JOURNAL OF AIRCRAFT
Vol. 42, No. 2, March–April 2005

Engineering Notes

ENGINEERING NOTES are short manuscripts describing new developments or important results of a preliminary nature. These Notes cannot exceed 6 manuscript pages and 3 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 Design of Transonic Airfoils Using Parallel Simultaneous Perturbation Stochastic Approximation

X. Q. Xing* and M. Damodaran†
Nanyang Technological University,
Singapore 639798, Republic of Singapore

Introduction

TRANSONIC airfoil design problems are typical design problems based on computational-fluid-dynamics (CFD) simulation, which is used to investigate the flowfield characteristics to measure the objective function values. It is well known that to simulate the whole flowfield characteristics using CFD is very time consuming. The expensive evaluations make it is crucial for developing efficient optimization methods to deal with the CFD-based problems. Stochastic methods such as simultaneous perturbation stochastic approximation (SPSA) reported in Xing and Damodaran,[1] simulated annealing (SA) reported in Wang and Damodaran,[2] and genetic algorithm (GA) reported in Quagliarella and Cioppa,[3] and so on have been successfully applied to transonic airfoil design problems. These methods have the advantage of yielding a global minimum and overcoming the limitations of deterministic gradient-based search methods, such as in Xing and Damodaran[1] and Eyi et al.,[4] which have a tendency of getting trapped in local minima. A major drawback of SPSA, SA, and GA methods in these problems is that these methods need thousands of function evaluations to get the optima.

Parallel methods are appealing alternative approaches for reducing computational time. Wang and Damodaran[2] have shown that a parallel version of simulated annealing algorithm can significantly reduce the number of design iterations running on each processor and the wall-clock time. Applications of parallel GA to improve the computational efficiency of aerodynamic design problems have been reported in Vicini and Quagliarella[5] and so on. In this work, a parallel SPSA method is explored for the inverse design of a transonic airfoil shape to assess and compare its performance with the parallel SA reported in Wang and Damodaran.[2]

Inverse Airfoil Shape Design Problem

The goal of inverse design problems is to determine the airfoil shape that will support a 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. A baseline NACA 0012 airfoil is chosen, and a steady flowfield around it at a Mach number of 0.73, Reynolds number 6.5×10^6, and angle of attack of 2.78 deg is computed for starting the design cycle iterations. The airfoil shape is updated by adding smooth perturbations $\Delta y_k(x)$ 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, K is the number of basis functions and has a value of 14 in this study, and δ_k is design variable, where seven of them are for upper surface and seven for lower surface. The impact of the choice of basis functions on inverse design using simulated annealing has been studied and reported in Lee.[6] For the same example studied in Wang and Damodaran,[2] Wagner functions outlined in Ramamoorthy and Padmavathi[7] are selected as the basis function:

$$f_1(x) = (\theta + \sin\theta)/\pi - \sin^2(\theta/2)$$

$$f_k(x) = \sin k\pi/k\pi + [\sin(k-1)\theta]/\pi \qquad \text{for} \qquad k > 1 \tag{2}$$

where $\theta = 2\sin^{-1}(\sqrt{x})$. Figure 1 shows the 14 shape functions $f_k(x)$.

For inverse design problems a typical objective function $F(X)$ to be minimized is defined as follows:

$$F(X) = \left[\frac{\sum_{m=1}^{M}\left(Cp_{t_m} - Cp_{b_m}\right)^2 \Delta S_m}{\sum_{m=1}^{M} \Delta S_m}\right]^{\frac{1}{2}} \tag{3}$$

where Cp_{t_m} is the pressure distribution of the target airfoil, which for this example is a RAE 2822 airfoil at the same Mach number and angle of attack to that of the baseline airfoil; Cp_{b_m} is the pressure

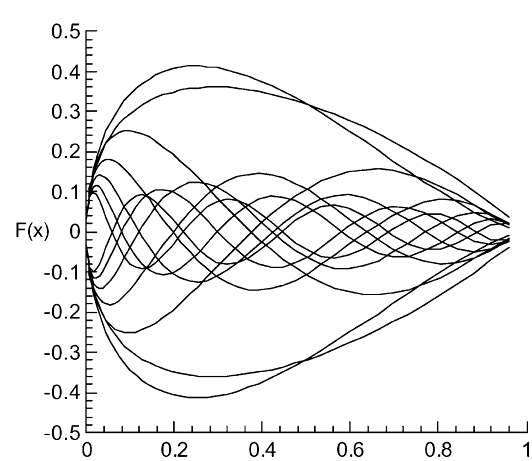


Fig. 1 Plot of Wagner functions.

