

A Cartesian Grid Embedded Boundary Method for the Heat Equation on Irregular Domains¹

Peter McCorquodale,* Phillip Colella,* and Hans Johansen†

**Applied Numerical Algorithms Group, Lawrence Berkeley National Laboratory, Berkeley, California 94720;*
and †*Department of Mechanical Engineering, University of California, Berkeley, California 94720*

E-mail: PWMcCorquodale@lbl.gov

Received March 22, 2001

We present an algorithm for solving the heat equation on irregular time-dependent domains. It is based on the Cartesian grid embedded boundary algorithm of Johansen and Colella (1998, *J. Comput. Phys.* **147**, 60) for discretizing Poisson's equation, combined with a second-order accurate discretization of the time derivative. This leads to a method that is second-order accurate in space and time. For the case in which the boundary is moving, we convert the moving-boundary problem to a sequence of fixed-boundary problems, combined with an extrapolation procedure to initialize values that are uncovered as the boundary moves. We find that, in the moving boundary case, the use of Crank–Nicolson time discretization is unstable, requiring us to use the L_0 -stable implicit Runge–Kutta method of Twizell, Gumel, and Arigu (1996, *Adv. Comput. Math.* **6**, 333). © 2001 Academic Press

Key Words: 35K15 initial value problems for second-order, parabolic equations; embedded boundary; moving boundaries.

1. INTRODUCTION

In this paper we present a numerical method for solving the parabolic initial-value problem

$$\psi_t = D\Delta\psi + f \text{ on } \Omega, \quad \psi(\mathbf{x}, 0) = \psi_0(\mathbf{x}) \quad (1)$$

¹Research supported at U. C. Berkeley by the U.S. Department of Energy Mathematical, Information and Computing Sciences Division, grants DE-FG03-94ER25205 and DE-FG03-92ER25140, and by the National Science Foundation Graduate Fellowship Program; and at the Lawrence Berkeley National Laboratory by the U.S. Department of Energy Mathematical, Information and Computing Sciences Division, contract DE-AC03-76SF00098.

with constant $D > 0$ on a bounded region Ω , and boundary conditions of either Neumann type

$$\frac{\partial \psi}{\partial n} = g_n(\mathbf{x}, t) \quad \text{on } \partial\Omega \quad (2)$$

or Dirichlet type

$$\psi = g_d(\mathbf{x}, t) \quad \text{on } \partial\Omega. \quad (3)$$

As in previous work on elliptic problems [6], our approach uses a finite-volume discretization, which embeds the domain in a regular Cartesian grid. We treat the solution as cell-centered on a rectangular grid, even when the cell centers are outside the domain.

For the time discretization, for the fixed-boundary problem we use either the Crank–Nicolson method or the method of Twizell, Gumel, and Arigu (TGA) [10]. We solve the moving-boundary problem by converting it to a sequence of fixed-boundary problems, and applying the TGA method to each. Our algorithm is stable and achieves second-order accuracy both on problems with fixed domain Ω and on problems with a time-dependent domain $\Omega(t)$ with boundaries moving with constant velocities. If the ratio of time step Δt to mesh spacing h is kept constant, then the solution error is $O(\Delta t^2 + h^2)$ as $h, \Delta t \rightarrow 0$.

Part of this work appeared in preliminary form in [7].

2. THE HEAT EQUATION FOR FIXED BOUNDARIES

2.1. Spatial Discretization

The underlying discretization of space is given by rectangular control volumes on a Cartesian grid: $\Upsilon_i = [(i - \frac{1}{2}\mathbf{u})h, (i + \frac{1}{2}\mathbf{u})h]$, $i \in \mathbb{Z}^d$, where d is the dimensionality of the problem, h is the mesh spacing, and \mathbf{u} is the vector whose entries are all ones. In the case of a fixed, irregular domain Ω , the geometry is represented by the intersection of Ω with the Cartesian grid. We obtain control volumes $V_i = \Upsilon_i \cap \Omega$ and faces $A_{i \pm \frac{1}{2}\mathbf{e}_s}$, that are the intersection of ∂V_i with the coordinate planes $\{\mathbf{x} : x_s = (i_s \pm \frac{1}{2})h\}$. Here, \mathbf{e}_s is the unit vector in the s direction. We also define A_i^B to be the intersection of the boundary of the irregular domain with the Cartesian control volume: $A_i^B = \partial\Omega \cap \Upsilon_i$. We will assume here that there is a one-to-one correspondence between the control volumes and faces and the corresponding geometric entities on the underlying Cartesian grid. The description can be generalized to allow for boundaries whose width is less than the mesh spacing, or sharp trailing edges.

In order to construct finite difference methods, we will need only a small number of real-valued quantities that are derived from these geometric objects.

- The areas/volumes, expressed in dimensionless terms: volume fractions $\kappa_i = |V_i|h^{-d}$, face apertures $\alpha_{i+\frac{1}{2}\mathbf{e}_s} = |A_{i+\frac{1}{2}\mathbf{e}_s}|h^{-(d-1)}$, and boundary apertures $\alpha_i^B = |A_i^B|h^{-(d-1)}$. We assume that we can compute estimates of the dimensionless quantities that are accurate to $O(h^2)$.

- The locations of centroids, and the average outward normal to the boundary

$$\mathbf{x}_i = \frac{1}{|V_i|} \int_{V_i} \mathbf{x} dV$$

$$\begin{aligned} \mathbf{x}_{i+\frac{1}{2}e_s} &= \frac{1}{|A_{i+\frac{1}{2}e_s}|} \int_{A_{i+\frac{1}{2}e_s}} \mathbf{x} dA \\ \mathbf{x}_i^B &= \frac{1}{|A_i^B|} \int_{A_i^B} \mathbf{x} dA \\ \mathbf{n}_i^B &= \frac{1}{|A_i^B|} \int_{A_i^B} \mathbf{n}^B dA, \end{aligned}$$

where \mathbf{n}^B is the outward normal to $\partial\Omega$, defined for each point on $\partial\Omega$. Again, we assume that we can compute estimates of these quantities that are accurate to $O(h^2)$.

