Tuncer, J. A. Ekaterinaris and M. F. Platzer, "A Zonal Viscous-Inviscid Interaction Method for 2-D Compressible Flows," FED-**147**, pp.19-28, Computational Aero- and Hydro-Acoustics, ASME 1993.

[17] V. N. Vatsa, M. J. Werle and J. M. Verdon, "Viscid/Inviscid Interaction at Laminar and Turbulent Symmetric Trailing Edges," AIAA-82-0165, presented at the 20th AIAA Aerospace Sciences Meeting, Orlando, Florida, January 1982.

[18] V. Torczon, "PDS: Direct Search Methods for Unconstrained Optimization on Either Sequential or Parallel Machines," CRPC report, TR92206, March 1992.

[19] W. Gropp, E. Lusk, and A. Skjellum, *Using MPI,* MIT Press, 1994

[20] C. D. Harris, "Two-Dimensional Aerodynamic Characteristics of the NACA-0012 Airfoil in the Langley 8-Foot Transonic Pressure Tunnel," NASA Technical Memorandum 81927, April 1981.

[21] M. Drela, "Two-Dimensional Transonic Aerodynamic Design and Analysis Using the Euler Equations," Dissertation, MIT, December 1985.

[22] D. A. Johnson and L. S. King, "A New Turbulence Closure Model for Boundary Layer Flows with Strong Adverse Pressure Gradients and Separation," AIAA-84-0175, presented at the 22nd AIAA Aerospace Sciences Meeting, Reno, Nevada, January 1984.

[23] H. Schlichting, *Boundary-Layer Theory,* McGraw Hill Book Company, p. 22, 1979.

Table 1: Test cases.

Case	airfoil	α_{num}	α_{exp}	M_∞	Re	x/c	ω
1	NACA-0012	1.37	1.86	0.700	9×10^6	0.00	0.1
2	RAE-2822	1.90	2.40	0.676	5.7×10^6	0.11	0.05
3	RAE-2822	2.10	2.55	0.725	6.5×10^6	0.03	0.05

Table 2: Evolution of aerodynamic coefficients and objective function values for Case 1, using 18 coefficients in the approximation.

No. of PDS iterations	C_L	C_D	C_{D_p}	C_{D_v}	Obj. function value
1	0.247	59	-5	64	0.0055
2	0.263	61	-3	64	0.0033
3	0.272	62	-1	63	0.0024
4	0.251	88	25	63	0.0023
10	0.246	84	21	63	0.0019

Table 3: Aerodynamic coefficients and objective function values for Case 1, using 10 coefficients in the approximation.

No. of PDS iterations	C_L	C_D	Obj. function value
5	0.250	72	0.00099
10	0.259	70	0.00083
Experiment	0.241	77	
Original code	0.239	64	

Table 4: Speed-up comparisons

Case	No. of coefficients in the approximation	Original iterations required	PDS iterations required	Speed-up with p processors				
				4	8	16	32	64
1	18	10	4	0.28	0.5	0.83	1.25	2.5
1	10	10	5	0.4	0.67	1.0	2.0	2.0
2	18	41	5	0.91	1.64	2.73	4.1	8.2
2	10	41	6	1.36	2.27	3.41	6.83	6.83
3	18	42	5	0.93	1.68	2.8	4.2	8.4
3	10	42	6	1.4	2.33	3.5	7.0	7.0

Table 5: Aerodynamic coefficients and objective function values for Case 2.

No. of coefficients in the approximation	No. of PDS iterations	C_L	C_D	Obj. function value
18	5	0.577	54	0.00012
10	6	0.634	16	0.00009
Experiment		0.566	85	
Original code		0.554	72	

Table 6: Evolution of aerodynamic coefficients and objective function values for Case 3.

No. of coefficients in the approximation	No. of PDS iterations	C_L	C_D	C_{D_p}	C_{D_v}	Obj. function value
18	2	0.724	76	12	64	0.00032
18	3	0.723	80	17	63	0.00026
18	4	0.722	75	12	63	0.00015
18	5	0.720	76	13	63	0.00014
10	1	0.726	77	14	63	0.00024
10	2	0.731	76	13	63	0.00017
10	3	0.743	74	11	63	0.00013
10	4	0.738	73	10	63	0.00011
10	5	0.740	74	11	63	0.000109
10	6	0.742	75	12	63	0.000107
Experiment		0.658	107			
Original code		0.657	80			

