AIAA-91-0241

Performance Analysis of DMR Method for Accelerations of Iterative Algorithms

S. Lee and G. Dulikravich
The Pennsylvania State University
University Park, PA

29th Aerospace Sciences Meeting
January 7-10, 1991/Reno, Nevada

PERFORMANCE ANALYSIS OF DMR METHOD FOR ACCELERATIONS OF ITERATIVE ALGORITHMS

Seungsoo Lee* and George S. Dulikravich**
Pennsylvania State University,
Department of Aerospace Engineering
University Park, PA 16802, U.S.A.

Abstract

Using incompressible stagnation, viscous flow as a testbed, a comparative analysis of the performance of the Distributed Minimal Residual (DMR) method was conducted. The DMR method was applied to explicit Runge-Kutta time-stepping and to the implicit Euler integration scheme in an artificial compressibility code with and without implicit residual smoothing. Parameters that were varied included: computational grid refinement, clustering, and skewness, CFL number, Reynolds number, artificial compressibility coefficient, frequency of application of the DMR, and a number of consecutive corrections combined. In all of the tests the DMR method demonstrated a consistent ability to reduce the number of iterations and the CPU time required. The DMR was also found to enhance the stability of the basic algorithms.

Introduction

The DMR method is a new scheme for improving convergence rates and stability of iterative algorithms used for the numerical integration of arbitrary systems of partial differential equations. The DMR method belongs to a class of time extrapolation algorithms[1] where each of the L equations in a system is iteratively updated using a separate sequence of weighting factors, ω, that are multiplying corrections, δq, from M preceding consecutive iterations.

$$q_1^{n+1} = q_1^n + \omega_1^n \delta q_1^n + \omega_1^{n-1} \delta q_1^{n-1} + \cdots + \omega_1^M \delta q_1^M$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

$$q_L^{n+1} = q_L^n + \omega_L^n \delta q_L^n + \omega_L^{n-1} \delta q_L^{n-1} + \cdots + \omega_L^M \delta q_L^M \qquad (1)$$

Here, superscript n designates the iteration level, while q are the variables. Thus, in the case of an iterative algorithm for the solution of a general system of partial differential equations of the form (in two-dimensional space for the sake of simplicity)

$$\frac{\partial Q}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} = 0 \qquad (2)$$

the local residual, R, at iteration level n+1 is given by

$$R_{ij}^{n+1} = \frac{\partial E_{ij}^{n+1}}{\partial \xi} + \frac{\partial F_{ij}^{n+1}}{\partial \eta} - D^2 (J\, Q_{ij}^{n+1}) + D (J\, Q_{ij}^{n+1}) \qquad (3)$$

Here Q_{ij}^{n+1} is the solution vector $Q_{ij}^{n+1} = Q_{ij}^{n+1}(q_1^{n+1}, q_2^{n+1}, \ldots, q_L^{n+1})_{ij}$, J is the Jacobian of a geometric transformation $\partial(\xi,\eta)/\partial(x,y)$, E_{ij}^{n+1} and F_{ij}^{n+1} are the components of the convective flux vector, $D^2 (J\, Q_{ij}^{n+1})$ is the dissipation and $D (J\, Q_{ij}^{n+1})$ is the artificial dissipation. Assume that the solution at the next iteration level n+1 is extrapolated from the previous M consecutive iteration levels. Then, we can say that

$$Q_{ij}^{n+1} = Q_{ij}^n + \sum_{m=1}^{M} \theta_{ij}^m \qquad (4)$$

where

---

* Postdoctoral Fellow. Member AIAA. Presently with Korean Defence Agency, Daejon, S. Korea.

** Associate Professor. Senior Member AIAA

$$\theta_{ij}^{m} = \begin{bmatrix} \omega_1^m \delta q_1^m \\ \omega_1^m \delta q_1^m \\ \vdots \\ \omega_1^m \delta q_1^m \end{bmatrix}_{ij} \quad (5)$$

Here, ω's are the acceleration (weighting or relaxation factors) to be calculated, δq^m are the corrections computed with the original non-accelerated scheme during each of the consecutive iterations, M denotes the total number of consecutive iteration levels combined, and L denotes the total number of equations in the system.

Optimal values of the weighting factors, ω_1^m, are determined from the condition that the L2-norm of the future global residual is simultaneously minimized with respect to each value of ω_1^m.

Using Taylor series expansion in artificial time, n, and neglecting all terms that are higher than first order in δn, one gets approximately that

$$R_{ij}^{n+1} = R_{ij}^{n} + \sum_{m=1}^{M} \left[\frac{\partial}{\partial \xi} A_{ij\cdot}^{n} + \frac{\partial}{\partial \xi} B_{ij\cdot}^{n} - D^2 J_{ij\cdot} + D J_{ij\cdot} \right] \theta_{ij}^{m} \quad (6)$$

The global residual for the entire domain can be defined as

$$\bar{R}^{n+1} = \sum_{ij} (R_{ij}^{n+1}) \cdot (R_{ij}^{n+1})^T \quad (7)$$

where the superscript T designates transpose of a vector, so that minimization of $\bar{R}^{n+1}$ leads to $\frac{\partial \bar{R}^{n+1}}{\partial \omega_r^m} = 0$, that is,

$$\sum_{n}^{M} \sum_{q}^{L} \omega_q^n c_{qr}^{nm} = b_r^m \quad (8)$$

representing the system of M×L linear algebraic equations for the L sets of M optimum acceleration factors ω. Coefficients c and b are known, since they are functions of the corrections from M previous iterations. Let us say that we want to combine corrections form M = 3 consecutive iterations to extrapolate the solution in an iterative algorithm for two-dimensional incompressible Navier-Stokes equations with heat transfer. In this case we need to solve M×L = 4×3 = 12 algebraic equations for 4×3 = 12 values of ω after every 10 iterations of the basic non-accelerated algorithm.

Earlier publications contain details of the DMR method as applied to Euler equations of compressible, inviscid flow[2,3,4] and to Navier-Stokes equations of incompressible, laminar flow without [5,6] and with[7-11] heat transfer. The purpose of this paper is to present an exhaustive and objective evaluation of the performance of the DMR method as applied to explicit and implicit algorithms for the solution of Navier-Stokes equations.

Test Cases

A stagnation-point steady viscous flow (Hiemenz flow) normal to a solid wall (Fig. 1a) was chosen as the test case, since the analytic solution to the Hiemenz flow is known[12], and the accuracy of the artificial compressibility[13] codes (the explicit Runge-Kutta method and the Euler implicit method) can be verified. This test case was used to examine the effectiveness of the DMR method and its sensitivity to a variety of flow and grid conditions. The inflow velocity components (u and v at the top boundary, j = jmax) and the pressure at both side boundaries (i = imin and i = imax) were specified by the values from the analytic solution to the Hiemenz flow. The remaining variables along the boundaries were evaluated by using characteristic formulation[14]. The flow corresponding to a Reynolds number of 400 based on the free stream velocity and the domain height[12] was computed with and without the DMR method applied to explicit and implicit integration schemes. The clustered computational grid consisted of 60×29 cells (Fig. 1b) with maximum aspect ratio AR_{max} = 6.65 at the bottom wall.