Using just these quantities, we can define conservative discretizations for the divergence operator. Let $\vec{F} = (F^1 \dots F^d)$ be a function of \mathbf{x} . Then

$$\begin{aligned} \nabla \cdot \vec{F} &\approx \frac{1}{|V_i|} \int_{V_i} \nabla \cdot \vec{F} dV = \frac{1}{|V_i|} \int_{\partial V_i} \vec{F} \cdot \mathbf{n} dA \\ &\approx \frac{1}{\kappa_i h} \left(\sum_{\pm=+,-} \sum_{s=1}^d \pm \alpha_{i\pm\frac{1}{2}e_s} F^s(x_{i\pm\frac{1}{2}e_s}) + \alpha_i^B \mathbf{n}_i^B \cdot \vec{F}(x_i^B) \right), \end{aligned} \tag{4}$$

where (4) is obtained by replacing the integrals of the normal components of the vector field \vec{F} with the values at the centroids.

We can use this idea to discretize the Laplacian, written as the divergence of a flux: $\Delta\psi = \nabla \cdot \vec{F}$, where $\vec{F} = \nabla\psi$. We follow the approach described in [6, 7]. The discretized solution values approximate the solution to the PDE at the rectangular cell centers: $U_i^n \approx \psi(\mathbf{i}h, n\Delta t)$. At first glance, this might be a cause for concern, since some of the centers of Cartesian cells Υ_i might not be contained in Ω . However, it is well known that, for any domain with smooth boundary, a smooth function can be extended to all of \mathbb{R}^d with a bound on the relative increase in the $C^{k,\beta}$ norms that depends only on the domain and (k, β) [5]. We assume that the values U_i on the covered cell centers approximate such an extension. We define the time-dependent inhomogeneous operator $L_I^h(t)$ as

$$(L_I^h(t)U)_i = \frac{1}{\kappa_i h} \left(\sum_{\pm=+,-} \sum_{s=1}^d \pm \alpha_{i\pm\frac{1}{2}e_s} F_{i\pm\frac{1}{2}e_s}^s + \alpha_i^B \mathbf{n}_i^B \cdot \vec{F}(x_i^B, t) \right). \tag{5}$$

The fluxes on the cell faces are computed from U by linearly interpolating between centered difference approximations. For example, for the first component ($s = 1$) in two dimensions,

$$\begin{aligned} F_{i+\frac{1}{2},j}^1 &= \eta \frac{(U_{i+1,j} - U_{i,j})}{h} + (1 - \eta) \frac{(U_{i+1,j\pm 1} - U_{i,j\pm 1})}{h} \\ \eta &= \frac{|y_{i+\frac{1}{2},j} - jh|}{h}, \end{aligned} \tag{6}$$

where $\pm = +(-)$ if $y_{i+\frac{1}{2},j} > jh (< jh)$.

Since $\vec{F} = \nabla\psi$, then

$$\mathbf{n}^B \cdot \vec{F}^B = \frac{\partial\psi}{\partial n}; \tag{8}$$

and so with Neumann boundary conditions (2), we set $\mathbf{n}_i^B \cdot \vec{F}(\mathbf{x}_i^B, t) = g_n(\mathbf{x}_i^B, t)$ in (5). With Dirichlet boundary conditions as from (3), we compute an estimate of $\frac{\partial \psi}{\partial n}$ by interpolating from the grid values and the values at the boundaries; for details, see [6]. For both Dirichlet and Neumann boundary conditions, these discretizations lead to linear systems with the same asymptotic conditioning properties as those of the corresponding operators in the absence of irregular boundaries, and they are amenable to the use of fast iterative solvers such as multigrid. Finally, we denote by L_H^h the operator $L_I^h(t)$ with homogeneous boundary conditions, $g_n = 0$ or $g_d = 0$.

2.2. TGA Temporal Discretization

We apply the method of Twizell, Gumel, and Arigu [10] to solve the initial-value problem

$$\begin{aligned} \frac{dU}{dt} &= L_I^h(t)U(t) + f(t) \\ U(0) &= U^0, \end{aligned} \tag{9}$$

where f is evaluated at the same cell centers as U .

We split the time step Δt such that

$$\begin{aligned} \mu_1 + \mu_2 + \mu_3 &= \Delta t \\ \mu_1 + \mu_2 + \mu_4 &= \Delta t/2. \end{aligned}$$

The update at step n uses the boundary values at the old and new times and also at an intermediate time t_{int} :

$$\begin{aligned} U^{n+1} &= (I - \mu_1 L_I^h(t_{\text{new}}))^{-1} (I - \mu_2 L_I^h(t_{\text{int}}))^{-1} \\ &\quad \times [(I + \mu_3 L_I^h(t_{\text{old}}))U^n + (I + \mu_4 L_H^h)f(t_{\text{avg}})\Delta t], \end{aligned} \tag{10}$$

where

$$\begin{aligned} t_{\text{old}} &= n\Delta t \\ t_{\text{new}} &= (n+1)\Delta t = t_{\text{old}} + \mu_1 + \mu_2 + \mu_3 \\ t_{\text{int}} &= t_{\text{new}} - \mu_1 = t_{\text{old}} + \mu_2 + \mu_3 \\ t_{\text{avg}} &= (t_{\text{old}} + t_{\text{new}})/2 = t_{\text{old}} + \mu_1 + \mu_2 + \mu_4. \end{aligned}$$

For a second-order L_0 -stable method, following [10], we pick $a > 1/2$ and

$$\begin{aligned} \mu_1 &= \frac{a - \sqrt{a^2 - 4a + 2}}{2} \Delta t \\ \mu_2 &= \frac{a + \sqrt{a^2 - 4a + 2}}{2} \Delta t \\ \mu_3 &= (1 - a)\Delta t \\ \mu_4 &= \left(\frac{1}{2} - a\right) \Delta t. \end{aligned}$$

For a method that uses real arithmetic only, the truncation error is minimized by taking $a = 2 - \sqrt{2} - \epsilon$, where ϵ is machine precision. In this formulation, the Crank–Nicolson method corresponds to $a = 1/2$, and hence $\mu_1 = \mu_4 = 0$ and $\mu_2 = \mu_3 = \Delta t/2$.

