Available online at www.sciencedirect.com
SCIENCE DIRECT®

Journal of Computational and Applied Mathematics 195 (2006) 274–291

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

www.elsevier.com/locate/cam

Resolving small-scale structures in Boussinesq convection by adaptive grid methods

Zhengru Zhang[a,*], Tao Tang[b]

[a]School of Mathematical Sciences, Beijing Normal University, Beijing 100875, PR China
[b]Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong

Received 15 August 2004
Available online 15 August 2004

Abstract

Inviscid Boussinesq convection is a challenging problem both analytically and numerically. Due to the complex dynamic development of small scales and the rapid loss of solution regularity, the Boussinesq convection pushes any numerical strategy to the limit. In E and Shu (Phys. Fluids 6 (1994) 49), a detailed numerical study of the Boussinesq convection in the absence of viscous effects is carried out using filtered pseudospectral method and a high-order accurate ENO schemes. In their computations, very fine grids have to be used in order to resolve the small-structures of the Boussinesq fluid. In this work, we will develop an efficient adaptive grid method for solving the inviscid incompressible flows, which can be useful in resolving extremely small-structures with reasonably small number of grid points. To demonstrate the effectiveness of the proposed method, the Boussinesq convection problem will be computed using the adaptive grid method.

MSC: 65M06; 65M50; 35Q35; 76B70

Keywords: Adaptive mesh method; Finite volume method; Incompressible flow; Boussinesq convection

1. Introduction

There are several reasons for the study of two-dimensional Boussinesq convection. It is a simple model to address the open problem about whether finite time singularity occurs for initially smooth

* Corresponding author.
E-mail addresses: zrzhang@bnu.edu.cn (Z. Zhang), ttang@math.hkbu.edu.hk (T. Tang).

doi:10.1016/j.cam.2005.03.087

flows in inviscid and incompressible three-dimensional Euler flows. The governing equations of the Boussinesq convection are analogous to those of three-dimensional axis-symmetric Euler flow with swirl, see, e.g., [14,15]. The understanding of the finite-time singularities may be crucial to explain small scale structures in viscous turbulent flows. The Boussinesq convection has also potential relevance to the study of atmospheric and oceanography turbulence, as well as other astrophysical situations where rotation and stratification play a dominant role. As previous numerical studies have shown [6,10,11], the complex dynamics and the rapid formation of small scales make this problem an extremely demanding test for any numerical techniques.

Mesh adaptation can be in the form of local mesh refinement or through a mapping from a logical or computational domain to the physical domain. In the local mesh refinement method (see, e.g., [3]), the adaptive mesh is generated by adding or removing grid points to achieve a desired level of accuracy, which allows a systematic error analysis. However, the local mesh refinement approach requires complicated data structures and fairly technical methods to communicate information among different levels of refinement. In the mesh redistribution approach, the adaptive methods keep the total number of grid points unchanged, and can cluster more grid points to areas with singularities or large solution gradients, see, e.g., [1,5,23] and a recent review article [17]. The basic idea of moving mesh method is to construct a transformation from a logical domain (or called computational domain) to the physical domain. Recent applications of this approach include Hamilton–Jacobi equations [19], incompressible flow simulations [7], multiphase flow simulations [8,25], drop formation [20] and deformation [26].

In this work, we design an adaptive mesh redistribution scheme to compute accurately the Boussinesq convection problems. Our solution-adaptive mesh is obtained by solving a set of nonlinear elliptic PDEs for the mesh map. A conservative interpolation is used to obtain the approximate solution on the resulting solution-adaptive mesh. The given PDEs are advanced one time step based on a second-order finite-volume approach. In some of the previous studies, straightforward high-order approximations are used but the conservation property of the original problem may be lost. In our adaptive algorithm, both the mesh generation and the PDE evolution are solved in the computational domain, and the conservation can also be kept. An immediate advantage of doing computations in the computational domain is that some existing faster solver such as multigrid can be easily used.

This paper is organized as follows. In Section 2, we present the adaptive mesh algorithm for nonlinear hyperbolic conservation laws. In Section 3, we apply the proposed algorithm to the Boussinesq system. Numerical results will be presented and discussed. Some concluding remarks will be given in the final section.

2. Adaptive grid method

In [12,13], a transformation from a logical domain to the physical domain is constructed by using harmonic mappings, see also [9]. One of the primary features of the numerical scheme is that the mesh redistribution part and the PDE evolution part are separated. In [24], the principle ideas used in [12,13] were further developed by the authors to solve convection–diffusion problems with small viscosity. One of the main objectives of this work is to extend the numerical schemes proposed in [24] to solve the two-dimensional Boussinesq problem. Again, our adaptive grid algorithm is formed by two independent parts: a mesh re-distribution part and a PDE evolution part, which can be demonstrated by Table 1.

Table 1
Outline of the numerical algorithm

0. Determine the initial mesh based on the initial function
1. Determine Δt based on CFL-type condition so that $t^{n+1} = t^n + \Delta t$
2. Advance the solution one time step based on an appropriate numerical scheme
3. Grid Restructuring
  a. Solve the mesh redistributing equation (a generalized Laplacian equation) by one Gauss-Seidel iteration, to get $\mathbf{x}^{(k),n}$
  b. Interpolate the approximate solutions on the new grid $\mathbf{x}^{(k),n}$
  c. Calculate a weighted average of the locally calculated monitor at each computational cell and the surrounding monitor values
  d. Perform the iteration procedure (a.)–(c.) on grid-motion and solution-interpolation until there is no significant change in calculating new grids from one iteration to the next

Start new time step (go to **1** above)

2.1. Mesh redistribution

The solution-adaptive mesh is obtained through a bijective map from a logical or computational domain to the physical domain. A fixed uniform mesh is used in the logical domain. Denote by $x(\xi, \eta)$, $y(\xi, \eta)$ the mesh map in two dimensions, where (ξ, η) are the coordinates in the logical domain. In the variational approach, the adaptive mesh is to find the minimizer of the following functional:

$$E[\xi, \eta] = \frac{1}{2} \int_{\Omega_\mathrm{p}} [\nabla \xi^\mathrm{T} G_1^{-1} \nabla \xi + \nabla \eta^\mathrm{T} G_2^{-1} \nabla \eta] \, \mathrm{d}x \, \mathrm{d}y,$$

where G_1 and G_2 are given symmetric positive definite matrices called *monitor functions*. Ω_p is solution domain in the physical space. More terms can be added to the variational form to control the property of the adaptive mesh [4].

In this work, the adaptive mesh is determined by the corresponding Euler–Lagrange equations:

$$\nabla \cdot (G_1^{-1} \nabla \xi) = 0, \quad \nabla \cdot (G_2^{-1} \nabla \eta) = 0.$$

One of the simplest choices of the monitor functions is of scalar type: $G_1 = G_2 = \Omega I$, where I is the identity matrix and $\Omega > 0$ is called weight function. One typical choice of the weight function is $\Omega = \sqrt{1 + |\nabla u|^2}$, where u is a solution of the underlying PDEs. This choice of the monitor function corresponds to Winslow's variable diffusion method [21]:

$$\nabla \cdot \left(\frac{1}{\Omega} \nabla \xi \right) = 0, \quad \nabla \cdot \left(\frac{1}{\Omega} \nabla \eta \right) = 0. \tag{2.1}$$

The above system is defined in the physical domain Ω_p. In practice, Ω_p may have complex geometry, and as a result it is difficult to solve the elliptic system (2.1) directly. To overcome this difficulty, we make coordinate transformation $x = x(\xi, \eta)$ and $y = y(\xi, \eta)$ for (2.1) to obtain:

$$\frac{x_\xi}{J}\left[\left(x_\eta \frac{1}{J\Omega}x_\eta + y_\eta \frac{1}{J\Omega}y_\eta\right)_\xi - \left(x_\xi \frac{1}{J\Omega}x_\eta + y_\xi \frac{1}{J\Omega}y_\eta\right)_\eta\right]$$
$$+ \frac{x_\eta}{J}\left[-\left(x_\eta \frac{1}{J\Omega}x_\xi + y_\eta \frac{1}{J\Omega}y_\xi\right)_\xi + \left(x_\xi \frac{1}{J\Omega}x_\xi + y_\xi \frac{1}{J\Omega}y_\xi\right)_\eta\right] = 0,$$
$$\frac{y_\xi}{J}\left[\left(x_\eta \frac{1}{J\Omega}x_\eta + y_\eta \frac{1}{J\Omega}y_\eta\right)_\xi - \left(x_\xi \frac{1}{J\Omega}x_\eta + y_\xi \frac{1}{J\Omega}y_\eta\right)_\eta\right]$$
$$+ \frac{y_\eta}{J}\left[-\left(x_\eta \frac{1}{J\Omega}x_\xi + y_\eta \frac{1}{J\Omega}y_\xi\right)_\xi + \left(x_\xi \frac{1}{J\Omega}x_\xi + y_\xi \frac{1}{J\Omega}y_\xi\right)_\eta\right] = 0. \tag{2.2}$$

