[10]Orlik-Rückemann, K. J., "Aerodynamic Aspects of Aircraft Dynamics at High Angles of Attack," *Journal of Aircraft*, Vol. 20, No. 9, 1983, pp. 737–751.

[11]Jategaonkar, R. V., Mönnich, W., Fischenberg, D., and Krag, B., "Data Gathering for C-160 Transall Flight Simulator, Part 1: Math Model and Aerodynamic Data Base," DLR-IB 111-93/37, Brunswick, Germany, May 1993.

[12]Jategaonkar, R. V., and Plaetschke, E., "Maximum Likelihood Parameter Estimation from Flight Test Data for General Non-Linear Systems," DFVLR-FB 83-14, Brunswick, Germany, March 1983.

Airfoil Design Utilizing Parallel Processors

Stephen C. Brawley* and Garth V. Hobson†
U.S. Naval Postgraduate School, Monterey, California 93943

Introduction

AN aerodynamic design scheme using parallel processors has been developed that significantly decreases the processing time required to optimize a desired performance. The parallel optimization scheme, when coupled with a flow solver, evaluates the aerodynamic performance of numerous geometries simultaneously. A test case was conducted utilizing the parallel optimization scheme and a similar sequential optimization scheme to design an airfoil to match the pressure distribution corresponding to a known shape. This design application demonstrates the practicality and versatility of aerodynamic design via optimization using parallel processors.

Computational fluid dynamics (CFD) has become a valuable engineering tool in both aerodynamic analysis and design. Airfoil optimization uses multivariable calculus to minimize an objective function selected by the designer. If the objective function is continuous, which is usually the case in airfoil design, the desired performance of the airfoil is optimized as the objective function is reduced.

Optimization methods for airfoil design have many advantages, including the flexibility of the designer to choose various design performance criteria and to use different types of flow solvers. However, their main disadvantage is the amount of computer processing time required for the design criteria to be optimized. Flowfield calculations over various geometries of airfoils that evaluate their aerodynamic performances constitute the vast majority of the computer time required.

To significantly speed up the design, the required processing time must be reduced. An investigation into gradient-based optimization schemes reveals that airfoil design can be treated as a parallel problem and can take advantage of the capabilities of parallel supercomputers. Parallel processors are used to conduct multiple CFD solutions for different airfoil geometries simultaneously to reduce the time required for airfoil design via optimization.

Presented as Paper 95-0125 at the AIAA 33rd Aerospace Sciences Meeting and Exhibit, Reno, NV, Jan. 9–12, 1995; received Feb. 19, 1995; revision received Oct. 1, 1995; accepted for publication Oct. 16, 1995. Copyright © 1995 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved.

*Adjunct Professor, Department of Aeronautics and Astronautics. Member AIAA.

†Associate Professor, Department of Aeronautics and Astronautics. Member AIAA.

Quasi-Newton Optimization Using Parallel Processors

An objective function appropriate for airfoil design is based upon the aerodynamic performance of the airfoil. Numerous evaluations of the objective function f are necessary for the gradient calculations and for line searches to locate a minimum. Since each objective function evaluation requires a CFD solution, the vast majority of computational time needed to design an airfoil is spent calculating the flowfield around various airfoil geometries.

Kennelly[1] developed the optimization routine QNMDIF based upon a quasi-Newton method and used the routine in airfoil design. In this research, parallel processors are used to simultaneously calculate the flowfields over multiple airfoil geometries for the estimation of the gradient vectors and in directional searches for minimum objective functions. Conducting the gradient calculations and line searches in parallel greatly increases the speed and efficiency of the design procedure.

The parallel quasi-Newton optimization routine, PARQNM, assigns multiple processors to simultaneously calculate objective functions with different sets of design variables. For a second-order-accurate estimation of each component of the gradient, two function evaluations are required. For example, the first gradient component is estimated from the central-differencing calculation

$$\frac{\partial f}{\partial x_1} = \frac{f(x_1 + \Delta x_1, x_2, \ldots, x_n)}{2\Delta x_1} - \frac{f(x_1 - \Delta x_1, x_2, \ldots, x_n)}{2\Delta x_1} \tag{1}$$

For n design variables, $2n$ processors are used in PARQNM to calculate all forward- and backward-difference function evaluations in parallel for the estimation of the gradient vector.

