Journal of Computational Physics **172,** 609–639 (2001)
doi:10.1006/jcph.2001.6844, available online at http://www.idealibrary.com on IDEAL®

An Efficient Dynamically Adaptive Mesh for Potentially Singular Solutions

Hector D. Ceniceros* and Thomas Y. Hou†

**Department of Mathematics, University of California Santa Barbara, Santa Barbara, California 93106; and*
†*Applied Mathematics, California Institute of Technology, Pasadena, California 91125*
E-mail: hdc@math.ucsb.edu; hou@ama.caltech.edu

Received May 2, 2000; revised May 29, 2001

We develop an efficient dynamically adaptive mesh generator for time-dependent problems in two or more dimensions. The mesh generator is motivated by the variational approach and is based on solving a new set of *nonlinear* elliptic PDEs for the mesh map. When coupled to a physical problem, the mesh map evolves with the underlying solution and maintains high adaptivity as the solution develops complicated structures and even singular behavior. The overall mesh strategy is simple to implement, avoids interpolation, and can be easily incorporated into a broad range of applications. The efficacy of the mesh is first demonstrated by two examples of blowing-up solutions to the 2-D semilinear heat equation. These examples show that the mesh can follow with high adaptivity a finite-time singularity process. The focus of applications presented here is however the baroclinic generation of vorticity in a strongly layered 2-D Boussinesq fluid, a challenging problem. The moving mesh follows effectively the flow resolving both its global features and the almost singular shear layers developed dynamically. The numerical results show the fast collapse to small scales and an exponential vorticity growth.

Key Words: semilinear heat equation, Euler singularity, Boussinesq flow, Rayleigh–Bénard convection, moving mesh.

1. INTRODUCTION

How can we compute accurately the collapse to very small length scales and the rapid loss of regularity of a time-evolving solution? A solution-adaptive mesh is indispensable for this task. There are many existing mesh-adaptive methods for this type of problem. Mesh adaptivity is usually in the form of local mesh refinements or through a bijective and continuous mesh mapping. The adaptive mesh can also be static or dynamic (continuously moving) [1, 3, 22, 33, 37, 39, 40, 44]. In local adaptive mesh refinement methods (see e.g. [7]), an adaptive mesh is obtained by adding or removing points to achieve a desired level

0021-9991/01 $35.00

of accuracy. This allows a systematic error analysis. However, local refinement methods require complicated data structures and fairly technical methods to communicate information among different levels of refinements. In the mapping approach, the mesh points are moved continuously in the whole domain to concentrate in regions where the solution has the largest variations. Due to strong nonlinear coupling of the mesh map with the underlying physical partial differential equation (PDE), *a priori* error estimate is difficult to obtain in this case. Nevertheless, it is possible to design mesh mappings that reflect closely the solution's geometry and regularity and that can be used to compute accurately finite-time singularity formation (see e.g. [12, 13]). These solution-adaptive mesh maps have the additional advantage of allowing the use of standard solvers as all the computations are performed in the logical domain using a uniform mesh. In this work, we propose a new dynamically adaptive mesh generator of this type.

Our adaptive mesh is motivated by the variational approach and is based on solving a simple set of *nonlinear* elliptic PDEs for the mesh map. The overall mesh strategy is cost efficient, easy to implement, and avoids interpolation. When coupled to a physical problem, the mesh evolves with the physical solution and maintains high adaptivity as the solution develops complicated structures. As we demonstrate, the proposed moving mesh can effectively be used to compute accurately multidimensional solutions that blow-up (become unbounded) in finite time as well as problems with complex and potentially singular dynamics.

Important physical phenomena that develop dynamically singular or nearly singular solutions in fairly localized regions (e.g., shear flows, shocks, multiphase flows, focusing waves, etc.) abound. The numerical investigation of these problems requires extremely fine meshes to resolve accurately the large and often nearly singular solution variations in small regions. The use of well-refined uniform meshes becomes computationally prohibitive when dealing with systems in two or three dimensions. Developing an effective adaptive mesh strategy for these problems becomes necessary. However, because of complicated solution structures and the global coupling of meshes at different length scales (especially for incompressible flows), it is very challenging to develop a robust and computationally stable adaptive mesh strategy. Particularly, a strategy with a mesh that can follow effectively the evolution of nearly singular layered solutions dynamically. In addition, it is important to compute the potentially singular solutions without introducing excessive artificial diffusion through frequent interpolations at different grid levels.

The design of our dynamically adaptive mesh was motivated by the fascinating and still open problem about whether a finite-time singularity can form out of smooth initial data in inviscid and incompressible 3-D Euler flows. This is not just a mathematical question. The finding and understanding of finite-time singularities may be crucial to explain small-scale structures in viscous turbulent flows. In this work, we apply our new dynamically adaptive mesh to investigate the production and concentration of vorticity in 2-D Boussinesq convection of a strongly layered fluid. The governing equations of Boussinesq convection are analogous to those of 3-D axi-symmetric Euler flow with swirl (see e.g. [42, 43]). As previous numerical studies have shown [26, 28–30, 42, 43], the complex dynamics and the rapid formation of small scales make this problem an extremely demanding test for any adaptive mesh technique. The numerical results presented here demonstrate that our adaptive mesh follows effectively the almost singular shear layers developed dynamically. The numerical solution remains very stable throughout the computation and as the physical solution becomes more singular, the adaptivity improves. Does the vorticity blow up in finite

time? The computations reveal that it only grows exponentially for the initial conditions we consider here. The importance of a nontrivial geometry for the potential singularity formation is supported by our numerics.

Traditionally, a mesh map is obtained as the solution to elliptic PDEs generated from a variational principle in the physical space (see Section 2). Information about the underlying physical solution is built into the mesh PDEs. In contrast, here we turn to the computational space to seek a mesh in which the nearly singular physical solution is better behaved (less localized). Using a variational principle in the computational space rather than in the physical domain as a guide, we propose a single set of *nonlinear* PDEs whose direct solution gives an efficient adaptive mesh. The information about localized singular regions is effectively spread in the computational domain. The nonlinear elliptic equations we propose are, to the best of our knowledge, a new mesh generator that, as we show here, is efficient, and can generate a good quality mesh. It can be implemented easily with fast Poisson solvers at the minimum cost of $O(N)$ operations, where N is the total number of grid points. Dynamic adaptivity is obtained naturally by following the moving mesh idea of Huang and Russell [33] which consists of solving alternately time-dependent flow equations associated with the mesh PDEs and the underlying physical equations. The overall result is a computationally efficient mesh that dynamically adapts to the complicated geometry of the time-varying and nearly singular solution, increasing the compression ratio (uniform grid size over smallest adaptive grid size) as a singularity is approached.

The paper is organized in two main parts. The first part (Sections 2–4) introduces our mesh strategy, demonstrates its efficiency in computing singular solutions, and provides a detailed guide about how to incorporate the dynamically adaptive mesh to compute time-dependent problems. Speciffically, in Section 2 we review the classical variational approach to mesh generation. In Section 3, we introduce our adaptive mesh guided by a variational principle. The effectiveness of the proposed mesh is illustrated with two extreme static examples and with the application of the moving mesh to compute the finite-time blowing-up of solutions to the 2-D semilinear heat equation. The simple steps to implement the adaptive mesh for a time-dependent problem are reviewed in Section 4. The second part (Sections 5–7) is devoted to the application of the dynamically adaptive mesh to investigate the baroclinic generation of vorticity and the collapse to small scales of a multilayered Boussinesq fluid. The governing equations of Boussinesq convection are presented in Section 5 and the numerical methodology for this problem is described in detail in Section 6. The numerical results are presented in Section 7. Finally, some concluding remarks are given in Section 8.

2. CLASSICAL VARIATIONAL MESH GENERATION

An adaptive mesh may be generated through a bijective map from a logical or computational domain to the physical domain. Typically, the mesh map transforms a uniform mesh in the logical space to cluster grid points at the regions of the physical domain where the solution has the largest gradients (see e.g. the books [35, 46]).

Let us denote by $(x(\xi, \eta), y(\xi, \eta))$ the mesh map in two dimensions. Here (ξ, η) are the computational coordinates or inverse map. In the variational approach, this map is provided by the minimizer of a functional of the following form:

$$E[\xi, \eta] = \frac{1}{2} \int_{\Omega_p} \left[\nabla \xi^T G_1^{-1} \nabla \xi \; + \; \nabla \eta^T G_2^{-1} \nabla \eta \right] dx\, dy, \tag{1}$$

where G_1 and G_2 are given symmetric positive definite matrices called monitor functions and $\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y})^T$. Here Ω_p denotes the physical domain. More terms can be added to the functional (1) to control other aspects of the mesh such as orthogonality (skewness) and mesh alignment with a given vector field [10, 11]. There are also adaptive meshes based directly on a discrete variational principle [15, 16].

The variational mesh is determined by the Euler–Lagrange equations associated with $E[\xi, \eta]$:

$$\nabla \cdot \left(G_1^{-1} \nabla \xi\right) = 0, \quad \nabla \cdot \left(G_2^{-1} \nabla \eta\right) = 0. \tag{2}$$

Specifically, if $u(x, y, t)$ is the solution at a given time t of the underlying PDE we are interested into solve for later times, then the monitor functions should depend on u. One of the simplest choices of monitor functions is $G_1 = G_2 = wI$, where I is the identity matrix and $w > 0$ is a weight function, for example $w = \sqrt{1 + u_x^2 + u_y^2}$. In this case, we obtain Winslow's variable diffusion method [47]:

$$\nabla \cdot \left(\frac{1}{w} \nabla \xi\right) = 0, \quad \nabla \cdot \left(\frac{1}{w} \nabla \eta\right) = 0. \tag{3}$$

In one dimension, this reduces to de Boor's *equidistribution principle* [21],

$$w x_\xi = C \quad \text{or} \quad \int_0^x w(x)\, dx = \xi C, \tag{4}$$

where C is a constant. This means that w is equally distributed in an averaged (integral) sense. But the choice of w is problem-dependent. For the interesting problem of the semilinear 1D heat equation, Budd, Huang, and Russell [13] have shown that, taking into account some scaling invariance of the solution, it is possible to select the equidistributed monitor function to accurately follow the finite time blow-up of the solution. Exploiting also the solution scaling invariance, Budd, Chen, and Russell [12] have obtained an optimal monitor function for the radially symmetric nonlinear Schrödinger equation. A different static method based on an iterative procedure on the Winslow map has been proposed by Ren and Wang [44]. Their method does not tailor the monitor function to the problem but relies instead on iteration and interpolation to statically redistribute the adaptive mesh. This appears to be successful in computing singularity formation for two 2-D problems where the location of singularity is fixed. Another iterative redistribution method, but this one dynamic, has been introduced recently by Li, Tang, and Zhang [36]. In contrast, we propose here an alternative efficient mesh generator obtained directly (without any iteration or interpolation) from a new set of simple nonlinear PDEs. The dynamically adaptive mesh can effectively follow with high adaption the rapid dynamics of potentially singular solutions.

3. AN EFFICIENT ADAPTIVE MESH FROM THE COMPUTATIONAL DOMAIN

We shall present here an efficient mesh generator motivated by a variational principle in the computational domain as opposed to the commonly used variational principle in the physical domain described in Section 2. The mesh generator we propose is then combined

naturally with the moving mesh idea of Huang and Russell [33] to achieve dynamic adaptivity. This idea consists of solving alternately time-dependent flow mesh PDEs and the underlying physical equations one time step at a time. This section is divided in four parts: the presentation of the mesh generator (the set of nonlinear elliptic equations), the moving mesh equations, some guidelines on how to select the mesh monitor function, and a set of examples including the blowing-up of solutions to the semilinear 2-D heat equation. These examples illustrate the efficiency of the adaptive mesh to accurately resolve singular behavior.

3.1. *A New Mesh Generator*

For simplicity of the presentation we limit our discussion to the 1-D and 2-D cases but our mesh generator generalizes straightforwardly to 3-D. To describe our approach, let us consider first a 1-D example and assume that we have given an underlying solution $u(x)$. Our approach is motivated by the following observation of Ren and Wang [44] (which is the starting point of their iterative method): with a good adaptive mesh $v(\xi) = u(x(\xi))$, i.e., the function in the computational space should be "better behaved." With this in mind, it is natural to look for the mesh map $x(\xi)$ that minimizes a measure of the gradient of v, say

$$\min_{x(\xi)} \int_{\Omega_c} \sqrt{1 + v_\xi^2}\, d\xi = \min_{x(\xi)} \int_{\Omega_c} \sqrt{1 + u_x^2(x(\xi))x_\xi^2}\, d\xi, \tag{5}$$

where Ω_c is the computational (logical) domain. The Euler–Lagrange equation associated with this variational problem is

$$\left(\frac{u_x^2}{\sqrt{1 + u_x^2 x_\xi^2}} x_\xi \right)_\xi = \frac{u_x u_{xx} x_\xi^2}{\sqrt{1 + u_x^2 x_\xi^2}}. \tag{6}$$

This is a nonlinear elliptic equation with a very stiff source term (right-hand side). Note that the source term contains a second-order derivative in the physical space. In practice, when u is nearly singular, the extremely large nonlinear source term imposes a numerical constraint so severe that it makes the numerical solution of (6) computationally infeasible. Moreover, since the coefficient in the elliptic term (left-hand side) of (6) can be zero, the equation is also degenerate.

