ELSEVIER

Comput. Methods Appl. Mech. Engrg. 137 (1996) 59–88

Computer methods in applied mechanics and engineering

A div–curl–grad formulation for compressible buoyant flows solved by the least-squares finite element method

Sheng-Tao Yu [a,*], Bo-Nan Jiang [b], Jie Wu [b], Nan-Suey Liu [c]

[a] *NYMA Technology, Inc., NASA Lewis Research Center, Brook Park, OH, USA*

[b] *Institute for Computational Mechanics and Propulsion, Ohio Aerospace Institute, NASA Lewis Research Center, Cleveland, OH, USA*

[c] *NASA Lewis Research Center, Cleveland, OH, USA*

Received 7 July 1995; revised 18 October 1995

Abstract

The present paper reports the development of the least-squares finite-element method for simulating compressible buoyant flows at low Mach numbers. We propose a div–curl–grad formulation with unknowns including vorticity, velocity, heat fluxes, temperature and pressure variation. The formulation is proved to be elliptic such that permissible boundary conditions become self-evident for a well posed flow problem. In contrast to conventional approaches, the present method evades the predicament of the 'singularity' problem of low-speed flows and no special treatment or artificial boundary condition is needed. Moreover, the assembled coefficient matrix is symmetric and positive-definite; its inversion is implemented by an element-by-element jacobi conjugate gradient method. As a numerical example, we calculate two-dimensional compressible buoyant flows inside a square enclosure at various Rayleigh numbers. For Rayleigh number one million, four secondary vortices were found embedded in the primary vortex. Due to significant temperature variations, the fluid flows are highly compressible in the interior. Along the walls, however, the flows are incompressible. The Nusselt number–Rayleigh number correlation deduced from the numerical result compared favorably with previously reported data.

1. Introduction

Perhaps the most neglected flow problem in finite-element simulations is the compressible buoyant flows at low Mach numbers, which on the contrary have been intensively studied using finite difference/volume methods. Because of heat addition, low-speed flows become compressible due to temperature and density variations. For example, motions of hot gases of a fire arise due to heat addition by combustion in an otherwise quiescent fluid in the presence of gravity. As another example, low-speed flows inside a chemical vapor deposition reactor are compressible due to strong heat radiation. Although the flow speed is slow, one must employ the compressible flow equations to simulate such flows. However, it is well known that the conventional methods, which can handle high-speed compressible flows easily, fail miserably when applied to these low-speed compressible flows.

The difficulty stems from two major issues: (1) the so-called 'singularity' problem of the low-Mach-number compressible flows, and (2) the lack of viable iterative methods for solving low-speed flows

*Corresponding author. E-mail: styu@rarefield.lerc.nasa.gov.

SSDI 0045-7825(95)00989-2

implicitly. In what follows, these two difficulties are briefly illustrated as the background of the present work.

1.1. Singularity problem for low-speed flows

In the setting of the mixed Galerkin method, it is well known that the incompressible Navier–Stokes equations have the so-called 'singular' behavior. The singularity stems from the exclusion of pressure as an unknown in the continuity equation. It has been explained [1] that the continuity equation is a kinematic constraint instead of an equal-footing governing equation such as the momentum equations. And pressure is the Lagrange multiplier of the system. In order to construct a stable and existing solution, pressure, i.e. the Lagrange multiplier, and the velocities, i.e. the solution of the governing equations, should be defined in different function spaces. Without observing this fact, instead treating them equally will render the equation set analytically singular.

According to Zienkiewicz and Wu [2] the numerical manifestation of this singularity can be illustrated by the structure of the linear algebraic system for the discretized flow equations:

$$\begin{pmatrix} \boldsymbol{K} & \boldsymbol{A} \\ \boldsymbol{B} & \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{p} \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}_1 \\ \boldsymbol{f}_2 \end{pmatrix}, \tag{1}$$

where $\boldsymbol{u}$ is the unknown vector of velocities for all grid nodes and $\boldsymbol{p}$ is that for pressure. With proper discretization, the matrix $\boldsymbol{K}$ is usually positive-definite. The upper half of the linear algebraic system, i.e. $\boldsymbol{K}\boldsymbol{u} + \boldsymbol{A}\boldsymbol{p} = \boldsymbol{f}_1$, represents the discretized momentum equations for every grid node and the lower part is that of the continuity equation. Due to the null diagonal of the global coefficient matrix, such systems are often singular. It has been shown that a necessary condition for the matrix to be invertible is that the number of $\boldsymbol{u}$ must be larger than that of $\boldsymbol{p}$. It should be emphasized, however, this condition is not sufficient.

For compressible flows, additional unknowns such as temperature and density are included. Pressure is related to temperature and density through the equation of state. Therefore, pressure could be eliminated from the flow equations as has been practised commonly in computing high-speed compressible flows. As a result, the abovementioned singularity problem ceases to exist; indeed, for high-speed compressible flows numerical difficulty of such nature never emerged.

For low-Mach-number compressible flows, pressure is rather uniform through out the whole flow field. Therefore, pressure can be considered as a small variation superimposed on a constant background such that the derivative of the constant background pressure can be dropped off. In this case, the pressure variation becomes the primitive unknown to be solved. In addition, as will be illustrated in the following section, the equation of state at low Mach numbers becomes a reciprocal correlation between temperature and density and the pressure variation is not directly related to them. As such, one cannot eliminate the pressure variation from the list of unknowns. Similar to the incompressible flows, the pressure variation does not appear in the continuity equation and the singularity problem emerges again. This situation can be illustrated by the following linear algebraic system,

$$\begin{pmatrix} \boldsymbol{K}_1 & \boldsymbol{A}_1 & \boldsymbol{A}_2 \\ \boldsymbol{K}_2 & \boldsymbol{B}_1 & \boldsymbol{B}_2 \\ \boldsymbol{K}_3 & \boldsymbol{C} & \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{T} \\ \boldsymbol{p}' \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}_1 \\ \boldsymbol{f}_2 \\ \boldsymbol{f}_3 \end{pmatrix}. \tag{2}$$

where $\boldsymbol{p}'$ is the pressure variation and $\boldsymbol{T}$ is the temperature. The upper part of the system, i.e. $\boldsymbol{K}_1\boldsymbol{u} + \boldsymbol{A}_1\boldsymbol{T} + \boldsymbol{A}_2\boldsymbol{p}' = \boldsymbol{f}_1$ and $\boldsymbol{K}_2\boldsymbol{u} + \boldsymbol{B}_1\boldsymbol{T} + \boldsymbol{B}_2\boldsymbol{p}' = \boldsymbol{f}_2$, are the discretized momentum and energy equations for all grid nodes and the lower part, i.e. $\boldsymbol{K}_3\boldsymbol{u} + \boldsymbol{C}\boldsymbol{T} = \boldsymbol{f}_3$, is that of the continuity equation. A diagonal null appears and the singularity problem is of concern.

Various methods were developed to overcome this 'singularity.' One approach is to ensure that the coefficient matrix with a null diagonal submatrix is not singular. In the setting of the mixed Galerkin

finite-element method, it has been quite clear that only certain combinations of the approximation functions for velocities and pressure are acceptable for stable solutions. i.e. the LBB condition [3] must be satisfied. Einset and Jensen [4] used the mixed Galerkin method and solved a low-Mach-number flow inside a chemical vapor deposition furnace. The final coefficient matrix of the mixed Galerkin method is usually non-symmetric; its inversion is not trivial.

Alternatively, in the same spirit of the non-equal-footingness of the flow equations, a fractional step procedure was proposed by Chorin [5]. In this method, one first calculates an intermediate velocity field by solving the momentum equations while ignoring the continuity equation. This intermediate velocity field is then corrected by the mass conservation constraint. The correction step was interpreted as a projection of velocities to a mass conservation function space and is usually facilitated by solving a Poisson equation. Essentially, by doing so, one introduces extra terms into the global matrix (2) to remove the the diagonal null and the singularity is resolved. It should be emphasized, however, due to the complexity of the method, error analysis is not available. Nevertheless, guided by engineering heuristics various computer programs were developed for low-Mach-number flows [6–9]. They are mixtures of the mixed interpolation and fractional step procedure. Usually, a staggered grid is employed, which, to some extent, is similar to the mixed Galerkin method. In the same spirit of the fractional step, the governing equations were solved sequentially. In addition to the lack of basic theory, the accuracy of the projection method is questionable [10] due to the employed artificial boundary conditions for pressure when solving the Poisson equation.

The abovementioned methods were designed on the premise that pressure must be included as a primitive variable. One can bypass the pressure and therefore the singularity problem by using the vorticity–velocity formulation. Essentially, by taking a curl of the momentum equations, the pressure derivatives are eliminated. Based on this approach, Ern et al. [11] developed a three-dimensional solver for low-Mach-number flows. The difficulty of this approach is the uncertainty of the boundary condition.

Recently, one noteworthy progress of simulating the low-speed compressible flows is the preconditioning methods by Merkle and Choi [12–14] and Turkel [15, 16] for both low-Mach-number flows and all-speed flows. The method is an extension of the computational schemes for aeronautical flows. The theory is based on the hyperbolic characteristic of the time marching method. The treatment is composed of two steps. First, according to Chorin [17], a temporal derivative of pressure together with a multiplicative variable β, i.e. the pseudo-compressibility term, is added to the continuity equation. As a result, the diagonal null in (2) is filled:

$$\begin{pmatrix} \boldsymbol{K}_1 & \boldsymbol{A}_1 & \boldsymbol{A}_2 \\ \boldsymbol{K}_2 & \boldsymbol{B}_1 & \boldsymbol{B}_2 \\ \boldsymbol{K}_3 & \boldsymbol{C} & \begin{bmatrix} \frac{1}{\beta\,\Delta t} & & 0 \\ & \frac{1}{\beta\,\Delta t} & \\ & & \ddots \\ 0 & & \frac{1}{\beta\,\Delta t} \end{bmatrix} \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{T} \\ \boldsymbol{p}' \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}_1 \\ \boldsymbol{f}_2 \\ \boldsymbol{f}_3 \end{pmatrix}. \tag{3}$$

The pseudo-compressibility term removes the singularity of the coefficient matrix, and a hyperbolic time marching for all equations becomes numerically viable even for incompressible flows.

Second, the propagation speed and damping effect of numerical waves are tuned up to speed up calculations towards the steady state. This is necessary due to the wide differences between the eigenvalues, i.e. u, $u+c$, and $u-c$, of the jacobian matrix at low Mach numbers ($u \ll c$). Numerically, the large difference between the eigenvalues results in unacceptable condition numbers of the discretized systems. This in turn makes the iterative methods for solving the algebraic system becoming drastically exasperated. When the eigenvalues are scaled to be of the same order of magnitude by premultiplying a preconditioning matrix to the equation set, the stability of this pseudo-time marching process is ensured. However, this method is based on conditioning the convective terms, little discussion for the

viscous terms was offered. In addition, since pressure becomes a primary unknown in this artificial system, one must specify its boundary condition. Usually, $\partial p/\partial \boldsymbol{n} = 0$ is used with $\boldsymbol{n}$ as the normal vector on the boundary. Similar to the projection methods, this pseudo boundary condition is derived based on a boundary-layer type assumption. For Stokes flows and recirculating flows, this artificial boundary condition poses significant error.

1.2. Iterative methods for unstructured meshes

Discretization of multi-dimensional Navier–Stokes equations using implicit methods results in large and sparse matrices to be inverted. Because the operator of the Navier–Stokes equations is not self-adjoint, conventional methods render the coefficient matrix asymmetric and/or non-positive-definite. As a result, the numerical inversion is a formidable task.

In the past, structured meshes were used as an integral part of the finite difference/volume methods. Due to its inherently ordered fashion of the data structure, the global matrix could be factorized approximately in either ADI or LU manner without increasing the bandwidth of the matrix. The structured-mesh methods were usually employed in conjunction with time-marching techniques and the numerical computation is guided by von Neumann stability analysis [18]. Generally, artificial damping is added to suppress the numerical instability caused by the factorization error. As a result, tremendous success has been obtained for large-scale simulations.

For unstructured-mesh methods, however, the coefficient matrix is no longer narrowly banded in an orderly fashion. Therefore, it cannot be factorized in a useful way. Instead of von Neumann analysis, one relies on the characteristics of the global matrix, such as the condition number, diagonal dominance, symmetry and so forth, for the strategy of choosing a viable iterative method. Since the matrices are often asymmetric and/or non-positive-definite, a robust iterative method which guarantees convergence does not exist. When the number of degrees of freedom is small, one can use the direct solver. For large multi-dimensional problems, however, the direct solver requires prohibitively large amount of memory and CPU time.

Currently, the most promising iterative methods are conjugate gradient (CG) type methods: the biconjugate gradient method (Bi-CG) [19], the biconjugate gradient stabilized method (Bi-CGSTAB) [20], the conjugate gradient squared method (CGS) [21], the conjugate gradient residual method (CGR) [22] and the generalized minimal residual method (GMRES) [23]. Note that, the fundamental strategy of the CG methods is an efficient search for the minimum, i.e. the solution. For non-positive-definite matrices, the existence of the minimum cannot be verified and the CG methods do not always provide the solution. As such, the application of these methods is an art guided by experience. For asymmetric but positive-definite matrices, the issues become the required computer memory and a robust preconditioning. For example, to use the GMRES method, one needs to stores at least 20 to 50 global vectors. The method usually takes hundreds of iterations for acceptable convergence. The efficiency of the calculations depends on the preconditioning procedure which is usually based on the incomplete LU factorization or the symmetric successive overrelaxation. Unfortunately, one has to design and tune up a special-purpose preconditioner for each problem of concern—no universal preconditioning procedure is available.

Here, we provide a brief account of applications of CG methods in CFD. Einset and Jensen [4] used CGS and GMRES methods for three-dimensional flows inside a chemical vapor deposition furnace. The simulated problem has about 61 000 degrees of freedom. Howard et al. [22] employed Bi-CG, CGR and CGS methods for solving a natural convection problem. They used about 10 000 degrees of freedom. Chin et al. [24] used CGS and Bi-CGSTAB methods to solve driven cavity flows, rearward-facing step flows and a channel flow. The largest problem has about 19 000 degrees of freedom. Strigberger et al. [25] used the GMRES method to solve a lid-driven cavity flow with the degrees of freedom about 6000. Venkatakrishnan et al. [26] used GMRES method with various preconditioners to simulate transonic flows over airfoils. The degrees of freedom were about 65 000. In general, the sizes of the reported simulations were lower than 100 000 degrees of freedom. Also note that these are fully coupled calculations. For fractional step type calculations, one can certainly solve larger problems. However, the generality of the computer code could be sacrificed.

1.3. The present work

The present work is an extension of the LSFEM developed by Jiang et al. [27–29]. In [27], div–curl–grad formulations and their ellipticity for incompressible flows were derived. In [28], Jiang and Povinelli showed that the LSFEM is optimal for elliptic problems in the sense that the global error is of the same order as the approximation errors. In [29], the versatility of the LSFEM was illustrated for various flow simulations.

The present work is part of an effort of using LSFEM to simulate chemically reactive flows. Previously, we used the LSFEM for computing quasi-one-dimensional flames [30]. Later on, a LSFEM solver for low-Mach-number flow was developed [31], in which the compressibility effect was employed as an unknown and a direct solver was used. In this paper, we propose a new, first-order, vorticity–velocity–pressure formulation for the low-Mach-number flows with veritable boundary consitions. In what follows, a brief account of the present paper is provided

In Section 2, we present the detailed derivation of the first-order formulation for low-Mach-number flows. Subsections included are the first-order equations, non-dimensionalization, the div–curl–grad system, ellipticity and boundary conditions. In Section 3, the LSFEM and the JCG method for solving the flow equations are elaborated. In the last section, we report the simulated results of compressible buoyancy flows inside a square enclosure. Four Rayleigh numbers and three sets of mesh are considered. The results are presented by contour plots and tabulated data.

