JOURNAL OF COMPUTATIONAL PHYSICS **106**, 286–298 (1993)

A Vorticity–Velocity Method for the Numerical Solution of 3D Incompressible Flows

G. GUJ AND F. STELLA

Dipartimento di Meccanica e Aeronautica, Via Eudossiana 18, 00184 Rome, Italy

Received October 3, 1990

A new method for the numerical solution of the 3D Navier–Stokes equations written in terms of vorticity–velocity is presented. The advantages of this formulation with respect to primitive variables and vorticity–vector–potential ones are discussed in view of physical as well as engineering applications. A suitable form of the continuum equations, the most appropriate discretization scheme, and variable location in order to guarantee the solenoidality of the velocity and vorticity fields are introduced and justified. A 3D lid driven cavity problem for $400 \leqslant \mathrm{Re} \leqslant 3200$ is chosen as a test case for comparison and validation purposes. A parallel implementation of the method as been performed on a shared memory architecture mainframe. Speedup results and efficiency considerations are given and discussed. © 1993 Academic Press, Inc.

1. INTRODUCTION

In the last decade, developments in computer hardware technology, in particular the availability of vector and parallel processors, have allowed the numerical treatment of complex three-dimensional (3D) flow fields. The discussion about the most appropriate mathematical formulation of the Navier–Stokes equation to simulate these flows is still open, being the choice strictly dependent on the problem domain and boundary conditions.

The mathematical formulations of the 3D Navier–Stokes equations may be classified into the following categories, depending on the choice of dependent variables, as primitive variables [1], vorticity–vector–potential [2], and vorticity–velocity [3, 4]. For a general review of the peculiarities of the three formulations we refer to [5], where a large number of references are mentioned and not repeated here.

In this paper we present a numerical method for the study of steady 3D flows which is based on the vorticity–velocity formulation of the Navier–Stokes equations. This formulation has the distinct advantage, with respect to the other two, when applied to some categories of problems. First, the vorticity–velocity ($\boldsymbol{\omega}$-$\mathbf{u}$) formulation is simpler than the primitive-variable one ($\mathbf{u}$, P), since the pressure does not appear explicitly in the field equations and thus the well-known difficulty connected with the determination of the pressure boundary value in incompressible flows is avoided [6]. Furthermore, as demonstrated in [7], the $\boldsymbol{\omega}$-$\mathbf{u}$ form has a striking advantage when applied to problems in a non-inertial frame of reference because the non-inertial effects only enter into the solution of the problem through the implementation of the boundary conditions. From the computational point of view the solution of the elliptic pressure equation in the primitive-variable formulation requires most of the computer effort, increasing with the number of mesh points; e.g., about 50% of CPU time is taken by the pressure solver using 24,500 mesh points [8], and about 80% using 116,000 mesh points [9]. Moreover, as pointed out by [10] in the study of external attached flows, the region in which the vorticity is computationally negligible is large and the rotational region is located in the boundary layer and in the wake. This means that the vorticity transport equation has to be solved only in these last-mentioned regions. On the other hand, the primitive variable formulation is more versatile when pressure boundary conditions are prescribed. Furthermore, in multiconnected problems the condition for the pressure to be single valued is explicitly satisfied [11].

On comparing the proposed $\boldsymbol{\omega}$-$\mathbf{u}$ method with the vorticity–vector–potential formulation, only the kinematic aspect has to be considered because the dynamic aspect is governed by the vorticity transport equation in both formulations. The adoption of the vector potential determines a velocity field that identically satisfies the mass conservation equation. But the vector potential itself is not uniquely defined; it is defined except for a gradient of a harmonic scalar quantity which allows for a multiplicity of boundary formulations for the vector potential equations [12]. Furthermore, the study of through-flow problems requires the introduction of a further dependent variable which is the scalar potential [12].

In the past few years the authors have been involved in

0021-9991/93 $5.00

the development and application of the vorticity–velocity form of Navier–Stokes equations. The basic 2D mathematical and numerical formulation is described in [13].

The purpose of the present paper is to give a systematic description of the 3D version of the method which has been previously applied by the authors with regards to Raileigh–Bénard convection [14, 15], implementation on a parallel computer [16], and comparison with vorticity–vector–potential formulation [17].

Particular attention is given here to the numerical technique which may be summarized in the following steps:

(1) The variables are located on a staggered grid in order to satisfy the continuity equation and the solenoidality constraint on the vorticity on individual cells.

(2) The equations are discretized by second-order accurate central differences on a uniform mesh.

(3) The conservative form is adopted for the vorticity transport equation in order to verify conservation of the mean vorticity and solenoidality of vorticity.

(4) The false transient method used in [13] is adopted to speed up the convergence of the numerical method, so that the governing equations are parabolized in time with a result that they are solved exactly at the steady state only.

(5) An alternating direction implicit (ADI) method of a scalar type proposed by [18] is used to integrate in time the six equations governing $\boldsymbol{\omega}$ and $\mathbf{u}$.

(6) The numerical procedure is implemented in a vectorized and parallelized code designed to run efficiently on a vector and parallel computer for a typical mesh up to $101 \times 101 \times 101$ points.

The method is applied to the test problem of the laminar steady flow at $\mathrm{Re} \leqslant 3200$ in a cubical driven cavity; comparison with other numerical [1] and experimental results [19] are presented.

2. VORTICITY–VELOCITY FORMULATION OF NAVIER–STOKES EQUATIONS

2.1. *Mathematical Formulation*

The nondimensional Navier–Stokes equation of a laminar flow of an incompressible Newtonian fluid is

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\nabla p + \frac{1}{\mathrm{Re}} \nabla^2 \mathbf{u}. \tag{1}$$

Here Re is $u'L/\nu$, where u' is the reference velocity, L is the reference length, and ν is the kinematic viscosity.

The Lagrange form of the advective term that is more suitable for deriving the vorticity transport equation is obtained from the following vectorial identity,

$$\mathbf{u} \times (\nabla \times \mathbf{u}) = \tfrac{1}{2}\nabla(\mathbf{u} \cdot \mathbf{u}) - (\mathbf{u} \cdot \nabla)\mathbf{u}, \tag{2}$$

and substituting the vorticity definition, $\boldsymbol{\omega} \times \mathbf{u}$ in (2), the form of the Navier–Stokes equation follows:

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{1}{2}\nabla(\mathbf{u} \cdot \mathbf{u}) + (\boldsymbol{\omega} \times \mathbf{u}) = -\nabla p + \frac{1}{\mathrm{Re}} \nabla^2 \mathbf{u}. \tag{3}$$

The vorticity transport equation is then derived by applying the curl operator to both terms of (3):

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + \nabla \times (\boldsymbol{\omega} \times \mathbf{u}) = \frac{1}{\mathrm{Re}} \nabla^2 \boldsymbol{\omega}. \tag{4}$$

This 3D form is called conservative in analogy with the well-known 2D conservative form of the vorticity transport equation that can be derived by a simple projection of (4) on the x_1–x_2 plane.

The velocity equations are derived by applying the curl operator to the vorticity definition,

$$\nabla \times \boldsymbol{\omega} = \nabla \times (\nabla \times \mathbf{u}) = \nabla(\nabla \cdot \mathbf{u}) - \nabla^2 \mathbf{u} \tag{5}$$

and using the continuity equation for incompressible flows $\nabla \cdot \mathbf{u} = 0$, the following kinematic relation is obtained:

$$\nabla^2 \mathbf{u} = -\nabla \times \boldsymbol{\omega}. \tag{6}$$

Note that (6) can be regarded also as a consequence of the Helmholtz decomposition theorem, in which only the vortical velocity is present.

2.2. *Boundary Conditions*

The present formulation in terms of derived variables leads to a simpler way to enforce the proper boundary conditions that what is allowed by the primitive variable formulation, whenever the pressure does not appear as a known value on the boundary. On the other hand, the problem of the difficulty of determining the vector–potential boundary conditions in the vorticity–vector–potential formulation [20] is overcome in the present one. In a general problem the boundary conditions associated with Eqs. (4) and (6) are:

— the boundary condition associated with Eq. (4) is the vorticity definition written on the boundary

$$\boldsymbol{\omega}_w = (\nabla \times \mathbf{u})_w. \tag{7}$$

As discussed in Section 2.4, this condition is essential for the conservation of the solenoidality of vorticity field, but it requires the solution of the coupled problem for the conservation of mass.

