Incompressible Navier-Stokes Solver with Extrapolation Method Suitable for Massively Parallel Computing*

(Application of Extrapolation Method to N-S Equations)

Kenjiro SHIMANO**, Yoshiyuki HAMAJIMA***
and Chuichi ARAKAWA***

In this paper, it is mathematically clarified by discussing the behavior of a nonlinear mapping that the extrapolation method described in the previous first report can be applied to numerical procedures for solving nonlinear equations. Guidelines for coping with nonlinear effects in extrapolating solutions of nonlinear equations are also presented. The coupled method is adopted as a numerical algorithm for incompressible flows because it is simple and its procedures can be efficiently parallelized. Parallel codes consisting of the coupled method and the extrapolation method are executed on a Fujitsu AP 1000 to solve 2-D steady flows. It is shown that the parallel efficiency of the coupled method is very high and that the extrapolation method keeps net speed-up high even when granularity becomes fine. It is concluded that these two methods constitute a promising N-S solver suitable for massively parallel computing of very fine granularity.

Key Words: Computational Fluid Dynamics, Numerical Analysis, Finite Volume Method, Incompressible Flow, Massively Parallel Computer, Extrapolation Method, Coupled Method

1. Introduction

In this study, the authors propose a combination of the coupled method and the extrapolation method as a numerical algorithm suitable for massively parallel computing. In the first report, three existing versions of the extrapolation method⁽¹⁾ and a newly developed one, LWE, were reviewed. The properties of the extrapolation method were clarified by applying it to numerical analysis of Poisson's equation, and its suitability for massively parallel computing was shown.

In this second report, it is mathematically proved that the extrapolation method described in the previous paper can be applied to numerical procedures

solving nonlinear equations. Guidelines for coping with nonlinear effects in extrapolating solutions of nonlinear equations are also presented.

The coupled method is adopted as a basic numerical algorithm for solving incompressible Navier-Stokes equations because it is simple and its procedures can be efficiently parallelized. The coupled method is reviewed in Section 3.

Parallel codes consisting of the coupled method and the extrapolation method are implemented on Fujitsu AP 1000, and it is shown that these two methods constitute a promising N-S solver suitable for massively parallel computing of very fine granularity.

2. Applicability to Nonlinear Equations

The extrapolation method reviewed in the first paper is based on the linear theory and its application to nonlinear equations is not straightforward. Smith et al.⁽²⁾ observed that the error approached zero quadratically by extrapolating a solution repeatedly for nonlinear equations. In their discussion, nonlinear terms were treated as small errors involved in linear

* Received 6th March, 1997. Japanese original: Trans. Jpn. Soc. Mech. Eng., Vol. 62, No. 603, B(1996), p. 3899-3904 (Received 21st December, 1995)

** Science University of Tokyo, Department of Mechanical Engineering, 1-3 Kagurazaka, Shinjuku-ku, Tokyo 162, Japan

*** University of Tokyo, Department of Mechanical Engineering, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113, Japan

mapping. Such an assumption implies weak non-linearity of a target equation and is not sufficient, since a flow at a high Reynolds number is of great interest in the CFD field.

In this section, the behavior of nonlinear mapping in the neighborhood of a fixed point is studied and the applicability of the extrapolation method to nonlinear equations is discussed. Guidelines for applying the extrapolation method to nonlinear equations are given. Discussion in this section is partly based on Wiggins⁽³⁾.

2. 1 Behavior of nonlinear mapping

An iterative procedure for solving a system of nonlinear equations is equivalent to nonlinear mapping in a vector space. Starting from an initial vector, an intermediate vector approaches the required fixed point.