Received 14 May 2004; revision received 18 August 2004; accepted for publication 24 August 2004.

*Research Fellow, Singapore-MIT Alliance, School of Mechanical and Production Engineering, 50 Nanyang Avenue; xqxing@ntu.edu.sg.

†Associate Professor, Singapore-MIT Alliance Fellow, School of Mechanical and Production Engineering, 50 Nanyang Avenue; mdamodaran@ntu.edu.sg. Associate Fellow AIAA.

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 the M coordinate points defining the airfoil contour.

The flow analysis module used for evaluating the objective function is based on the finite volume formulation of the unsteady Navier–Stokes equations for two-dimensional viscous flow. 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 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 Jameson and coworkers[8,9]

Parallel Implementation of SPSA

This study focuses on parallel simultaneous perturbation stochastic approximation, details of which can be found in Spall.[10] Parallel SPSA is implemented by carrying out the search process on multiple processors (say, p). Given the initial design variables, each processor performs the same SPSA search process with its own random perturbations to the design variables. Therefore, each processor receives the objective function value of itself and the corresponding design variables after a design cycle. Then, p values of the objective function are gathered at the end of each design cycle, and the best solution is chosen from the results of the p processors. The design variables corresponding to the best solution at each design cycle are used as the design variables to begin the next design cycle on different processors. Iterations are terminated when the termination criteria are satisfied. Computations were carried on SGI Origin 3000 system using message-passing-interface library.

Results and Discussions

The sensitivity of the converged values of the objective functions on the size of the computational mesh for the test case is demonstrated by carrying out the optimization using different grid sizes on a single processor. Figure 2 shows the influence of grid size on the optimization achievement, which is defined by the decrease of the objective function value, i.e., $(F_0 - F)/F_0$, where F is the objective function defined by Eq. (3) and F_0 is the value of F at the start of iteration. From Fig. 2, it can be seen that the objective function value has a convergence tendency as the grid size increases. It can produce grid-independent solutions with a grid size of 128×48.

The convergence criteria for the parallel SPSA are set to be same as that of parallel SA reported by Wang and Damodaran[2] in order to compare the performance of the two parallel methods. The optimization process is terminated as the objective function reduces to 0.006, 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 NACA 0012 is 0.11420. Optimization results of parallel SA can be found in Wang and Damodaran.[2]

Figure 3 shows the variation of the objective function with the number of design cycles on different combinations of processors (only results from processor number zero are given) obtained by parallel SPSA method. It can be found that the objective function

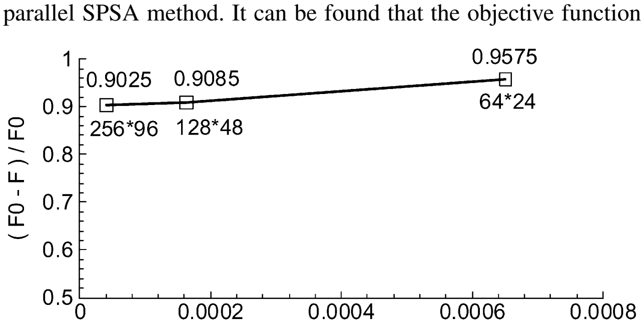


Fig. 2 Influence of grid size on the objective function values.

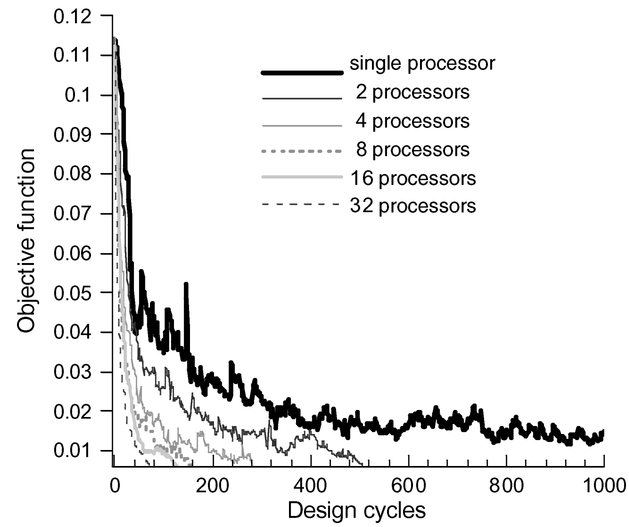


Fig. 3 Convergence histories using parallel SPSA.