— the boundary condition associated with the elliptic velocity equation (6) is the velocity vector assigned on the edges of the solution domain

$$\mathbf{u} = \bar{\mathbf{u}} \tag{8}$$

(e.g., $\mathbf{u} = \mathbf{0}$ on a solid wall of a body, corresponding to no slip and impermeability conditions).

The specialized form of these boundary conditions is presented below for the test case.

2.3. *Mass Conservation*

In the present formulation, the solenoidality constraints on the velocity field are implicitly satisfied by the equation system (4) and (6), because this has been derived by the differential manipulation from the Navier–Stokes and mass conservation equations in primitive-variable form, as described in Section 2.1. Let us note that the kinematic equations (6) and the velocity boundary conditions (8) cannot assure a solenoidal velocity field for an arbitrary vorticity distribution [21], so this property may be assured only by coupling equations (4) and (6). In any event, an analogous coupling of the dynamic and kinematic aspects is required also for primitive-variable and vorticity–vector-potential formulations as was demonstrated by [6, 22]. In analogy with vorticity–vector–potential formulation [22], the decoupling of the two aspects should require in the present formulation as well, either a numerical coupling between kinematic and dynamic equations or an integral condition on the vorticity field. To these authors' knowledge, as also stated by [5], this integral condition has not yet been derived. Therefore, we pay particular attention in the coupled solution of Eqs. (4) and (6), as is explained in Section 3.1.

2.4. *Solenoidality of Vorticity Field*

A very important condition to enforce in the computation of flow fields is the solenoidality of vorticity. This condition is identically satisfied in 2D problems, due to the orthogonality between the vorticity vector and the plane of motion. On the other hand, this condition has to be enforced in some way in 3D problems. For this purpose, let us demonstrate now that the form of the advective term in Eq. (4) is essential for a straightforward satisfaction of this constraint. In fact by applying the divergence operator to Eq. (4) the advective term vanishes and the equation governing the divergence of vorticity reduces to the simple diffusive equation

$$\frac{\partial D\boldsymbol{\omega}}{\partial t} - \frac{1}{\mathrm{Re}} \nabla^2 D\boldsymbol{\omega} = 0, \tag{9}$$

where $D\boldsymbol{\omega}$ is the divergence of vorticity

$$D\boldsymbol{\omega} = \nabla \cdot \boldsymbol{\omega} \tag{10}$$

that has to be zero as stated above. Equation (10) is completed by the initial condition $D\boldsymbol{\omega}(t=0) = 0$ and by the boundary condition

$$[D\boldsymbol{\omega}]_w = [\nabla \cdot \nabla \times \mathbf{u}]_w = 0 \tag{11}$$

which is homogeneus, since the vorticity definition is satisfied: (i) explicitly on the boundary by Eq. (7), (ii) implicitly in the bulk of the fluid field by the kinematic relations (6).

Equation (10) governs the conservation of the divergence of $\boldsymbol{\omega}$, via a diffusive homogeneous equation of $D\boldsymbol{\omega}$ with homogeneous boundary and initial conditions. This equation admits only the trivial solution $D\boldsymbol{\omega}(t) = 0$. Furthermore, if a nonphysical initial condition is enforced, leading to $D\boldsymbol{\omega}(t=0) \neq 0$, $D\boldsymbol{\omega}(t)$ tends to zero for t going to infinity. This is very important if a steady state solution is to be found asymptotically via an iterative procedure. These considerations confirm that the solenoidality condition on the vorticity is implicitly imposed at the continuum level by writing the vorticity transport equation (4) with the vorticity definition used to determine boundary conditions.

3. DISCRETIZED SCHEME

3.1. *Time Integration and Iterative Procedure*

As was explained in Section 2, the solenoidality constraints on velocity and vorticity fields require a coupled solution of the system given by Eqs. (4) and (6), with boundary conditions (7) and (8). Unfortunately, the number of unknowns in 3D domains discourages the use of a direct solver for the entire problem. Furthermore, an iterative procedure is also needed due to the presence of the non-linear advective term in Eq. (4).

In the proposed numerical method the necessary iterative procedure for steady state solution is based on a false transient method [23], which solves the parabolic equation (4) and the following parabolized version of the velocity equation

$$\alpha \frac{\partial \mathbf{u}}{\partial \mathbf{t}} - \nabla^2 \mathbf{u} - \nabla \times \boldsymbol{\omega} = \mathbf{0}, \tag{12}$$

where α is a relaxation parameter. It is obvious that the steady state solution is reached when the time derivatives approach zero. On the other hand, the solution of an unsteady flow requires an inner loop on the iterative procedure for each physical time step [24]. The time integration procedure is performed using a simple ADI algorithm on each scalar equation [18].

3.2. *Spatial Discretization*

The finite difference approximation (FD) of the governing equations are derived by replacing the time derivatives with forward differences and the spatial first- and second-order derivatives with second-order central differences.

The location of the staggered variables is shown in Fig. 1. Assuming the corner (i, j, k) as the reference point, the following localization of the variable results:

- u_1 $(i, j-\frac{1}{2}, k-\frac{1}{2})$
- u_2 $(i-\frac{1}{2}, j, k-\frac{1}{2})$
- u_3 $(i-\frac{1}{2}, j-\frac{1}{2}, k)$
- ω_1 $(i-\frac{1}{2}, j, k)$
- ω_2 $(i, j-\frac{1}{2}, k)$
- ω_3 $(i, j, k-\frac{1}{2})$.

The staggering of the variable location (Fig. 1) is chosen not only to obtain the maximum accuracy of the discretized derivatives, but, also to ensure the conservation of mass and vorticity at the discrete level; something which, as explained in the previous section, is not explicitly nor identically imposed in the present formulation. By analogy with the two-dimensional case [13], it is possible to satisfy mass conservation, to round-off-error level, if the velocity component u_i is located at the middle of the face of the computational cell which is normal to x_i. In this way the computational molecule may be regarded as a control volume for mass conservation, which is satisfied to round-off-error level on each individual computational cell. Similarly, each vorticity component is located at the mid-point of the edge of the cell parallel to the corresponding axis, and a control volume, shifted by a half spatial step in the three directions, is considered to impose the solenoidality of vorticity. Furthermore, the proposed staggered variable location guarantees in the two-point formula an accuracy of $(\delta x/2)^2$ in the discretization of the first derivative instead of δx^2, that can be obtained using a not staggered variable location. This is particularly clear in the computation of the right-hand side of Eq. (6).

The correct location of the variables of the problem is found by using the general rule of applying the discrete operators. This rule cannot be used in a straightforward manner in discretizing the nonlinear terms. In fact, as is shown in the following discrete manipulation of Eq. (4), the staggered variables location is not sufficient to ensure the solenoidality of vorticity field in the *discrete form*, but it is also required that, in obtaining the FD approximation for the advective term, $\nabla \times (\boldsymbol{\omega} \times \mathbf{u})$, all needed averaging is performed on the product $(\boldsymbol{\omega} \times \mathbf{u})$ and not on the individual $\boldsymbol{\omega}$ and $\mathbf{u}$ terms.