Let us consider the sequential nonlinear mapping in the vector space $\mathbf{R}^N$ with the fixed point $\mathbf{o}$;

$$\mathbf{e} \mapsto \mathbf{J}\mathbf{e} + \mathbf{F}(\mathbf{e}). \quad (1)$$

The matrix $\mathbf{J}$ represents linear terms. $\mathbf{F}(\mathbf{e})$ is a vector representing nonlinear terms. It consists of only quadratic terms and satisfies $\mathbf{F}(\mathbf{o}) = \mathbf{o}$. For simplicity, it is assumed that the matrix $\mathbf{J}$ is in the Jordan standard form and that all the eigenvalues have a magnitude of less than unity. Note that $\mathbf{J}$ is equivalent to the iteration matrix $\mathbf{M}$.

Let $\mathbf{e}_k$ be the vector obtained after the k -th mapping starting from the initial vector $\mathbf{e}_0$. The vector $\mathbf{e}_k$ satisfies the relation

$$\mathbf{e}_{k+1} = \mathbf{J}\mathbf{e}_k + \mathbf{F}(\mathbf{e}_k). \quad (2)$$

The nonlinear terms in the $\mathbf{F}(\mathbf{e})$ are $O(|\mathbf{e}|^2)$. Obviously, the behavior of the nonlinear mapping becomes approximately the same as that of the linear mapping $\mathbf{e} \mapsto \mathbf{J}\mathbf{e}$ as an intermediate vector approaches the fixed point $\mathbf{o}$. However, it is necessary to extrapolate solutions at an early step of iterations when a residual is in order of $10^{-1} \sim 10^{-2}$. It is impossible to neglect terms of $O(|\mathbf{e}|^2)$. Therefore, in this discussion, quadratic terms are taken into consideration and terms of $O(|\mathbf{e}|^3)$ are neglected.

The vector $\mathbf{e}_k$ is transformed to $\mathbf{y}_k$ by

$$\begin{cases} \mathbf{e}_k \mapsto \mathbf{y}_k \\ \mathbf{e}_k = \mathbf{y}_k + \mathbf{h}(\mathbf{y}_k), \end{cases} \quad (3)$$

where $\mathbf{h}(\mathbf{y})$ is a vector of which elements are given as quadratic forms of $\mathbf{y}$ elements. Clearly $\mathbf{h}(\mathbf{o}) = \mathbf{o}$. The mapping for $\mathbf{y}_k$ is obtained after a little modification,

$$\mathbf{y}_{k+1} = \mathbf{J}\mathbf{y}_k + \mathbf{J}\mathbf{h}(\mathbf{y}_k) - \mathbf{h}(\mathbf{J}\mathbf{y}_k) + \mathbf{F}(\mathbf{y}_k) + O(3). \quad (4)$$

If we choose $\mathbf{h}(\mathbf{y})$ such that

$$\mathbf{J}\mathbf{h}(\mathbf{y}) - \mathbf{h}(\mathbf{J}\mathbf{y}) + \mathbf{F}(\mathbf{y}) = \mathbf{o}, \quad (5)$$

a simple expression of the vector $\mathbf{y}_k$ without quadratic terms is obtained;

$$\mathbf{y}_{k+1} = \mathbf{J}\mathbf{y}_k + O(3). \quad (6)$$

The quadratic form $\mathbf{h}(\mathbf{y})$ satisfying Eq. (5) exists

only if for all i, j, n

$$\lambda_i \lambda_j - \lambda_n \neq 0. \quad (7)$$

Let us suppose that condition (7) is satisfied. The mapping (6) is approximately linear because terms of $O(3)$ are negligible in the neighborhood of the fixed point $\mathbf{o}$. Then the n -th component of the vector $\mathbf{y}_k$ can be expressed by the n -th eigenvalue of $\mathbf{J}$

$$(\mathbf{y}_k)_n = a_n \lambda_n^k \quad (8)$$

where a_n represents the coefficient dependent of the initial vector. Substituting the R. H. S of (8) to (3) gives

$$(\mathbf{e}_k)_n = a_n \lambda_n^k + \sum_{i,j} A_{ijn} (\lambda_i \lambda_j)^k \quad (9)$$

