Available online at www.sciencedirect.com
SCIENCE DIRECT®

ELSEVIER

Journal of Computational Physics 198 (2004) 295–309

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

High order marching schemes for the wave equation in complex geometry

Jing-Rebecca Li [*,1], Leslie Greengard [2]

Courant Institute of Mathematical Sciences, New York University, 251 Mercer Street, New York, NY 10012-1110, USA

Received 11 August 2003; received in revised form 16 December 2003; accepted 5 January 2004
Available online 26 February 2004

Abstract

We present a new class of explicit marching schemes for the wave equation in complex geometry. They rely on a simple embedding of the domain in a uniform Cartesian grid, which allows for efficient and automatic implementation but creates irregular cells near the boundary. While existing explicit finite difference schemes are generally restricted in the size of the time step that can be taken by the dimensions of the smallest cell, the schemes described here are capable of taking time steps dictated by the uniform grid spacing. This should be of significant benefit in a wide variety of simulation efforts.

Keywords: Small cell; Stability; Wave equation

1. Introduction

Many problems in electromagnetics, optics, and acoustics require the solution of the time-dependent wave equation in complex geometry. While frequency domain techniques are often both accurate and efficient, they tend to become inefficient when transient behavior needs to be resolved. Moreover, the Fourier transform which maps from the time to the frequency domain is of questionable utility in the presence of material inhomogeneities and non-linearities. In the present paper, we limit our discussion to the isotropic, homogeneous case, but we expect that the underlying notion of constructing stable marching schemes can be extended to more complicated governing equations.

We are particularly interested in designing numerical marching schemes that are applicable on Cartesian grids with embedded boundaries as discussed, for example, in [1,2]. These have a number of advantages

[*] Corresponding author.
E-mail addresses: jingli@cims.nyu.edu (J.-R. Li), greengard@cims.nyu.edu (L. Greengard).

[1] The work of this author was supported by the US Department of Energy under grant DE-FGO2-00ER25053 and by the National Science Foundation under grant DMS-9983190.

[2] The work of this author was supported in part by the US Department of Energy under grant DE-FG02-00ER25053 and HRL under contract MDA972-01-C-0006.

doi:10.1016/j.jcp.2004.01.021

over body fitted grids in terms of efficiency, memory consumption, automation, mesh refinement, and convergence analysis. In complicated domains, however, small cells are created wherever the irregular boundary intersects the Cartesian grid. Classical stability analysis [16,17] suggests that the the size of the time step is dependent on the size of the smallest cell in the spatial discretization, rendering such schemes impractical unless some additional techniques are introduced to overcome this restriction. More precisely, we would like to allow time steps on the order of the size of the uniform cells in the grid away from the domain boundary.

Existing approaches to this "small cell" problem include cell merging (see, e.g. [6,15]), large time step generalizations of Godunov's method [12], and the rotated grid *h*-box method [5]. In cell merging, one removes small cells near the boundary, which tends to result in a loss of accuracy there. In the generalized Godunov and *h*-box methods, one keeps the small cells, enlarging the numerical domain of dependence near the small cells in some way. These schemes have not been carried out to high order accuracy. Recent work on Maxwell's equations [7] has demonstrated the feasibility of obtaining second order accuracy using ideas that are somewhat related to those discussed here. Even more closely related is the second order accurate method for the scalar wave equation found in [11]. Nevertheless, for large scale wave propagation problems, high precision is critical to avoid numerical dispersion errors. Hence, there is room for significant improvement in the performance of numerical methods in terms of order of accuracy.

In this paper we show experimentally that high order schemes in complex geometry are feasible in two space dimensions. These schemes follow the work of [3,13], which introduced a new approach to marching in time that appears to be remarkably insensitive to the presence of small cells. These are three time level schemes, based on an exact evolution formula for wave propagation. Section 2 describes the original evolution formula of [3], while Sections 3–5 provide the essential details of the implementation. Section 6 summarizes the results of our numerical experiments, and Section 7 contains our conclusions. It is worth emphasizing that we have no formal results concerning stability. We hope that the analytic formalism presented here leads to a direct proof of convergence.

An independent approach, also motivated by [3] is described in [18]. Rather than relying on quadrature, the authors of the latter paper derive marching schemes from a Lax–Wendroff finite difference formulation and enforce stability as a condition on the local stencil itself. Finally, we restrict our attention here to interior problems. Exterior problems require non-reflecting boundary conditions which can be imposed on a disk, using the technique of [4], or in general domains, using the recently-developed fast plane-wave time-domain schemes [8,10,14].