Let us consider as an example the two terms which follow from the vorticity transport equation for ω_1 and ω_3 differentiated with respect to x_1 and x_3, respectively:

$$\left\{\frac{\delta}{\delta x_1}\left[-\frac{\delta}{\delta x_3}(\omega^3 u^1)\right]\right\}_{ijk} + \left\{\frac{\delta}{\delta x_3}\left[\frac{\delta}{\delta x_1}(\omega^3 u^1)\right]\right\}_{ijk}, \tag{13}$$

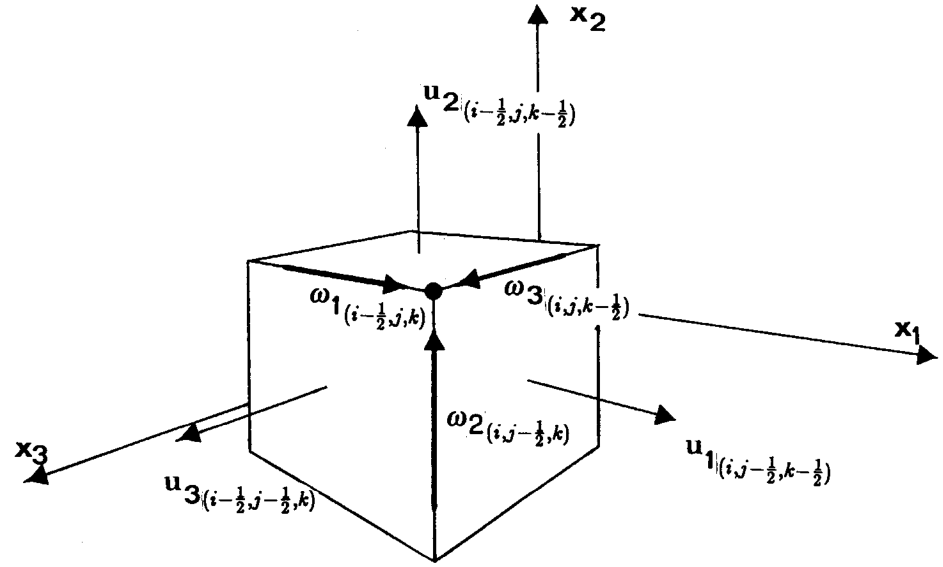


FIG. 1. Computational molecule showing the 3D staggered mesh. The reference point (i, j, k) is marked with (•).

where δ represents the discrete derivative. Here and in the Eqs. (14)–(19) the superscript defines the quantities computed by an interpolation operation and the subscript defines quantities located in the computational mesh points (Fig. 1). The first term of Eq. (13) may be manipulated as

$$\left\{\frac{\delta}{\delta x_1}\left[-\frac{\delta}{\delta x_3}(\omega^3 u^1)\right]\right\}_{ijk} = \frac{1}{\Delta x_1}\left\{\left[-\frac{\delta}{\delta x_3}(\omega^3 u^1)\right]_{i+1/2jk} - \left[-\frac{\delta}{\delta x_3}(\omega^3 u^1)\right]_{i-1/2jk}\right\}, \tag{14}$$

where

$$\left[\frac{\delta}{\delta x_3}(\omega^3 u^1)\right]_{i-1/2jk} = \frac{1}{\Delta x_3}\{(\omega^3 u^1)_{i-1/2jk+1/2} - (\omega^3 u^1)_{i-1/2jk-1/2}\} \tag{15}$$

and

$$(\omega^3 u^1)_{i-1/2jk-1/2} = \tfrac{1}{2}[(\omega_3 u^1)_{ijk-1/2} + (\omega_3 u^1)_{i-1jk-1/2}]. \tag{16}$$

The velocity u^1 in $i-1, j, k-\frac{1}{2}$ is

$$(u^1)_{i-1jk-1/2} = \tfrac{1}{2}[u_{1_{i-1j+1/2k-1/2}} + u_{1_{i-1j-1/2k-1/2}}]. \tag{17}$$

In a similar manner, the complete discrete form of the first term of (14) may be found. The second term of (13) in discrete form is given by

$$\left\{\frac{\delta}{\delta x_3}\left[\frac{\delta}{\delta x_1}(\omega^3 u^1)\right]\right\}_{ijk} = \frac{1}{\Delta x_3}\left\{\left[\frac{\delta}{\delta x_1}(\omega^3 u^1)\right]_{ijk+1/2} - \left[\frac{\delta}{\delta x_1}(\omega^3 u^1)\right]_{ijk-1/2}\right\}, \tag{18}$$

where the explicit expression of the second term on the right-hand side is

$$\left[\frac{\delta}{\delta x_1}(\omega^3 u^1)\right]_{ijk-1/2} = \frac{1}{2\Delta x_1}[(\omega_3 u^1)_{i+1jk-1/2} - (\omega_3 u^1)_{i-1jk-1/2}]. \tag{19}$$

Let us note now that the second term of the right-hand side of (19) is identically balanced by the corresponding term in (16) and, similarly, the other terms will cancel each other. This means that the solenoidality of vorticity is governed by Eq. (9) at the discrete level as well because the discrete term $\nabla \times (\boldsymbol{\omega} \times \mathbf{u})$ is identically zero. Consequently $\nabla \cdot \boldsymbol{\omega}$ is satisfied to the round-off-error level in the computed solution.

3.3. *Vectorization and Parallelization Technique*

The proposed ADI method is particularly suited for both vectorization and parallelization. For example, let us consider the first component of the kinematic relations (12), which due to the splitting operated by the Samarskii–Andreyev ADI, is reduced to a tridiagonal system of the form

$$a_i u_{i-1jk} + b_i u_{ijk} + c_i u_{i+1jk} = d_i, \tag{20}$$

where

$$\begin{aligned} a_i &= -\frac{\Delta t}{\alpha\, \Delta x^2} \\ b_i &= 1 + \frac{2\Delta t}{\alpha\, \Delta x^2} \\ c_i &= -\frac{\Delta t}{\alpha\, \Delta x^2} \\ d_i &= R(u_{ijk}) \end{aligned} \tag{21}$$

and $R(u_{ijk})$ is the residual of the discretized form of Eq. (12). In this way $N_j \times N_k$ linear systems of type (20) are obtained. Each system is written for a line of unknowns (j, k). As is clear from definitions (21) the various linear systems are decoupled because the coefficients a_i, b_i, c_i, d_i do depend on the values of the unknowns at the previous time step. So the linear systems can be written together and solved simultaneously, and the complete vectorization and parallelization of the code is obtainable.

This technique is particularly suited for 3D problems because the length of the generated vector (number of different linear systems) is in general very large, being connected with the product of the two residual dimensions N_j and N_k as shown in the previous example. Hence, for a typical 3D mesh of 101 × 101 × 101 the length of the vector is of the order of 10.000.

Due to the limitation of the section size of the vector register (e.g., 128 or 256 for IBM3090VF and 64 for CRAY-XMP) it is not possible to process the whole vector in a single step. Hence, a combination of vector and parallel strategies can be used to speed up the numerical solver. Therefore, the vector is split in segments devoted to different tasks and processed by the first available CPU [16]. The

TABLE I

Mesh Sensitivity to Degrees of Freedom (Symmetry Assumed) and Extrapolation for Zero Mesh Size

Mesh	DOF	$u_{1\,\min}$	Percentage of error	$u_{2\,\max}$	Percentage of error	$u_{2\,\min}$	Percentage of error
A	92256	−0.22036	21.9	0.19656	20.6	−0.36198	17.1
B	230850	−0.24553	12.9	0.21547	12.9	−0.38389	12.1
C	727218	−0.26290	6.8	0.23097	6.7	−0.40942	6.2
D	2509446	−0.27255	3.4	0.23939	3.3	−0.42334	3.1
Extr.		−0.28203	—	0.24747	—	−0.43669	—

Note. $u_{1\,\min}$ is evaluated on the vertical centerline; $u_{2\,\max}$ and $u_{2\,\min}$ are evaluated on the horizontal centerline; percent errors are evaluated with respect to the extrapolated solution. Mesh types are: (A) $31 \times 31 \times 16$, (B) $45 \times 45 \times 19$, (C) $67 \times 67 \times 27$, (D) $101 \times 101 \times 41$.

numerical implementation of the parallel version of the code is obtained using IBM Parallel FORTRAN DISPATCH statement.

The length of the segments per task should be selected to be large enough to do enough work to justify the process allocation, but small enough to allow a flexible process dispatching.

In the frame of an optimal use of a multiple vector facilities, the vectorial architecture suggests to select a dimension of the task that equals the section size of the vector register (128 for a IBM3090 mod.E) or a multiple of it. Preliminary tests have demonstrated that the task with 128 elements guarantees a negligible relative dispatching overhead and therefore has been selected for the present work.

