of the grid size). The solution is considered converged when the maximum absolute residual is one order of magnitude smaller than the truncation error. The convergence history for the stream-function vorticity formulation is similar. The residual of the linear Poisson equation for the stream function is always of machine accuracy and the residual of the vorticity transport equation reduces quadratically; convergence is obtained after six iterations. For the biharmonic equation, convergence takes six iterations and is also quadratic. The solutions in terms of the wall skin-friction distribution are presented in Fig. 2. The results for the three formulations are in excellent agreement and correlate well with the skin-friction distribution presented by Briley.[8]

Next, the model problem of a separated flow in a symmetrical diffuser, introduced by Inoue,[9] is examined. The diffuser problem is solved using the stream-function vorticity and the biharmonic formulation. The inlet and outlet boundary of the diffuser are at $x = -1.0$ and $x = 3.0$, respectively. The shape factor of the diffuser wall $A = -0.08^9$. The centerline is located at $y = 1.0$ and the Reynolds number based on this reference length and the free-stream velocity is $R = 6250$. The inflow conditions provide the initial conditions for the entire flowfield. Convergence is quadratic and machine zero is reached in 6-7 iterations. The solutions in terms of the wall skin-friction distribution are shown in Fig. 3. The results for the two formulations are in excellent agreement and correlate well with Inoue's results, except for the outflow conditions. In Fig. 3 the solution for the higher Reynolds number, $R = 12500$, with a larger separation region is also presented.

The biharmonic equation is the most efficient of the three formulations in terms of CPU time and storage requirements. The biharmonic program is more than two-times faster and requires more than a factor two less memory than the stream-function vorticity program. The primitive variable method is the slowest among the three formulations and it puts severe demands on the computer storage requirements. In this case the bandwidth is $0(6N)$ and this coupled with the increase in the number of variables results in more than a four-fold increase in storage and CPU time as compared to the biharmonic program.

Conclusions

Three formulations of the two-dimensional Navier-Stokes equations are solved numerically using Newton's method and a direct solution routine for banded matrices. The fully implicit solution techniques use second-order central differencing for all the terms and are shown to be reliable and to provide quadratic convergence. The biharmonic formulation is most efficient in terms of CPU time and memory without loss of accuracy.

Finally, while it is well known that iterative methods (line overrelaxation or ADI) for biharmonic equations have very slow rates of convergence, the present study, using direct solvers, indicates that the biharmonic formulation is the most recommended.

References

[1] Roache, P. J., and Ellis, M. A., "The BID Method for the Steady-State Navier-Stokes Equations," *Computers and Fluids,* Vol. 3, 1975, pp. 305-320.

[2] Cebeci, T., Hirsh, R. S., Keller, H. B., and Williams, P. G., "Studies of Numerical Methods for the Plane Navier-Stokes Equations," *Computer Methods in Applied Mechanics and Engineering,* Vol. 27, 1981, pp. 13-44.

[3] Vanka, S. P., and Leaf, G. K., "An Efficient Finite-Difference Calculation Procedure for Multi-Dimensional Fluid Flows," AIAA Paper 83-1569, 1983.

[4] Schreiber, R., and Keller, H. B., "Driven Cavity Flow by Efficient Numerical Techniques," *Journal of Computational Physics,* Vol. 49, 1983, pp. 310-333.

[5] Bender, E. E., and Khosla, P. K., "Solution of the Two-Dimensional Navier-Stokes Equations using Sparse Matrix Solvers," AIAA Paper 87-0603, 1987.

[6] Dongarra, J. J., Moler, C. B., Bunch, J. R., and Stewart, G. W., "LINPACK User's Guide," SIAM, Philadelphia, 1979.

[7] Van Dam, C. P., Hafez, M., and Ahmad, J., "Comparison of Solution Techniques for Partially-Parabolized Navier-Stokes Equations," *Computational Fluid Dynamics,* edited by G. de Vahl Davis and C. Fletcher, North Holland, 1988, pp. 737-748.

[8] Briley, W. R., "A Numerical Study of Laminar Separation Bubbles using the Navier-Stokes Equations," *Journal of Fluid Mechanics,* Vol. 47, 1971, pp. 713-736.

[9] Inoue, O., "Separated Boundary Layer Flows with High Reynolds Numbers," *Lecture Notes in Physics,* Vol. 141, Springer-Verlag, New York, 1981.