Note that system (2.2) is more complicated than the original Euler–Lagrange equation. An alternative approach, as proposed by Ceniceros and Hou [6], is to consider a functional defined in the computational domain directly:

$$\tilde{E}[x, y] = \frac{1}{2}\int_{\Omega_c} (\tilde{\nabla}^{\mathrm{T}} x G_1 \tilde{\nabla} x + \tilde{\nabla}^{\mathrm{T}} y G_2 \tilde{\nabla} y)\,\mathrm{d}\xi\,\mathrm{d}\eta, \tag{2.3}$$

where G_1, G_2 are again the monitor functions and $\tilde{\nabla} = (\partial_\xi, \partial_\eta)^{\mathrm{T}}$. The corresponding Euler–Lagrange equation is

$$(G_1 x_\xi)_\xi + (G_1 x_\eta)_\eta = 0,$$
$$(G_2 y_\xi)_\xi + (G_2 y_\eta)_\eta = 0. \tag{2.4}$$

In the following computation, we take the monitor function with the simplest form $G_1 = G_2 = \Omega I$. Then the Eq. (2.4) is reduced to

$$\tilde{\nabla} \cdot (\Omega \tilde{\nabla} x) = 0, \quad \tilde{\nabla} \cdot (\Omega \tilde{\nabla} y) = 0. \tag{2.5}$$

In our computation, we use Gauss–Seidel (GS) iteration to approximate the solution of the above system. The iteration is continued until there is no significant change in calculating new grids from one iteration to the next. In practice, a few iterations (say 3–5) are required at each time level, so the cost for generating new mesh is not too expensive. In order to obtain a smooth mesh distribution, we need to apply the following low-pass filter to the discrete monitor function:

$$\Omega_{j,k} \leftarrow \frac{4}{16}\Omega_{j,k} + \frac{2}{16}(\Omega_{j+1,k} + \Omega_{j-1,k} + \Omega_{j,k+1} + \Omega_{j,k-1})$$
$$+ \frac{1}{16}(\Omega_{j-1,k-1} + \Omega_{j-1,k+1} + \Omega_{j+1,k-1} + \Omega_{j+1,k+1}).$$

Usually, the above smooth technique is carried out 3–5 times at each GS iterative step.

After each GS iterative step, we need to pass the solution information from the old mesh $(x_{j,k}, y_{j,k})$ to the newly obtained mesh $(\tilde{x}_{j,k}, \tilde{y}_{j,k})$. This can be realized by using the conservative interpolation technique proposed by Tang and Tang [18]:

$$|\tilde{A}_{j+\frac{1}{2},k+\frac{1}{2}}|\tilde{u}_{j+\frac{1}{2},k+\frac{1}{2}} = |A_{j+\frac{1}{2},k+\frac{1}{2}}|u_{j+\frac{1}{2},k+\frac{1}{2}} - \left[(c^x u)_{j+1,k+\frac{1}{2}} - (c^x u)_{j,k+\frac{1}{2}}\right] - \left[(c^y u)_{j+\frac{1}{2},k+1} - (c^y u)_{j+\frac{1}{2},k}\right], \tag{2.6}$$

where $c^x_{j,k} = x_{j,k} - \tilde{x}_{j,k}$, $c^y_{j,k} = y_{j,k} - \tilde{y}_{j,k}$. The above formula is obtained using the classical perturbation theory. It is obvious that the discretization form (2.6) satisfies the mass–conservation in the following discrete sense:

$$\sum_{j,k} |\tilde{A}_{j+\frac{1}{2},k+\frac{1}{2}}|\tilde{u}_{j+\frac{1}{2},k+\frac{1}{2}} = \sum_{j,k} |A_{j+\frac{1}{2},k+\frac{1}{2}}|u_{j+\frac{1}{2},k+\frac{1}{2}},$$

where $|A_{j+\frac{1}{2},k+\frac{1}{2}}|$ and $|\tilde{A}_{j+\frac{1}{2},k+\frac{1}{2}}|$ means the areas of the corresponding control cells. Some theoretical properties of this conservative interpolation can be found in [18].

2.2. *Numerical solution to PDEs*

The governing equations of the Boussinesq equations can be written in conservative form. To demonstrate the principal ideas for the PDE evolution, let us consider the 2D conservation system in general form:

$$u_t + f(u)_x + g(u)_y = 0, \quad (x, y) \in \Omega_{\mathrm{p}}, \tag{2.7}$$

where Ω_{p} denotes the physical domain. To allow flexibility in handling complex geometry and in using fast solution solvers (such as multi-grid methods), we transform the underlying PDEs using the coordinate transformations $x = x(\xi, \eta)$, $y = y(\xi, \eta)$ and then solve the resulting equations in the computational domain equipped with a (fixed) uniform mesh. The cell-centered finite volume method will be employed to solve the transformed PDEs. Note that

$$u_x = \frac{1}{J}[(y_\eta u)_\xi - (y_\xi u)_\eta], \quad u_y = \frac{1}{J}[-(x_\eta u)_\xi + (x_\xi u)_\eta],$$

where $J = x_\xi y_\eta - x_\eta y_\xi$ is the Jacobian of the coordinate transformation. Using the above formulas, the (2.7) becomes

$$u_t + \frac{1}{J}(y_\eta f(u) - x_\eta g(u))_\xi + \frac{1}{J}(x_\xi g(u) - y_\xi f(u))_\eta = 0, \quad (\xi, \eta) \in \Omega_{\mathrm{c}}, \tag{2.8}$$

where Ω_{c} is the computational domain with a uniform grid (ξ_j, η_k). For convenience, we write the above equation in a simpler form:

$$u_t + \frac{1}{J}F(u)_\xi + \frac{1}{J}G(u)_\eta = 0. \tag{2.9}$$

The above system is again of the conservative form and can be solved using a finite-volume approach. Denote the control cell $[\xi_j, \xi_{j+1}] \times [\eta_k, \eta_{k+1}]$ by $A_{j+\frac{1}{2},k+\frac{1}{2}}$ and the cell average value by

$$\bar{u}^n_{j+\frac{1}{2},k+\frac{1}{2}} = \frac{1}{\Delta\xi\Delta\eta}\int_{A_{j+\frac{1}{2},k+\frac{1}{2}}} u(\xi,\eta,t^n)\,\mathrm{d}\xi\,\mathrm{d}\eta.$$

Note that

$$\begin{aligned}
&\frac{1}{\Delta\xi\Delta\eta}\int_{\xi_j}^{\xi_{j+1}}\int_{\eta_k}^{\eta_{k+1}} \frac{1}{J} w_\xi\,\mathrm{d}\xi\,\mathrm{d}\eta \\
&\quad = \frac{1}{\Delta\xi\Delta\eta}\frac{1}{J_{j+\frac{1}{2},k+\frac{1}{2}}}\int_{\xi_j}^{\xi_{j+1}}\int_{\eta_k}^{\eta_{k+1}} w_\xi\,\mathrm{d}\xi\,\mathrm{d}\eta + \mathcal{O}(\Delta\xi^2) \\
&\quad = \frac{1}{J_{j+\frac{1}{2},k+\frac{1}{2}}}\left(\frac{w_{j+1,k+\frac{1}{2}} - w_{j,k+\frac{1}{2}}}{\Delta\xi}\right) + \mathcal{O}(\Delta\xi^2),
\end{aligned}$$

where a mid-point rule is used in the first step. Similar approach can be used to treat the term involving $J^{-1}w_\eta$. Integrating Eq. (2.9) over the cell $[t^n, t^{n+1}] \times A_{j+\frac{1}{2},k+\frac{1}{2}}$ in the computational domain leads to

$$\bar{u}^{n+1}_{j+\frac{1}{2},k+\frac{1}{2}} = \bar{u}^n_{j+\frac{1}{2},k+\frac{1}{2}} - \frac{\Delta t^n}{J_{j+\frac{1}{2},k+\frac{1}{2}}}\left(\frac{\bar{F}^n_{j+1,k+\frac{1}{2}} - \bar{F}^n_{j,k+\frac{1}{2}}}{\Delta\xi} + \frac{\bar{G}^n_{j+\frac{1}{2},k+1} - \bar{G}^n_{j+\frac{1}{2},k}}{\Delta\eta}\right). \tag{2.10}$$

