Computational Mathematics and Mathematical Physics, Vol. 43, No. 3, 2003, pp. 433–445.
Translated from Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki, Vol. 43, No. 3, 2003, pp. 453–466.

Numerical Simulation of Viscous Incompressible Flow in a Cubic Cavity

T. G. Elizarova and O. Yu. Milyukova
Institute of Mathematical Modeling, Russian Academy of Sciences,
Miusskaya pl. 4a, Moscow, 125047 Russia
e-mail: telizar@yahoo.com

Received February 22, 2002; in final form, May 30, 2002

Abstract—The results of numerical simulations of the three-dimensional isothermal viscous incompressible flow in a cavity with a moving lid are presented. The system of quasi-hydrodynamic equations is used as a mathematical model. Computations were performed on a distributed-memory multiprocessor parallel computer. Parallel computation of Poisson's equation is discussed in detail. Numerical results obtained for the Reynolds numbers 100 and 1000 on progressively refined grids are presented.

INTRODUCTION

Numerical simulation of three-dimensional unsteady flows is a promising area of modern computational fluid dynamics. Multidimensional simulations of viscous incompressible flows became possible due to progress in numerical methods for computing such flows in terms of the so-called natural (velocity–pressure) variables and to the high performance of multiprocessor computers [1].

The problem of fluid motion in a cavity is a well known and demanding benchmark test used to validate numerical techniques for flow simulation and evaluate their efficiency. However, publications containing the results of multidimensional numerical computations of such flows are scarce [2–5]. At moderate Reynolds numbers (Re < 1000), the flow is a steady laminar vortex with center near the center of the domain. The flow in the symmetry plane of the cavity is virtually two-dimensional and is adequately described by the available two-dimensional models. The corresponding velocity distributions obtained by applying different techniques are mutually consistent and agree with the results of physical experiments [6]. With increasing Reynolds number, the flow structure becomes increasingly complicated: the flow becomes stratified at Re ~ 2000, unsteady at higher velocities, and ultimately turbulent.

In this study, the cavity flow is simulated by using the system of quasi-hydrodynamic (QHD) equations describing viscous incompressible flows [7, 8]. In [8], phenomenological derivation and analysis of the QHD equations were presented and a number of exact analytical solutions to the system were obtained. The QHD equations were validated by computing two-dimensional viscous incompressible flows in [9–11]. The QHD equations describing viscous incompressible flows have several advantages over the conventional Navier–Stokes equations, which provide a basis for developing efficient numerical algorithms for solving fluid-dynamics equations. Specifically, the system of QHD equations contains additional dissipative terms ensuring regularization of numerical solutions. The QHD system is supplemented with natural boundary conditions for pressure required to solve Poisson's equation. When the QHD equations are approximated on a spatial grid, the values of velocity and pressure in the finite-difference equations are determined at the same grid points. This allows one to avoid the use of the so-called staggered grids [12].

Simulation of a multidimensional flow is a formidable task that requires the use of a high-performance computer to be accomplished. The numerical algorithm described here was implemented on a cluster distributed-memory computer with the use of data exchange based on the MPI (Message Passing Interface) standard. This interface is currently used in all major computing facilities containing several tens to several thousands of processors. The problem was solved in the velocity–pressure variables. The efficiency of the numerical algorithm is largely determined by the efficiency of computation of Poisson's equation for pressure. For this reason, special emphasis was placed on the implementation of the numerical method for solving Poisson's equation in an (x, y, z) geometry on a multiprocessor computer. A parallel version of a modified incomplete Cholesky conjugate gradient method (MICCG(0)) (see [13]) was proposed for solving the second boundary value problem.

The algorithm developed in this study is designed to solve both time-independent and time-dependent problems on large spatial grids. The results of computations of a cavity flow obtained on grids with up to

4×10^6 points in space for Re = 100 and 1000 are presented. For Re = 1000, the convergence of solutions computed on progressively refined grids is demonstrated.

1. QUASI-HYDRODYNAMIC EQUATIONS

In index notation, the QHD equations of isothermal viscous incompressible flow are written as follows:

$$\nabla_i u^i = \nabla_i w^i, \tag{1.1}$$

$$\frac{\partial u^k}{\partial t} + \nabla_i(u^i u^k) + \frac{1}{\rho}\nabla^k p = \nabla_i \Pi^{ik}_{NS} + \nabla_i(u^i w^k) + \nabla_i(w^i u^k). \tag{1.2}$$

Here, summation is performed over repeated indices. The Navier–Stokes viscous stress tensor is expressed as $\Pi^{ik}_{NS} = \nu(\nabla^k u^i + \nabla^i u^k)$. The components of mass flux density are calculated by the formulas

$$j^k = \rho(u^k - w^k), \quad k = 1, 2, 3, \tag{1.3}$$

where

$$w^k = \tau\left(u^j \nabla_j u^k + \frac{1}{\rho}\nabla^k p\right). \tag{1.4}$$

The following notation is used in (1.1)–(1.4): $\rho = \text{const} > 0$ is the fluid density, p is the dynamic pressure, $\mathbf{u} = (u^1, u^2, u^3)$ is velocity, $\nu = \eta/\rho$ is kinematic viscosity, $\tau = \nu/c_s^2$ is a relaxation time, and c_s is sonic velocity.

The QHD equations differ from the Navier–Stokes equations by additional conservative terms with the parameter τ having the dimension of time. As $\tau \longrightarrow 0$, the QHD equations reduce to the Navier–Stokes equations. In the time-independent equations, the additional conservative terms are of order $O(\tau^2)$ as $\tau \longrightarrow 0$. The laminar boundary-layer approximation corresponding to the QHD equations is equivalent to the classical Prandtl equations [7, 8]. In [8], a number of exact analytical solutions to the QHD equations were obtained. As $\tau \longrightarrow 0$, these exact solutions tend to the corresponding exact solutions to the Navier–Stokes equations. In this sense, the system of QHD equations approximates the Navier–Stokes equations.

For incompressible flows, the convergence of numerical solutions to the QHD equations to the corresponding solutions to the Navier–Stokes equations with decreasing τ was demonstrated in [9–11] by using two-dimensional convective flows in rectangular cavities as examples.

In dimensionless form, the QHD system of equations for three-dimensional isothermal flows is written in Cartesian coordinates as

$$\frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} + \frac{\partial y_z}{\partial z} = \frac{\partial w_x}{\partial x} + \frac{\partial w_y}{\partial y} + \frac{\partial w_z}{\partial z}, \tag{1.5}$$

$$\begin{aligned}\frac{\partial u_x}{\partial t} + \frac{\partial(u_x^2)}{\partial x} + \frac{\partial(u_x u_y)}{\partial y} + \frac{\partial(u_x u_z)}{\partial z} + \frac{\partial p}{\partial x} &= \frac{2}{\mathrm{Re}}\frac{\partial^2 u_x}{\partial x^2} + \frac{1}{\mathrm{Re}}\frac{\partial}{\partial y}\left(\frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x}\right) + \frac{1}{\mathrm{Re}}\frac{\partial}{\partial z}\left(\frac{\partial u_x}{\partial z} + \frac{\partial u_z}{\partial y}\right)\\ &+ 2\frac{\partial(u_x w_x)}{\partial x} + \frac{\partial(u_y w_x)}{\partial y} + \frac{\partial(u_x w_y)}{\partial y} + \frac{\partial(u_z w_x)}{\partial z} + \frac{\partial(u_x w_z)}{\partial z},\end{aligned} \tag{1.6}$$