The method of gradient calculation used in PARQNM has several advantages over the method used in the sequential quasi-Newton routine QNMDIF. Most importantly, all function evaluations are done in parallel instead of sequentially. Also, the central-difference estimation of each gradient component used in the parallel routine is more accurate than the forward-difference estimation used in QNMDIF. In QNMDIF, if the forward-difference estimation of the gradient fails to provide a direction that reduces the objective function in a line search, valuable processing time will be wasted before computing the gradient derivatives based upon a central-difference approximation.

A parallel line search was developed that minimizes a multivariable objective function more efficiently and many times faster than the line search used in the sequential optimization routine. After the direction of search P is calculated based upon the gradient vector, the new set of design variables X becomes a function of a scalar q as shown in Eq. (2):

$$X^{k+1} = X^k + qP^k \tag{2}$$

Different values of q are assigned to the processors in equal intervals between its minimum and maximum values selected by the designer. Each processor then simultaneously computes the objective function for a unique set of design variables. The new set of design variables corresponding to the minimum objective function is then sent to all processors in a global message.

The parallel line search conducts all expensive function evaluations in parallel, unlike the sequential version that requires numerous function evaluations calculated sequentially using parabolic interpolation. Also, the parallel line search helps protect against convergence to a local minimum instead of a global minimum because of more function evaluations along the line of search.

Comparison of Sequential and Parallel Optimization Schemes in Airfoil Design

A test case was conducted using the quasi-Newton sequential and parallel optimization schemes to design airfoils that approach the pressure distribution around a target symmetric airfoil in subsonic flight conditions and at 0-deg angle of attack.

Geometry and Flow Solver

The baseline airfoil used in this application is a NACA 0008 symmetric airfoil. The values of the thickness of the airfoil at eight different positions along the chord were varied. A geometry package developed by the McDonnell Douglas Corporation was utilized to fit an eighth-order Chebychev polynomial through the collocation points to describe the airfoil's thickness distribution.[2]

The flowfield properties and aerodynamic performance of various airfoil shapes were calculated by a two-stage Runge–Kutta Euler flow solver, RK2EULER.[3] RK2EULER is easily vectorizable and updates the flowfield properties around an airfoil faster each iteration on vector processors than similar implicit or Crank–Nicholson flow solvers. The flowfield properties were initialized at freestream conditions with the density set to unity and the pressure set to the reciprocal of the ratios of specific heats, and 1500 flowfield iterations were conducted on each airfoil. The number of flowfield iterations was set to ensure a reduction of three orders of magnitude in the summations of density residuals throughout the flowfield. GRAPE was used to generate 133 × 34 grid points around each airfoil.[4]

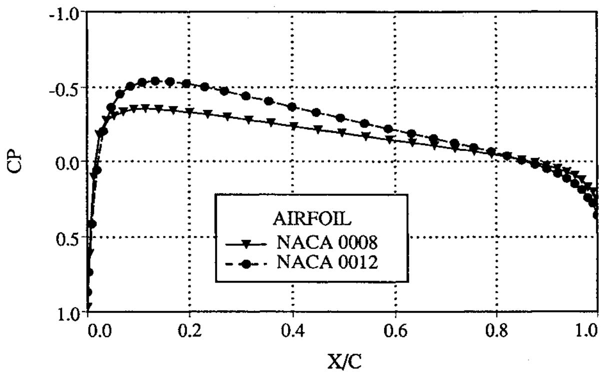

Fig. 1 Pressure distributions of baseline and target airfoils.

Design Criterion

The goal of this test case was to use the optimization routines to design airfoils to match the inviscid pressure distribution around a NACA 0012 airfoil. The flowfield properties and pressure distribution around each airfoil shape in the design process were calculated using the inviscid Euler flow solver. The objective function associated with each airfoil shape was determined by summing the square of the difference between the desired and calculated coefficients of pressure C_p,

$$f = \sum_{i=1}^{73} (C_{p-\text{calculated}_i} - C_{p-\text{target}_i})^2 \tag{3}$$

where the 73 points were located around the airfoil surface. The majority of varied thicknesses along the airfoil were near the leading edge of the airfoil; this is where the largest difference between the baseline and target pressure distribution exists as shown in Fig. 1.

