AIAA'89

AIAA-89-0097
Acceleration of Iterative Algorithms for Euler Equations of Gasdynamics

Seungsoo Lee, George S. Dulikravich
and Daniel J. Dorney

The Pennsylvania State University
University Park, PA 16802

27th Aerospace Science Meeting
January 9-12, 1989
Reno, Nevada

ACCELERATION OF ITERATIVE ALGORITHMS FOR EULER EQUATIONS OF GASDYNAMICS

Seungsoo Lee, Graduate Assistant
George S. Dulikravich, Associate Professor
Daniel J. Dorney, Graduate Assistant

Department of Aerospace Engineering
Pennsylvania State University, University Park, PA 16802

ABSTRACT

A new algorithm for the acceleration of iterative schemes 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. The DMR (Distributed Minimal Residual) method allows a separate sequence of optimal weighting factors to be used for each equation in the system. The acceleration scheme was applied to the system of time-dependent Euler equations of inviscid gasdynamics in conjunction with the finite volume Rational Runge-Kutta (RRK) explicit time-stepping algorithm. Using DMR without multigridding, between 30% and 70% of the total computational efforts were saved in the subsonic compressible flow calculations. DMR method in its present version seems to be especially suitable for stiff systems of equations. It required less than double amount of storage of the original non-accelerated algorithm.

INTRODUCTION

One of the successful, explicit methods used to solve Euler and Navier-Stokes equations governing compressible flows subject to the various flow conditions is the Rational Runge-Kutta (RRK) time-stepping algorithm [1,2]. It is based on the finite volume technique with 2nd-4th order blended non-physical (artificial) dissipation [1]. Several attempts have been made to accelerate the iterative convergence of this method. They include local time stepping [1], implicit residual smoothing [1], enthalpy damping [1] and multigrid techniques [3]. Also, an extrapolation procedure based on the power method and the Minimal Residual Method (MRM) were applied [3] to the finite volume Runge-Kutta method together with multigridding. In the MRM [3], a weighted combination of the corrections at consecutive iteration levels is extrapolated and the weights are chosen to minimize the L_2 norm of the future residual. The extrapolation was performed without considering the specific properties of the governing equations. The GNLMR (Generalized Non-Linear Minimal Residual) method [4,5,6,7] utilizes the information from the governing equations. It has been applied successfully to a number of scalar linear and nonlinear partial differential equations.

Both MRM and GNLMR method use the same values of optimal weights for the corrections to every equation in a system. Nevertheless, since each component of the solution vector in a system of equations has its own convergence speed, the sequence of optimal weights could be allowed to vary from equation to equation. The authors believe that this concept underlying the Distributed Minimal Residual (DMR) method is similar to the general idea behind the preconditioning techniques. With the preconditioning, the eigenvalues of the system are changed so that the different CFL (Courant-Friedrichs-Levy) number can be used for each characteristic variable. This paper presents the theory constituting the DMR method and demonstrates the advantages of the new algorithm with a number of computational examples. Applications of the DMR to the system of Euler equations of inviscid gasdynamics are presented. The formulation can be equally well applied to other systems of differential equations and to other types of numerical integration algorithms.

TIME-DEPENDENT EULER EQUATIONS OF INVISCID GASDYNAMICS

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

$$\frac{\partial Q}{\partial t} + \frac{\partial E}{\partial \xi} + \frac{\partial F}{\partial \eta} = 0 \qquad (1)$$

where the global solution vectors combining mass, x-momentum, y-momentum and energy conservation equations are defined as

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

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

Here, ρ, u, v, p, e_o are the 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 transformation $\partial(\xi,\eta)/\partial(x,y)$, respectively.

The contravariant components U and V of the velocity vector in the body-conforming (ξ,η) coordinate system are given by

$$U = \xi_x\, u + \xi_y v \tag{3}$$

$$V = \eta_x\, u + \eta_y v \tag{4}$$

The total energy per unit mass for a calorically perfect gas is

$$e_o = c_v T + \frac{1}{2}(u^2+v^2) \tag{5}$$

where c_v is the specific heat at constant volume and T is the absolute temperature. The determinant of the Jacobian geometric transformation matrix is

$$D = \xi_x\, \eta_y - \xi_y \eta_x \tag{6}$$

FINITE VOLUME RUNGE-KUTTA TIME-STEPPING ALGORITHM

In the finite volume method [1], the governing equations are integrated over each computational cell in the (ξ,η) computational plane. With the help of the divergence theorem, the surface integral is transformed into a sum of line integrals. These integrals are discretized with the assumption that the fluxes are constant along the cell faces. Each quantity at the cell face is evaluated as the average of the values at the neighboring cell centers (cell centered scheme).

The cell centered finite volume method is identical to the central difference scheme on a uniform grid. It is known that the central difference scheme produces odd-even decoupling. To suppress this tendency,the artificial dissipation terms are added to the discretized equation [1]. The mixture of 2nd and 4th order artificial dissipation terms [1] was used.

$$dQ = d_\xi Q + d_\eta Q \tag{7}$$

where d is the artificial dissipation operator and Q is the vector defined in Eq. 2. The two terms on the right hand side of Eq. 7 are contributions from the two computational directions. They can be written [1] as:

$$d_\xi Q = d_{i+1/2j} - d_{i-1/2j}; \quad d_\eta Q = d_{ij+1/2} - d_{ij-1/2} \tag{8}$$

The terms on the right hand sides of Eq. 8 are similar [1]. For example,

$$d_{i+1/2j} = \frac{1}{D\,\Delta t}\{\varepsilon^{(2)}_{i+1/2j}\,(Q_{i+1j} - Q_{ij})$$

$$- \varepsilon^{(4)}_{i+1/2j}\,(Q_{i+2j} - 3Q_{i+1j} + 3Q_{ij} - Q_{i-1j})\} \tag{9}$$

where the second and fourth order coefficients multiplying the flux derivative terms are flow adaptive coefficients. The scaling with the area D and the local time step, Δt, is included [8] to correspond to the formulation of the Euler equations in the transformed plane. A pressure sensor is introduced to locate regions requiring large amounts of artificial dissipation. It is based on the second derivative of pressure [1,9]

$$\nu_{ij} = \frac{|p_{i+1j} - 2p_{ij} + p_{i-1j}|}{p_{i+1j} + 2p_{ij} + p_{i-1j}} \tag{10}$$

The flow adaptive coefficients are then calculated [1] as:

$$\varepsilon^{(2)}_{i+1/2/} = k^{(2)}\max(\nu_{i+1j}, \nu_{ij})$$

$$k^{(2)} \cong 1/4$$

$$\varepsilon^{(4)}_{i+1/2j} = \max(0, (k^{(4)} - \varepsilon^{(2)}_{i+1/2j}))$$

$$k^{(4)} \cong 1/256 \tag{11}$$

The system of time-dependent Euler equations is known to be of hyperbolic type and the boundary conditions should be applied according to the direction of the characteristics. At the inflow and outflow boundaries, the incoming Riemann invariant is specified and the outgoing Riemann invariant is extrapolated from the interior points. Also, the entropy and the tangential velocity are prescribed at the inflow. At the outflow, these quantities are extrapolated from the interior of the domain.