Although Eq. (6) cannot be used in practice to generate a solution-adaptive mesh, it provides important information regarding the spreading of the singular regions of u. Indeed, through numerical experiments we notice that both the elliptic and the source term contribute to the spreading of u in the computational domain. However, by switching off the source term we observe that the elliptic term alone is sufficient to produce an effective spreading of the singular regions in the computational space. As a consequence, a candidate for a good mesh generator is obtained by setting to zero the right-hand side of (6) and by modifying the elliptic coefficient to avoid degeneracy:

$$\left(\frac{1 + u_x^2}{\sqrt{1 + u_x^2 x_\xi^2}} x_\xi \right)_\xi = 0. \tag{7}$$

This equation has still a very singular coefficient. For computational purposes it is better to replace u_x in the coefficient numerator by the smoother quantity $u_x x_\xi = v_\xi$. Thus, we can write our nonlinear mesh equation in the following simple form:

$$\frac{\partial}{\partial \xi}(w x_\xi) = 0 \quad \text{with } w = \sqrt{1 + u_x^2 x_\xi^2}. \tag{8}$$

Note that this equation has the same form as the 1-D Winslow equation (this will not be the case in 2-D) except that now the weight function involves the derivative in the computational space. The mesh equation (8) generalizes naturally to higher dimensions and, unlike the classical mesh equations (2), maintains a very simple structure. For example in 2-D, our adaptive mesh generator becomes

$$\nabla' \cdot (w \nabla' x) = 0, \quad \nabla' \cdot (w \nabla' y) = 0, \tag{9}$$

where $\nabla' = (\frac{\partial}{\partial \xi}, \frac{\partial}{\partial \eta})^T$ and $w = \sqrt{1 + |\nabla' u|^2}$ as a particular choice of monitor function. More generally we will take a monitor function of the form

$$w = \sqrt{1 + \beta^2 |\nabla' u|^2 + g^2(u)}, \tag{10}$$

where β is a scaling constant and $g(u)$ is a function of u chosen to reflect the leading order dynamic growth rate of the time-dependent problem to be solved. We elaborate more on this but first some remarks about (9).

It is important to note that although the system (9) has the same form as that of the length functional method described in the book by Knupp and Steinberg [35] (Eq. (6.52) on p. 130) the two systems are fundamentally different. The length functional equations are linear and uncoupled whereas the system (9) is nonlinear and coupled. Equations (9) are, to the best of our knowledge, a new mesh generator. In connection with the length functional linear equations Dvinsky [25] (see also [35]) has shown that folded grids can result for nonconvex domains and thus there is the possibility that the mesh generator (9) could face the same problem (e.g., for smooth u). Nevertheless, in our experience with rectangular (convex) domains we have found that (9) produces smooth good quality meshes.

Let us now go back to the monitor function (10). For simplicity consider the 1-D case. Suppose for example that $u(x)$ is very localized with a large derivative and the computational and physical domains are the same, say [0, 1]. An optimal compression ratio would be obtained for a mesh such that $u_\xi = O(\|u\|_\infty)$ because the localized physical region would be spread completely in the computational domain [0, 1]. Using Eq. (8) with $w = \sqrt{1 + \beta^2 u_\xi^2}$, it can be shown that this is so if β is of order $\|u_x\|_\infty \|u^2\|_\infty^{-1}$. Before we address the dynamical aspect of the adaptive mesh and provide some guideline on how to select the function g for time-dependent problems, let us illustrate the effectiveness of the mesh generator (9) with the following two static examples.

EXAMPLE 1. Let $u = ce^{-c^2(x^2+y^2)}$ with $c = 100$ and solve (9) with $w = \sqrt{1 + \beta^2 |\nabla' u|^2}$. Note that $\|\nabla u\|_\infty \|u^2\|_\infty^{-1} = O(1)$ so we take $\beta = 1$. We use only $N = 128^2$ points. The numerical method we employ to solve (9) is discussed in detail in the next section.

The function u represented in the physical space, i.e., in the (x, y) coordinates, is shown in Fig. 1a. Note that $u(x, y)$ has a very sharp δ-function form and a uniform grid would require thousands of points per dimension to resolve it. In the transformed (ξ, η) space, u

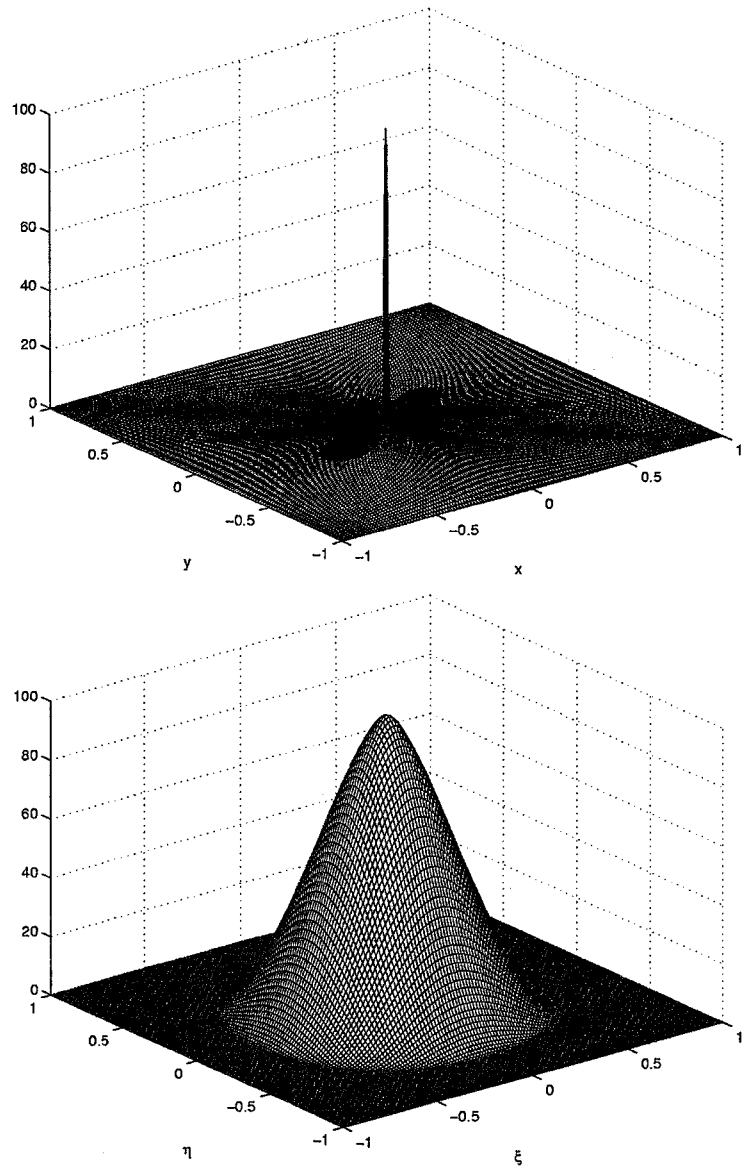

FIG. 1. $u = ce^{-c^2(x^2+y^2)}$ with $c = 100$. (a) Physical space: $u(x, y)$ (top). (b) Computational space: $v(\xi, \eta) = u(x(\xi, \eta), y(\xi, \eta))$ (bottom).

has a much wider support, as Fig. 1b shows, and decays smoothly toward the computational domain boundary. The adaptive mesh for the whole physical domain is shown in Fig. 2a. There is a very high density concentration of grid points in the vicinity of the peak. Figure 2b gives a close-up of this region. The compression ratio, i.e., the ratio of the uniform grid size and the smallest adaptive grid size, for this example is about 40.

EXAMPLE 2. Let $u = e^{-c^2(x^2+y^2)}$ with $c = 100$. Again, $w = \sqrt{1 + \beta^2|\nabla' u|^2}$ but now $\beta = c$ as $\|\nabla u\|_\infty \|u^2\|_\infty^{-1} = O(c)$. The function u in the computational space, i.e., $v(\xi, \eta) = u(x(\xi, \eta), y(\xi, \eta))$ appears in Fig. 3a and the corresponding adaptive mesh for the whole physical domain is shown in Fig. 3b. The mesh performs just as well as for

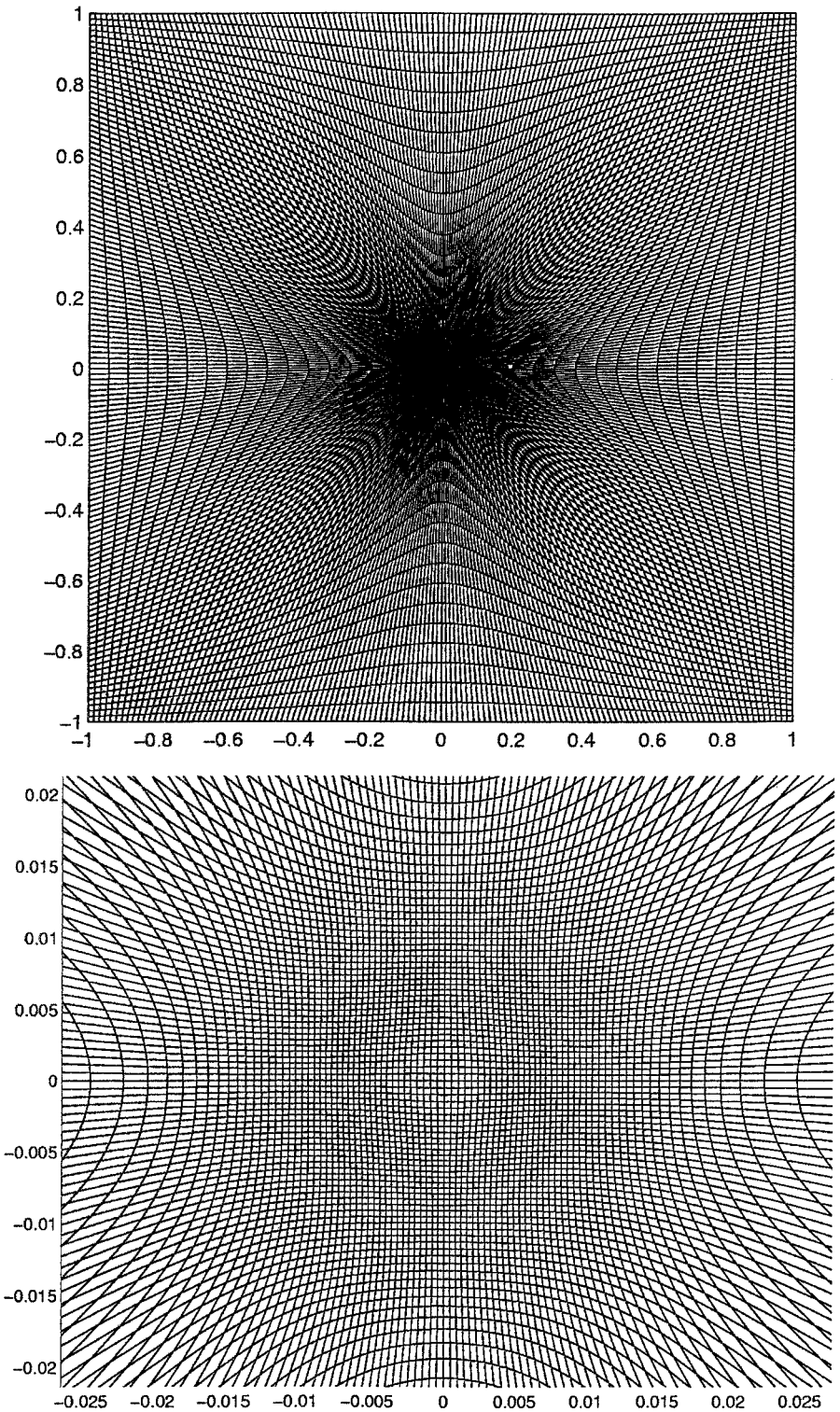

FIG. 2. The adaptive mesh for $u = ce^{-c^2(x^2+y^2)}$ with $c = 100$ and $N = 128^2$. (a) The whole physical domain and (b) a close-up of the mesh around the peak of u.