In case of the explicit Runge-Kutta (RK) method, the maximum allowable CFL number of 2.8 was used, while the von Neumann number was σ = 0.4. The Implicit Residual Smoothing (IRS) method[15] used CFL = 5.0. Smoothing parameter θ = 1.0 was used in both x and y direction. A small amount of fourth order artificial dissipation (ε = 0.05) was added to obtain a smooth solution[16]. Using numerical experimentation it was found that the optimal artificial compressibility coefficient for this test case was β = 2, and that the fastest convergence is obtained when the DMR method is applied every 10 iterations by combining corrections from 3 consecutive iterations.

A first order Euler time integration implicit code was also exercised for the test case of the Hiemenz flow with the same flow and grid conditions as in the test case for the explicit code, except that CFL = 10.0 and ε = 0.25 were used. The optimal value of the artificial compressibility coefficient was found by numerical experiments to be β = 5. The DMR method was found to give the fastest convergence when applied every 5 iterations by combining weighted corrections from 4 consecutive

iterations. Figure 2 shows the computed results (using both explicit and implicit codes), demonstrating an excellent agreement with the analytic solutions.

Performance Analysis

The convergence properties of the artificial compressibility method and the DMR method depend also on the artificial compressibility coefficient, β. To find the optimal artificial compressibility coefficient for this test case, a series of computations were made. The Reynolds number was kept at Re = 400 and the same grid was used where AR_{max}= 6.65. The RK method was tested with the DMR and with the IRS. The results are summarized in Fig. 3 and Fig. 4. The number of iterations and the CPU time are the values needed for the initial residual to be reduced by 10 orders of magnitude on CRAY-YMP computer. Both non-accelerated schemes show high dependency on the artificial compressibility coefficient, β. When combined with the DMR method, they are considerably less sensitive to the artificial compressibility factor β. For the entire range of β, it was found that the DMR method gives better convergence rates showing 25-50% reduction in CPU time with the RK method and 25-60% reduction in CPU time with the Euler implicit method. The combination of the DMR method and the IRS was also applied, but failed to give further reduction in CPU time.

The success of the DMR method also depends on how often the DMR method is applied and how many consecutive solutions are combined each time. Tables 1 and 2 indicate that the best convergence rates for this test case were obtained by applying the DMR method every 10 iterations with weighted corrections from 2 consecutive iterations combined for the RK method, while application of the DMR method every 5 iterations with weighted corrections form 4 consecutive iterations combined gives the best results for the Euler implicit scheme. Due to the progressive increase in the memory size, combining corrections from more than 5 consecutive iterations was not considered here. The optimal values of the artificial compressibility coefficient, β, were used for both explicit and implicit methods.

The dependency of the DMR method on the various flow conditions and grid conditions were explored through a number of computations. To isolate the effect of the DMR method, the artificial compressibility coefficients, β, were kept at their optimal values for both the explicit and the implicit method. The DMR method was applied by using the optimal frequency of the DMR applications and the optimal number of combined consecutive solutions for both basic methods. The other parameters, such as CFL number and von Neumann number, were kept the same as in the previous tests. The Reynolds number of 400 and maximum grid aspect ratio AR_{max} = 6.65 were used as basic values, unless specifically mentioned for each test case. For the sake of comparison, the IRS method was tested along with the RK method. The purpose of these series of tests was to show the dependency of performance of the DMR method on each of these parameters.

The equivalent speed of sound, a, in incompressible flows is a function of not only the solution, but also the grid conditions.

$$a = \sqrt{U^2 + \beta(\xi_x^2 + \xi_y^2)} \qquad (9)$$

Here U is the contravariant velocity component normal to a constant ξ grid line, and ξ_x, ξ_y are metric derivatives. It is expected that the convergence behavior is also significantly affected by the grid clustering and grid skewness. First, the performance of the DMR method was examined as a function of the grid clustering toward the bottom wall. Figures 5 and 6 show the number of iterations and the actual CPU time required by the RK method and by the Euler implicit method to reduce the initial residual by 10 orders of magnitude as a function of the grid aspect ratio. As expected, the basic codes suffer from the rapidly deteriorating convergence rates with the increase in grid clustering. The effectiveness of the DMR method in conjunction with the RK method deteriorates also with the exceeding grid clustering. Compared to the performance of the DMR method, the IRS offers less savings. Actually, the IRS fails to converge with a highly clustered grid, thus showing high sensitivity to grid clustering. The reduction of the sensitivity of the IRS to grid clustering could be attempted by allowing the residual smoothing parameter θ to vary throughout the domain[17]. In the present study, however, this cumbersome and unreliable approach was not adopted. On the other hand, the DMR method gives consistent acceleration of the Euler implicit method in the range of grid aspect ratios where the Euler implicit method converges, although the number of required iterations rapidly increases with strong grid clustering.

The skewness (non-orthogonality) of a computational grid also influences the convergence behavior of an iterative scheme, since the equivalent speed of sound depends on the grid skewness as well as the grid clustering. The sensitivity of the DMR method to grid skewness was examined by inclining vertical grid lines shown in Fig. 1. The grid skew angle, φ, is defined as the angle between the vertical and the ξ = constant grid lines. As can be seen in Figures 7 and 8, both explicit and implicit schemes are less sensitive to the grid non-orthogonality than to the grid clustering. The DMR method with the RK method gives consistently superior convergence

rates compared to the IRS method. For the skew angle larger than $\varphi = 10°$ the IRS method fails to converge, while the combination of the DMR method with the IRS converges until $\varphi = 15°$, indicating the stabilizing effects of the DMR method. When applied alone, the DMR method successfully accelerates both the RK and the Euler implicit method for skew angles as large as $\varphi = 30°$, offering 35-55% reduction in CPU time.

In general, the convergence rates of iterative algorithms vary as the number of grid cells increase. Three levels of grid were used to examine the effects of the grid refinement. The grids consisted of 30x15, 60x30, and 120x60 cells, and were generated with the same clustering function[18]. The results are summarized in Fig. 9 and 10. Note that the vertical axes of these two plots use logarithmic scale, while the horizontal axis is the ratio of the average size of the grid cell to the average size of the coarse grid cell. The DMR method shows the same general trend as the RK method, the IRS, and the Euler implicit method. The results of the DMR method with the Euler implicit method on the finest grid were obtained by applying the DMR method every 10 iterations with weighted corrections from 3 consecutive iterations combined, since the DMR method failed to enhance the convergence rate on the finest grid when applied every 5 iterations with corrections from 4 consecutive iterations combined. This demonstrates the importance of the choice of frequency of the DMR applications and the number of consecutive iterations combined.