At the solid wall, the normal momentum equation is used to evaluate the wall pressure. The contravariant velocity component U at the ghost cells inside the solid body is extrapolated, while the contravariant velocity component V is reflected from the wall.

An explicit Runge-Kutta time-stepping [2,1] scheme is used to evolve the solution in time. The 4th order Runge-Kutta scheme is given by

$$Q^{(0)} = Q^n \tag{12}$$

$$Q^{(1)} = Q^n - \frac{\Delta t}{4}\,(NQ^{(0)} - dQ^{(0)}) \tag{13}$$

$$Q^{(2)} = Q^n - \frac{\Delta t}{3}\,(NQ^{(1)} - dQ^{(0)}) \tag{14}$$

$$Q^{(3)} = Q^n - \frac{\Delta t}{2}\,(NQ^{(2)} - dQ^{(0)}) \tag{15}$$

$$Q^{(4)} = Q^n - \Delta t\,(NQ^{(3)} - dQ^{(0)}) \tag{16}$$

$$Q^{n+1} = Q^{(4)} \tag{17}$$

where N is the discretization operator of the finite volume method. The artificial dissipation is evaluated at the beginning of each Runge-Kutta global step and it was not updated during the intermediate steps. Linear stability analysis indicates that the explicit Runge-Kutta scheme is stable if CFL ≤ 2.8. If the grid spacing in (ξ,η) plane is uniform $\Delta\xi = \Delta\eta = 1$, then the time step is given [9] by

$$\Delta t_\xi = \frac{1}{|U| + a\,(\xi_x^{\,2} + \xi_y^{\,2})^{1/2}} \tag{18}$$

$$\Delta t_\eta = \frac{1}{|V| + a\,(\eta_x^{\,2} + \eta_y^{\,2})^{1/2}} \tag{19}$$

where a is the local speed of sound and the combined time step [9] is

$$\Delta t = \left(\frac{\Delta t_\xi\, \Delta t_\eta}{\Delta t_\xi + \Delta t_\eta}\right) \text{CFL} \tag{20}$$

DISTRIBUTED MINIMAL RESIDUAL METHOD (DMR)

~~Global~~ Local residual of the finite volume method at time level t can be expressed as

$$r^t = \iint \frac{\partial Q}{\partial t}\, dS = -\iint \left(\frac{\partial E}{\partial \xi} + \frac{\partial F}{\partial \eta}\right) dS \tag{21}$$

where S is the surface of the single grid cell and components Q, E and F of the generalized solution vector are defined in Eq. 2.

We plan to use corrections from M consecutive time levels to update the value of Q to (t+1) global time level. Thus,

$$Q^{t+1} = Q^t + \sum_m^M \Omega^m \tag{22}$$

where

$$\Omega^m = \begin{bmatrix} \omega_1^{\,m}\, \Delta_1^{\,m} \\ \omega_2^{\,m}\, \Delta_2^{\,m} \\ \cdots\dot{m}\cdots\dot{m} \\ \omega_L^{\,m}\, \Delta_L^{\,m} \end{bmatrix} \tag{23}$$

and Δ_ℓ^m are the corrections and ω_ℓ^m are the weights for each of the $\ell = 1\ldots, L$ equations in the system (Eq. 2) at each of the $m = 1, \ldots,$ M consecutive global time levels. Therefore, upon substituting Eq. 22 in Eq. 21, the new local residual for the single cell will be

$$r^{t+1} = -\iint \left[\frac{\partial}{\partial \xi} E\,(Q^t + \sum_m^M \Omega^m) + \frac{\partial}{\partial \eta} F\,(Q^t + \sum_m^M \Omega^m)\right] dS \tag{24}$$

Using a Taylor series expansion truncated after the first term results in

$$r^{t+1} = -\iint \left[\frac{\partial E}{\partial \xi} + \frac{\partial F}{\partial \eta}\right] dS - \sum_m^M \iint \left[\frac{\partial}{\partial \xi}\left(\frac{\partial E}{\partial Q}\,\Omega^m\right) + \frac{\partial}{\partial \eta}\left(\frac{\partial F}{\partial Q}\,\Omega^m\right)\right] dS \tag{25}$$

or

$$r^{t+1} = r^t - \sum_m^M \iint \left[\frac{\partial}{\partial \xi}\left(\frac{\partial E}{\partial Q}\,\Omega^m\right) + \frac{\partial}{\partial \eta}\left(\frac{\partial F}{\partial Q}\,\Omega^m\right)\right] dS \tag{26}$$

Define the global residual R^t as a sum of the squares of the local residuals, that is,

$$R^t = \sum_i^I \sum_j^J (r^t)^* \,(r^t) \tag{27}$$

where I and J define the grid size and the superscript * designates the transpose of an array. Then, the global residual at the next global time level will be

$$R^{t+1} = \sum_i^I \sum_j^J \left\{r^t - \sum_n^M \iint \left[\frac{\partial}{\partial \xi}\left(\frac{\partial E}{\partial Q}\,\Omega^n\right) + \frac{\partial}{\partial \eta}\left(\frac{\partial F}{\partial Q}\,\Omega^n\right)\right] dS\right\}^* \cdot \left\{r^t - \sum_m^M \iint \left[\frac{\partial}{\partial \xi}\left(\frac{\partial E}{\partial Q}\,\Omega^m\right) + \frac{\partial}{\partial \eta}\left(\frac{\partial F}{\partial Q}\,\Omega^m\right)\right] dS\right\} \tag{28}$$

To minimize R^{t+1}, it is necessary to use the values of ω_ℓ^m that satisfy

$$\frac{\partial R^{t+1}}{\partial \omega_\ell^m} = 0 \tag{29}$$

for all m and ℓ. Thus, from Eq. 29 and Eq. 28 it follows that

$$\sum_i^I \sum_j^J \left\{r^t - \sum_n^M \iint \left[\frac{\partial}{\partial \xi}\left(\frac{\partial E}{\partial Q}\,\Omega^n\right) + \frac{\partial}{\partial \eta}\left(\frac{\partial F}{\partial Q}\,\Omega^n\right)\right] dS\right\}^* \cdot \left\{\iint \left[\frac{\partial}{\partial \xi}\left(\frac{\partial E}{\partial Q}\,\frac{\partial \Omega^m}{\partial \omega_\ell^m}\right) + \frac{\partial}{\partial \eta}\left(\frac{\partial F}{\partial Q}\,\frac{\partial \Omega^m}{\partial \omega_\ell^m}\right)\right] dS\right\} = 0 \tag{30}$$