2. Exact integral evolution formula

It was shown in [3] that if $u(\mathbf{x},t)$ is a solution to the homogeneous wave equation

$$u_{tt} = \nabla^2 u, \tag{1}$$

in $\mathbb{R}^d$, then there exists a kernel $G_d(r,\tau)$ such that

$$u(\mathbf{x},t+\tau) = 2u(\mathbf{x},t) - u(\mathbf{x},t-\tau) + \int_{B_\tau(\mathbf{x})} G_d(|\mathbf{x}-\boldsymbol{\xi}|,\tau)\ \nabla^2 u(\boldsymbol{\xi},t)\,\mathrm{d}\boldsymbol{\xi}, \tag{2}$$

where $B_\tau(\mathbf{x}) = \{\boldsymbol{\xi}, |\boldsymbol{\xi}-\mathbf{x}| \leqslant \tau\}$ denotes the closed ball in $\mathbb{R}^d$ of radius τ centered at $\mathbf{x}$. Moreover,

$$G_1(r,\tau) = \tau - r, \tag{3}$$

$$G_2(r,\tau) = \frac{\ln(\tau + \sqrt{\tau^2 - r^2}) - \ln r}{\pi}, \tag{4}$$

$$G_3(r,\tau) = \frac{1}{2\pi r}. \tag{5}$$

We will use the exact formula (2) to construct a class of explicit three time level schemes to solve (1). In particular we modify (2) to solve (1) in a bounded domain D subject to either Dirichlet conditions

$$u(\mathbf{x},t) = g(\mathbf{x},t), \tag{6}$$

or Neumann conditions

$$\frac{\partial u}{\partial v} = h(\mathbf{x},t) \tag{7}$$

on the boundary ∂D, where v is the unit outward normal. We begin by restricting the region of integration in the formula (2) to define $\tilde{u}(\mathbf{x},t+\tau)$:

$$\tilde{u}(\mathbf{x},t+\tau) := 2u(\mathbf{x},t) - u(\mathbf{x},t-\tau) + \int_{B_\tau(\mathbf{x})\bigcap\mathscr{D}} G_d(|\mathbf{x}-\boldsymbol{\xi}|,\tau)\ \nabla^2 u(\boldsymbol{\xi},t)\,\mathrm{d}\boldsymbol{\xi}. \tag{8}$$

It is easy to verify that $\tilde{u}$ satisfies the wave equation for $\tau > 0$. As a result, the function Ψ given by

$$\Psi(\mathbf{x},t+\tau) := u(\mathbf{x},t+\tau) - \tilde{u}(\mathbf{x},t+\tau), \quad \mathbf{x} \in D,$$

also satisfies the wave equation and takes on zero initial data. Suppose now that one wants to impose the boundary condition given by (6). Then we must have

$$\Psi(\mathbf{x},t+\tau) = g(\mathbf{x},t+\tau) - \tilde{u}(\mathbf{x},t+\tau), \quad \mathbf{x} \in \partial D. \tag{9}$$

Similarly, if the boundary condition (7) is to be satisfied, we must have

$$\frac{\partial \Psi}{\partial v}(\mathbf{x},t+\tau) = h(\mathbf{x},t+\tau) - \frac{\partial \tilde{u}}{\partial v}(\mathbf{x},t+\tau), \quad \mathbf{x} \in \partial D. \tag{10}$$

In either case, the wave equation for Ψ can be solved using hyperbolic potential theory [9]. For short times τ, this involves a *local boundary integral equation* for which the cost is negligible.

In one space dimension, the solution is available analytically. Assuming that we are working on an interval $[a,b]$ with $a < x \ll b$, an elementary calculation shows that for the boundary condition in (6)

$$\Psi(x,t+\tau) = \begin{cases} 0 & \text{if } x-a \geqslant \tau, \\ g(a,t+\tau-x+a) - \tilde{u}(a,t+\tau-x+a) & \text{if } x-a \leqslant \tau. \end{cases} \tag{11}$$

For the boundary condition in (7),

$$\Psi(x,t+\tau) = \begin{cases} 0 & \text{if } x-a \geqslant \tau, \\ -\int_\tau^{x-a}\left(h(a,t+\tau-\xi) - \frac{\partial \tilde{u}}{\partial v}(a,t+\tau-\xi)\right)\mathrm{d}\xi & \text{if } x-a \leqslant \tau. \end{cases} \tag{12}$$

A similar formula applies near the right boundary point $x = b$.