The one-dimensional Lax–Friedrichs numerical flux

$$\bar{f}(a,b) = \tfrac{1}{2}[f(a) + f(b) - \max|f'(u)|(b-a)]$$

will be applied to $\bar{F}$, $\bar{G}$ respectively:

$$\bar{F}_{j,k+\frac{1}{2}} = \bar{F}\left(u^-_{j,k+\frac{1}{2}}, u^+_{j,k+\frac{1}{2}}\right), \quad \bar{G}_{j+\frac{1}{2},k} = \bar{G}\left(u^-_{j+\frac{1}{2},k}, u^+_{j+\frac{1}{2},k}\right). \tag{2.11}$$

In order to compute (2.11), we construct a piecewise linear approximation as follows:

$$u^-_{j,k+\frac{1}{2}} = \bar{u}_{j-\frac{1}{2},k+\frac{1}{2}} + \frac{\Delta\xi}{2}s_{j-\frac{1}{2},k+\frac{1}{2}}, \quad u^+_{j,k+\frac{1}{2}} = \bar{u}_{j+\frac{1}{2},k+\frac{1}{2}} - \frac{\Delta\xi}{2}s_{j+\frac{1}{2},k+\frac{1}{2}}$$

$$s_{j+\frac{1}{2},k+\frac{1}{2}} = \left(\mathrm{sign}\left(s^-_{j+\frac{1}{2},k+\frac{1}{2}}\right) + \mathrm{sign}\left(s^+_{j+\frac{1}{2},k+\frac{1}{2}}\right)\right)\frac{|s^+_{j+\frac{1}{2},k+\frac{1}{2}}s^-_{j+\frac{1}{2},k+\frac{1}{2}}|}{|s^+_{j+\frac{1}{2},k+\frac{1}{2}}| + |s^-_{j+\frac{1}{2},k+\frac{1}{2}}|}$$

$$s^-_{j+\frac{1}{2},k+\frac{1}{2}} = \frac{\bar{u}_{j+\frac{1}{2},k+\frac{1}{2}} - \bar{u}_{j-\frac{1}{2},k+\frac{1}{2}}}{\Delta\xi}, \quad s^+_{j+\frac{1}{2},k+\frac{1}{2}} = \frac{\bar{u}_{j+\frac{3}{2},k+\frac{1}{2}} - \bar{u}_{j+\frac{1}{2},k+\frac{1}{2}}}{\Delta\xi}.$$

A semi-discretized difference equations can be obtained from the fully discretized (2.10), which will be solved by a 3-stage Runge–Kutta method proposed by Shu and Osher [16]. More precisely, for the ODE system $u'(t) = L(u)$ we use

$$u_{jk}^{(1)} = u_{jk}^{n} + \Delta t L(u_{jk}^{n}),$$

$$u_{jk}^{(2)} = \frac{3}{4}u_{jk}^{n} + \frac{1}{4}\left[u_{jk}^{(1)} + \Delta t L\left(u_{jk}^{(1)}\right)\right],$$

$$u_{jk}^{n+1} = \frac{1}{3}u_{jk}^{n} + \frac{2}{3}\left[u_{jk}^{(2)} + \Delta t L\left(u_{jk}^{(2)}\right)\right].$$

The above ODE solver satisfies the total variation non-increasing property.

3. E and Shu's problem revisit

E and Shu [10] studied the small-scale structures in two-dimensional Boussinesq convection in the absence of viscous effects. The governing equations of continuity and motion for an incompressible, inviscid fluid in the presence of gravity are, respectively,

$$\rho_t + \mathbf{u} \cdot \nabla\rho = 0, \tag{3.1}$$

$$\mathbf{u}_t + \mathbf{u} \cdot \nabla\mathbf{u} = -\nabla p + \rho g\mathbf{j}, \tag{3.2}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{3.3}$$

where p is the pressure, ρ is the density (usually this should be the temperature and denoted by T or θ, but we are accustomed to call it density, and therefore denote it by ρ), $\mathbf{u} = (u, v)$ is the velocity, g is the gravitational constant, and $\mathbf{j}$ is the unit vector in the upward vertical direction.

In two-dimensional, the above system of equations can be re-written using the stream function-vorticity formulation. Let $\omega = v_x - u_y$ be the vorticity. The velocity $\mathbf{u} = (u, v)$ is determined by the stream function ψ:

$$u = \psi_y, \quad v = -\psi_x. \tag{3.4}$$

It is convenient to re-write (3.1)–(3.3) in the stream-function vorticity formulation:

$$\rho_t + \mathbf{u} \cdot \nabla\rho = 0, \tag{3.5}$$

$$\omega_t + \mathbf{u} \cdot \nabla\omega = -\rho_x, \tag{3.6}$$

$$-\Delta\psi = \omega. \tag{3.7}$$

Several existing studies show that this problem is extremely difficult both numerically and analytically. Although short time existence can be shown for sufficiently smooth initial conditions, it is unclear if a solution can lose regularity and become singular in finite time. Following Beale et al. [2], E and Shu [10] proved that if a singularity develops in the Boussinesq flow at a finite time T^*, such as

$\|\mathbf{u}(\cdot, T^*)\|_m + \|\rho(\cdot, T^*)\|_m = +\infty$, then

$$\int_0^{T^*} |\omega(\cdot, t)|_\infty \, dt = +\infty, \qquad \int_0^{T^*}\int_0^t |\rho_x(\cdot, s)|_\infty \, ds \, dt = +\infty, \tag{3.8}$$

where $\|f(\cdot)\|_m$ denotes the m-norm in Sobolev space and $|f(\cdot)|_\infty = \max_{(x,y)\in R^2}|f(x, y)|$. It is assumed that $m > 2$ and that the initial conditions for $\mathbf{u}$ and ρ lie in $H^m(R^2)$. In particular, E and Shu provide the minimum rate of self-similar blow-up if a singularity develops at T^*:

$$|\omega(\cdot, t)|_\infty \sim \frac{c_1}{T^* - t}, \quad |\rho_x(\cdot, t)|_\infty \sim \frac{c_2}{(T^* - t)^2}. \tag{3.9}$$

In [10], the gravitational constant g is normalized to be 1 and the following initial boundary conditions are considered:

$$\omega(x, y, 0) = 0, \quad \rho(x, y, 0) = 50\rho_1(x, y)\rho_2(x, y)[1 - \rho_1(x, y)], \tag{3.10}$$

where

$$\rho_1(x, y) = \begin{cases} \exp\left(1 - \dfrac{\pi^2}{\pi^2 - x^2 - (y-\pi)^2}\right) & x^2 + (y-\pi)^2 \leqslant \pi^2 \\ 0 & \text{otherwise,} \end{cases} \tag{3.11}$$

$$\rho_2(x, y) = \begin{cases} \exp\left(1 - \dfrac{(1.95\pi)^2}{(1.95\pi)^2 - (x-2\pi)^2}\right) & |x - 2\pi| < 1.95\pi \\ 0 & \text{otherwise.} \end{cases} \tag{3.12}$$

The initial density contour and the corresponding adaptive mesh are plotted in Fig. 1. Obviously, the given data are smooth. The initial adaptive grid is obtained by using 100×100 mesh.