2. Theoretical model

2.1. The first-order equations

Two-dimensional, compressible, viscous flow equations are of concern:

$$u\frac{\partial \rho}{\partial x} + v\frac{\partial \rho}{\partial y} + \rho\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) = 0, \tag{4}$$

$$\rho u\frac{\partial u}{\partial x} + \rho v\frac{\partial u}{\partial y} + \frac{\partial p}{\partial x} = \mu\frac{\partial}{\partial x}\left[2\frac{\partial u}{\partial x} - \frac{2}{3}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)\right] + \mu\frac{\partial}{\partial y}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right), \tag{5}$$

$$\rho u\frac{\partial v}{\partial x} + \rho v\frac{\partial v}{\partial y} + \frac{\partial p}{\partial y} = \mu\frac{\partial}{\partial y}\left[2\frac{\partial v}{\partial y} - \frac{2}{3}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)\right] + \mu\frac{\partial}{\partial x}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) - \rho g, \tag{6}$$

$$\rho C_p u\frac{\partial T}{\partial x} + \rho C_p v\frac{\partial T}{\partial y} - \left(u\frac{\partial p}{\partial x} + v\frac{\partial p}{\partial y}\right) = \Phi + k\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right) - \rho g v, \tag{7}$$

$$\Phi = 2\mu\left[\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 + \frac{1}{2}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)^2\right] - \frac{2}{3}\mu\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)^2, \tag{8}$$

where ρ is the density; u and v, the velocities in the respective directions; T, the temperature; and Φ, the viscous dissipation. Physical properties such as the viscosity μ, the conductivity k, and the constant pressure specific heat C_p are assumed constant throughout the flow field. Note that the coordinate system is chosen so that the gravity is in the negative y direction.

In order to reduce the equations to a first-order system, new variables are introduced,

$$\theta = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}, \tag{9}$$

$$\omega = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}, \tag{10}$$

$$q_x = k\frac{\partial T}{\partial x}, \tag{11}$$

$$q_y = k\frac{\partial T}{\partial y}, \tag{12}$$

where θ, the dilatation is a measure the compressibility; ω is the vorticity; and, q_x and q_y are the heat conduction fluxes in the respective directions. As a result, we obtain the following equations:

$$u\frac{\partial \rho}{\partial x} + v\frac{\partial \rho}{\partial y} + \rho\theta = 0, \tag{13}$$

$$\rho u\frac{\partial u}{\partial x} + \rho v\frac{\partial u}{\partial y} + \frac{\partial p}{\partial x} = \mu\left(\frac{4}{3}\frac{\partial \theta}{\partial x} - \frac{\partial \omega}{\partial y}\right), \tag{14}$$

$$\rho u\frac{\partial v}{\partial x} + \rho v\frac{\partial v}{\partial y} + \frac{\partial p}{\partial y} = \mu\left(\frac{4}{3}\frac{\partial \theta}{\partial y} + \frac{\partial \omega}{\partial x}\right) - \rho g, \tag{15}$$

$$\rho C_p u\frac{\partial T}{\partial x} + \rho C_p v\frac{\partial T}{\partial y} - \left(u\frac{\partial p}{\partial x} + v\frac{\partial p}{\partial y}\right) = \mu\left[\frac{4}{3}\theta^2 + \omega^2 + 4\left(\frac{\partial u}{\partial y}\frac{\partial v}{\partial x} - \frac{\partial u}{\partial x}\frac{\partial v}{\partial y}\right)\right] + \frac{\partial q_x}{\partial x} + \frac{\partial q_y}{\partial y} - \rho g v. \tag{16}$$

The alternative rule of partial differentiation for the heat conduction fluxes must be satisfied:

$$\frac{\partial q_x}{\partial y} - \frac{\partial q_y}{\partial x} = 0. \tag{17}$$

This irrotational condition for the heat conduction fluxes is necessary for constructing the div–curl–grad system to be presented in the next section. The above governing equations are closed by the equation of state,

$$p = \rho RT, \tag{18}$$

where R is the gas constant. Noted that θ is introduced to facilitate the derivation and it will not appear as a dependent variable in the final equation set.

2.2. Non-dimensionalization

The governing equations are non-dimensionalized by appropriate parameters:

$$\rho^* = \frac{\rho}{\rho_\infty}, \qquad u^* = \frac{u}{U_\infty}, \qquad v^* = \frac{v}{U_\infty},$$

$$\theta^* = \frac{\theta L}{U_\infty}, \qquad \omega^* = \frac{\omega L}{U_\infty}, \qquad p'^* = \frac{p'}{\rho_\infty U_\infty^2},$$

$$T^* = \frac{T}{T_\infty}, \qquad x^* = \frac{x}{L}, \qquad y^* = \frac{y}{L},$$

$$q_x^* = \frac{q_x}{\rho_\infty U_\infty T_\infty C_p}, \qquad q_y^* = \frac{q_y}{\rho_\infty U_\infty T_\infty C_p},$$

where ρ_∞, U_∞, T_∞ and L are reference values of density, velocity, temperature and a length scale. Note that special care is taken in nondimensionalizing the derivatives of the pressure. Since we are interested in the low-speed flows, the pressure distribution is rather uniform. Therefore, we consider pressure as composed of a small variations p' and a uniform background $\bar{p}$, i.e. $p = \bar{p} + p'$, where $\bar{p} = \rho_\infty RT_\infty$. The background pressure $\bar{p}$ then can be dropped out in the spatial derivatives. The pressure variation

p' exists due to the flow velocities and, therefore, is non-dimensionalized by a reference kinetic energy $\rho_\infty U_\infty^2$. The background material for this derivation can be found in [32–34]. The non-dimensionalized equations are

$$u\frac{\partial\rho}{\partial x}+v\frac{\partial\rho}{\partial y}+\rho\theta=0\,, \tag{19}$$

$$\rho u\frac{\partial u}{\partial x}+\rho v\frac{\partial u}{\partial y}+\frac{\partial p'}{\partial x}=\frac{1}{\mathrm{Re}}\left(\frac{4}{3}\frac{\partial\theta}{\partial x}-\frac{\partial\omega}{\partial y}\right), \tag{20}$$

$$\rho u\frac{\partial v}{\partial x}+\rho v\frac{\partial v}{\partial y}+\frac{\partial p'}{\partial y}=\frac{1}{\mathrm{Re}}\left(\frac{4}{3}\frac{\partial\theta}{\partial y}+\frac{\partial\omega}{\partial x}\right)-\frac{1}{2\epsilon\,\mathrm{Fr}}\rho\,, \tag{21}$$

$$\rho u\frac{\partial T}{\partial x}+\rho v\frac{\partial T}{\partial y}-(\gamma-1)M^2\left(u\frac{\partial p'}{\partial x}+v\frac{\partial p'}{\partial y}\right)=\frac{(\gamma-1)M^2}{\mathrm{Re}}\left[\frac{4}{3}\theta^2+\omega^2+4\left(\frac{\partial u}{\partial y}\frac{\partial v}{\partial x}-\frac{\partial u}{\partial x}\frac{\partial v}{\partial y}\right)\right]+\frac{\partial q_x}{\partial x}+\frac{\partial q_y}{\partial y}-\frac{(\gamma-1)M^2}{2\epsilon\,\mathrm{Fr}}\rho v \tag{22}$$

$$\theta=\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}, \tag{23}$$

$$\omega=\frac{\partial v}{\partial x}-\frac{\partial u}{\partial y}, \tag{24}$$

$$q_x=\frac{1}{\mathrm{Pe}}\frac{\partial T}{\partial x}, \tag{25}$$

$$q_y=\frac{1}{\mathrm{Pe}}\frac{\partial T}{\partial y}, \tag{26}$$

$$\frac{\partial q_x}{\partial y}-\frac{\partial q_y}{\partial x}=0\,. \tag{27}$$

Note that the superscript * has been dropped for convenience. The dimensionless numbers in the equations are defined as

$$\mathrm{M}=\frac{U_\infty}{\sqrt{\gamma RT_\infty}},\qquad \mathrm{Fr}=\frac{U_\infty^2}{2\epsilon gL},\qquad \mathrm{Ra}=\frac{2\epsilon gL^3}{\nu\alpha},$$

$$\mathrm{Pr}=\frac{\nu}{\alpha},\qquad \mathrm{Re}=\sqrt{\frac{\mathrm{Ra\,Fr}}{\mathrm{Pr}}},\qquad \mathrm{Pe}=\mathrm{Re\,Pr},$$

$$\alpha=\frac{k}{\rho_\infty C_p},\qquad \gamma=\frac{C_p}{C_v},$$

where M is the Mach number, Fr the Froude number, Ra the Rayleigh number, Pr the Prandtl number, Re the Reynolds number, Pe the Peclet number, α the thermal diffusivity, and γ the ratio of specific heats. The temperature difference parameter ϵ is defined as

$$\epsilon = \frac{\Delta T}{2T_\infty} = \frac{T_h - T_c}{T_h + T_c}, \tag{28}$$

where T_h and T_c are the specified hot and cold temperatures in a thermal convection problem. In addition, the non-dimensionalized equation of state is given by

$$1 + \gamma M^2 p' = \rho T. \tag{29}$$

2.3. The div–curl–grad formulation

For low-Mach-number flows ($M \ll 1$), the pressure derivative terms, the viscous dissipation terms and the buoyancy term in the energy equation (Eq. (22)), become negligible. In addition, according to the equation of state (Eq. (29)), the density and temperature become reciprocals of each other. With strong heat addition, the temperature and, therefore, the speed of sound is high, the assumption of a small Mach number can be widely applied to combustion phenomena and material processing procedures. Specifically, for Mach number less than one tenth, the approximate formulation is regarded as very accurate. This treatment is commonplace for low-Mach-number flows and can be found in [4, 32–34].

To proceed, the reciprocal correlation of temperature and density is used to obtain the following equation:

$$\partial\rho = -\frac{1}{T^2}\,\partial T. \tag{30}$$

Substituting it into the continuity equation (Eq. (19)), we get

$$u\frac{\partial T}{\partial x} + v\frac{\partial T}{\partial y} = T\theta. \tag{31}$$

Note that temperature was chosen as the primitive variable and density is replaced by the reciprocal of temperature. Substituting the definition of q_x and q_y, i.e. (25) and (26), into (31), we get the following algebraic equation for θ:

$$\theta = \frac{\mathrm{Pe}}{T}\left(uq_x + vq_y\right). \tag{32}$$

The left-hand side of Eq. (31) is the material derivative of temperature, which can be substituted into the energy equation (22), and we get

$$\frac{\partial q_x}{\partial x} + \frac{\partial q_y}{\partial y} - \theta = 0. \tag{33}$$

Note that the pressure derivatives, viscous dissipation and the gravitational source term have been neglected in (22) for low Mach numbers. This new form of the energy equation is a linear first-order equation in terms of heat fluxes. The nonlinear convective terms are now absorbed into θ, which is actually an algebraic expression in terms of temperature, velocities, and heat conduction fluxes, i.e. Eq. (32).

Similarly, we want to transform the nonlinear convective terms of the momentum equations into algebraic expressions. To this end, the total pressure b is introduced:

$$b = p' + \frac{1}{2T}\left(u^2 + v^2\right). \tag{34}$$

The convective terms of the momentum equations can be reformulated as

$$\rho u\frac{\partial u}{\partial x} + \rho v\frac{\partial u}{\partial y} + \frac{\partial p'}{\partial x} = \frac{\partial b}{\partial x} - \frac{v\omega}{T} + \frac{q_x\,\mathrm{Pe}}{2T^2}\left(u^2 + v^2\right), \tag{35}$$

$$\rho u\frac{\partial v}{\partial x} + \rho v\frac{\partial v}{\partial y} + \frac{\partial p'}{\partial y} = \frac{\partial b}{\partial y} + \frac{u\omega}{T} + \frac{q_y\,\mathrm{Pe}}{2T^2}\left(u^2 + v^2\right). \tag{36}$$

To proceed, we introduce another new variable $B = \mathrm{Re}\,b - 4/3\theta$ and the first-order equations are obtained for u, v, B, ω, q_x, q_y and T:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = \frac{\mathrm{Pe}}{T}\,(uq_x + vq_y)\,, \tag{37}$$

$$\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} = \omega\,, \tag{38}$$

$$\frac{\partial \omega}{\partial y} + \frac{\partial B}{\partial x} = \mathrm{Re}\left[\frac{v\omega}{T} - \frac{q_x\,\mathrm{Pe}}{2T^2}\,(u^2+v^2)\right], \tag{39}$$

$$-\frac{\partial \omega}{\partial x} + \frac{\partial B}{\partial y} = -\mathrm{Re}\left[\frac{u\omega}{T} + \frac{q_y\,\mathrm{Pe}}{2T^2}\,(u^2+v^2)\right] - \frac{1}{2\epsilon\,\mathrm{Fr}\,T}\,, \tag{40}$$

$$\frac{\partial T}{\partial x} = q_x\,\mathrm{Pe}, \tag{41}$$

$$\frac{\partial T}{\partial y} = q_y\,\mathrm{Pe}, \tag{42}$$

$$\frac{\partial q_x}{\partial x} + \frac{\partial q_y}{\partial y} = \frac{\mathrm{Pe}}{T}\,(uq_x + vq_y)\,, \tag{43}$$

$$\frac{\partial q_y}{\partial x} - \frac{\partial q_x}{\partial y} = 0\,. \tag{44}$$

These equations can be presented concisely by the notation of vector analysis:

$$\nabla\cdot\boldsymbol{u} = \frac{\mathrm{Pe}}{T}\,(uq_x + vq_y)\,, \tag{45}$$

$$\nabla\times\boldsymbol{u} = \boldsymbol{\omega}, \tag{46}$$

$$[\nabla\cdot\boldsymbol{\omega} = 0]\,, \tag{47}$$

$$\nabla\times\boldsymbol{\omega} + \nabla B = \boldsymbol{f}, \tag{48}$$

$$\nabla\cdot\boldsymbol{q} = \frac{\mathrm{Pe}}{T}\,(uq_x + vq_y)\,, \tag{49}$$

$$\nabla\times\boldsymbol{q} = 0\,, \tag{50}$$

$$\nabla T = \mathrm{Pe}\,\boldsymbol{q}, \tag{51}$$

where the right-hand side vector $\boldsymbol{f}$ in Eq. (48) is defined as

$$\boldsymbol{f} = \begin{pmatrix} \mathrm{Re}\left(\dfrac{v\omega}{T} - \dfrac{q_x\,\mathrm{Pe}}{2T^2}(u^2+v^2)\right) \\ -\mathrm{Re}\left(\dfrac{u\omega}{T} + \dfrac{q_y\,\mathrm{Pe}}{2T^2}(u^2+v^2)\right) - \dfrac{1}{2\epsilon\,\mathrm{Fr}\,T} \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \end{pmatrix}. \tag{52}$$

Note that all the right-hand sides are algebraic relations of unknowns, and they have nothing to do with the classification of the equation set. The two-dimensional case is of concern, and the unknown vectors are defined as

$$\boldsymbol{u} = (u, v, 0)\,,$$

$$\boldsymbol{\omega} = (0, 0, \omega)\,,$$

$$\boldsymbol{q} = (q_x, q_y, 0)\,.$$

Vorticity has only one component which is perpendicular to the two-dimensional plane and therefore the its divergence-free condition is automatically satisfied; (47) is put inside a bracket.

Apparently, these equations are composed of div–curl–grad systems: (45) and (46) are a div–curl system for velocities; (47) and (48) are a div–curl system for vorticity and scalar B; and, (49), (50) and (51) are a div–curl–grad system for the heat fluxes and temperature. As such, we arrive at a system with eight equations and seven unknowns. In the following section, we shall show that this seemingly 'over-determined' system due to the inconsistency between the number of the unknowns and the number of equations can be easily circumvented by introducing one dummy variable and the resultant equation set is elliptic and well-posed.