In two dimensions, with the boundary conditions given by (6), we represent the solution as a double-layer potential:

$$\Psi(\mathbf{x}, t+\tau) = D[\mu](\mathbf{x}, \tau) := \int_0^\tau \int_{\xi(\gamma)\in\partial D} \frac{\partial}{\partial\nu(\gamma)} G(\|\mathbf{x} - \xi(\gamma)\|, \tau - \tau')\mu(\gamma, \tau')\,\mathrm{d}\gamma\,\mathrm{d}\tau', \tag{13}$$

for $\mathbf{x} \in D$, where $G(r, \tau)$ is the free-space Green's function

$$G(r, \tau) = \begin{cases} 0 & \text{if } r > \tau, \\ 1/\sqrt{\tau^2 - r^2} & \text{if } r < \tau, \end{cases}$$

γ is some parameterization of ∂D, $\nu(\gamma)$ is the unit outward normal at $\xi(\gamma)$, and $\mu(\gamma, \tau')$ is the unknown double-layer density. Taking the limit as $\mathbf{x}$ approaches a point $\xi(\gamma')$ on the boundary, a standard computation [9] yields the relation

$$\Psi(\xi(\gamma'), t+\tau) = -\frac{1}{2}\mu(\gamma', \tau) + \int_0^\tau \int_{\xi(\gamma)\in\partial D} \frac{\partial}{\partial\nu(\gamma)} G(\|\xi(\gamma') - \xi(\gamma)\|, \tau - \tau')\mu(\gamma, \tau')\,\mathrm{d}\gamma\,\mathrm{d}\tau'. \tag{14}$$

Thus, the density μ is obtained by solving the following integral equation on the boundary:

$$-\frac{1}{2}\mu(\gamma', \tau) + D_0[\mu](\gamma', \tau) = g(\xi(\gamma'), t+\tau) - \tilde{u}(\xi(\gamma'), t+\tau), \tag{15}$$

where the compact boundary integral operator is

$$D_0[\mu](\gamma', \tau) := \int_0^\tau \int_{\xi(\gamma)\in\partial D} \frac{\partial}{\partial\nu(\gamma)} G(\|\xi(\gamma') - \xi(\gamma)\|, \tau - \tau')\mu(\gamma, \tau')\,\mathrm{d}\gamma\,\mathrm{d}\tau'. \tag{16}$$

If the boundary conditions are given by (7), we represent the solution in the form of a single-layer potential:

$$\Psi(\mathbf{x}, t+\tau) = S[\sigma](\mathbf{x}, \tau) := \int_0^\tau \int_{\xi(\gamma)\in\partial D} G(\|\mathbf{x} - \xi(\gamma)\|, \tau - \tau')\sigma(\gamma, \tau')\,\mathrm{d}\gamma\,\mathrm{d}\tau', \tag{17}$$

for $\mathbf{x} \in D$, where $\sigma(\gamma, \tau')$ is an unknown single-layer density. Taking the appropriate limits, the density σ must solve the following integral equation on the boundary:

$$\frac{1}{2}\sigma(\gamma', \tau) + K_0[\sigma](\gamma', \tau) = h(\xi(\gamma'), t+\tau) - \frac{\partial\tilde{u}}{\partial\nu}(\xi(\gamma'), t+\tau), \tag{18}$$

where the compact boundary integral operator is

$$K_0[\sigma](\gamma', \tau) := \int_0^\tau \int_{\xi(\gamma)\in\partial D} \frac{\partial}{\partial\nu(\gamma')} G(\|\xi(\gamma') - \xi(\gamma)\|, \tau - \tau')\sigma(\gamma, \tau')\,\mathrm{d}\gamma\,\mathrm{d}\tau'.$$

Definition 2.1. We refer to the function $\Psi(\mathbf{x}, t+\tau)$ as the boundary correction.

Our numerical approach to solving the wave Eq. (1) in the presence of a boundary is straightforward. At every time step, we discretize (8), march $\tilde{u}$ forward in time, and add the boundary correction $\Psi(\mathbf{x}, t+\tau)$.

Remark 2.1. For the sake of simplicity, we assume $u(\mathbf{x},0)$ and $u(\mathbf{x},\Delta t)$ are given, ignoring "startup issues" for the three level scheme.

3. Discretization

Let the domain D be contained within the square region $[a,b]\times[a,b]$ on which we have imposed an $(N+1)\times(N+1)$ uniform Cartesian grid

$$(x_i,y_j)=(a+ih,b+jh),$$

with $h=(b-a)/N$. If the grid point $(x_i,y_j)\in D$, it is referred to as an *interior grid point*. Otherwise it is referred to as an *exterior grid point*. We assume that the boundary is smooth and that it is provided as a sequence of L points $\xi_1,\dots,\xi_L$ which we refer to as *boundary points*. For the sake of simplicity, we require that the boundary point separation is at least as fine as the underlying volume grid (the arc length between ξ_i and ξ_{i+1} is at most h).