or

$$\sum_i^I \sum_j^J (r^t)^* \left\{\iint \left[\frac{\partial}{\partial \xi}\left(\frac{\partial E}{\partial Q}\,\frac{\partial \Omega^m}{\partial \omega_\ell^m}\right) + \frac{\partial}{\partial \eta}\left(\frac{\partial F}{\partial Q}\,\frac{\partial \Omega^m}{\partial \omega_\ell^m}\right)\right] dS\right\} = \sum_i^I \sum_j^J \sum_n^M \left\{\iint \left[\frac{\partial}{\partial \xi}\left(\frac{\partial E}{\partial Q}\,\Omega^n\right) + \frac{\partial}{\partial \eta}\left(\frac{\partial F}{\partial Q}\,\Omega^n\right)\right] dS\right\}^* \cdot \left\{\iint \left[\frac{\partial}{\partial \xi}\left(\frac{\partial E}{\partial Q}\,\frac{\partial \Omega^m}{\partial \omega_\ell^m}\right) + \frac{\partial}{\partial \eta}\left(\frac{\partial F}{\partial Q}\,\frac{\partial \Omega^m}{\partial \omega_\ell^m}\right)\right] dS\right\} \tag{31}$$

where $\frac{\partial \Omega^m}{\partial \omega_\ell^m} = \{\Delta_k^m \, \delta_{k\ell}\}$ and $\delta_{k\ell}$ is the Kronecker delta.

$$(32)$$

Notice that

$$\frac{\partial E}{\partial Q} \Omega^n = \sum_q^L \omega_q^n \frac{\partial E}{\partial Q} \frac{\partial \Omega^n}{\partial \omega_q^n} \qquad (33)$$

and

$$\frac{\partial F}{\partial Q} \Omega^n = \sum_q^L \omega_q^n \frac{\partial F}{\partial Q} \frac{\partial \Omega^n}{\partial \omega_q^n} \qquad (34)$$

Let

$$A_\ell^m = \iint \left[\frac{\partial}{\partial \xi} \left(\frac{\partial E}{\partial Q} \frac{\partial \Omega^m}{\partial \omega_\ell^m}\right) + \frac{\partial}{\partial \eta} \left(\frac{\partial F}{\partial Q} \frac{\partial \Omega^m}{\partial \omega_\ell^m}\right)\right] dS \qquad (35)$$

Note that A_ℓ^m is not a function of ω's. Then, Eq. (31) becomes

$$\sum_i^I \sum_j^J (r^t)^* A_\ell^m = \sum_i^I \sum_j^J \sum_n^M \sum_q^L \omega_q^n (A_q^n)^* A_\ell^m \qquad (36)$$

Let

$$C_{q\ell}^{nm} = \sum_i^I \sum_j^J (A_q^n)^* A_\ell^m \qquad (37)$$

and

$$B_\ell^m = \sum_i^I \sum_j^J (r^t)^* A_\ell^m \qquad (38)$$

Then

$$\sum_n^M \sum_q^L \omega_q^n C_{q\ell}^{nm} = B_\ell^m \qquad (39)$$

or

$$\sum^M (\omega_1^n C_{1\ell}^{nm} + \omega_2^n C_{2\ell}^{nm} + \omega_3^n C_{3\ell}^{nm} + \ldots\ldots + \omega_L^n C_{L\ell}^{nm}) = B_\ell^m \qquad (40)$$

resulting in a system of LxM equations for the LxM unknown optimal acceleration factors ω_ℓ^m. In the case of two-dimensional Euler equations, L = 4. Thus, we must solve simultaneously the following system of 4xM equations in order to determine the 4xM optimal values of ω_ℓ^m.

$$\left[\begin{array}{cccc|ccc}
C_{11}^{11} & C_{21}^{11} & C_{31}^{11} & C_{41}^{11} & C_{11}^{21} & \ldots. & C_{41}^{M1} \\
C_{12}^{11} & C_{22}^{11} & C_{32}^{11} & C_{42}^{11} & C_{12}^{21} & \ldots. & C_{42}^{M1} \\
C_{13}^{11} & C_{23}^{11} & C_{33}^{11} & C_{43}^{11} & C_{13}^{21} & \ldots. & C_{43}^{M1} \\
C_{14}^{11} & C_{24}^{11} & C_{34}^{11} & C_{44}^{11} & C_{14}^{21} & \ldots. & C_{44}^{M1} \\
\hline
C_{11}^{12} & C_{21}^{12} & C_{31}^{12} & C_{41}^{12} & C_{11}^{22} & \ldots. & C_{41}^{M2} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
C_{14}^{1M} & C_{24}^{1M} & C_{34}^{1M} & C_{44}^{1M} & C_{14}^{2M} & \ldots. & C_{44}^{M4}
\end{array}\right]
\cdot
\left[\begin{array}{c}
\omega_1^1 \\ \omega_2^1 \\ \omega_3^1 \\ \omega_4^1 \\ \hline \omega_1^2 \\ \vdots \\ \omega_4^M
\end{array}\right]
=
\left[\begin{array}{c}
B_1^1 \\ B_2^1 \\ B_3^1 \\ B_4^1 \\ \hline B_1^2 \\ \vdots \\ B_4^M
\end{array}\right]
\qquad (41)$$

We have decided to combine four consecutive time steps (M=4). Since the twodimensional Euler equations form a system that has four equations (L = 4), these four sequences of four optimal values of ω can then be used in Eq. 23 and Eq. 22 to update the solution to the next global time level t+1.

The matrices $\partial E/\partial Q$ and $\partial F/\partial Q$ that are needed for evaluation of the coefficients in the above matrix are given as:

$$\frac{\partial E}{\partial Q} = \begin{bmatrix} 0 & \xi_x & \xi_y & 0 \\ \frac{\gamma-1}{2}\xi_x(u^2+v^2) - uU & (2-\gamma)\xi_x u+U & \xi_y u-(\gamma-1)\xi_x v & (\gamma-1)\xi_x \\ \frac{\gamma-1}{2}\xi_y(u^2+v^2) - vU & \xi_x v-(\gamma-1)\xi_y u & (2-\gamma)\xi_y v+U & (\gamma-1)\xi_y \\ [(\gamma-1)(u^2+v^2) - \gamma e]U & h\xi_x-(\gamma-1)uU & h\xi_y-(\gamma-1)vU & \gamma U \end{bmatrix} \qquad (42)$$

$$\frac{\partial F}{\partial Q} = \begin{bmatrix} 0 & \eta_x & \eta_y & 0 \\ \frac{\gamma-1}{2}\eta_x(u^2+v^2) - uV & (2-\gamma)\eta_x u+V & \eta_y u-(\gamma-1)\eta_x v & (\gamma-1)\eta_x \\ \frac{\gamma-1}{2}\eta_y(u^2+v^2) - vV & \eta_x v-(\gamma-1)\eta_y u & (2-\gamma)\eta_y v+V & (\gamma-1)\eta_y \\ [(\gamma-1)(u^2+v^2) - \gamma e]V & h\eta_x-(\gamma-1)uV & h\eta_y-(\gamma-1)vV & \gamma V \end{bmatrix} \qquad (43)$$

where h is the specific enthalpy per unit mass and γ in the ratio of specific heats for a calorically perfect gas.