2.4. *Ellipticity of the formulation*

Since there are seven unknowns, the equation set cannot be classified by the ordinary method, which usually requires an even number of unknowns to form complex conjugate eigenvalues for elliptic systems. The first-order equations, i.e. (45) to (51), are composed of two div–curl systems and a div–curl–grad system. The first and second div–curl sets are constructed by two equations and two unknowns, i.e. $u - v$ in (45) and (46) and $\omega - B$ in (47) and (48). The third div–curl–grad set, however, has three unknowns T, q_x and q_y determined by four equations. Therefore, we introduce a dummy variable κ into the third div–curl–grad system:

$$\nabla \cdot \boldsymbol{q} = \frac{\mathrm{Pe}}{T}\,(uq_x + vq_y)\,, \tag{53}$$

$$\mathrm{rot}\,\boldsymbol{q} = 0, \tag{54}$$

$$-\mathrm{curl}\,\kappa + \nabla T = \mathrm{Pe}\,\boldsymbol{q}\,. \tag{55}$$

The technique of introducing a dummy variable to facilitate the classification of the first-order partial differential equations was originally due to Chang and Gunzburger [35]. The special notations rot and curl are used to accomodate the two-dimensional formulation and they are defined as

$$\mathrm{rot}\,\boldsymbol{q} = \frac{\partial q_y}{\partial x} - \frac{\partial q_x}{\partial y}, \tag{56}$$

$$\mathrm{curl}\,\kappa = \begin{pmatrix} \dfrac{\partial \kappa}{\partial y} \\ -\dfrac{\partial \kappa}{\partial x} \end{pmatrix}. \tag{57}$$

Taking a rot of Eq. (55) leads to $\Delta\kappa = 0$, because $\mathrm{rot}\,\boldsymbol{q} = \mathrm{rot}\,\nabla T = 0$. Thus, with an appropriate boundary condition, the solution of κ is a uniform constant in the computational domain. As a result, the introduction of the dummy variable κ does not change anything in the system. Facilitated by κ, we now have eight equations and eight unknowns:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = \frac{\mathrm{Pe}}{T}\,(uq_x + vq_y)\,, \tag{58}$$

$$\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} = \omega\,, \tag{59}$$

$$\frac{\partial B}{\partial x} + \frac{\partial \omega}{\partial y} = f_1\,, \tag{60}$$

$$\frac{\partial \omega}{\partial x} - \frac{\partial B}{\partial y} = -f_2\,, \tag{61}$$

$$\frac{\partial q_x}{\partial x} + \frac{\partial q_y}{\partial y} = \frac{\mathrm{Pe}}{T}\,(uq_x + vq_y)\,, \tag{62}$$

$$\frac{\partial q_y}{\partial x} - \frac{\partial q_x}{\partial y} = 0\,, \tag{63}$$

$$\frac{\partial \kappa}{\partial x} + \frac{\partial T}{\partial y} = q_y\,\mathrm{Pe}\,, \tag{64}$$

$$\frac{\partial T}{\partial x} - \frac{\partial \kappa}{\partial y} = q_x\,\mathrm{Pe}\,, \tag{65}$$

where the right-hand side terms f_1 and f_2 are defined in Eq. (52). We then cast the equation set into a matrix form:

$$\boldsymbol{A}_1 \frac{\partial \boldsymbol{q}}{\partial x} + \boldsymbol{A}_2 \frac{\partial \boldsymbol{q}}{\partial y} + \boldsymbol{S} = 0\,, \tag{66}$$

in which the unknown vector $\boldsymbol{q}$ is defined as

$$\boldsymbol{q} = (u, v, B, \omega, q_x, q_y, \kappa, T)^{\mathrm{T}}, \tag{67}$$

and the right-hand side vector is

$$\boldsymbol{S} = -(f_0, \omega, f_1, -f_2, f_0, 0, q_y\,\mathrm{Pe}, q_x\,\mathrm{Pe})^{\mathrm{T}}, \tag{68}$$

where f_0 is equivalent to θ and is defined in Eq. (32). The coefficient matrices are

$$\boldsymbol{A}_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \tag{69}$$

and

$$\boldsymbol{A}_2 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \end{pmatrix}. \tag{70}$$

The characteristic polynomial associated with the system is

$$\det\left(\boldsymbol{A}_1 - \boldsymbol{A}_2 \frac{\mathrm{d}y}{\mathrm{d}x}\right) = 0, \tag{71}$$

where $\mathrm{d}y/\mathrm{d}x = \lambda$ is the eigenvalues of the system and is the direction of the characteristic surface. The first-order system is said to be hyperbolic if all eigenvalues are real. If all eigenvalues are complex, the system is elliptic. And if some are real and other complex, the system is considered as mixed type. In [36], background material about the classification of the first-order system can be found. Here, the eigenvalues can be obtained by the following procedure,

$$\begin{aligned}
\det(\boldsymbol{A}_1 - \boldsymbol{A}_2\lambda) &= \det\begin{pmatrix} \boldsymbol{C} & 0 & 0 & 0 \\ 0 & \boldsymbol{C} & 0 & 0 \\ 0 & 0 & \boldsymbol{C} & 0 \\ 0 & 0 & 0 & \boldsymbol{C} \end{pmatrix} \\
&= (\det \boldsymbol{C})^4 \\
&= \left(1+\lambda^2\right)^4 \\
&= 0,
\end{aligned} \tag{72}$$

where the submatrix $\boldsymbol{C}$ is

$$\boldsymbol{C} = \begin{pmatrix} 1 & -\lambda \\ \lambda & 1 \end{pmatrix}. \tag{73}$$

Thus, there is no real root for the eigenvalue; indeed, the equation set is elliptic.

2.5. Permissible boundary conditions

Because the system of such equations is elliptic, the required boundary conditions are those for a standard boundary value problem. Since the equation set is first-order, only Dirichlet boundary conditions are permissible. Facilitated by the dummy variable κ, we have eight unknowns governed by eight equations. Therefore, on each boundary we need four boundary conditions. For the purpose of discussion, we divide the system of equations into two groups: the flow equations, Eqs. (58)–(61); and the heat equations, Eqs. (62)–(65). On each boundary, each group requires two boundary conditions. We propose the following boundary conditions in Table 1.

where the outward normal vector for the boundary is denoted by $\boldsymbol{n}$ and c's are the specified distributions of the flow variable. Without losing generality, the symmetry condition is assumed to be with respect to the x axis. Note that for certain combination, a null boundary condition for the dummy variable κ is invoked in order to make the proposed system well posed. It should be emphasized, however, this is

Table 1
Boundary conditions

Conditions	Flow equations	Heat equations
Wall	$u = v = 0$	$T = c, \quad \boldsymbol{q} \times \boldsymbol{n} = c$ $\boldsymbol{q} \cdot \boldsymbol{n} = c, \quad (\kappa = 0)$
Specified inlet (outlet)	$u = c, \quad v = c$	$T = c, \quad \boldsymbol{q} \times \boldsymbol{n} = c$ $\boldsymbol{q} \cdot \boldsymbol{n} = c, \quad (\kappa = 0)$
Symmetry	$v = 0, \quad \omega = 0$	$q_y = 0, \quad (\kappa = 0)$

only for the purpose of discussion, because the dummy variable is not used in the numerical solution. These null boundary conditions are put in parentheses.

In the conventional vorticity–velocity methods, there has been endeavors to derive the wall conditions for vorticity. For projection methods and the preconditioning methods, one always needs to impose artificial boundary condition for pressure. Here, we have shown that, since only two boundary conditions are required from the flow equations, no wall boundary condition for vorticity or pressure is needed because the noslip condition ($u = v = 0$) is enough to make the system well posed. Similarly, no boundary condition for vorticity or pressure is needed for specified inflow or outflow.

For heat equations, there are two types of boundary conditions: (1) specified temperature condition; and, (2) specified heat flux condition. For the first case, it is necessary to impose an accompanying consistency condition for heat flux tangential to the wall, i.e. $\boldsymbol{q} \times \boldsymbol{n} = \boldsymbol{q}_\tau = c$, where $\boldsymbol{q}_\tau$ is the projection of $\boldsymbol{q}$ in the direction tangential to the boundary. For two-dimensional equations, this cross product results in a scalar relation. This tangential heat flux condition can be deduced from the specified wall temperatures. Although not shown in the present paper, this boundary condition for tangential heat fluxes is crucial for accurate solutions. We shall address this issue in our future publication.

For the specified-heat-flux boundary condition, we activate a null boundary condition for κ in addition to the specified normal heat flux. Note that κ is a predetermined constant over the whole flow field and the adoption of this pseudo-boundary condition poses no theoretical difficulty.

3. The least-squares finite element method

As mentioned in the previous section, the dummy variable κ is not included in the numerical solution. Therefore, the first-order system of eight equations and seven unknowns are solved by the LSFEM. A vector form of these equations are considered,

$$\boldsymbol{A}_1 \frac{\partial \boldsymbol{q}}{\partial x} + \boldsymbol{A}_2 \frac{\partial \boldsymbol{q}}{\partial y} + \boldsymbol{S} = 0\,, \tag{74}$$

where each entry of the right-hand side vector $\boldsymbol{S}$ is a nonlinear algebraic equation of seven dependent variables to be solved, i.e.

$$s_i = s_i(q_1, q_2, \ldots, q_7) \quad i = 1, 2, \ldots, 8 \tag{75}$$

where $q_j, j = 1, \ldots, 7$ are the dependent variables. These non-linear terms are linearized by Newton's method in the following fashion:

$$s_i^{n+1} = s_i^n + \sum_{j=1}^{7} \left(\frac{\partial s_i}{\partial q_j} \right)^n \Delta q_j\,, \tag{76}$$

where the superscript n denotes the previous iteration step and $n+1$ is the current step. $\Delta q_j = q_j^{n+1} - q_j^n$ is the increment of the flow variables in each Newton's iteration. After manipulation, we obtain a new set of equations in vector form ready for finite-element discretization,

$$\boldsymbol{A}_0^n\, \Delta \boldsymbol{q} + \boldsymbol{A}_1^n \frac{\partial \Delta \boldsymbol{q}}{\partial x} + \boldsymbol{A}_1^n \frac{\partial \boldsymbol{q}^n}{\partial x} + \boldsymbol{A}_2^n \frac{\partial \Delta \boldsymbol{q}}{\partial y} + \boldsymbol{A}_2^n \frac{\partial \boldsymbol{q}^n}{\partial y} + \boldsymbol{S}^n = 0\,. \tag{77}$$

To proceed, the governing equations are cast into the following operator form:

$$\boldsymbol{L}\, \Delta \boldsymbol{q} = \boldsymbol{f}\,, \tag{78}$$

where the linear operator $\boldsymbol{L}$ is defined as

$$\boldsymbol{L} = \boldsymbol{A}_0^n + \boldsymbol{A}_1^n \frac{\partial}{\partial x} + \boldsymbol{A}_2^n \frac{\partial}{\partial y}\,. \tag{79}$$

The right-hand side vector is

$$\boldsymbol{f} = -\boldsymbol{A}_1^n \frac{\partial \boldsymbol{q}^n}{\partial x} - \boldsymbol{A}_2^n \frac{\partial \boldsymbol{q}^n}{\partial y} - \boldsymbol{S}^n. \tag{80}$$

We then define the least-squares functional of the residual $\boldsymbol{R} = \boldsymbol{L}\,\Delta\boldsymbol{q} - \boldsymbol{f}$ for admissible $\Delta\boldsymbol{q}$ as

$$J(\Delta\boldsymbol{q}) = \int_{\Omega} \boldsymbol{R}^{\mathrm{T}} \cdot \boldsymbol{R}\,\mathrm{d}\Omega. \tag{81}$$

Minimizing the least-squares functional $J(\Delta\boldsymbol{q})$ with respect to $\Delta\boldsymbol{q}$ leads to

$$\delta J(\Delta\boldsymbol{q}) = 0\,. \tag{82}$$

That is,

$$\int_{\Omega} (\boldsymbol{L}\delta\,\Delta\boldsymbol{q})^{\mathrm{T}} \cdot (\boldsymbol{L}\,\Delta\boldsymbol{q} - \boldsymbol{f})\;\mathrm{d}\Omega = 0\,, \tag{83}$$

where δ denotes the variation of the function. Let $\delta\Delta\boldsymbol{q} = \boldsymbol{v}$, and Eq. (83) can be written as

$$\int_{\Omega} (\boldsymbol{L}\boldsymbol{v})^{\mathrm{T}}(\boldsymbol{L}\,\Delta\boldsymbol{q})\,\mathrm{d}\Omega = \int_{\Omega} (\boldsymbol{L}\boldsymbol{v})^{\mathrm{T}}\boldsymbol{f}\,\mathrm{d}\Omega\,. \tag{84}$$

To employ the finite element method, the computational domain is decomposed into N_e elements and the element shape functions Φ_i's are introduced. The discretized solution in each element $\Delta\boldsymbol{q}_h^e(t,x,y)$ can be expressed as

$$\Delta\boldsymbol{q}_h^e(t,x,y) = \sum_{i=1}^{N_{\mathrm{n}}} \Phi_i(x,y)(\Delta\boldsymbol{Q}_i(t))^e, \tag{85}$$

where N_{n} is the number of nodes per element and the $(\Delta\boldsymbol{Q}_i(t))^e$ are the nodal values of $\Delta\boldsymbol{q}$. The test function $\boldsymbol{v}$ is chosen as

$$\boldsymbol{v}(x,y) = \Phi_i(x,y)\boldsymbol{I}, \tag{86}$$

where $\boldsymbol{I}$ is the identity matrix. Substituting Eqs. (85) and (86) into (84) gives the linear algebraic equation

$$\boldsymbol{K}^n\,\Delta\boldsymbol{Q} = \boldsymbol{F}^n, \tag{87}$$

where $\Delta\boldsymbol{Q}$ denotes the global nodal values of $\Delta\boldsymbol{q}(x,y)$ and the final global matrix is

$$\boldsymbol{K}^n = \sum_{e=1}^{N_e} (\boldsymbol{K}^n)^e\,. \tag{88}$$

That is, the global matrix $\boldsymbol{K}^n$ is assembled from the element matrix $(\boldsymbol{K}^n)^e$, which is defined as

$$(\boldsymbol{K}_{ij}^n)^e = \int_{\Omega_e} (\boldsymbol{L}\Phi_i)^{\mathrm{T}} \cdot (\boldsymbol{L}\Phi_j)\,\mathrm{d}\Omega\,. \tag{89}$$

The final right-hand side vector $\boldsymbol{F}^n$ is assembled from the element vector $(\boldsymbol{F}_i^n)^e$, which is given as

$$(\boldsymbol{F}_i^n)^e = \int_{\Omega_e} (\boldsymbol{L}\Phi_i)^{\mathrm{T}} \cdot \boldsymbol{f}\,\mathrm{d}\Omega\,. \tag{90}$$

An important feature of the least-squares finite element method can be observed in Eqs. (84) and (89). That is, the final global matrix is symmetric. This symmetry is obtained by the self inner product of $\boldsymbol{L}\Phi$ in Eq. (89). In addition, as long as the solution exists, the global matrix is also positive-definite.

For inverting the matrix, we used the JCG method [37], which is considered as an efficient and straightforward approach for inverting a symmetric, positive-definite matrix. Because the Jacobi preconditioning in the JCG method consists of modifying only the diagonal terms of the global matrix, the preconditioned global matrix does not suffer from any fill-in. In this way, significant savings of computer memory can be obtained and no numerical accuracy is compromised. Therefore, in the present calculations, we only store the global vectors for the unknowns, unknown increments and the geometric data. We consider this merit of the LSFEM especially attractive for large scale calculations.

4. Results and discussion

The numerical example is the simulation of buoyancy-driven gas flows in a square enclosure. As shown in Fig. 1, the configuration consists of two insulated horizontal walls and two vertical walls at different temperatures, T_h and T_c. Fig. 1 also illustrates the specified boundary conditions. This problem has been extensively studied based on the incompressible flow equations with Boussinesq approximation, which is appropriate only for small temperature differences between T_h and T_c. In practice, however, a large temperature difference is frequently encountered, and the compressible formulation must be employed. Previously, Chenoweth and Paolucci [38] used a SIMPLE type method and performed an in-depth study of the flow field. As a result, heat transfer correlations are deduced and reported, including relationships between Nusselt number, Rayleigh number, aspect ratio of geometry, temperature difference and so forth. By using the preconditioning method, Choi and Merkle [14] also successfully calculated the flow field. Their results compared favorably with Chenoweth's data.

Flow features of the buoyancy-driven cavity flow depend on Rayleigh number Ra, Froude number Fr, the aspect ratio of the cavity and the temperature difference parameter ϵ. For the present study, four Rayleigh numbers, $\mathrm{Ra} = 10^3$, 10^4, 10^5 and 10^6, are considered with a temperature difference parameter $\epsilon = 0.6$, i.e. $T_h/T_c = 4$. In all four cases, the Froude number and the aspect ratio are unity. For grid refinement study, three meshes were used: 33×33, 65×65 and 129×129. In all three meshes, grid lines were clustered near all four wall to resolve high gradients of the flow properties near walls. However, the grid stretching is moderate with the ratio $h_{\max}/h_{\min} = 3.7$ for all calculations.

There are two loops of iterations: the outer loop updates the coefficient matrices and source terms, i.e. Eq. (76), by Newton's method; while the inner loop solves the variable increment Δq by inverting the global matrix using the JCG method. Fig. 2 shows a typical numerical convergence of the inner iterations by the JCG method. The case shown here is $\mathrm{Ra} = 10^6$ with the 129×129 mesh. In all calculations, we used bi-quadratic shape functions and the integration is done by using a two-point Gauss quadrature. Fig. 2 shows that the error drops to be lower than the sixth decimal digit in about 300 JCG iterations. For the present calculations, the specified criteria for stopping the inner iterations are $error \leqslant 10^{-7}$ or 300 iterations, whichever is satisfied first. The *error* in the JCG method is defined as the marching distance of the CG method divided by the absolute value of the global unknown vector.

Fig. 3 shows a typical convergence of the outer loop by Newton's method for the same case. In about six iterations, the magnitude of Δq reduces about seven orders of magnitude. Since the criterion of the inner iteration by the JCG method is set at 10^{-7}, the convergence of the outer iterations levels off at approximately the same order of magnitude. Fig. 3 also shows a quadratic convergence, which is a typical characteristic of Newton's method.