$$\begin{aligned}\frac{\partial u_y}{\partial t} + \frac{\partial(u_x u_y)}{\partial x} + \frac{\partial(u_y^2)}{\partial y} + \frac{\partial(u_z u_y)}{\partial z} + \frac{\partial p}{\partial y} &= \frac{1}{\mathrm{Re}}\frac{\partial}{\partial x}\left(\frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x}\right) + \frac{2}{\mathrm{Re}}\frac{\partial^2 u_y}{\partial y^2} + \frac{1}{\mathrm{Re}}\frac{\partial}{\partial z}\left(\frac{\partial u_y}{\partial z} + \frac{\partial u_z}{\partial y}\right)\\ &+ \frac{\partial(u_x w_y)}{\partial x} + \frac{\partial(u_y w_x)}{\partial x} + 2\frac{\partial(u_y w_y)}{\partial y} + \frac{\partial(u_z w_y)}{\partial z} + \frac{\partial(u_y w_z)}{\partial z},\end{aligned} \tag{1.7}$$

$$\begin{aligned}\frac{\partial u_z}{\partial t} + \frac{\partial(u_x u_z)}{\partial x} + \frac{\partial(u_y u_z)}{\partial y} + \frac{\partial(u_z^2)}{\partial z} + \frac{\partial p}{\partial z} &= \frac{1}{\mathrm{Re}}\frac{\partial}{\partial x}\left(\frac{\partial u_x}{\partial z} + \frac{\partial u_z}{\partial x}\right) + \frac{1}{\mathrm{Re}}\frac{\partial}{\partial y}\left(\frac{\partial u_y}{\partial z} + \frac{\partial u_z}{\partial y}\right) + \frac{2}{\mathrm{Re}}\frac{\partial^2 u_z}{\partial z^2}\\ &+ \frac{\partial(u_x w_z)}{\partial x} + \frac{\partial(u_z w_x)}{\partial x} + \frac{\partial(u_y w_z)}{\partial y} + \frac{\partial(u_z w_y)}{\partial y} + 2\frac{\partial(u_z w_z)}{\partial z},\end{aligned} \tag{1.8}$$

where

$$\begin{aligned} w_x &= \tau\left(u_x\frac{\partial u_x}{\partial x} + u_y\frac{\partial u_x}{\partial y} + u_z\frac{\partial u_x}{\partial z} + \frac{\partial p}{\partial x}\right), \\ w_y &= \tau\left(u_x\frac{\partial u_y}{\partial x} + u_y\frac{\partial u_y}{\partial y} + u_z\frac{\partial u_y}{\partial z} + \frac{\partial p}{\partial y}\right), \\ w_z &= \tau\left(u_x\frac{\partial u_z}{\partial x} + u_y\frac{\partial u_z}{\partial y} + u_z\frac{\partial u_z}{\partial z} + \frac{\partial p}{\partial z}\right). \end{aligned} \tag{1.9}$$

Here, the unknown quantities are the velocity components $u_x = u_x(x, y, z, t)$, $u_y = u_y(x, y, z, t)$, and $u_z = u_z(x, y, z, t)$ and pressure $p = p(x, y, t)$.

The pressure field corresponding to a known velocity field is computed by solving Poisson's equation

$$\begin{aligned} \frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} + \frac{\partial^2 p}{\partial z^2} &= \frac{1}{\tau}\left(\frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} + \frac{\partial u_z}{\partial z}\right) - \frac{\partial}{\partial x}\left(u_x\frac{\partial u_x}{\partial x} + u_y\frac{\partial u_x}{\partial y} + u_z\frac{\partial u_x}{\partial z}\right) \\ &- \frac{\partial}{\partial y}\left(u_x\frac{\partial u_y}{\partial x} + u_y\frac{\partial u_y}{\partial y} + u_z\frac{\partial u_y}{\partial z}\right) - \frac{\partial}{\partial z}\left(u_x\frac{\partial u_z}{\partial x} + u_y\frac{\partial u_z}{\partial y} + u_z\frac{\partial u_z}{\partial z}\right), \end{aligned} \tag{1.10}$$

which is equivalent to (1.5) when $\tau = \text{const}$.

2. STATEMENT OF THE PROBLEM

Consider a three-dimensional isothermal flow in a cubic cavity with edge length H. The lid on the top of the cavity is moving at a constant velocity U_0. The computational domain and the coordinate system employed are schematized in Fig. 1.

The flow is described by Eqs. (1.6)–(1.10) rewritten in dimensionless form by using the relations

$$x = \tilde{x}H, \quad y = \tilde{y}H, \quad z = \tilde{z}H, \quad u_x = \tilde{u}_xU_0, \quad u_y = \tilde{u}_yU_0, \quad u_z = \tilde{u}_zU_0,$$

$$p = \tilde{p}\rho U_0^2, \quad t = (\tilde{t}H)/U_0, \quad \text{Re} = (U_0H)/\nu.$$

Equations (1.6)–(1.10) are supplemented with boundary conditions. On stationary rigid surfaces, velocity is subject to the no-slip condition

$$\mathbf{u} = 0.$$

On the surface $y = 1$, the conditions $u_x = U_0$, $u_y = 0$, and $u_z = 0$ are set. The boundary conditions for pressure follow from the impermeability condition and have the form

$$\partial p/\partial \mathbf{n} = 0, \tag{2.1}$$

where $\mathbf{n}$ is the normal vector to a surface. In particular, $\partial p/\partial x = 0$ on the face $x = 0$, whereas condition (2.1) is equivalent to $\partial p/\partial x = 0$ and $\partial p/\partial y = 0$ on the edge $x = 0$, $y = 0$. At the vertex $x = 0$, $y = 0$, and $z = 0$, it holds that $\partial p/\partial x = 0$, $\partial p/\partial y = 0$, and $\partial p/\partial z = 0$.

As the initial condition, we set $u_x = u_y = u_z = 0$. The initial pressure was assumed constant in the entire flow: $p = 0$. To eliminate ambiguity in computing pressure, its value at the vertex (1, 1, 1) was held constant: $p(1, 1, 1) = 1$.

The additional terms containing τ were treated as regularizers. In the computations, the value of τ was varied and was chosen to ensure the accuracy and stability of the algorithm. In this study, we used $\tau = 0.01$ for Re = 100 and $\tau = 0.001$ for Re = 1000. Two-dimensional computations of similar problems have shown that this choice ensures sufficient stability and accuracy of the numerical algorithm [9, 10].

3. NUMERICAL ALGORITHM

Numerical solution of Eqs. (1.6)–(1.10) was based on a finite-difference method. A uniform grid was used, with mesh size h along all coordinate axes. All quantities were calculated at grid points. The boundary of the computational domain was set at half-integer grid points; i.e., the mesh size was equal to $h/2$ at the

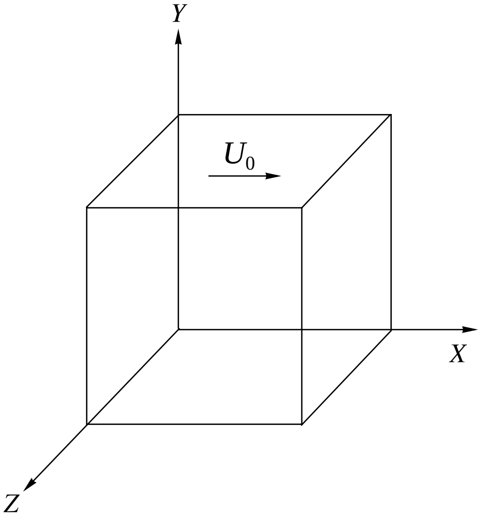

Fig. 1.