Acceleration of Iterative Algorithms for Euler Equations of Gasdynamics

Seungsoo Lee* and George S. Dulikravich†
Pennsylvania State University, University Park,
Pennsylvania 16802

Introduction

ONE of the successful, explicit methods used to solve Euler and Navier-Stokes equations governing compressible flows is the finite-volume, Runge-Kutta, time-stepping algorithm.[1] Several attempts have been made to accelerate the iterative convergence of this method. These acceleration methods are based on local time stepping,[1] implicit residual smoothing,[1] enthalpy damping,[1] and multigrid techniques.[2] Also, an extrapolation procedure based on the power method and the Minimal Residual Method (MRM) were applied[2] to the Jameson's multigrid algorithm. The MRM has not been shown to accelerate the scheme without multigriding. It uses same values of optimal weights for the corrections to every equation in a system. If each component of the solution vector in a system of equations is allowed to have its own convergence speed, then a separate sequence of optimal weights could be assigned to each equation. This idea is the essence of the Distributed Minimal Residual (DMR) method,[3] which is based on the General Nonlinear Minimal Residual (GNLMR) concept.[4]

Time-Dependent Euler Equations

The system of time-dependent Euler equations of gasdynamics in a two-dimensional space can be written in a general conservative form[1] as

$$\frac{\partial Q}{\partial \tau} + \frac{\partial E}{\partial \xi} + \frac{\partial F}{\partial \eta} = 0 \tag{1}$$

where the global solution vectors combining mass, ξ-momentum, η-momentum, and energy conservation equations are

Received May 7, 1988; revision received Dec. 4, 1988; presented as AIAA Paper 89-0097 at the AIAA Aerospace Sciences Meeting, Reno, NV, Jan. 9-12, 1989. Copyright © 1989 American Institute of Aeronautics and Astronautics, Inc. All rights reserved.

*Graduate Student, Department of Aerospace Engineering.

†Associate Professor, Department of Aerospace Engineering. Senior Member AIAA.

defined as

$$Q = \frac{1}{D}\begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho e_0 \end{bmatrix} \qquad E = \frac{1}{D}\begin{bmatrix} \rho U \\ \rho u U + \xi_x p \\ \rho v U + \xi_y p \\ (\rho e_0 + p)U \end{bmatrix}$$

$$F = \frac{1}{D}\begin{bmatrix} \rho V \\ \rho u V + \eta_x p \\ \rho v V + \eta_y p \\ (\rho e_0 + p)V \end{bmatrix} \tag{2}$$

Here τ, ρ, u, v, p, e_0 are time, density, x and y components of the velocity vector, thermodynamic pressure, and mass-specific total energy, respectively. In addition, U, V, ξ, η, and D are the contravariant velocity vector components, non-orthogonal curvilinear computational coordinates, and determinant of the Jacobian of the transformation $\partial(\xi,\eta)/\partial(x,y)$ respectively.

Distributed Minimal Residual Method

Local residual of the finite-volume method at the global iteration level t can be expressed as

$$r^t = \int\!\!\int \frac{\partial Q^t}{\partial \tau}\,\mathrm{d}S = -\int\!\!\int \left(\frac{\partial E^t}{\partial \xi} + \frac{\partial F^t}{\partial \eta}\right)\mathrm{d}S \tag{3}$$

where S is the surface of the single grid cell, and the components Q^t, E^t, and F^t are defined in Eq. (2). In the DMR, corrections from M consecutive iteration levels are used to update the value of Q to $(t + 1)$ global iteration level. Thus,

$$Q^{t+1} = Q^t + \sum_m^M \Omega^m \tag{4}$$

where

$$\Omega^m = \begin{bmatrix} \omega_1^m & \Delta_1^m \\ \vdots & \vdots \\ \omega_L^m & \Delta_L^m \end{bmatrix} \tag{5}$$

Here, Δ_1^m are the corrections computed with the basic algorithm and ω_1^m are the weights for each of the $1 = 1\ldots, L$ equations in the system [Eq. (2)] at each of the $m = 1,\ldots, M$ consecutive iteration levels. Therefore, the new local residual for the single grid cell will be

$$r^{t+1} = -\int\!\!\int \left(\frac{\partial E^{t+1}}{\partial \xi} + \frac{\partial F^{t+1}}{\partial \eta}\right)\mathrm{d}S \tag{6}$$