Fig. 4 shows the comparison of the present result with previous data reported by Chenoweth et al. [38]. The x-axis is the Rayleigh number in a logarithmic scale and the y-axis is the Nusselt number. Flow

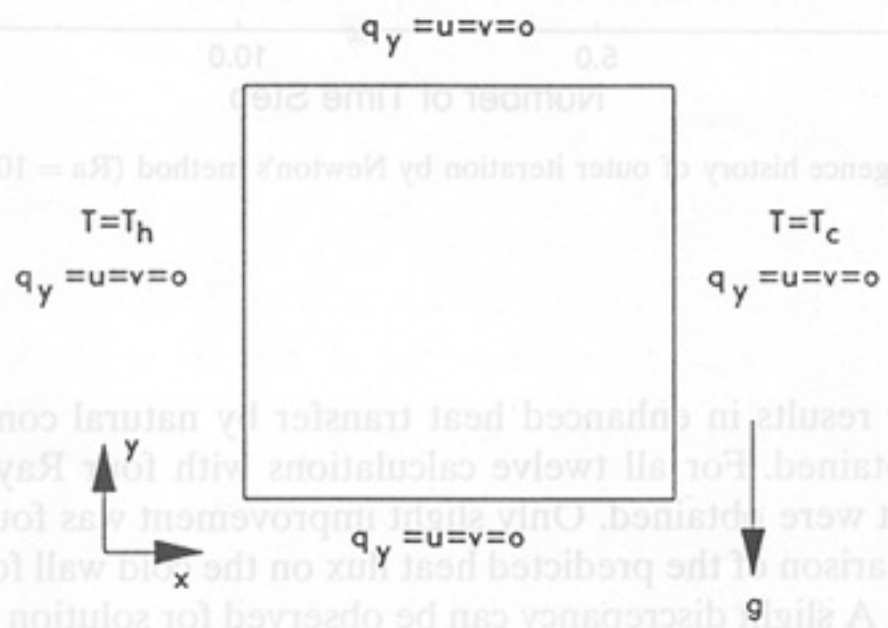

Fig. 1. Schematic of natural convection inside a square enclosure.

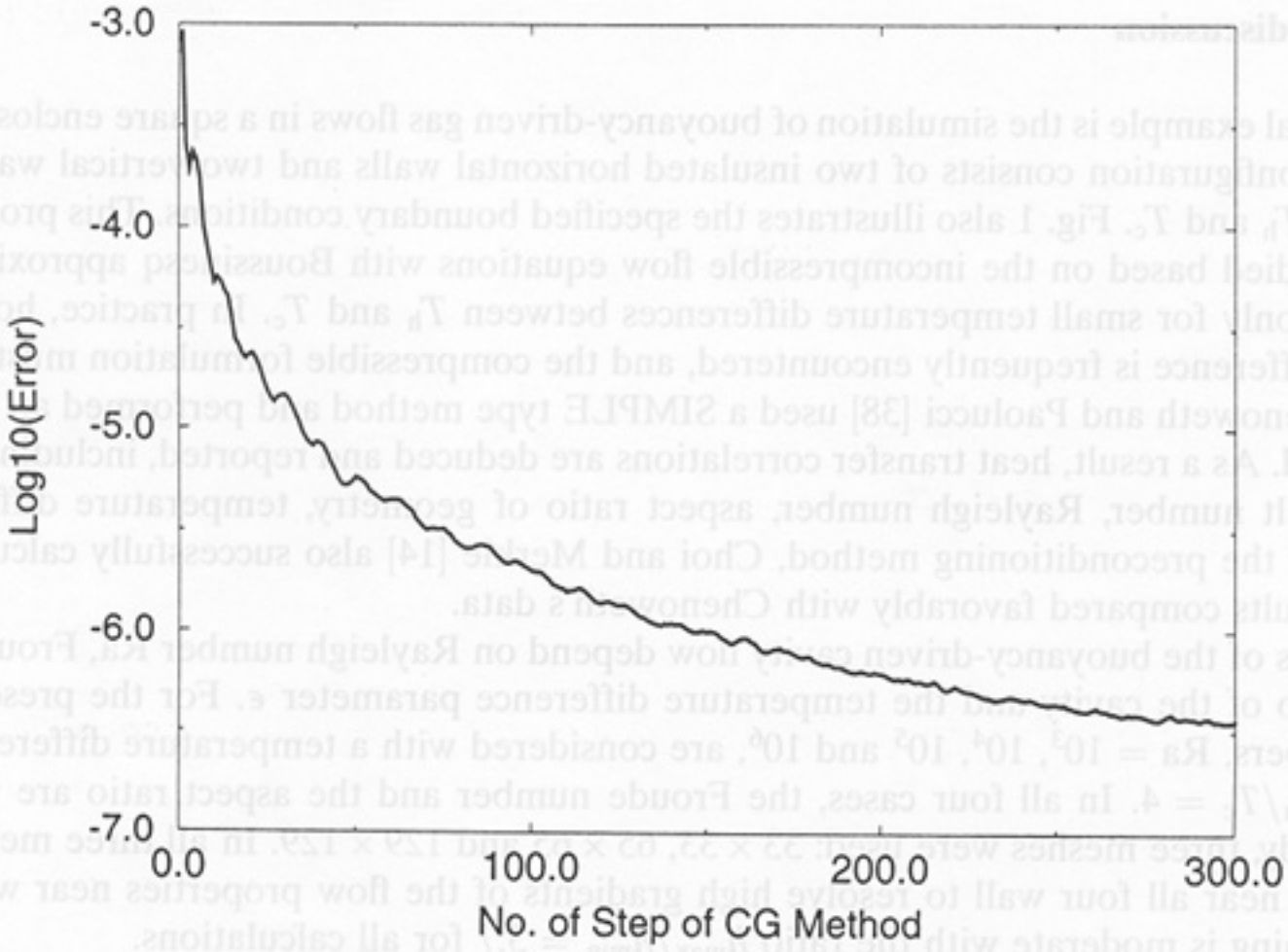

Fig. 2. A typical convergence history of inner iteration by the JCG method ($Ra = 10^6$, 129×129 mesh).

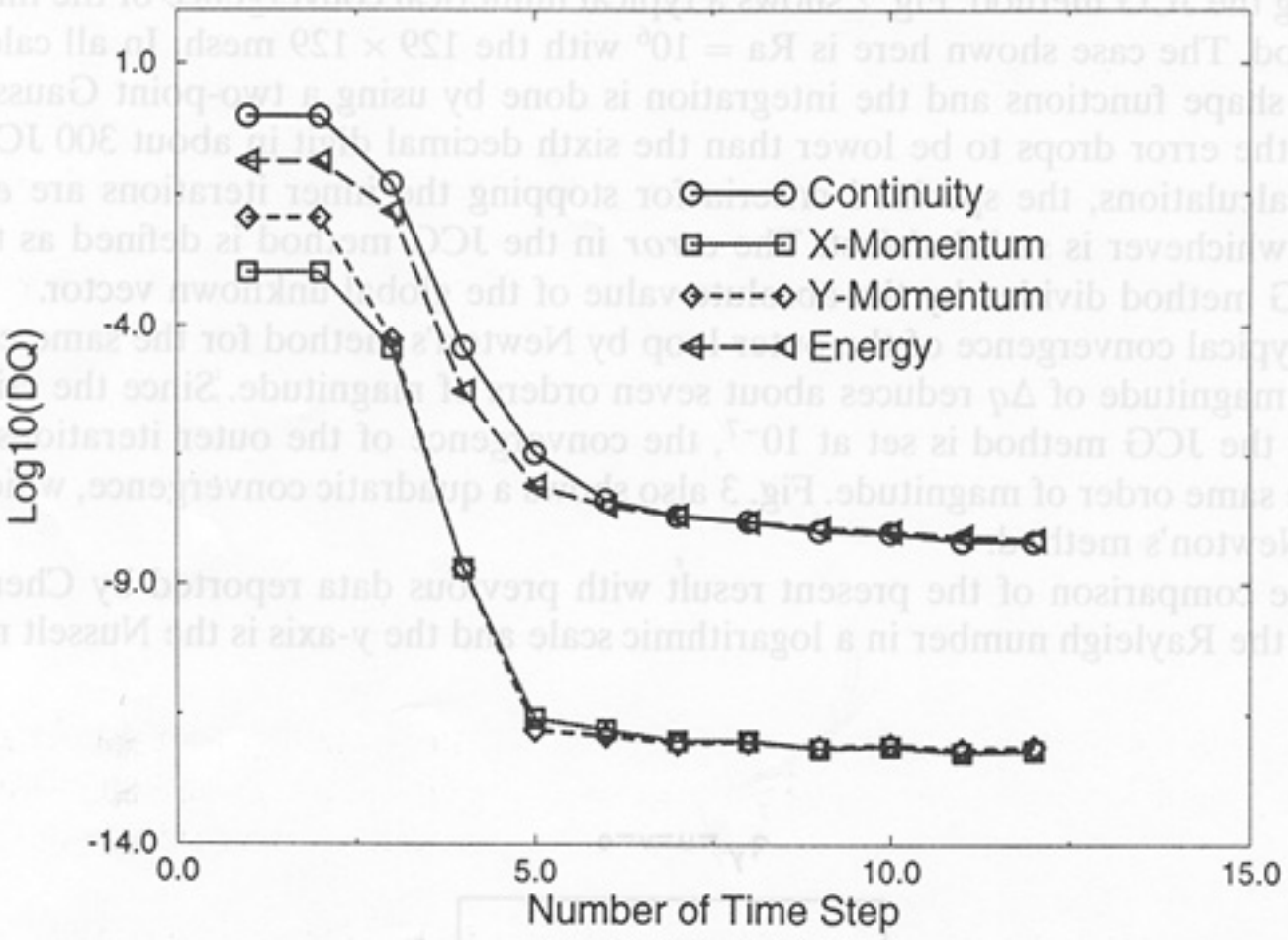

Fig. 3. A typical convergence history of outer iteration by Newton's method ($Ra = 10^6$, 129×129 mesh).

with large Rayleigh number results in enhanced heat transfer by natural convection and, therefore, a larger Nusselt number is obtained. For all twelve calculations with four Rayleigh numbers and three meshes, favorable agreement were obtained. Only slight improvement was found for solution using the fine mesh. Fig. 5 shows comparison of the predicted heat flux on the cold wall for the case of $Ra = 10^6$ by using three different meshes. A slight discrepancy can be observed for solution between the 33×33 mesh and the other two meshes. For 65×65 and 129×129 meshes, the solution is essentially same; a mesh-

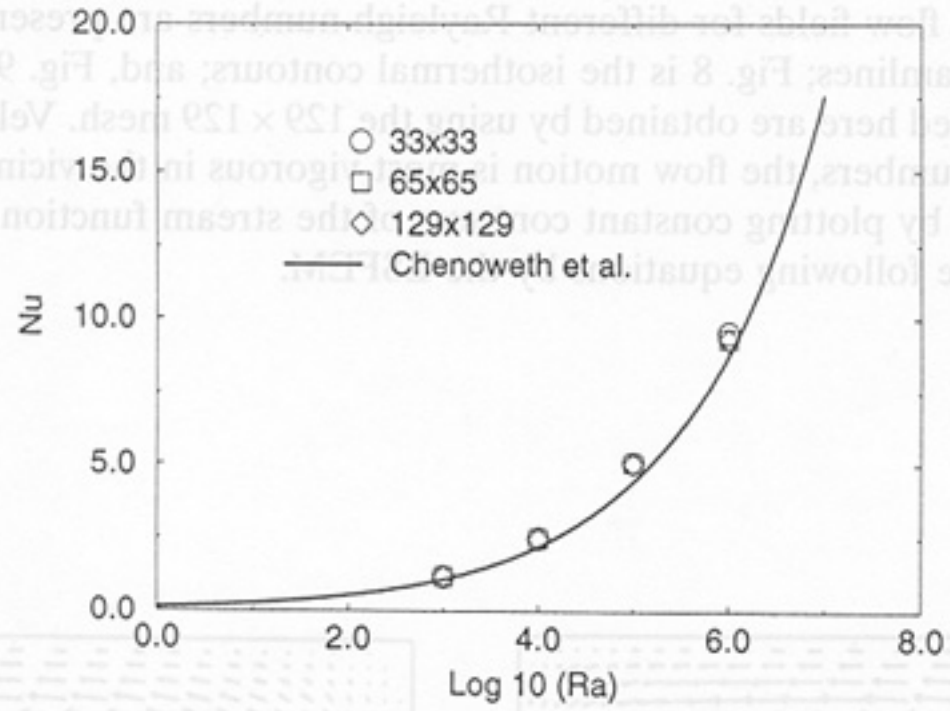

Fig. 4. The comparison of the calculated Nusselt numbers with Chenoweth and Paolucci's correlation.

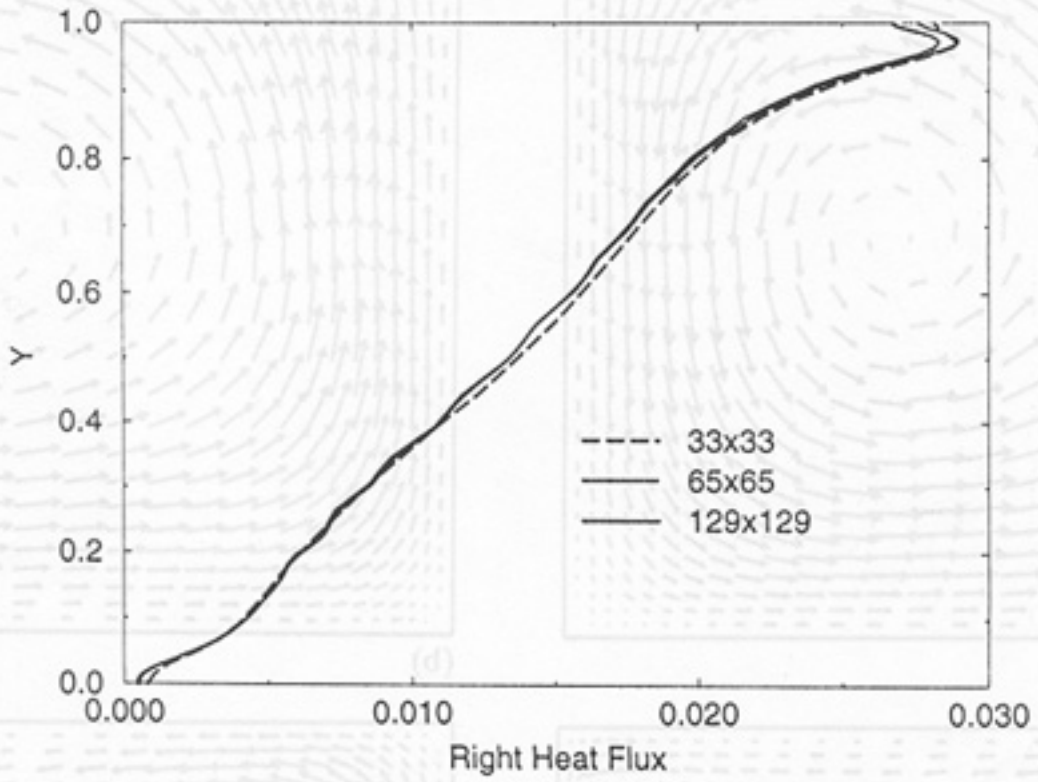

Fig. 5. Mesh refinement calculations ($Ra = 10^6$).

independent solution is obtained. For self-consistency test, we integrated q_x along the heat conducting walls to check the global energy balance. The differences between the integrated heat fluxes of the hot wall and cold wall are tabulated in Table 2. The data are presented in percentage with respect to the averaged values.

Errors of the global energy conservation for all calculations are less than one percentage. Perhaps this is the reason that good agreements were found in Fig. 4 for all meshes. The global heat transfer effect is relatively insensitive to the mesh size.

