An International Journal
computers & mathematics
with applications

PERGAMON Computers and Mathematics with Applications 37 (1999) 49–60

Parallel Multigrid Finite Volume Computation of Three-Dimensional Thermal Convection

P. WANG* AND R. D. FERRARO
Jet Propulsion Laboratory, California Institute of Technology
MS 168-522, 4800 Oak Grove Drive
Pasadena, CA 91109-8099, U.S.A.

(Received May 1998; accepted June 1998)

Abstract—A parallel implementation of the finite volume method for three-dimensional, time-dependent, thermal convective flows is presented. The algebraic equations resulting from the finite volume discretization, including a pressure equation which consumes most of the computation time, are solved by a parallel multigrid method. A flexible parallel code has been implemented on the Intel Paragon, the Cray T3D, and the IBM SP2 by using domain decomposition techniques and the MPI communication software. The code can use 1D, 2D, or 3D partitions as required by different geometries, and is easily ported to other parallel systems. Numerical solutions for air (Prandtl number Pr= 0.733) with various Rayleigh numbers up to 10^7 are discussed. Published by Elsevier Science Ltd.

Keywords—Thermal convection, Parallel computation, Finite volume, Multigrid.

NOMENCLATURE

a, b, d	coefficients of the finite-volume equations	h	height of cavity
$\bar{A}$	area of a control-volume face	l	length of cavity
A	coefficient matrix of the finite-volume equation	$\mathrm{N_u}$	Nusselt number
d	width of cavity	p	nondimensional pressure
g	acceleration due to gravity	p^*	best estimate of pressure
		p'	pressure correction

*Author to whom all correspondence should be addressed.

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, and was sponsored by the National Research Council and the National Aeronautics and Space Administration while one of the authors (P.Wang) held a NRC-JPL Research Associateship.

Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government, the National Research Council, or the Jet Propulsion Laboratory, California Institute of Technology.

This research was performed in part using the CSCC parallel computer system operated by Caltech on behalf of the Concurrent Supercomputing Consortium. Access to this facility was provided by NASA. The Cray Supercomputer used in this investigation was provided by funding from the NASA Offices of Mission to Planet Earth, Aeronautics, and Space Science. Access to the Intel Paragon at the Jet Propulsion Laboratory, California Institute of Technology, was provided by the NASA High Performance Computing and Communications Program.

0898-1221/99/$ - see front matter. Published by Elsevier Science Ltd.
PII: S0898-1221(99)00113-3

R	Rayleigh number
T	nondimensional temperature
u, v, w	nondimensional velocity components
u^*, v^*, w^*	velocity components based on p^*
u', v', w'	velocity corrections
x, y, z	nondimensional coordinates

GREEK SYMBOLS

α	under-relaxation factor
β	coefficient of thermal expansion
κ	thermal diffusivity
ν	kinematic viscosity
σ	Prandtl number

1. INTRODUCTION

Natural convection driven by imposed horizontal density gradients finds many applications in engineering: reactor cooling systems, crystal growth procedures, and solar-energy collectors. The most numerically studied form of this problem is the case of a rectangular cavity with differentially heated sidewalls. The two-dimensional version of this problem has received considerable attention [1–6], but for the three-dimensional case, very few results have been obtained, mainly due to the limits in computing power. The significant computational resources of modern, massively parallel supercomputers promise to make such studies feasible. In order to determine such flow structures and heat transfer, numerical simulations using the Navier-Stokes equations and the energy equation on parallel systems are discussed.

In the early simulation methods, vorticity and stream function were usually the calculated variables, but the use of primitive variables has attracted many computational researchers for three-dimensional simulations. A number of numerical algorithms have been developed for incompressible fluid flows, such as MAC [7], the Projection Method [8], and others [9]. The SIMPLE algorithm of Patankar and Spalding [10] not only provided a remarkably successful implicit method, but has dominated for a decade the field of numerical simulations of incompressible flows [11]. This algorithm based on the finite volume formulation lends itself to easy physical interpretation, and it ensures that if conservation is satisfied for each control volume, it is also satisfied for the entire calculation domain. A clear and detailed description of SIMPLE is given by Patankar [12].

Modelling large-scale three-dimensional time dependent fluid flows poses many challenges. Current numerical methods perform well on fast, single-processor, vector computers; however, the expense of performing such computation is extreme in terms of needed memory and processing time. With the rapid development of parallel computation, modelling large scale three-dimensional, time dependent flows has become realistic by using a large number of processors. Currently, many large scale scientific simulations have been carried out on various parallel systems such as the Intel Paragon, the Cray T3D, and the IBM SP2.

In our present study, the SIMPLE scheme is chosen as the algorithm for the thermal convective flow problems on parallel systems. It is not only because the scheme is used widely in the convective heat transfer community, but also because it is easy to implement on a parallel system. Since the derived equations are only dependent on its local control volume information, domain decomposition techniques can be easily applied. Here we present a numerical study of thermal convection in a large Rayleigh number range using parallel systems. Section 2 describes the mathematical formulation of the three-dimensional, time dependent, thermal cavity flows. The numerical approach for those flows is given in Section 3. The detailed parallel implementation and the code performance are described in Section 4. The numerical solutions for air (Prandtl number Pr=0.733) with various Rayleigh numbers up to 10^7 are discussed in Sections 5. The summary of the present work is given in Section 6.