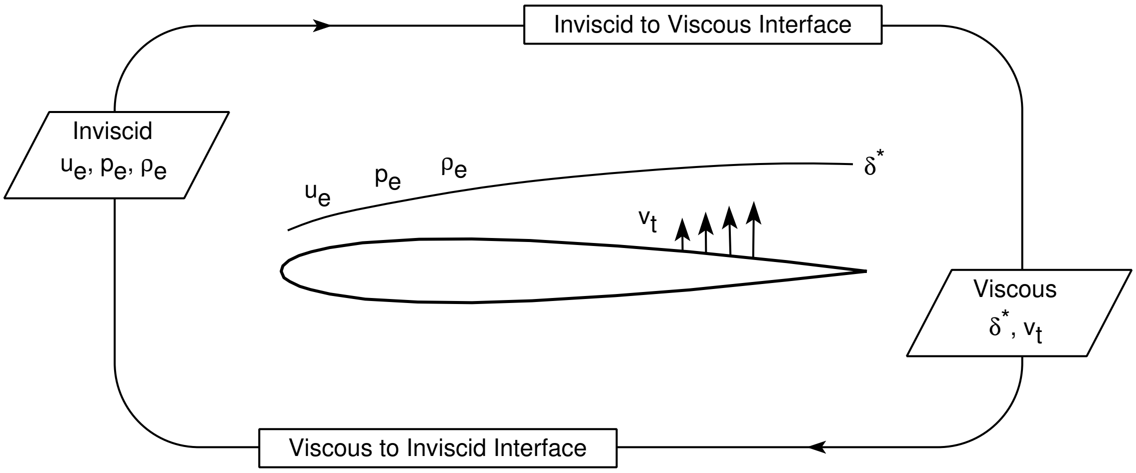

Figure 1: The VII global iteration scheme using either the displacement thickness (δ^*) or the transpiration velocity (v_t) approach.

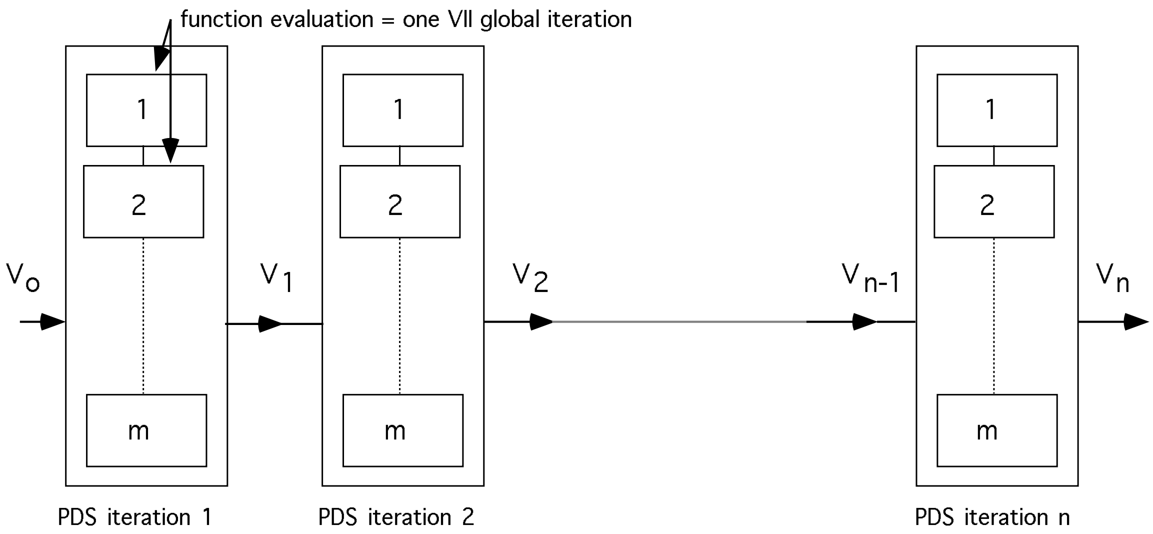

Figure 2: The PDS iteration scheme.