PARQNM and QNMDIF were terminated either when the objective function was reduced to less than 10% of its original value, or after a maximum of 20 optimization cycles. This criterion ensures a convergence towards the target airfoil shape and keeps the amount of required processing time to a reasonable amount.

Results

The parallel quasi-Newton optimization design was performed using 16 i860 processors on the Intel iPSC/860 hypercube computer. The sequential optimization design was performed using a UNIX workstation with a single i860 processor. The parallel optimization scheme completed the airfoil design test case 18 times faster than the sequential case and in fewer optimization cycles. The utilization of parallel processors significantly decreased the processing time necessary for the airfoil design test case and increased the efficiency of the optimization scheme.

Figure 2 compares the convergence history of the objective function using the two quasi-Newton optimization schemes. The parallel application required each of the 16 processors to calculate eight objective functions in parallel and was completed in roughly 4 h. The sequential optimization scheme showed a much lower convergence rate. The final objective function was reduced to 15% of its initial value after completing 407 flowfield evaluations in 72 h.

The parallel airfoil design was more efficient than the sequential airfoil design because of the utilization of multiple processors for the parallel gradient calculation and for parallel

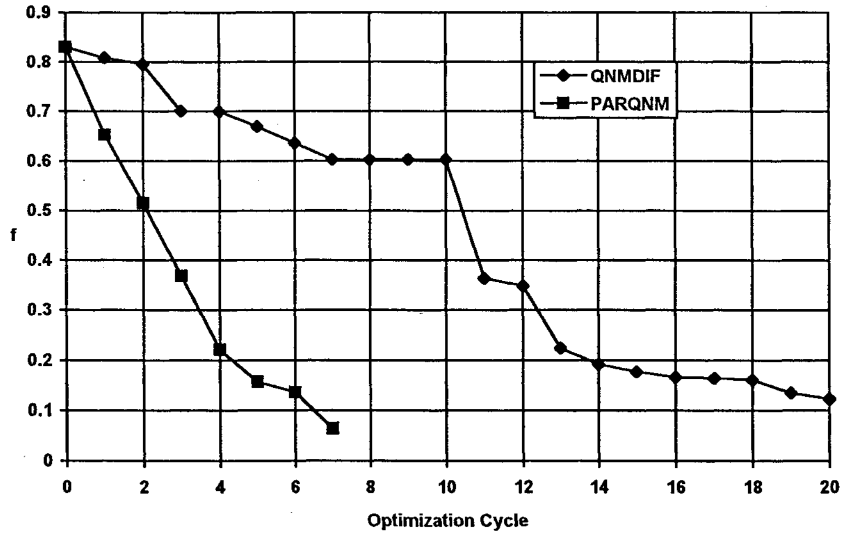

Fig. 2 Parallel and sequential convergence histories.

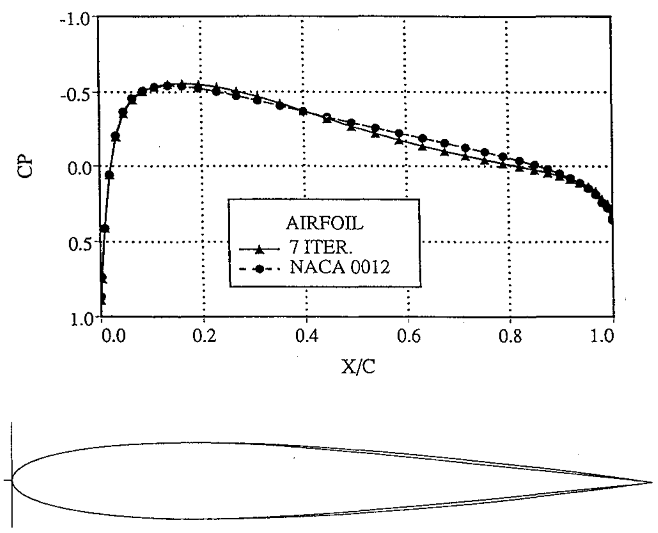

Fig. 3 Pressure distributions and shapes of design and target airfoils.

directional searches. The parallel optimization scheme used central-difference estimations of the derivatives for the calculation of each component of the gradient vector. The sequential optimization scheme first attempted forward-difference estimations of the derivatives because less objective function evaluations were required. When the directional search used by QNMDIF did not reduce the objective function based upon the forward-difference estimation of the gradient, QNMDIF then recomputed central-difference estimations and conducted additional directional searches. This resulted in great inefficiencies for the sequential airfoil design test case.