In addition to the computer memory required by the original non-accelerated scheme [1], additional memory is needed to implement the DMR. If the grid points are IxJ and we use M global consecutive time levels to update the solution, then for the two-dimensional problem the extra memory requirement is approximately L x (2+M) x (I-2) x (J-2) and for the three-dimensional Euler equations the extra memory requirement is approximately L x (3+M) x (I-2) x (J-2) x (K-2). In the two-dimensional case this represents approximately 150% increase and in the three-dimensional case this represents approximately 175% increase in memory requirement over the original non-accelerated [1] algorithm.

Three different methods were tested for the boundary conditions on the residuals in the integrals of Eq. 35. The first method was to set the residuals at the ghost cells to be zero. The second method calculates the residuals at the ghost cells from the boundary conditions. The third method extrapolates the residuals from the interior of the flowfield. It was found that the third method gives the best results.

RESULTS

All computations were performed on a VAX 11/8550 computer in a single precision mode. The first sequence of tests of DMR was performed on the internal two-dimensional (L=4) flow problems by combining four consecutive global time steps (M=4). This means that a 16x16 matrix (Eq. 41) needs to be inverted. Figure 1 shows the computational grid for a 10% thick circular bump in a two dimensional channel. The grid size is 65x17 points. The calculations were started with uniform flow and the DMR was applied once after every 30 iterations. Figure 1 shows the convergence histories of subsonic flow calculations with $M_\infty = 0.5$. The number of iterations needed to achieve the same level of residual is reduced almost by 60%. The convergence with the DMR shows smaller oscillations than that of the original [1] scheme. It is expected that this behavior continues to the machine accuracy. The saving in computational time is about 50% for this test case.

The constant pressure contour plots of the entire flow field for both non-accelerated finite volume RRK scheme and DMR accelerated finite volume RRK scheme are shown in Fig. 4 and Fig. 5, respectively. The difference between the two is not discernable in these contour plots thus confirming that DMR method does not adversely influence the quality of the solution.

Results of the second test case are presented in Figs. 6,7,8 and 9. The entire flow field is subsonic with $M_\infty = 0.55$. For this test case, the saving was almost 40% in CPU time. It is noticeable that the convergence history shows more oscillatory behavior than for the case with M = 0.5. Another subsonic ($M_\infty = 0.6$) test case was tested and the results are shown in Figs. 10, 11, 12 and 13 demonstrating that a considerable amount of computation effort was saved.

Figs. 14 and 15 show the convergence histories for the transonic shocked flow case with $M_\infty = 0.675$ which is less than the flow choking Mach number of this channel. Results indicate that with the DMR, the convergence rate is not improved.

Similar trends were observed when solving Euler equations for a flow around a circle. An O-type grid consisting of 64x32 grid cells was used. For a moderately compressible subsonic flow ($M_\infty = 0.3$), DMR saves (Figs. 18 and 19) approximately 45% of CPU time. It generates results (Fig. 20) that are practically indistinguishable from the non-accelerated scheme. When the critical free stream Mach number $M_\infty = 0.4$ was used, Fig. 21 indicates and Fig. 22 confirms that the DMR method in its present form offers practically no gain when compared with the non-accelerated algorithm although the computed surface Mach numbers (Fig. 23) are equally accurate. Thus, both Ni's bump case and circle case indicate that DMR method in its present formulation offers no advantages at transonic speeds. On the other hand, the system of Euler equations becomes stiff as the Mach number decreases, thus rapidly reducing the convergence rate of the non-accelerated scheme. When using $M_\infty = 0.1$ (an almost incompressible flow), Figs. 24 and 25 demonstrate that DMR offers over 70% savings in the CPU time over the non-accelerated scheme. Fig. 26 indicates difference in the computed surface C_p values after 1200 iterations.

In order to account for the different local characteristic behavior of the transonic flow, it should be possible to use different sets of weights for different regions of the flowfield. Also, the artificial dissipation terms could be incorporated in the formulation of the DMR. In addition, the optimal frequency of applying the DMR needs to be investigated. In the present investigations, DMR was applied by combining four consecutive time steps after every thirty time steps.

Notice that all numerical results were obtained without the standard acceleration techniques such as explicit and implicit residual smoothing, enthalpy damping, multigridding and vectorization. These methods could be added to further accelerate the algorithm.

CONCLUSIONS

A conceptually new method termed Distributed Minimal Residual (DMR) has been developed and successfully applied to the acceleration of an explicit finite volume iterative algorithm for the numerical solution of a nonlinear system of Euler equations governing inviscid gasdynamics. The main idea of using a separate sequence of optimal acceleration factors for each of the equations in the system was theoretically formulated a numerically demonstrated with a number of examples. This means that the partial differential equations governing mass, x-momentum, y-momentum and energy conservation were accelerated according to their own separate optimal sequences of acceleration factors that have a common objective of minimizing the global residual of the entire system at each global time level. The method seems to offer significant time savings especially for stiff systems of differential equations.

ACKNOWLEDGEMENTS

This work was supported by the Air Force Office of Scientific Research/Numerical Mathematics Program under the Grant AFOSR-87-0121 supervised by Dr. John P. Thomas, Dr. Arje Nachman and Lt. Col. David Nelson. The manuscript was typed by Ms. Amy Myers. The graphics was produced on the Apple Computer system donated to our Interdisciplinary Computational Fluid Dynamics group.

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 No. 81-1259, Palo Alto, CA, June, 1981.

2. Wambecq, A., "Rational Runge-Kutta Methods for Solving Systems of Ordinary Differential Equations," Computing, Vol. 20, 1978, pp. 333-342.

3. 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.

4. Kennon, S. R., and Dulikravich, G. S., "Optimal Acceleration Factors for Iterative Solution of Linear and Nonlinear Differential Systems," Computer Methods in Applied Mechanics and Engineering, 47, 1984, pp. 357-367.

5. Kennon, S. R., "Optimal Acceleration Factors for Iterative Solution of Linear and Nonlinear Differential Systems," AIAA Paper 85-0162, January 1985, Reno, Nevada.

6. Huang, C. Y., Kennon, S. R., and Dulikravich, G. S., "Generalized Nonlinear Minimal Residual (GNLMR) Method for Iterative Algorithms," Journal of Computational and Applied Mathematics, 16, 1986, pp. 215-232.

7. Huang, C. Y., and Dulikravich, G. S., "Fast Iterative Algorithms Based on Optimized Explicit Time-Stepping," Computer Methods in Applied Mechanics and Engineering, 63, August 1987, pp. 15-36.

8. Caughey, D. A. and Turkel, E., "Effects of Numerical Dissipation on Finite-Volume Solutions of Compressible Flow Problems, AIAA paper 88-0621, Reno, NV, January 1988.

9. MacCormack, R. W. and Baldwin, B. S. "A Numerical Method for Solving the Navier-Stokes Equations With Application to Shock-Boundary Layer Interaction," AIAA paper 75-1, January 1975.