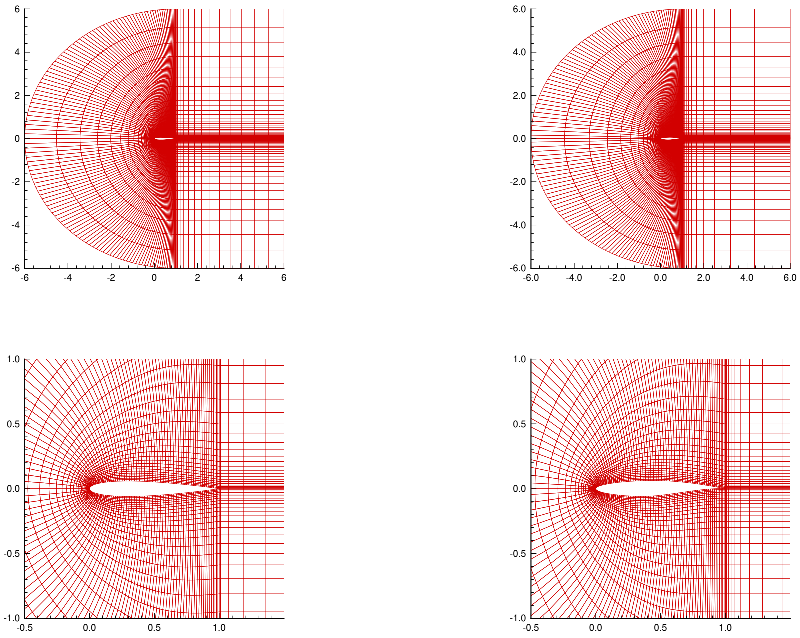

Figure 3: The NACA-0012 (left) and RAE-2822 (right) airfoil grids.

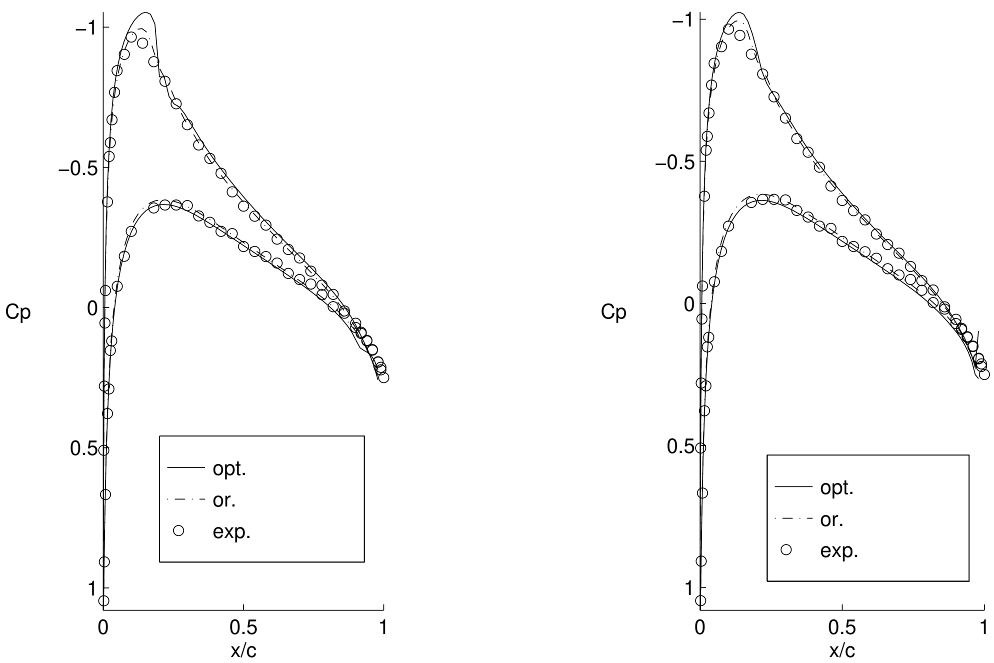

Figure 4: Surface pressure coefficient distribution over the NACA-0012 airfoil at $\alpha_{num} = 1.37$, $M_\infty = 0.700$ and Re $= 9\ \times 10^6$ (Case 1). Results obtained from the optimized (opt.) code using 18 (left) and 10 (right) transpiration velocity coefficients and the original (or.) code, compared to experimental data (exp.).

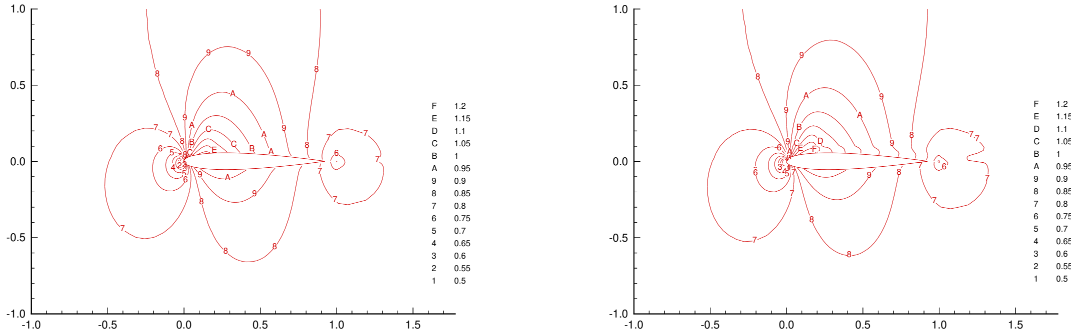

Figure 5: The Mach isolines about the NACA-0012 airfoil obtained from the original code (left) and the enhanced code (right) using 18 transpiration velocity coefficients.