2. MATHEMATICAL FORMULATION

The flow domain is a rectangular cavity of $0 < x < l$, $0 < y < d$, and $0 < z < h$. The appropriate governing equations, subject to the Boussinesq approximation, can be written in

nondimensional form as

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}=0, \tag{1}$$

$$\frac{1}{\sigma}\left(\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}+w\frac{\partial u}{\partial z}\right)=-\frac{\partial p}{\partial x}+\nabla^2 u, \tag{2}$$

$$\frac{1}{\sigma}\left(\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}+w\frac{\partial v}{\partial z}\right)=-\frac{\partial p}{\partial y}+\nabla^2 v, \tag{3}$$

$$\frac{1}{\sigma}\left(\frac{\partial w}{\partial t}+u\frac{\partial w}{\partial x}+v\frac{\partial w}{\partial y}+w\frac{\partial w}{\partial z}\right)=-\frac{\partial p}{\partial z}+\nabla^2 w+RT, \tag{4}$$

$$\frac{\partial T}{\partial t}+u\frac{\partial T}{\partial x}+v\frac{\partial T}{\partial y}+w\frac{\partial T}{\partial z}=\nabla^2 T. \tag{5}$$

The dimensionless variables in these equations are the fluid velocities u, v, w , the temperature T, and the pressure p, where $\sigma = \nu/\kappa$ is the Prandtl number and $R = g\beta\Delta T h^3/\kappa\nu$ is the Rayleigh number. Here ν is the kinematic viscosity, κ is the thermal diffusivity, β is the coefficient of thermal expansion, and g is the acceleration due to gravity. The rigid end walls on x=0, l are maintained at constant temperatures T_0 and $T_0 + \Delta T$, respectively, while the other boundaries are assumed to be insulating. So the boundary conditions on the rigid walls of the cavity are

$$u = v = w = 0, \quad \text{on } x = 0, l,\ y = 0, d,\ z = 0, h, \tag{6}$$

$$T = 0, \quad \text{on } x = 0, \tag{7}$$

$$T = 1, \quad \text{on } x = l, \tag{8}$$

$$\frac{\partial T}{\partial z} = 0, \quad \text{on } z = 0,\ h, \tag{9}$$

$$\frac{\partial T}{\partial y} = 0, \quad \text{on } y = 0,\ d. \tag{10}$$

In general, the motion is controlled by the parameters σ, R, and the flow domain.

3. NUMERICAL APPROACH

An efficient and practical numerical approach for three-dimensional, time-dependent, thermal-convective flow problems is studied. This implementation is based on the widely used finite volume method (SIMPLE [12]) with an efficient and fast elliptic multigrid scheme for predicting incompressible fluid flows. A normal staggered grid configuration is used and the conservation equations are integrated over a macro control volume (Figure 1: i control volume) in the stag-

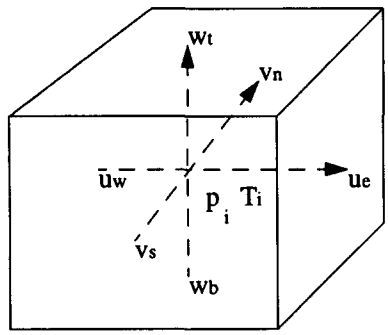


Figure 1. The staggered grid configurations. Velocities are stored at the six surfaces of the control volume marked by $(u_e, u_w, v_n, v_s, w_t, w_b)$, and the temperature and pressure are stored at the center of the control volume (p_i, T_i).

gered grid. Here, velocities are stored at the six surfaces of the control volume marked by $(u_e, u_w, v_n, v_s, w_t, w_b)$, and the temperature and pressure are stored at the center of the control volume (p_i, T_i). Since the solution of the pressure equation derived from the SIMPLE scheme can represent as much as 80% of the total cost for solving the fluid flow problem [11], it is, therefore, a high priority to solve for p in an efficient manner. Here, a multigrid scheme is applied to the discretized equations, which acts as a convergence accelerator and reduces the cpu time significantly for the whole computation. Local, flow-oriented, upwind interpolation functions have been used in the scheme to prevent the possibility of unrealistic oscillatory solutions at high Rayleigh numbers.