where A_{ijn} is the coefficient defined by a_i, a_j and the quadratic form $\mathbf{h}(\mathbf{y})$. The first term in the R. H. S of (9) also appears in the corresponding linear mapping $\mathbf{e} \mapsto \mathbf{J}\mathbf{e}$. The second term including the product $\lambda_i \lambda_j$ arises from nonlinearity. It is comprehensible that the product $\lambda_i \lambda_j$ behaves like an eigenvalue. The product $\lambda_i \lambda_j$ is called *substantial eigenvalue* in this paper. According to Eq. (9), one component of the error vector $\mathbf{e}_k$ is expressed in powers of eigenvalues and substantial eigenvalues. Terms including small eigenvalues or small substantial eigenvalues can be neglected for sufficiently large integer k , and only p dominant terms are taken into account. Let us consider a *pseudo-characteristic* polynomial equation such that eigenvalues and substantial eigenvalues satisfy,

$$\sum_{j=0}^p c'_j \lambda^j = 0 \quad (10)$$

where λ stands for an eigenvalue or a substantial eigenvalue. If its coefficients $c'_0, \dots, c'_p$ are used as weights, all the eigenvalues and the substantial eigenvalues considered here should vanish. Therefore it can be concluded that the extrapolation method based on the linear theory is applicable to iterative procedures for solving nonlinear equations when condition (7) is satisfied.

If condition (7) is not satisfied for n , $(\mathbf{y}_k)_n$ is written in the form

$$(\mathbf{y}_k)_n = a_n \lambda_n^k + b_n k \lambda_n^k. \quad (11)$$

For simplicity, we assume another component in the form of (8). The term including $k \lambda_n^k$ appears in the R. H. S of Eq. (11). Such terms do not vanish by the weighted average using the coefficients of the pseudo-characteristic equation considered above, but it is possible to eliminate the terms by adopting a set of coefficients $\{c'_0, c'_1, \dots, c'_p\}$ such that

$$\sum_{j=0}^p c'_j (k+j) \lambda_n^{k+j} = 0. \quad (12)$$

The coefficients c'_j are determined by (12) together with the following equation which is similar to the pseudo-characteristic equation;

$$\sum_{j=0}^p c_j' \lambda^j = 0 \quad (13)$$

where λ stands for an eigenvalue or a substantial eigenvalue.

Then the extrapolation method is applicable even when $\lambda_i \lambda_j - \lambda_n = 0$. Extension to cases in which two or more components do not satisfy condition (7) is straightforward. Note that the numerical procedures for extrapolating solutions of nonlinear equations are all the same as linear equations.

2.2 Guidelines for coping with nonlinear effect

Nonlinearity can be interpreted as an increase of the apparent degree of freedom by the appearance of substantial eigenvalues and/or $k\lambda^k$. Especially when nonlinearity is strong, A_{ijn} in Eq. (9) becomes large, and many terms including $\lambda_i \lambda_j$ become dominant. One has to choose a large value for p to cope with the nonlinear effect. However, a large p leads to a high cost of memory storage and a very long time for extrapolation. To avoid this, the authors take another strategy in which intermediate solutions are stored every q -th step for some $q > 1$. This means that the iteration matrix M is replaced by M^q , and consequently the eigenvalues $\{\lambda_1, \lambda_2, \dots, \lambda_N\}$ are replaced by $\{\lambda_1^q, \lambda_2^q, \dots, \lambda_N^q\}$. By this modification, separation of dominant eigenvalues from the others can be achieved, which results in a good extrapolation.

Strategies proposed in this study for coping with the nonlinear effect are summarized as

- to make the number of considered eigenvalues large
- to store intermediate solutions every q -th step.

3. Coupled Method

The coupled method based on Vanka's SCGS (Symmetric Coupled Gauss-Seidel) method⁽⁴⁾ is adopted in solving the incompressible Navier-Stokes equations.