3. MOVING BOUNDARIES

We can generalize the approach for parabolic problems described above to the case of boundaries that move. Specifically, the domain Ω is now a function of time, $\Omega = \Omega(t)$, and the various geometric quantities can also be computed in a time-dependent way: $\kappa_i(t)$, $\alpha_{i+\frac{1}{2}e_s}(t)$, $\mathbf{x}_{i+\frac{1}{2}e_s}(t)$, etc. In this paper, we restrict our study to rigid-body motions, such that each connected component of the boundary has a motion of the form

$$\partial\Omega(t) = \partial\Omega(0) + \mathbf{r}(t).$$

The time step is assumed to satisfy a CFL condition with respect to the velocity $\mathbf{v} = \frac{d\mathbf{r}}{dt}$:

$$\max_{1 \leq s \leq d} |v_s| \frac{\Delta t}{h} < 1.$$

In [7], the starting point for the moving boundary case was a quadrature formula for the update of the solution obtained by integrating the conservation law over the region in space-time given by $\{V_i(t): t_{\text{old}} \leq t \leq t_{\text{new}}\}$. This was combined with a hybrid temporal differencing scheme, using Crank–Nicolson in regular cells, and backward-Euler at irregular cells. The resulting method is second-order accurate, provided $\Delta t = O(h)$. However, for the case of Dirichlet boundary conditions, the method exhibited oscillatory behavior and was unstable to some types of forcing at the moving boundary. In [7] this behavior was attributed to the combination of the neutral stability of Crank–Nicolson at high wave numbers and the presence of eigenvalues of L_H^h with nontrivial imaginary parts, corresponding to eigenmodes with oscillatory behavior near the boundary.

In the present approach, we solve the moving-boundary problem by defining an equivalent fixed-boundary problem for each time step. Specifically, we solve at each time step the discretization (10) of the following fixed-boundary problem:

$$\begin{aligned} \psi_i^{\text{fixed}} &= D\Delta\psi^{\text{fixed}} + f, \\ \text{where } \psi^{\text{fixed}} &= \psi^{\text{fixed}}(\mathbf{x}, t), \quad \mathbf{x} \in \Omega(t_{\text{new}}), \quad t_{\text{old}} \leq t \leq t_{\text{new}}. \end{aligned} \tag{11}$$

The boundary conditions on the fixed boundary are computed by interpolating values from the moving boundary to the points on the fixed boundary $\partial\Omega(t_{\text{new}})$ at times t_{old} and t_{int} . To obtain a stable algorithm, it is necessary to use the L_0 -stable TGA time discretization instead of Crank–Nicolson, a fact that we will demonstrate below. This loss of stability in the case of Crank–Nicolson is consistent with the analysis described above. The interpolation process used to obtain initial and boundary values as the boundary moves interacts with the marginally stable behavior of the fixed-boundary algorithm to produce an unstable method.

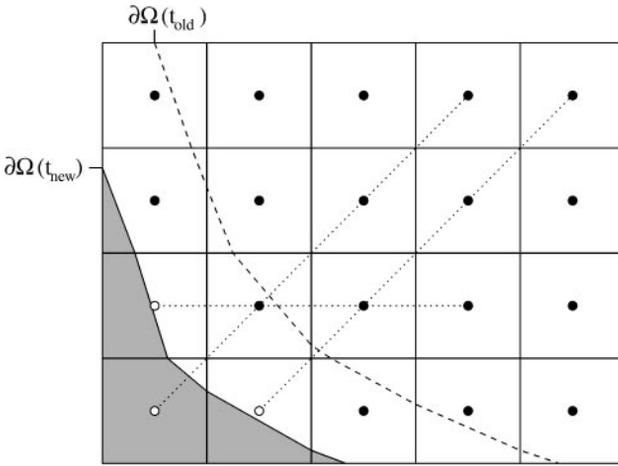


FIG. 1. Centers of cells in $\Omega(t_{old})$ are shown with solid circles, and centers of cells in $\Omega(t_{new}) \setminus \Omega(t_{old})$ are shown with unfilled circles. To estimate the value of U^n at one of these latter points, we interpolate quadratically from values at the centers of three other cells in $\Omega(t_{old})$ forming a line with the new cell center. We pick whichever line is closest in direction to the normal to the boundary at time t_{new} .

The steps required in setting up the fixed-boundary problem (11) are:

1. Extend the domain of U^n to $\Omega(t_{new})$, and define the newly uncovered values by interpolation.
2. Compute boundary values at $(\mathbf{x}_i^B(t_{new}), t_{old})$ and $(\mathbf{x}_i^B(t_{new}), t_{int})$.

In Step 1, to estimate the value of U^n at the center $\mathbf{x}_i(t_{new})$ of a newly uncovered cell in $\Omega(t_{new}) - \Omega(t_{old})$, we use a quadratic interpolant from three other cells in $\Omega(t_{old})$, such that the centers of these cells form a line with $\mathbf{x}_i(t_{new})$. We choose whichever line passing through the centers of the new cell and one of its immediate neighbors has a direction closest to that of the normal $\mathbf{n}_i^B(t_{new})$ (see Fig. 1).

In Step 2, we use the vector displacements

$$\begin{aligned} \delta_{old} &= \mathbf{r}(t_{new}) - \mathbf{r}(t_{old}) \\ \delta_{int} &= \mathbf{r}(t_{new}) - \mathbf{r}(t_{int}) \end{aligned}$$

(see Fig. 2).

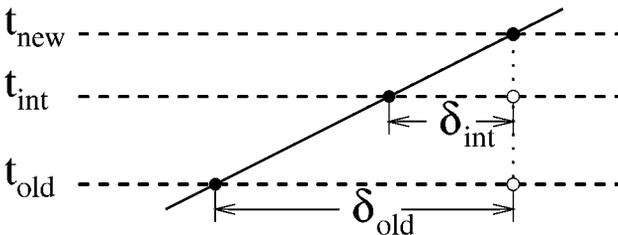


FIG. 2. From known values at points shown with solid circles on the moving boundary, we extrapolate to find values of ψ (Dirichlet) or $\frac{\partial \psi}{\partial n}$ (Neumann) at points shown with unfilled circles, representing times t_{old} and t_{int} on the boundary at time t_{new} .