A brief summary of the SIMPLE method is outlined here, and details that are omitted here may be found in the original reference [12]. For a guessed pressure field p^*, the imperfect velocity field (u^*, v^*, w^*) based on this pressure will result from the solution of the following discretization equations:

$$a_e u_e^* = \sum a_{unb} u_{nb}^* + b_u + (p_i^* - p_E^*)\,\bar{A}_e, \tag{11}$$

$$a_n v_n^* = \sum a_{vnb} v_{nb}^* + b_v + (p_i^* - p_N^*)\,\bar{A}_n, \tag{12}$$

$$a_t w_t^* = \sum a_{wnb} w_{nb}^* + b_w + (p_i^* - p_T^*)\,\bar{A}_t, \tag{13}$$

where the summation is over the appropriate neighbor points. $\bar{A}_e, \bar{A}_n, \bar{A}_t$ are the areas of the faces of the i control volume at e, n, t, respectively. $a_e, a_n, a_t, a_{unb}, a_{vnb}, a_{wnb}$ and b_u, b_v, b_w are the coefficients of the finite-volume equations. The correct velocity fields are computed by the velocity-correction formula:

$$u_e = u_e^* + de\,(p_i' - p_E'), \tag{14}$$

$$v_n = v_n^* + dn\,(p_i' - p_N'), \tag{15}$$

$$w_t = w_t^* + dt\,(p_i' - p_T'), \tag{16}$$

where

$$de = \frac{\bar{A}_e}{a_e}, \tag{17}$$

$$dn = \frac{\bar{A}_n}{a_n}, \tag{18}$$

$$dt = \frac{\bar{A}_t}{a_t}, \tag{19}$$

and the correct pressure is computed by

$$p = p^* + \alpha p'. \tag{20}$$

α is an under-relaxation factor for the pressure. In the present case, a value of 0.5 is used. p' will result from the solution of the following discretization equation:

$$a_p p_i' = \sum a_{pnb} p_{nb}' + b_p, \tag{21}$$

which is derived from the continuity equation (1) and the velocity-correction equations (14)–(16) described below.

The continuity equation (1) for the control volume in Figure 1 is

$$(u\bar{A})_w - (u\bar{A})_e + (v\bar{A})_s - (v\bar{A})_n + (w\bar{A})_b - (w\bar{A})_t = 0. \tag{22}$$

Introducing equations (14)–(16) into (22) leads to

$$a_p p_i' = a_e p_e' + a_w p_w' + a_n p_n' + a_s p_s' + b_p, \tag{23}$$

which is simplfied in (21) by notations. Here

$$a_e = \left(\bar{A}d\right)_e, \tag{24}$$

$$a_w = \left(\bar{A}d\right)_w, \tag{25}$$

$$a_n = \left(\bar{A}d\right)_n, \tag{26}$$

$$a_s = \left(\bar{A}d\right)_s, \tag{27}$$

$$a_t = \left(\bar{A}d\right)_t, \tag{28}$$

$$a_b = \left(\bar{A}d\right)_b, \tag{29}$$

$$a_p = a_e + a_w + a_n + a_s + a_t + a_b, \tag{30}$$

$$b_p = \left(u^*\bar{A}\right) - \left(u^*\bar{A}\right)_e + \left(v^*\bar{A}\right)_n - \left(v^*\bar{A}\right)_s + \left(w^*\bar{A}\right)_t - \left(w^*\bar{A}\right)_b \tag{31}$$

The temperature will be solved by the following discretization equation:

$$a_T T_i = \sum a_{tnb} T_{nb} + b_T, \tag{32}$$

where $a_p, a_T, a_{pnb}, a_{tnb}$ and b_p, b_T are the coefficients of the pressure and temperature equations resulting from the finite-volume method.

Here five discretized equations for u^*, v^*, w^*, p', T need to be solved at each time level. Typically, this solution step dominates the costs associated with finite volume implementation, with the pressure equation having the highest cost. Therefore, the choice of solution methodology is perhaps one of the most important implementation issues addressed here.

Direct solution methods, such as Gaussian elimination or LU decomposition, have been commonly used in solving linear algebraic equations, and there have been several efforts toward developing parallel implementation of such solvers [13–15]. However, there are some inherent disadvantages of employing these techniques for the parallel solution of large problems. The three-dimensional, time dependent thermal convective flow problems typically result in structured matrices characterized by extremely large bandwidths. The direct solution of such matrices requires a great amount of memory for storage. Since five algebraic equations will be solved, storing the associated matrices becomes extremely expensive.

Here iterative solution techniques are chosen for the above algebraic equations. Since only local coefficients of finite-volume equations are required for an iterative scheme, the total memory is far less than those of direct methods. More importantly, using domain decomposition techniques, iterative methods are readily implemented in a parallel system with a message passing library. But, iterative methods may face converge problems. If the problem size is very large, the number of iterations needed to satisfactorily solve an equation could be significant, or sometimes, the iterative method might not converge at all. So choosing an efficient and fast converging iterative scheme is essential in the present study.

Here a multigrid scheme is applied in solving the above equations. The main idea is to use the solution on a coarse grid to revise the required solution on a fine grid since an error of wavelength λ is most easily eliminated on a mesh of size Δh, where $\lambda \approx \Delta h$. Thus, a hierarchy of grids of different mesh sizes is used to solve the fine grid problem. It has been proven theoretically and practically that the multigrid method has a better rate of convergence than do other iterative methods. Here we do not give a full theoretical analysis of the algorithm, which is described in detail elsewhere [16,17]. In the present computation, a V-cycle scheme with a flexible number of grid levels is implemented with successive over-relaxation as the smoother. Injection and linear interpolation are used as restriction operators and interpolation operators, respectively. A detailed description of a parallel multigrid method is given in the next section, and implementations of sequential multigrid methods can be found in [18].