Using a Taylor series expansion of E^{t+1} and F^{t+1} in time and truncating it after the first term results in

$$r^{t+1} = r^t - \sum_m^M \int\!\!\int \left\{\frac{\partial}{\partial \xi}\left(\frac{\partial E^t}{\partial Q^t}\,\Omega^m\right) + \frac{\partial}{\partial \eta}\left(\frac{\partial F^t}{\partial Q^t}\,\Omega^m\right)\right\}\mathrm{d}S \tag{7}$$

Define the global residual R^{t+1} at the global iteration level $(t + 1)$ as a sum of the squares of the local residuals, that is,

$$R^{t+1} = \sum_i^I \sum_j^J (r^{t+1})^*(r^{t+1}) \tag{8}$$

where I and J define the grid size, and the superscript * designates the transpose. The objective is to find optimum values of L sequences of M values of ω_1^m that will minimize the global residual R^{t+1} at the next global iteration level $(t + 1)$. To minimize R^{t+1}, it is necessary to use the values ω_1^m that satisfy

$$\frac{\partial R^{t+1}}{\partial \omega_1^m} = 0 \tag{9}$$

for all m and 1. Thus, from Eqs. (7), (8), and (9) it follows that

$$\sum_i^I \sum_j^J \left\{r^t - \sum_n^M \int\!\!\int \left[\frac{\partial}{\partial \xi}\left(\frac{\partial E^t}{\partial Q^t}\,\Omega^n\right) + \frac{\partial}{\partial \eta}\left(\frac{\partial F^t}{\partial Q^t}\,\Omega^n\right)\right]\mathrm{d}S\right\}^*$$
$$\times \left\{\int\!\!\int \left[\frac{\partial}{\partial \xi}\left(\frac{\partial E^t}{\partial Q^t}\,\frac{\partial \Omega^m}{\partial \omega_1^m}\right) + \frac{\partial}{\partial \eta}\left(\frac{\partial F^t}{\partial Q^t}\,\frac{\partial \Omega^m}{\partial \omega_1^m}\right)\right]\mathrm{d}S\right\} = 0 \tag{10}$$

where

$$\frac{\partial \Omega^m}{\partial \omega_1^m} = \{\Delta_k^m \delta_{kl}\} \tag{11}$$

and δ_{kl} is the Kronecker delta. Notice that

$$\Omega^n = \sum_q^L \frac{\partial \Omega^n}{\partial \omega_q^n}\,\omega_q^n \tag{12}$$

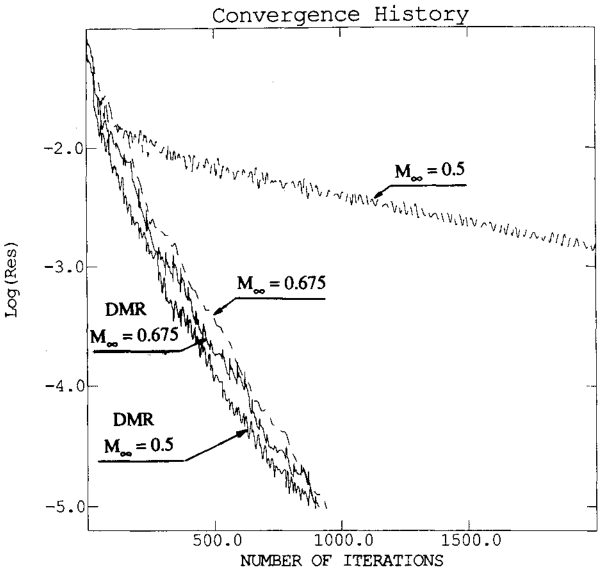

Fig. 1 Convergence histories in terms of iteration numbers: non-accelerated- (--------) and DMR accelerated (————) algorithm: channel flow.

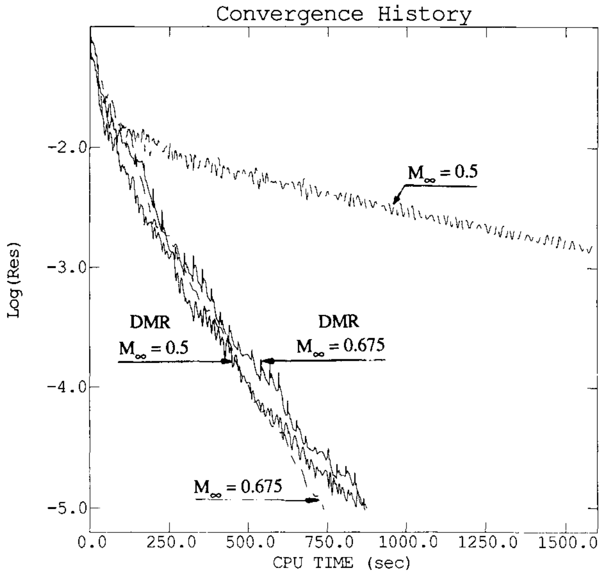

Fig. 2 Convergence histories in terms of the CPU time: non-accelerated- (--------) and DMR accelerated (————) algorithm: channel flow.