A series of test runs were made in order to show how the DMR method behaves as the Reynolds number varies. Figures 11 and 12 summarize the results. These figures show the number of iterations and the CPU time required to reduce the initial residual by 10 orders of magnitude. Both explicit and implicit methods show best convergence when the Reynolds number is approximately 500. This behavior of the basic schemes can be understood by noticing that the non-skewed grid clustering was optimal for this Reynolds number. The Euler implicit scheme appears to be more sensitive to the variation of the Reynolds number than the RK method. While IRS is more costly than the DMR method and diverges for Re > 500, the DMR method maintains stability and gives 50-70% reduction in CPU time.

The performance of the RK, the Euler implicit, and the DMR method varies also as the CFL number varies. Figures 13 and 14 show that the DMR method accelerates the basic schemes over the entire range of the CFL numbers used. It is noticeable that the IRS converges slower when compared with the RK method at the same CFL number. For CFL numbers greater than 12, the non-accelerated Euler implicit method fails to converge. When combined with the DMR method it converges. For both explicit and implicit schemes, application of the DMR method decreases sensitivity to the choice of the CFL number.

The accuracy of the schemes was examined on the three levels of computational grid after the residual was reduced by 10 orders of magnitude. The first column of the Table 3 and Table 4 shows the ratio of the average length of the grid cell to the average length of the cell on the coarse grid. The second column shows the L-2 norm of the combined relative error, and the third column shows the L-2 norm of the point-averaged error of the schemes. As can be seen in Tables 3 and 4, both the RK method and the Euler implicit method show the convergence characteristics typical of the first order schemes, even though the second order spatial differencing was used for both schemes in the computational domain. This can be understood by noticing that the first order boundary treatment was used.

Conclusions

Based on the extensive testing of the DMR method using viscous incompressible Hiemenz flow as an example, the following conclusions can be drawn. The DMR method performs equally well with explicit and implicit algorithms. It consistently provides reduction in the number of iterations and the computing time, while requiring additional computer memory for corrections from up to four previous consecutive iterations. The method is capable of significantly increasing reliability of the basic algorithms by reducing their sensitivity to the choice of various user specified flow field parameters and the computational grid quality.

Acknowledgments

Authors are grateful for the free use of NAS facility CRAY computer time provided by Dr. Robert Stubbs of NASA Lewis Research Center and Dr. George S. Deiwert of NASA Ames Research Center. Graphics was generated on the equipment donated by the Apple Computer, Inc.

References

1. Huang, C. Y., Dulikravich, G. S and Kennon, S. R., "Generalized Non-Linear Minimal Residual (GNLMR) Method for Iterative Algorithms," Journal of Comput. and Applied Math., Vol. 16, Nov. 1986, pp. 215-232.
2. Lee, S., Dulikravich, G. S. and Dorney, D., "Distributed Minimal Residual (DMR) Method for Explicit Algorithms Applied to Nonlinear Systems," presented at the Conference on Iterative Methods for Large Linear Systems, Austin, Texas, Oct. 19-21, 1988.
3. Dulikravich, G. S., Dorney, D. J. and Lee, S.,

"Iterative Acceleration and Physically Based Dissipation for Euler Equations of Gasdynamics," Proceedings of ASME WAM'88, Symposia on Advances and Applications in Computational Fluid Dynamics, edited by O. Baysal, FED - Vol. 66, 1988, pp. 81-92.

4. Lee, S., Dulikravich, G. S. and Dorney, D. J., "Acceleration of Iterative Algorithms for Euler Equations of Gasdynamics," AIAA Paper 89-0097, Reno, NV, Jan. 1989; Also, AIAA Journal, Vol. 28, No. 5, May 1990, pp. 939-942.
5. Lee, S. and Dulikravich, G. S., "Accelerated Computation of Viscous, Steady Incompressible Flows," ASME paper 89-GT-45, Gas Turbine and Aeroengine Congress and Exposition, Toronto, Canada, June 4-8, 1989.
6. Lee, S. and Dulikravich, G. S., "A Fast Iterative Algorithm for Incompressible Navier-Stokes Equations," Proceedings of the 10th Brazilian Congress of Mechanical Engineering, Rio de Janeiro, Dec. 7-10, 1989.
7. Lee, S. and Dulikravich, G. S., "Computer Simulation of Convective Cooling Effectiveness Inside Turning Passages," Proceedings of the 16th Northeast Bioengineering Conference, ed. R. Gaumount, Penn State University, PA, March 26-27, 1990.
8. Lee, S. and Dulikravich, G.S.,"Acceleration of Iterative Algorithms for Incompressible Navier-

pp. 12-26.

14. Chakravarthy, S. R., "Euler Equations - Implicit Schemes and Implicit Boundary Conditions," AIAA paper 82-0228, AIAA 20th Aerospace Sciences Meeting, Orlando, Florida, Jan. 11-14, 1982.
15. Jameson, A., Schmidt, W., and Turkel, E., "Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Scheme," AIAA paper 81-1259, Palo Alto, CA, June, 1981.
16. Steger, J. L. and Kutler, P., "Implicit Finite-Difference Procedure for the Computation of Vortex Wakes," AIAA Journal, Vol. 15, No. 7, July 1977, pp. 581-590.
17. Martinelli, L, Computations of Viscous Flows with a Multigrid Method, Ph. D. Thesis, The Department of Mechanical and Aerospace Engineering, Princeton University, 1987.
18. Anderson, D. A., Tannehill, J. C. and Pletcher, R. H., Computational Fluid Mechanics and Heat Transfer, McGraw-Hill Book Company, New York, New York, 1984.

Mountain Conference on Iterative Methods, ed. Manteuffel, Copper Mountain, Colorado, April 1-5, 1990.

9 Lee, S. and Dulikravich, G.S.,"Distributed

Number of Iterations (non-accelerated RK method) I0 = 4005
CPU Time (non-accelerated RK method) T0 = 74.71 seconds

	M = 1	M = 2	M = 3	M = 4	M = 5

Minimal Residual (DMR) Method for Acceleration of Iterative Algorithms", Proceedings of the CFD Symposium in Aeropropulsion, ed. M.-S. Liu, NASA Lewis Research Center, Cleveland, OH, April 24-26, 1990; Also to appear in Computational Methods in Applied Mechanics and Engineering.