In this work, we will apply the adaptive mesh method to the system (3.5)–(3.7). All the three equations are transformed to the computational domain on which the resulting system is solved with a uniform mesh. The density ρ, vorticity ω and stream-function ψ are defined at the cell center, while the velocity $\mathbf{u}$ is defined at the cell edges. The discretization for the stream-function Eq. (3.7) leads to a large sparse positive definite linear system. We use a multigrid preconditioned conjugate gradient (MGPCG) iteration method to calculate the stream-function ψ, and compute the velocity $\mathbf{u}$ using the Eq. (3.4) with a central difference approach. Both the mesh equation and the Boussinesq equations (3.5)–(3.7) are discretized in the logical domain which is equipped with a (fixed) uniform mesh.

3.1. Evolution of the bubble

In [10], E and Shu employed two high-order numerical methods, namely Fourier-collocation spectral method and ENO3 method, to solve the Boussinesq system (3.5)–(3.7) with initial values (3.10)–(3.12) on a uniform mesh. The small structures are resolved with very fine meshes, namely 1500^2 mesh points for the spectral method and 512^2 points for the ENO3 method. In this subsection, we will present some adaptive results, hoping to resolve the small structures of the Boussinesq flow with less mesh points. The monitor function used for (2.5) is $\Omega = \sqrt{1 + 0.2|\tilde{\nabla}\rho|^2}$.

In Fig. 2, the density and vorticity contours at $t = 1.6$ are plotted using 300^2 and 400^2 mesh points, respectively. At this time, the flow looks like a rising bubble. It is observed that there is almost no

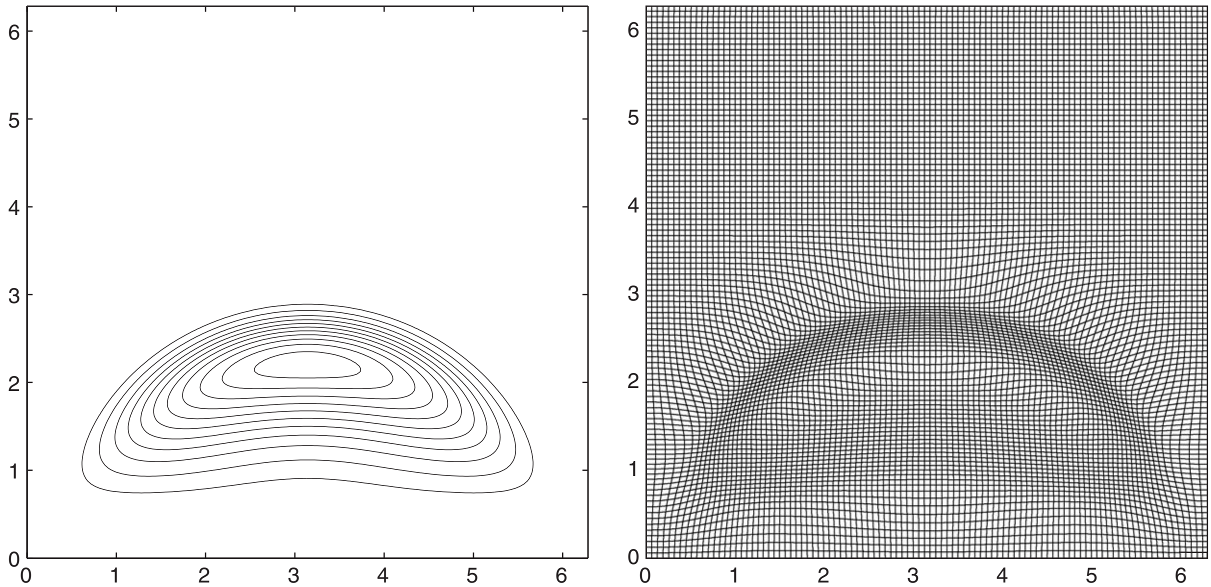

Fig. 1. Initial density contours (left) and adaptive mesh redistribution $N_x = N_y = 100$.

difference between the numerical solutions obtained by using 300^2 and 400^2 mesh points, implying that it is sufficient to resolve the flow at $t = 1.6$ with the 300^2 mesh points.

The solution contours at $t = 3$ is plotted in Figs. 3 and 4. By now the outer boundary of the bubble has become a sharp front, and the flow becomes more complicated and many small scale structures are formed. It is observed in [10] that as the bubble rises, it leaves behind a long and thin filament of light fluid. They claimed that the thin filament is a check on the amount of numerical diffusion present in a numerical scheme: a low-order method with numerical viscosity will destroy the thin filament. It is seen from Fig. 3 that the thin filament is obtained by using 300^2 grid. In fact, the numerical results with finer grids are graphically indistinguishable with those presented in [10] which were computed on a 1500^2 grid.

It is observed in Fig. 5 that the maximum and minimum density are almost preserved, which is in good agreement with the physical property. Relatively speaking, the maximum of density is not perfectly satisfactory, which is caused by the numerical scheme. Our algorithm is only second order accurate, thus numerical viscosity is introduced, and consequently sharpness is slightly smeared.

In Figs. 6 and 7, we plot the solution at some selected cut-lines in order to see the detailed small structures. Clearly, at $t = 2.5$, both the density and the vorticity are very steep in some local regions. Also it is seen that the density is axis-symmetric and the vorticity is anti-axis symmetric. No oscillation is caused by the numerical scheme. It is seen from Fig. 7, the density along the symmetry axis becomes sharper, while the maximum is preserved. The density reaches its maximum at the top of the cap. The evolution of the density along the symmetry axis is similar to the formation of a one-dimensional shock out of smooth initial data.

In Figs. 8 and 9, we plot the solution at $t = 3.16$. At this time, the flow becomes even more complicated, and many sharp small structures are developed.