Definition 3.1. We denote by $U^n_{i,j}$ the value of the approximate solution at the grid point (x_i,y_j) at time $t_n:=n\Delta t$.

As indicated in Remark 2.1 above, we assume that $U^0_{i,j}$ and $U^1_{i,j}$ are given for all interior grid points.

The values on the right hand side of (8) are required for all interior grid points. In order to compute the integral at the next time step, the function $\nabla^2 u$ at the current time is required in the region of integration $B_\tau(\mathbf{x})\bigcap\mathscr{D}$. We go further and approximate u at the current time as a function in the entire domain $\mathscr{D}$ and then take its analytical Laplacian. This function is a piecewise polynomial function obtained from interpolating the computed solution U at the current time. The pieces of this function are defined on subregions of $\mathscr{D}$ of the following form:

$$D_{i,j}:=C_{i,j}\bigcap D,$$

where $C_{i,j}$ is the square subdomain

$$C_{i,j}:=[x_i,x_{i+1}]\times\left[y_j,y_{j+1}\right].$$

We refer to $D_{i,j}$ as a *cell*. If $D_{i,j}=C_{i,j}$, then $D_{i,j}$ is called a *regular* cell. If $D_{i,j}$ is empty, we ignore it. If $D_{i,j}$ is neither regular nor empty, it is an *irregular* cell.

On each cell, we use bivariate polynomial basis functions and denote the representation on $D_{i,j}$ by $U^n_{i,j}(x,y)$. The polynomial $U^n_{i,j}(x,y)$ is obtained by fitting, in a least squares sense, a subset of the current data at points in or near $D_{i,j}$. To be more precise, let p be the desired spatial approximation order. We then assume $U^n_{i,j}(x,y)$ has the form:

$$U^n_{i,j}(x,y):=\sum_{k+l\leqslant p}a_{k,l}x^ky^l. \tag{19}$$

The number of unknown coefficients $a_{k,l}$ is $(p+2)(p+1)/2$. We then look for a set of points around the cell $D_{i,j}$ at which to evaluate (19) to set up the least squares problem.

We denote by $R_{i,j}$ the set of interpolation points (also called an *interpolation configuration*) and let $R_{i,j}$ consists of Q grid points and F boundary points. We require that $R_{i,j}$ satisfies the following three conditions:

(1) All grid points contained in $R_{i,j}$ must be interior grid points.
(2) There are more data points than unknowns in the least squares problem, that is

$$Q + F \geqslant (p+2)(p+1)/2. \tag{20}$$

(3) There are sufficiently many points in each linear dimension to allow interpolation of order p in that dimension.

Let us consider the easiest case first, where the cell $D_{i,j}$ is not only regular ($D_{i,j} = C_{i,j}$) but far from the boundary. We will assume that $D_{i,j}$ is sufficiently far away from the boundary so that all the grid points contained in the rectangular lattice $L_{i,j} := \{(x_k, y_m) \mid k = k_0, \ldots, i, i+1, \ldots, k_f, m = m_0, \ldots, j, j+1, \ldots, m_f\}$ are interior grid points, where the dimensions of the lattice satisfies two conditions: $(k_f - k_0 + 1)(m_f - m_0 + 1) \geqslant (p+2)(p+1)/2$ and $(k_f - k_0) \geqslant p$, $(m_f - m_0) \geqslant p$. For such $D_{i,j}$ and $L_{i,j}$, $L_{i,j}$ is an acceptable interpolation configuration for $D_{i,j}$ because it satisfies conditions 1–3.

Definition 3.2. We call the lattice $L_{i,j}$ a regular interpolation configuration. If $D_{i,j}$ uses $L_{i,j}$ as the set of interpolation points, it is an interpolation-regular cell. If the interpolation set for $D_{i,j}$ is not $L_{i,j}$, then $D_{i,j}$ is an interpolation-irregular cell.

In Fig. 1, for example, to produce a third order bivariate polynomial approximation on cell 4 around the point **r**, the set of interpolation points is the lattice $L_{i,j}$ with $k_0 = i-1, k_f = i+1, m_0 = j-1, m_f = j+1$.