10. Lee, S. and Dulikravich, G.S.,"Accelerated Computation of Viscous Incompressible Flows With Heat Transfer", Proceedings of the First International Symposium on Experimental and Computational Aerothermodynamics of Internal Flows , ed. N.-X. Chen, Beijing, P.R. China, July 8-12, 1990; Also, to appear in Numerical Methods in Heat Transfer: Fundamentals.
11. Lee, S.,"Acceleration of Iterative Algorithms for Euler and Navier-Stokes Equations", Ph.D. Dissertation, Penn State University, Dept. of Aerospace Eng., University Park, PA, May 1990.
12. Panton, R. L., Incompressible Flow, John Wiley and Sons, Inc., New York, New York, 1984.
13. Chorin, A. J., "A Numerical Method for Solving Incompressible Viscous Flow Problems," Journal of Computational Physics, Vol. 2, 1967,

	Iter.	CPU	Iter.	CPU	Iter.	CPU	Iter.	CPU	Iter.	CPU
10	2441	46.8	1631	33.0	1571	34.0	1442	33.7	1423	36.5
20	2801	53.1	2282	44.4	2281	44.4	2021	40.7	1681	37.2

Table 1 Number of iteration steps combined and frequency of application of the DMR method with the RK method (Left-hand side column indicates number of non-accelerated iterations between applications of the DMR method)

Number of Iterations (non-accelerated Euler implicit method) I0 = 458 CPU Time (non-accelerated Euler implicit method) T0 = 197.45 seconds

	1		2		3		4		5	
	Iter.	CPU	Iter.	CPU	Iter.	CPU	Iter.	CPU	Iter.	CPU
5	385	166.1	215	93.3	196	85.6	187	82.4	186	82.8
10	415	178.8	248	107.1	237	102.7	211	91.8	209	91.4

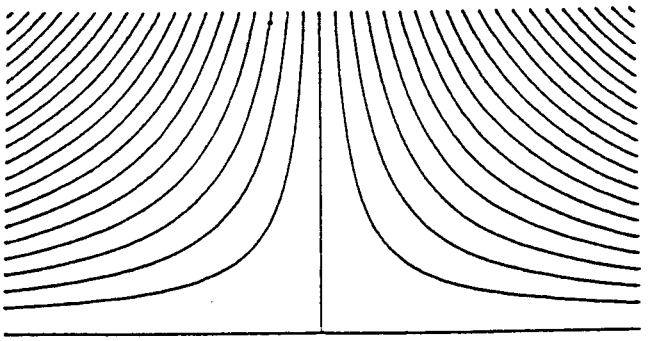
Table 2 Number of iteration steps combined and frequency of application of the DMR method with the Euler implicit method (Left-hand side column indicates number of non-accelerated iterations between applications of the DMR method)

$\frac{\Delta x}{\Delta x_0}$	$\frac{\|\| Q - Q_e \|\|_2}{\|\| Q_e \|\|_2}$	$\frac{\|\| Q - Q_e \|\|_2}{\text{number of grid points}}$
1.0	3.71×10^{-03}	1.29×10^{-04}
0.5	8.84×10^{-04}	1.54×10^{-05}
0.25	2.19×10^{-04}	1.92×10^{-06}

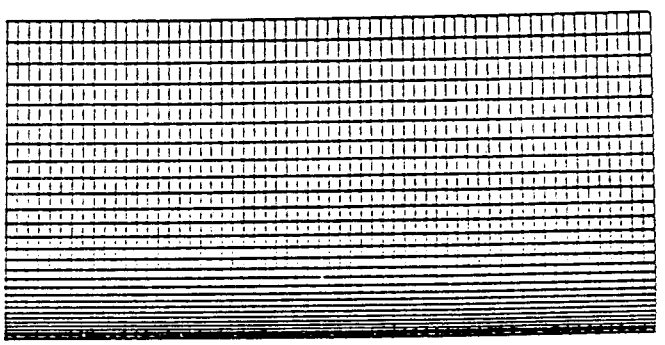
Table 3 Convergence test for the Runge-Kutta time stepping method

$\frac{\Delta x}{\Delta x_0}$	$\frac{\|\| Q - Q_e \|\|_2}{\|\| Q_e \|\|_2}$	$\frac{\|\| Q - Q_e \|\|_2}{\text{number of grid points}}$
1.0	3.43×10^{-03}	1.20×10^{-04}
0.5	8.35×10^{-04}	1.46×10^{-05}
0.25	2.24×10^{-04}	1.96×10^{-06}

Table 4 Convergence test for the Euler implicit method

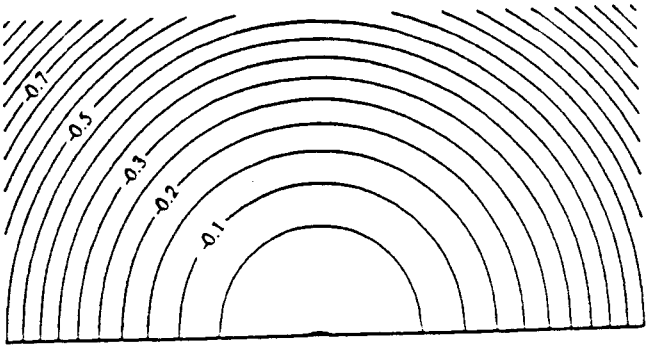


(a) Stream line patterns for Hiemenz flow (analytic solution)

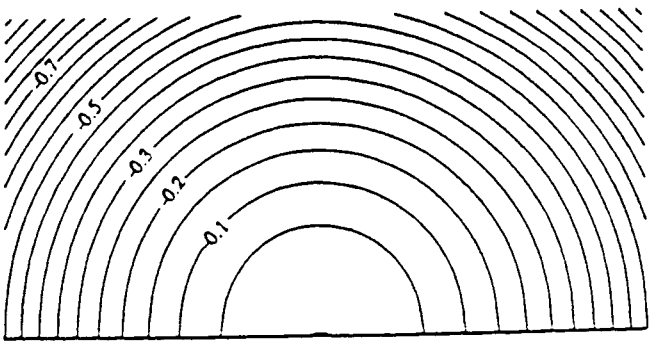


(b) Computational grid

Fig. 1 An H-type computational grid of 60x29 cells used for Hiemenz flow computations (AR = 6.65)

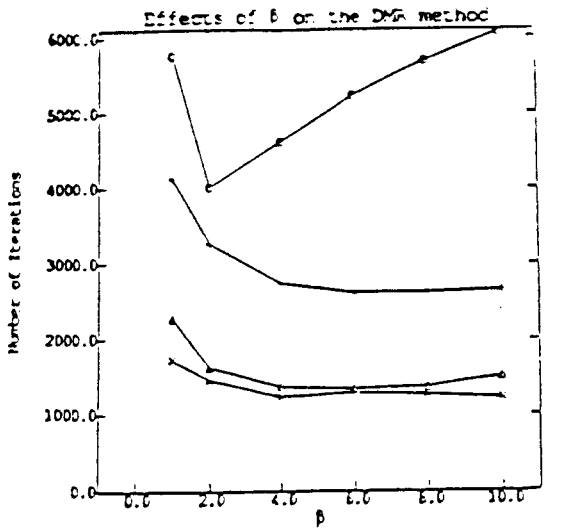


(a) Isobars (RK method)