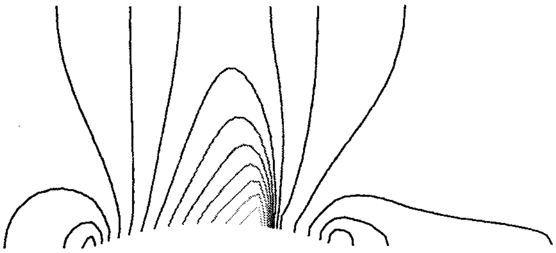

Fig. 3 Constant pressure contours for DMR-accelerated algorithm: transonic channel flow with $M_\infty = 0.675$.

Let

$$A_1^m = \iint \left[\frac{\partial}{\partial \xi}\left(\frac{\partial E^t}{\partial Q^t}\frac{\partial \Omega^m}{\partial \omega_1^m}\right) + \frac{\partial}{\partial \eta}\left(\frac{\partial F^t}{\partial Q^t}\frac{\partial \Omega^m}{\partial \omega_1^m}\right)\right] \mathrm{d}s \qquad (13)$$

Note that A_1^m is not a function of ω's. Then, Eq. (10) becomes

$$\sum_i^I \sum_j^J \sum_n^M \sum_q^L (A_q^n)^* A_1^m \omega_q^n = \sum_i^I \sum_j^J (r^t)^* A_1^m \qquad (14)$$

Let

$$C_{ql}^{nm} = \sum_i^I \sum_j^J (A_q^n)^* A_1^m B_1^m = \sum_i^I \sum_j^J (r^t)^* A_1^m \qquad (15)$$

The result is a system of $L \times M$ algebraic equations

$$\sum_n^M (\omega_l^n C_{1l}^{nm} + \omega_2^n C_{2l}^{nm} + \omega_3^n C_{3l}^{nm} + \ldots\ldots + \omega_L^n C_{Ll}^{nm}) = B_l^m \qquad (16)$$

for the $L \times M$ unknown optimal acceleration factors ω_l^m. The DMR applied to the finite-volume scheme[1] in the two-dimensional case needs approximately a 150% increase, and in the three-dimensional case, it needs approximately a 175% increase in computer memory over the original nonaccelerated algorithm.[1] Boundary conditions on the residuals in the integrals of Eq. (13) used extrapolation of the residuals from the interior of the flowfield.

Results

All computations were performed on a VAX 11/8550 computer in a single precision mode. The first sequence of tests was performed on the internal two-dimensional ($L = 4$) flow problems by combining four consecutive time steps ($M = 4$). This means that a 16 × 16 matrix [Eq. (16)] needs to be inverted. The test geometry was a 10% thick circular half airfoil on a wall of a straight two-dimensional channel. The H-type grid size was 65 × 17 points. The calculations were started with uniform flow, and the DMR was applied once after every 30 steps performed with the original unaccelerated code.[1] Figures 1 and 2 depict the convergence histories of flow calculations with $M_\infty = 0.5$ and $M_\infty = 0.675$. For the entirely subsonic flow ($M_\infty = 0.5$), the number of iterations needed to achieve the same level of residual is reduced almost by 60%, and the saving in computational time is about 50%. Both figures indicate that DMR in its present version does not accelerate transonic flow ($M_\infty = 0.675$) computations. Superimposed constant pressure contours (see Fig. 3) of the entire flowfield for both the nonaccelerated and the DMR accelerated schemes confirm that the DMR method does not adversely influence the quality of the solution.

The second test case was a flow around a circle. An O-type radially clustered grid consisting of 64 × 32 grid cells was used. We applied DMR after every 60 iterations by combining four consecutive time levels. When the critical freestream Mach number ($M_\infty = 0.4$) was used, Figs. 4 and 5 indicate that the DMR method in its present form offers practically no gain. At very low freestream Mach numbers, the system of Euler equations becomes very stiff, thus, rapidly reducing the convergence rate of the nonaccelerated scheme. On the other hand, when using $M_\infty = 0.1$, the DMR offers over 70% savings in the CPU time (see Fig. 5) over the nonaccelerated scheme.