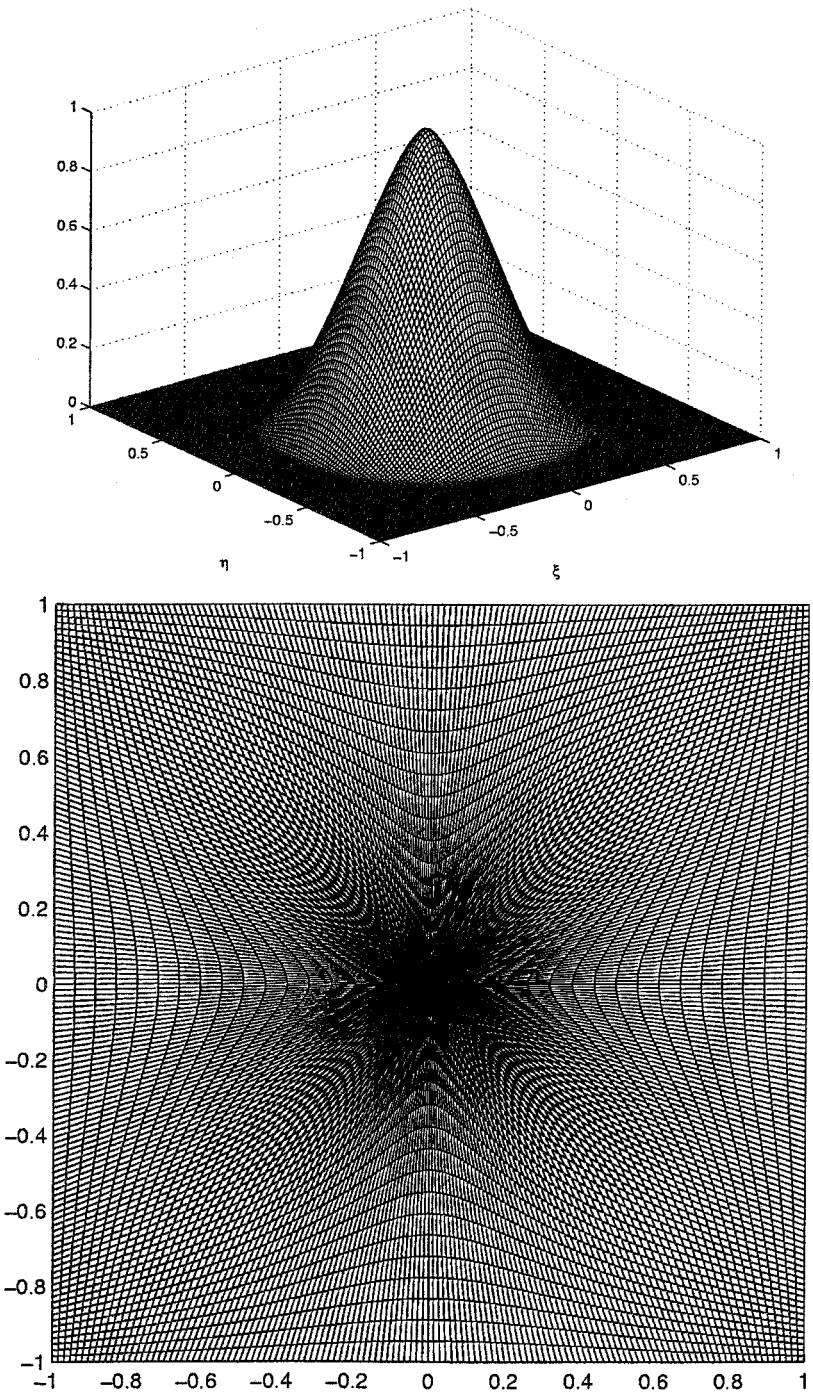

FIG. 3. $u = e^{-c^2(x^2+y^2)}$ with $c = 100$. Adaptive mesh obtained using $\beta = c$ in the monitor function. (a) The function in the computational space, i.e., $v(\xi, \eta) = u(x(\xi, \eta), y(\xi, \eta))$. (b) The adaptive mesh for the whole physical domain.

Example 1. In fact, the mesh distribution and the functions in the computational space look the same for both examples. However, if we take $\beta = 1$ instead of $\beta = c$ the mesh adaptivity would be limited and will deteriorate as c is increased.

3.2. *The Moving Mesh*

While our adaptive mesh can effectively resolve very singular functions we still need to provide a mechanism for dynamically adjusting the mesh to possible rapid changes of time-dependent solutions. There are several methods to obtain a moving mesh (see e.g. [1, 3, 22, 33, 37, 39, 40]). Here, we adopt the so-called moving mesh PDE approach [31–33] in which a time-dependent PDE is introduced to determine the motion of the mesh. Both the moving mesh PDE (MMPDE) and the underlying physical equations are solved simultaneously or alternately. This approach has the advantage of avoiding interpolation between old and new grids which is necessary in the static methods. Interpolation may introduce too much numerical smoothing in problems in which the resolution of small scales is important and thus, desirably, it should be avoided.

Recently Huang and Russell [33] have introduced a very robust class of MMPDEs derived from the gradient flow equations associated with the mesh variational principle. Here, we apply the same idea directly to our proposed mesh equations (9).

A standard method to solve (9) is to consider the equations

$$x_\tau = \nabla' \cdot (w \nabla' x), \tag{11}$$

$$y_\tau = \nabla' \cdot (w \nabla' y), \tag{12}$$

where τ is an artificial time. Then, beginning with an initial guess, we march in "time" to steady state. Any discrete marching scheme to solve (11) and (12) can be regarded as an iterative method to solve the nonlinear system (9).

At $t = 0$, we can find the solution to (11)-(12) up to steady state to obtain a mesh that adapts well to the initial data. With this initial adaptive mesh, the solution u can be updated (using the underlying PDE) one time step. Then a new mesh is obtained using the updated u in the monitor function. However, since u changes only very little in one time step, it is not necessary to solve again (11) and (12) all the way to steady state. Besides, the initial mesh is already a very good initial guess. Thus, it is natural to march only one time step in (11) and (12) (or equivalently to do only one iteration) at a time. In other words, taking τ as the actual time, equations (11)-(12) are our MMPDEs. Therefore, we proceed solving the moving mesh and the underlying PDEs alternately one time step at a time [33].

3.3. *Selecting the Monitor Function for the Dynamic Mesh*

As noted by Budd, Huang, and Russell [13] in the case of problems with finite-time blow-up, if the monitor function and the MMPDE are not chosen properly, the moving mesh may not share the underlying solution rapid dynamics and can fail to adapt as the singularity is approached. We give next some guidelines on how to select g in the monitor function (10) so that the adaptive mesh can follow even the fast dynamics encountered in a finite-time singularity formation process.

Let us illustrate the main idea with a well-known example of a problem with finite time blow-up: the semilinear heat equation

$$u_t = \Delta u + f(u) \quad u(x,0) = u_0(x) > 0; \quad x \in \Omega, \tag{13}$$

where

$$\frac{f(u)}{u} \to \infty \quad \text{as } u \to \infty.$$

For definiteness let us concentrate on the two-dimensional case with $\Omega = [0,1] \times [0,1]$ and with homogeneous Dirichlet boundary conditions, i.e., $u = 0$ on the boundary. This equation is a simple model for combustion [5], and it is well known (see e.g. [27]) that if u_0 is sufficiently large then the solution u will become unbounded in finite time.

To select the appropriate monitor function for this singular problem we note that as the solution to (13) grows, its dynamics are dictated by the nonlinear term $f(u)$ so that the leading order growth rate of the solution gradient is $f'(u)$, i.e, neglecting the diffusion term

$$\frac{\partial \nabla u}{\partial t} \sim f'(u) \nabla u. \tag{14}$$

To adapt efficiently, the dynamic mesh has to evolve at this rate which implies that $J f'(u) \sim$ constant, where J is the Jacobian of the mesh transformation. Simple asymptotics indicate that $J \sim w^{-1}$ and therefore $g(u) \sim f'(u)$ as $u \to \infty$. If $f'(u)$ is nonsingular for the range of u being considered, we can simply choose $g(u) = f'(u)$. The monitor function becomes

$$w = \sqrt{1 + \beta^2(t) |\nabla' u|^2 + (f'(u))^2}. \tag{15}$$

Note that $\beta(t) = \|\nabla u\|_\infty \|u^2\|_\infty^{-1}$ is now time-dependent. We demonstrate the capability of the mesh to capture the finite-time blowing-up of a solution to (13) and of a variant this equation with convection in the following two examples. The implementation details of the numerical methodology to include the dynamic mesh are addressed in the following section.

EXAMPLE 3. In this example we consider Eq. (13) with $f(u) = 4\sqrt{1+u^5}$ and the following initial condition

$$u_0(x,y) = 20 \sin^2(2\pi x) \sin^2(\pi y). \tag{16}$$

The initial condition has two humps along the x-direction. These humps grow rapidly to collapse into a pair of spikes where the solution becomes unbounded. Figure 4 presents the numerical solution at $t = 0.00258$ both in the physical and in the computational space. At this time, $\|u\|_\infty = 1.36 \times 10^7$. Despite u being so singular, with only $N = 128^2$ in the computational space, the adaptive mesh clearly resolves the blowing-up solution, maintaining it smooth in the logical domain as Fig. 4b shows. A close-up of the mesh near one of the spikes is given in Fig. 5 where the scale of the extremely high compression can be more clearly appreciated.

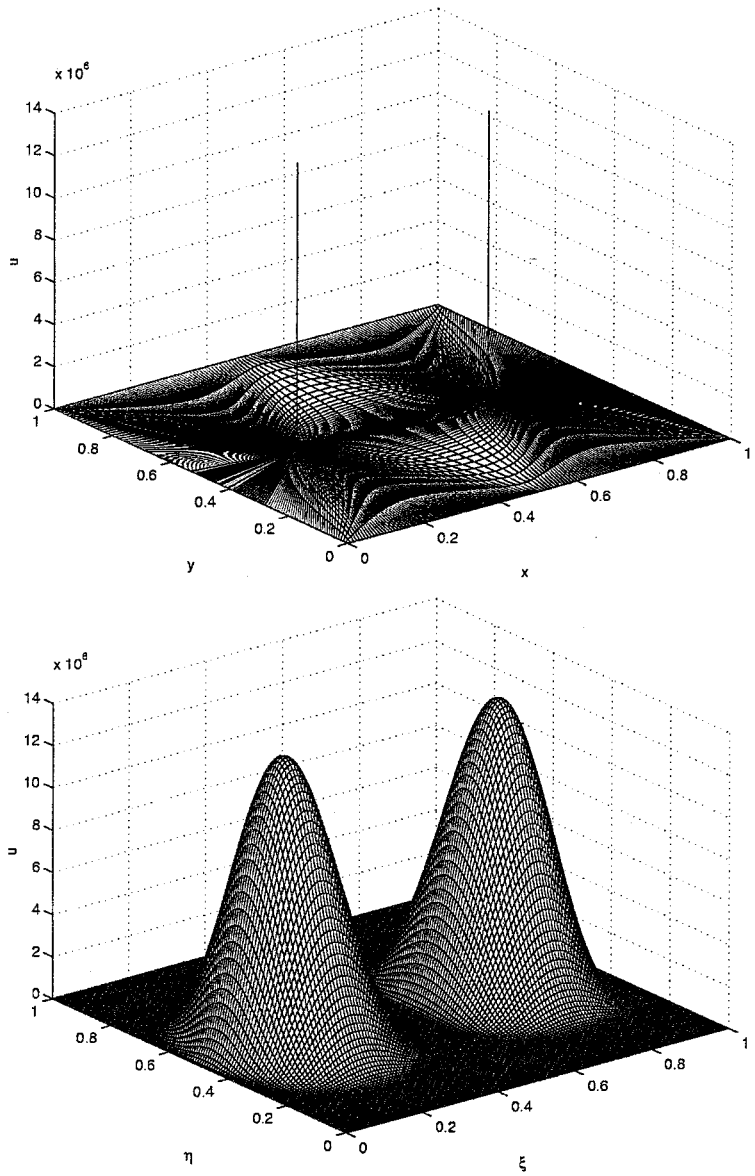

FIG. 4. Numerical solution to the semilinear heat equation (13) with $f(u) = 4\sqrt{1+u^5}$ at $t = 0.00258$ (a) Solution in the physical space (top) and (b) in the computational space (bottom); $\max u = 1.36 \times 10^7$, $N = 128^2$.