(b) Isobars (Euler implicit method)

Fig. 2 Computed solutions for Hiemenz flow (Re = 400)



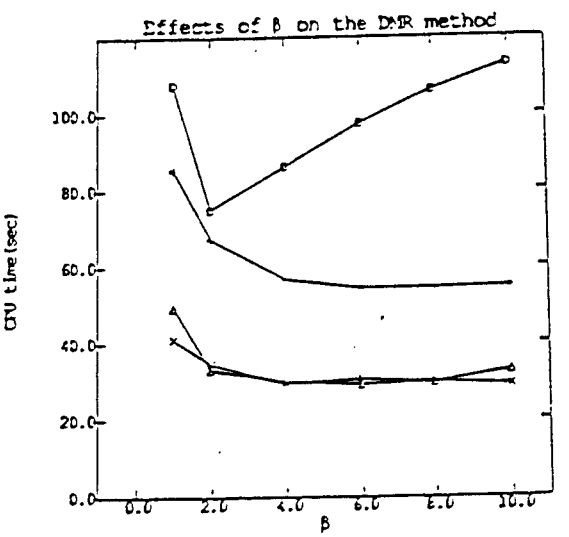
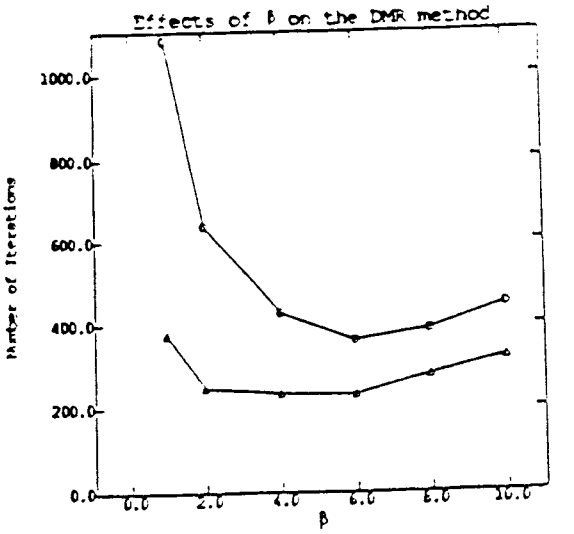


Fig. 3 Effects of the artificial compressibility coefficient on the convergence (RK; -o-o-, DMR; -Δ-Δ-, IRS; -+-+-, DMR+IRS; -x-x-)



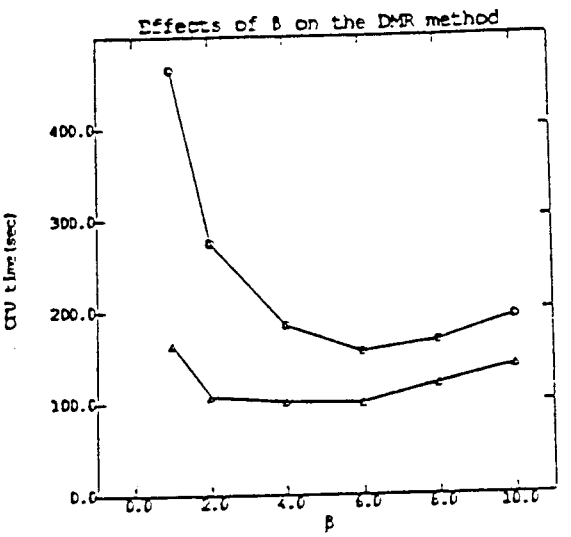
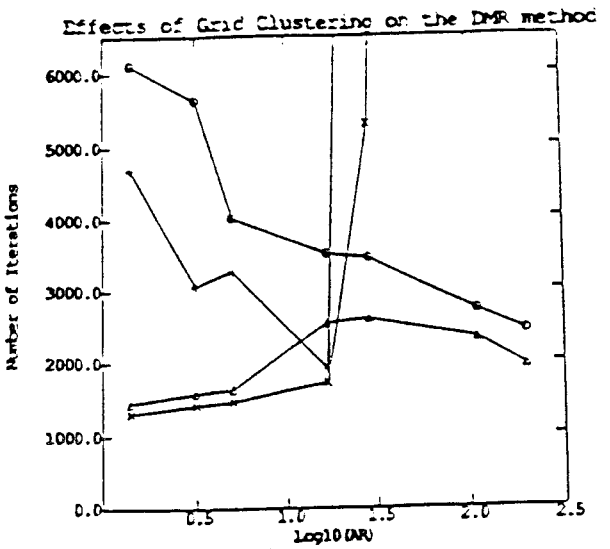


Fig. 4 Effects of the artificial compressibility coefficient on the convergence (Euler implicit; -o-o-, DMR; -Δ-Δ-)



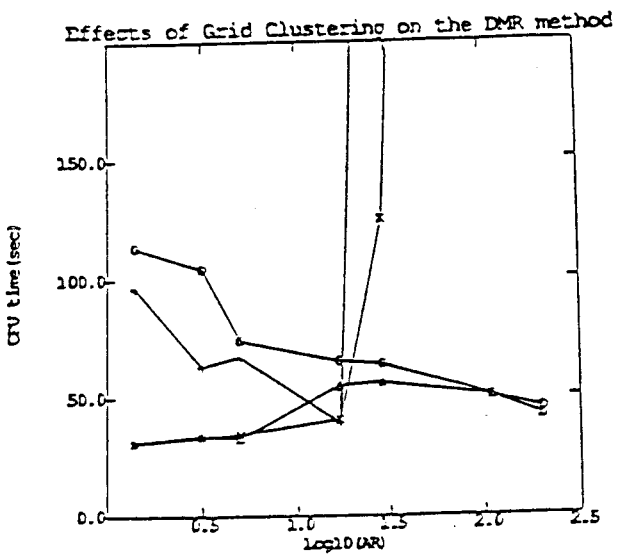
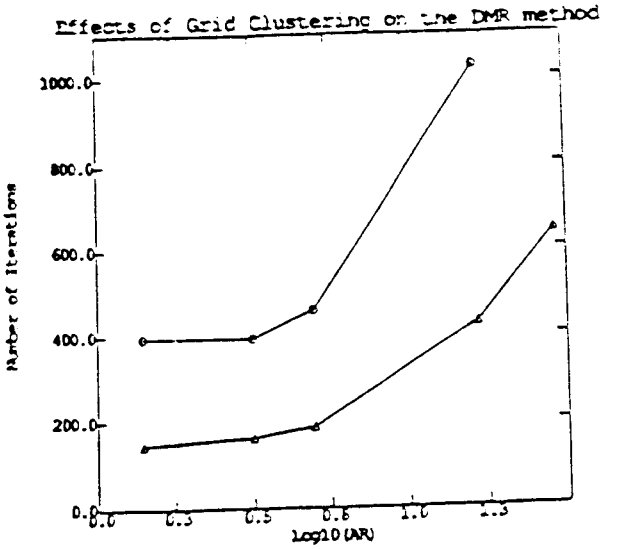