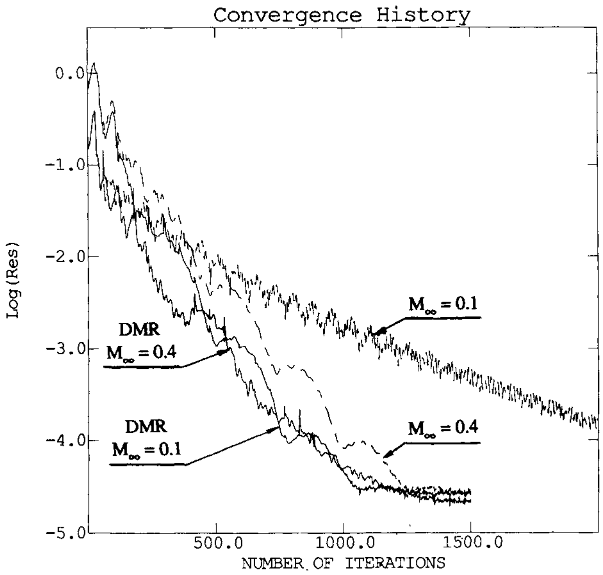

Fig. 4 Convergence histories in terms of iteration numbers: nonaccelerated- (--------) and DMR accelerated (————) algorithm: circle flow.

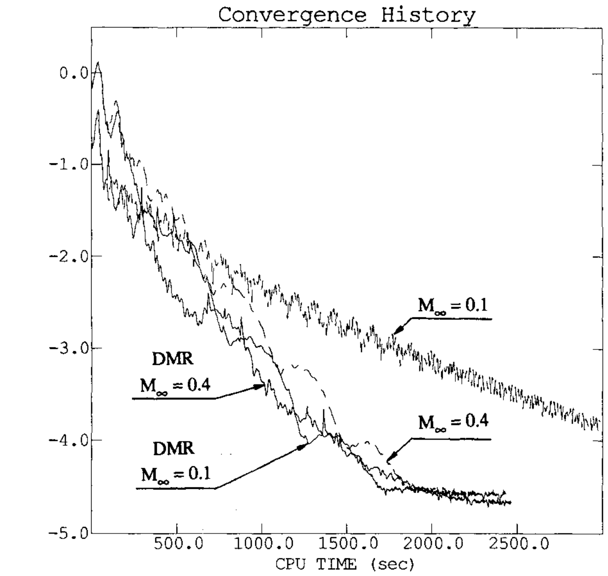

Fig. 5 Convergence histories in terms of the CPU time: nonaccelerated- (--------) and DMR accelerated (————) algorithm: circle flow.

Notice that all numerical results were obtained without the standard acceleration techniques such as explicit and implicit residual smoothing, enthalpy damping, multigriding, and vectorization. These methods could be added to further accelerate the algorithm. The method seems to offer substantial time savings when applied to compressible flow codes at low Mach numbers.

Conclusions

A new method for the acceleration of explicit iterative algorithms for the numerical solution of systems of partial differential equations has been developed. The method is based on the idea of allowing each partial differential equation in the system to approach the converged solution at its own optimal speed and at the same time to communicate with the rest of the equations in the system. The DMR method computes a separate sequence of optimal acceleration factors to be used for each component of the general solution vector. The acceleration scheme was applied to the system of time-dependent Euler equations of inviscid gasdynamics in conjunction with the finite-volume, Runge-Kutta, explicit, time-stepping algorithm. Using DMR without multigriding, between 30% and 70% of the total computational efforts were saved in the subsonic compressible flow calculations. The DMR method seems to be especially suitable for stiff systems of equations and can be applied to other systems of differential equations and other numerical algorithms.

Acknowledgment

This work was supported by the Air Force Office of Scientific Research/Numerical Mathematics Program under the grant AFOSR 87-0121 supervised by Thomas, Nelson, and Nachman.

References

[1] 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.

[2] Hafez, M., Parlette, E., and Salas, M. D., "Convergence Acceleration of Iterative Solutions of Euler Equations for Transonic Flow Computations," AIAA Paper 85-1641, July 1985.

[3] Dulikravich, G. S., Dorney, D. J., and Lee, S., "Iterative Acceleration and Physically Based Dissipation for Euler Equations of Gasdynamics," ASME WAM '88, Proceedings of Symposia on Advances and Applications in Computational Fluid Dynamics, edited by O. Baysal, Chicago, IL, FED-Vol. 66, 1988, pp. 81-92.