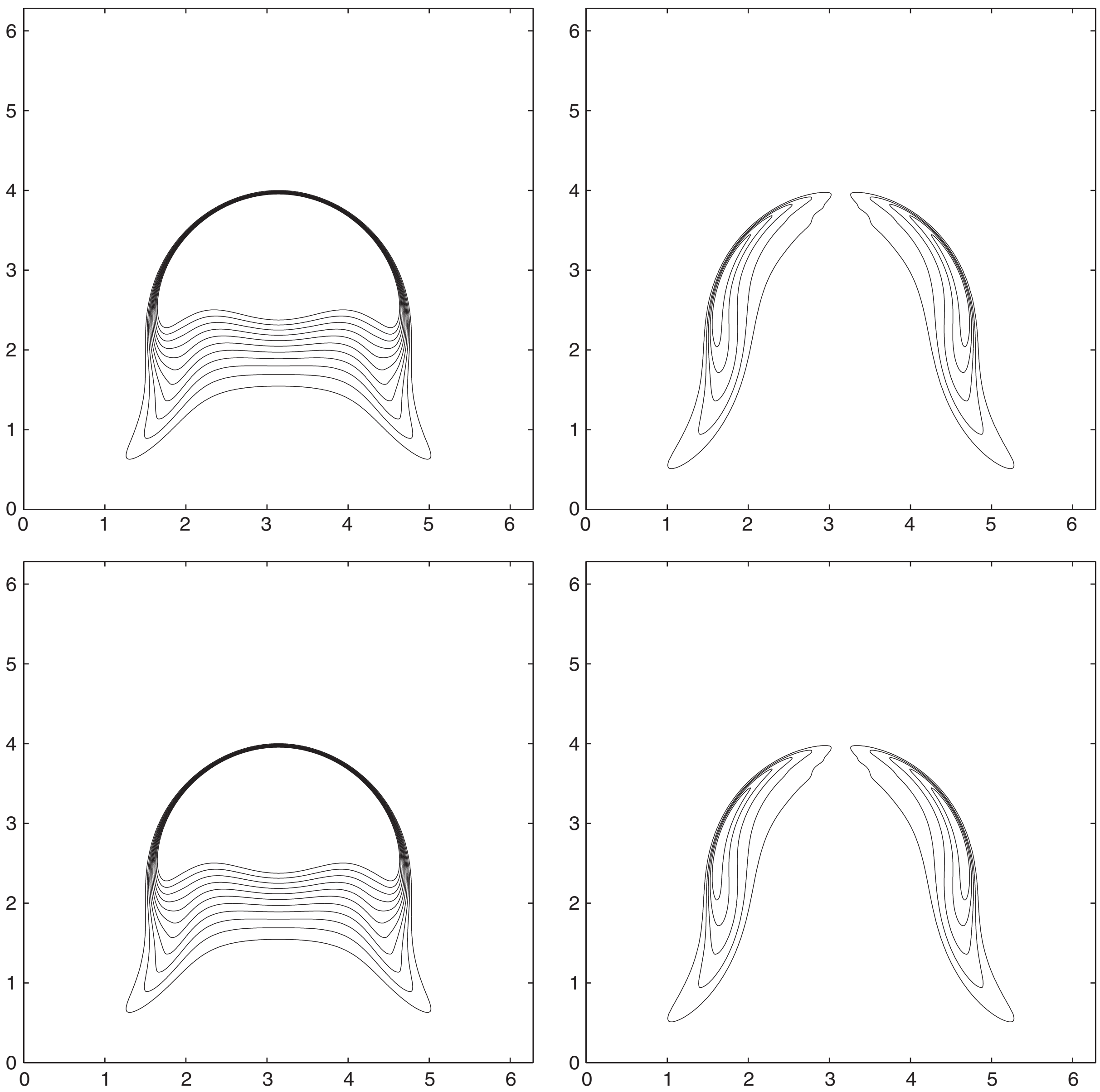

Fig. 2. Adaptive solution for the Boussinesq problem at $t = 1.6$. Left: density contour; right: vorticity contour. Top: $N_x = N_y = 300$; bottom: $N_x = N_y = 400$.

3.2. Some computational details

In Table 2, we show the compressed ratio of the adaptive mesh at different time levels. From this table, it is found that at the initial time the adaptive mesh is almost uniform due to the smoothness of the initial data. As the time gets larger, the compressed ratio becomes bigger, which implies the regularity of the

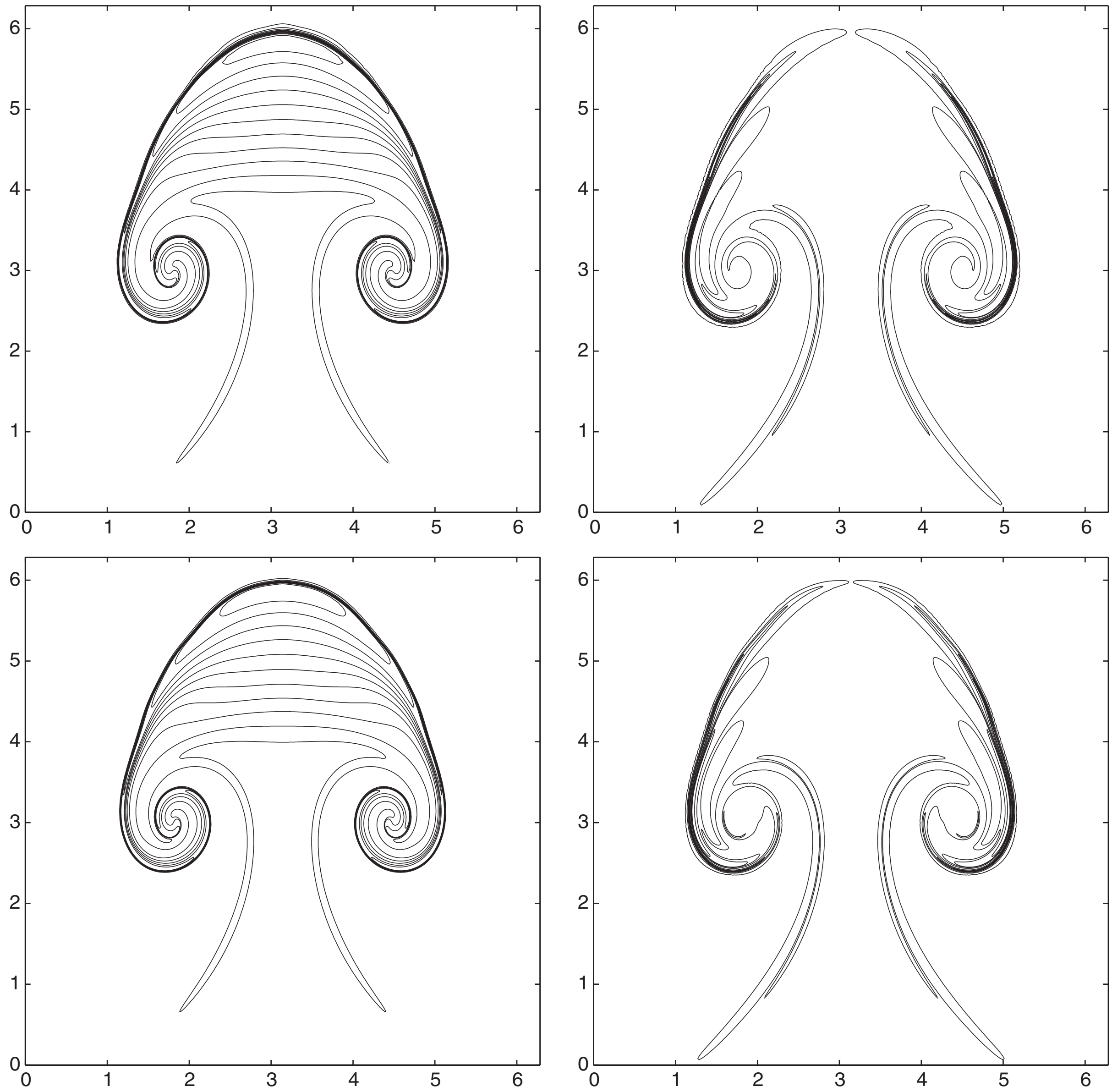

Fig. 3. Adaptive solution for the Boussinesq problem at $t = 3$. Left: density contour; right: vorticity contour. Top: $N_x = N_y = 300$; bottom: $N_x = N_y = 400$.

solution becomes poorer. In Table 2, the maximum and minimum meshes are defined as

$$\max \Delta x = \max_{j,k} \{\Delta x_{j,k}\}, \quad \min \Delta x = \min_{j,k} \{\Delta x_{j,k}\},$$

where $\Delta x_{j,k} = \max_{p,q \in \{1,2,3,4\}} \{|x_p - x_q|\}$, with x_p, x_q the four nodes of the cell $A_{j,k}$. The $\max \Delta y$ and $\min \Delta y$ are defined similarly. Also in Table 2 $|A_{j,k}|$ denotes the area of the cell $A_{j,k}$.

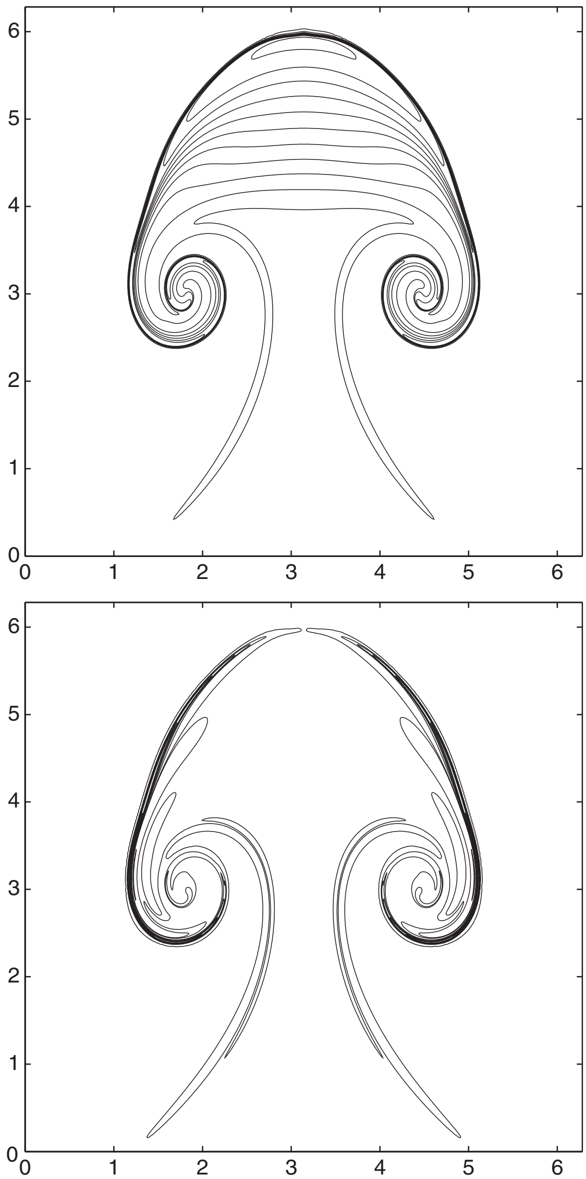

Fig. 4. Adaptive solution for the Boussinesq problem at $t = 3$ with $N_x = N_y = 500$. Top: density ρ contour; bottom: vorticity ω contour.