A set of algebraic equations is obtained by the finite volume method on a staggered grid.

$$\begin{aligned} AP_{i,j}^u u_{i,j} &= AE_{i,j}^u u_{i+1,j} + AW_{i,j}^u u_{i-1,j} + AN_{i,j}^u u_{i,j+1} \\ &\quad + AS_{i,j}^u u_{i,j-1} + \Delta y (P_{i-1,j} - P_{i,j}) + SU_{i,j} \\ AP_{i,j}^v v_{i,j} &= AE_{i,j}^v v_{i+1,j} + AW_{i,j}^v v_{i-1,j} + AN_{i,j}^v v_{i,j+1} \end{aligned} \quad (14)$$

Fig. 1 Staggered grid

$$+ AS_{i,j}^v v_{i,j-1} + \Delta x (P_{i,j-1} - P_{i,j}) + SV_{i,j} \quad (15)$$

$$\Delta y (u_{i,j} - u_{i+1,j}) + \Delta x (v_{i,j} - v_{i,j+1}) = 0 \quad (16)$$

where, AP , AE , etc. are coefficients evaluated by the finite volume method. In the coupled method, the four velocity components and one pressure component corresponding to one staggered cell (control volume) shown in Fig. 1 are simultaneously updated. This can be achieved by inverting a 5×5 matrix for each cell, namely.

$$\begin{bmatrix} AP_{i,j}^u & -AE_{i,j}^u & 0 & 0 & \Delta y \\ -AW_{i+1,j}^u & AP_{i+1,j}^u & 0 & 0 & -\Delta y \\ 0 & 0 & AP_{i,j}^v & -AN_{i,j}^v & \Delta x \\ 0 & 0 & -AS_{i,j+1}^v & AP_{i,j+1}^v & -\Delta x \\ \Delta y & -\Delta y & \Delta x & -\Delta x & 0 \end{bmatrix} \times \begin{bmatrix} u_{i,j} \\ u_{i+1,j} \\ v_{i,j} \\ v_{i,j+1} \\ P_{i,j} \end{bmatrix} = \begin{bmatrix} f_{i,j}^u \\ f_{i+1,j}^u \\ f_{i,j}^v \\ f_{i,j+1}^v \\ 0 \end{bmatrix} \quad (17)$$

Contributions of variables located outside of the cell are included in elements of the R. H. S vector such as $f_{i,j}^u$. The procedures for the coupled method are

- (1) to evaluate coefficients of algebraic equations
- (2) inner iteration: relaxation over all the cells using Eq. (17)
- (3) outer iteration: to repeat (1) and (2) until the solution converges.

Staggered cells are grouped by red-black as shown in Fig. 2, and relaxation on one group is carried out after another in the inner iteration stage. If a regular order is adopted, relaxation may be less effective because some velocity components are updated immediately after the last update. By adopting the red-black ordering, for instance, about a 10% reduction of iterations is observed in calculating the lid-driven square cavity flow at $Re=1000$ with 128×128 control volumes on a single processor.

Parallelization of the coupled method is achieved by domain decomposition as shown in Fig. 2 where

Fig. 2 Red-Black ordering and domain decomposition

connecting boundaries are represented as wide lines. Red/black variables along the connecting boundaries are sent to relevant neighboring processors after each red/black group relaxation. Although velocity components located on connecting boundaries belong to two processors, they are not updated at the same time due to the red-black ordering, and the iterative procedures are efficiently parallelized.

The coupled method is suitable for massively parallel computing for the following reasons;

1. Velocities and pressures are simultaneously sent/received, which reduces the overhead time required for calling a message passing library.
2. Operations to invert the 5×5 matrix which occupies a large part of CPU time can be completely parallelized.
3. The algorithm is quite simple without an additional equation such as Poisson's equation for pressure.