With Dirichlet boundary conditions, we are given the values of $\psi(\mathbf{x}_i^B(t), t) = g_d(\mathbf{x}_i^B(t), t)$ for any t . We interpolate $g_d(\mathbf{x}_i^B(t_{\text{new}}), t)$ at $t = t_{\text{old}}, t_{\text{int}}$ by

$$g_d(\mathbf{x}_i^B(t_{\text{new}}), t_{\text{old}}) = g_d(\mathbf{x}_i^B(t_{\text{old}}), t_{\text{old}}) + \vec{G}_i \cdot \delta_{\text{old}} + O(h^2) \tag{12}$$

$$g_d(\mathbf{x}_i^B(t_{\text{new}}), t_{\text{int}}) = g_d(\mathbf{x}_i^B(t_{\text{int}}), t_{\text{int}}) + \vec{G}_i \cdot \delta_{\text{int}} + O(h^2), \tag{13}$$

where $\vec{G}_i = \nabla\psi(\mathbf{x}_i(t_{\text{new}}), t_{\text{old}}) + O(h)$ is an estimate of the gradient in cell i , obtained from U^n . In particular, each component G_i^s is computed separately by differentiating the quadratic interpolant through $U_i^n, U_{i\pm e_s}^n$, and $U_{i\pm 2e_s}^n$, where the sign of \pm is chosen so that all points are in $\Omega(t_{\text{new}})$, and therefore U^n has been computed. For example, for the first component in two dimensions,

$$G_{i,j}^1 = \pm \frac{1}{h} \left(-\frac{3}{2}U_{i,j}^n + 2U_{i\pm 1,j}^n - \frac{1}{2}U_{i\pm 2,j}^n \right).$$

With a smooth boundary and smooth ψ and g_d , the error term in (12) is $O(h\Delta t)$ and in (13) is $O(h\mu_1)$. Assuming $\Delta t = O(h)$, then the error in both is $O(h^2)$.

With Neumann boundary conditions, we are given the values of $\frac{\partial\psi}{\partial n}(\mathbf{x}_i^B(t), t) = g_n(\mathbf{x}_i^B(t), t)$ for any t . For the new problem, we use the estimates

$$g_n(\mathbf{x}_i^B(t_{\text{new}}), t_{\text{old}}) = g_n(\mathbf{x}_i^B(t_{\text{old}}), t_{\text{old}}) + \vec{D}_i \cdot \delta_{\text{old}} + O(h^2) \tag{14}$$

$$g_n(\mathbf{x}_i^B(t_{\text{new}}), t_{\text{int}}) = g_n(\mathbf{x}_i^B(t_{\text{int}}), t_{\text{int}}) + \vec{D}_i \cdot \delta_{\text{int}} + O(h^2), \tag{15}$$

where \vec{D}_i has components $D_i^s = \frac{\partial}{\partial n}(\frac{\partial\psi}{\partial x_s})(\mathbf{x}_i(t_{\text{new}}), t_{\text{old}}) + O(h)$ computed as follows:

$$D_i^s = \sum_{r=1}^d (\mathbf{n}_i^B(t_{\text{new}}) \cdot \mathbf{e}_r) \frac{\partial^2\psi}{\partial x_r \partial x_s}(\mathbf{x}_i(t_{\text{new}}), t_{\text{old}}). \tag{16}$$

The second derivatives in (16) are estimated with a three-point stencil

$$\frac{\partial^2\psi}{\partial x_r^2}(\mathbf{x}_i(t_{\text{new}}), t_{\text{old}}) = \frac{U_{i-e_r}^n - 2U_i^n + U_{i+e_r}^n}{h^2} \tag{17}$$

if U_{i-e_r}, U_i , and U_{i+e_r} have all been computed. Otherwise, we use an estimate of the derivative at $x_{i\pm e_r}(t_{\text{new}})$ by replacing i in (17) by either $i + e_r$ or $i - e_r$, as appropriate. Cross derivatives $\frac{\partial^2\psi}{\partial x_r \partial x_s}$ in (16) are computed with a four-point stencil. For example, if neither U_{i-e_r} , nor U_{i-e_s} has been computed, then we use

$$\frac{\partial^2\psi}{\partial x_r \partial x_s}(\mathbf{x}_i(t_{\text{new}}), t_{\text{old}}) = \frac{U_{i+e_r+e_s}^n - U_{i+e_r}^n + U_i^n - U_{i+e_s}^n}{h^2}.$$

We use this same formula if $U_{i+e_r+e_s}^n$ has been computed but no other $U_{i\pm e_r\pm e_s}^n$ has been. Finally, if both U_{i-e_r} and U_{i+e_r} have been computed but U_{i-e_s} has not, then we use

$$\frac{\partial^2\psi}{\partial x_r \partial x_s}(\mathbf{x}_i(t_{\text{new}}), t_{\text{old}}) = \frac{U_{i+e_r+e_s}^n - U_{i+e_r}^n + U_{i-e_r}^n - U_{i-e_r+e_s}^n}{2h^2}.$$

The error terms in (14) and (15) are $O(h\Delta t)$ and $O(h\mu_1)$, which become $O(h^2)$ assuming $\Delta t = O(h)$.

4. RESULTS

Our examples of problem (1) are in two dimensions. All of the test problems have as their solution

$$\psi(x, y, t) = \frac{4 \exp\left(-\frac{x^2 + y^2}{5(t+1)}\right)}{5\pi(t+1)} \quad (18)$$

satisfying

$$\begin{aligned} \psi_t &= \Delta\psi + f, \\ f(x, y, t) &= \frac{4(x^2 + y^2 - 5(t+1))}{125\pi(t+1)^3} \exp\left(-\frac{x^2 + y^2}{5(t+1)}\right). \end{aligned} \quad (19)$$

We solve (19) numerically on a rectangular domain with three elliptically shaped holes, with boundary conditions computed using the exact solution (18). In the moving-boundary problem, the holes move with constant velocities. With both fixed and moving boundaries, we solve two separate problems with different boundary conditions:

- Dirichlet conditions on all boundaries;
- Dirichlet conditions on the fixed external boundaries, but Neumann conditions on the boundaries of the ellipses.

We advance the solution in time from $t = 0$ to $t = 1$ using a mesh spacing h and corresponding time step Δt such that $\Delta t/h \approx 5/\pi$ and Δt divides 1. The values used are shown in Table I.