To compute the stream function Eq. (3.7), we need to write it in the computational domain, which leads to a large sparse positive definite linear system of the form $Lx = f$. Since the coefficient matrix L depends on the coordinate transformation, we have to solve the linear system at each time level. This requires a fast solver for the calculation of $Lx = f$. We employ the MGPCG method which is briefly

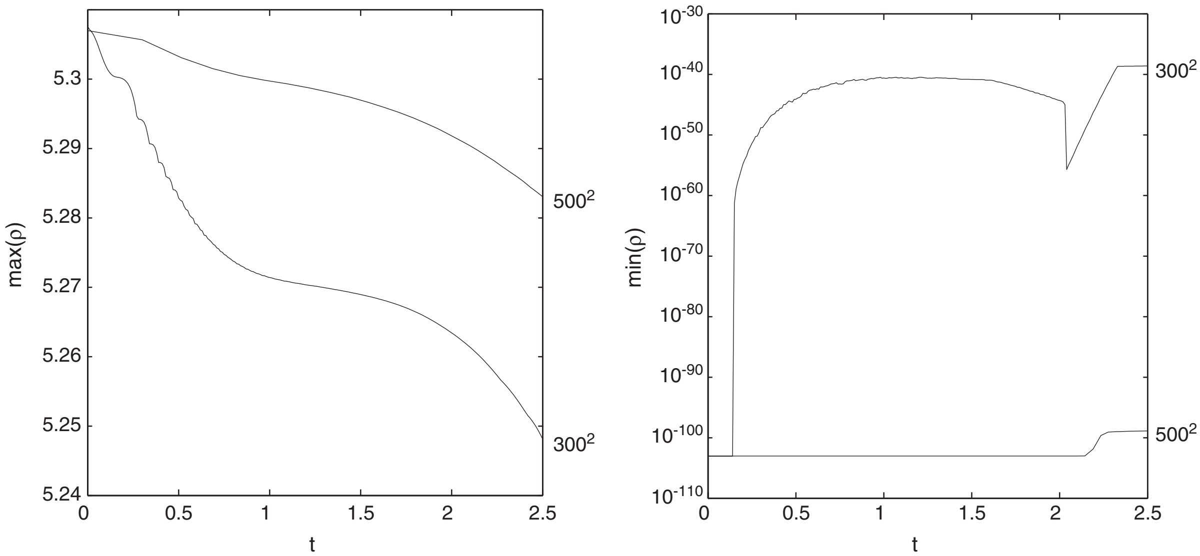

Fig. 5. Time history of maximum (left) and minimum (right) density.

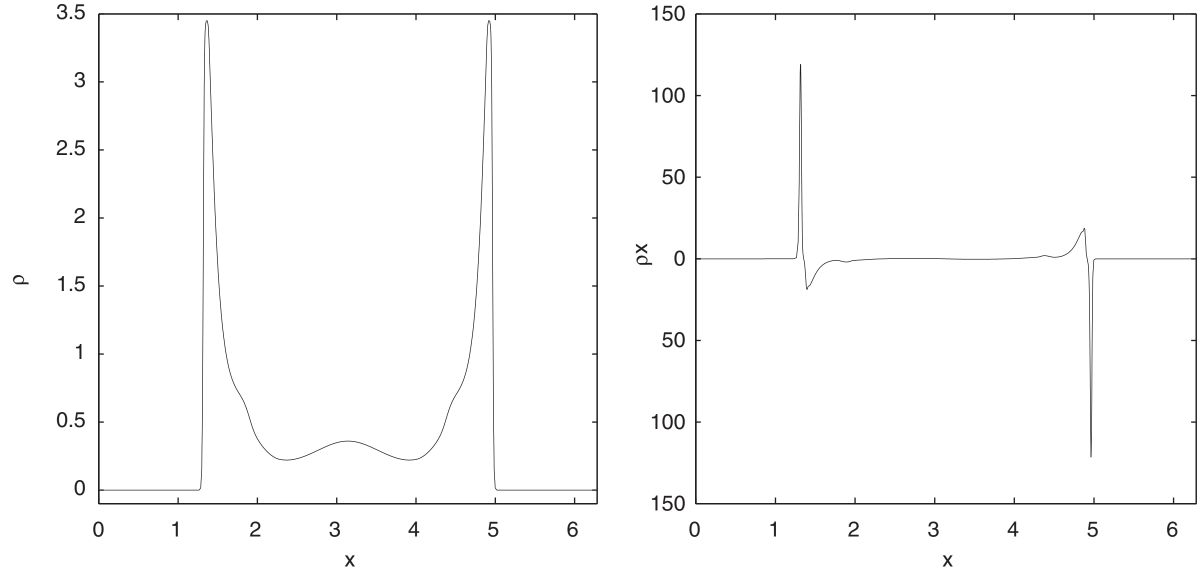

Fig. 6. Adaptive mesh solution for the Boussinesq problem with $N_x = N_y = 500$. Left: density ρ at $t = 2.5$, $y = \pi$. Obviously, the numerical result is stable and no oscillation occurs. Right: ρ_x at $t = 2.5$, $y = \pi$.

outlined below:

- Given the initial guess $x^0, r^0 = f - Lx^0,\ L_l \tilde{r}^0 = r^0,\ p^0 = \tilde{r}^0$.
- $\alpha_i = (\tilde{r}^i, r^i)/(p^i, L_l p^i)$.
- $x^{i+1} = x^i + \alpha_i p^i$.
- $r^{i+1} = r^i - \alpha_i L_l p^i$, check convergence.

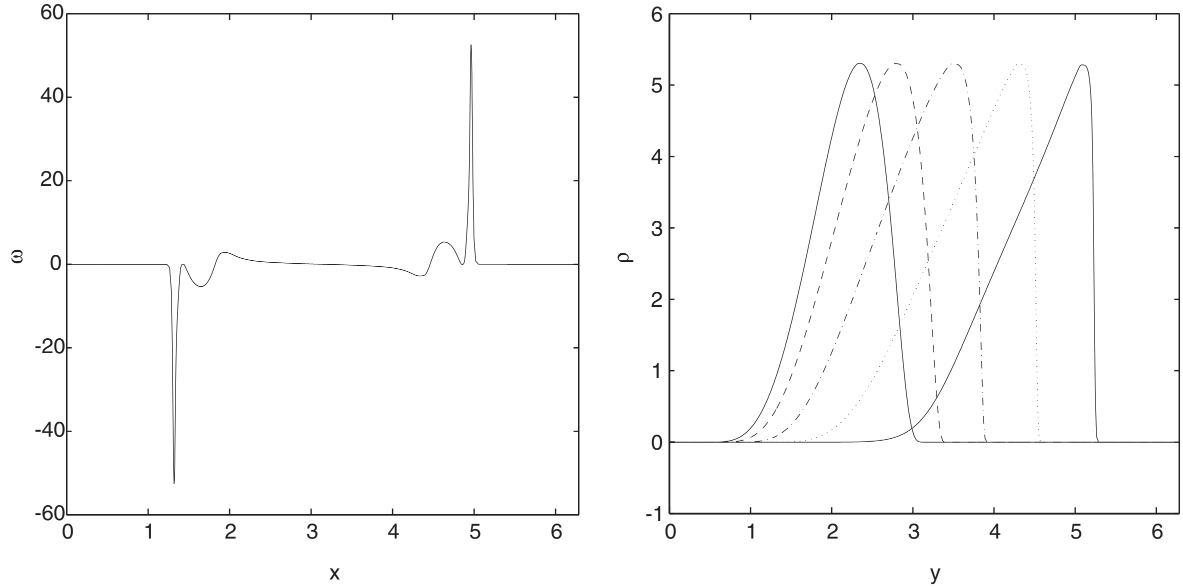

Fig. 7. Adaptive mesh solution for the Boussinesq problem with $N_x = N_y = 500$. Left: vorticity ω at $t = 2.5$, $y = \pi$, right: evolution of density along the symmetry axis $x = \pi$ at $t = 0.5$, 1, 1.5, 2, 2.5.