4. Application to Incompressible N-S Equations

The two flow-fields illustrated schematically in Fig. 3 are simulated on an AP 1000. Both of the flows are laminar. The equations are discretized using the finite volume method, and the hybrid scheme⁽⁵⁾ is used to evaluate the convection-diffusion terms. Calculation is terminated when the non-dimensional residual for each equation is below 10^{-4} .

As for the square cavity, the upper wall moves from the left to the right. The flows are at $Re=100$ and 1000 based on the wall velocity and the length of the square. A uniform staggered mesh with 256×256 control volumes is used.

The sudden expansion flow is at $Re=50$ based on the inlet velocity and the step height. The length of the computational domain along the x -axis is 20 times the inlet width. The number of control volumes is 256

$\times 128$. The grid step Δx is given in the manner of geometric progression and cells are clustered around the inlet.

The code adopting the coupled method is parallelized and implemented on a Fujitsu AP 1000. The number of processors is $256 (=16 \times 16)$ and $512 (=32 \times 16)$. A smaller number of processors than 256 is not adopted here, because this study focuses on quite fine granularity.

Table 1 presents the efficiency ϵ and speed-up S achieved on the AP 1000 without extrapolation. Somehow, high efficiencies are observed in the table. For instance, the efficiency of the cavity flow at $Re=1000$ on 256 PE is 92.7% while that by Sakamoto and Kawamura⁽⁶⁾, who solved the same flow on the same parallel computer, is about 70%. Their computational conditions such as the number of grid points and number of processors were identical to ours except that they adopted the MAC method and its revised versions. Our result is sufficient proof that the coupled method is suitable for massively parallel computing.

Then the extrapolation method is applied. The number of considered eigenvalues, p , and the interval at which histories are stored, q , are determined by trial and error. The final choices are $(p, q) = (19, 16)$ for the cavity flow at $Re=100$, $(29, 16)$ for the cavity flow at $Re=1000$ and $(29, 10)$ for the sudden expansion flow. The wavenumber parameter n_{ws} for LWE is taken as $n_{ws}=7$ for the cavity flows and $n_{ws}=6$ for the sudden expansion flow. These values for n_{ws} are also determined by trial and error.

Residual reduction by ROLE is illustrated in Figs. 4-7 for the cavity flows. The numbers in those figures represent the number of iterations at which extrapolation is executed. Rapid reduction of residuals after extrapolation can be found in all cases and demonstrates the validity of our discussion in Section 2. As observed in application to the Poisson's equation in the first report, later extrapolation tends to give a larger accelerative effect, since separation of dominant eigenvalues from small ones is achieved in later extrapolation.

Comparing the results in terms of the Reynolds number, it is observed that the accelerative effect at $Re=1000$ is poorer than at $Re=100$. This is attributed to a larger A_{ijn} in Eq. (9) resulting from stronger nonlinearity. To cope with the stronger nonlinearity

Fig. 3 Simulated flows

Table 1 Efficiency and speed-up (No extrapolation)

PE	Cavity $Re = 100$		Cavity $Re = 1000$		Sudden Expansion	
	$\epsilon[\%]$	S	$\epsilon[\%]$	S	$\epsilon[\%]$	S
256	91.4	234	92.7	237	82.4	211
512	80.4	412	84.2	431	71.4	366

Fig. 4 History of residuals (ROLE)
Cavity $Re=100$; 16×16 PEs

Fig. 5 History of residuals (ROLE)
Cavity $Re=100$; 32×16 PEs

Fig. 6 History of residuals (ROLE)
Cavity $Re=1000$; 16×16 PEs

of $Re=1000$, a larger value for $p(=29)$ than $p=19$ for $Re=100$ was chosen, which is, however, insufficient.

ROGE, MPE and LWE have the features of ROLE mentioned above in common.