EXAMPLE 4. We now consider a variant of Eq. (13) to include convection and with a different nonlinearity as follows:

$$u_t + \cos(\pi(x + 0.2))uu_x = \Delta u + 4u^2. \tag{17}$$

With the added nonlinear convection term, the above equation does not seem to have a self-similar scaling. Although it is expected that without convection the solution would behave similarly as that in Example 3, more interesting dynamics will develop in the presence of this particular convection. As Figure 6a shows, the convection makes the two maxima interact. At $t = 0.02$, the two peaks have already merged into one (noncircular) peak as seen in

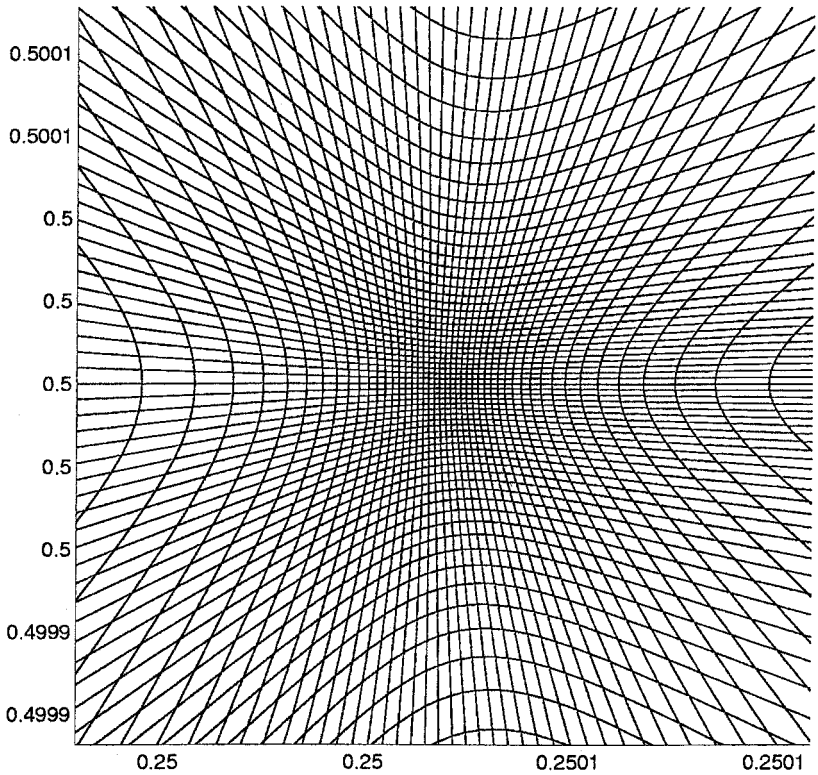

FIG. 5. Close-up of the mesh around one spike of the numerical solution to the semilinear heat equation (13) with $f(u) = 4\sqrt{1+u^5}$ at $t = 0.00258$.

Fig. 6b. From this point on, the solution grows rapidly developing a concentrated elliptical spike centered at (0.3, 0.5). Figure 7 presents the numerical solution at $t = 0.0436$ when $\|u\|_\infty = 5 \times 10^8$. Again the adaptive mesh maintains dynamically a smoothly resolved solution in the computational space (Fig. 7b).

Note that we have not made use of any a priori information of the underlying solution but only incorporated the leading order dynamic growth rate into the monitor function. It should also be noted that although this monitor function appears to be optimal in the sense of the extremely high compression ratio achieved for this particular class of blow-up problems, it may not yield the optimal mesh in other situations. The selection of the monitor function is problem-dependent and the scaling strategy presented here should be viewed as a guideline only. High compression comes at the expense of significant mesh deformation outside the most singular region and can affect largely the accuracy of the solution there. For some problems, for example the incompressible Boussinesq flow, we consider in the second part of this paper, the solution needs to be resolved accurately in the whole physical domain. In these cases, a compromise should be sought so that, while keeping good adaptivity, the mesh does not deform excessively.

4. SIMPLE STEPS TO IMPLEMENT THE ADAPTIVE MESH

Following Huang and Russell [33], we use the alternate solution procedure to incorporate our dynamically adaptive mesh to the numerical computation of initial value problems.

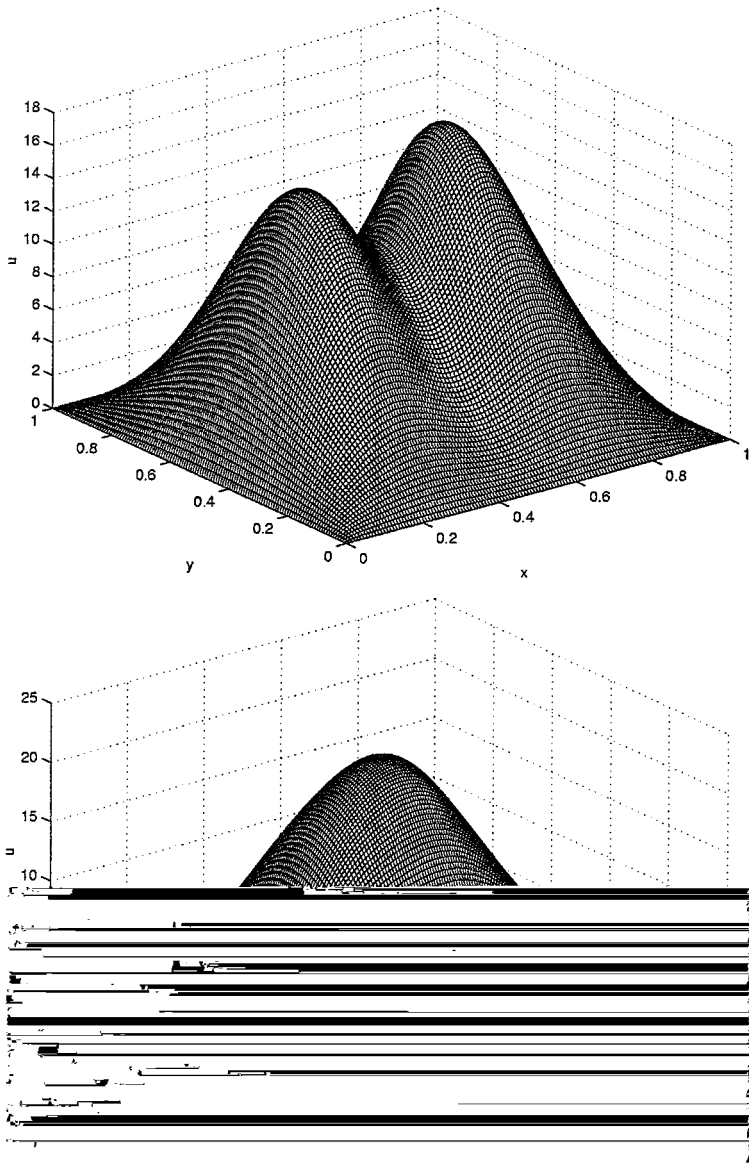

FIG. 6. Numerical solution to the semilinear heat equation with convection (17). (a) Solution at $t = 0.01$ and (b) at $t = 0.02$. $N = 128^2$.

As pointed out in [33], this procedure makes it very easy to combine the adaptive mesh computation with existing solvers for the underlying PDE. The implementation consists of two simple steps:

1. Express the underlying PDE in terms of the computational coordinates (ξ, η).
2. Integrate in time alternately the MMPDEs and the transformed PDE.

Except for the computation of the mesh, which we explain in detail at the end of this section, the algorithm is as in [33]. However, for completeness we now describe each step.

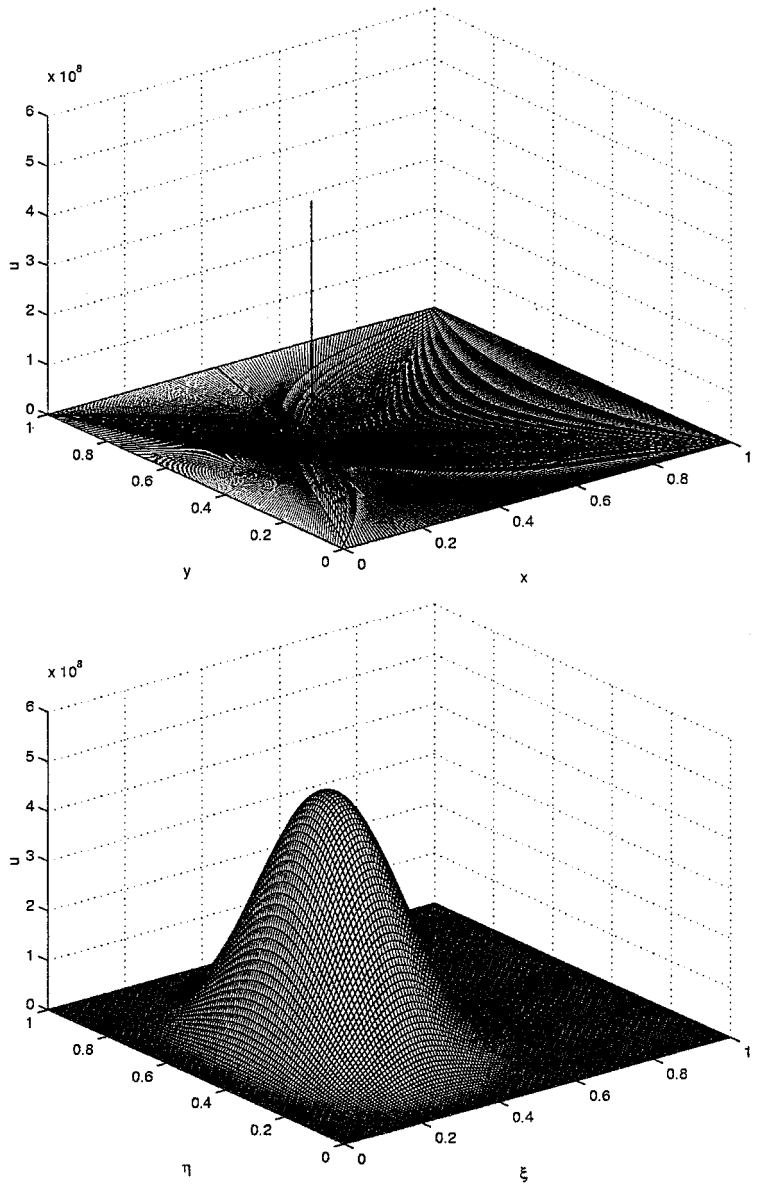

FIG. 7. Numerical solution to the semilinear heat equation (17) at $t = 0.0436$. (a) Solution in the physical space (top) and (b) in the computational space (bottom); $\max u = 5 \times 10^8$, $N = 128^2$.

4.1. *Transforming the Underlying PDE*

Assume that the underlying PDE is of the form

$$u_t = f(t, x, y, u, u_x, u_y, u_{xx}, u_{xy}, u_{yy}), \quad (x, y) \in \Omega_p \text{ and } t > 0, \tag{18}$$

with u satisfying $u(x, 0) = u_0(x)$ and appropriate boundary conditions. Here u can be vector-valued and thus (18) can be a system of physical PDEs. We first express (18) as

$$\dot{u} - u_x\dot{x} - u_y\dot{y} = f(t, x, y, u, u_x, u_y, u_{xx}, u_{xy}, u_{yy}), \tag{19}$$

where the "$\dot{}$" stands for the time derivative keeping ξ and η fixed. Note that we get an additional convection term accounting for the mesh motion. Here $(\dot{x}, \dot{y})$ is the mesh velocity.

Because both the mesh equations and the underlying PDE are solved in the computational domain, the spatial derivatives in (19) need to be written in terms of the computational variables using the following transformation formulas:

$$
\begin{aligned}
u_x &= \frac{1}{J}[(y_\eta u)_\xi - (y_\xi u)_\eta], \\
u_y &= \frac{1}{J}[-(x_\eta u)_\xi + (x_\xi u)_\eta], \\
u_{xx} &= \frac{1}{J}\left[\left(J^{-1}y_\eta^2 u_\xi\right)_\xi - (J^{-1}y_\xi y_\eta u_\eta)_\xi - (J^{-1}y_\xi y_\eta u_\xi)_\eta + \left(J^{-1}y_\xi^2 u_\eta\right)_\eta\right], \\
u_{xy} &= \frac{1}{J}\left[-(J^{-1}x_\eta y_\eta u_\xi)_\xi + (J^{-1}x_\xi y_\eta u_\eta)_\xi + (J^{-1}x_\eta y_\xi u_\xi)_\eta - (J^{-1}x_\xi y_\xi u_\eta)_\eta\right], \\
u_{yy} &= \frac{1}{J}\left[\left(J^{-1}x_\eta^2 u_\xi\right)_\xi - (J^{-1}x_\xi x_\eta u_\eta)_\xi - (J^{-1}x_\xi x_\eta u_\xi)_\eta + \left(J^{-1}x_\xi^2 u_\eta\right)_\eta\right],
\end{aligned}
$$

where $J = x_\xi y_\eta - x_\eta y_\xi$ is the Jacobian of the coordinate (mesh) transformation. Once these formulas are substituted into the right-hand side of (19), the underlying PDE can be discretized and solved in time alternately with the MMPDEs.

4.2. *The Alternate Solution Procedure*

In its simplest form, this procedure can be described as follows [33]. Given the approximate physical solution u^n and the adaptive mesh $\mathbf{x}^n = (x^n, y^n)$ at a time $t^n = n\Delta t$:

1. Compute the monitor function $w^n = w(x^n, y^n, u^n)$.
2. Compute the new mesh $\mathbf{x}^{n+1}$ by integrating the MMPDEs for one time step.
3. Compute the approximation of the physical solution u^{n+1} by integrating for one time step the transformed underlying PDE, using the new mesh $\mathbf{x}^{n+1}$ and the mesh velocity $\dot{\mathbf{x}}$.

At $t = 0$, the monitor function $w = w(x^0, y^0, u_0)$ is computed and the MMPDEs are solved numerically to steady state to obtain a good initial adaptive mesh. To generate the initial mesh at $t = 0$, one can use the uniform grid as the initial condition for the time dependent mesh equation.