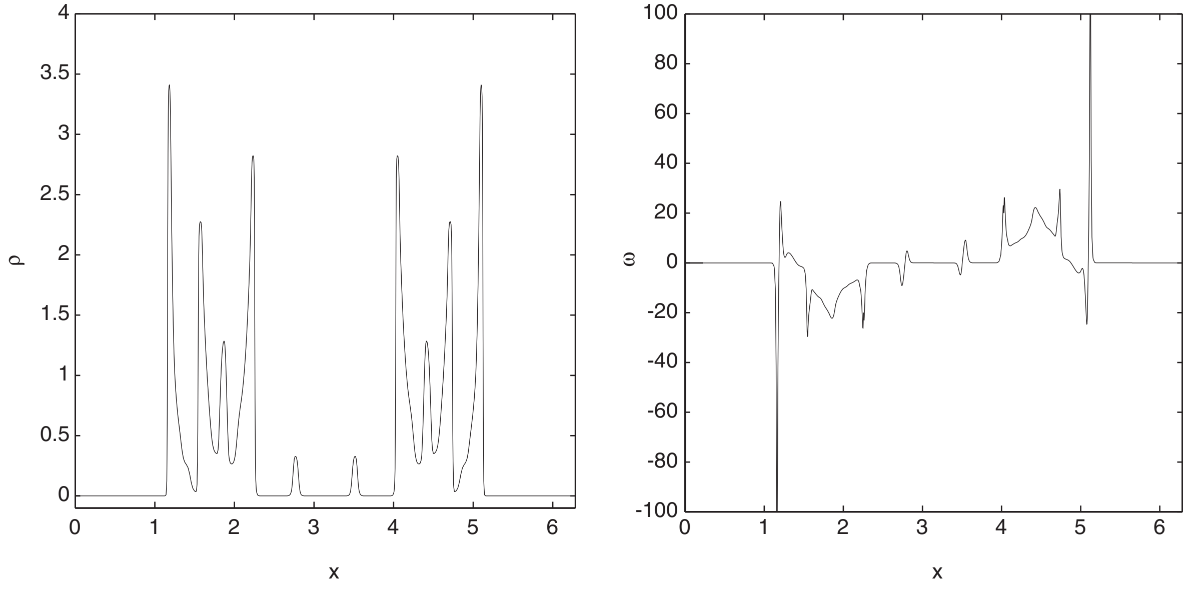

Fig. 8. Adaptive mesh solution for the Boussinesq problem with $N_x = N_y = 500$. A cut of density (left) and vorticity (right) profile through the roll at $t = 3.16$. Complicated small scales are observed.

- relax $L_l \tilde{r}^{i+1} = r^{i+1}$ using multigrid method.
- $\beta_i = (\tilde{r}^{i+1}, r^{i+1})/(\tilde{r}^i, r^i)$.
- $p^{i+1} = \tilde{r}^{i+1} + \beta_i p^i$.

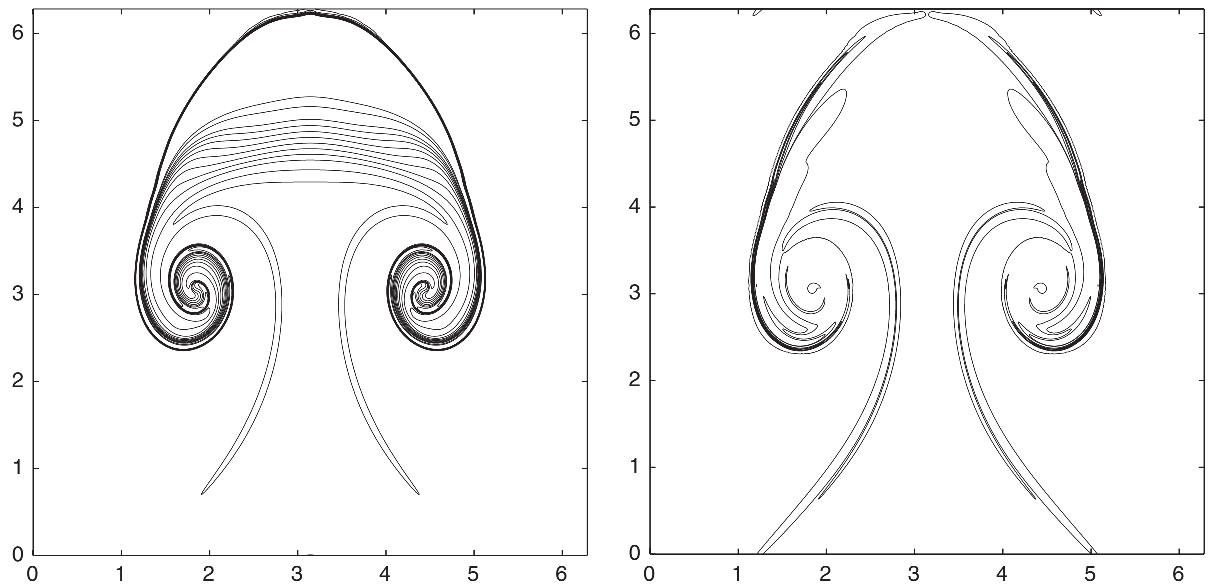

Fig. 9. Adaptive mesh solution for the Boussinesq problem with $N_x = N_y = 500$ at $t = 3.16$. Left: density ρ contour; right: vorticity ω contour.

Table 2
Boussinesq problem: adaptive mesh with $N_x = N_y = 400$

		$t=0$	$t=1.6$	$t=2.5$	$t=3$
Δx	max Δx	0.0158	0.0436	0.0732	0.0798
	min Δx	0.0157	0.0109	0.0043	0.0028
	max/min	1.0058	4.0032	7.0691	28.1504
Δy	max Δy	0.0158	0.0475	0.0904	0.2476
	min Δy	0.0156	0.0122	0.0039	0.0027
	max/min	1.0131	3.8882	22.8822	91.2458
$\|A_{jk}\|$	max $\|A_{jk}\|$	2.4756e−04	6.4874e−04	0.0011	0.0036
	min $\|A_{jk}\|$	2.4459e−04	1.2542e−04	3.8312e−05	2.6960e−05
	max/min	1.0121	5.1727	28.6289	133.1909

The largest and smallest sizes and their ratio at $t = 0, 1.6, 2.5, 3$.

In the first step the initial guess for x^0 can be taken as the related numerical results obtained in the previous time level. The matrix L_l is an approximation to the original problem matrix L. In our computation, we employ a robust multigrid method that uses matrix-dependent prolongation written by Zeeuw [22]. This particular multigrid method can efficiently handle the high-contrast variable coefficients introduced by the mesh map. Zeeuw's multigrid code is designed for 9-point scheme with Dirichlet boundary condition. Here, the Boussinesq problem is subject to periodic boundary condition, so we apply the multigrid method to $L_l\tilde{r} = r$ with homogeneous boundary condition. The outer CG loop satisfies periodic boundary

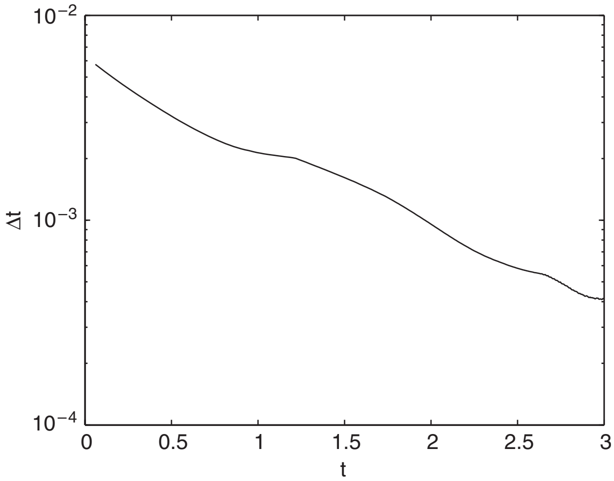

Fig. 10. The variation of time step with $N_x = N_y = 500$.