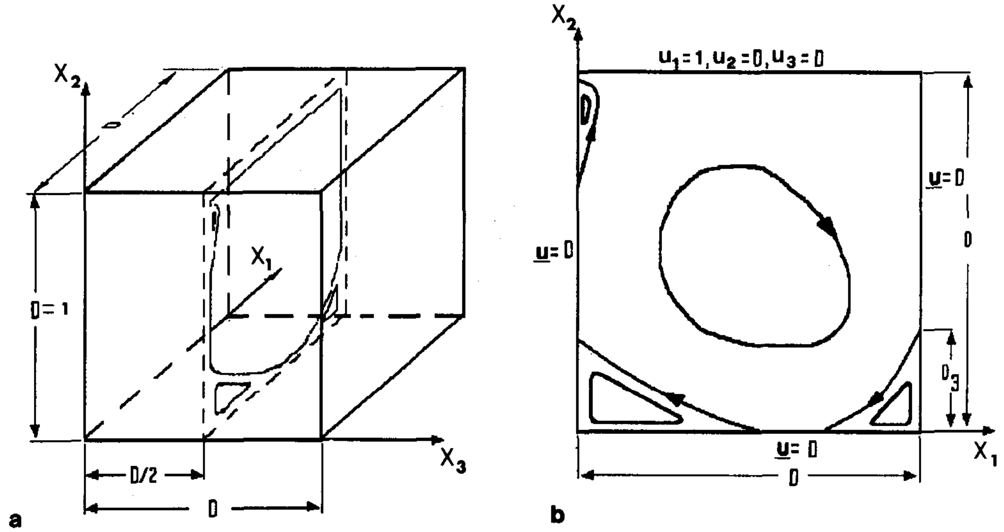
Furthermore, this way of parallelization does not modify the natural sequence of the program and the obtained results are bit by bit identical.

4. RESULTS

The lid-driven flow in a cubic cavity ($1 \times 1 \times 1$) shown in Fig. 2 is the natural extension to 3D of the 2D driven cavity test case, which has been widely used for validation and comparison purposes. The reason of this choice results from the simplicity of the geometry and from the number of numerical and experimental results available. In the present work we consider a Re range up to 3200 for which steady transversal vortices are expected to be present. A typical mesh up to $101 \times 101 \times 81$ points is employed. Symmetric boundary conditions on the third direction have been adopted in many of the computations to reduce the number of unknowns. A mesh of $101 \times 101 \times 101$ is used only in the parallel computations.

4.1. *Mesh Sensitivity*

The study of mesh dependence has been conducted first to validate the grid adopted in obtaining the results presented in the paper. We have selected the case at Re = 1000. The mesh on a half of the domain is varied from $31 \times 31 \times 16$ to $101 \times 101 \times 41$ (symmetry assumed), corresponding to a number of degrees of freedom (DOF) from 92,256 to 2,509,446. The results of the mesh sensitivity analysis are presented in Table I in terms of the maximum vertical and



[illegible]

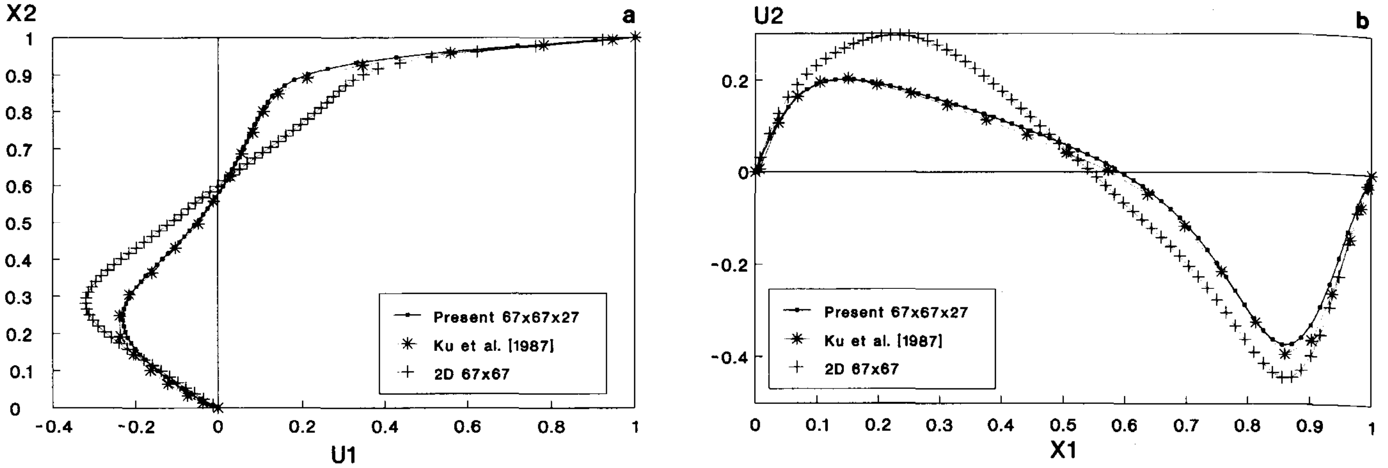


FIG. 3. Velocity profiles for Re = 400: (a) u_1 component on vertical centerline; (b) u_2 component on horizontal centerline.

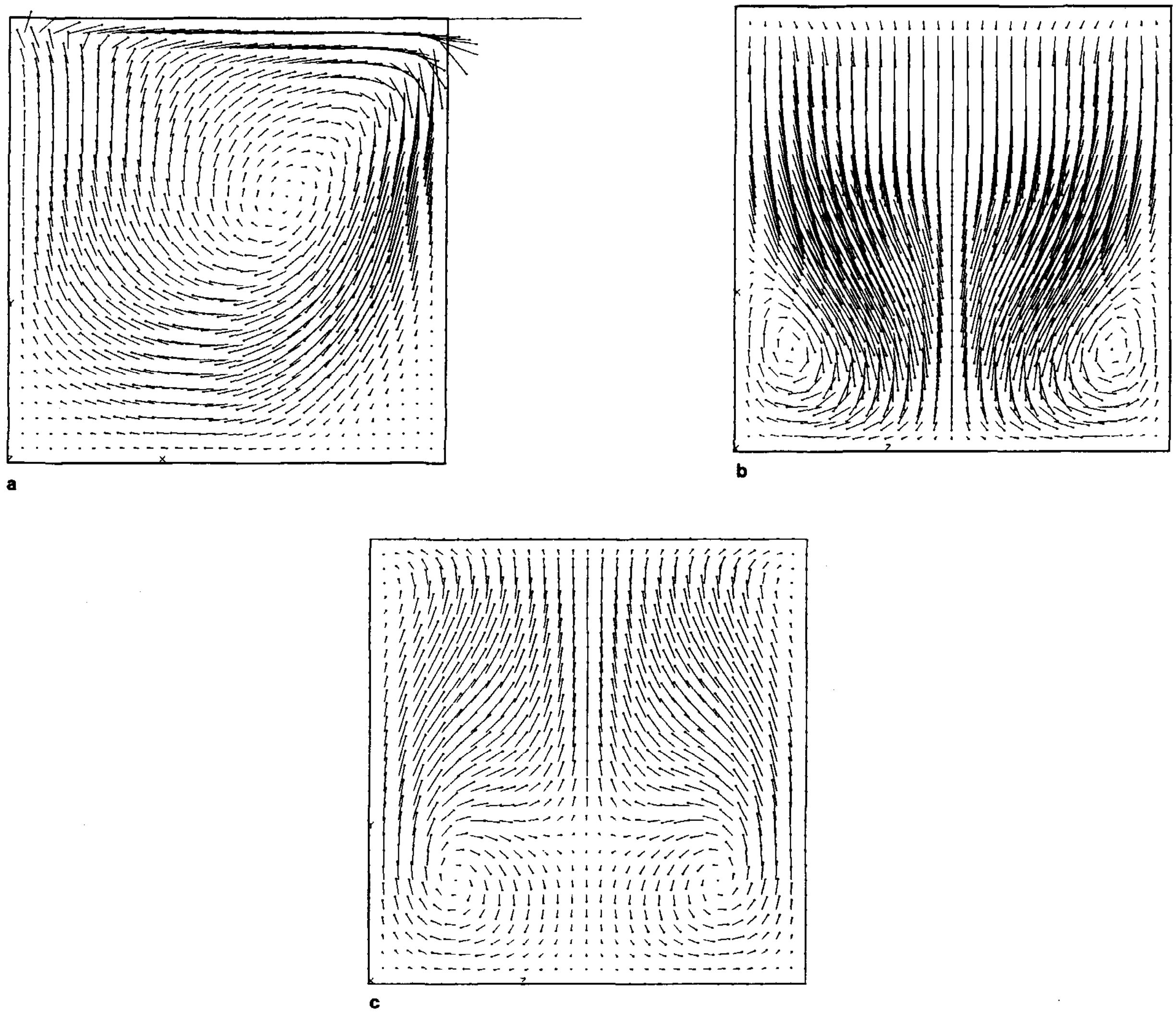


FIG. 4. Velocity vectors plots for Re = 400 projected on: (a) x_1–x_2 midplane; (b) x_1–x_3 midplane; (c) x_2–x_3 midplane.