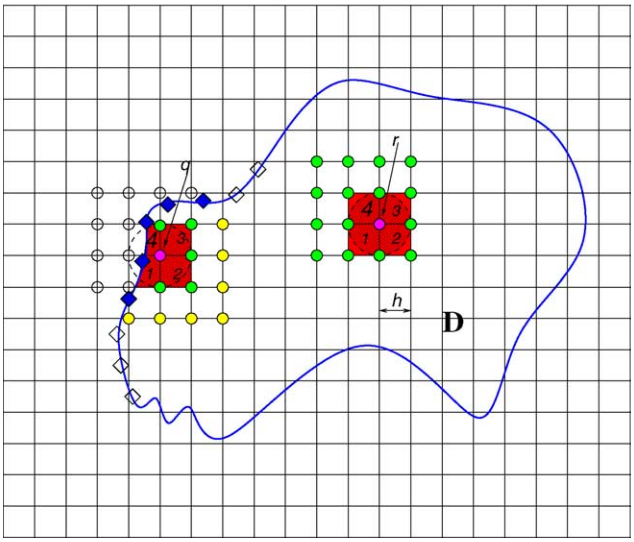


Fig. 1. A domain D is superimposed on a Cartesian mesh. The points **r** and **q** are interior grid points whose domains of dependence Ω for $\Delta t = h$ are outlined by dashed circles. The computation of $\tilde{u}(\mathbf{r}, t+\tau)$ and $\tilde{u}(\mathbf{q}, t+\tau)$ requires integration over the intersection of Ω with four cells labeled 1, 2, 3, and 4. For the point **r** all four cells are regular in shape. If we interpolate the approximate solution as a third order bivariate polynomial in each cell, 16 points on a 4×4 lattice around the cell are used whenever possible and this configuration is called a regular interpolation configuration. The interpolation configuration for cell 4 around the point **r** is regular and is shown. It can be seen that all four cells around **r** have regular interpolation configurations. Hence the stencil for **r** is regular and is the same as for all other interior grid points sufficiently far from the boundary. On the other hand, for the point **q**, cells 2 and 3 are regular in shape whereas cells 1 and 4 are irregular in shape. The 4×4 lattice around cell 4 contains points outside of the domain and cannot be used in interpolation. The set of interpolation points used for this cell is irregular and contains 13 interior grid points (six solid disks and seven shaded disks) and five boundary points (shaded diamonds). It can be seen that the 4×4 lattices around cells 1, 2, and 3 also contain grid points outside of the boundary; hence, these cells will also have irregular interpolation configurations.

They are shown as solid disks around cell 4 (the lower left corner point of cell 4 being (x_i, y_j)). In this case, cell 4 is interpolation-regular for third order interpolation.

For cells nearer the boundary, the lattice $L_{i,j}$ will contain grid points which are not interior and hence cannot be used. In order to have sufficiently many points for interpolation, we include in the interpolation configuration the boundary points ξ_j which are within the convex hull of the lattice $[x_{k_0}, x_{k_f}] \times [y_{m_0}, y_{m_f}]$. If the conditions 2–3 are still not satisfied, we systematically enlarge the region by decrementing k_0, m_0 and incrementing k_f, m_f until they are.

In Fig. 1, for example, additional interpolation points for cell 4 near the point $\mathbf{q}$ are indicated by the shaded circles on the volume grid and shaded diamonds on the boundary. Thus, this cell is interpolation-irregular. For each interior grid point, the integral in (8) is computed for each of the four cells bordering it, which may be shape-irregular as well as interpolation-irregular. The integral for irregular cells requires more computation than for regular cells.

Remark 3.1. *Cell averaging*: Once the polynomial function $U^n_{i,j}(x,y)$ is created for each cell, we define a single valued function $U^n(x,y)$ on all of D. If (x,y) lies in the interior of a cell $C_{i,j}$, then $U^n(x,y) = U^n_{i,j}(x,y)$. If (x,y) lies on an edge between cells $C_{i,j}$ and $C_{i',j'}$, then $U^n(x,y) = \frac{1}{2}[U^n_{i,j}(x,y) + U^n_{i',j'}(x,y)]$. If (x,y) lies at a corner between cells $C_{i,j}$, $C_{i+1,j}$, $C_{i,j+1}$, $C_{i+1,j+1}$, then $U^n(x,y) = \frac{1}{4}[U^n_{i,j}(x,y) + U^n_{i+1,j}(x,y) + U^n_{i,j+1}(x,y) + U^n_{i+1,j+1}(x,y)]$.

4. Algorithm

Although we do not have a complete specification of all the necessary tools at this point, an outline of the algorithm will serve to focus the subsequent discussion.