We compute the solution error after time step n as the difference between the computed solution and the exact solution at the final time $n\Delta t$,

$$\xi_i^n = U_i^n - \psi(ih, n\Delta t).$$

We display the max norm of the solution error

$$\|\xi^n\|_\infty = \max_i |\xi_i^n|$$

and the volume-weighted 1-norm

$$\|\xi^n\|_1 = \frac{\sum_i |\xi_i^n \kappa_i|}{\sum_i \kappa_i}.$$

TABLE I
Mesh Spacing h , Time step Δt , and Number of Steps Used in Runs

h	Δt	steps
0.1000	0.1667	6
0.0500	0.0769	13
0.0250	0.0400	25
0.0125	0.0200	50

TABLE II
Parameters for Ellipses in (20), Where $\gamma = \sqrt{2}/15 = 0.09428$

i	p_i	q_i	a_i	b_i
1	-6γ	-5γ	3γ	2γ
2	10γ	-7γ	2γ	1γ
3	7γ	3γ	1.5γ	2γ

4.1. Fixed Boundaries

For the fixed-boundary problems, we solve on the domain

$$\Omega = \Omega^\square - \Omega_1^\circ - \Omega_2^\circ - \Omega_3^\circ,$$

where $\Omega^\square = [-1.5, 1.5] \times [-1, 1]$, and $\Omega_1^\circ, \Omega_2^\circ, \Omega_3^\circ$ are interiors of ellipses,

$$\Omega_i^\circ = \left\{ (x, y) : \frac{(x - p_i)^2}{a_i^2} + \frac{(y - q_i)^2}{b_i^2} \leq 1 \right\}, \quad (20)$$

where the centers and axis lengths are set as in Table II. These are chosen as multiples of an irrational number so as to reduce dependencies on the discretization.

The exact solution (18) to (19) at $t = 1$ is shown as a contour plot in Fig. 3. We define the rate of convergence between two norms, e_1 and e_2 , with two different mesh spacings, h_1 and h_2 , as

$$r = \log\left(\frac{e_1}{e_2}\right) / \log\left(\frac{h_1}{h_2}\right).$$

Then $r = 2$ indicates a method that is second-order accurate.

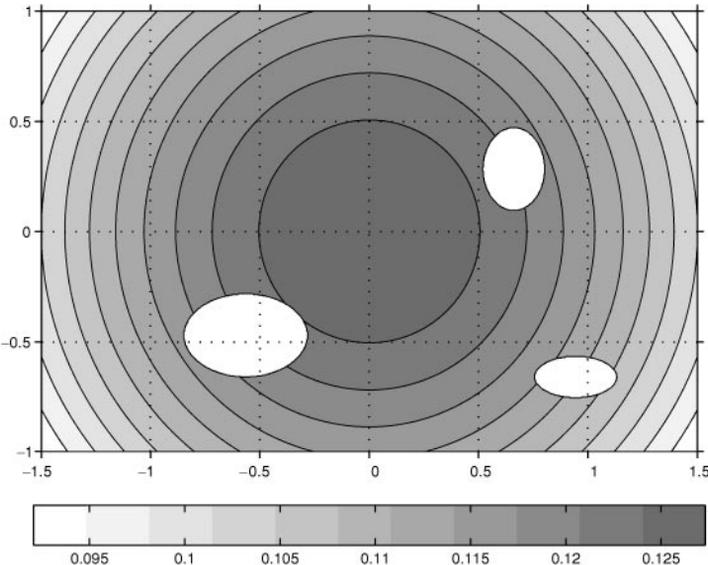


FIG. 3. Contour plot of exact solution (18) to (19) at $t = 1$ with fixed boundaries.

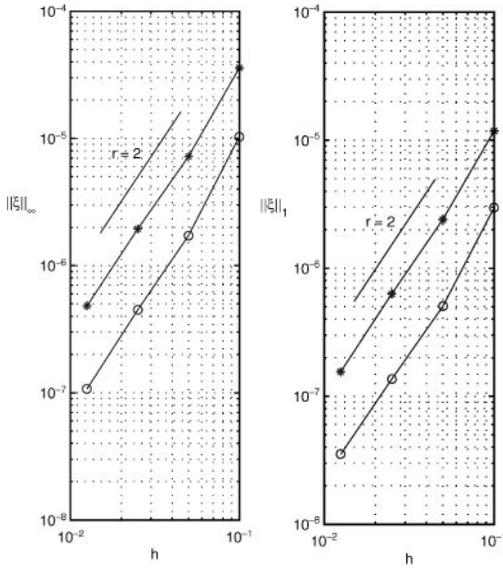


FIG. 4. Solution error at $t = 1$ using Crank–Nicolson (stars) and TGA (circles), for the Dirichlet problem for (19) with fixed boundaries. Left-hand plot shows max norm, right-hand plot shows 1-norm. We see that both $\|\xi\|_\infty = O(h^2)$ and $\|\xi\|_1 = O(h^2)$, indicating second-order accuracy.

Figure 4 shows both the max-norm and the 1-norm of the solution error at $t = 1$ for the Dirichlet problem. Figure 5 shows the same quantities for the Neumann problem. As these figures show, both the Crank–Nicolson and TGA methods are second-order accurate.

The solution error at $t = 1$ for the finest mesh spacing used ($h = \frac{1}{80}$) is plotted in Fig. 6 for both the Crank–Nicolson and TGA methods applied to the Dirichlet problem. For the Neumann problem, the error in these methods is plotted in Fig. 7.

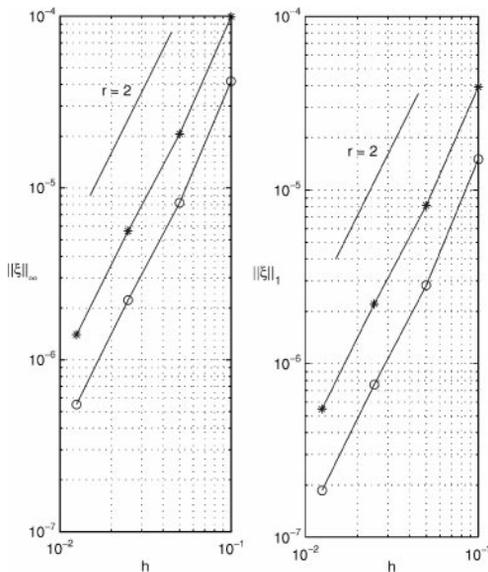


FIG. 5. Solution error at $t = 1$ using Crank–Nicolson (stars) and TGA (circles), for the Neumann problem for (19) with fixed boundaries. Left-hand plot shows max norm, right-hand plot shows 1-norm. We see that both $\|\xi\|_\infty = O(h^2)$ and $\|\xi\|_1 = O(h^2)$, indicating second-order accuracy.