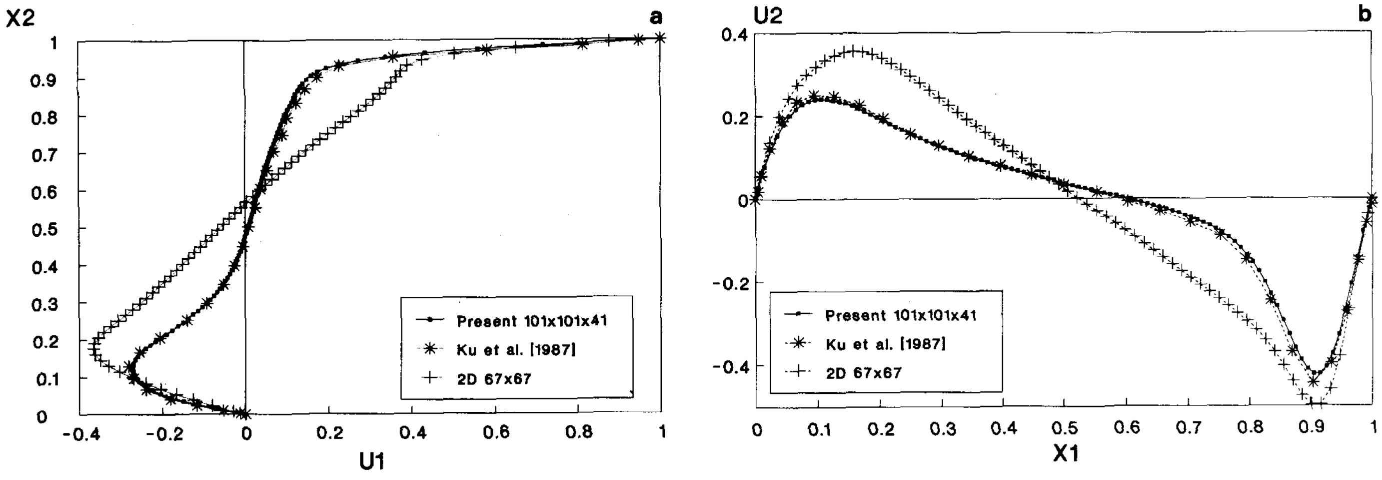


FIG. 5. Velocity profiles for Re = 1000 on: (a) vertical centerline; (b) horizontal centerline.

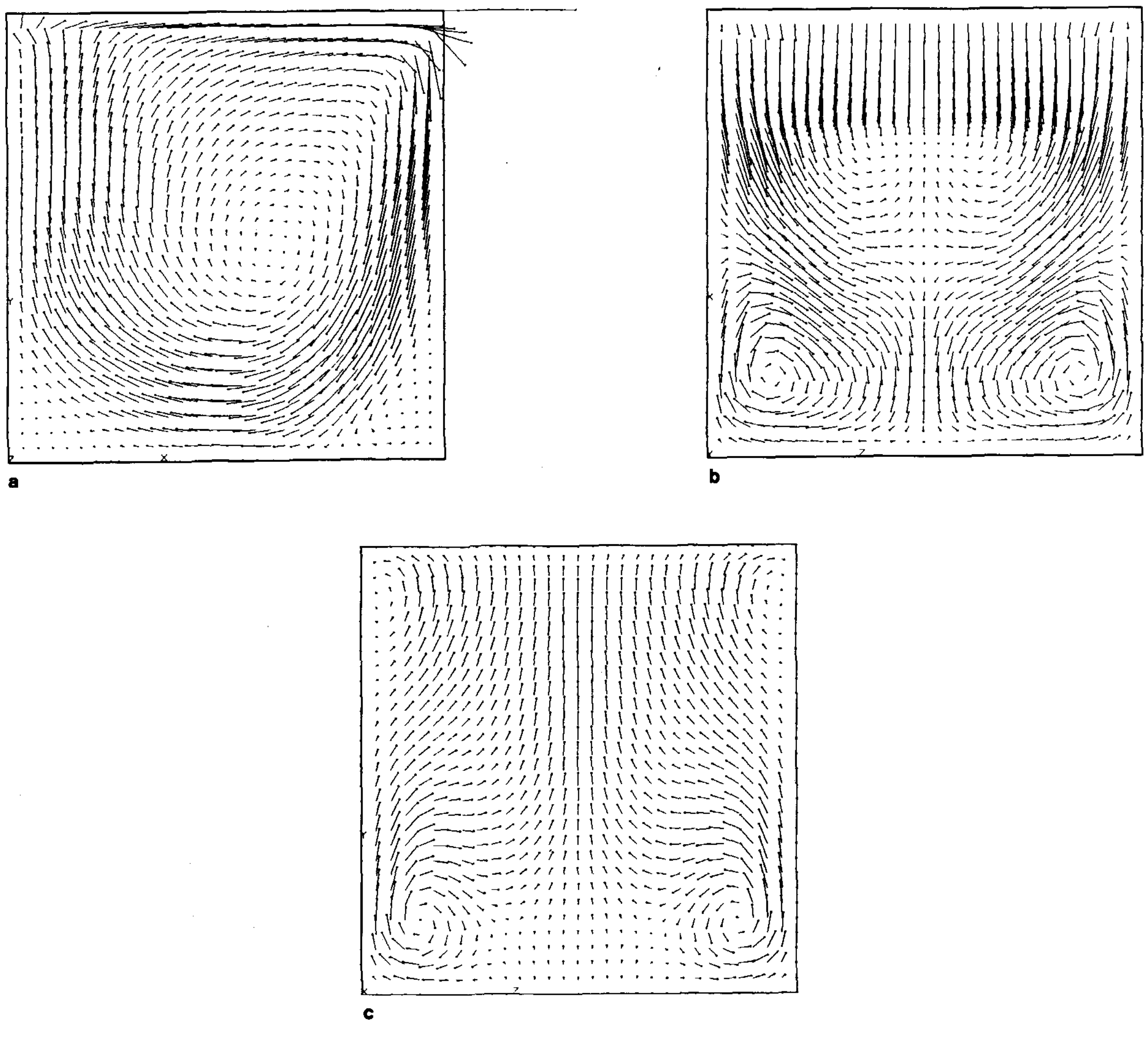


FIG. 6. Velocity vectors plots for Re = 1000 projected on: (a) x_1–x_2 midplane; (b) x_1–x_3 midplane; (c) x_2–x_3 midplane.