For time step $n = 1, \ldots$

1. From the current data $\{U^n_{i,j}\}$ compute a bivariate piecewise polynomial interpolant $U^n(\mathbf{x}) = U^n(x,y)$.
2. Differentiate $U^n(\mathbf{x})$ analytically to obtain $\nabla^2 U^n(\mathbf{x})$.
3. Evolve the solution for one time step $\tau = \Delta t$ according to formula (8) as follows:

$$U^{n+1}_{i,j} := 2U^n(x_i, y_j) - U^{n-1}(x_i, y_j) + \int_{B_\tau(x_i,y_j) \bigcap \mathscr{D}} G_d(|(x_i, y_j) - \xi|, \tau)\ \nabla^2 U^n(\xi)\, \mathrm{d}\xi. \tag{21}$$

The integral in (21) can be carried out analytically or by quadrature.

4. For boundary value problems, the boundary correction term must be included.

$$U^{n+1}_{i,j} := U^{n+1}_{i,j} + \Psi(x_i, y_j, t_{n+1}).$$

4.1. For Dirichlet conditions, solve the integral Eq. (15) and add in Ψ given by (13).

4.2. For Neumann conditions, solve the integral Eq. (18) and add in Ψ given by (17).

Remark 4.1. *Breaking time symmetry:* Note that the first two quantities on the right hand side of the discrete evolution formula (21) are cell-averaged interpolants, according to Remark 3.1. We referred to this as strong u-consistency in our previous paper [13]. This averaging procedure breaks time-symmetry in the evolution formula and, by smoothing, acts like a low-pass filter. (A similar observation is made in [18].) This notion is supported by numerical experiments because a variety of schemes which are stable when $U^n(x_i, y_j)$ is used are not stable if the (unaveraged) grid value $U^n_{i,j}$ is used. In this paper, we only give numerical results for schemes which use the averaging procedure.

Remark 4.2. The integral

$$\int_{B_\tau(x_i,y_j)\bigcap\mathscr{D}} G_d(|(x_i,y_j)-\xi|,\tau)\ \nabla^2 U^n(\xi)\,\mathrm{d}\xi, \tag{22}$$

in (21) is computed in four pieces. Since $\Delta t = h$, the domain of integration $(B_\tau(x_i,y_j)\bigcap\mathscr{D})$ is the union of the four subdomains $\{B_\tau(x_i,y_j)\bigcap D_{i,j},\ B_\tau(x_i,y_j)\bigcap D_{i-1,j},\ B_\tau(x_i,y_j)\bigcap D_{i,j-1},\ B_\tau(x_i,y_j)\bigcap D_{i-1,j-1}\}$. For a given polynomial basis function, this integral is exactly the same for all regular shaped cells but is different for each irregular shaped cell near the boundary. We pre-compute and store these integrals for each needed polynomial basis function.

5. The boundary correction step

We turn now to a discussion of the integral Eq. (15) for Dirichlet boundary conditions. Since the boundary ∂D is sufficiently smooth, we assume it is parameterized by the differentiable function $\xi(\gamma):[\alpha,\beta]\to\partial D$ with L boundary discretization points $\{\xi_l\}=\xi(\gamma_l)$. We divide the boundary into L segments, centered at successive points $\{\xi_l=\xi(\gamma_l)\}$. More precisely, we let

$$I_l = \left[\gamma_l - \frac{1}{2}(\gamma_l-\gamma_{l-1}),\quad \gamma_l + \frac{1}{2}(\gamma_{l+1}-\gamma_l)\right].$$

(Indexing, of course, is done modulo L since the boundary is periodic.) On the segment I_l we assume $\mu(\gamma,\tau)$ is a bivariate polynomial of order q in γ and τ,

$$\mu_l(\gamma,\tau) := \sum_{p_1+p_2\leqslant q} a_{p_1,p_2}(\gamma-\gamma_l)^{p_1}\tau^{p_2}. \tag{23}$$

We choose the unknowns to be the values $\mu_l^k := \mu(\gamma_l,\tau_k)$, $l=1,\ldots,L$, $k=0,\ldots,K$, where $\tau_k = k\frac{\tau}{K}$. From these, we fit the polynomial in (23) using μ_m^k, $m=l_0,\ldots,l,l+1,\ldots,l_f$, with $(l_f-l_0+1)=q+1$, and $k=0,\ldots,q$.

With the preceding formulation in mind, it is straightforward but tedious to compute all integrals in (15) either analytically or by quadrature for all piecewise polynomial basis functions describing the double layer density μ.

Since the integral equation is of the second kind, it is well-suited for iterative solution. In fact, it can be shown [9] that the norm of the double layer operator D_0 given in (16) is of the order $\mathrm{O}(\tau)=\mathrm{O}(\Delta t)$. Thus q terms in the Neumann series (fixed point iteration) yields an error of the order $\mathrm{O}(\Delta t^{q+1})$.

Neumann conditions are handled in a similar fashion.