4.3. *Solving the Mesh Equations*

We now describe how to solve efficiently the MMPDEs (11) and (12). Note that this is a system of nonlinear elliptic equations and the elliptic coefficient is the monitor function w. Because high-order derivatives of the mesh map are hidden in w, a straightforward discretization of (11) and (12) fails because of a severe time step stability constraint. A natural alternative would be the ADI method but it also fails in practical situations because of the strong nonlinearity. There is however a simple, efficient, and robust way to solve the mesh equations. This is the following semi-implicit discretization [24],

$$
\frac{x^{n+1} - x^n}{\Delta t} = a\Delta_h' x^{n+1} + \nabla_h' \cdot (w^n \nabla_h' x^n) - a\Delta_h' x^n, \tag{20}
$$

$$\frac{y^{n+1}-y^n}{\Delta t} = a\Delta_h' y^{n+1} + \nabla_h' \cdot (w^n \nabla_h' y^n) - a\Delta_h' y^n, \tag{21}$$

where $a = \max w^n$. Here Δ_h' and ∇_h' are the standard second-order approximations to the operators Δ' and ∇' (the Laplacian and the gradient with respect to (ξ, η)), respectively. Note that equations are solved in a square computational domain with a uniform grid. Thus, the adaptive mesh can be obtained with fast solvers at the cost of inverting a Laplacian per time step, i.e., in $O(N)$ operations with N being the total number of grid points.

Note also that the discretization of the mesh equations does not affect the accuracy of the underlying physical solution in an analytical sense. In fact, it is common to use some temporal or spatial *smoothing* on the monitor function or directly on the mesh map (x, y) to obtain smoother meshes. As in [33], we apply the following low-pass filter four times to the monitor function:

$$\begin{aligned} w_{i,j} \leftarrow{}& \frac{4}{16} w_{i,j} + \frac{2}{16}(w_{i+1,j} + w_{i-1,j} + w_{i,j+1} + w_{i,j-1}) \\ &+ \frac{1}{16}(w_{i-1,j-1} + w_{i-1,j+1} + w_{i+1,j-1} + w_{i+1,j+1}). \end{aligned} \tag{22}$$

4.4. *The Numerical Method for the Semilinear Heat Equation*

To solve the semilinear heat equation in conjuction with the adaptive mesh, we first write it as

$$\dot{u} = J^{-1}\nabla' \cdot (A\nabla' u) + u_x \dot{x} + u_y \dot{y} + f(u), \tag{23}$$

where A is a positive definite matrix with the transformation coefficients for the Laplacian and $u_x = J^{-1}[(y_\eta u)_\xi - (y_\xi u)_\eta]$ and $u_y = J^{-1}[-(x_\eta u)_\xi + (x_\xi u)_\eta]$. On (23) we perform the semi-implicit time discretization,

$$\frac{u^{n+1}-u^n}{\Delta t} = b\Delta' u^{n+1} + J^{-1}\nabla' \cdot (A\nabla' u^n) - b\Delta' u^n + u_x^n \dot{x}^n + u_y^n \dot{y}^n + f(u^n), \tag{24}$$

where $b = \max \frac{\rho(A)}{|J|}$ with $\rho(A)$ being the spectral radius of A. The term $b\Delta' u^{n+1}$ serves as a majorizing preconditioner which can be inverted easily, just as in the discretization (20) and (21) for the mesh equations. The spatial discretization is standard second order. An important thing to note is that solving the semilinear heat equation requires adaptive time stepping as well. We reduce Δt according to the leading growth rate of the solution in the form $\Delta t = \Delta t_0 / \| f'(u) \|_\infty$.

5. BOUSSINESQ CONVECTION AND POTENTIAL SINGULARITY FORMATION

The Boussinesq equations are based on the observation that there are flows for which the temperature varies little, and therefore the density varies little, yet in which the buoyancy drives the motion. For a layer of this type of fluid, the density ρ obeys the relation [23]

$$\rho = \rho_0[1 - \alpha(T - T_0)], \tag{25}$$

where T denotes the temperature, α is the constant coefficient of volume expansion, and ρ_0 is the density at T_0, the temperature at the bottom of the layer. We assume that T_0 is

the highest temperature as in Rayleigh–Bénard thermal convection experiments. Because for a typical liquid $(\rho - \rho_0)/\rho_0 = \alpha(T_0 - T) \ll 1$, the density variations are neglected everywhere except in the buoyancy term. The motion of a layer of *inviscid* Boussinesq fluid is described by the equations

$$\mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla\left(\frac{p}{\rho_0} + gy\right) - \alpha g(T_0 - T)\mathbf{j}, \tag{26}$$

$$T_t + \mathbf{u} \cdot \nabla T = 0, \tag{27}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{28}$$

where $\mathbf{u}$ represents the velocity field, p is the pressure, g is the gravitational constant, and $\mathbf{j}$ is the unit vector in the upward vertical direction. This type of flow is relevant to the study of atmospheric and oceanographic turbulence and in many other situations where stratification plays a significant role.

In 2-D, which is our case of interest, it is convenient to write this system of equations in the stream function-vorticity formulation. Letting $\theta = T_0 - T$ and taking the curl on Eq. (26) we have the following system of scalar equations:

$$\omega_t + \mathbf{u} \cdot \nabla \omega = -\bar{g}\theta_x, \tag{29}$$

$$\theta_t + \mathbf{u} \cdot \nabla \theta = 0, \tag{30}$$

$$-\Delta \psi = \omega, \tag{31}$$

$\omega = v_x - u_y$ (not to be confused with the monitor function w) is the vorticity and $\bar{g} = \alpha g$ is a scaled gravity constant. The stream function ψ determines the velocity $\mathbf{u} = (u, v)$ as

$$u = \psi_y, \quad v = -\psi_x. \tag{32}$$

It is well-known that the Boussinesq equations are similar to those describing 3-D axi-symmetric Euler flows with swirl (nonzero azimuthal velocity); see e.g. [42, 43]. Because of this analogy, Boussinesq convection provides, like the axi-symmetric flow, a computationally feasible (two-dimensional) framework to investigate potential finite-time singularity formation, a mystery yet to be solved. Grauer and Sideris [29] were the first to explore the possibility of finite-time singularities in the axi-symmetric Euler flow. Their work has stimulated a very dynamic research in this direction (e.g. [14, 26, 28, 30, 34, 42, 43]).

The problem is difficult. While short-time existence can be shown for sufficiently smooth conditions, it is unclear if a solution can lose its regularity and become singular in finite time. The key issue is the presence of a vorticity production mechanism, namely $-\bar{g}\theta_x$ in the Boussinesq equations. Following Beale, Kato, and Majda [4], E and Shu [26] show that if a singularity develops in the Boussinesq flow at a finite time t^*, such that $\|\mathbf{u}(\cdot, t^*)\|_m + \|\theta(\cdot, t^*)\|_m = +\infty$, then

$$\int_0^{t^*} |\omega(\cdot, t)|_\infty \, dt = +\infty \quad \text{and} \quad \int_0^{t^*}\int_0^t |\theta_x(\cdot, s)|_\infty \, ds \, dt = +\infty, \tag{33}$$

where $\|f(\cdot)\|_m$ denotes the usual Sobolev m-norm and $|f(\cdot)|_\infty = \max_{\mathbf{x} \in R^2} |f(\mathbf{x})|$. It is assumed that $m > 2$ and that the initial conditions $\mathbf{u}(\mathbf{x}, 0)$ and $\theta(\mathbf{x}, 0)$ lie in $H^m(R^2)$.

In particular, this result tells us the minimum rate of self-similar blow-up if this occurs [26]:

$$|\omega(\cdot, t)|_\infty \sim \frac{c_1}{t^* - t}, \tag{34}$$

$$|\theta_x(\cdot, t)|_\infty \sim \frac{c_2}{(t^* - t)^2}. \tag{35}$$

There are several numerical studies of possible singularity formation in 3-D Euler flows and in 2-D Boussinesq convection [6, 8, 9, 18, 26, 28–30, 34, 42, 43]). While the studies differ in their conclusions, they all show that this is an extremely difficult problem both numerically and analytically. Vorticity production rapidly leads to the formation of small scales and the computations quickly run out of resolution. Thus, an adaptive mesh strategy is absolutely necessary. The early computations of Pumir and Siggia [42] already use a simple form of adaptive mesh via a coordinate transformation of a fixed type. However, their mesh does not adjust to the geometry of the solution but mainly concentrates at the point where the vorticity is maximum. Outside this region, the flow is not well resolved and for an incompressible fluid it is essential to resolve the flow globally to avoid energy losses. Recently, Grauer, Marliani, and Germaschewski [28] have performed an outstanding computation of a fully 3-D ideal incompressible flow using adaptive mesh refinements (AMR). However, one of the drawbacks of their method is the artificial numerical dissipation introduced by the frequent interpolation associated with the AMR technique.

The accurate computation of inviscid Boussinesq flow is thus challenging and constitutes a real demanding test for our dynamically adaptive mesh. Here, we explore an interesting scenario for the potential formation of a finite-time singularity by considering strongly layered convection in a channel.

6. IMPLEMENTATION DETAILS FOR BOUSSINESQ FLOW IN A CHANNEL

We now discuss a few implementation issues specific to the Boussinesq equations (26)–(28) for a channel geometry. In this case, the flow is bounded by horizontal walls on the top and bottom of the layer, and it is assumed to be periodic in the horizontal direction.

As explained in Sections 3 and 4, initially the mesh equations have to be solved to steady state but afterwards only for one time step at a time. Considering that the flow is periodic in the horizontal direction we impose $x(\xi, \eta) - \xi$ to be periodic in ξ. The implicit discrete mesh equations (20) and (21) are inverted by applying the Fast Fourier transform (FFT) in ξ, and then using a tridiagonal solver on the resulting system. We take (1-D) uniform meshes as boundary conditions for the mesh map on the top and bottom walls. More general boundary conditions for the mesh can be obtained by solving corresponding 1-D mesh equations. Our criterion for steady state is that consecutive iterations differ by less than 10^{-10}. The number of iterations to get to steady state varies depending on the smoothness of the initial data. This is a one-time overhead of our adaptive grid method.

Once we write the Boussinesq equations in terms of the derivatives in the computational coordinates (ξ, η) and transform the time derivative as in (19), we do a second-order central difference discretization in space. With some additional work, higher order discretizations are also feasible.

To compute the flow velocity (u, v) we need to solve first for the stream function ψ. In the computational variables ξ and η, the stream function equation (31) becomes an elliptic equation with variable coefficients. This equation is subjected to Dirichlet boundary conditions ($\psi = 0$) on the top and bottom of the computational domain and periodic boundary conditions in the horizontal direction. For this particular initial condition, vorticity remains to be zero on the top and bottom of the computational domain until the plume reaches the boundary. For this reason, we have applied zero vorticity boundary condition throughout our computations.

We construct an efficient solver for the transformed stream function equation by preconditioning the Conjugate Gradient (CG) method with a robust multigrid method that uses matrix-dependent prolongation [48]. This particular multigrid handles efficiently the high-contrast variable coefficients introduced by the mesh map. The CG method corrects locally the solution to enforce the horizontal periodic boundary conditions. Our stopping criterion for the CG method is that the maximum difference between consecutive iterations is less than 10^{-8}. The multigrid tolerance is set to 10^{-7}. Typically it takes one or two CG iterations and the multigrid performs also one or two iterations every time it is called. Thus, ψ is effectively obtained in $O(N)$ operations per time step.

After solving for ψ, we compute the flow velocity from (32) using centered differences. The alternate solution time-marching procedure is then applied using a second order Adams–Bashforth method. The mesh velocity is also computed with second-order accuracy as $\dot{\mathbf{x}} = (\mathbf{x}^{n+1} - \mathbf{x}^{n-1})/(2\Delta t)$. Thus, the overall method is second order both in space and time. Higher order multistep or Runge–Kutta methods can be easily implemented.

To reduce the dispersive error inherent in centered differences, we filter θ and ω separately in ξ and η every time step using the following fourth-order filter [38]: $u_j \leftarrow \frac{1}{16}(-u_{j-2} + 4u_{j-1} + 10u_j + 4u_{j+1} - u_{j+2})$. This filter can effectively eliminate the small amplitude mesh-scale oscillations without affecting the accuracy of the physical solution. The second-order filtering $u_j \leftarrow \frac{1}{4}(u_{j-1} + 2u_j + u_{j+1})$, which is used frequently in the literature, seems to introduce excessive numerical diffusion to the physical solution.

7. NUMERICAL RESULTS

We present in this section numerical results for Boussinesq convection without viscosity regularization using our dynamically adaptive mesh. Throughout the numerical experiments the scaled gravity constant $\bar{g}$ is taken to be 10. We begin by describing our initial conditions which correspond to a multilayer fluid. We then examine the detailed time evolution of the flow.