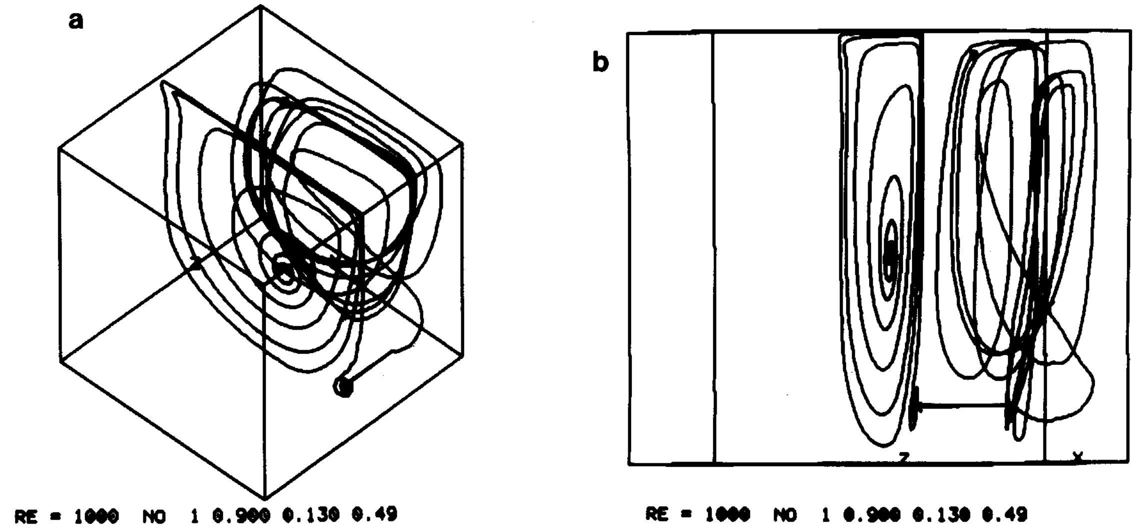


FIG. 7. Particle tracking showing that the flow in the downstream secondary eddy is towards the end wall (Re = 1000).

reversal velocities. The convergence analysis based upon the data presented in Table I indicates that the order of the method is around 1.7.

4.2. *Driven Cavity*

The results for Re = 400 are shown in Figs. 3 and 4. In Fig. 3 the velocity profiles of the u_1 component on the vertical centerline and the u_2 component on the horizontal centerline of the plane $x_3 = 0.5$ are compared with the results of [1], obtained by a pseudo-spectral method using $25 \times 25 \times 25$ modes. A good agreement is shown both in positions and values of the extremes velocity. The 2D solution [13] is also superimposed in Fig. 3 so that the side wall effect is clearly exhibited by the reduction of the extreme velocities. The velocity vector plots projected onto three orthogonal midplanes are displayed in Fig. 4. The plots in planes $x_1 - x_3$ (b) and $x_2 - x_3$ (c) clearly demonstrate that the flow is completely 3D even at this low Re. Let us note that not only the flow structure of the velocity field seems similar to that shown in Fig. 11c of [1], but also, that the position of the transversal vortex core (Fig. 4b) quantitatively evaluated from Fig. 11 of [1], agrees satisfactorily (maximum difference 5%) with that computed in the present work (Table II).

Results for Re = 1000 are presented in Figs. 5–9. The velocity profiles obtained by $101 \times 101 \times 41$ mesh (symmetry assumed) shown in Fig. 5 are compared with the 3D

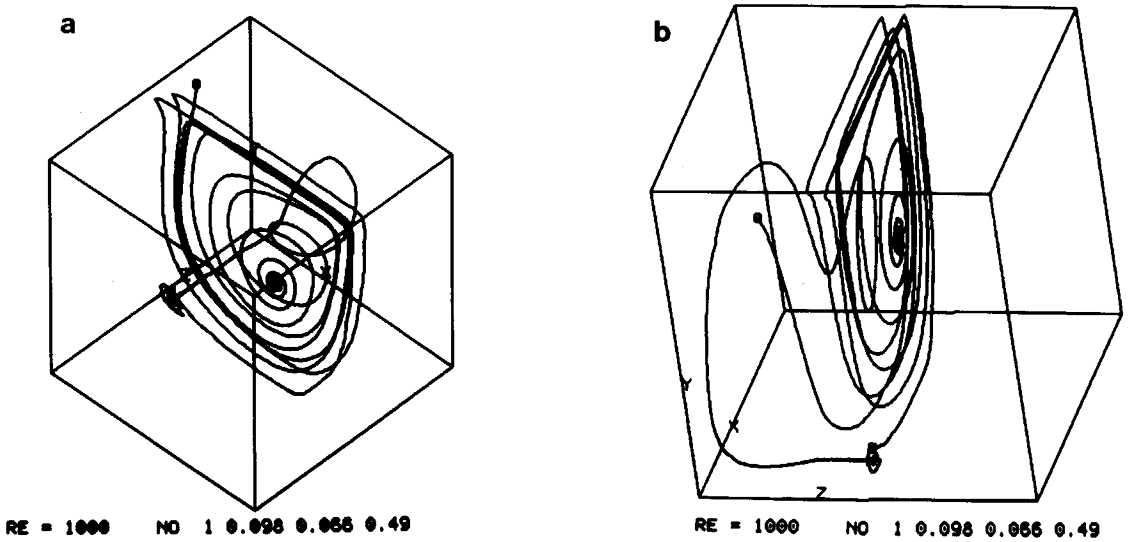


FIG. 8. Particle tracking showing that the flow in the upstream secondary eddy is towards the end wall (Re = 1000).

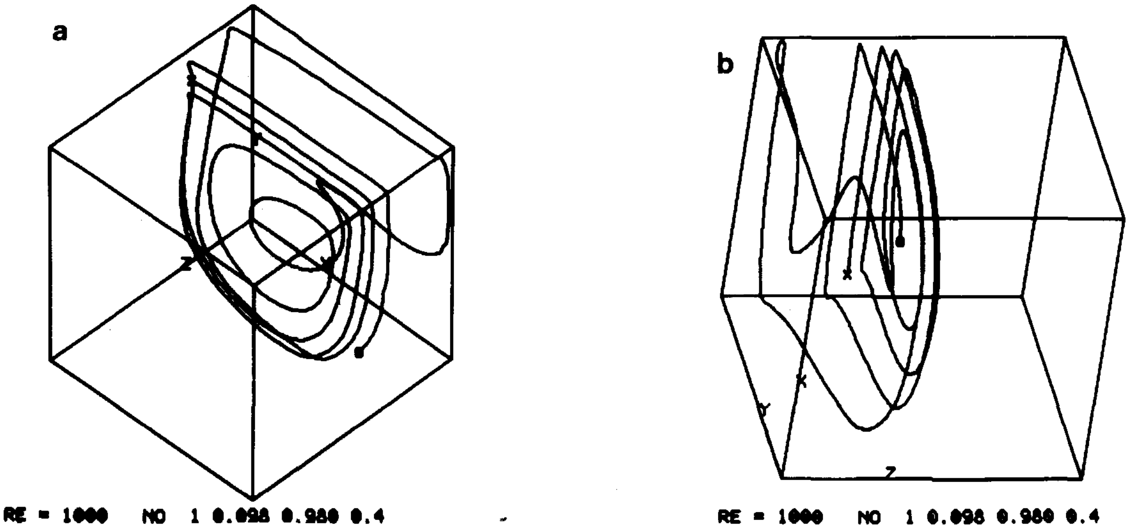


FIG. 9. Particle tracking showing that the spiralling out is toward the end wall and the spiralling in is toward the middle plane (Re = 1000).