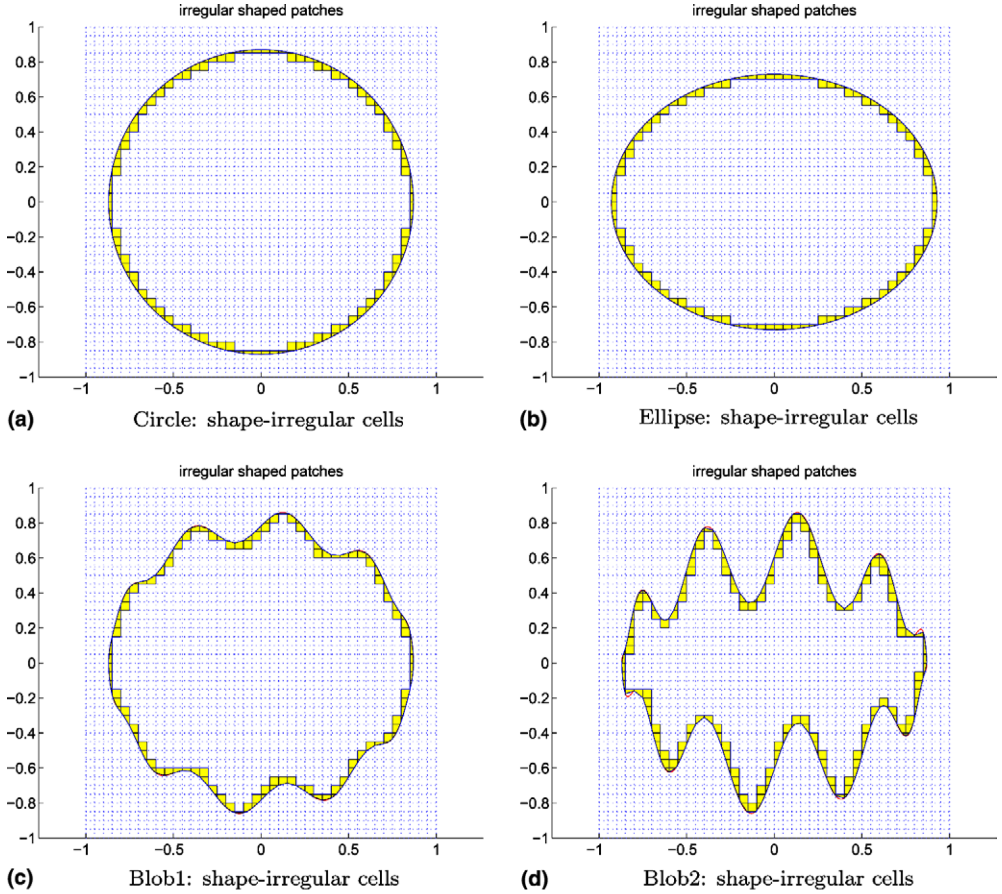
6. Numerical results

The numerical results of this section came from running the numerical schemes described in the previous sections in four geometries: a disk, an ellipse, and two non-convex shapes named “blob1” and “blob2”. The latter two are parameterized by

$$x(\gamma) = 0.87\cos(2\pi\gamma),$$
$$y(\gamma) = 0.87\sin(2\pi\gamma)(1-b+b\sin(20\pi\gamma)),$$

where $b = 0.1$ for “blob1” and $b = 0.3$ for “blob2”. These four shapes are shown in Figs. 2 and 3. Fig. 2 shows the cells cut by the boundary which are irregular in shape (40×40 mesh). Fig. 3 shows the cells which are interpolation-irregular. (The regular configuration is a 6×6 lattice used to interpolate a fourth degree bivariate polynomial.) The irregular cells require special treatment and contribute to the irregularity of the stencils at points near the boundary.

A convergence study was performed using the four geometries and a variety of test functions, including $\sin(ax + by \pm \sqrt{a^2 + b^2}t)$, $(ax + by \pm \sqrt{a^2 + b^2}t)^m$, $\log(|ax + by \pm \sqrt{a^2 + b^2}t|)$, and $e^{ax+by\pm\sqrt{a^2+b^2}t}$. We show in Figs. 4 and 5 the approximation errors (at $t = 2$) using schemes based on second, fourth, and sixth order bivariate polynomial approximations in the volume as well as on the boundary. For the function $\sin(x + y - \sqrt{2}t)$ in Fig. 4, the orders of accuracy of the three schemes are 2, 4, and 6, respectively, as expected. The erratic convergence of the sixth order scheme in the “blob1” geometry seems to warrant explanation. We suspect that there is some mild instability which arises when the boundary is insufficiently sampled compared with the underlying volume grid. In our algorithm, we place just enough boundary points uniformly (in arc length) around the boundary so that the arc length between successive points is approximately (and no larger than) the spacings Δx and Δy in the volume discretization. Thus, a boundary with a larger arc length will automatically have a finer boundary discretization. For example, in the case of



(a) Circle: shape-irregular cells

(b) Ellipse: shape-irregular cells

(c) Blob1: shape-irregular cells

(d) Blob2: shape-irregular cells

Fig. 2. Cells cut by the boundary are called shape-irregular and are shaded in the 40×40 meshes shown.