7.1. *The Initial Conditions*

As initial data we take $\omega(\mathbf{x}, 0) \equiv 0$ and $\theta(\mathbf{x}, 0)$ defining a stratified fluid with three constant regions θ_1, θ_2, and $\bar{\theta} = (\theta_1 + \theta_2)/2$ connected by two thin layers in the following form:

$$\theta(x, y, 0) = \begin{cases} \theta_2 + (\bar{\theta} - \theta_2) H_\delta(0.5 + y_s(x) - y) & \text{if } y \geq 0.5, \\ \theta_1 + (\bar{\theta} - \theta_1) H_\delta(y + y_s(x) - 0.5) & \text{if } y < 0.5, \end{cases} \tag{36}$$

where

$$y_s(x) = \delta + \epsilon + \epsilon \sin 2\pi(x + 0.75), \tag{37}$$

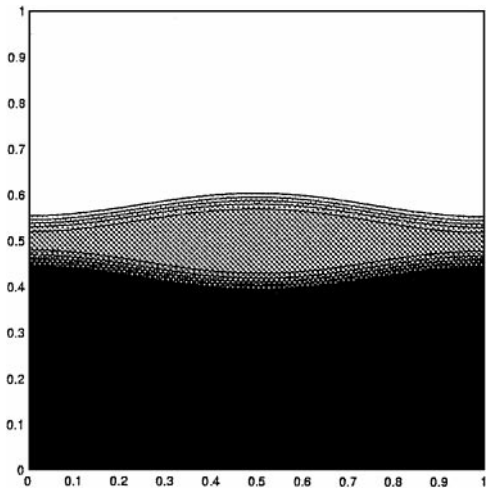

FIG. 8. Initial temperature distribution shown in a filled contour (level set) plot.

and $H_\delta(x)$ is mollified Heaviside function given by [17]:

$$H_\delta(x) = \begin{cases} 0 & \text{if } x < -\delta, \\ (x+\delta)/(2\delta) + \sin(\pi x/\delta)/(2\pi) & \text{if } |x| \le \delta, \\ 1 & \text{if } x > \delta. \end{cases} \tag{38}$$

Here, we take $\theta_1 = -1$, $\theta_2 = 1$, and $\bar{\theta} = 0$. By setting $\delta = 0.025$ and $\epsilon = 0.04$, we obtain two thin symmetric layers saparating smoothly the three constant values of θ. Hereafter we will refer to θ as the temperature field.

Figure 8 shows the temperature distribution at $t = 0$. The initial adaptive mesh is generated by solving to steady state equations (20) and (21) using the monitor function $w = \sqrt{1 + |\nabla' \theta|^2}$. We choose the scaling coefficient $\beta = 1$ here to avoid excessive grid deformation dynamically resulting from the global coupling nature of the incompressible flow. Figure 9 presents the initial adaptive mesh for a region covering the two central thin layers. The mesh shown was obtained using $N = 128^2$ points but in all the

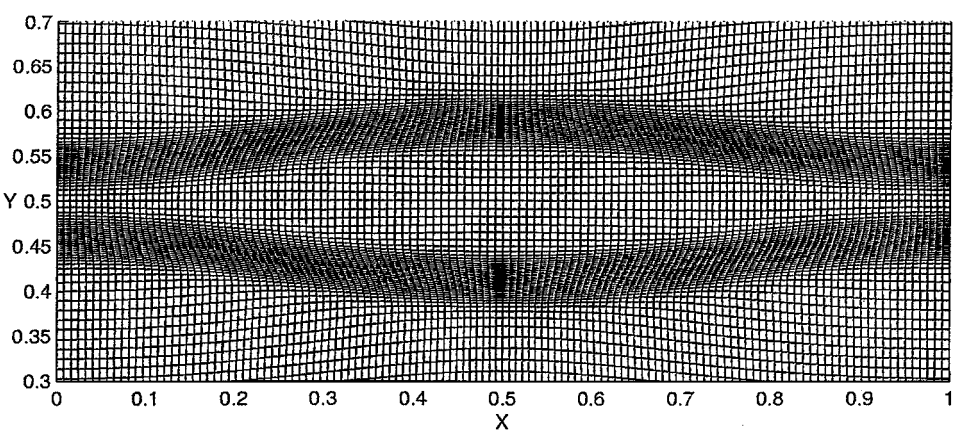

FIG. 9. Initial adaptive mesh covering the central fluid layers for $N = 128^2$.

TABLE I
Time Stepping History

Time interval	Δt
$0.0 - 0.5$	1.0×10^{-4}
$0.5 - 0.6$	5.0×10^{-5}
$0.6 - 0.7$	2.5×10^{-5}
$0.7 - 0.8$	1.25×10^{-5}

computations that follow, we use $N = 512^2$ points in the whole computational domain $[0, 1] \times [0, 1]$.

7.2. *Flow Evolution and Small-Scale Structure Development*

We now present the time evolution of the layered Boussinesq inviscid fluid with initial zero vorticity and temperature given by (36)–(38). Although for these particular conditions, the flow has four-fold symmetry, we do not use this property to achieve higher resolution but instead compute the solution in the whole domain, $[0, 1] \times [0, 1]$. We take $N = 512^2$ points, and Δt is reduced adaptively to comply with the CFL condition and for accuracy sake. We start with $\Delta t = 1 \times 10^{-4}$ and end the computations with $\Delta t = 1.25 \times 10^{-5}$. Table I gives a detailed record of the time stepping we employ. Convergence runs using 128^2 and 256^2 for $t \leq 0.4$ were also performed confirming second-order accuracy. For the exact solutions, the maximum and minimum values of θ are preserved in time. This provides a useful diagnostics for the numerics. Our computations maintain the global extrema of θ within three to four digits for the majority of the computed time interval. All the computations were carried out in a 450 MHz PC computer using double precision.

The time evolution of both the temperature and the vorticity fields is depicted in Figs. 10 and 11. At $t = 0.5$ (Figs. 10a and 10b), the initial $\theta = 0$ central region of the fluid has become a rounded bubble with a thin front. The vorticity field at this time is concentrated into four small symmetric regions with alternate signs, producing a fast vertical convection and squeezing the flow in at the center. The vorticity is zero outside the four small regions. While the maximum vorticity is attained at the steepest parts of the bubble, $|\nabla\theta|_\infty \equiv \max\|\nabla\theta\|_{L^2}$ occurs at the thinnest section of the arms. At $t = 0.6$ (Figs. 10c and 10d), the flow central region begins to evolve into two symmetric bubbles with a sharp cap. The maximum vorticity has almost doubled its value, from 36.21 at $t = 0.5$ to 62.13 at $t = 0.6$.

A rapid transition then follows and the bubbles unfold into thermal plumes with a mushroom shape structure as Fig. 11a shows. At $t = 0.7$ the support of the vorticity is already collapsing to the sides and the stem of the plumes (Fig. 11b) in extremely thin layers. Across these thin layers the vorticity field has a large and sharp variation. The maximum vorticity at $t = 0.7$ is 135.34. In the axi-symmetric flow analogy, the thin vortical layers correspond to vortex sheets in an incipient roll-up. At $t = 0.8$ (Figs. 11c and 11d), the stem connecting the two mushroom plumes which is almost collapsing encloses the region of maximum vorticity (232.40 at this time).

Figure 12 gives a close-up of the dynamically adaptive mesh around one roll of the upper thermal plume at $t = 0.8$. The adaptive mesh is able to follow closely the fast flow

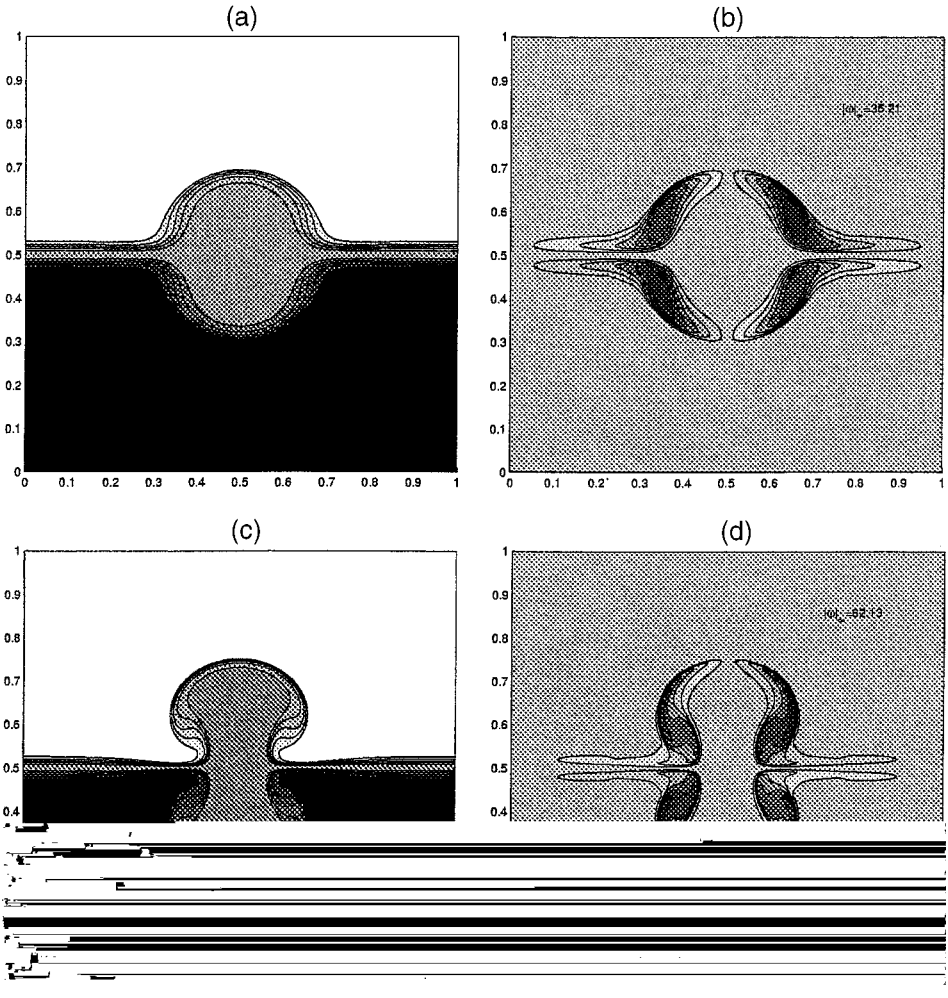

FIG. 10. Temperature and vorticity filled contour plots at $t = 0.5$ and $t = 0.6$. (a) θ at $t = 0.5$, (b) ω at $t = 0.5$, (c) θ at $t = 0.6$, and (d) ω at $t = 0.6$. Ten contours (level sets) are shown in each plot. The vorticity support is concentrated in four small symmetric regions among which the vorticity alternates signs $(+ - / - +)$. Within each support region, the darker the area the larger the vorticity in absolute value.

dynamics maintaining good adaptivity in regions of complex geometry, even up to this very singular stage. In fact, as Table II demonstrates, the more singular the solution gets the higher the mesh compression ratio (uniform grid size to smallest adaptive grid size). At $t = 0.8$ we obtain a compression ratio close to 9 giving an effective resolution corresponding to that of a 4600^2 point uniform mesh. But any compression ratio is meaningless if the solution is not globally resolved as it is required in incompressible flows. Our adaptive mesh not only achieves high compression ratios but, as Fig. 12 demonstrates, it also covers all the most singular regions with a sufficiently spread fine grid. As a result, the solution is effectively resolved globally even when it becomes extremely localized and nearly singular.

We now examine in more detail the latest stage of the fluid motion and the time behavior of important flow quantities. Figure 13 gives a close-up of 10 vorticity contours

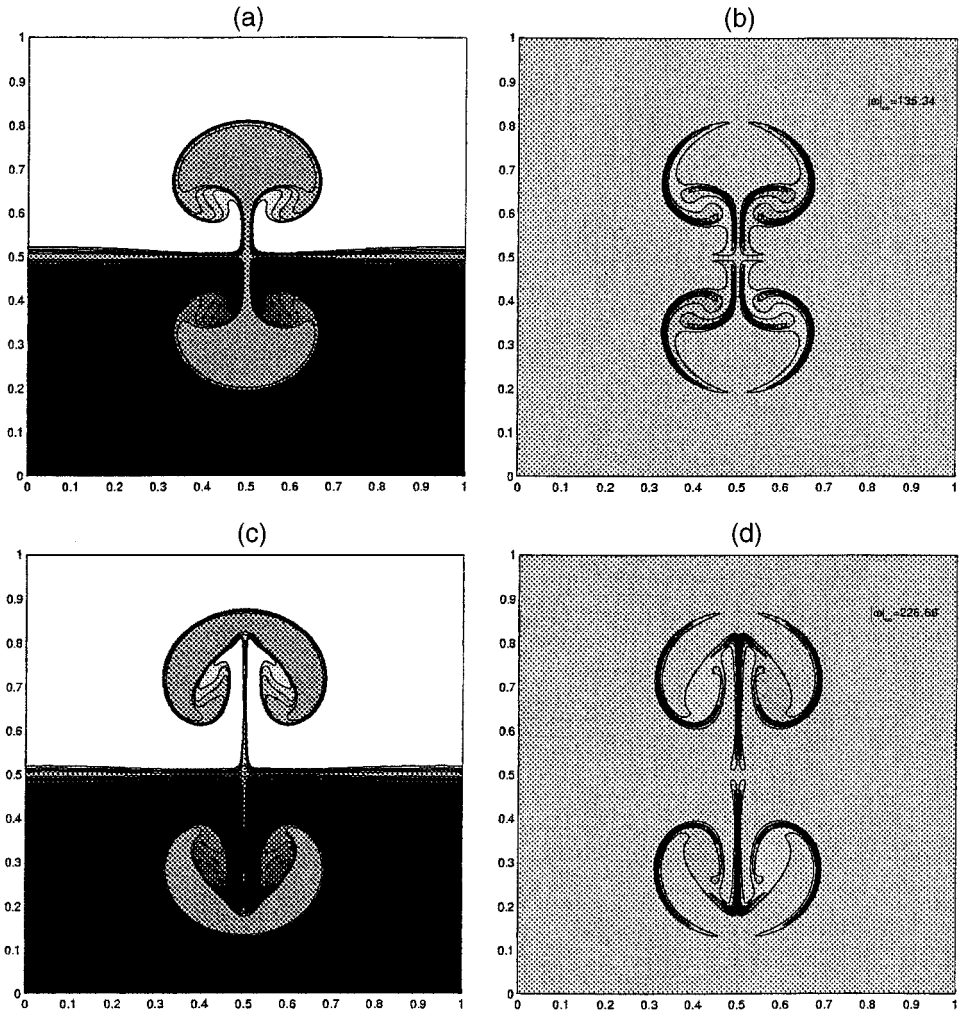

FIG. 11. Temperature and vorticity filled contour plots at $t = 0.7$ and $t = 0.8$. (a) θ at $t = 0.7$, (b) ω at $t = 0.7$, (c) θ at $t = 0.8$, and (d) ω at $t = 0.8$. Ten contours (level sets) are shown in each plot. The vorticity support is concentrated in four small symmetric regions among which the vorticity alternates signs $(+ - / - +)$. Within each support region, the darker the area the larger the vorticity in absolute value.