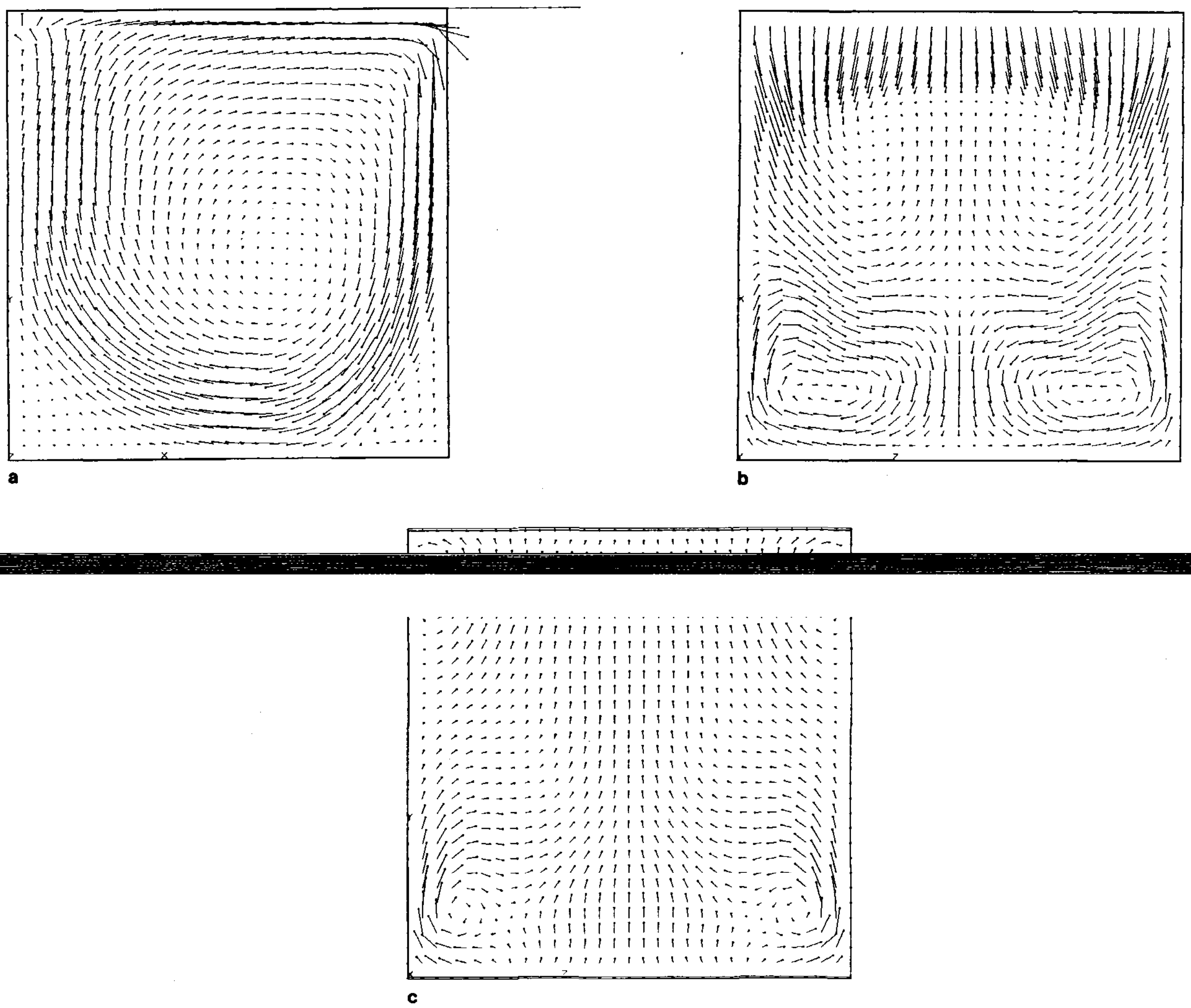


FIG. 10. Velocity vectors plots for Re = 2000 projected on: (a) x_1–x_2 midplane; (b) x_1–x_3 midplane; (c) x_2–x_3 midplane.

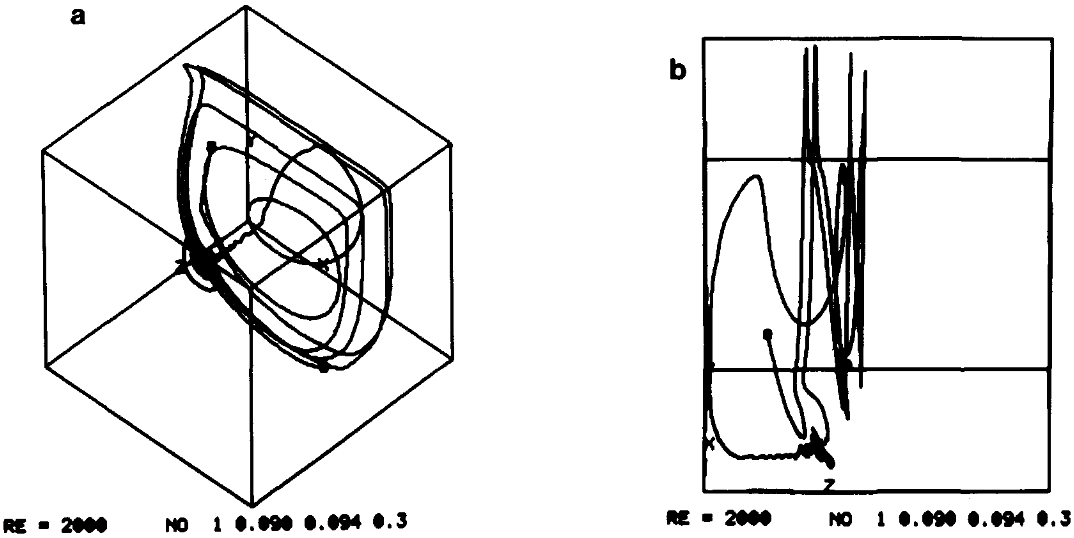


FIG. 11. Particle tracking showing the inclination of the trajectory in the upstream secondary eddy due to the presence of cross rolls (Re = 2000).

solution obtained by [1] with 30 × 30 × 30 modes. The 2D solution obtained with 67 × 67 mesh is also superimposed in Fig. 5, and the side wall effect is exhibited in a manner similar to that of Fig. 3. The velocity vector plots of Fig. 6b show the onset on the transversal plane of a counterrotating vortex, which grows with Re, see Fig. 10b. The two corner vortices in the longitudinal plane Figs. 7–8 counterrotating with respect to the main one give a flow from the symmetry plane to the end wall (Figs. 7 and 8). In the same direction there is the spiralling out of the main roll (Fig. 9). On the contrary, in the core region of the main vortex a spiralling in is present, giving a net flux from the end wall toward the symmetry plane (Fig. 9). The spiralling has the form discussed by Mallison and de Vahl Davis [2]. Even at this Re the qualitative and quantitative comparison with Fig. 14 of [1] gives a satisfactory agreement (maximum difference 6% in Table II).

The velocity–vector plot of Fig. 10 clearly demonstrates the formation of steady transversal vortices, which are present in both $x_1 - x_3$ and $x_2 - x_3$ planes (Figs. 10b and 10c), respectively. As it is clear from the trackings of Fig. 11b the strength of these transversal rolls is small compared with that of the main one so that the effect of these cross rolls is only to give an inclination of the trajectories. This inclination is evident in the path of particles in the downstream secondary eddy in Fig. 11b. The overall structure for Re = 2000 (Figs. 11 and 12) does not change too much from the one discussed for Re = 1000. At Re = 2000 a very small recirculation appears upstream at the top of the cavity (see Fig. 12). The corner vortex found experimentally by [19], for a cavity with a spanwise aspect ratio (SAR) 3:1 is not present in the obtained numerical results for SAR of 1:1.

Let us observe that the transversal rolls at the bottom of Fig. 10b tend to stretch at increasing Re. A preliminary study conducted at Re just beyond the critical value for

TABLE II

Positions of the Center of the Transversal Vortices

Re	Ku *et al.* [1]		Present	
	x_2	x_3	x_2	x_3
400	0.216	0.201	0.227	0.197
1000	0.146	0.135	0.155	0.134
2000	—	—	0.155	0.120

Note. The mesh is 31 × 31 × 31.

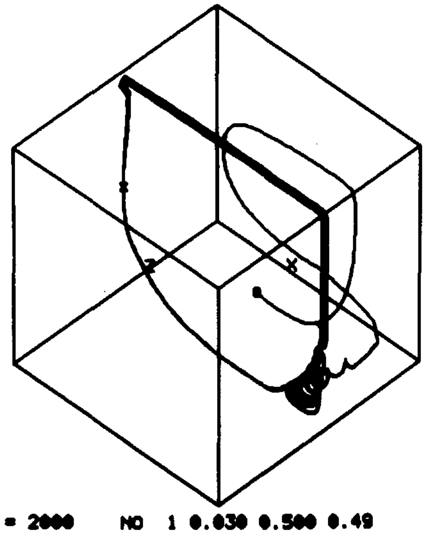


FIG. 12. Particle tracking showing the downstream secondary eddy (Re = 2000).