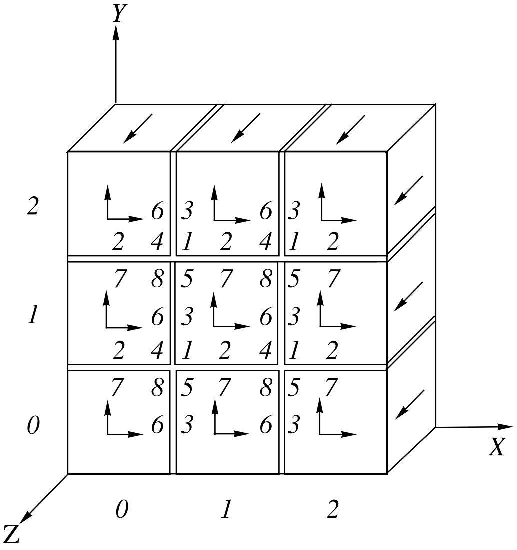

Fig. 2.

boundary. The first derivatives were calculated as

$$\frac{\partial f}{\partial x} \approx \frac{1}{h}\left(\frac{f_{i+1,j,k}+f_{ijk}}{2} - \frac{f_{ijk}+f_{i-1,j,k}}{2}\right).$$

The second derivatives were approximated by the expressions

$$\frac{\partial}{\partial x}k\frac{\partial f}{\partial x} = \frac{1}{h}\left(k_{i+0.5,j,k}\frac{f_{i+1,j,k}-f_{ijk}}{h} - k_{i-0.5,j,k}\frac{f_{ijk}-f_{i-1,j,k}}{h}\right),$$

$$\frac{\partial}{\partial x}k\frac{\partial f}{\partial y} = \frac{1}{h}\left(k_{i+0.5,j,k}\frac{f_{i+0.5,j+0.5,k}-f_{i+0.5,j-0.5,k}}{h} - k_{i-0.5,j,k}\frac{f_{i-0.5,j+0.5,k}-f_{i-0.5,j-0.5,k}}{h}\right),$$

where $k_{i\pm0.5,j,k} = 0.5(k_{i\pm1,j,k} + k_{ijk})$, $f_{i\pm0.5,j\pm0.5,k} = 0.25(f_{i\pm1,j\pm1,k} + f_{ijk} + f_{i\pm1,j,k} + f_{i,j\pm1,k})$. Thus, the values of functions at half-integer grid points were determined as the half-sums of their values at integer grid points, while their values at the centers of cell faces, which are required to calculate mixed derivatives, were approximated by the arithmetic means of their values at the adjacent grid points. The derivatives $\partial f/\partial y$, $\partial f/\partial z$, $(\partial/\partial y)(k\partial f/\partial y)$, $(\partial/\partial z)(k\partial f/\partial z)$, $(\partial/\partial x)(k\partial f/\partial z)$, $(\partial/\partial y)(k\partial f/\partial x)$, $(\partial/\partial y)(k\partial f/\partial z)$, $(\partial/\partial z)(k\partial f/\partial x)$, and $(\partial/\partial z)(k\partial f/\partial y)$ were approximated by analogous expressions.

The boundary conditions for velocity were approximated with second-order accuracy by introducing fictitious grid points along the outer boundaries of the domain. Approximate boundary conditions for pressure were obtained by extrapolating Poisson's equation to the domain boundary. In particular, the condition $\partial p/\partial x = 0$ set on the face $x = 0$ was approximated by the equation

$$-0.5p_{1,j-1,k} - 0.5p_{1,j,k-1} + 3p_{1jk} - p_{2jk} - 0.5p_{1,j+1,k} - 0.5p_{1,j,k+1} = 0;$$

the conditions $\partial p/\partial x = 0$ and $\partial p/\partial y = 0$ set on the edge with $x = 0$ and $y = 0$, by the equation

$$-0.25p_{1,1,k-1} + 1.5p_{11k} - 0.5p_{21k} - 0.5p_{12k} - 0.25p_{1,1,k+1} = 0.$$

At the vertex $x = 0$, $y = 0$, $z = 0$,

$$0.75p_{111} - 0.25p_{211} - 0.25p_{121} - 0.25p_{112} = 0.$$

Analogous equations are written for other grid points located on the domain boundary. At the point with coordinates $x = 1$, $y = 1$, and $z = 1$, we set $p_{N_1,N_2,N_3} = 1$, where N_1, N_2, and N_3 denote the number of grid points along the coordinates x, y, and z, respectively.

To solve the three-dimensional finite-difference equation for pressure $Ay = f$, we used a modified incomplete Cholesky conjugate gradient method (MICCG(0)) (see [13]) or its parallel version proposed in Section 4 below. The iterative process was terminated as soon as the condition $(Ay^l - f, Ay^l - f) < \varepsilon^2$ was satisfied, where $\varepsilon = 0.00001$ and l is the number of iteration step.

The velocity field on the next time layer was determined by using an explicit scheme with time step Δt. The flow was assumed to be steady if

$$\max_{ijk}\left(\frac{u_x^{n+1} - u_x^n}{\Delta t}\right)_{ijk} < 0.001, \quad \max_{ijk}\left(\frac{u_y^{n+1} - u_y^n}{\Delta t}\right)_{ijk} < 0.001, \quad \max_{ijk}\left(\frac{u_z^{n+1} - u_z^n}{\Delta t}\right)_{ijk} < 0.001.$$

This algorithm is a generalization of that proposed in [9–11] for solving the two-dimensional problem.

4. ALGORITHM OF PARALLEL IMPLEMENTATION

The parallel implementation of the solution of the finite-difference equations approximating system (1.6)–(1.10) is based on the approach known as domain decomposition (geometric parallelism). The three-dimensional computational domain is partitioned into $p = p_1 \times p_2$ subdomains in the directions OX and OY as illustrated by Fig. 2 for $p_1 = p_2 = 3$, and a two-dimensional array of $p = p_1 \times p_2$ processors is organized into a computer. Each processor executes computations in the corresponding geometric domain.

The algorithm of parallel implementation of the solution of the explicit finite-difference scheme for system (1.6)–(1.8) is analogous to that described in [14]. Prior to computing the velocity components $(u_x)_{ijk}^{n+1}$, $(u_y)_{ijk}^{n+1}$, and $(u_z)_{ijk}^{n+1}$ on the $(n + 1)$th time layer, the values of $(u_x)_{ijk}^n$, $(u_y)_{ijk}^n$, $(u_z)_{ijk}^n$, and p_{ijk}^n at the boundary grid points in the adjoining subdomains lying in planes *2* and *3* and on lines *1* are transferred. They are required to calculate grid functions at the boundary points of subdomains along lines *4* and *5* and in planes *6* and *7*. Data are transferred in packets.

The finite-difference counterpart of (1.10),

$$-a_{ijk}y_{i-1,j,k} - b_{ijk}y_{i,j-1,k} - g_{ijk}y_{i,j,k-1} + c_{ijk}y_{ijk} - a_{i+1,j,k}y_{i+1,j,k} - b_{i,j+1,k}y_{i,j+1,k} - g_{i,j,k+1}y_{i,j,k+1} = f_{ijk}$$

$(i = 1, 2, \ldots, N_1 - 1, j = 1, 2, \ldots, N_2 - 1, k = 1, 2, \ldots, N_3 - 1)$, or the equation $Ay = f$ was solved on a single-processor computer by means of MICCG(0). In this method, the number of iteration steps required to ensure convergence is $O(\ln(2/\varepsilon)\sqrt{N_h})$, where N_h is the number of grid points along a coordinate axis and ε is the admissible relative error. The major difficulty in parallelizing MICCG(0) lies in the recursive procedure used to calculate the inverse of the preconditioner matrix. To overcome this difficulty, the grid points are reordered and the preconditioner matrix is reconstructed [15–20].