[4] Huang C. Y., and Dulikravich, G. S., "Fast Iterative Algorithms Based on Optimized Explicit Time-Stepping," *Computer Methods in Applied Mechanics and Engineering,* Vol. 63, Aug. 1987, pp. 15-36.

Oscillatory Shock Motion Caused by Transonic Shock Boundary-Layer Interaction

B. H. K. Lee*
National Research Council, Ottawa, Canada

Introduction

PERIODIC shock motions on airfoils at transonic flow conditions had been observed experimentally[1-4] for more than a decade. They have also been detected from numerical solutions of the Navier-Stokes equations[5] and recently by an unsteady viscous-inviscid interaction method.[6] Attempts have been made to formulate a model to predict the unsteady shock motion, but so far a satisfactory explanation of the mechanism of self-sustained shock oscillation and a method to estimate the frequency about which the shock wave oscillates are still lacking.

Spark schlieren photographs[1] of the flowfield over a supercritical airfoil with flow separation have indicated clearly the presence of upstream moving waves originating at the trailing edge and near-wake region. They are associated with wake fluctuations due to unsteady shock motions. A possible mechanism of the self-sustained shock-wave oscillation caused by unsteady transonic shock boundary-layer interaction on a supercritical airfoil with fully separated flow at the shock wave is illustrated in Fig. 1. The case of a shock wave oscillating on the upper airfoil surface about a mean position is considered (corresponding to Tijdeman's[7] type A shock motion). Because of the movement of the shock, pressure waves are formed which propagate downstream in the separated flow region at a velocity a_p. On reaching the trailing edge, the disturbances generate upstream moving waves at velocity a_u. These waves will interact with the shock and impart energy to maintain its oscillation. The loop is then completed and the period of the shock wave oscillation should agree with the time it takes for a disturbance to propagate from the shock to the trailing edge plus the duration for an upstream moving wave to reach the shock from the trailing edge. In this Note, experimental results supporting this model for self-sustained shock oscillation are presented.

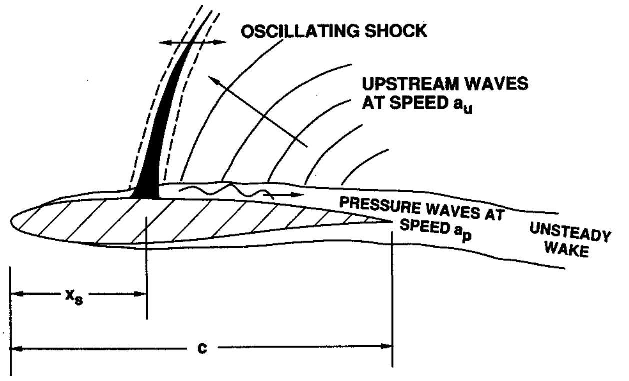

Fig 1 Model of self-sustained shock oscillation.

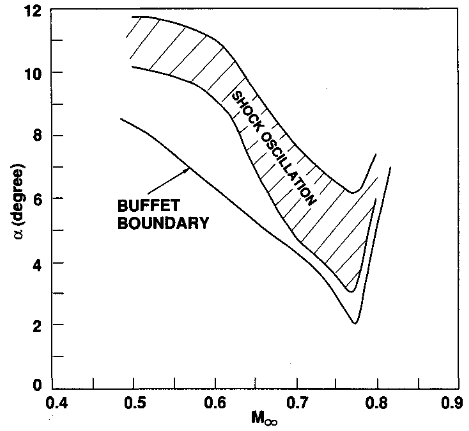

Fig 2 Region of shock oscillation for BGK No. 1 airfoil.

Experiments

The investigation was carried out in the high Reynolds number Two-Dimensional Test Facility[8] of the National Aeronautical Establishment. The airfoil tested was the BGK No. 1[4] with design Mach number and lift coefficient of 0.75 and 0.63, respectively. The chord c was 10 in., and thickness-to-chord ratio was 11.8%. The Reynolds number based on the chord was 20×10^6. In addition to the 50 pressure orifices on the airfoil upper surface and 20 on the lower surface for steady measurements, unsteady pressure data were obtained from 16 fast-response miniature transducers installed on the upper sur-

Received Nov. 17, 1988; revision received June 20, 1989. Copyright © 1989 by B. H. K. Lee, Published by the American Institute of Aeronautics and Astronautics, Inc. with permission.

*Senior Research Officer, National Aeronautical Establishment. Member AIAA.