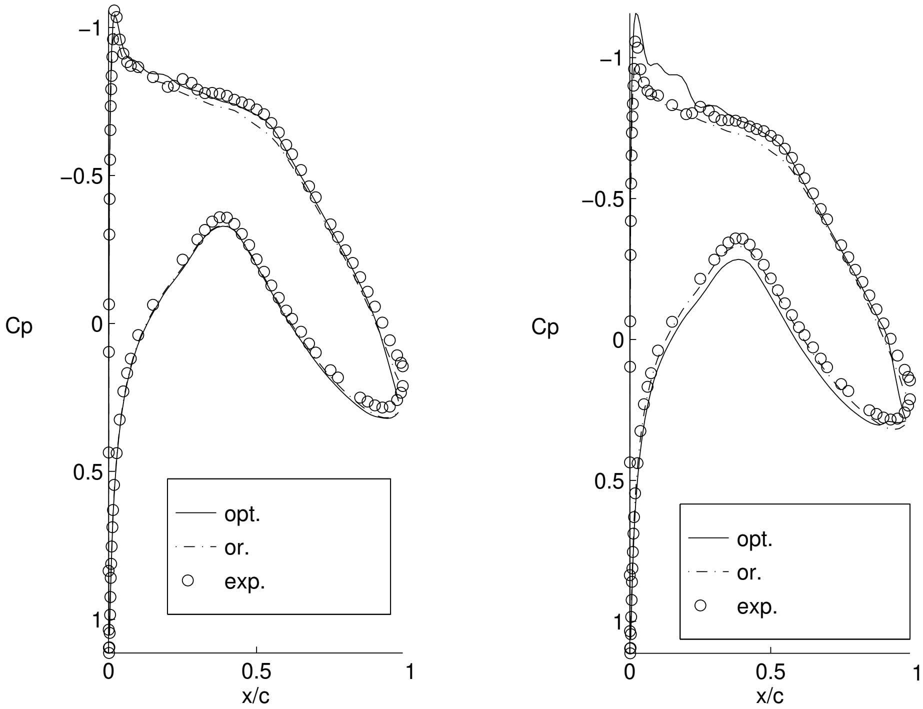

Figure 6: Surface pressure coefficient distribution about the RAE-2822 airfoil at $\alpha_{num} = 1.90$, $M_\infty = 0.676$ and Re $= 5.7 \times 10^6$ (Case 2). Results obtained from the optimized (opt.) code using 18 (left) and 10 (right) transpiration velocity coefficients and the original (or.) code, compared to experimental data (exp.).

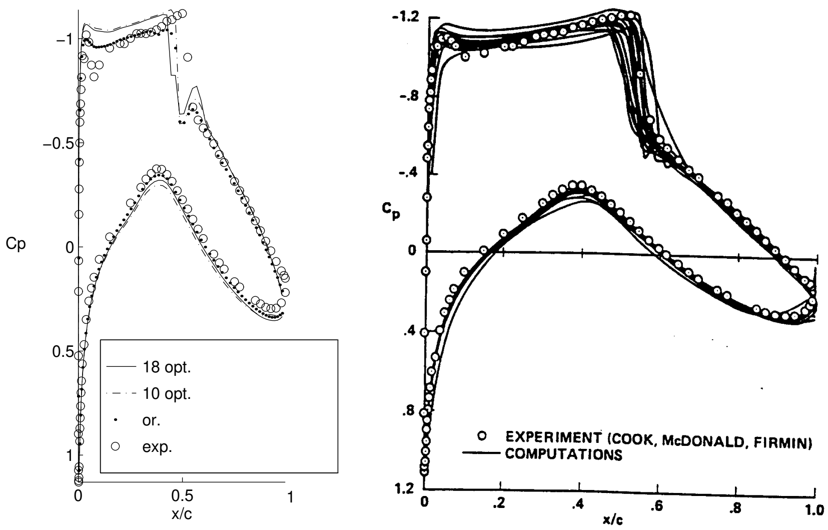

Figure 7: Surface pressure coefficient distribution about the RAE-2822 airfoil at $\alpha_{num} = 2.10$, $M_\infty = 0.725$ and $\mathrm{Re} = 6.5 \times 10^6$ (Case 3). Results (left) obtained from the optimized (opt.) code using 18 and 10 transpiration velocity coefficients and the original (or.) code, compared to experimental data (exp.). Results for $\alpha_{num} = 2.92$, $M_\infty = 0.725$ and $\mathrm{Re} = 6.5 \times 10^6$ (right) taken from Holst [2].