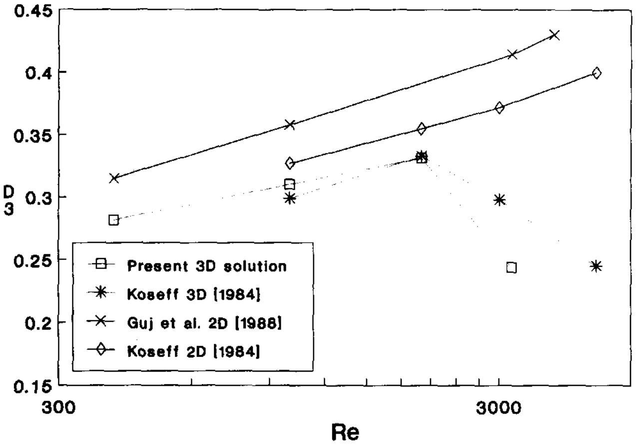


FIG. 13. Dimension of the downstream secondary eddy D_3 as a function of Re.

transition to the unsteady periodic solutions shows that each of these rolls seems to split in two, leading to a flow structure similar to that found at Re = 3200 numerically by [8] in a cube and experimentally by [19] in a box with a SAR of 3:1. At Re = 3200 [8] found that the solution falls in the unsteady region of the parameter space, but we find, using 31 × 31 × 31 mesh points, a steady solution at Re = 3200.

Finally, in Fig. 13 the dimension of downstream secondary eddy D_3 as a function of Re is shown. The present numerical results are compared with the experiments by [19], finding a very good agreement, in particular for $\text{Re} \leqslant 2000$. For Re larger than 2000 the behaviour is similar to that discussed by [19], but, due to the coarse mesh, the numerical values are sensibly different. In Fig. 13 the numerical 2D results [13] and quasi-2D experimental measurements [19] are also reported, showing that D_3 increase with Re in contrast with the 3D case.

4.3. *Parallel Speedup*

The results of parallelization of the program in terms of effectiveness is given in Table III for a typical 3D mesh of 101 × 101 × 101. The most important parameter to analyze is the parallel speed-up ratio (ρ), defined as the ratio of one processor elapsed time to the parallel elapsed time (E_p) using N processors:

$$\rho = \frac{E_p(N=1)}{E_p}. \tag{22}$$

TABLE III

Total Elapsed Time [s] and Speedup Ratio as a Function of the Number of Processors

N	E_p	ρ
1	119.58	1.00
2	63.29	1.89
3	45.63	2.62
4	37.00	3.23
5	32.59	3.67
6	30.14	3.97

Note. The mesh is 101 × 101 × 101.

TABLE IV

Degree of Parallelism (α_p), Unbalanced Load (u_l), and Total Amount ($\alpha_p + u_l$) as a Function of the Number of Processors

N	$\alpha_p \times 100$	$u_l \times 100$	$(\alpha_p + u_l) \times 100$
2	94.2	1.3	95.5
3	92.8	2.5	95.3
4	92.1	3.7	95.8
5	90.9	5.0	95.9
6	89.8	6.0	95.8

A speedup value of 3.97 (see Table III) is obtained by running the code with all six processors available. Although far from the ideal speedup of six, this is a nice result for a fully vectorized code using a widely applicable methodology. The main reason of the limited value of the speedup is due to the unbalancing load among the different processors. In the present case, using a vector of 10,000 elements and a machine with a section size of 128, we have 10,000/128 = 78 + 16/128, that is, 78 complete vector tasks and one incomplete. The 78 tasks are dispatched on the six processors and we have 13 complete tasks for the whole machine. The last incomplete task has to be solved on one processor with the other five waiting. A quantitative evaluation of this side effect is reported in Table IV. The effect of the unbalancing of processors load is particularly evident when working on a dedicated machine, but not on a multi-user environment, because in that situation the real load of each processor is not predictable.

5. CONCLUSIONS

The test case considered and the results obtained have highlighted several attractive features of the proposed vorticity–velocity method for the solution of the Navier–Stokes equations.

- 1. The results are very accurate as is shown by the comparisons with available 3D numerical data for the cubic driven cavity flow.
- 2. A mesh sensitivity analysis has been conducted for DOF form 92,256 to 2,509,446 at Re = 1000 showing a convergence speed to the asymptotic solution of the order of 1.7.

- 3. The solution algorithm, based on a false transient method is particularly efficient and stable from the numerical point of view.
- 4. The staggered mesh, the conservative formulation, and the particular discrete scheme adopted ensure that both vorticity and velocity fields are solenoidal everywhere in the problem domain.
- 5. A parallel implementation of the method has been performed on a shared memory architecture mainframe. A speed up of 3.97 is obtained on a six-processor IBM3090VF/600E.
- 6. Velocity profiles, particle tracks, and description of the overall flow structure in the cubic driven cavity are given for Re from 400 to 2000, at which the solution is found to be steady.

ACKNOWLEDGMENTS

The authors acknowledge CNR for the support received for this research with Grant No. 89.00050.69.

REFERENCES

1. H. C. Ku, R. S. Hirsh, and T. Taylor, *J. Comput. Phys.* **70**, 439 (1987).
2. G. Mallison and G. de Vahl Davis, *J. Fluid Mech.* **83**, 1 (1977).
3. S. C. R. Dennis, D. B. Ingham, and R. N. Cook, *J. Comput. Phys.* **33**, 325 (1979).
4. G. A. Osswald, K. N. Ghia, and U. Ghia, *in* AIAA Paper, 87-1139, p. 408, 1987.
5. P. M. Gresho, *Annu. Rev. Fluid Mech.* **23**, 413 (1991).
6. L. Quartapelle and M. Napolitano, *J. Comput. Phys.* **62**, 340 (1986).
7. C. G. Speziale, *J. Comput. Phys.* **73**, 476 (1987).
8. C. Y. Perng and R. L. Street, *Int. J. Numer. Methods Fluids* **9**, 341 (1989).
9. D. Gosman, Computational Dynamics Limited, private communication, 1989.
10. L. Morino, in *Developments in Boundary Element Methods*: *Nonlinear Problems in Fluid Dynamics*, edited by P. K. Banerjee and L. Morino (Elsevier Appl. Sci, Balking, 1990), Vol. 6.
11. F. Stella and G. Guj, *Int. J. Numer. Methods Fluids* **9**, 1285 (1989).
12. J. A. Reizes, E. Leonardi, and G. de Vahl Davis, in *Computational Technique and Applications*, edited by J. Noye and C. Fletcher (North Holland, Amsterdam, 1984), p. 903.
13. G. Guj, F. Stella, *Int. J. Numer. Methods Fluids* **8**, 405 (1988).
14. F. Stella, *Problema di Rayleigh–Bénard in Domini Limitati*, Ph.D. thesis, Dottorato di Ricerca in Meccanica Applicata, Roma, 1989.
15. F. Stella, G. Guj, and E. Leonardi, "The Rayleigh–Bénard problem in intermediate limited domains," Proceedings, IV Convegno Italiano di Meccanica Computazionale, Padova, June 1989 (unpublished).
16. F. Stella and G. Guj, in *Proceedings*, *Int. Conf. Parallel Computing Achievements*, *Problems and Prospects*, *Capri*, *Italy*, *1990.*
17. F. Stella, G. Guj, E. Leonardi, and G. de Vahl Davis, "The Velocity–Vorticity and the Vector Potential–Vorticity Formulations in Three-dimensional Natural Convection," Proceedings, III Convegno Italiano di Meccanica Computazionale, Palermo, June 1988 (unpublished).
18. A. Samarskii and V. Andreyev, *URSS Comput. Maths. and Math. Phys.* **3**, No. 6, (1963), 1373.
19. J. R. Koseff and R. L. Street, *ASME J. Fluids Eng.* **106**, 390 (1984).
20. S. M. Richardson and A. R. H. Cornish, *J. Fluid Mech.* **82**, 309 (1977).
21. F. Stella and G. Guj, "Vorticity–Velocity Formulation in the Computation of Flows in Multiconnected Domains," Proceedings, IX AIDAA Congress, October 1987, p. 200 (unpublished).
22. L. Quartapelle and F. Valz-Gris, *Int. J. Numer. Methods Fluids* **1**, 129 (1981).
23. G. Mallison and G. de Vahl Davis, *J. Comput. Phys.* **12** (4), 435 (1973).
24. M. Behnia, G. de Vahl Davis, F. Stella and G. Guj, in "Proceedings, of Numerical Simulation of Oscillatory Convection in Low *Pr* Fluids, in *GAMM-Workshop*, *Marseille*, *France*, *1988*, edited by B. Roux (Vieweg, Braunschweig, 1990), p. 19.