Table 2
Errors of overall heat transfer(%)

Ra Mesh	33× 33	65× 65	129× 129
10^3	0.040	0.009	0.006
10^4	0.165	0.133	0.132
10^5	0.630	0.500	0.475
10^6	0.316	0.223	0.210

In Figs. 6–9, simulated flow fields for different Rayleigh numbers are presented; Fig. 6 is the velocity vectors; Fig. 7 is the streamlines; Fig. 8 is the isothermal contours; and, Fig. 9 is the vorticity contours. All contour plots presented here are obtained by using the 129×129 mesh. Velocity vectors, Fig. 6, show that for large Rayleigh numbers, the flow motion is most vigorous in the vicinity of walls. In Fig. 7, the streamlines are obtained by plotting constant contours of the stream function ϕ. The distribution of ϕ is obtained by solving the following equations by the LSFEM.

$$\frac{\partial \phi}{\partial x} = -\frac{v}{T},$$

$$\frac{\partial \phi}{\partial y} = \frac{u}{T}.$$

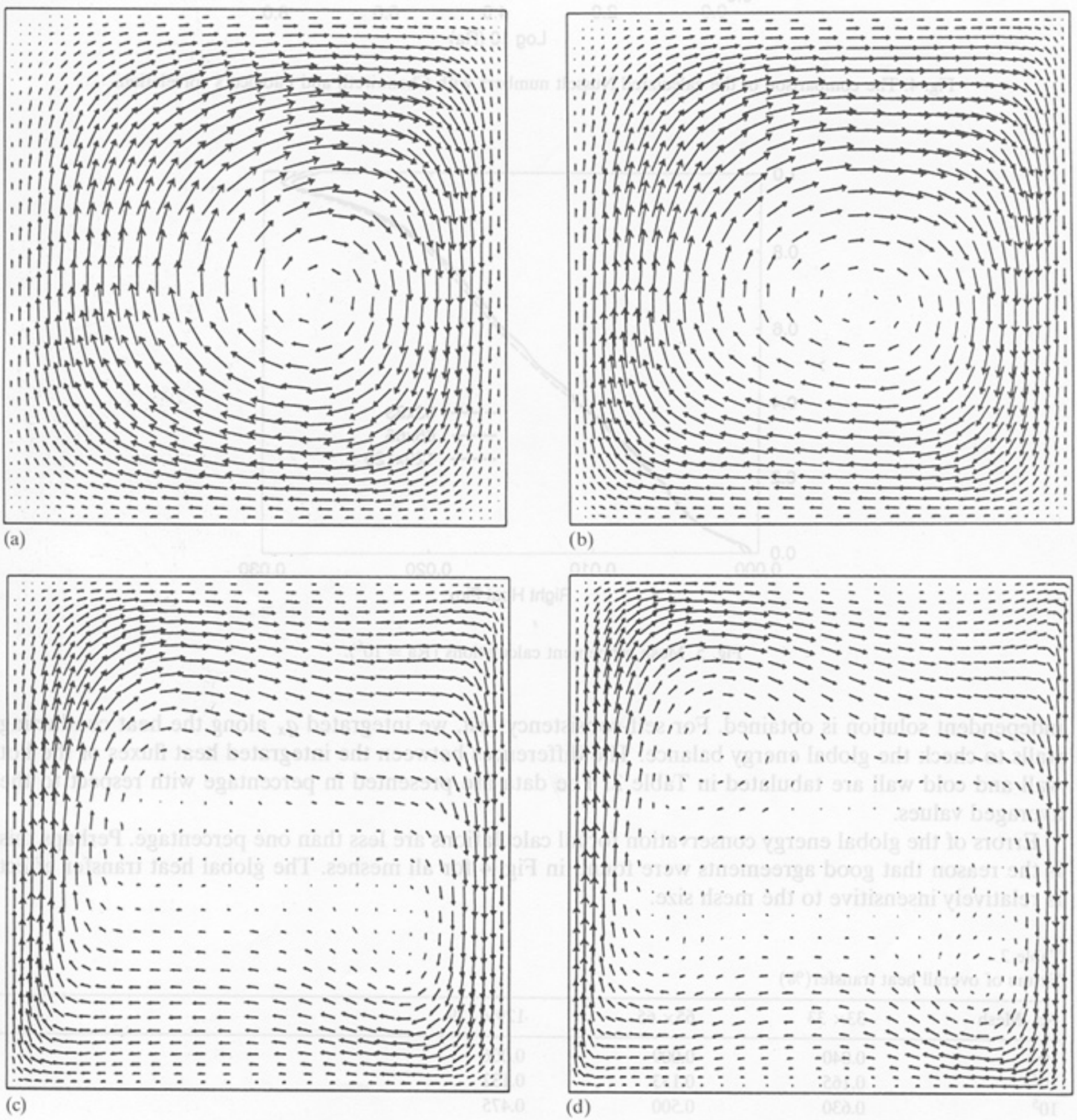

Fig. 6. Velocity vectors of the simulated flow fields: (a) $\mathrm{Ra} = 10^3$; (b) $\mathrm{Ra} = 10^4$; (c) $\mathrm{Ra} = 10^5$; (d) $\mathrm{Ra} = 10^6$.

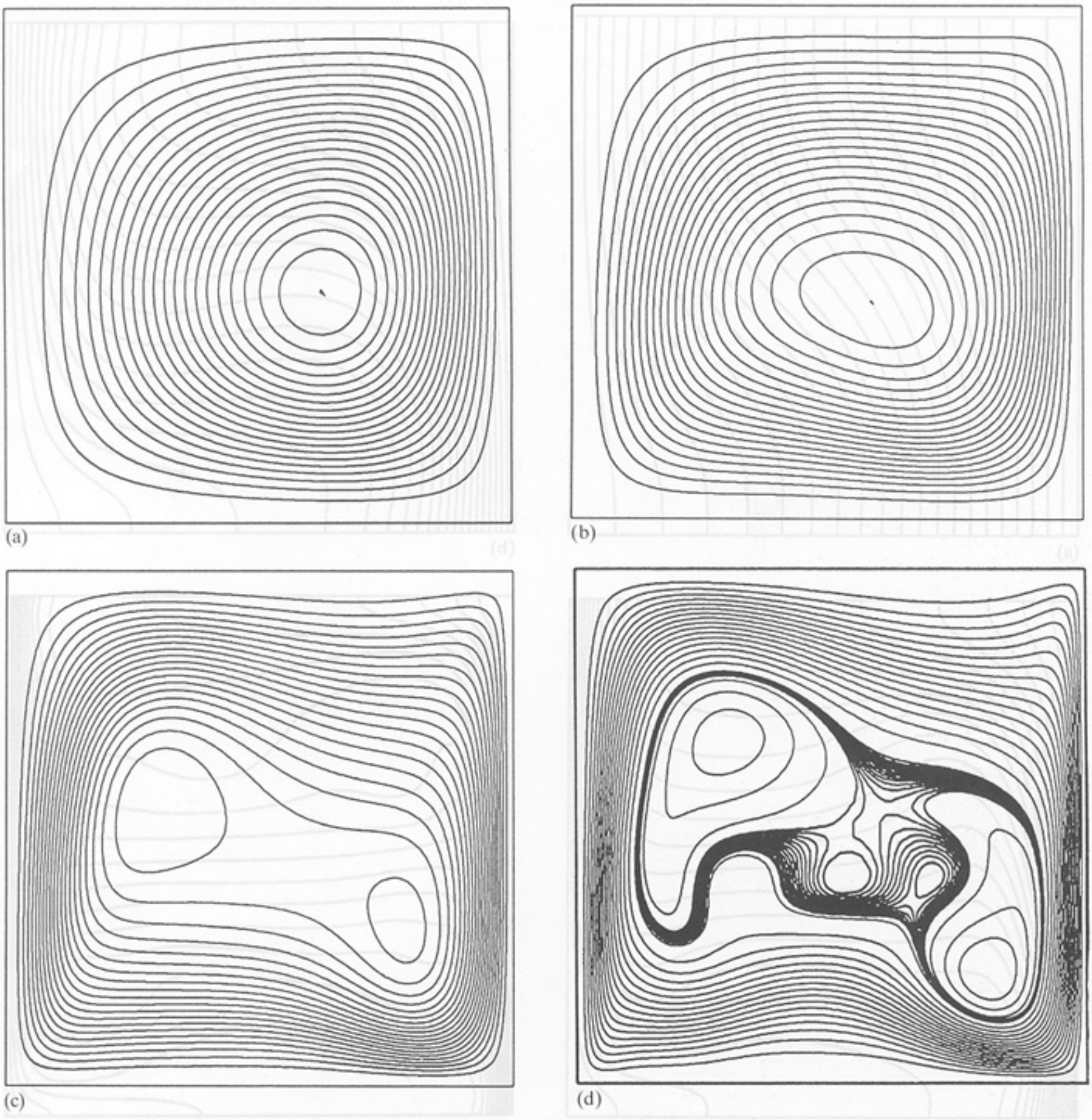

Fig. 7. Streamlines of the simulated flow fields: (a) $Ra = 10^3$; (b) $Ra = 10^4$; (c) $Ra = 10^5$; (d) $Ra = 10^6$.

Since this is a post-processing procedure, u, v and T are known throughout the flow field. To close the equation system, $\phi = 0$ along all walls is imposed.

It is well known that the Boussinesq approximation displays a fully antisymmetric flow field with respect to the center of the cavity. The present calculation based on the compressible formulation shows an asymmetric flow field which has been observed experimentally. For $Ra = 10^3$ and 10^4, a shift of the vortex center towards the cold wall is observed. At $Ra = 10^5$ and 10^6, secondary rolls embedded in the primary eddy are observed. Note that for $Ra = 10^6$, four secondary vortices are observed. Although not shown, it is interesting to note that only two embedded vortices were resolved if the 33×33 mesh was used. The result of the 65×65 mesh is about the same as that of the 129×129 mesh. No previously reported results showed such resolution. This is probably due to the merits of the LSFEM: (1) no artificial boundary condition for pressure is used; (2) no added artificial damping is employed; and (3) first-order

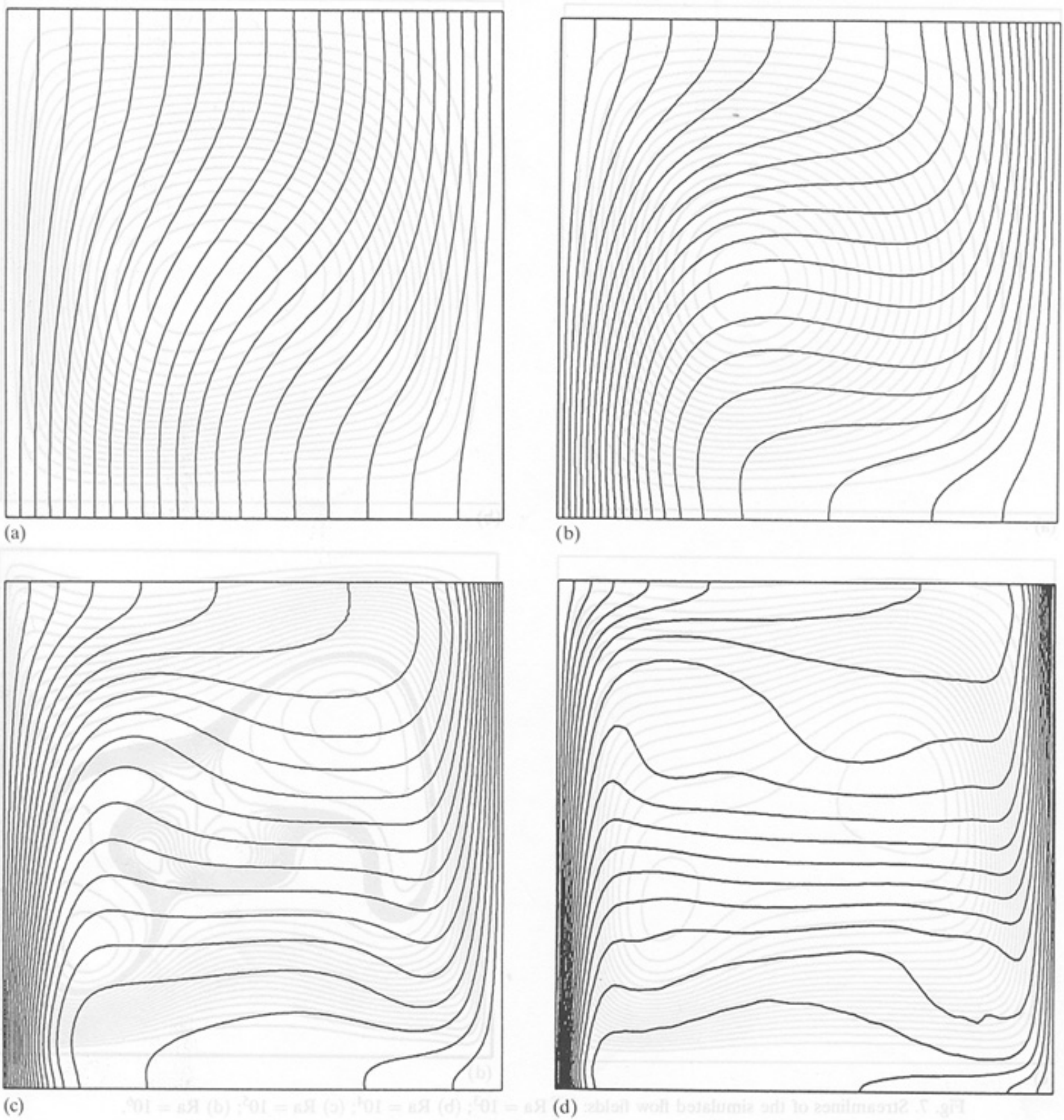

Fig. 8. Contours of constant temperature of the simulated flow fields: (a) $Ra = 10^3$; (b) $Ra = 10^4$; (c) $Ra = 10^5$; (d) $Ra = 10^6$.

derivative variables such as vorticity and heat fluxes are discretized in the same order of accuracy as velocities, pressure and temperature.

In Fig. 8, isothermal contours show steep temperature gradients near heat conducting walls. For the case of large Rayleigh number, the overall heat transfer is enhanced by fluid motion. In vorticity contours, Fig. 9, it is obvious that the distribution of vorticity profiles along walls is quite complex. In general, vorticity and its derivatives along walls cannot be predetermined. Fig. 10 shows the contours of the compressibility effect θ for the four cases. θ is of interest for physical interpretation and it can be calculated by using Eq. (32) in a post-processing procedure. Note that, θ is null along walls. That is, the flow is incompressible on walls even though the flow field inside the enclosure is compressible. This can be easily verified by inspecting Eq. (32), in which one can clearly see that the noslip condition results in the incompressible wall condition. For lower Rayleigh number case, the compressible region is

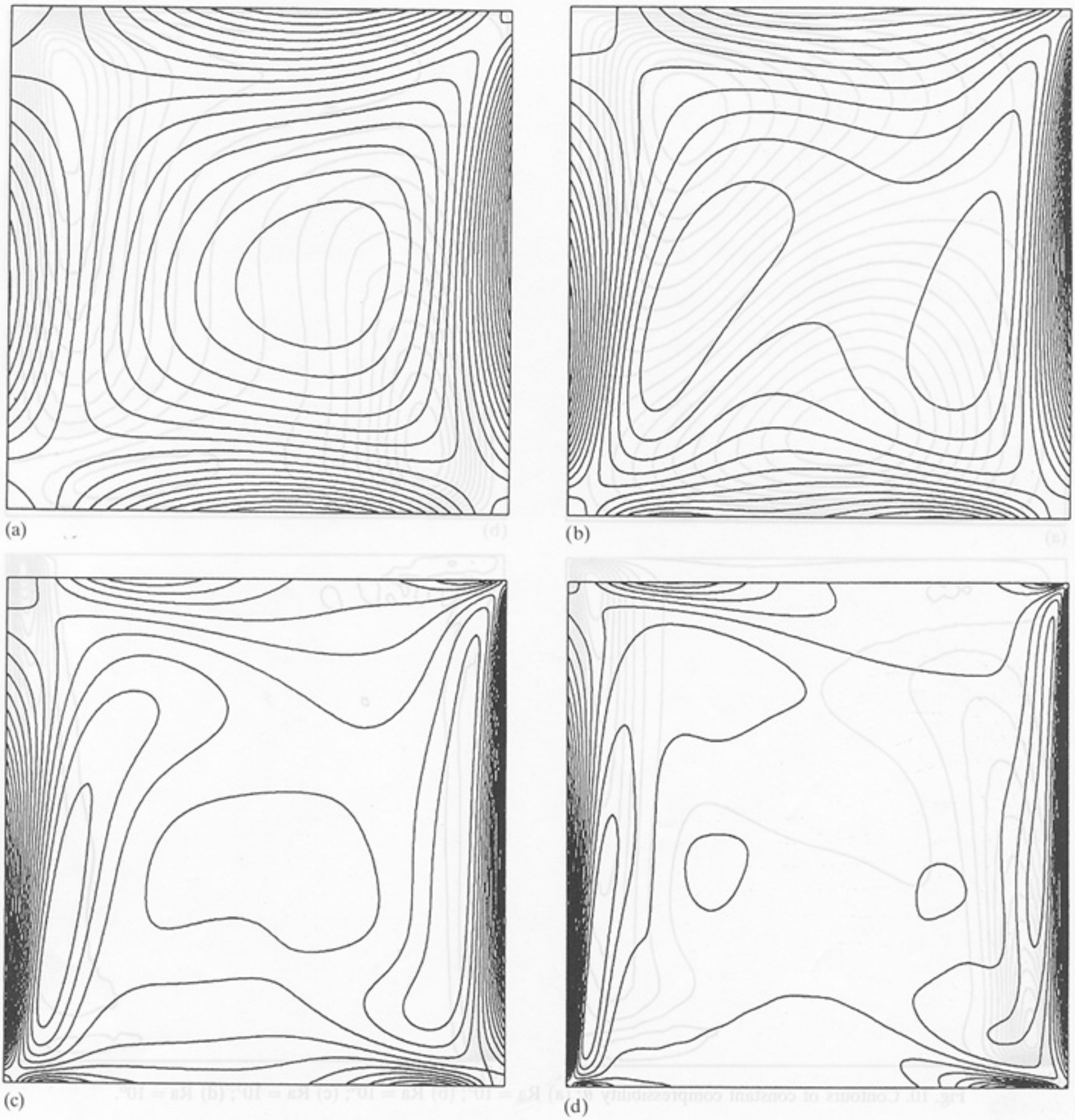

Fig. 9. Contours of constant vorticity of the simulated flow fields: (a) Ra = 10^3; (b) Ra = 10^4; (c) Ra = 10^5; (d) Ra = 10^6.

larger than that of higher Rayleigh number case, and the centers of the compressible regions are away from walls. For high Rayleigh number flow, the compressible region is restricted near the lower-left and upper-right corners with steep gradients to both vertical and horizontal walls. This is the reason for clustering the grid nodes near all four walls instead of just the two heat conducting walls.