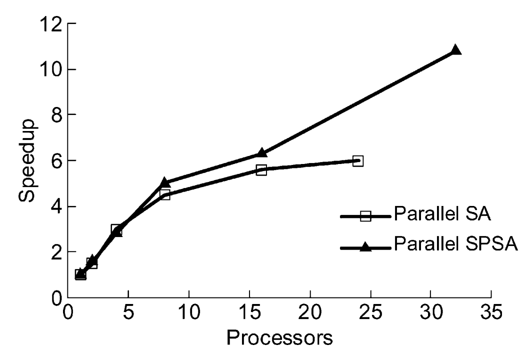


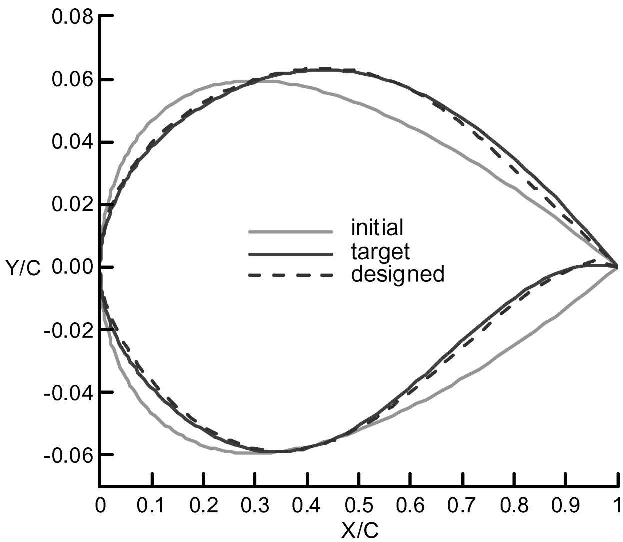
Fig. 4 Comparison of the speedup of parallel SPSA and parallel SA.

reduced to a value around 0.013 in about 800 design cycles on a single processor, after which the convergence speed becomes very slow. It is difficult to reduce the objective function value to 0.006 on a single processor, while parallel SPSA can reduce the objective function to a value less than 0.006 in a fewer design cycles, as shown in Fig. 3. This suggests that parallel SPSA can increase the possibility of finding the global optimum. This can be explained by the stochastic nature of SPSA. Choice of tuning parameters is critical for the performance of a stochastic method. The random number generator or random seeds also affects the performance of a stochastic method. Different random number generator or random seeds as well as different tuning parameters will lead to different convergence speed and different final design results. For this inverse design problem, it is difficult to reduce the objective function value to less than 0.01 using a single processor with the selected random number seeds and tuning parameters, suggesting other random number seeds or tuning parameters should be tried to search the global optimum, which means time-consuming reiterations of the design process, while, with parallel SPSA, several processors or tens of processors search the optimum. Every processor has its own random number seeds and tuning parameters. Consequently, the ability to get the global optimum is increased, and therefore the objective function value can be reduced to a value less than 0.006 in a reasonable computation cost on each processor.