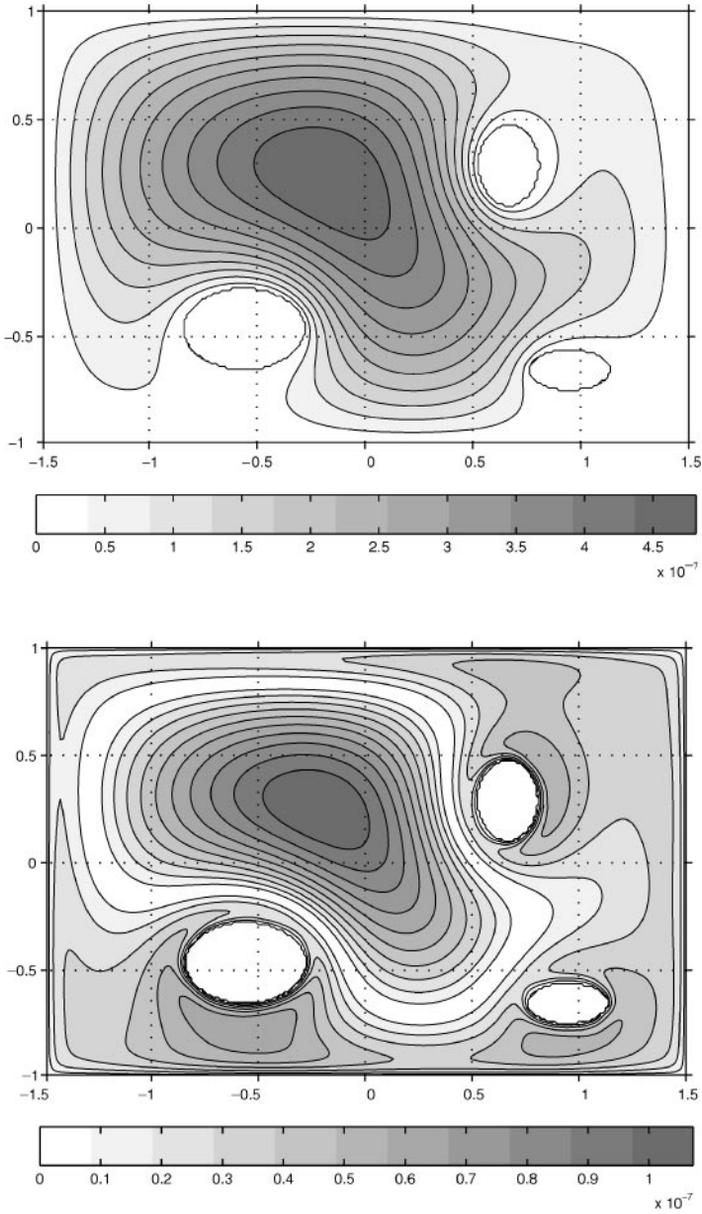


FIG. 6. Contour plots of absolute value of solution error to (19) at $t = 1$ for fixed boundaries, Dirichlet boundary conditions, $h = 0.0125$. The top figure is for Crank–Nicolson method, and the bottom figure for TGA method.

4.2. Moving Boundaries

In the moving-boundary problems, we solve on the time-dependent domain

$$\Omega(t) = \Omega^\square - \Omega_1^\circ(t) - \Omega_2^\circ(t) - \Omega_3^\circ(t),$$

where $\Omega^\square = [-1.5, 1.5] \times [-1, 1]$, and the initial ellipse interiors $\Omega_1^\circ(0)$, $\Omega_2^\circ(0)$, and $\Omega_3^\circ(0)$ are as defined in (20) and Table II. In our moving-boundary problem, the axis lengths a_i

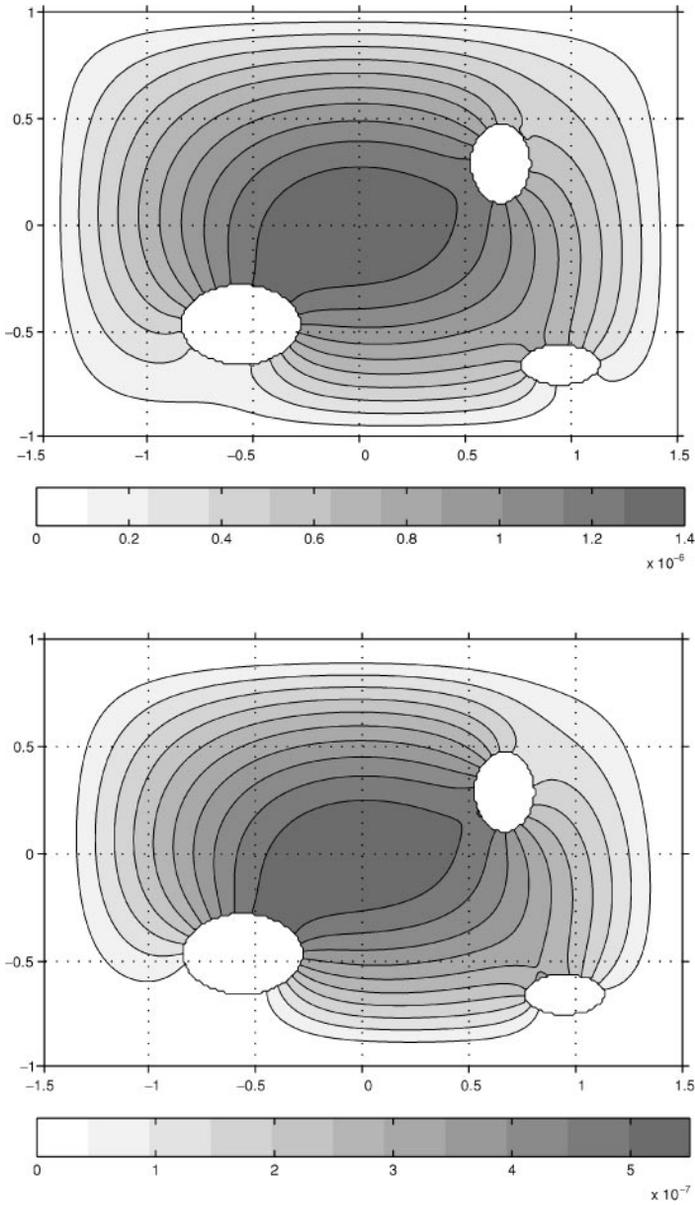


FIG. 7. Contour plots of absolute value of solution error to (19) at $t = 1$ for fixed boundaries, Neumann boundary conditions, $h = 0.0125$. The top figure is for Crank–Nicolson method, and the bottom figure for TGA method.

and b_i do not vary, but the ellipse centers (p_i, q_i) move at constant velocities,

$$(p_i(t), q_i(t)) = (p_i(0) + u_i t, q_i(0) + v_i t),$$

where u_i and v_i are listed in Table III. Note that the timesteps Δt and the mesh spacings h from Table I satisfy a CFL condition $\frac{1}{5} < \max\{|u_i|, |v_i|\} \Delta t / h \leq \frac{1}{3}$.