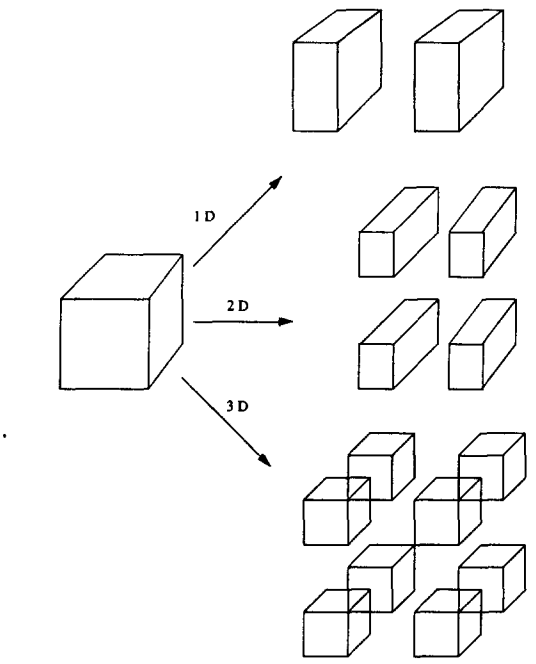


Figure 2. 1D, 2D, and 3D partitions on a flow domain for parallel systems which can be applied according to different geometries.

4. PARALLEL IMPLEMENTATION

In order to achieve load balance and to exploit parallelism as much as possible, a general and portable parallel structure (Figure 2) based on domain decomposition techniques is designed for the three-dimensional flow domain. It has 1D, 2D, and 3D partition features which can be chosen according to problem geometry. Those partition features provide the best choice to achieve load balance so the communication will be minimized. For example, if the geometry is a square cavity, the 3D partitioner can be used, while if the geometry is a shallow cavity with a large aspect ratio, the 1D partitioner in x direction can be applied. Here, MPI is used for communication which is required when subdomains on each processor need neighboring boundary data information. Because of the portability of this software, the code with MPI can be executed on any parallel system which supports the MPI library. Because of the complexity of the implementation, the code is written in C so flexible data structures can be used. The whole parallel computation is carried out by executing the following sequence on each subdomain with a communication procedure added at each iteration step for all flow fields.

1. Choose a domain decomposition structure by applying a 1D, 2D, or 3D partition.
2. Guess the flow field at the initial time step including the velocity u, v, w, the temperature T, and the pressure field p.
3. Advance the flow to the next time step: update the coefficient b .
4. Solve the temperature equation (34) using the multigrid method and exchange data information on partition boundaries.
5. Evaluate the coefficients of the momentum equations and solve the velocity equations (11)–(13) by the multigrid method and exchange data information on partition boundaries.
6. Solve the pressure equation (21) by the multigrid method and exchange data information on partition boundaries.
7. Correct the velocity and the pressure fields by using (14)–(16) and (20).
8. Cycle 4 to 7 until convergence is achieved.
9. Go to next time level and exchange data information on boundaries.
10. Cycle 3 to 9 until convergence for a steady solution is achieved.

The parallel multigrid solver for the pressure equation is summarized as the following: for the pressure equation $a_p p'_i = \sum a_{pnb} p'_{nb} + b_p$, the discrete form is rewritten as $AP^h = b^h$ at mesh level h. Here p', b_p are represented by P, b, and A is the local coefficient matrix of the finite volume pressure equation. And the parallel V-cycle scheme $P^h \leftarrow MP^h(P^h, b^h)$ for N grid levels is outlined as the following.

1. Do $k = 1, N-1$.
   Relax n_1 times on $A^h P^h = b^h$ with a given initial guess P_0^h, and after each relaxation iteration, exchange edge values with neighbors.
   $b^{2h} \leftarrow I_h^{2h}(b^h - A^h P^h), \quad p^{2h} \leftarrow 0$
   Enddo
2. $k = N$ (the coarsest grid), solve $A^h P^h = b^h$.
3. Do $k = N-1, 1$.
   Correct $P^h \leftarrow P^h + I_{2h}^h P^{2h}$.
   Relax n_2 times on $A^h P^h = b^h$ with initial guess P_0^h, and after each relaxation iteration, exchange edge values with neighbors.
   Enddo

Here I_h^{2h} and I_{2h}^h are restriction operators and interpolation operators, respectively, and in the present study, injection and linear interpolation are used. A similar multigrid scheme is used for the velocity and temperature equations.

The algorithm telescopes down to the coarsest grid, which can be a single interior grid, and then works its way back to the finest grid. Currently, the parallel V-cycle scheme with a flexible number of grid level is implemented, which can be adjusted according to the grid size used. The successive over-relaxation was chosen as the smoother, and two times iterations were performed at each grid level.

For low Rayleigh numbers, the initial conditions throughout the flow domain can be set to $T = x/l$ and $u = v = w = p = 0$. For higher Rayleigh numbers, the initial conditions can be generated from a steady flow at a lower Rayleigh number. The application is implemented on the Intel Paragon, the Cray T3D, and the IBM SP2, but it can be easily ported to other distributed memory systems with the MPI library.