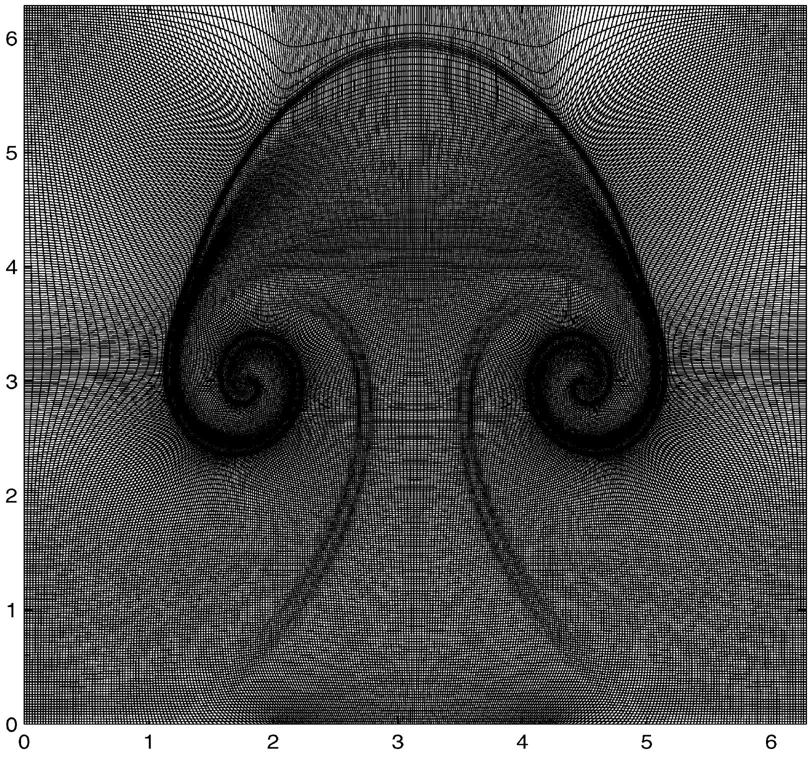

Fig. 11. Adaptive mesh for Boussinesq problem with $N_x = N_y = 300$ at $t = 3$.

condition. Both the tolerances for the outer CG loop and the inner multigrid loop are taken as 10^{-8}. In average, when the MPPCG is called at each time level, it takes 2–3 CG iterations and 3–4 multigrid iterations. Therefore, the cost for computing the stream function is not too expensive.

In our computation, the CFL number is chosen as 0.5. The variation of time step with time is plotted in Fig. 10. As expected the time step becomes smaller when the solution singularity is developed. In this case, the minimum size of the cell volume will become very small, see Table 2. In Fig. 11, we present the mesh redistribution at $t = 3$. Obviously, more grid points are clustered in the regions with sharp solutions or small structures.

4. Concluding remarks

In this work, an adaptive moving mesh method is applied to the two-dimensional incompressible Boussinesq system. Both the mesh generation equation and the underlying Boussinesq problem are solved in the computational domain. Particular attention is paid to preserve the conservation properties of the system. A fast solution solver MGPCG is applied to solve the large linear system. The numerical results demonstrate that our numerical approach can resolve the small-scale structures developed in the Boussinesq convection. There is no numerical oscillation occurred throughout the bubble evolution.

Acknowledgements

The research of the authors was supported by the Hong Kong Research Grants Council (Hong Kong RGC Grant 2033/99P).

References

[1] M.J. Baines, Moving Finite Elements, Oxford University Press, Oxford, 1994.
[2] J.T. Beale, T. Kato, A. Majda, Remarks on the breakdown of smooth solutions for 3-D Euler equations, Commun. Math Phys. 94 (1984) 61–66.
[3] M.J. Berger, P. Colella, Local adaptive mesh refinement for shock hydrodynamics, J. Comput. Phys. 82 (1989) 64–84.
[4] J.U. Brackbill, An adaptive grid with directional control, J. Comput. Phys. 108 (1993) 38–50.
[5] W.M. Cao, W.Z. Huang, R.D. Russell, An *r*-adaptive finite element method based upon moving mesh PDEs, J. Comput. Phys. 149 (1999) 221–244.
[6] H.D. Ceniceros, T.Y. Hou, An efficient dynamically adaptive mesh for potentially singular solutions, J. Comput. Phys. 172 (2001) 609–639.
[7] Y. Di, R. Li, T. Tang, P.W. Zhang, Moving mesh finite element methods for the incompressible Navier–Stokes equations, SIAM J. Sci. Comput. 26 (2005) 1036–1056.
[8] Y. Di, R. Li, T. Tang, P.W. Zhang, Level set calculations for incompressible two-phase flows on a dynamically adaptive grid, Preprint, 2005.
[9] A.S. Dvinsky, Adaptive grid generation from harmonic maps on Riemannian manifolds, J. Comput. Phys. 95 (1991) 450–476.
[10] W.-N. E, C.-W. Shu, Small-scale structures in Boussinesq convection, Phys. Fluids 6 (1994) 49–58.
[11] R. Grauer, C. Marliani, K. Germaschewski, Adaptive mesh refinement for singular solutions of the incompressible Euler equations, Phys. Rev. Lett. 80 (1998) 4177–4180.
[12] R. Li, T. Tang, P. Zhang, Moving mesh methods in multiple dimensions based on harmonic maps, J. Comput. Phys. 170 (2001) 562–588.
[13] R. Li, T. Tang, P. Zhang, A moving mesh finite element algorithm for singular problems in two and three space dimensions, J. Comput. Phys. 177 (2002) 365–393.
[14] A. Pumir, E.D. Siggia, Development of singular solutions to the axisymmetric Euler equations, Phys. Fluids A 4 (1992) 1472–1491.

[15] A. Pumir, E.D. Siggia, Finite-time singularities in the axisymmetric three-dimensional Euler equations, Phys. Rev. Lett. 68 (1992) 1511–1514.
[16] C.-W. Shu, S. Osher, Efficient implement of essentially non-oscillatory shock-wave schemes, II, J. Comput. Phys. 83 (1989) 32–78.
[17] T. Tang, Moving mesh methods for computational fluid dynamics, in: Z.-C. Shi, Z. Chen, T. Tang, D. Yu (Eds.), Recent Advances in Adaptive Computations, Contemporary Mathematics, American Mathematical Society, 2005.
[18] H.Z. Tang, T. Tang, Moving mesh methods for one- and two-dimensional hyperbolic conservation laws, SIAM J. Numer. Anal. 41 (2001) 487–515.
[19] H.Z. Tang, T. Tang, P.-W. Zhang, An adaptive mesh redistribution method for nonlinear Hamilton–Jacobi equations in two- and three-dimensions, J. Comput. Phys. 188 (2003) 543–572.
[20] E.D. Wilkes, S. Phillips, O. Basaran, Computational and experimental analysis of dynamics of drop formation, Phys. Fluids 11 (1999) 3577–3598.
[21] A. Winslow, Numerical solution of the quasi-linear Poisson equation on a nonuniform triangle mesh, J. Comput. Phys. 1 (1967) 149–172.
[22] P.M. De Zeeuw, Matrix-dependent prolongation and restrictions in a blackbox multigrid solver, J. Comput. Phys. 33 (1990) 1–27.
[23] P.A. Zegeling, Moving Grid Methods, Utrecht University Press, Utrecht, 1992.
[24] Z.R. Zhang, T. Tang, An adaptive mesh redistribution algorithm for convection-dominated problems, Commun. Pure Appl. Anal. 1 (2002) 57–73.
[25] X. Zheng, J. Lowengrub, A. Anderson, V. Cristini, Adaptive unstructured volume remeshing II: applications to two- and three-dimensional level-set simulations of multiphase flow, Preprint, 2005.
[26] A.Z. Zinchenko, M.A. Rother, R.H. Davis, A novel boundary-integral algorithm for viscous interaction of deformable drops, Phys. Fluids 9 (1997) 1493–1511.