around the upper plume at $t = 0.8$ in both the physical and the computational space. The physical length scale is so small that the contours appear to be collapsing at the sides and stem of the plume in Fig. 13a. However, in the computational space (Fig. 13b), the vorticity has a much wider support. As a result, the contours can be clearly distinguished and found to be well resolved. The maximum of vorticity occurs on the stem at the point marked with a star in Fig. 13 and the minimum at the mirror image of this point. Figure 14 presents a slice of the vorticity at $t = 0.8$ through its maximum point both in the physical and in the computational space. As Fig. 14a demonstrates that the vorticity is strongly concentrated in a narrow support and shows two extremely large and sharp spikes around the center. These spikes appear much smoother in the computational space as shown in Fig. 14b.

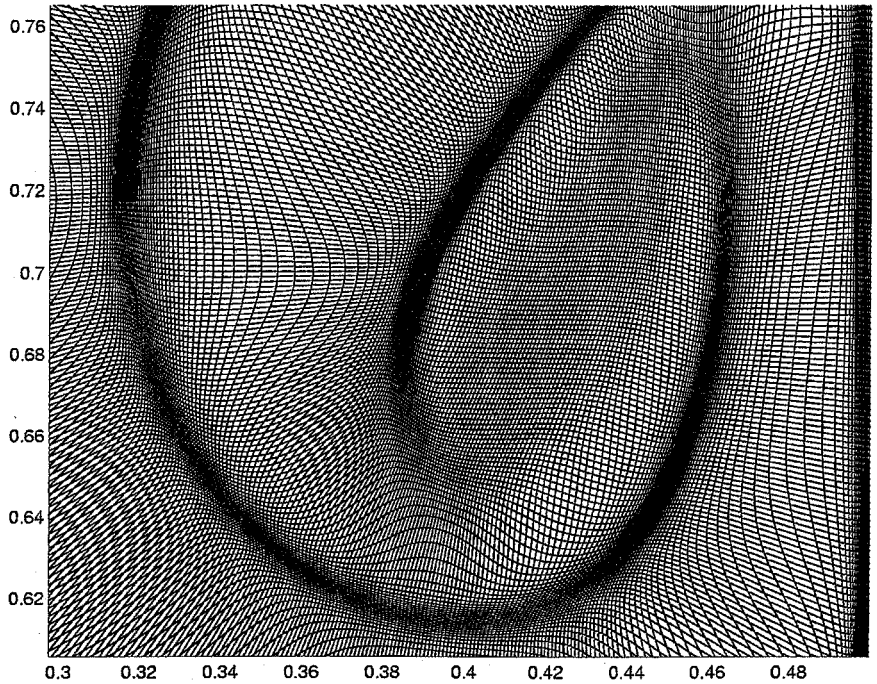

FIG. 12. Close-up of the adaptive mesh around one roll of the upper thermal plume at $t = 0.8$. $N = 512^2$.

Is the maximum vorticity growing fast enough to develop a finite-time singularity? Figure 15 shows the growth in time of the maximum vorticity plotted in a semi-log scale. After a rapid transient stage at the beginning, $|\omega|_\infty$ grows clearly exponentially (linear behavior in the semi-log plot) up to $t = 0.5$. Then the growth accelerates but still at a seemingly exponential rate. Just before $t = 0.7$, the growth of the maximum vorticity, which occurs on the sides of the plumes, begins to saturate. Soon after this, the maximum vorticity shifts to the stem of the plumes and continues to grow for a short time before showing signs of saturation close to $t = 0.8$. It is conceivable that the apparent saturation is due to the very simple geometry of the flow in the vicinity of the maximum point. This situation is analogous to that occurring when two parallel vortex tubes are placed close to each other. The axial strain saturates as the core of the tubes greatly deforms to avoid reconnection [2, 41, 45]. The importance of nontrivial geometry for potential

TABLE II
Mesh Compression Ratios

Time	Compression
0.0	4.34
0.5	5.53
0.6	6.41
0.7	7.44
0.8	8.83

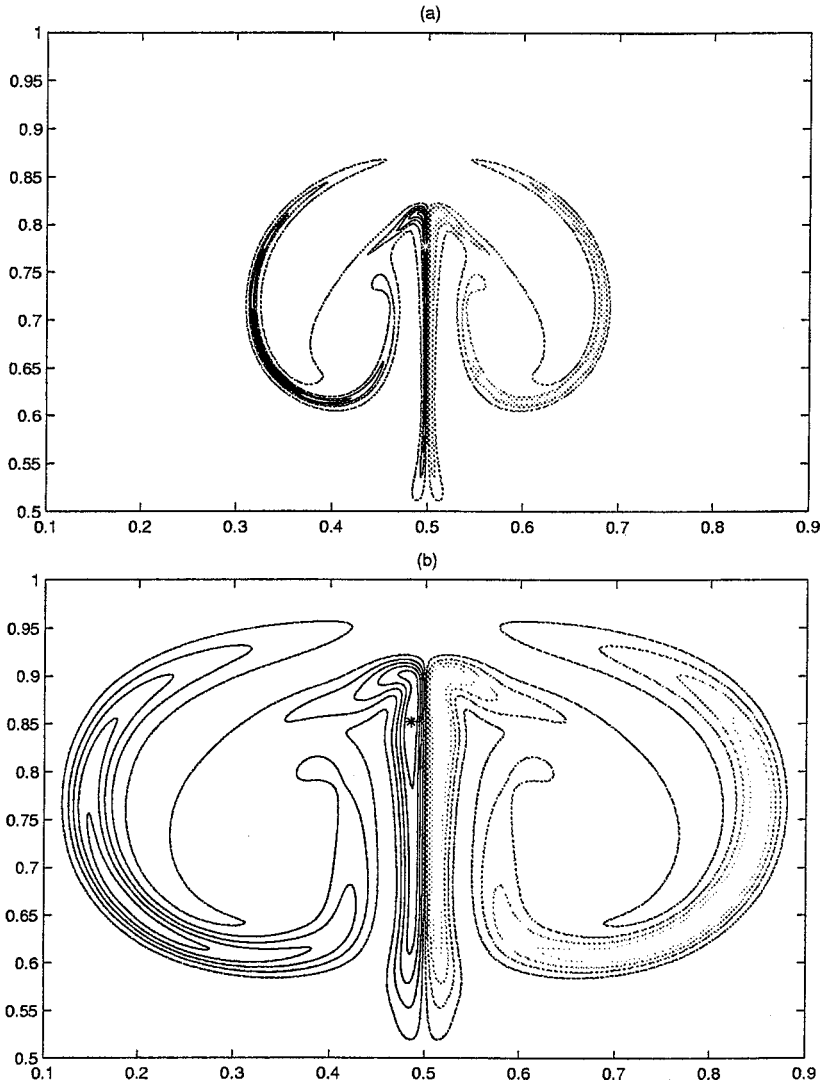

FIG. 13. Vorticity contours in the upper plume at $t = 0.8$ in (a) the physical space and (b) the computational space. The stars mark the point of maximum vorticity $|\omega|_\infty = 226.68$. The vorticity is zero on the symmetry line $x = 0.5$, is positive on the region enclosed by the left contours, and negative on the right counterpart.

finite-time singularity development was suggested by Constantin, Majda, and Tabak [20] for quasi-geostrophic flows and by Constantin, Fefferman, and Majda [19] for the 3-D Euler equations.

The different phases of the flow can be also connected to the behavior of $|\nabla\theta|_\infty$ and of the vorticity generating term $|\theta_x|_\infty$. Figure 16 shows the growth in time of these quantities. Two phases stand out: the accelerated growth of $|\nabla\theta|_\infty$ from $t = 0.50$ to $t = 0.69$ and the apparent saturation beginning at $t = 0.75$.

In summary, the time growth of $|\omega|_\infty$, $|\nabla\theta|_\infty$, and $|\theta_x|_\infty$ gives no indication of a finite-time singularity development for the initial conditions we consider. But the numerics support the importance of the local geometry for potential singularity formation.

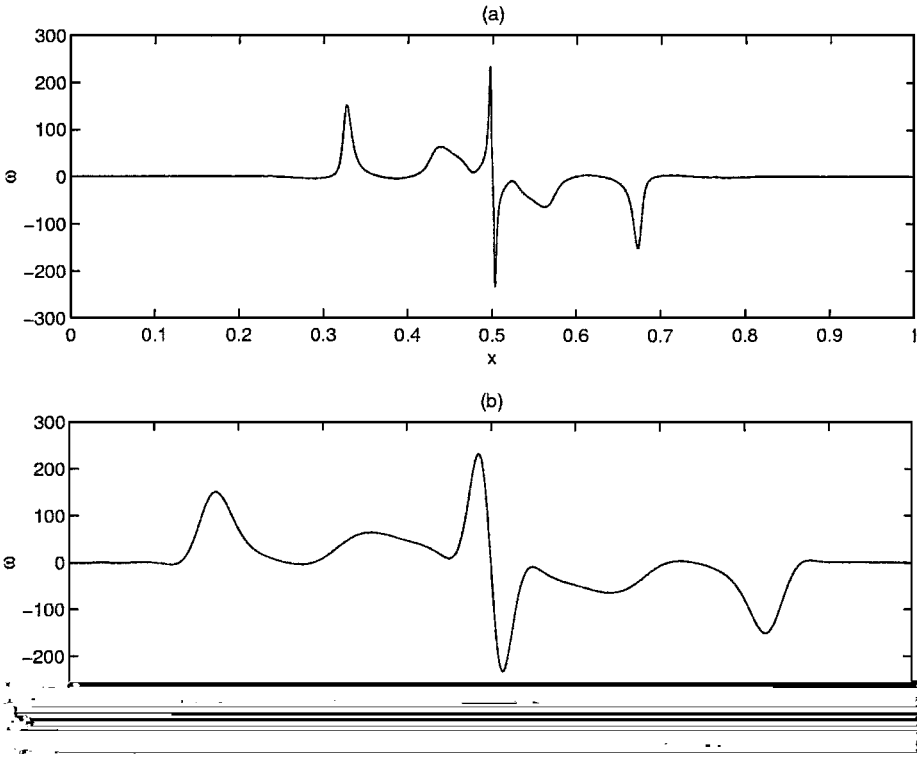

FIG. 14. Slice of the vorticity at $t = 0.8$ through its maximum at $\eta = 0.8521$ in (a) the physical space and (b) the computational space.

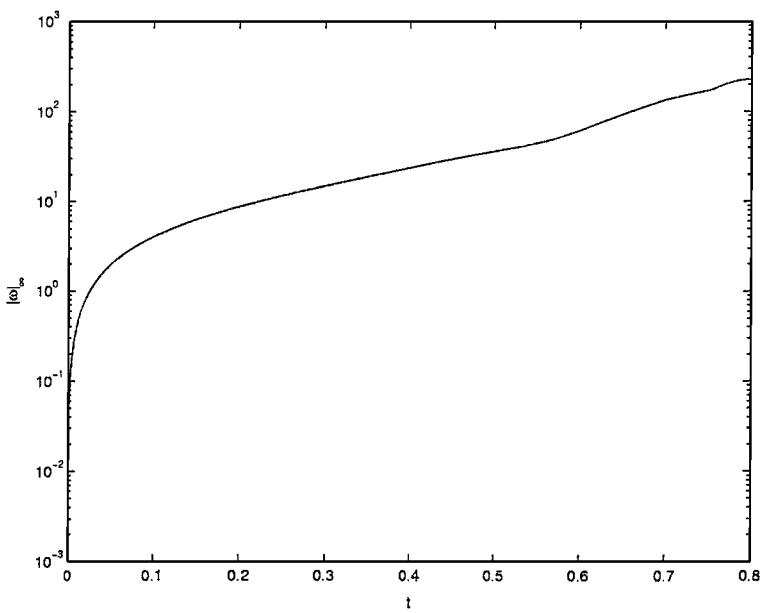

FIG. 15. Growth of the maximum of vorticity in time.