The parallel line search was more efficient in reducing the objective function than the sequential line search. The minimum variation for the directional search was determined by the value of the estimated machine precision, and the maximum variation of the thickness was set to 1% chord to ensure only small perturbations of the airfoil shape. The sequential line search evaluated a maximum of eight objective functions in each direction of search. The parallel line search was more thorough than the sequential search because it evaluated 16 objective functions in the direction of search including the maximum and minimum points.

The final shape of the design airfoil using PARQNM is shown with the NACA 0012 target airfoil with their resulting pressure distributions in Fig. 3. The design airfoil's shape is nearly identical to the target airfoil from the leading edge to the point of maximum thickness where the greatest changes in pressure occur.

Conclusions

This work applies recent advances in parallel supercomputing technology to an intuitively parallel problem. Through the use of parallel supercomputers, applications of optimization methods based upon gradient methods are faster and more efficient. The parallel optimization routine performs second-order-accurate gradient estimations and more thorough directional searches in parallel that increases the speed and the efficiency of the quasi-Newton routine. For the particular case of airfoil design via optimization that requires multiple calculations of expensive objective functions, the utilization of the parallel quasi-Newton routine can result in design solutions many times faster than with a sequential optimization routine.

References

[1]Kennelly, R. A., "Improved Method for Transonic Airfoil Design-by-Optimization," AIAA Paper 83-1864, Jan. 1983.

[2]Verhoff, A., Stookesberry, D., and Cain, A., "An Efficient Approach to Optimal Aerodynamic Design, Part 1: Analytic Geometry and Aerodynamic Sensitivities," AIAA Paper 93-0099, Jan. 1993.

[3]Brawley, S. C., "Aerodynamic Design Using Parallel Processors," Ph.D. Dissertation, U.S. Naval Postgraduate School, Monterey, CA, Sept. 1993.

[4]Sorenson, R. L., "A Computer Program to Generate Two-Dimensional Grids About Airfoils and Other Shapes by the Use of Poisson's Equation," NASA TM-81198, May 1980.

Analytical Expression of Induced Drag for a Finite Elliptic Wing

Masami Ichikawa*
Okayama Prefectural University, Okayama-ken 719-11, Japan

Introduction

THE two available formulas for calculating induced drag for a thin wing differ in their physical concepts. One is called the near field and the other is the far field. Since both equations agree with each other in the thin wing theory, induced drag calculations by these formulas have been used to examine numerical planar lifting-surface methods.[1,2] Most of these lifting-surface methods have only demonstrated the agreement of two kinds of induced drags, but have not addressed the accuracy of predicted induced drags. For example, Wagner's result for a variable-sweep wing shows a good agreement, but its predicted distribution for the sectional induced drag is excessively wavy.[3] Until now, analytical verifications have been unavailable for even a circular wing in incompressible flow.[4] Recently, the author has obtained such solutions by using Kida's method.[5] This Note shows the procedure for obtaining analytical solutions of the induced drag of an elliptic wing in incompressible steady flow, based on linearized theory. Furthermore, some numerical results are presented.

Formulation

Reference 5 gives the exact lifting-surface solution of the elliptic wing shown in Fig. 1 using the acceleration potential. This solution consists of two parts: one is the solution for the wing of the geometrical parameter k, defined in Fig. 1, less than unity, and the other is that for wing of larger than unity. Only the former case is discussed here.

Far-Field Induced Drag Solution

The total induced drag D_i acting on a wing surface is estimated by the following relation:

$$D_i = -\rho \int_{-a}^{+a} \Gamma(x) \cdot w_i(x) \, \mathrm{d}x \tag{1}$$

where ρ denotes the air density, $\Gamma(x)$ is the circulation distribution, and $w_i(x)$ is the induced downwash velocity distribution on the wing surface by a planar wake over the wingspan.

Received Aug. 26, 1995; revision received Dec. 20, 1995; accepted for publication Jan. 5, 1996. Copyright © 1996 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved

*Assistant Professor, Department of System Engineering.