Fig. 5 Effects of grid clustering on the convergence (RK; -o-o-, DMR; -Δ-Δ-, IRS; -+-+-, DMR+IRS; -x-x-)



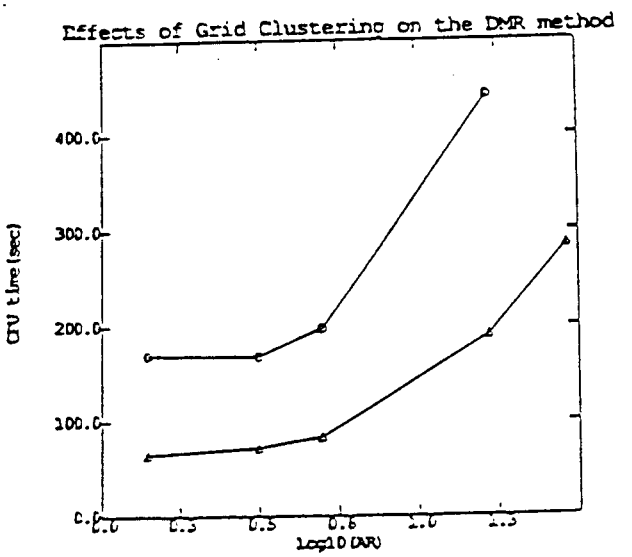
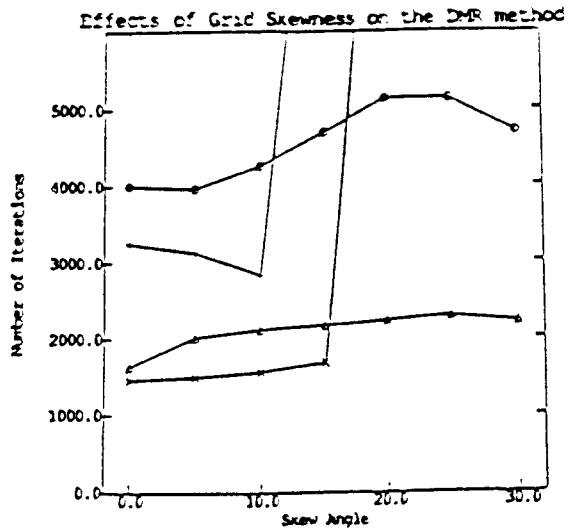


Fig. 6 Effects of grid clustering on the convergence (Euler implicit; -o-o-, DMR; -Δ-Δ-)



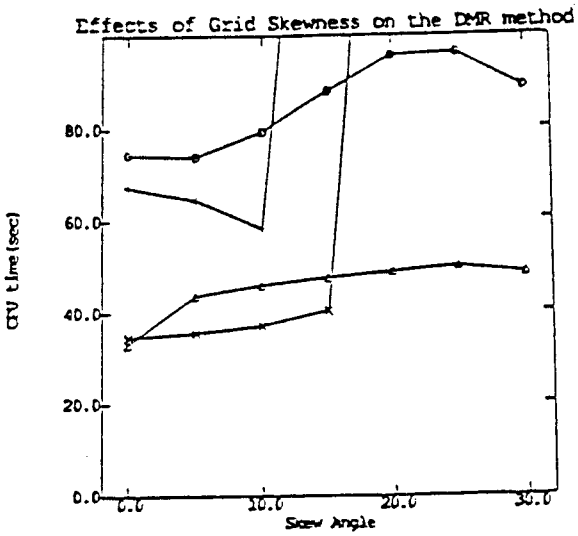
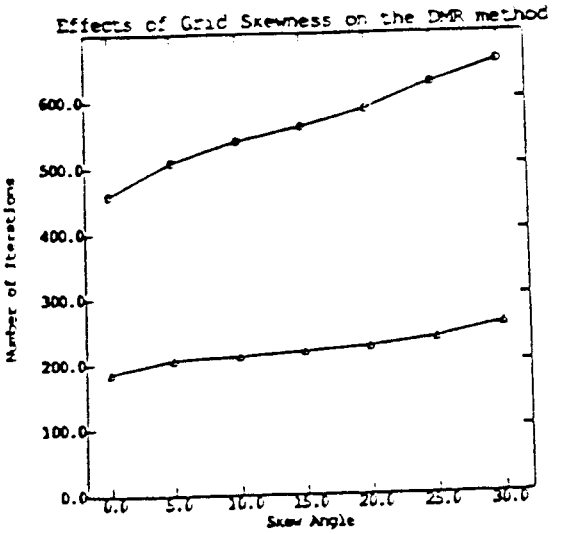


Fig. 7 Effects of grid skewness on the convergence (RK; -o-o-, DMR; -Δ-Δ-, IRS; -+-+-, DMR+IRS; -x-x-)



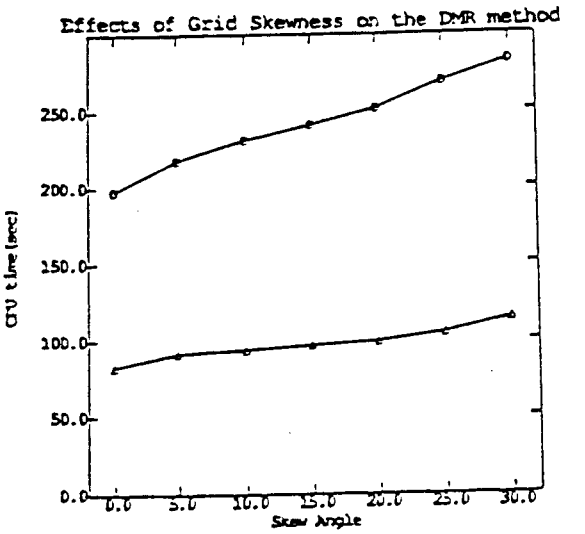
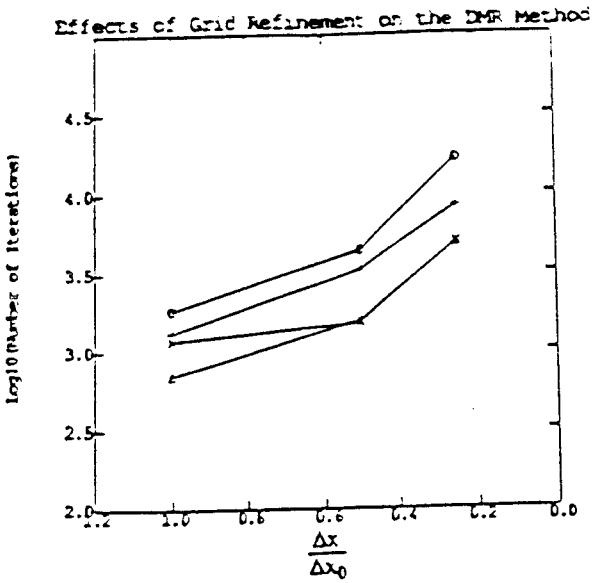