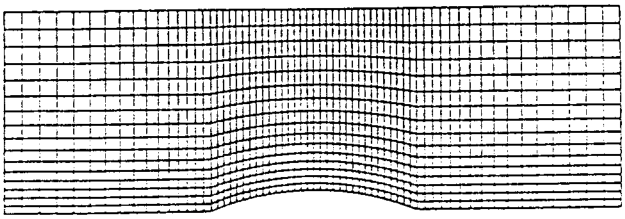

Figure 1. Computational grid for a 10% thick circular arc airfoil on the bottom wall of a straight two-dimensional channel (Ni's bump).

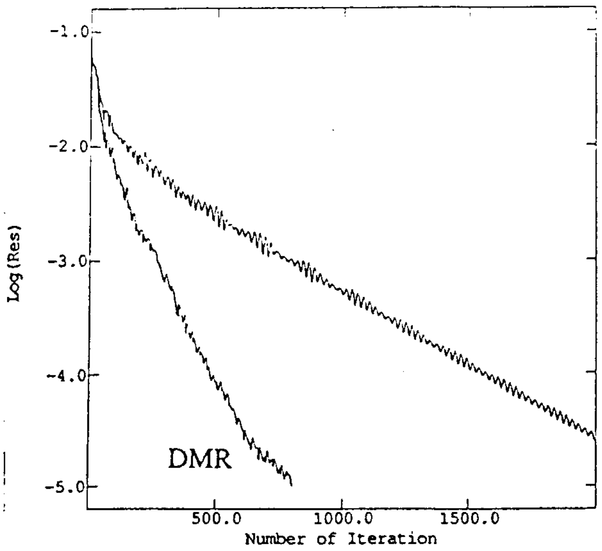

Figure 2. Comparison of convergence rates in terms of iteration numbers: non-accelerated and DMR accelerated algorithm for Ni's bump with $M_\infty = 0.5$.

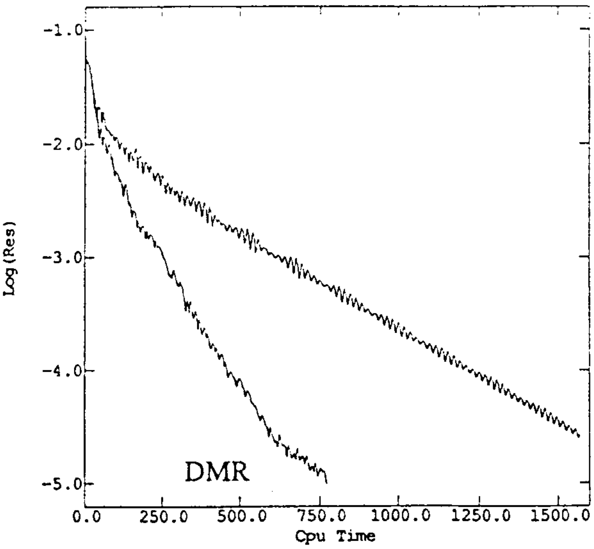

Figure 3. Comparison of convergence rates in terms of the CPU time: non-accelerated and DMR accelerated algorithm for Ni's bump with $M_\infty = 0.5$.

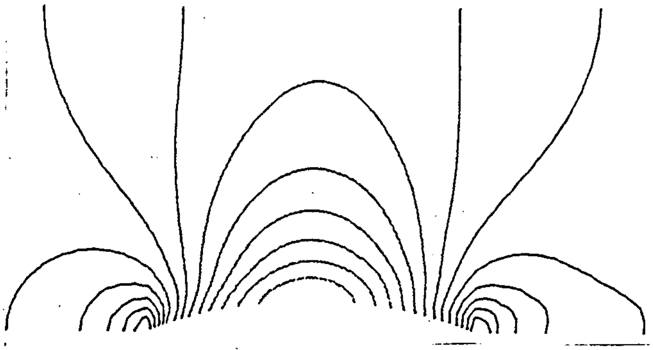

Figure 4. Constant pressure contours for non-accelerated algorithm for Ni's bump with $M_\infty = 0.5$.

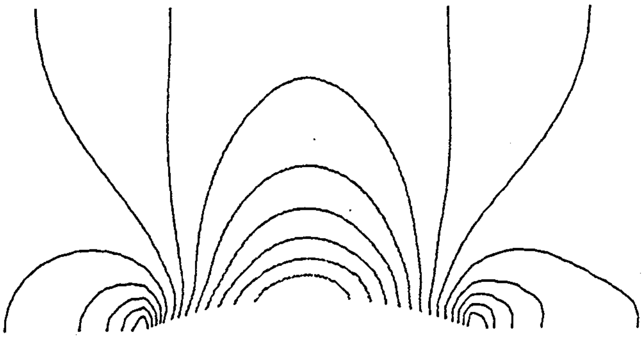

Figure 5. Constant pressure contours for DMR accelerated algorithm for Ni's bump with $M_\infty = 0.5$.

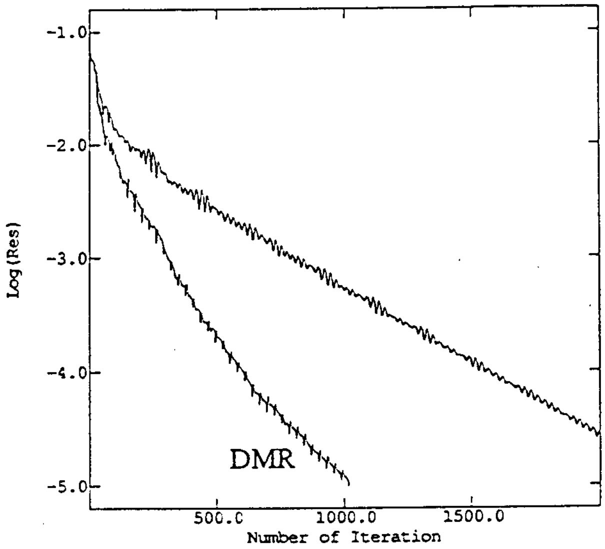

Figure 6. Comparison of convergence rates in terms of interation numbers: non-accelerated and DMR accelerated algorithm for Ni's bump with $M_\infty = 0.55$.

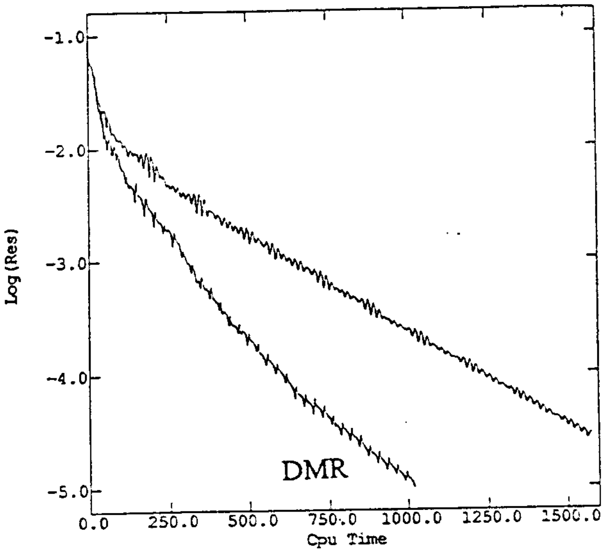

Figure 7. Comparison of convergence rates in terms of the CPU time: non-accelerated and DMR accelerated algorithm for Euler Ni's bump with $M_\infty = 0.55$.