Tables 2-4 show the speed-up of iterations, S_{iter} , by which one can see how fast the solution converges using the extrapolation method. The S_{iter} is defined by

Fig. 7 History of residuals (ROLE)
Cavity $Re=1000$; 32×16 PEs

Table 2 Speed-up of iterations
Cavity $Re=100$

PE	n_{ex}	ROLE	MPE	ROGE	LWE
16×16 $iter = 5066$	2144	1.697	1.963	1.501	1.865
	2344	1.902	1.809	1.591	1.839
	2544	1.934	1.991	1.810	1.931
32×16 $iter = 5248$	2144	1.673	1.460	1.541	1.832
	2344	2.012	1.735	1.635	1.790
	2544	1.945	1.679	1.796	1.994

Table 3 Speed-up of iterations
Cavity $Re=1000$

PE	n_{ex}	ROLE	MPE	ROGE	LWE
16×16 $iter = 7048$	2990	1.231	1.356	1.286	1.433
	3344	1.237	1.239	1.249	1.588
	3744	1.573	1.399	1.538	1.701
32×16 $iter = 7079$	2990	1.228	1.321	1.207	1.357
	3344	1.203	1.192	1.275	1.581
	3744	1.632	1.489	1.360	1.674

Table 4 Speed-up of iterations
Sudden Expansion

PE	n_{ex}	ROLE	MPE	ROGE	LWE
16×16 $iter = 7066$	2601	1.792	1.495	1.420	2.150
	2990	1.794	1.967	1.539	1.999
	3390	1.734	2.083	1.863	1.990
32×16 $iter = 7107$	2601	1.874	1.885	1.573	2.162
	2990	1.878	2.109	1.741	2.001
	3390	1.920	2.060	1.559	1.987

$$S_{iter} = \frac{iter}{iter_{ex}} \quad (18)$$

where $iter$ and $iter_{ex}$, respectively, represent the number of iterations without extrapolation and that with extrapolation. The time for extrapolation is not considered in S_{iter} . n_{ex} in the tables represents the number of iterations at which extrapolation is executed. LWE gives the highest S_{iter} for the cavity flow at $Re=1000$. As for the cavity flow at $Re=100$ and the sudden expansion flow, MPE and LWE show good

Fig. 8 Total CPU time ratio (16×16 PEs) Fig. 9 Total CPU time ratio (32×16 PEs)

acceleration. Also, ROLE is comparable to MPE and LWE for the cavity flow at $Re=100$. Judging from these results, techniques such as LWE and MPE considering some global information are very effective for reducing the number of iterations. Furthermore, since LWE gives high values of S_{iter} for all the flows, it is ascertained that extracting and separately extrapolating low frequency terms is a good strategy for improving the accelerative property of the extrapolation method. It is noteworthy that LWE works well even for a high Reynolds number flow. However, the superiority of LWE to the others is not very significant. Further study on how to extract low frequency errors is required for improvement of LWE. Also, application of LWE to problems involving complex geometry is another important subject to be tackled. The fact that ROLE is better than ROGE for almost all the flows is incomprehensible. It seems to be an accidental result.

Figs. 8 and 9 show total CPU time ratios, which are interpreted as reciprocals of net speed-up by extrapolation. Bars are drawn in these figures only for the cases of maximum net speed-up. The four versions of the extrapolation method do not vary greatly in the reduction ratio of CPU time; $0.6 \sim 0.7$ for the cavity at $Re=1000$ and $0.5 \sim 0.6$ for the cavity at $Re=100$ and the sudden expansion. It is noteworthy that the reduction ratio of CPU time hardly depends on the number of processors, as observed in application to Poisson's equation (see the first report).