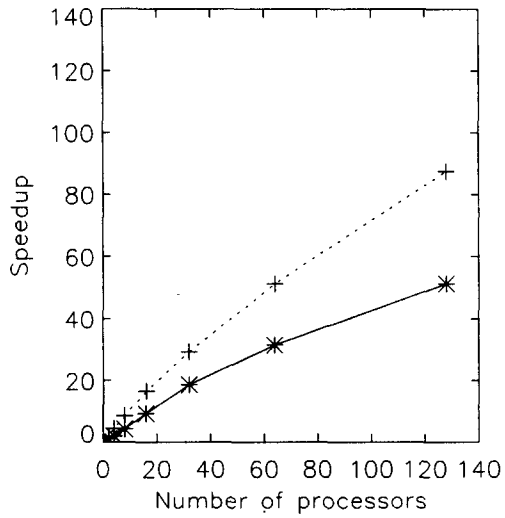


Figure 3. Speed up of the parallel 3D code on the IBM SP2 (*) and Intel Paragon (+).

The speedup measurements of the parallel code are performed on the Intel Paragon, and the IBM SP2. A moderate grid size $64 \times 64 \times 64$ with a test model of $R = 10^5$, $\sigma = 0.733$ in a

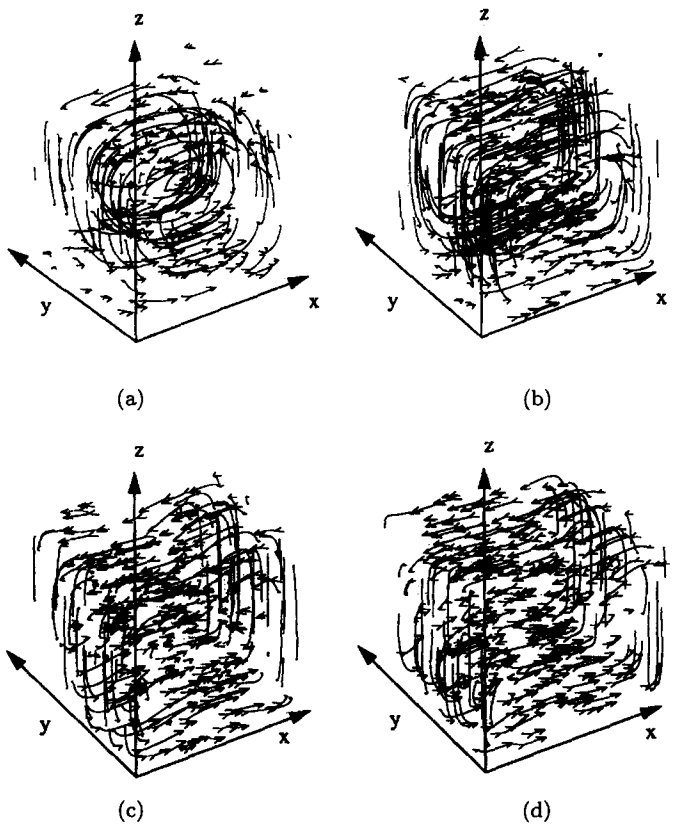


Figure 4. Velocity on the whole domain for Rayleigh numbers 14460 (a), 10^5 (b), 10^6 (c), 10^7 (d).

square box for fixed time steps is used as our test problem. Figure 3 shows the speed up from one processor to 128 processors on the Paragon (+) and the SP2 (*). The line is nearly linear on the Paragon, but on the SP2 the curve bends slightly when the number of processors reaches 64. This is because the computation speed of the SP2 is about five time faster than that of the Paragon for the present problem, but the communication speed on the SP2 is not. So for the same grid size, the ratio of computation to communication changes more rapidly on the SP2.

5. RESULTS AND DISCUSSION

Various numerical tests have been carried out on the 3D code. The results show that the numerical scheme is robust and efficient, and the general parallel structure allows us to use different partitions to suit various physical domains. Here numerical results for the velocity and temperature fields with $R = 14660, 10^5, 10^6, 10^7$, $\sigma = 0.733$ in $0 \leq x, y, z \leq 1$ are presented. The grid sizes $64 \times 64 \times 64$ and $128 \times 128 \times 128$ with 2D and 3D partitions are used for the computation. The computation is stopped when the following conditions are satisfied, corresponding to the achievement of a steady-state solution:

$$\max_{i,j} |T_{i,j}^{k+1} - T_{i,j}^{k}| < \epsilon_1, \tag{33}$$

$$\max_{i,j} |u_{i,j}^{k+1} - u_{i,j}^{k}| < \epsilon_2, \tag{34}$$

$$\max_{i,j} |v_{i,j}^{k+1} - v_{i,j}^{k}| < \epsilon_2, \tag{35}$$

$$\max_{i,j} |w_{i,j}^{k+1} - w_{i,j}^{k}| < \epsilon_2, \tag{36}$$

where k is the time level index and ϵ_1 and ϵ_2 are usually taken to be 10^{-6}.