In the parallel version of MICCG(0), the preconditioner matrix has the form

$$B = (D^{-1} + A_1)D[D^{-1} + (A_1)^{\mathrm{T}}],$$

where the matrix A_1 defines an operator $\hat{A}_1$ in the space of grid functions on the set of grid points:

$$(\hat{A}_1 y)_{ijk} = \begin{cases} -a_{ijk}y_{i-1,j,k} - b_{ijk}y_{i,j-1,k} - g_{ijk}y_{i,j,k-1}, & (i,j,k) \in \omega_0, \\ -g_{ijk}y_{i,j,k-1}, & (i,j,k) \in \omega_1, \\ -a_{ijk}y_{i-1,j,k} - g_{ijk}y_{i,j,k-1}, & (i,j,k) \in \omega_2, \\ -b_{ijk}y_{i,j-1,k} - g_{ijk}y_{i,j,k-1}, & (i,j,k) \in \omega_3, \\ -a_{ijk}y_{i-1,j,k} - a_{i+1,j,k}y_{i+1,j,k} - g_{ijk}y_{i,j,k-1}, & (i,j,k) \in \omega_4, \\ -b_{ijk}y_{i,j-1,k} - b_{i,j+1,k}y_{i,j+1,k} - g_{ijk}y_{i,j,k-1}, & (i,j,k) \in \omega_6, \\ -a_{ijk}y_{i-1,j,k} - a_{i+1,j,k}y_{i+1,j,k} - b_{ijk}y_{i,j-1,k} - g_{ijk}y_{i,j,k-1}, & (i,j,k) \in \omega_6, \\ -b_{ijk}y_{i,j-1,k} - b_{i,j+1,k}y_{i,j+1,k} - a_{ijk}y_{i-1,j,k} - g_{ijk}y_{i,j,k-1}, & (i,j,k) \in \omega_7, \\ -a_{ijk}y_{i-1,j,k} - b_{ijk}y_{i,j-1,k} - a_{i+1,j,k}y_{i+1,j,k} - b_{i,j+1,k}y_{i,j+1,k} - g_{ijk}y_{i,j,k-1}, & (i,j,k) \in \omega_8. \end{cases}$$

Here, ω_0 is the set of interior grid points of all subdomains; ω_2, ω_3, ω_6, and ω_7 are the sets of grid points located in planes *2, 3, 6,* and *7*, respectively; and ω_1, ω_4, ω_5, and ω_8 are the sets of grid points located on lines *1, 4, 5,* and *8*, respectively (see Fig. 2).

The entries d_{ijk} of the diagonal matrix D are determined by the condition $A\mathbf{e} + GD_A\mathbf{e} = B\mathbf{e}$, where $\mathbf{e} = (1, \dots, 1)^{\text{т}}$; D_A is the diagonal part of A; and G is the diagonal matrix with diagonal entries σ_{ijk} defined as

$$\sigma_{ijk} = \begin{cases} \xi_\alpha h + \xi_0 h^2, \text{ if } (i, j, k) \in \omega_\alpha, \text{ for } \alpha = 1, 2, 3, 4, 5, \\ \xi_0 h^2 \text{ otherwise;} \end{cases}$$

where $\xi_\alpha > 0$ for $0 \le \alpha \le 5$. We set

$$\tilde{\xi}_1 = 2\pi/3, \quad \tilde{\xi}_2 = \tilde{\xi}_3 = \tilde{\xi}_4 = \tilde{\xi}_5 = \pi/3, \quad \tilde{\xi}_0 = 0.5\pi^2. \tag{4.1}$$

The entries d_{ijk} of the diagonal matrix D are calculated as

$$d_{ijk}^{-1} = \begin{cases} \tilde{c}_{ijk} - a_{ijk}(a_{ijk} + b_{i-1, j+1, k} + \kappa_i^1 a_{i-1, j, k} + g_{i-1, j, k+1})d_{i-1, j, k} - \\ - b_{ijk}(b_{ijk} + a_{i+1, j-1, k} + \kappa_j^2 b_{i, j-1, k} + g_{i, j-1, k+1})d_{i, j-1, k} - \\ - g_{ijk}(g_{ijk} + a_{i+1, j, k-1} + b_{i, j+1, k-1})d_{i, j, k-1}, \quad (i, j, k) \in \omega_0, \\ \tilde{c}_{ijk} - g_{ijk}(g_{ijk} + a_{i+1, j, k-1} + a_{i, j, k-1} + b_{i, j+1, k-1} + b_{i, j, k-1})d_{i, j, k-1}, \quad (i, j, k) \in \omega_1, \\ \tilde{c}_{ijk} - a_{ijk}(a_{ijk} + b_{i-1, j, k} + b_{i-1, j+1, k} + \kappa_i^1 a_{i-1, j, k} + g_{i-1, j, k+1})d_{i-1, j, k} - \\ - g_{ijk}(g_{ijk} + a_{i+1, j, k-1} + b_{i, j+1, k-1} + b_{i, j, k-1})d_{i, j, k-1}, \quad (i, j, k) \in \omega_2, \\ \tilde{c}_{ijk} - b_{ijk}(b_{ijk} + a_{i+1, j-1, k} + a_{i, j-1, k} + \kappa_j^2 b_{i, j-1, k} + g_{i, j-1, k+1})d_{i, j-1, k} - \\ - g_{ijk}(g_{ijk} + a_{i+1, j, k-1} + a_{i, j, k-1} + b_{i, j+1, k-1})d_{i, j, k-1}, \quad (i, j, k) \in \omega_3, \\ \tilde{c}_{ijk} - a_{ijk}(a_{ijk} + b_{i-1, j, k} + b_{i-1, j+1, k} + g_{i-1, j, k+1})d_{i-1, j, k} - \\ - a_{i+1, j, k}(a_{i+1, j, k} + b_{i+1, j, k} + b_{i+1, j+1, k} + a_{i+2, j, k} + g_{i+1, j, k+1})d_{i+1, j, k} - \\ - g_{ijk}(g_{ijk} + b_{i, j+1, k-1} + b_{i, j, k-1})d_{i, j, k-1}, \quad (i, j, k) \in \omega_4, \\ \tilde{c}_{ijk} - b_{ijk}(b_{ijk} + a_{i, j-1, k} + a_{i+1, j-1, k} + g_{i, j-1, k+1})d_{i, j-1, k} - \\ - b_{i, j+1, k}(b_{i, j+1, k} + a_{i+1, j+1, k} + a_{i, j+1, k} + b_{i, j+2, k} + g_{i, j+1, k+1})d_{i, j+1, k} - \\ - g_{ijk}(g_{ijk} + a_{i+1, j, k-1} + a_{i, j, k-1})d_{i, j, k-1}, \quad (i, j, k) \in \omega_5, \\ \tilde{c}_{ijk} - a_{ijk}(a_{ijk} + b_{i-1, j+1, k} + g_{i-1, j, k})d_{i-1, j, k} - b_{ijk}(b_{ijk} + \\ + \kappa_j^2 b_{i, j-1, k} + g_{i, j-1, k+1})d_{i, j-1, k} - a_{i+1, j, k}(a_{i+1, j, k} + b_{i+1, j+1, k} + \\ + a_{i+2, j, k} + g_{i+1, j, k+1})d_{i+1, j, k} - g_{ijk}(g_{ijk} + b_{i, j+1, k-1})d_{i, j, k-1}, \quad (i, j, k) \in \omega_6, \\ \tilde{c}_{ijk} - b_{ijk}(b_{ijk} + a_{i+1, j-1, k} + g_{i, j-1, k+1})d_{i, j-1, k} - a_{ijk}(a_{ijk} + \\ + \kappa_i^1 a_{i-1, j, k} + g_{i-1, j, k+1})d_{i-1, j, k} - b_{i, j+1, k}(b_{i, j+1, k} + a_{i+1, j+1, k} + \\ + b_{i, j+2, k} + g_{i, j+1, k+1})d_{i, j+1, k} - g_{ijk}(g_{ijk} + a_{i+1, j, k-1})d_{i, j, k-1}, \quad (i, j, k) \in \omega_7, \\ \tilde{c}_{ijk} - a_{ijk}(a_{ijk} + g_{i-1, j, k+1})d_{i-1, j, k} - b_{ijk}(b_{ijk} + g_{i, j-1, k+1})d_{i, j-1, k} - \\ - a_{i+1, j, k}(a_{i+1, j, k} + a_{i+2, j, k} + g_{i+1, j, k+1})d_{i+1, j, k} - b_{i, j+1, k}(b_{i, j+1, k} + \\ + b_{i, j+2, k} + g_{i, j+1, k+1})d_{i, j+1, k} - g_{ijk}^2 d_{i, j, k-1}, \quad (i, j, k) \in \omega_8, \end{cases}$$