A contour plot of the exact solution to (19) at $t = 1$ is shown in Fig. 8. Figure 9 shows both the max-norm and the 1-norm of the solution error at $t = 1$ for the Dirichlet problem.

TABLE III
Components of Velocities of Ellipses

i	u_i	v_i
1	-0.10	0.20
2	-0.15	0.15
3	-0.20	0.20

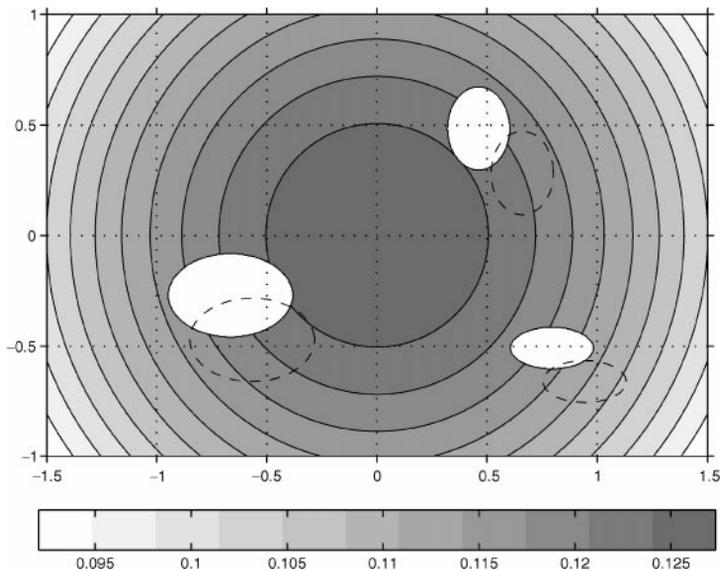


FIG. 8. Contour plot of exact solution to (19) at $t = 1$ for moving-boundary problem. The dashed ellipses indicate the boundaries at $t = 0$.

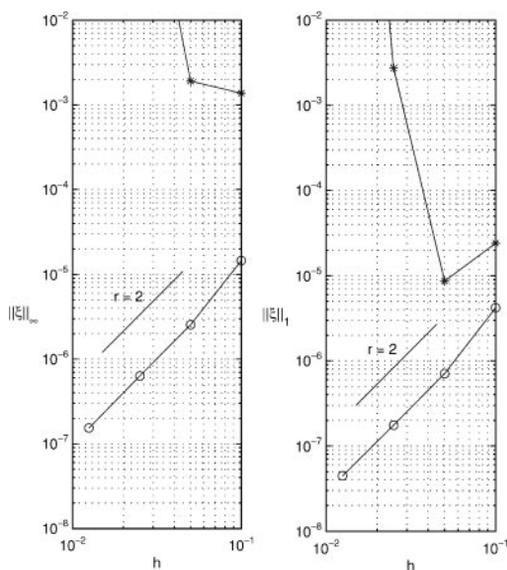


FIG. 9. Solution error at $t = 1$ in TGA method (circles) and Crank–Nicolson method (stars), with Dirichlet conditions for (19) on moving boundaries. Left-hand plot shows max norm; right-hand plot shows 1-norm.

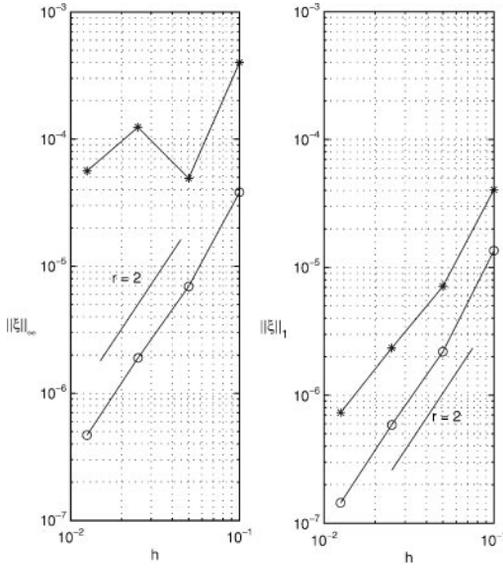


FIG. 10. Solution error at $t = 1$ in TGA method (circles) and Crank–Nicolson (stars), with Neumann conditions for (19) on moving boundaries. Left-hand plot shows max norm; right-hand plot shows 1-norm.

Figure 10 shows the same quantities for the Neumann problem. We see that when applied to these problems, the TGA method is second-order accurate in both norms. The Crank–Nicolson is second-order accurate in 1-norm for the Neumann problem but is zeroth-order in max norm, and diverges in both norms for the Dirichlet problem with moving boundaries.

The solution error at $t = 1$ for the finest mesh spacing used ($h = \frac{1}{80}$) in the TGA method solving (19) is plotted in Fig. 11 for the Dirichlet problem and Fig. 12 for the Neumann problem.

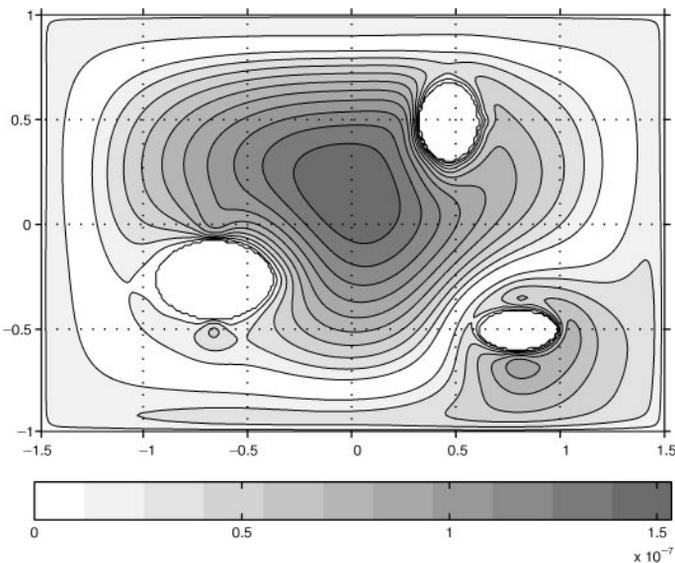


FIG. 11. Contour plot of absolute value of solution error to (19) at $t = 1$ for moving boundaries, Dirichlet boundary conditions, $h = 0.0125$ in TGA method.