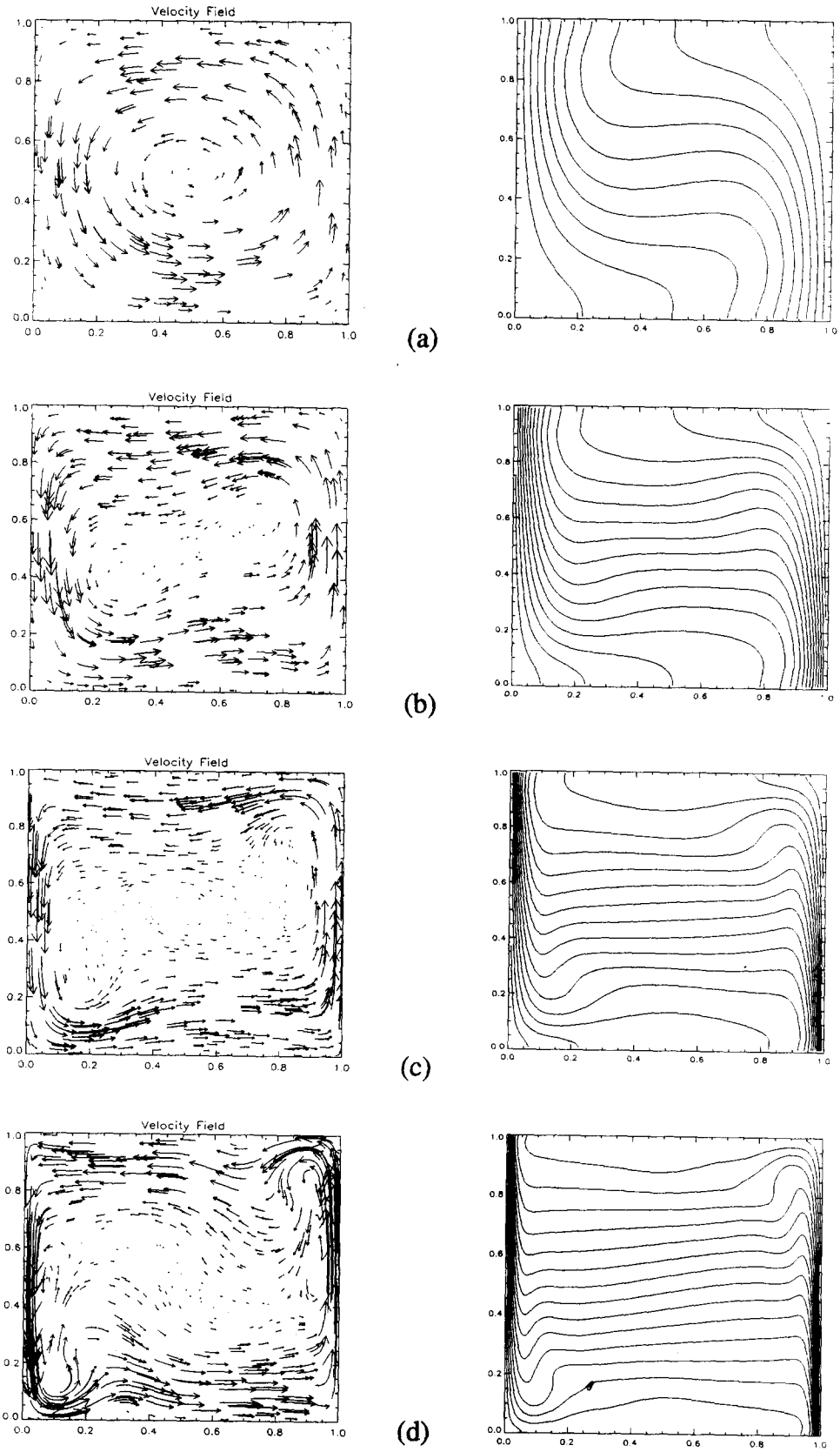


Figure 5. Velocity (left), temperature (right) on $y = 0.5$ for Rayleigh numbers 14460 (a), 10^5 (b), 10^6 (c), 10^7 (d).

Table 1. The overall Nusselt number N_u, the local minimum Nusselt number and the local maximum Nusselt number at the cold wall.

R	σ	N_u	N_{umin}	N_{umax}
10^5	0.733	4.615	0.667	9.148
10^6	0.733	10.575	0.849	25.035
10^7	0.733	21.175	0.901	62.264

Table 2. The overall Nusselt number on the cold wall and the middle plane ($x = 0.5$).

R	N_{uw}	N_{uc}	Discrepancy
10^5	4.615	4.667	-0.052
10^6	10.575	10.414	0.161
10^7	21.175	21.149	0.026

The velocities on the whole flow domain are displayed in Figure 4, which give complete pictures of the three-dimensional flow with various Rayleigh numbers. In Figure 5a, the velocities and the temperatures for $R = 14660$ are illustrated on $y = 0.5$. The flow patterns on the $x - z$ plane are similar to the two-dimensional problems [4,5] with low Rayleigh numbers. It is easy to see that the flow rises from the hot side, travels horizontally, and sinks on the cold side. For $R = 10^5$ in Figure 5b, the velocity field has a slightly tilt, and it no longer remains a single main circulation. Two rolls are formed in the x-z plane velocity field, and the thermal boundary layers are observed on the two sidewalls. When $R = 10^6$ in Figure 5c, the centers of the rolls move further toward the two sidewalls, and the thermal boundary layers are getting thinner. In the middle of the flow domain, besides the two rolls near the sidewalls, a weak roll in the center is visible. Once the Rayleigh number reaches $R = 10^7$ in Figure 5d, the centers of the two rolls move to the lower left corner and the upper right corner, respectively. A stronger roll dominates the middle region.