where $\tilde{c}_{ijk} = c_{ijk}(1 + \sigma_{ijk})$,

$$\kappa_i^1 = \begin{cases} 1, \text{ if } i = M_1 k_1 + 1, \quad 1 \le k_1 \le p_1 - 1, \\ 0 \text{ otherwise,} \end{cases}$$

$$\kappa_j^2 = \begin{cases} 1, \text{ if } j = M_2 k_2 + 1, \quad 1 \le k_2 \le p_2 - 1, \\ 0 \text{ otherwise,} \end{cases}$$

Table

Re	N_h	Δt	n	t
100	22	0.002	3764	7.53
	42	0.0025	3038	7.59
1000	42	0.001	29666	29.67
	82	0.002	16329	32.66
	162	0.002	16428	32.86

$M_\alpha = N_\alpha/p_\alpha$, $k_1 = [i/M_1]$, $k_2 = [j/M_2]$, $[l]$ is the integral part of l, and (k_1, k_2) is the index of a subdomain $(0 \le k_\alpha \le p_\alpha - 1;\ \alpha = 1, 2)$.

By following [19], it can be shown that the parallel version of MICCG(0) is convergent if the number of iteration steps is at least $O(\ln(2/\varepsilon)\sqrt{N_h})$ for any particular pair of p_1 and p_2. When p_1 and p_2 are sufficiently large, $O(\ln(2/\varepsilon)\sqrt{N_h}\max(\sqrt{p_1}, \sqrt{p_2}))$ iteration steps are required. According to theoretical analyses and computations, the required number of iteration steps slowly increases with the number of processors. Note that the values of ξ_α in (4.1) minimize the estimated number of iteration steps of the parallel version of MICCG(0) required to solve the Dirichlet problem for Poisson's equation.

A parallel calculation of d_{ijk} is started by all processors simultaneously from lines *1* (see Fig. 2) and is continued at the interior points of the subdomains in planes *2* and *3*. The values of d_{ijk} calculated in planes *2* and *3* and on lines *1* are transferred to adjacent processors. Then, the calculation continues on lines *4* and *5* and in planes *6* and *7* and terminates on lines *8*. Prior to calculating d_{ijk} on lines *8*, the values of d_{ijk} on the corresponding lines *4* and *5* are transferred from the adjacent processors. The order of calculation is indicated by arrows in Fig. 2. All data are transferred in packets. Both stages of the inversion of the matrix B, $(D^{-1} + A_1)\overline{w^k} = Ay^k - f$ and $[D^{-1} + (A_1)^{\mathrm{T}}]w^k = D^{-1}w^k$, are implemented in a similar manner at each kth iteration step. The parallelization of the remaining procedures involved in the preconditioned conjugate gradient algorithm is organized by analogy with that applied in [18] to two-dimensional problems.

Note that the computation of pressure by solving Poisson's equation may take 60% to 90% of the total run time, depending on the required number of iteration steps. For this reason, the efficiency of parallelization of the entire algorithm is determined by the efficiency of parallel solution of Poisson's equation.

5. NUMERICAL RESULTS

The flow in a cubic cavity with a moving lid was computed for the Reynolds numbers Re = 100 and Re = 1000 on uniform spatial grids with equal number of grid points along all coordinates ($N_1 = N_2 = N_3 = N_h$). At Re = 100, we used grids with $N_h = 22$ and $N_h = 42$. The number of grid points in computations with Re = 1000 was N_h = 42, 82, or 162. The time step Δt used in computations is shown in the table together with the step number n and instant t corresponding to the onset of a steady flow. The computations were performed

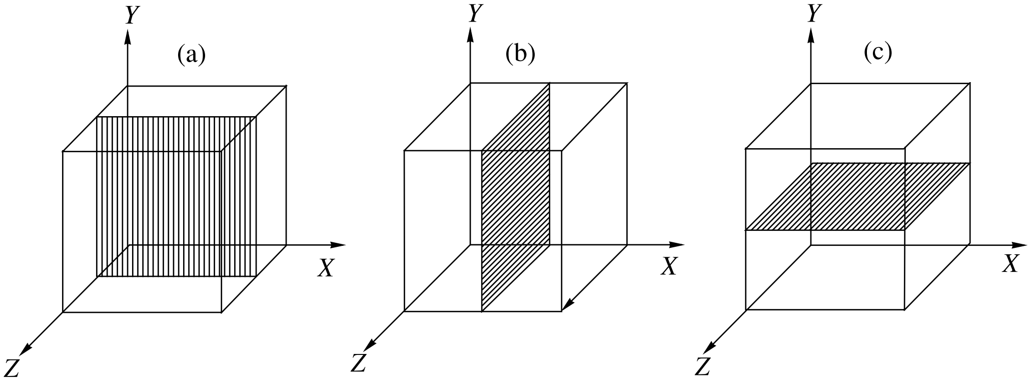

Fig. 3.

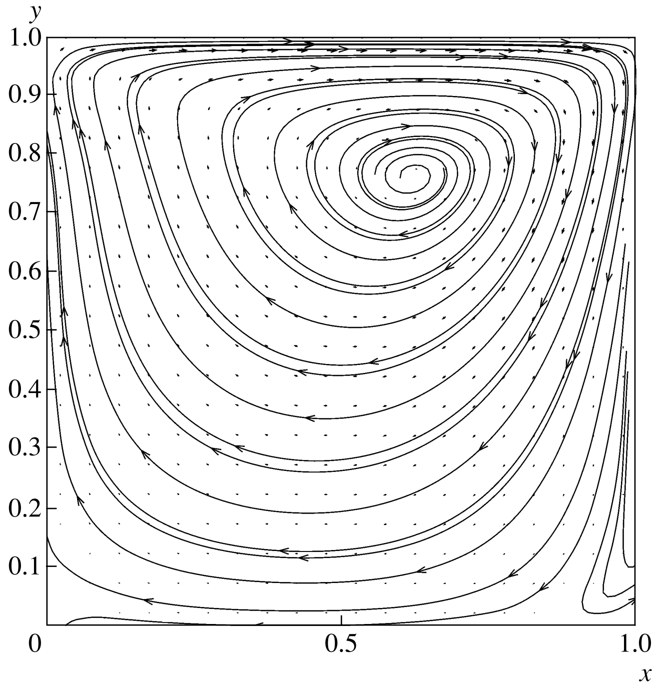

Fig. 4.