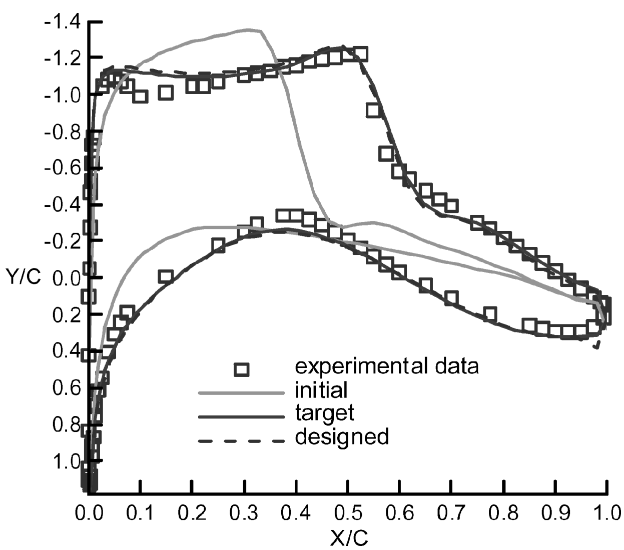
The number of design cycles reduced to 506 when two processors are used, 285 when four processors are used, 158 when eight processors are used, 128 when 16 processors are used, and 74 when 32 processors are used. The corresponding speedups based on these number of the design cycles/function evaluations are 1.6, 2.8, 5.0, 6.3, and 10.8, respectively, whereas, using parallel SA, the speedups based on the functions evaluation are 1.5, 3.0, 4.5, 5.6, and 6.0 corresponding to 2, 4, 8, 16, and 24 processors, respectively, reported in Wang and Damodaran.[2] Figure 4 compares the speedup vs number of processors of parallel SPSA and parallel SA. From Fig. 4 it can be seen that the speedups are almost same for the two parallel methods when the number of processors is less than eight, whereas speedups of parallel SPSA are higher than those of parallel SA when the number of processors is more than eight.

A gradient-based Broydon–Fletcher–Goldfarb–Shanno method was applied to the same problem and it was observed that about 350 function evaluations were required to reduce the objective function to 0.006. Therefore, by using eight or more processors, the parallel SPSA method can reduce the function evaluations on each processor to the same order as that obtained by deterministic methods for this inverse design problem.

Figure 5 compares the initial, target, and designed airfoil shapes obtained using parallel SPSA implemented on 16 processors as well as the corresponding airfoil surface-pressure distributions. It can be seen that the airfoil shapes and surface pressure distributions of the designed airfoil have good agreement with those of the target airfoil defined by a RAE 2822 airfoil. The calculation pressure distributions also have a good agreement with experimental data,[11] suggesting that the numerical simulation has a reasonable accuracy.



a) Airfoil shapes



b) Airfoil surface-pressure distributions

Fig. 5 Comparisons of airfoil shapes and surface-pressure distributions.

Conclusions

Parallel simultaneous perturbation stochastic approximation (SPSA) has been implemented on multiple processors to deal with the inverse airfoil design problem, in which the objective functions are evaluated with a computational-fluid-dynamics solver. Results clearly demonstrated that parallel SPSA is a feasible approach, which not only reduces the number of objective function evaluations on each processor by reducing the calculation time (wall-clock time), but also increases the chance of finding the global optimum. Compared with a parallel simulated-annealing method, the parallel SPSA method is more efficient computationally when using eight or more processors, and it is easy to parallelize the serial SPSA code. Future study will focus on the implementation of parallel SPSA on other single/multidisciplinary problems for testing its robustness for complex design problems.

References

[1]Xing, X. Q., and Damodaran, M., "Application of Simultaneous Perturbation Stochastic Approximation Method for Aerodynamic Shape Design Optimization," AIAA Paper 2002-5643, Sept. 2002.

[2]Wang, X., and Damodaran, M., "Inverse Design of Transonic Airfoils Using Parallel Simulated Annealing and CFD," *AIAA Journal*, Vol. 40, No. 4, 2002, pp. 791–794.

[3]Quagliarella, D., and Cioppa, A. D., "Genetic Algorithms Applied to the Aerodynamic Design of Transonic Airfoils," AIAA Paper 94-1896-CP, June 1994.

[4]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, Dec. 1994, pp. 447–461.

[5]Vicini, A., and Quagliarella, D., "A Multi-Objective Approach to Transonic Wing Design by Means of Genetic Algorithms," NATO RTO AVT, RTO-MP-35, June 2000.

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

[7]Ramamoorthy, P., and Padmavathi, K., "Airfoil Design by Optimization," *Journal of Aircraft*, Vol. 14, No. 2, 1977, pp. 219–221.

[8]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.

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

[10]Spall, J. C., "An Overview of the Simultaneous Perturbation Method for Efficient Optimization," *Johns Hopkins APL Technical Digest*, Vol. 19, No. 4, 1998, pp. 482–492.

[11]Cook, P. H., McDonald, M. A., and Firmin, M. C. P., "Aerofoil RAE2822—Pressure Distributions, and Boundary Layer and Wake Measurement," AGARD, Advisory Rept. 138 Paper A6, May 1979.