Fig. 8 Effects of grid skewness on the convergence (Euler implicit; -o-o-, DMR; -Δ-Δ-)



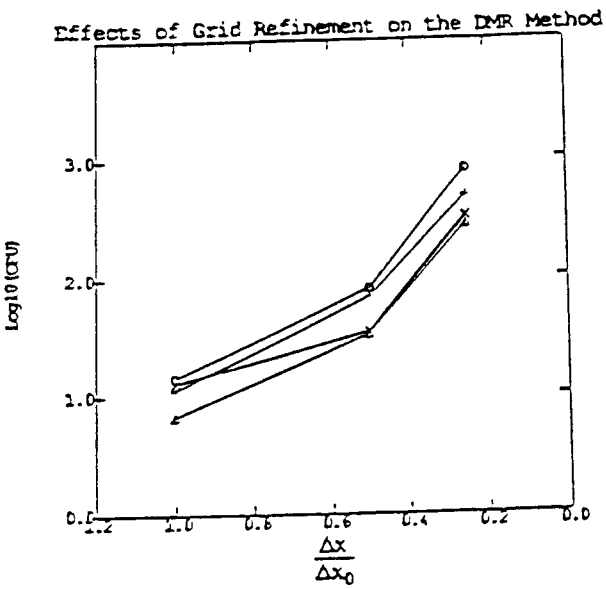
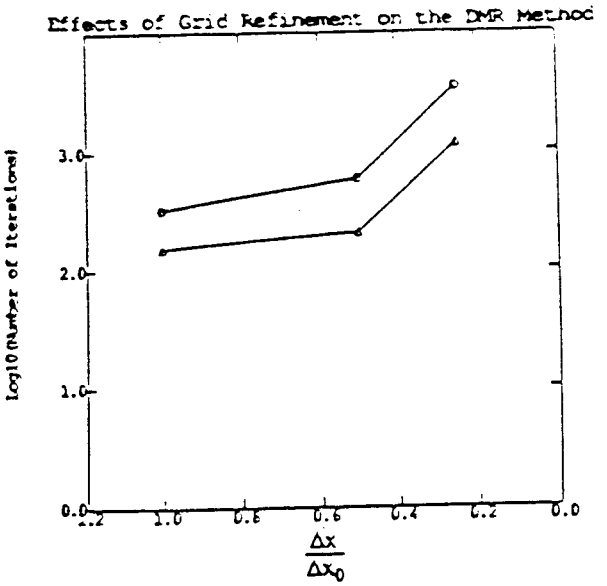


Fig. 9 Effects of grid refinement on the convergence (RK; -o-o-, DMR; -Δ-Δ-, IRS; -+-+-, DMR+IRS; -x-x-)



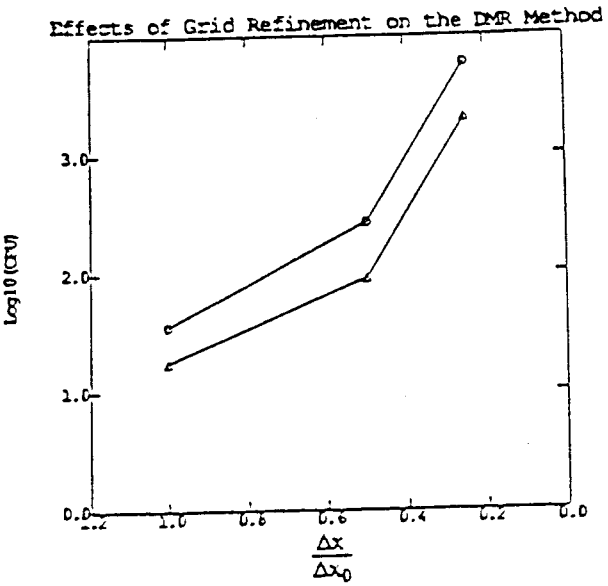
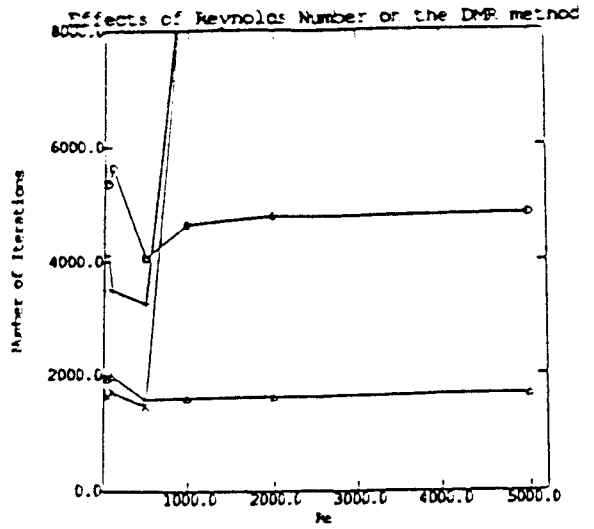


Fig. 10 Effects of grid refinement on the convergence (Euler implicit; -o-o-, DMR; -Δ-Δ-)



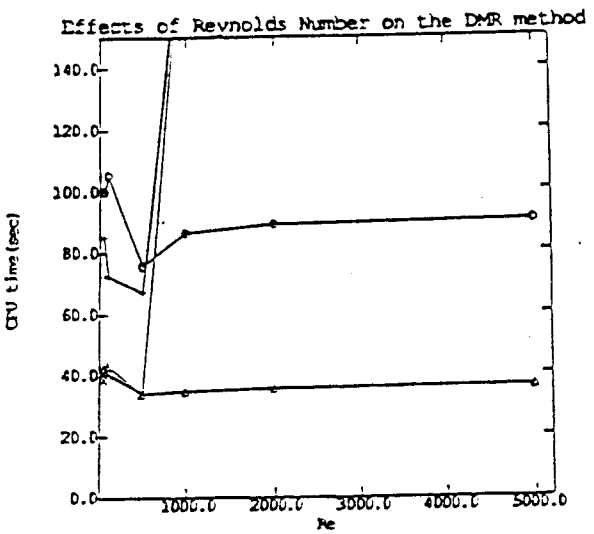
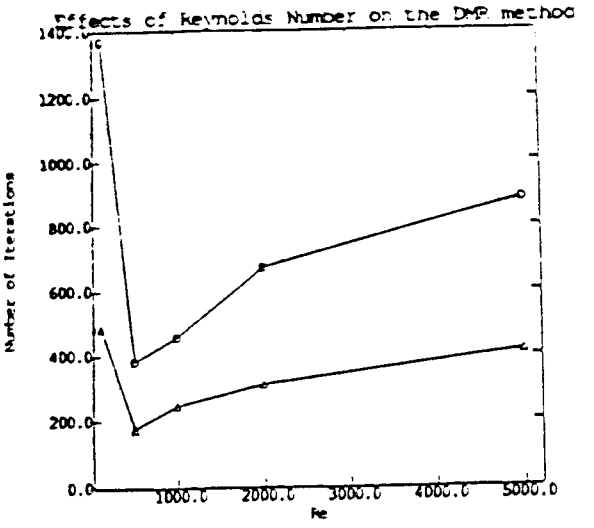