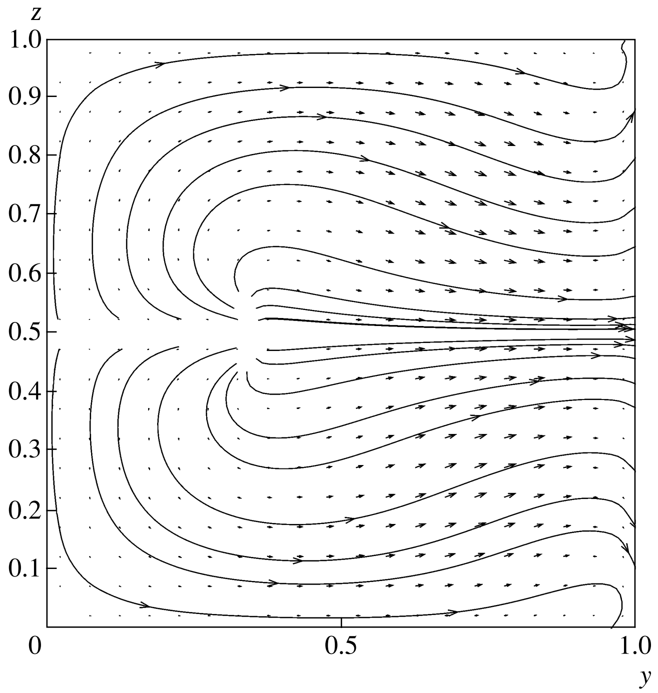

Fig. 5.

on an MVS-1000M computer with a single processor for the grid with N_h = 22 and with four, 16, and 25 processors for the grids with N_h = 42, 82, and 162, respectively.

In the case of Re = 1000 and N_h = 82, computations were performed on 50 time layers, and the run time was 21.18, 5.69, and 4.10 min for four, 16, and 25 processors, respectively. Practical computations demonstrate that the use of a greater number of processors is warranted for a greater number of grid points.

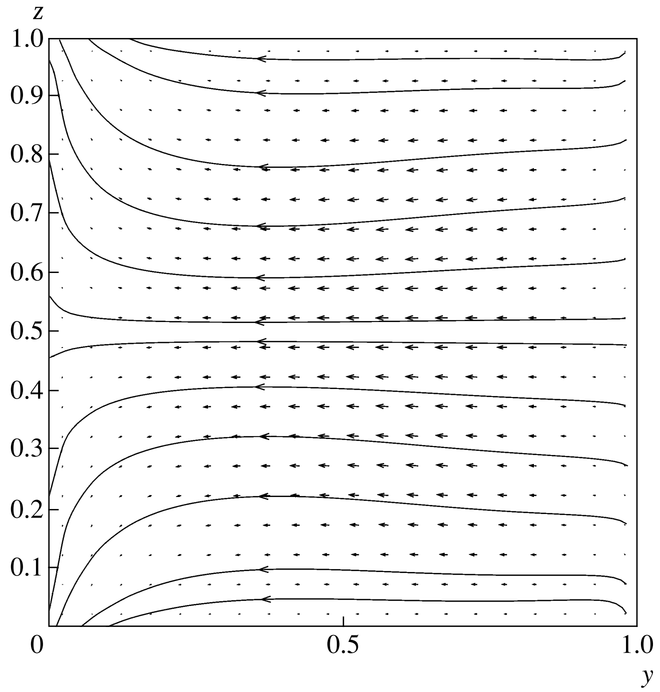

Fig. 6.

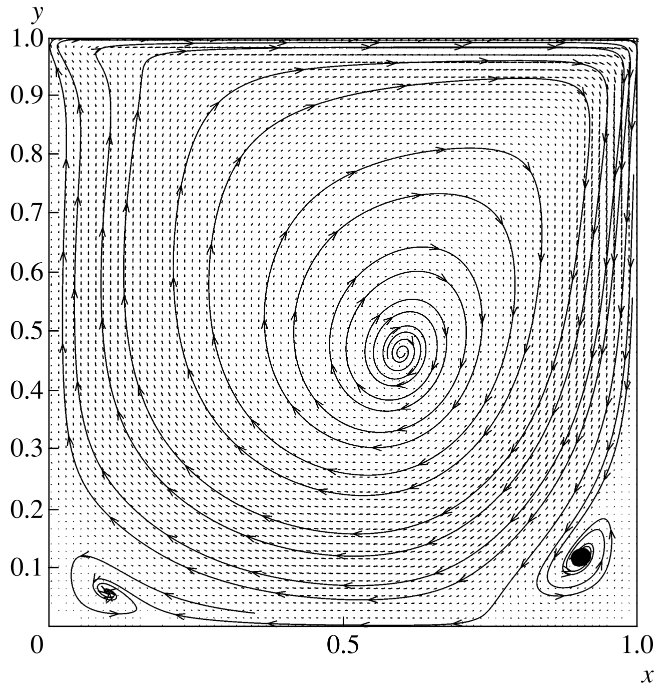

Fig. 7.

It is well known that the efficiency of an algorithm parallelized on a constant number of processors increases with the number of grid points, whereas it decreases as the number of processors increases while the number of grid points remains constant (see [14, 17]). When computations are to be performed on finer grids with the use of a greater number of processors, either the parallelization of MICCG(0) should be based on different orderings of unknowns or the elliptic equation should be solved by using a parallel version of the regularized alternating triangular method developed for three-dimensional problems in [21].

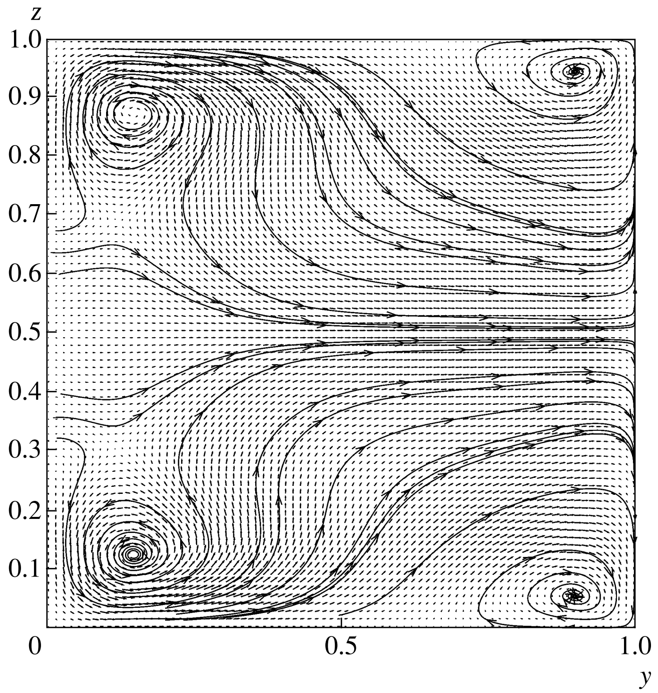

Fig. 8.

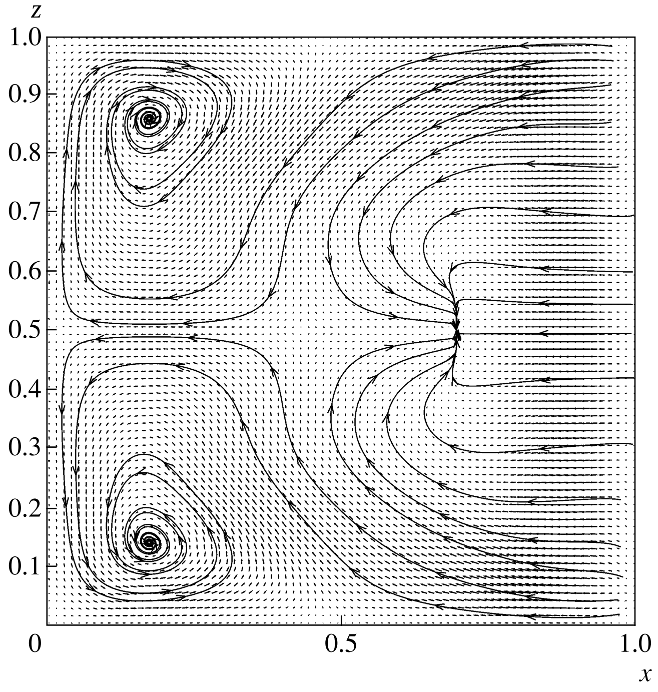

Fig. 9.