For quantitative presentation, Figs. 11–16 show distributions of velocities, heat fluxes and temperature along the central lines and walls for flow fields of four different Rayleigh numbers. For high Rayleigh number flows, steep gradients developed near walls. In Figs. 13 and 14, for Rayleigh number 10^6 case, the distributions of heat fluxes along vertical walls have kinks near the lower-left and upper-right corners of the square enclosure. This phenomenon has also been observed for incompressible flow calculation using the Boussinesq model for buoyancy effect. The data presented in Figs. 11–16 are selectively tabulated in Tables 3–8 for easy access.

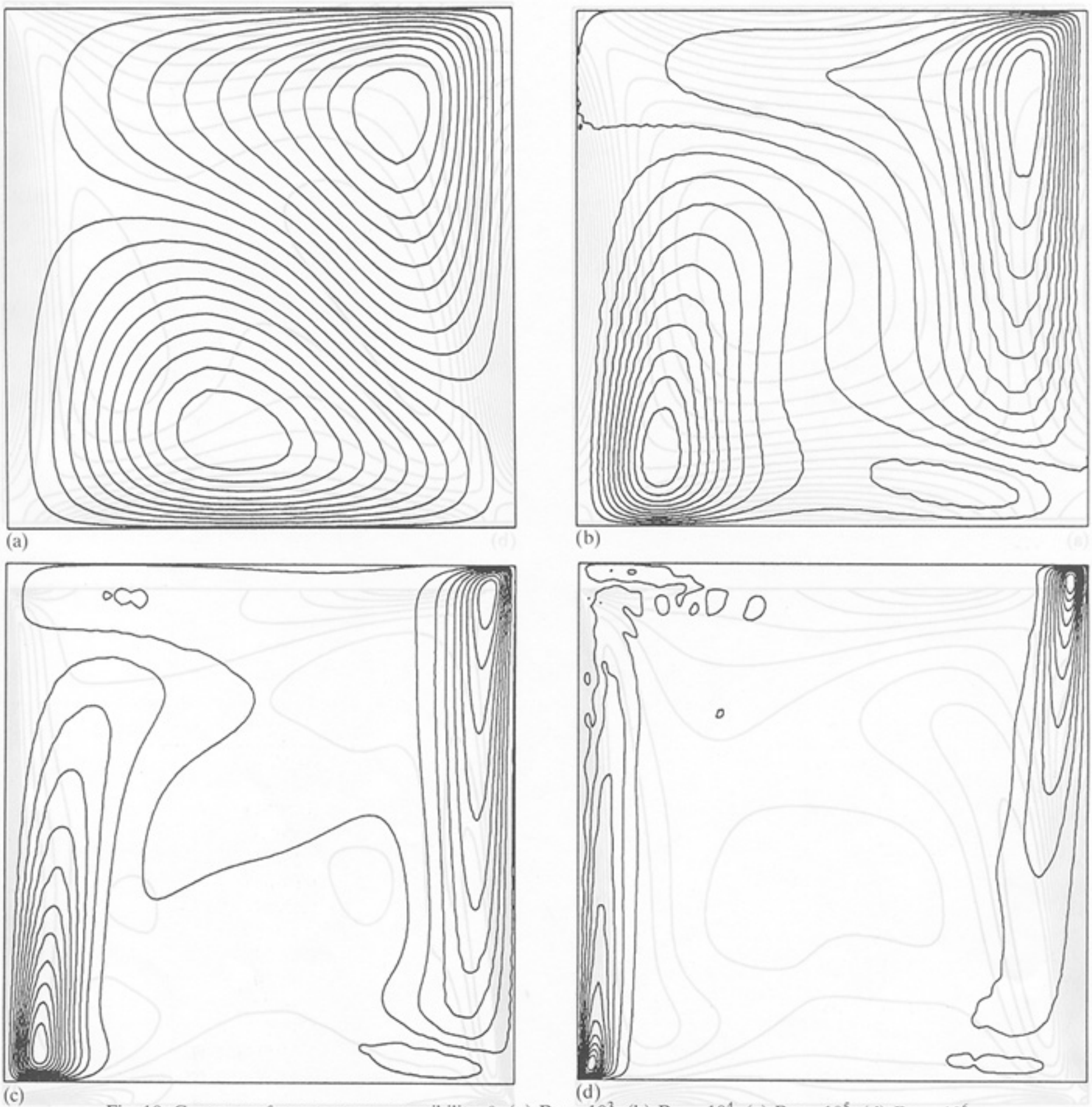

Fig. 10. Contours of constant compressibility θ: (a) Ra = 10^3; (b) Ra = 10^4; (c) Ra = 10^5; (d) Ra = 10^6.

Table 3
Velocity u profile along the vertical centerline

Y	Ra = 10^3	10^4	10^5	10^6
1.00000	0.00000	0.00000	0.00000	0.00000
0.98500	0.02775	0.03861	0.03306	0.02540
0.96863	0.05457	0.07587	0.06578	0.05365
0.95077	0.08000	0.11101	0.09709	0.08338
0.93129	0.10352	0.14318	0.12550	0.11186
0.91002	0.12456	0.17141	0.14914	0.13503
0.88682	0.14253	0.19469	0.16603	0.14852
0.86150	0.15683	0.21207	0.17434	0.14929
0.83387	0.16686	0.22264	0.17289	0.13704
0.80373	0.17205	0.22573	0.16152	0.11414
0.77083	0.17189	0.22085	0.14134	0.08478
0.73494	0.16593	0.20786	0.11476	0.05378
0.69577	0.15378	0.18694	0.08525	0.02614

Table 3 (*continued*)

Y	Ra = 10^3	10^4	10^5	10^6
0.65304	0.13511	0.15868	0.05687	0.00627
0.60641	0.10970	0.12416	0.03331	−0.00378
0.55552	0.07744	0.08496	0.01653	−0.00537
0.50000	0.03850	0.04282	0.00566	−0.00175
0.44448	−0.00262	0.00256	−0.00254	0.00174
0.39359	−0.04057	−0.03286	−0.01171	−0.00127
0.34696	−0.07404	−0.06459	−0.02351	−0.01332
0.30423	−0.10216	−0.09307	−0.03764	−0.03161
0.26506	−0.12434	−0.11785	−0.05296	−0.05005
0.22917	−0.14026	−0.13788	−0.06793	−0.06378
0.19627	−0.14980	−0.15199	−0.08084	−0.07073
0.16613	−0.15301	−0.15924	−0.09000	−0.07047
0.13850	−0.15010	−0.15915	−0.09413	−0.06385
0.11318	−0.14143	−0.15178	−0.09269	−0.05336
0.08998	−0.12747	−0.13766	−0.08590	−0.04201
0.06872	−0.10880	−0.11767	−0.07451	−0.03170
0.04923	−0.08605	−0.09284	−0.05943	−0.02300
0.03137	−0.05989	−0.06429	−0.04154	−0.01549
0.01500	−0.03099	−0.03304	−0.02154	−0.00818
0.00000	0.00000	0.00000	0.00000	0.00000

Table 4
Velocity v profile along the horizontal centerline

X	Ra = 10^3	10^4	10^5	10^6
0.00000	0.00000	0.00000	0.00000	0.00000
0.01500	0.02601	0.06269	0.12068	0.17838
0.03137	0.05124	0.12028	0.21629	0.28123
0.04923	0.07532	0.17100	0.28093	0.30510
0.06872	0.09783	0.21287	0.30981	0.26473
0.08998	0.11833	0.24388	0.30196	0.19169
0.11318	0.13637	0.26224	0.26251	0.11659
0.13850	0.15147	0.26682	0.20282	0.05580
0.16613	0.16316	0.25759	0.13748	0.01423
0.19627	0.17100	0.23596	0.07942	−0.00819
0.22917	0.17458	0.20487	0.03609	−0.01494
0.26506	0.17351	0.16838	0.00883	−0.01256
0.30423	0.16747	0.13083	−0.00529	−0.00756
0.34696	0.15612	0.09587	−0.01078	−0.00328
0.39359	0.13911	0.06570	−0.01174	−0.00111
0.44448	0.11595	0.04082	−0.01085	−0.00039
0.50000	0.08592	0.02014	−0.00926	−0.00041
0.55552	0.05131	0.00260	−0.00733	−0.00081
0.60641	0.01582	−0.01389	−0.00523	−0.00106
0.65304	−0.01949	−0.03201	−0.00320	0.00002
0.69577	−0.05350	−0.05347	−0.00199	0.00251
0.73494	−0.08490	−0.07909	−0.00315	0.00543
0.77083	−0.11234	−0.10874	−0.00913	0.00724
0.80373	−0.13450	−0.14117	−0.02300	0.00543
0.83387	−0.15019	−0.17382	−0.04755	−0.00160
0.86150	−0.15850	−0.20306	−0.08398	−0.01560
0.88682	−0.15884	−0.22458	−0.13032	−0.04115
0.91002	−0.15097	−0.23401	−0.17988	−0.08491
0.93129	−0.13498	−0.22749	−0.22100	−0.14941
0.95077	−0.11124	−0.20214	−0.23840	−0.22235
0.96863	−0.08033	−0.15617	−0.21622	−0.26583
0.98500	−0.04298	−0.08882	−0.14060	−0.21647
1.00000	0.00000	0.00000	0.00000	0.00000

Table 5
Temperature profile along the top insulated wall

X	Ra = 10^3	10^4	10^5	10^6
0.00000	1.60000	1.60000	1.60000	1.60000
0.01500	1.58663	1.58712	1.58234	1.56896
0.03137	1.57205	1.57311	1.56343	1.53724
0.04923	1.55617	1.55795	1.54352	1.50596
0.06872	1.53890	1.54164	1.52289	1.47563
0.08998	1.52012	1.52421	1.50172	1.44718
0.11318	1.49974	1.50568	1.48019	1.42082
0.13850	1.47763	1.48612	1.45845	1.39701
0.16613	1.45368	1.46558	1.43671	1.37556
0.19627	1.42774	1.44419	1.41531	1.35573
0.22917	1.39963	1.42207	1.39461	1.33868
0.26506	1.36914	1.39940	1.37486	1.32318
0.30423	1.33595	1.37631	1.35627	1.31007
0.34696	1.29963	1.35290	1.33890	1.29879
0.39359	1.25954	1.32912	1.32292	1.28912
0.44448	1.21470	1.30477	1.30842	1.28123
0.50000	1.16361	1.27918	1.29519	1.27437
0.55552	1.10910	1.25318	1.28371	1.26884
0.60641	1.05489	1.22696	1.27380	1.26407
0.65304	1.00054	1.19855	1.26450	1.25964
0.69577	0.94595	1.16614	1.25475	1.25521
0.73494	0.89125	1.12824	1.24322	1.25089
0.77083	0.83681	1.08375	1.22830	1.24615
0.80373	0.78310	1.03201	1.20789	1.24086
0.83387	0.73065	0.97300	1.17933	1.23495
0.86150	0.67998	0.90732	1.13946	1.22764
0.88682	0.63155	0.83616	1.08451	1.21587
0.91002	0.58571	0.76127	1.01131	1.19209
0.93128	0.54270	0.68466	0.91758	1.14328
0.95077	0.50264	0.60846	0.80368	1.04947
0.96863	0.46555	0.53460	0.67382	0.89032
0.98500	0.43138	0.46472	0.53567	0.66018
1.00000	0.40000	0.40000	0.40000	0.40000

Table 6
Temperature T profile along the bottom insulated wall

X	Ra = 10^3	10^4	10^5	10^6
0.00000	1.60000	1.60000	1.60000	1.60000
0.01500	1.57059	1.52996	1.43504	1.26622
0.03137	1.53851	1.45388	1.26157	0.96758
0.04923	1.50353	1.37187	1.08975	0.76437
0.06872	1.46541	1.28456	0.93391	0.65989
0.08998	1.42393	1.19329	0.80649	0.61718
0.11318	1.37885	1.10021	0.71404	0.60264
0.13850	1.33002	1.00824	0.65391	0.59821
0.16613	1.27733	0.92092	0.61775	0.59676
0.19627	1.22079	0.84188	0.59688	0.59641
0.22917	1.16058	0.77398	0.58449	0.59625
0.26506	1.09708	0.71861	0.57635	0.59711
0.30423	1.03092	0.67545	0.57005	0.59940
0.34696	0.96302	0.64284	0.56470	0.60292
0.39359	0.89457	0.61837	0.55926	0.60390
0.44448	0.82692	0.59941	0.55351	0.60131
0.50000	0.76146	0.58336	0.54732	0.59440
0.55552	0.70415	0.56924	0.54109	0.58681

Table 6 (*continued*)

X	$Ra = 10^3$	10^4	10^5	10^6
0.60641	0.65795	0.55650	0.53459	0.57887
0.65304	0.62010	0.54425	0.52789	0.57443
0.69577	0.58847	0.53213	0.52061	0.57179
0.73494	0.56153	0.52004	0.51266	0.57058
0.77083	0.53815	0.50797	0.50390	0.56871
0.80373	0.51754	0.49591	0.49457	0.56438
0.83387	0.49913	0.48397	0.48480	0.55730
0.86150	0.48252	0.47219	0.47463	0.54682
0.88682	0.46743	0.46065	0.46432	0.53415
0.91002	0.45365	0.44939	0.45414	0.51913
0.93128	0.44102	0.43850	0.44412	0.50371
0.95077	0.42942	0.42802	0.43381	0.48539
0.96863	0.41877	0.41805	0.42294	0.46410
0.98500	0.40898	0.40871	0.41170	0.43669
1.00000	0.40000	0.40000	0.40000	0.40000

Table 7
Heat conduction q_x distribution along the left hot wall

Y	$Ra = 10^3$	10^4	10^5	10^6
1.00000	0.03362	0.01039	0.00372	0.00150
0.98500	0.03363	0.01041	0.00375	0.00154
0.96863	0.03368	0.01048	0.00383	0.00167
0.95077	0.03376	0.01062	0.00400	0.00196
0.93128	0.03390	0.01085	0.00430	0.00240
0.91002	0.03411	0.01120	0.00477	0.00298
0.88682	0.03444	0.01173	0.00544	0.00363
0.86150	0.03491	0.01248	0.00631	0.00430
0.83387	0.03556	0.01350	0.00741	0.00498
0.80373	0.03647	0.01484	0.00869	0.00565
0.77083	0.03768	0.01658	0.01016	0.00633
0.73494	0.03926	0.01873	0.01179	0.00703
0.69577	0.04129	0.02135	0.01358	0.00777
0.65304	0.04383	0.02443	0.01552	0.00859
0.60641	0.04692	0.02800	0.01761	0.00955
0.55552	0.05057	0.03204	0.01987	0.01070
0.50000	0.05473	0.03652	0.02235	0.01206
0.44448	0.05890	0.04097	0.02489	0.01353
0.39359	0.06255	0.04493	0.02735	0.01497
0.34696	0.06562	0.04835	0.02974	0.01636
0.30423	0.06810	0.05121	0.03204	0.01766
0.26506	0.07004	0.05352	0.03423	0.01893
0.22917	0.07148	0.05529	0.03626	0.02017
0.19627	0.07250	0.05654	0.03809	0.02144
0.16613	0.07319	0.05731	0.03970	0.02271
0.13850	0.07361	0.05765	0.04103	0.02405
0.11318	0.07385	0.05765	0.04198	0.02544
0.08998	0.07396	0.05738	0.04252	0.02693
0.06872	0.07400	0.05695	0.04259	0.02837
0.04923	0.07399	0.05646	0.04220	0.02941
0.03137	0.07397	0.05603	0.04146	0.02957
0.01500	0.07395	0.05573	0.04066	0.02876
0.00000	0.07394	0.05563	0.04029	0.02797