Table 5 Speed-up of iterations (QUICK)

Cavity $Re = 100$ $p = 16, q = 6$			Cavity $Re = 1000$ $p = 22, q = 6$		
PE	n_{ex}	S_{iter}	PE	n_{ex}	S_{iter}
16×16 $iter = 1625$	487	1.238	16×16 $iter = 1613$	559	1.502
	587	1.494		659	1.835
	687	1.772		759	1.929
8×8 $iter = 1470$	487	1.105	8×8 $iter = 1479$	559	1.419
	587	1.778		659	1.312
	687	1.699		759	1.679

The net speed-up is kept high even for fine granularity because the time for extrapolation is only a small part of the whole computational time: even when fine granularity makes the parallel efficiency of extrapolation low, it affects the net speed-up only slightly. The extrapolation method is an appropriate choice for massively parallel computation of very fine granularity.

As a general rule, when a different discretizing scheme is adopted, an acceleration technique requires modification of the program. However, this is not true for the extrapolation method. The results shown in Table 5 are obtained by applying ROLE to the coupled method with the QUICK scheme instead of the hybrid scheme. Simulated flows are the cavity flows at $Re = 100$ and 1000 . The number of control volumes is 128×128 . The extrapolation parameters, p and q , are shown in the table. The subroutine of ROLE is exactly the same as that used in the hybrid scheme. This demonstrates that the extrapolation method is easy to handle. The acceleration effect of the extrapolation method in the QUICK code is roughly comparable with that in the hybrid scheme: most S_{iter} 's are in the range $1.5 \sim 2.0$.

Problems remain with the extrapolation method, and some improvement is required. For example, the number of histories referred to must be reduced in order to avoid high costs for memory storage while further accelerative performance is required, especially for high Reynolds number flows. The performance of the extrapolation method in practical engineering problems such as turbulent flows with complex geometry is still unknown at present. However, it has been confirmed in this study that the coupled method and the extrapolation method are suitable for massively parallel computing of fine granularity. Their combination is a promising numerical technique for realizing the full performance of massively parallel computers.

5. Concluding Remarks

The applicability of the extrapolation method to a nonlinear system of equations was discussed by

considering the behavior of nonlinear mapping. The extrapolation method was introduced to the coupled method for solving incompressible Navier-Stokes equations on an AP 1000. The reduction ratios of CPU time were 0.6~0.7 for the cavity flow at $Re=1\,000$ and 0.5~0.6 for both the cavity flow at $Re=100$ and the sudden expansion flow. The conclusions are summarized as follows.

- In applying the extrapolation method to nonlinear equations, the number of considered eigenvalues must be large and histories must be stored every q -th step ($q>1$).

- The coupled method which gives a higher parallel efficiency than the MAC method and its revised versions is fit for massively parallel computing.

- The extrapolation method keeps the accelerative property high for fine granularity in contrast to the multigrid method.

- Even though the extrapolation method needs some improvement, the combination of the extrapolation method and the coupled method is a promising numerical technique for realizing the performance of massively parallel computers.

Acknowledgement

The authors would like to thank Fujitsu Research Co. for permission to use the AP 1000.

References

- (1) Leland, R. W. and Rellett, J. S., A Comparison of Parallel Extrapolation Methods, Proc. of the sixth GAMM-Seminar, (1991), p. 177, Viewveg.
- (2) Smith, D. A., Ford, W. F. and Sidi, A., Extrapolation Methods for Vector Sequences, SIAM Rev. Vol. 29, No. 2 (1987), p. 199.
- (3) Wiggins, S., Introduction to Applied Nonlinear Dynamics System and Chaos, (1990), Springer-Verlag, New York.
- (4) Vanka, S. P., Block-Implicit Multigrid Solution of Navier-Stokes Equations in Primitive Variables, J. Comp. Phys. Vol. 65 (1986), p. 138.
- (5) Patankar, S. V., Numerical Heat Transfer and Fluid Flow, (1980), McGraw-Hill.
- (6) Sakamoto, M. and Kawamura, H., Parallel Computation of a Two-Dimensional Flow Field Using the MAC Methods, Trans. Jpn. Soc. Mech. Eng., (in Japanese), Vol. 60, No. 569 (1994), p. 99.