The streamlines and distributions of velocity components in the three central cross sections of the cavity with the coordinates $z = 0.5$ (Fig. 3a), $x = 0.5$ (Fig. 3b), and $y = 0.5$ (Fig. 3c) are shown in subsequent figures. The cross section $z = 0.5$ is the symmetry plane of the problem. Figures 4–6 illustrate the flow patterns computed in these cross sections for Re = 100 on the grid with $N_h = 22$. Specifically, Fig. 4 depicts the velocity components u_x and u_y and streamlines in the cross section $z = 0.5$; Fig. 5, the velocity components u_y and u_z and streamlines in the cross section $x = 0.5$; Fig. 6, the velocity components u_x and u_z and streamlines in the

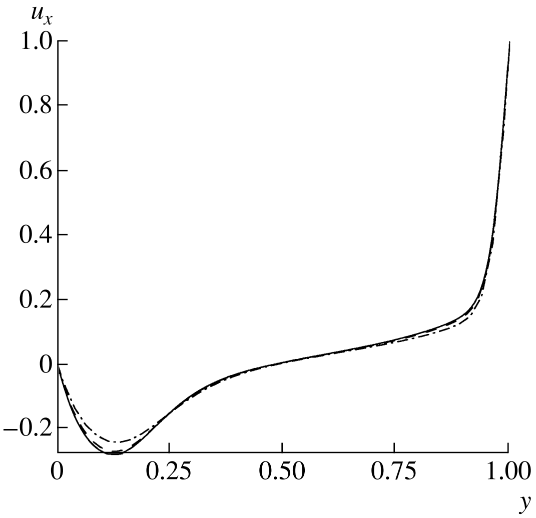

Fig. 10.

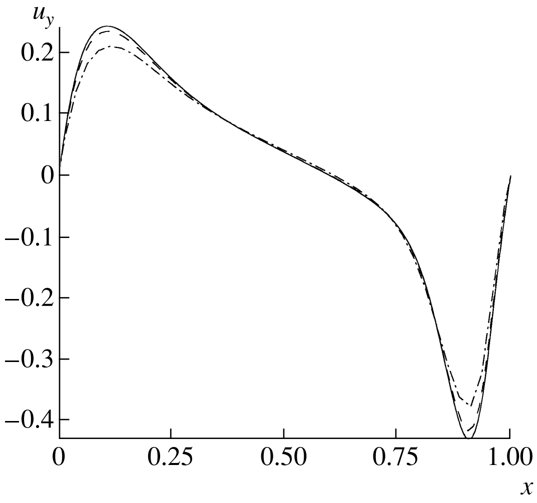

Fig. 11.

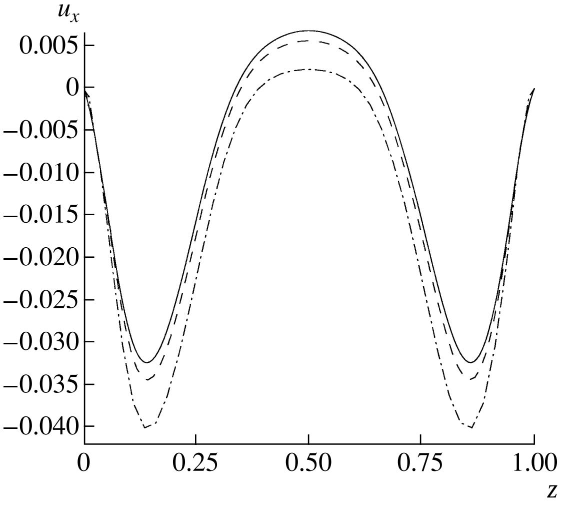

Fig. 12.

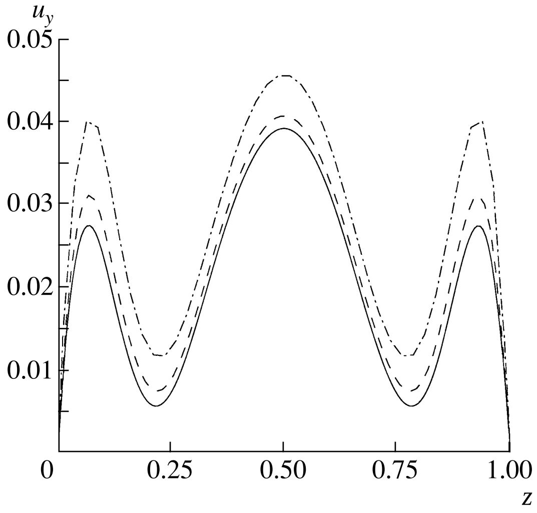

Fig. 13.

cross section $y = 0.5$. In this flow, the z-components of velocity are small, and results computed in this study are in good agreement with those obtained in two-dimensional computations [9] despite the complex flow pattern in the domain. Similar results were obtained in these cross sections on grids with $N_h = 22$ and 42.

Figures 7–9 show the streamlines computed in the same cross sections for Re = 1000 on the grid with $N_h = 82$. The last two demonstrate that the flow pattern becomes substantially more complicated with increasing Re as additional sources and sinks appear in these planes. As Re increases further, the flow splits into several vortices, which become unstable as the lid velocity is increased.

The divergent flow patterns computed at the bottom of the cavity ($y \approx 0$), its sides ($z \approx 0$ and $z \approx 1$), and the faces located at $x \approx 1$ and $x \approx 0$ for Re = 100 and 1000 are virtually identical to those presented in [4].

Figures 10–14 show the one-dimensional distributions of velocity components computed on progressively refined grids for Re = 1000. Solid, dashed, and dot-and-dash curves correspond to $N_h = 162$, 82, and 42, respectively. They graphically illustrate the convergence of numerical solution due to grid refinement.

In particular, Figures 10 and 11 depict the distributions of the horizontal and vertical velocity components ($u_y(x)$ at $z = 0.5$, $x = 0.5$ and $u_x(y)$ at $z = 0.5$, $y = 0.5$) in the symmetry plane of the cavity. These distributions, obtained on the grid with $N_h = 82$, are in good agreement with those presented in [2, 4] as results of computations performed on nonuniform grids refined toward domain boundaries with a number of grid

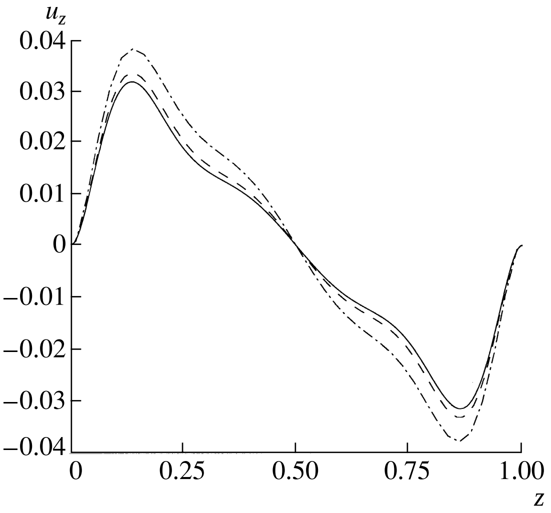

Fig. 14.

points ~6×10^4. Figures 12–14 illustrate the dependence of u_x, u_y, and u_z on z at $x = 0.5$ and $y = 0.5$. One can see that the velocity components plotted as functions of z are an order of magnitude smaller than those plotted as depending on location in the symmetry plane of the problem. As a function of z, the velocity exhibits a stronger dependence on the mesh size as compared to its variation in the symmetry plane.