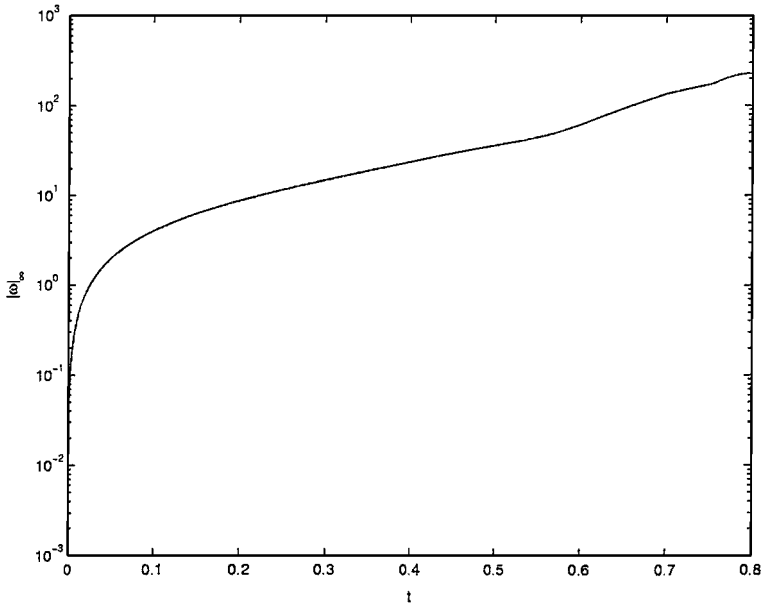

FIG. 16. Growth of $|\nabla\theta|_\infty$ and $|\theta_x|_\infty$ in time.

8. CONCLUSIONS

We have presented in this work a new dynamically adaptive mesh generator for computing time-dependent solutions that can develop singular or near singular behavior. The efficient mesh map is obtained as the solution of a set of simple nonlinear PDEs which can be solved at minimal cost. The overall dynamic mesh strategy is easy to implement, avoids interpolation, and can be used in conjunction with existing time-integration solvers.

Although the focus of application for the adaptive mesh here was the problem of inviscid Boussinesq convection, we have also demonstrated with a pair of examples that the mesh can effectively follow 2-D finite-time blowing-up behavior without losing its very high adaptivity and thus capturing the singularity accurately.

Inviscid Boussinesq convection of an incompressible fluid is a challenging problem both analytically and numerically. Because of the complex dynamic development of small scales and the solution's rapid loss of regularity, Boussinesq convection pushes any adaptive mesh strategy to the limit. Our adaptive mesh follows the complex evolution of the almost singular flow with very good adaptivity. Moreover, the numerical solution remains stable throughout the entire computation. In the numerical study, we have found that the baroclinically generated vorticity becomes highly localized in thin layers and its maximum appears to be growing exponentially in time. Using the axi-symmetric flow analogy, the thin layers correspond to vortex sheets that roll up and form the envelope of thermal plumes. The maximum vorticity ultimately develops in the stem of the plumes, a geometrically simple region that appears to lead to the saturation of the vorticity growth. This behavior supports the theory about the importance of a nontrivial geometry for the potential finite-time singularity formation.

At present, our adaptive mesh has not incorporated other mesh attributes such as skewness and orthogonality, that may be important in other applications. It seems plausible to include

these additional properties, starting again by a variational principle in the computational domain with the corresponding extra terms as in [10, 11]. In general, the monitor function in the mesh generator should be problem-dependent as this function ultimately determines the compression and deformation of the mesh.

Through numerical experience we have found that the mesh generator produces meshes of good quality in rectangular domains. However, because the nonlinear mesh PDEs (9) have the same form as the linear PDEs of the length functional mesh [35], it is conceivable that our mesh generator may fail in some instances of nonconvex domain as is the case for the length functional mesh [25].

It seems also natural to combine the adaptive mesh with upwinding or ENO solvers for free boundary problems, for example in conjunction with capturing schemes such as the Level Set Method. This is currently under investigation and will be reported elsewhere.

ACKNOWLEDGMENTS

The authors thank Xiao-Ping Wang for insightful conversations during the early stage of this work. We also thank Bob Russell for a number of valuable comments and suggestions and for bringing to our attention important issues related to the performance of existing variational mesh generators and their historical development. Research was in part supported by National Science Foundation Grant DMS-9704976 and Army Research Office Grant DAAD19-99-1-0141.

REFERENCES

1. S. Adjerid and J. E. Flaherty, A moving finite element method with error estimation and refinement for one-dimensional time dependent partial differential equations, *SIAM J. Numer. Anal.* **23**, 778 (1986).
2. C. Anderson and C. Greengard, The vortex ring merger problem at infinite Reynolds-number, *Commun. Pure Appl. Math.* **42**(8), 1123 (1989).
3. M. J. Baines, *Moving Finite Elements* (Claredon, Oxford, 1994).
4. J. T. Beale, T. Kato, and A. Majda, Remarks on the breakdown of smooth solutions for the 3D incompressible Euler equations, *Commun. Math. Phys.* **94**, 61 (1984).
5. J. Bebernes and D. Eberly, *Mathematical Problems from Combustion Theory*, Applied Mathematical Sciences (Springer-Verlag, New York, 1989).
6. J. B. Bell and D. L. Marcus, Vorticity intensification and transition to turbulence in the 3-dimensional Euler equations, *Commun. Math. Phys.* **147**(2), 371 (1992).
7. M. J. Berger and P. Collela, Local adaptive mesh refinement for shock hydrodynamics, *J. Comput. Phy.* **82**, 62 (1989).
8. M. E. Brachet, D. I. Meiron, S. A. Orszag, B. G. Nickel, R. H. Morf, and U. Frisch, Small-scale structure of the Taylor–Green vortex, *J. Fluid Mech.* **130**, 411 (1983).
9. M. E. Brachet, M. Meneguzzi, A. Vincent, H. Politano, and P. L. Sulem, Numerical evidence of smooth self-similar dynamics and possibility of subsequent collapse for 3-dimensional ideal flows. *Phys. Fluids A* **4**(12):2845 (1992).
10. J. U. Brackbill, An adaptive grid with directional control, *J. Comput. Phys.* **108**, 38 (1993).
11. J. U. Brackbill and J. S. Slatzman, Adaptive zoning for singular problems in two dimensions, *J. Comput. Phys.* **46**, 342 (1982).
12. C. J. Budd, S. Chen, and R. D. Russell, New self-similar solutions of the nonlinear Schrödinger equation with moving mesh computations, *J. Comput. Phys.* **152**, 756 (1999).
13. C. J. Budd, W. Huang, and R. D. Russell, Moving mesh methods for problems with blow-up, *SIAM J. Sci. Comput.* **17**(2), 305 (1996).
14. R. E. Caflisch, Singularity formation for complex solutions of the 3D incompressible Euler equations, *Physica D* **67**(1-3), 1 (1993).

15. J. E. Castillo, A discrete variational grid generation method, *SIAM J. Sci. Stat. Comput.* **12**, 454 (1991).
16. J. E. Castillo and J. S. Otto, A practical guide to direct optimization for planar grid-generation, *Comput. Math. Appl.* **37**(9), 123 (1999).
17. Y. C. Chang, T. Y. Hou, B. Merriman, and S. Osher, A level set formulation of Eulerian interface capturing methods for incompressible fluid flows, *J. Comput. Phys.* **124**, 449 (1996).
18. A. J. Chorin, The evolution of a turbulent vortex, *Commun. Math. Phys.* **83**(4), 517 (1982).
19. P. Constantin, C. Fefferman, and A. J. Majda, Geometric constraints on potentially singular solutions for the 3-D Euler equations, *Commun. Part. Diff. Eq.* **21**(3-4), 559 (1996).
20. P. Constantin, A. J. Majda, and E. G. Tabak, Singular front formation in a model for quasigeostrophic flow, *Phys. Fluids* **6**(1), 9 (1994).
21. C. de Boor, *Good Approximation by Splines with Variable Knots II*, Springer Lecture Notes Series (Springer-Verlag, Berlin, 1973).
22. E. A. Dorfi and L. O'C. Drury, Simple adaptive grids for 1-D initial value problems, *J. Comput. Phys.* **69**, 175 (1987).
23. P. G. Drazin and W. H. Reid, Hydrodynamic stability. Cambridge monographs on mechanics and applied mathematics (Cambridge Univ. Press, New York, 1981).
24. T. Dupont, Private communication.
25. A. S. Dvinsky, Adaptive grid generation from harmonic maps on riemannian manifolds, *J. Comput. Phys.* **95**, 450 (1991).
26. W. E and C.-H. Shu, Small-scale structures in Boussinesq convection, *Phys. Fluids* **1**, 49 (1994).
27. A Friedman and B. McLeod, Blow-up of Positive solutions of semilinear heat-equations, *Indiana Univ. Math. J.* **34**(2), 425 (1985).
28. R. Grauer, C. Marliani, and K. Germaschewski, Adaptive mesh refinement for singular solutions of the incompressible Euler equations, *Phys. Rev. Lett.* **80**(19), 4177 (1998).
29. R. Grauer and T. C. Sideris, Numerical computation of 3D incompressible ideal fluids with swirl, *Phys. Rev. lett.* **67**(25), 6511 (1991).
30. R. Grauer and T. C. Sideris, Finite time singularities in ideal fluids with swirl, *Physica D* **88**, 116 (1995).
31. W. Huang, Y. Ren, and R. D. Russel, Moving mesh methods based on moving mesh partial differential equations, *J. Comput. Phys.* **113**, 279 (1994).
32. W. Huang, Y. Ren, and R. D. Russel, Moving mesh partial differential equations (MMPDEs) based on the equidistribution principle, *SIAM J. Numer. Anal.* **31**, 709 (1994).
33. W. Huang and R. D. Russell, Moving mesh strategy based on a gradient flow equation for two-dimensional problems, *SIAM J. Sci. Comput.* **20**(3), 998 (1999).
34. R. M. Kerr, Evidence for a singularity of the 3-dimensional, incompressible Euler equations, *Phys. Fluids A* **5**(7), 1725 (1993).
35. P. Knupp and S. Steinberg, Fundamentals of Grid Generation (CRC Press, Boca Raton, FL, 1993).
36. R. Li, T. Tang, and P. Zhang, Moving mesh methods in multiple dimensions based on harmonic maps, *J. Comput. Phys.*, in press.
37. G. Liao, F. Liu, C. de la Pena, D. Peng, and S. Osher, Level-set based deformation methods for adaptive grids, *J. Comput. Phys.* **159**, 103 (2000).
38. M. S. Longuet-Higgins and E. D. Cokelet, The deformation of steep surface waves on water I. A numerical method of computation, *Proc. R. Soc. Lond. A.* **350** (1976).
39. K. Miller and R. N. Miller, Moving finite elements I, *SIAM J. Numer. Anal.* **18**, 1019 (1981).
40. L. R. Petzold, Observations on an adaptive moving grid method for one-dimensional systems of partial differential equations, *Appl. Numer. Math.* **3**, 347 (1987).
41. A. Pumir and E. Siggia, Collapsing solutions to the 3-D Euler equations, *Phys. Fluids A* **2**(2), 220 (1990).
42. A. Pumir and E. D. Siggia, Development of singular solutions to the axisymmetric Euler equations, *Phys. Fluids A* **4**(7), 1472 (1992).

43. A. Pumir and E. D. Siggia, Finite-time singularities in the axisymmetric three-dimensional Euler equations, *Phys. Rev. Lett.* **68**(10), 1511 (1992).
44. W. Ren and X.-P. Wang, An iterative grid redistribution method for singular problems in multiple dimensions, *J. Comput. Phys.* **159**, 246 (2000).
45. M. J. Shelley, D. I. Meiron, and S. A. Orszag, Dynamic aspects of vortex reconnection of perturbed anti-parallel vortex tubes, *J. Fluid Mech.* **246**, 613 (1993).
46. J. F. Thompson, Z. U. A. Warsi, and C. W. Mastin, *Numerical Grid Generation* (North-Holland, New York, 1985).
47. A. Winslow, Numerical solution of the quasi-linear Poisson equation on a nonuniform trainagle mesh, *J. Comput. Phys.* **1**, 149 (1967).
48. P. M. De Zeeuw, Matrix-dependent prolongation and restrictions in a blackbox multigrid solver, *J. Comput. Applied Math.* **33**(1), 1 (1990).