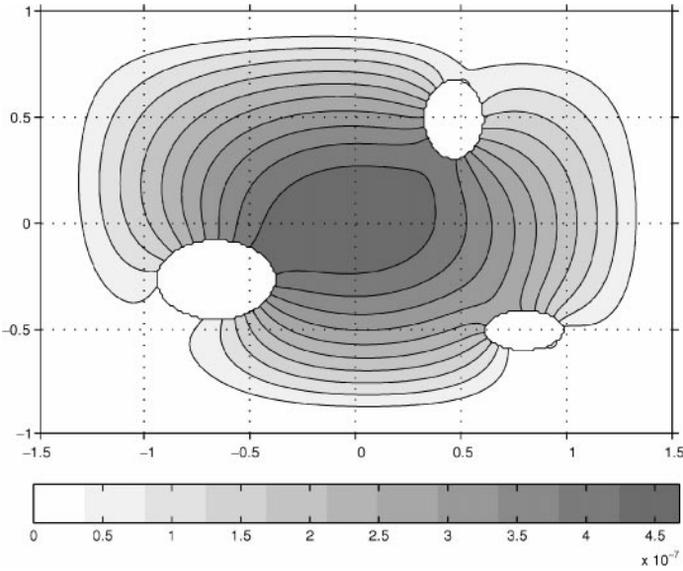


FIG. 12. Contour plot of absolute value of solution error to (19) at $t = 1$ for moving boundaries, Neumann boundary conditions, $h = 0.0125$ in TGA method.

5. FUTURE WORK

The method described here, together with that in [6] for elliptic PDEs and [8] for hyperbolic PDEs, provide the fundamental components required for developing second-order accurate methods for a broad range of continuum mechanics problems in irregular geometries based on the predictor–corrector approach in [2]. Similar approaches based on formally inconsistent discretizations at the irregular boundary have been used previously and observed to be stable [1, 9], so we expect that the extension to the more accurate boundary discretization should be straightforward. For embedded boundary methods to be practical, it is necessary to use them in conjunction with block-structured adaptive mesh refinement, particularly in three dimensions. This is routine for the case in which the embedded boundary is contained in the finest level of refinement [6], but requires some additional discretization design when the embedded boundary crosses coarse–fine interfaces.

One issue that has not been completely addressed is discrete conservation. For the case of fixed boundaries, both the Crank–Nicolson and the TGA algorithms are in discrete conservation form; i.e., the divided difference in time of the old and new values can be written as a difference of fluxes of the form (4). In that case, the difference in the volume-weighted sums of the dependent variables over any discrete subdomain is equal to the sum of fluxes across the boundaries of the subdomain. This is not the case for the moving boundary algorithm, since the conversion of the moving-boundary problem to a sequence of fixed-boundary problems does not satisfy the appropriate summation-by-parts identity. One possible way to correct this problem is to compute an estimate of the failure to conserve based on a space–time quadrature formula, which is used to construct a conservative and stable increment of the solution that restores overall conservation, analogous to what is done in the hyperbolic case [3, 4]. Such an approach was proposed in [7], but the modified update triggered the boundary instability of the hybrid Crank–Nicolson method used there.

We expect that such a method would have no stability problems because of the L_0 stability of the TGA time discretization.

REFERENCES

1. A. S. Almgren, J. B. Bell, P. Colella, and T. Marthaler, A Cartesian mesh method for the incompressible Euler equations in complex geometries, *SIAM J. Sci. Comput.* **142**, 1 (1997).
2. J. B. Bell, P. Colella, and H. M. Glaz, A second order projection method for the incompressible Navier–Stokes equations, *J. Comput. Phys.* **85**, 257 (1989).
3. J. B. Bell, P. Colella, and M. Welcome, A conservative front-tracking for inviscid compressible flow, in *Proceedings of the Tenth AIAA Computational Fluid Dynamics Conference* (AIAA Press, Washington, DC, June 1991), pp. 814–822.
4. I.-L. Chern and P. Colella, *A Conservative Front Tracking Method for Hyperbolic Conservation Laws*, Technical Report UCRL-97200 (Lawrence Livermore National Laboratory, July 1987).
5. N. Gilbarg and N. S. Trudinger, *Elliptic Partial Differential Equations of Second Order* (Springer-Verlag, New York/Berlin, 1977).
6. H. Johansen and P. Colella, A Cartesian grid embedded boundary method for Poisson’s equation on irregular domains, *J. Comput. Phys.* **147**, 60 (1998).
7. H. S. Johansen, *Cartesian Grid Embedded Boundary Methods for Elliptic and Parabolic Partial Differential Equations on Irregular Domains*, Ph.D. thesis (Dept. of Mechanical Engineering, Univ. of California, Berkeley, December 1997).
8. D. Modiano and P. Colella, A higher-order embedded boundary method for time-dependent simulation of hyperbolic conservation laws, in *Proceedings of the FEDSM 00—ASME Fluids Engineering Simulation Meeting, Boston, MA, June 2000*.
9. R. B. Pember, A. S. Almgren, W. Y. Crutchfield, L. H. Howell, J. B. Bell, P. Colella, and V. E. Beckner, *An Embedded Boundary Method for the Modeling of Unsteady Combustion in an Industrial Gas-fired Furnace*, Technical Report UCRL-JC-122177, LLNL, October 1995. (Presented at the 1995 Fall Meeting of the Western States Section of the Combustion Institute, Stanford University.)
10. E. H. Twizell, A. B. Gumel, and M. A. Arigu, Second-order, L_0 -stable methods for the heat equation with time-dependent boundary conditions, *Adv. Comput. Math.* **6**, 333 (1996).