Table 8
Heat conduction q_x distribution along the right cold wall

Y	$Ra = 10^3$	10^4	10^5	10^6
1.00000	0.07890	0.05130	0.03439	0.02670
0.98500	0.07896	0.05150	0.03486	0.02778
0.96863	0.07914	0.05198	0.03558	0.02826
0.95077	0.07939	0.05257	0.03612	0.02720
0.93128	0.07967	0.05315	0.03634	0.02555
0.91002	0.07992	0.05365	0.03625	0.02402
0.88682	0.08007	0.05399	0.03585	0.02280
0.86150	0.08005	0.05413	0.03521	0.02163
0.83387	0.07978	0.05402	0.03442	0.02073
0.80373	0.07916	0.05366	0.03353	0.01975
0.77083	0.07811	0.05298	0.03258	0.01900
0.73494	0.07652	0.05194	0.03153	0.01814
0.69577	0.07430	0.05048	0.03038	0.01743
0.65304	0.07135	0.04857	0.02906	0.01647
0.60641	0.06758	0.04616	0.02761	0.01566
0.55552	0.06294	0.04321	0.02596	0.01440
0.50000	0.05741	0.03971	0.02412	0.01338
0.44448	0.05158	0.03592	0.02223	0.01187
0.39359	0.04617	0.03222	0.02043	0.01085
0.34696	0.04131	0.02865	0.01876	0.00945
0.30423	0.03707	0.02523	0.01714	0.00861
0.26506	0.03348	0.02199	0.01554	0.00750
0.22917	0.03053	0.01897	0.01395	0.00698
0.19627	0.02819	0.01621	0.01237	0.00607
0.16613	0.02639	0.01375	0.01080	0.00554
0.13850	0.02505	0.01165	0.00927	0.00506
0.11318	0.02410	0.00996	0.00780	0.00456
0.08998	0.02345	0.00867	0.00644	0.00399
0.06872	0.02304	0.00775	0.00525	0.00329
0.04923	0.02280	0.00717	0.00433	0.00235
0.03137	0.02267	0.00683	0.00374	0.00136
0.01500	0.02261	0.00667	0.00344	0.00070
0.00000	0.02260	0.00663	0.00336	0.00054

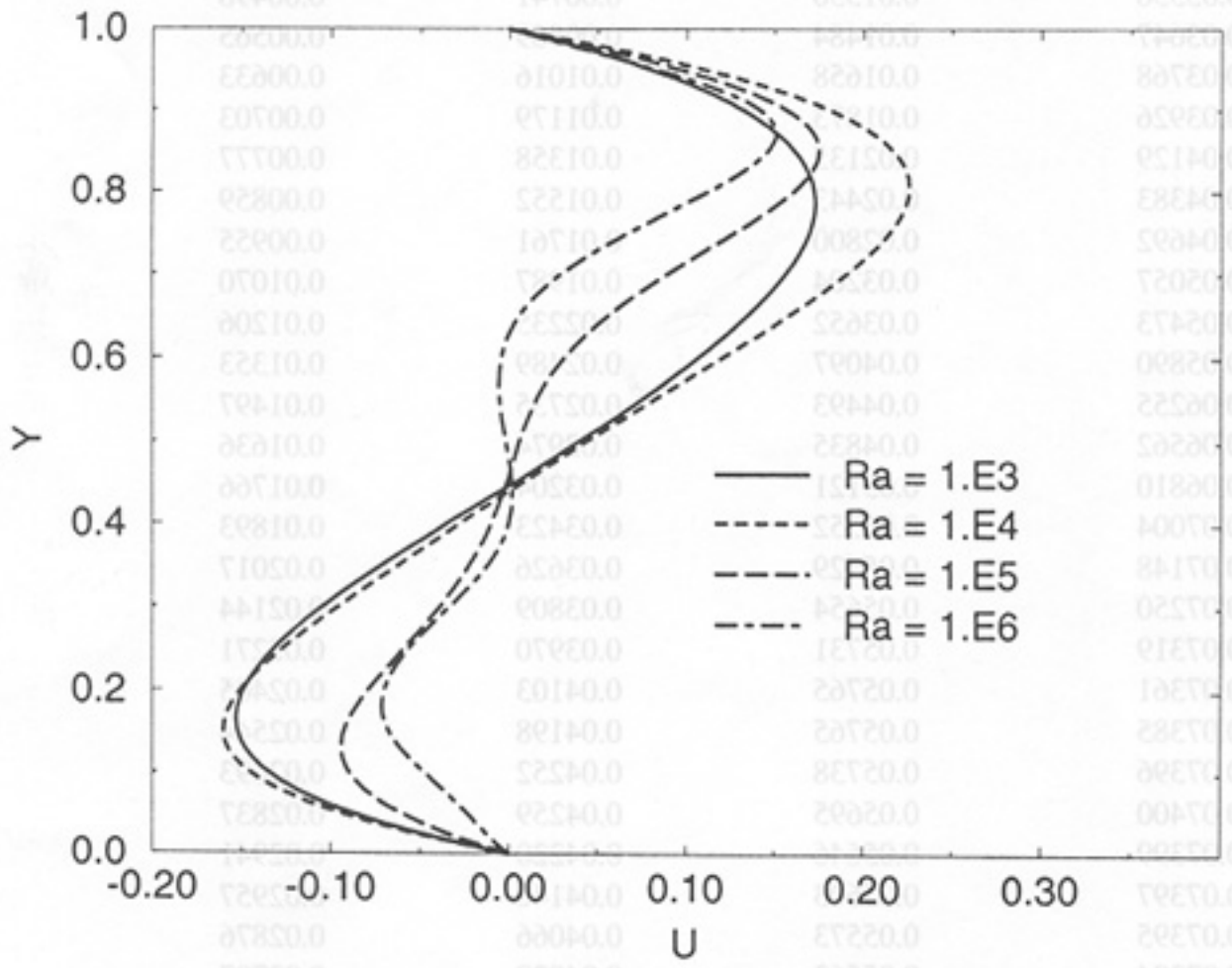

Fig. 11. Velocity u distribution along the vertical centerline.

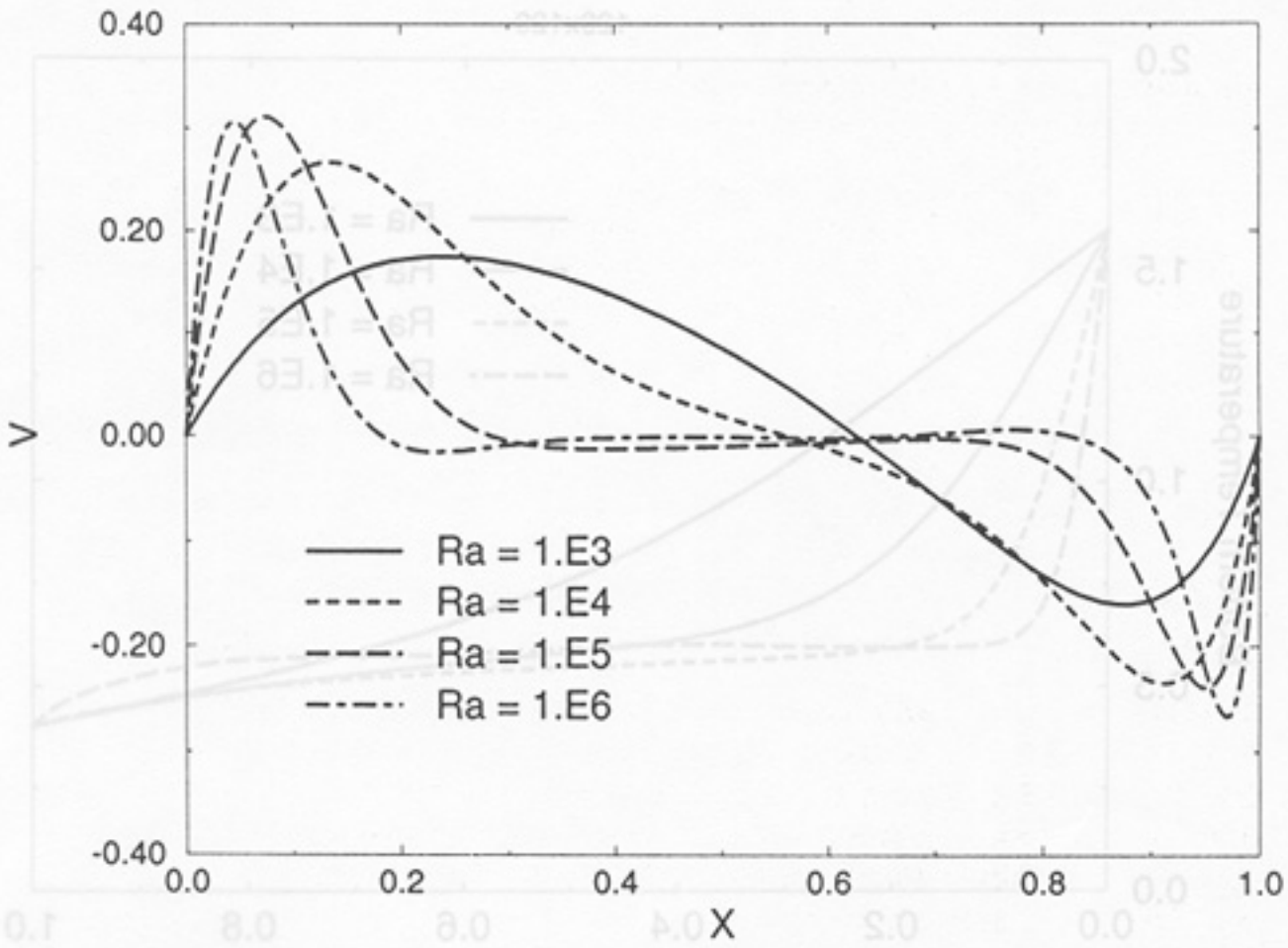

Fig. 12. Velocity v distribution along the horizontal centerline.

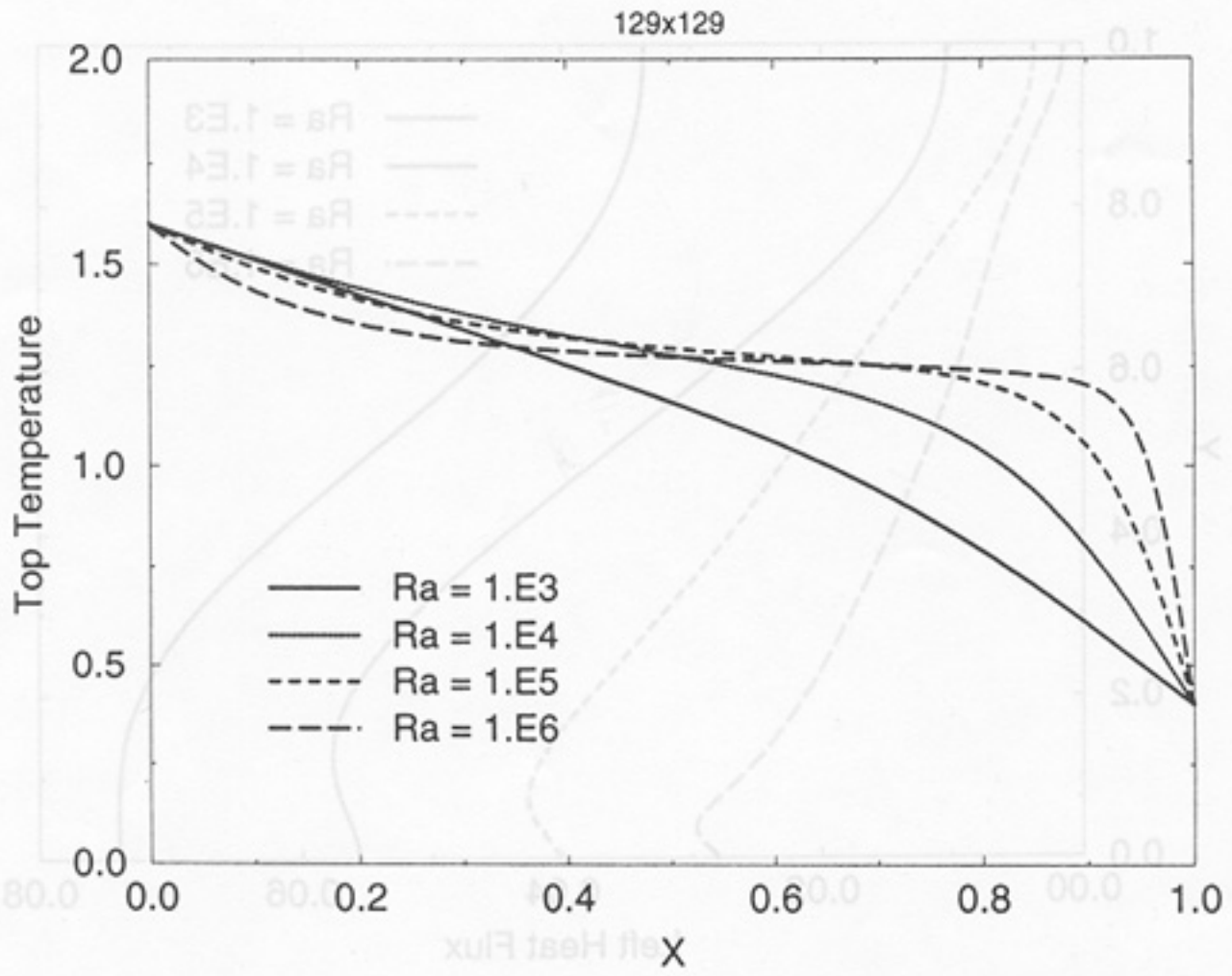

Fig. 13. Temperature T distribution along the top insulated wall.

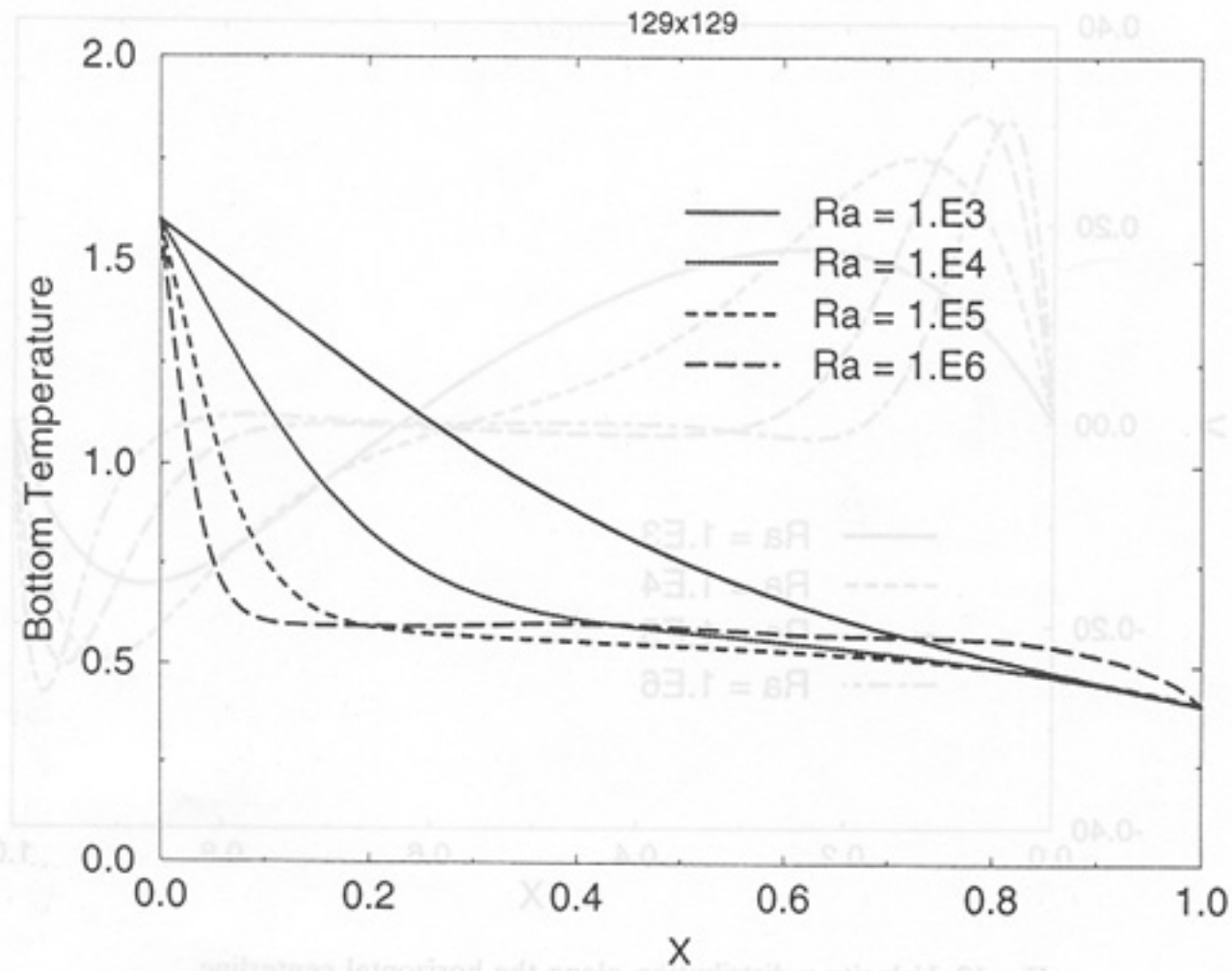

Fig. 14. Temperature T distribution along the lower insulated wall.