The core temperature field remains nearly linear, but the structures near the sidewalls become more complicated. Compared with $R = 10^6$, much stronger temperature gradients cover nearly the whole sidewall, and vary more rapidly near the lower left and the upper right corners. On the x-z plane, the flow fields have similar flow patterns as those of the two-dimensional cases, but the 3D solutions give complete pictures of the flows which are more realistic and interesting.

The heat transfer along the cold wall and the center ($x = 0.5$) plane is characterized by the local Nusselt numbers,

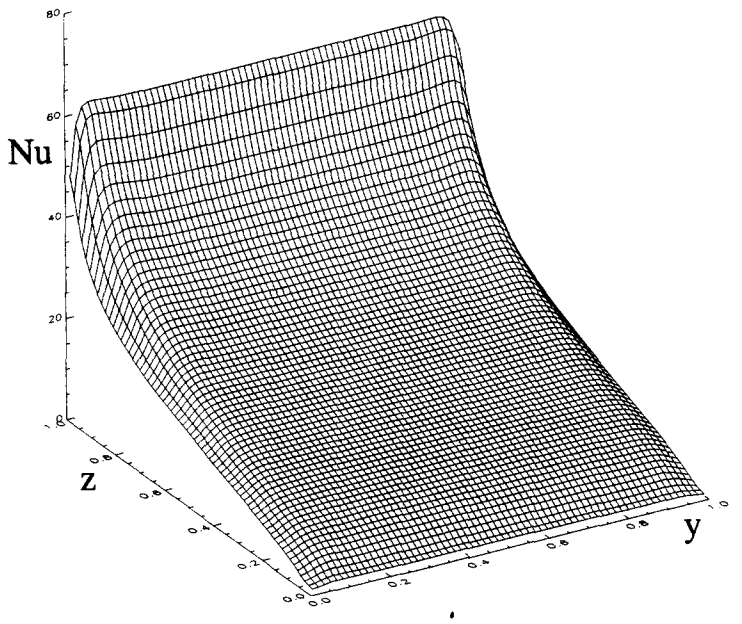
$$N_{u\,loc} = \left.\frac{\partial T}{\partial x}\right|_{x=0}, \tag{37}$$

$$N_{u\,loc} = \left.\frac{\partial T}{\partial x}\right|_{x=0.5} - uT|_{x=0.5}, \tag{38}$$

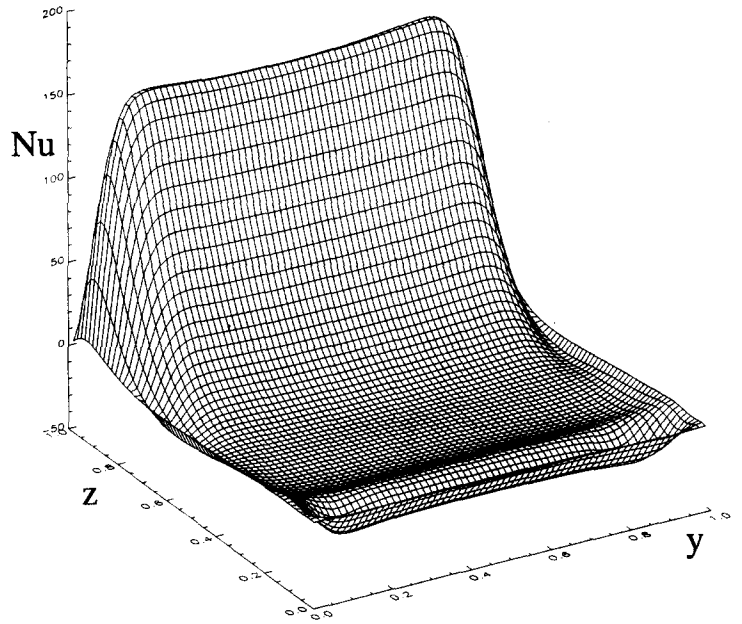
and the overall Nusselt number on the y-z plane is computed by

$$N_u = \int_0^1 \int_0^1 \frac{\partial T}{\partial x}|_x - uT|_x \, dy\, dz. \tag{39}$$

The local minimum, the local maximum, and the overall Nusselt number N_u on the cold wall are shown in Table 1, and indicate how the outward heat transfer is significantly enhanced as R increases. For Rayleigh number 10^7, Figure 6 shows the local Nusselt numbers on the cold wall and on the middle plane ($x = 0.5$), which gives quite different local heat transfer properties. On the cold wall, the local Nusselt numbers increase sharply when $z \to 1.0$ because of the thin thermal boundary layers set up near the walls, and the local Nusselt numbers vary little in y



(a)



(b)

Figure 6 The local Nusselt numbers on the cold wall (a), and on the middle y-z plane (b).

direction except near the south wall ($y = 0.0$) and the north wall ($y = 1.0$). On the midplane, the local Nusselt numbers change significantly in z direction, and have some negative values near the lower half zone ($z < 0.5$). The range of the local minimum and maximum Nusselt number is much larger than that on the cold wall. As the total Nusselt number N_u can be commonly used as an indicator of the approach to a steady state, the overall Nusselt number should remain the same if a steady state is achieved. So the Nusselt number N_{uw} on the cold wall and the Nusselt number N_{uc} on the midplane can be used to provide a check on the accuracy of the numerical solution. Table 2 lists the Nusselt numbers in each case with a different Rayleigh number to test the accuracy of the numerical results. This result indicates very good consistency.