These one-dimensional graphs can be used as a basis for a quantitative comparison of the results of three-dimensional cavity flow computations performed with the use of different numerical algorithms.

Note that the numerical algorithm for solving the Navier–Stokes equations somewhat similar to the one described above was applied by Fedoseyev in [5] to compute a number of flows, including steady cavity flows at high Reynolds numbers up to Re = 40 000. The basic idea of the algorithm was to regularize the continuity equation by rewriting it as (1.1), where the components of the vector $\mathbf{w}$ are calculated as $w^k = \tau\nabla^k p$. The remaining Navier–Stokes equations were written in the standard form, without any modification. The boundary condition $\partial p/\partial n = 0$ was used for pressure on the rigid wall. This approach made it possible to simulate the flows on relatively coarse grids.

CONCLUSIONS

In this paper, we present the results of numerical simulations of three-dimensional cavity flows based on the system of QHD equations written in an Eulerian coordinate system. To solve the system of equations obtained as a finite-difference approximation, we proposed an algorithm for parallel implementation, which made it possible to compute the problem on a cluster computer. To solve the second boundary value problem for three-dimensional Poisson's equation, we proposed and implemented a parallel version of MICCG(0). The numerical results obtained are compared with those presented in other publications. The convergence of the numerical solution on progressively refined grids is demonstrated.

The results presented here can be used in testing algorithms designed to compute three-dimensional flows. The algorithm developed in this study can be used to perform computations of three-dimensional unsteady flows in rectangular domains with reasonable time complexity.

ACKNOWLEDGMENTS

This work was supported by the Russian Foundation for Basic Research, project no. 01-01-00061, and by INTAS, grant no. 2000-0617.

REFERENCES

1. Fortov, V.E., Levin, V.K., Savin, G.I., *et al.*, The Supercomputer MVS-1000M and Prospects of Its Application, *Inform.-Analit. Zh. Nauka Prom-st Rossii*, 2001, no. 11(55), pp. 49–52.
2. Pokhilko, V.I., Solution of the Navier–Stokes Equations in a Cubic Cavity, *Preprint of Inst. of Math. Modeling, Russ. Acad. Sci.*, Moscow, 1994, no. 11.
3. Isaev, S.A., Luchko, N.N., Sudakov, A.G., and Sidorovich, T.V., Numerical Simulation of the Laminar Recirculating Flow in a Square Cavity with a Moving Boundary at High Reynolds Numbers, *Inzh.-Fiz. Zh.*, 2002, vol. 75, no. 1, pp. 54–60.
4. Isaev, S.A., Luchko, N.N., Sudakov, A.G., *et al.*, Numerical Simulation of the Laminar Recirculating Flow in a Cubic Cavity with a Moving Boundary, *Inzh.-Fiz. Zh.*, 2002, vol. 75, no. 1, pp. 49–53.
5. Fedoseyev, A.I., A Regularization Approach to Solving the Navier–Stokes Equations for Problems with Boundary Layer, *Comput. Fluid Dyn. J.*, Special Number 2001 (Proc. 8th ISCFD 1999 at ZARM, Bremen), pp. 317–324, paper II-16 *(http://uahtitan.uah.edu/alex/cfd_journal_ns99.pdf).*
6. Koseff, J.R. and Street, R.L., The Lid-Driven Cavity Flow: A Synthesis of Qualitative and Quantitative Observations, *Trans. ASME J. Fluids Eng.*, 1984, vol. 106, pp. 390–398.

7. Sheretov, Yu.V., Quasi-Hydrodynamic Equations as a Model for Compressible Viscous Heat-Conducting Flows, *Primenenie funktsional'nogo analiza v teorii priblizhenii* (Application of Functional Analysis in Approximation Theory), Tver: Tver. Gos. Univ., 1997, pp. 127–155.
8. Sheretov, Yu.V., *Matematicheskoe modelirovanie techeniya zhidkosti i gaza na osnove kvazigazodinamicheskikh i kvazigidrodinamicheskikh uravnenii* (Mathematical Modeling of Flows Based on Quasi-Gasdynamic and Quasi-Hydrodinamic Equations), Tver: Tver. Gos. Univ., 2000.
9. Gurov, D.B., Elizarova, T.G., and Sheretov, Yu.V., Numerical Simulation of Cavity Flows Based on the Quasi-Hydrodynamic System of Equations, *Mat. Modelirovanie*, 1996, vol. 8, no. 7, pp. 33–44.
10. Elizarova, T.G., Kalachinskaya, I.S., Klyuchnikova, A.V., and Sheretov, Yu.V., Application of Quasi-Hydrodynamic Equations in the Modeling of Low-Prandtl Thermal Convection, *Zh. Vychisl. Mat. Mat. Fiz.*, 1998, vol. 38, no. 10, pp. 1732–1742.
11. Elizarova, T.G. and Sheretov, Yu.V., Theoretical and Numerical Analysis of Quasi-Gasdynamic and Quasi-Hydrodynamic Equations, *Zh. Vychisl. Mat. Mat. Fiz.*, 2001, vol. 41, no. 2, pp. 239–255.
12. *Handbook of Computational Fluid Mechanics*, Peyret, R., Ed., London: Academic, 2000.
13. Gustafsson, I., A Class of First Order Factorization Methods, *BIT*, 1978, vol. 18, no. 1, pp. 142–156.
14. Chetverushkin, B.N., *Kineticheski-soglasovannye skhemy v gazovoi dinamike* (Kinetically Consistent Schemes in Gas Dynamics), Moscow: Mosk. Gos. Univ., 1999.
15. Duff, I.S. and Meurant, G.A., The Effect of Ordering on Preconditioned Conjugate Gradient, *BIT*, 1989, vol. 29, no. 4, pp. 635–657.
16. Duff, I.S. and van der Vorst, H.A., Developments and Trends in the Parallel Solution of Linear Systems, *Parallel Computing*, 1999, vol. 25, nos. 13/14, pp. 1931–1970.
17. Milyukova, O.Yu. and Chetverushkin, B.N., Parallel Version of the Alternating Triangular Method, *Zh. Vychisl. Mat. Mat. Fiz.*, 1998, vol. 38, no. 2, pp. 228–238.
18. Milyukova, O.Yu., Parallel Version of the Generalized Alternating Triangular Method for Elliptic Equations, *Zh. Vychisl. Mat. Mat. Fiz.*, 1998, vol. 38, no. 12, pp. 2002–2012.
19. Milyukova, O.Yu., Parallel Iterative Method with a Factorized Preconditioning Matrix for Elliptic Equations, *Differ. Uravn.*, 2000, vol. 36, no. 7, pp. 953–962.
20. Ortega, J.M., *Introduction to Parallel and Vector Solution of Linear Systems*, New York: Plenum, 1988. Translated under the title *Vvedenie v parallel'nye i vektornye metody resheniya lineinykh sistem*, Moscow: Mir, 1991.
21. Milyukova, O.Yu., Parallel Versions of the Alternating Triangular Conjugate-Gradient Method for Three-Dimensional Elliptic Equations, *Zh. Vychisl. Mat. Mat. Fiz.*, 2002, vol. 36, no. 10, pp. 1472–1481.