Fig. 11 Effects of Reynolds number on the convergence
(RK; -o-o-, DMR; -Δ-Δ-, IRS; -+-+-, DMR+IRS; -×-×-)



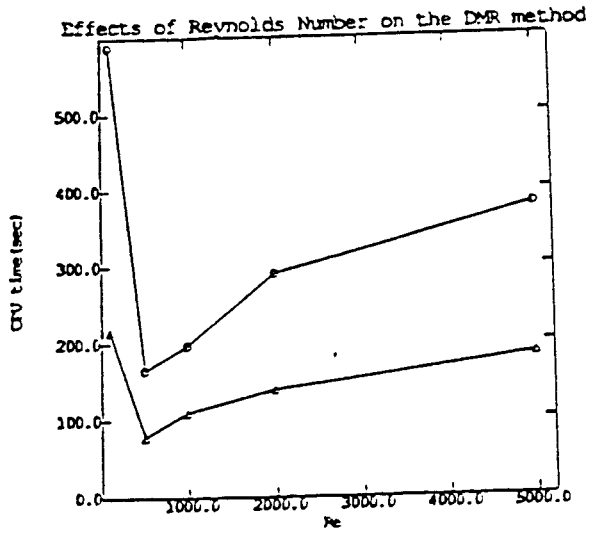
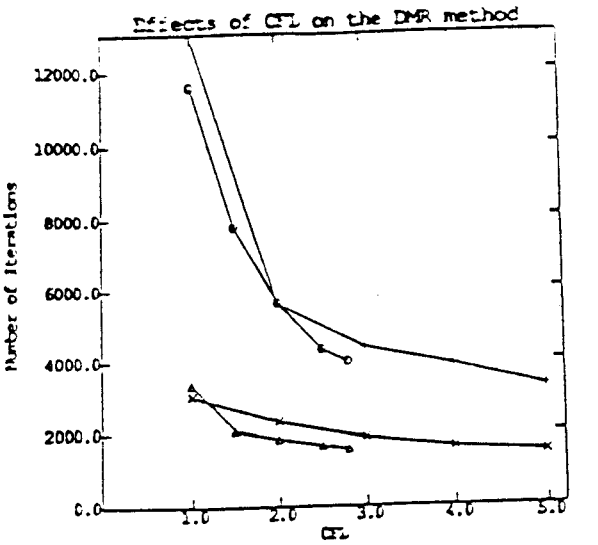


Fig. 12 Effects of Reynolds number on the convergence
(Euler implicit; -o-o-, DMR; -Δ-Δ-)



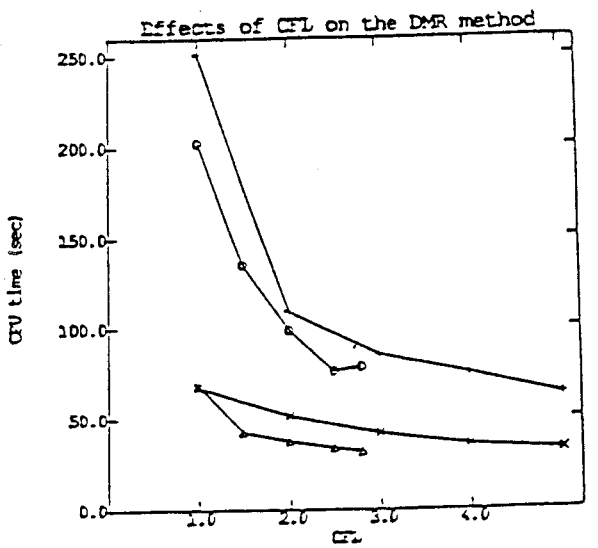
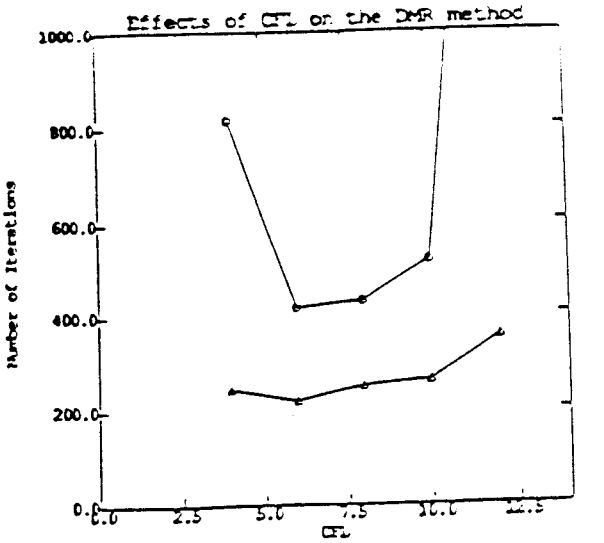


Fig. 13 Effects of CFL number on the convergence (RK; -o-o-, DMR; -Δ-Δ-, IRS; -+-+-, DMR+IRS; -x-x-)



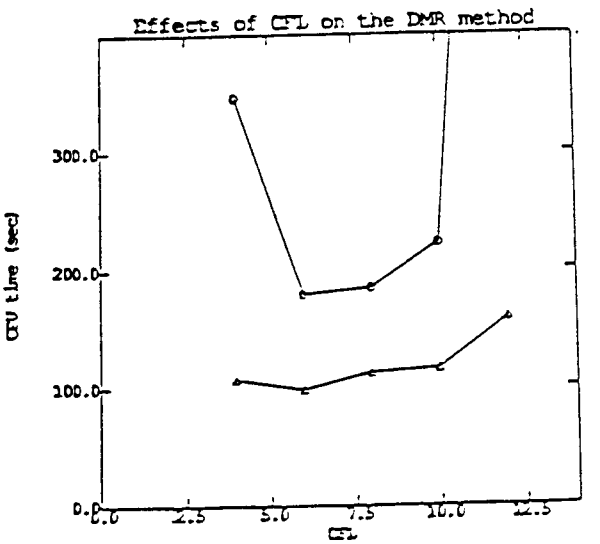


Fig. 14 Effects of CFL number on the convergence (Euler implicit; -o-o-, DMR; -Δ-Δ-)