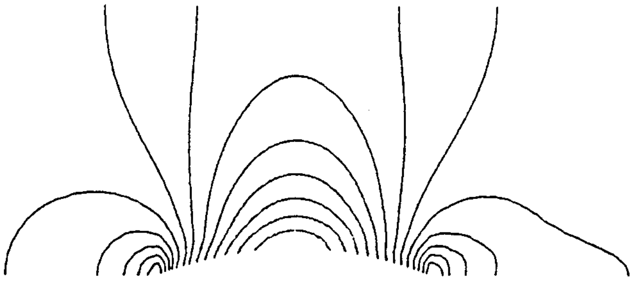

Figure 8. Constant pressure contours for non-accelerated algorithm for Ni's bump with $M_\infty = 0.55$.

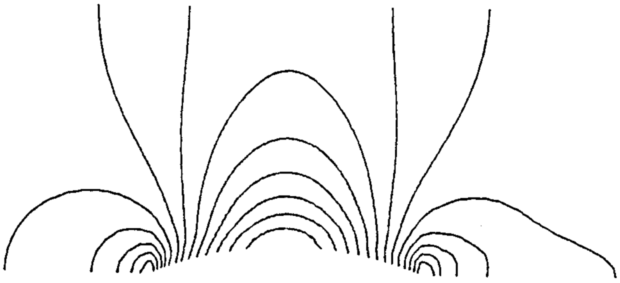

Figure 9. Constant pressure contours for DMR accelerated algorithm for Ni's bump with $M_\infty = 0.55$.

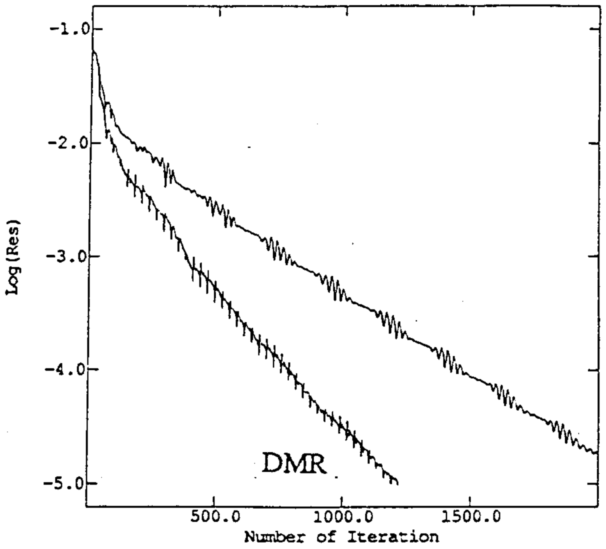

Figure 10. Comparison of convergence rates in terms of iteration numbers: non-accelerated and DMR accelerated algorithm for Ni's bump with $M_\infty = 0.6$.

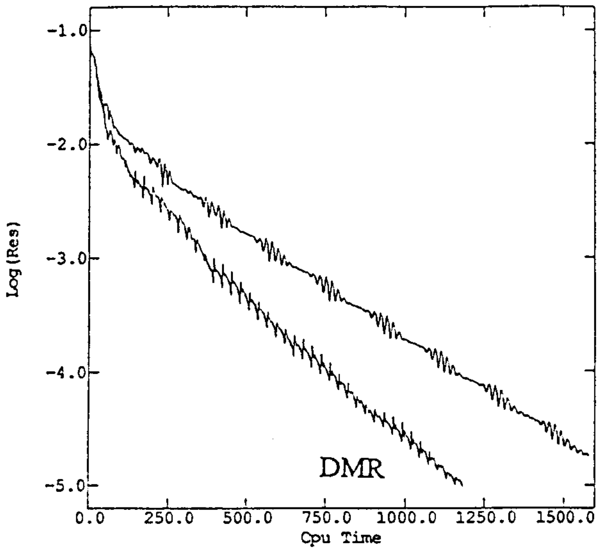

Figure 11. Comparison of convergence rates in terms of the CPU time: non-accelerated and DMR accelerated algorithm for Ni's bump with $M_\infty = 0.6$.

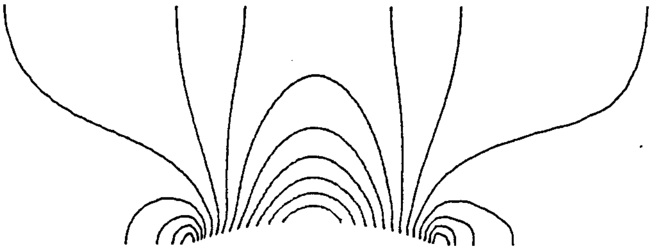

Figure 12. Constant pressure contours for non-accelerated algorithm for Ni's bump with $M_\infty = 0.6$.

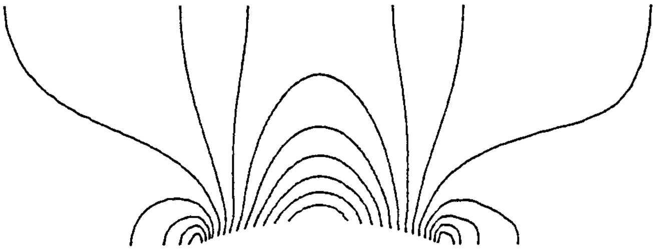

Figure 13. Constant pressure contours for DMR accelerated algorithm for Ni's bump with $M_\infty = 0.6$.

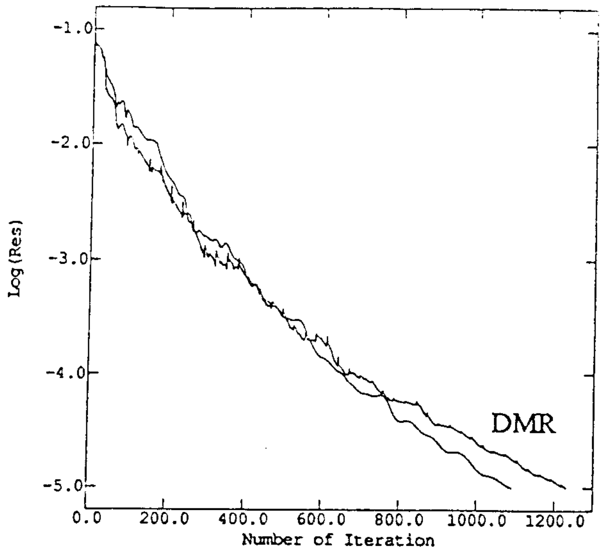

Figure 14. Comparison of convergence rates in terms of iteration numbers: non-accelerated and DMR accelerated algorithm for Ni's bump with M_∞= 0.675.