6. CONCLUSIONS

We have successfully implemented the finite volume method with an efficient and fast multigrid scheme to solve for three-dimensional, time-dependent, incompressible fluid flows on distributed memory systems. The parallel code is numerically robust, computationally efficient, and portable to parallel architectures which support MPI for communications. The present parallel code has a very flexible partition structure which can be used for any rectangular geometry by applying a 1D, 2D, or 3D partitioning to achieve load balance. This feature allows us to study various thermal cavity flows with different geometries. In addition to the Prandtl number and the Rayleigh number, the geometry is the other major factor determining the flow structure [5]. The code shows very good speedup which allows us to simulate large scale problems using a larger number of processors. We have carried out some high Rayleigh number flow simulations which demonstrate the capabilities of this parallel code.

The new three-dimensional numerical results are obtained for various Rayleigh numbers ranging from 14,660 to 10^7. They give a complete description of the three-dimensional flow, with the flow field gradually changing to a multiple roll structure from a single flow circulation as the Rayleigh number increases. Very thin flow boundary layers are formed on the two sidewalls. The overall Nusselt numbers on the cold wall with various Rayleigh numbers are also calculated which show the strong dependence of Rayleigh numbers. In spite of the difficulties associated with large Rayleigh number simulation, our results illustrated here clearly demonstrate the great potential for applying this approach to solving much higher Rayleigh number flow in realistic, three-dimensional geometries using parallel systems with large grid sizes. Much higher Rayleigh numbers computations and transient features of thermal convection in 3D are under investigation.

REFERENCES

1. D.E. Cormack, L.G. Leal and J.H. Seinfeld, Natural convection in a shallow cavity with differentially heated end walls, Part 2. Numerical solutions, *J. Fluid Mech.* **65**, 231–246, (1974).
2. G. de Vahl Davis and I.P. Jones, Natural convection in a square cavity A comparison exercise, *Int. J. Numer. Methods in Fluids* **3**, 227–248, (1983).
3. P.G. Daniels, P.A. Blythe and P.G. Simpkins, Onset of multicellular convection in a shallow laterally heated cavity, *Proc. Roy. Soc. A* **411**, 327–350, (1987).
4. N.G. Wright, Multigrid solutions of elliptic fluid flow problems, Ph.D Dissertation, University of Leeds, United Kingdom, (1988).
5. P. Wang, Thermal convection in slender laterally-heated cavities, Ph.D Dissertation, City University, London, (1993).
6. P. Wang and P.G. Daniels, Numerical solutions for the flow near the end of a shallow laterally heated cavity, *J. Eng. Math.* **28**, 211–226, (1994).
7. F.H. Harlow and J.E. Welch, Numerical calculations of time-dependent viscous incompressible flow of fluid with free surface, *Phys. Fluids* **8** (12), 2182–2189, (1965).
8. A.J. Chorin, Numerical solution of Navier-Stokes equations, *Math. Comp.* **22**, 745–762, (1968).
9. P.J. Roache, *Computational Fluid Dynamics*, Hermosa, Albuquerque, NM, (1976).
10. S.V. Patankar and D.B. Spalding, A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows, *Int. J. Heat Mass Transfer* **15**, 1787, (1972).
11. J.P. Van Doormaal and G.D. Raithby, Enhancements of the SIMPLE method for predicting incompressible fluid flows, *Numerical Heat Transfer* **7**, 147–163, (1984).
12. S.V. Patankar, *Numerical Heat Transfer and Fluid Flow*, Hemisphere, New York, (1980).
13. J. Dongarra and L. Johnsson, Solving banded systems on a parallel processor, *Parallel Comput.* **5**, 219–246, (1987).
14. I.S. Duff, Direct solvers, *Comput. Phys. Rep.* **11**, 21–50, (1989).
15. H.Q. Ding and R.D. Ferraro, A general purpose sparse matrix parallel solvers package, In *9^{th} International Parallel Processing Symposium*, pp. 70–76, (1995).
16. A. Brandt, Multilevel adaptive solutions to boundary-value problems, *Math. Comp.* **31**, 333–390, (1977).
17. S.F. McCormick, *Multilevel Adaptive Methods for Partial Differential Equations, Frontiers on Applied Mathematics*, SIAM, Philadelphia, PA, (1989).
18. W.L. Briggs, *A Multigrid Tutorial*, Society for Industrial and Applied Mathematics, Philadelphia, PA, (1987).