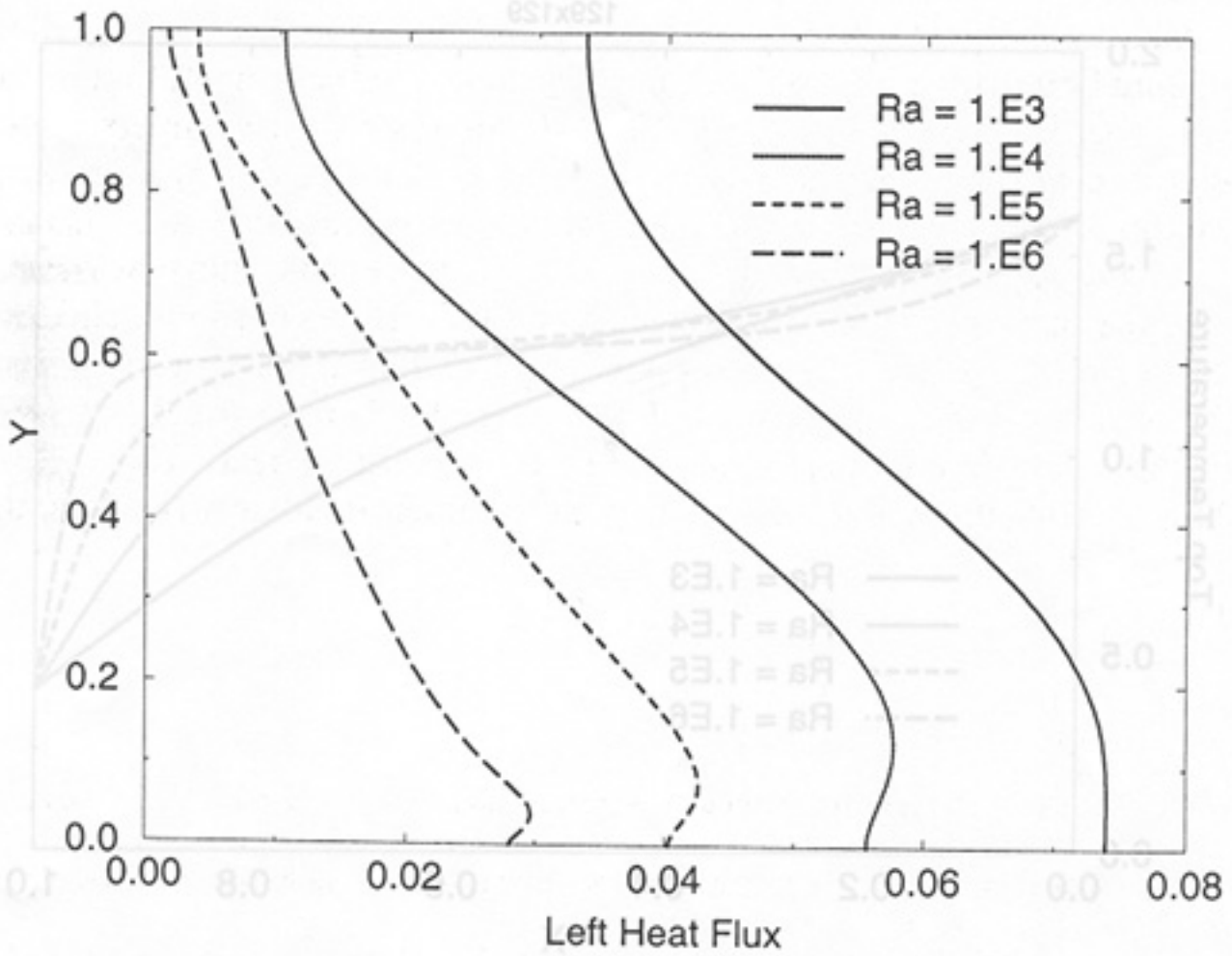

Fig. 15. Heat flux q_x distribution along the left hot wall.

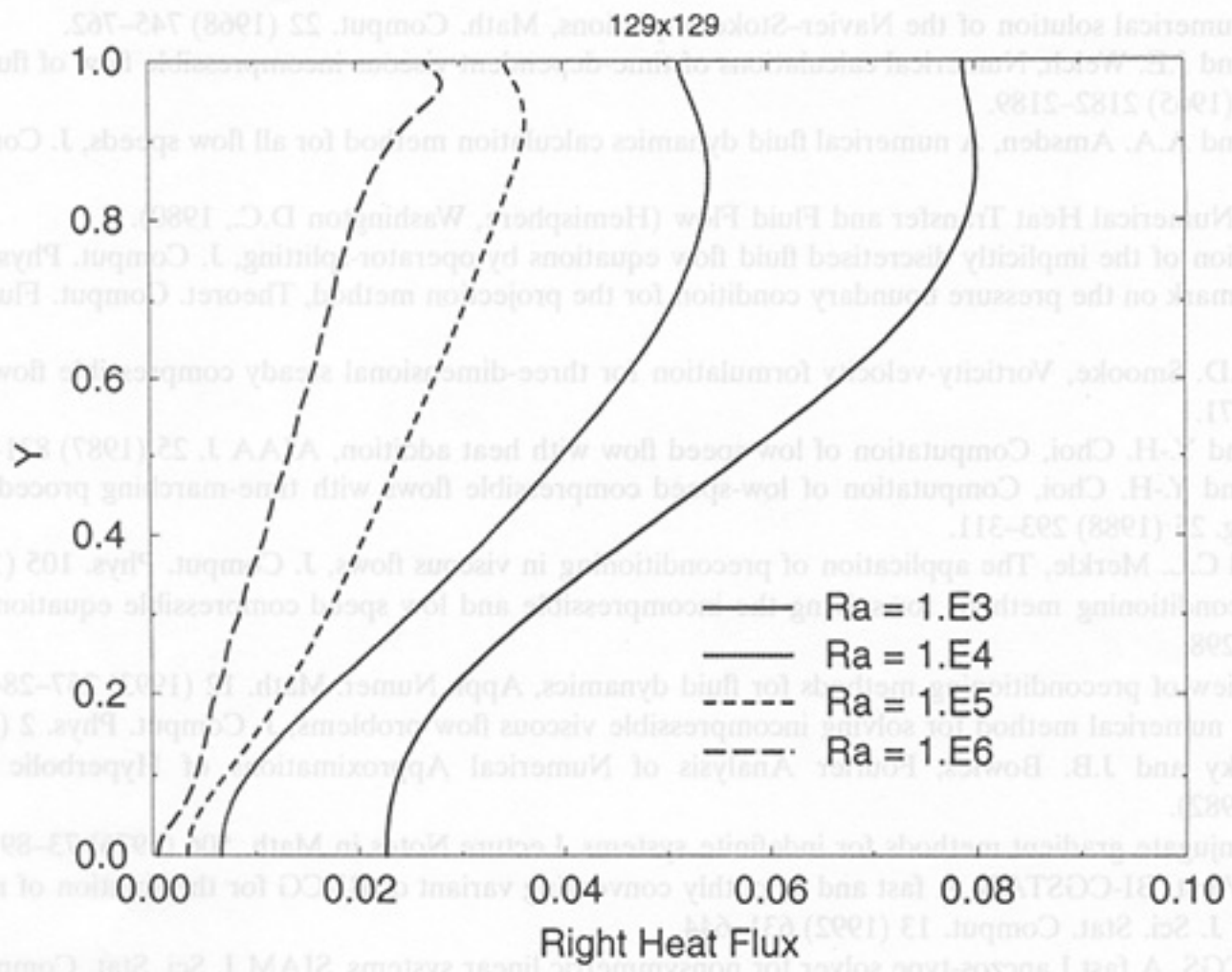

Fig. 16. Heat flux q_x distribution along the right cold wall.

5. Concluding remarks

In this paper, we have reported the development of a LSFEM for simulating compressible, viscous buoyant flows at low Mach numbers. In order to use the LSFEM. the flow equations are reduced to first order by introducing variables such as vorticity and heat conduction fluxes. In the vector analysis notation, the first-order equation set is cast to two div–curl and one div–curl–grad systems. The classification of the equations is facilitated by introducing a dummy variable and we proved that the system is elliptic. As such, the required boundary conditions become veritable and the formulation is suitable for LSFEM solution. Since the coefficient matrix of the LSFEM method is always symmetric and positive-definite, the inversion of the coefficient matrix was carried out by an element-by-element JCG method. As an numerical example, buoyancy-driven flows inside a square enclosure are calculated. The LSFEM crisply resolves the the asymmetric flow features of the compressible buoyant flows, which cannot be obtained b solving the incompressible flow equations with the Boussinesq model. For Rayleigh number one million, four secondary vortices were observed embedded in the the primary vortex. The accuracy of the simulated result was verified by comparing the calculated Nusselt number with Chenoweth's data, by a mesh refinement calculation and by the energy conservation check. The present result indicates that the LSFEM is a viable method for calculating multi-dimensional, low-Mach-number, compressible, buoyant flows.

References

[1] J.T. Oden, A short course of finite element methods for computational fluid dynamics, Lecture note, Knoxville, Tennessee, July, 1993.
[2] O.C. Zienkiewicz and J. Wu, Incompressibility without tears—how to avoid restrictions on mixed formulation, Int. J. Numer. Methods Engrg. 32 (1991) 1189–1203.
[3] J.T. Oden and G.F. Carey, Finite Elements, Mathematical Aspects, Vol. IV (Prentice-Hall, Englewood Cliffs, NJ, 1983).
[4] E.O. Einset and K.F. Jensen, A finite element solution of three-dimensional mixed convection gas flows in horizontal channels using preconditioned iterative matrix methods, Int. J. Numer. Methods Fluids 14 (1992) 817–841.

[5] A.J. Chorin, Numerical solution of the Navier–Stokes equations, Math. Comput. 22 (1968) 745–762.
[6] F. H. Harlow and J.E. Welch, Numerical calculations of time-dependent viscous incompressible flow of fluid with free surface, Phys. Fluids 8 (1965) 2182–2189.
[7] F. H. Harlow and A.A. Amsden, A numerical fluid dynamics calculation method for all flow speeds, J. Comput. Phys. 8 (1971) 197–213.
[8] S.V. Patankar, Numerical Heat Transfer and Fluid Flow (Hemisphere, Washington D.C., 1980).
[9] R.I. Issa, Solution of the implicitly discretised fluid flow equations by operator-splitting, J. Comput. Phys. 62 (1985) 40–65.
[10] R. Temam, Remark on the pressure boundary condition for the projection method, Theoret. Comput. Fluid Dynam. 3 (1991) 181–184.
[11] A. Ern and M.D. Smooke, Vorticity-velocity formulation for three-dimensional steady compressible flows, J. Comput. Phys. 105 (1993) 58–71.
[12] C.L. Merkle and Y.-H. Choi, Computation of low-speed flow with heat addition, AIAA J. 25 (1987) 831–838.
[13] C.L. Merkle and Y.-H. Choi, Computation of low-speed compressible flows with time-marching procedures, Int. J. Numer. Methods Engrg. 25 (1988) 293–311.
[14] Y.-H. Choi and C.L. Merkle, The application of preconditioning in viscous flows, J. Comput. Phys. 105 (1993) 207–223.
[15] E. Turkel, Preconditioning methods for solving the incompressible and low speed compressible equations, J. Comput. Phys. 72 (1987) 277–298.
[16] E. Turkel, Review of preconditioning methods for fluid dynamics, Appl. Numer. Math. 12 (1993) 257–284.
[17] A.J. Chorin, A numerical method for solving incompressible viscous flow problems, J. Comput. Phys. 2 (1967) 12–26.
[18] R. Vichnevetsky and J.B. Bowles, Fourier Analysis of Numerical Approximations of Hyperbolic Equations (SIAM, Philadelphia, 1982).
[19] R. Fletcher Conjugate gradient methods for indefinite systems, Lecture Notes in Math. 506 (1976) 73–89.
[20] H.A. van der Vorst, BI-CGSTAB: A fast and smoothly converging variant of BI-CG for the solution of nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 13 (1992) 631–644.
[21] P. Sonneveld, CGS, A fast Lanczos-type solver for nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 10 (1989) 36–52.
[22] D. Howard, W.M. Connolley and J.S. Rollett, Unsymmetric conjugate gradient methods and sparse direct methods in finite element flow simulation, Int. J. Numer. Methods Fluids 10 (1990) 925–945.
[23] Y. Saad, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 7 (1986) 856–869.
[24] P. Chin, E.F.D'Azevedo, P.A. Forsyth and W.P. Tang, Preconditioned conjugate gradient methods for the incompressible Navier–Stokes equations, Int. J. Numer. Methods Fluids 15 (1992) 273–295.
[25] J. Strigberger, G. Baruzzi, W. Habashi and M. Fortin, Some special purpose preconditioners for conjugate gradient-like methods applied to CFD, Int. J. Numer. Methods Fluids 16 (1993) 581–596.
[26] V. Venkatakrishnan and D.J. Mavriplis, Implicit solver for unstructured meshes, J. Comput. Phys. 105 (1993) 83–91.
[27] B.N. Jiang, T.L. Lin and L.A. Povinelli, Large-scale computation of incompressible viscous flow by least-squares finite element method, Comput. Methods Appl. Mech. Engrg. 114 (1994) 213–231.
[28] B.N. Jiang and L.A. Povinelli, Optimal least-squares finite element method for elliptic problems, Comput. Methods Appl. Mech. Engrg. 102 (1993) 199–212.
[29] B.N. Jiang and L.A. Povinelli, Least-squares finite element method for fluid dynamics, Comput. Methods Appl. Mech. Engrg. 81 (1990) 13–37.
[30] S.T. Yu, B.N. Jiang, N.S. Liu and J. Wu, Simulation of an H_2/O_2 flame by the LSFEM, Int. J. Numer. Methods Fluids 23 (1996) 65–74.
[31] S.T. Yu, B.N. Jiang, N.S. Liu and J. Wu, The least-squares finite-element method for low-mach-number compressible viscous flows, Int. J. Numer. Methods Engrg. 38 (1995) 3591–3610.
[32] R.G. Rehm and H.R. Baum, The equations of motions for thermally driven, buoyant flows, J. Res. Nat. Bureau of Standards 83 (1978) 297–308.
[33] A. Majda and J. Sethian, The derivation and numerical solution of the equations for zero mach number combustion, Combst. Sci. Tech. 42 (1985) 185–205.
[34] M. Hafez, M. Soliman and S. White, A unified approach for numerical solution of viscous compressible and incompressible flows over adiabatic and isothermal walls, Fifth Symposium on Numerical and Physical Aspects of Aerodynamic Flows, California State University, Long Beach, California, 1992.
[35] C.L. Chang and M.D. Gunzburger, A finite element method for first-order elliptic system in three dimensions, Appl. Math. Comput. 23 (1987) 171–184.
[36] C. Hirsch, Numerical computation of internal and external flows, Vol. 1: Fundamentals of Numerical Discretization, Wiley Series in Numerical Methods in Engineering (John Wiley and Sons, New York, 1988) 139–140.
[37] G.H. Golub and C.F. van Loan, Matrix Computations (The Johns Hopkins University Press, Baltimore, MD, 1983).
[38] D.R. Chenoweth and S. Paolucci, Natural convection in an enclosed vertical air layer with large horizontal temperature differences, J. Fluid Mech. 169 (1986) 173–210.