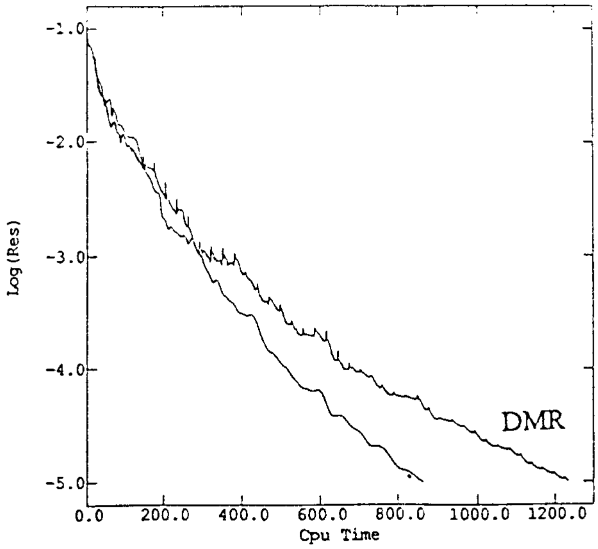

Figure 15. Comparison of convergence rates in terms of the CPU time: non-accelerated and DMR accelerated algorithm for Ni's bump with M_∞= 0.675.

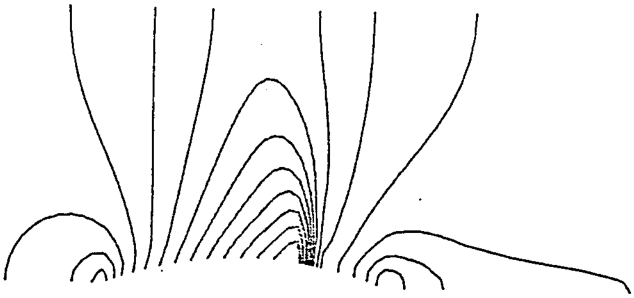

Figure 16. Constant pressure contours for non-accelerated algorithms for Ni's bump with M_∞= 0.675.

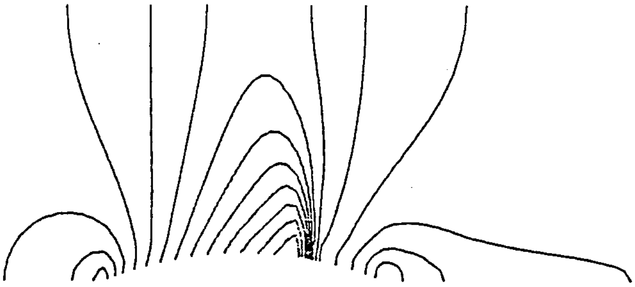

Figure 17. Constant pressure contours for non-accelerated algorithm for Ni's bump with M_∞= 0.675.

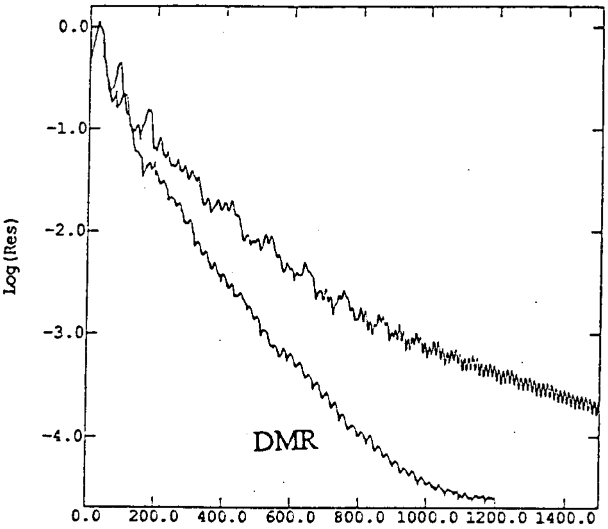

Figure 18. Comparison of convergence rates in terms of iteration numbers: non-accelerated and DMR accelerated algorithm for circle with M_∞= 0.3.

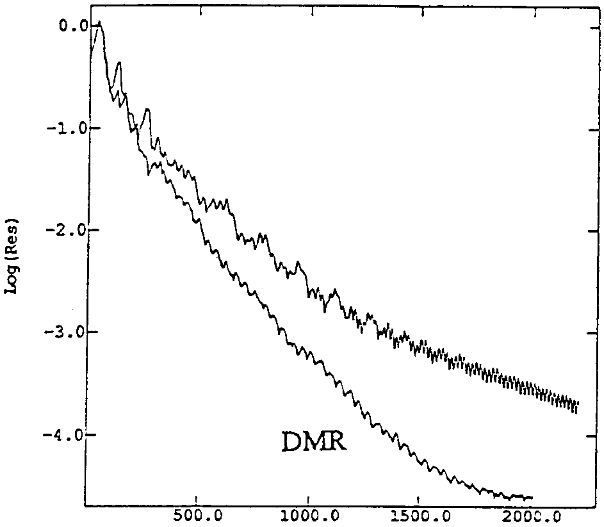

Figure 19. Comparison of convergence rates in terms of the CPU time: non-accelerated and DMR accelerated algorithm for circle with M_∞= C.3.

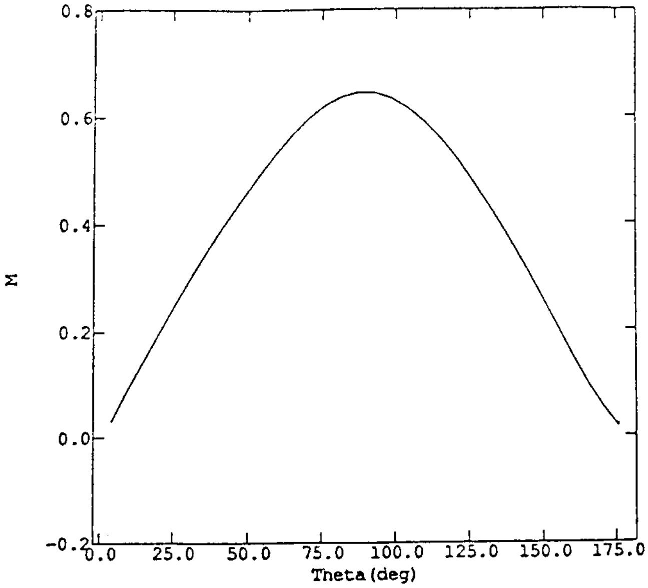

Figure 20. Surface Mach numbers for non-accelerated and DMR accelerated algorithm for circle with $M_\infty = 0.3$.

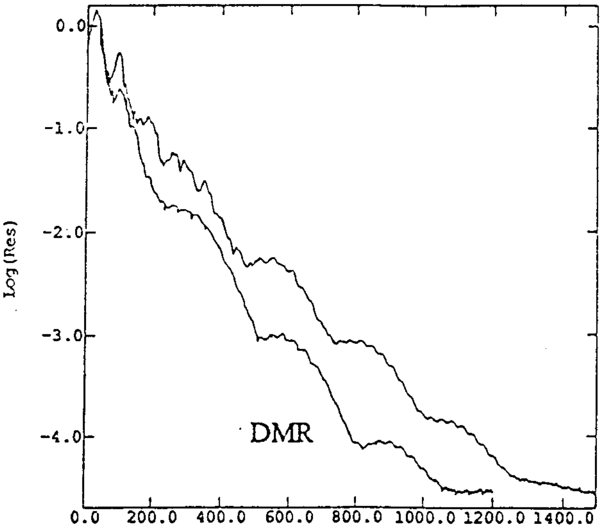

Figure 21. Comparison of convergence rates in terms of iteration numbers: non-accelerated and DMR accelerated algorithm for circle with $M_\infty = 0.4$.

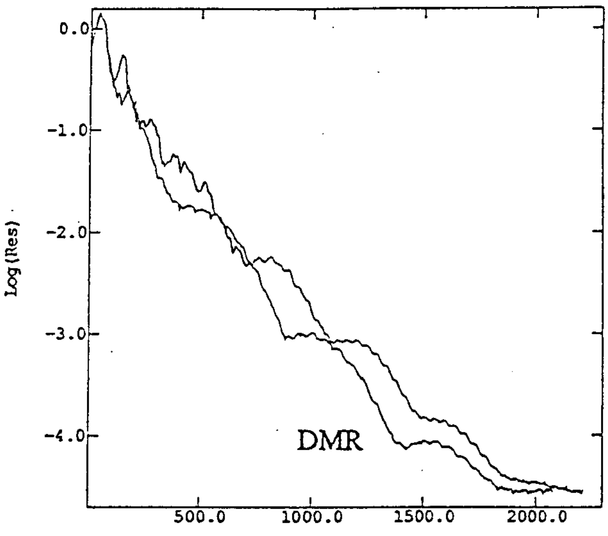

Figure 22. Comparison of convergence rates in terms of the CPU time: non-accelerated and DMR accelerated algorithm for circle with $M_\infty = 0.4$.

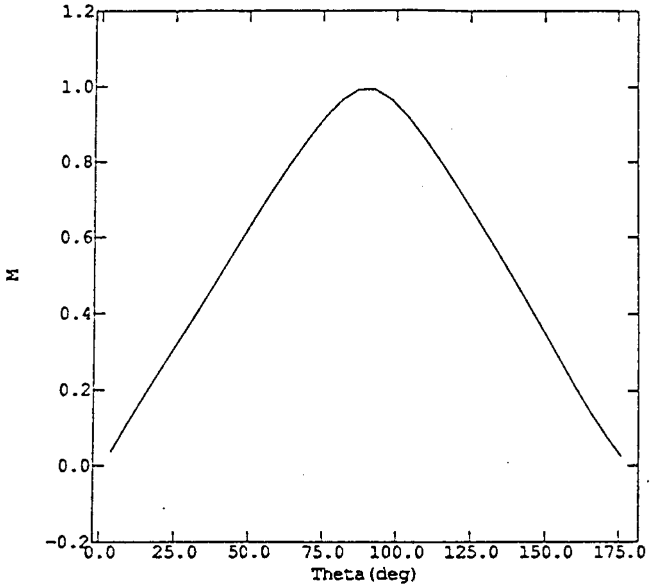

Figure 23. Surface Mach numbers for non-accelerated and DMR accelerated algorithm for circle with $M_\infty = 0.4$.

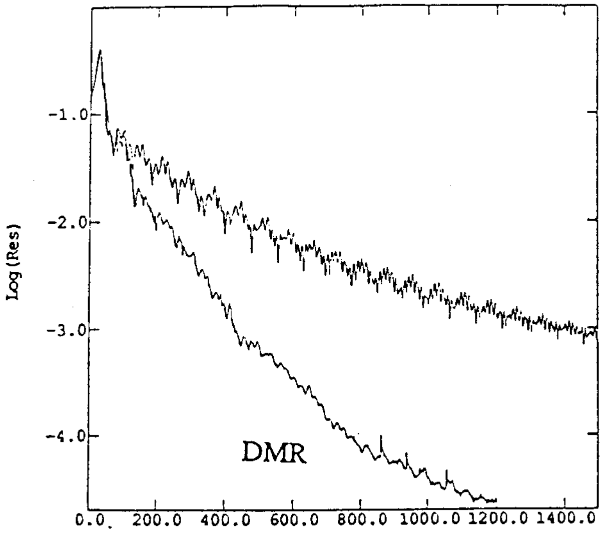

Figure 24. Comparison of convergence rates in terms of iteration numbers: non-accelerated and DMR accelerated algorithm for circle with M_{∞}= 0.1.

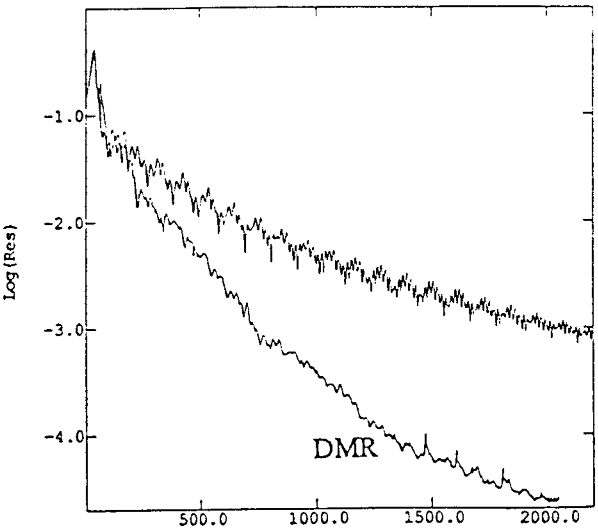

Figure 25. Comparison of convergence rates in terms of the CPU time: non-accelerated and DMR accelerated algorithm for circle with M_{∞}= 0.1.

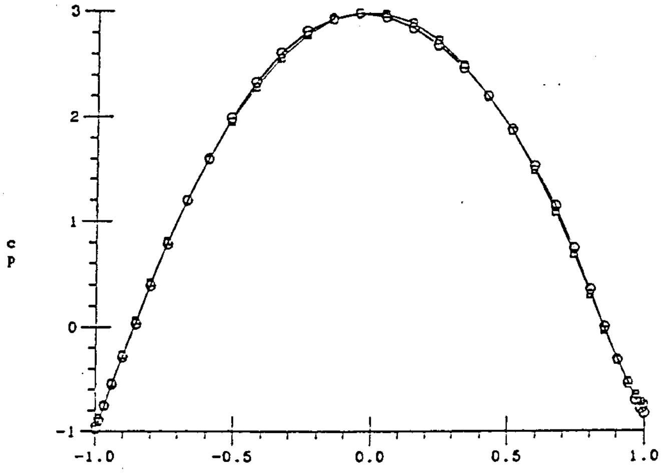

Figure 26. Comparison of surface pressure coefficients: non-accelerated (-□-□-□-) and DMR (-ө-ө-ө-ө-) accelerated algorithm for circle with M_{∞}= 0.1.