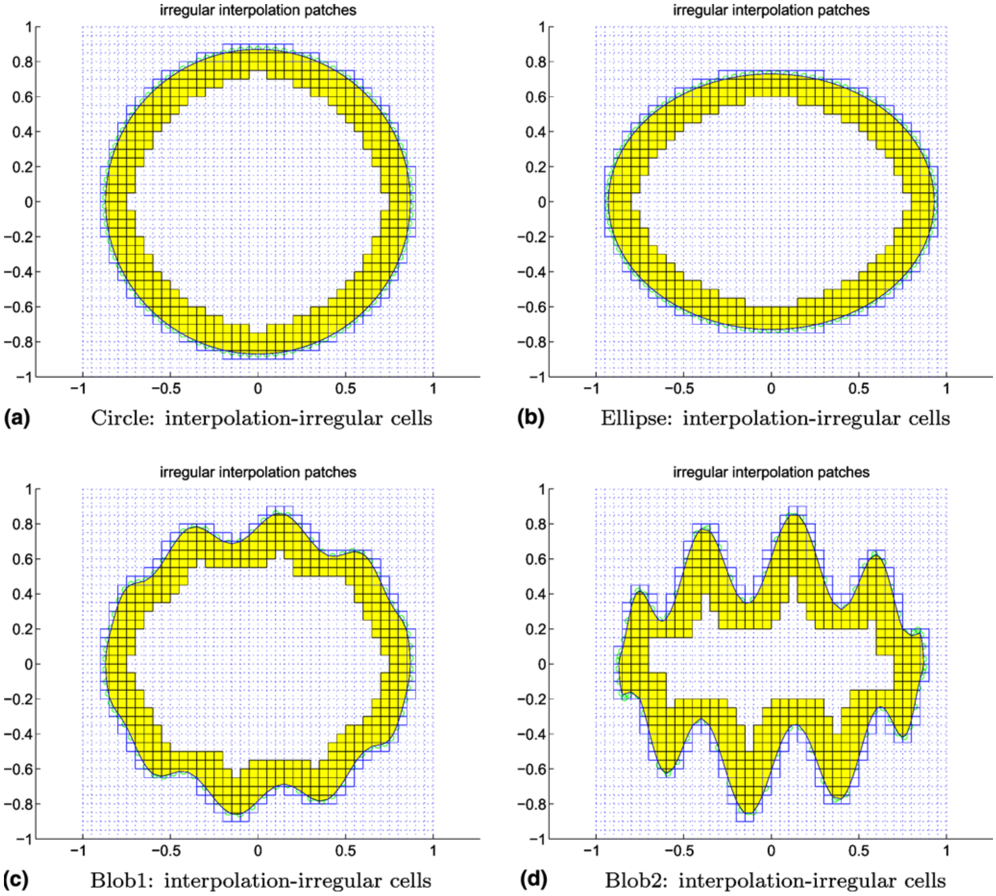

irregular interpolation patches
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
−1
−0.5
0
0.5
1
(a)
Circle: interpolation-irregular cells
(b)
Ellipse: interpolation-irregular cells
(c)
Blob1: interpolation-irregular cells
(d)
Blob2: interpolation-irregular cells

Fig. 3. For fourth order accuracy, the regular interpolation configuration is a 6 × 6 lattice. As a result, a number of cells (shaded) near the boundary are obliged to use an irregular set of interpolation points.

the erratically behaved data point for "blob1", the volume discretization was 160 × 160. The shape "blob1" required 697 boundary points, the more concave shape "blob2" required 1422 boundary points, and the circle and the ellipse required 438 and 483, respectively. We believe the fact that "blob2" exhibited the expected convergence behavior at this volume discretization while "blob1", which would have been expected to be a better-behaved example, did not is due to the fact that "blob2" had more than twice the number of discretization points on the boundary. We tried adding a small number of additional boundary points to "blob1" to improve the behavior of this particular data point, but were not successful. We believe that adding a very large number of boundary points is not called for in this case. Rather, refining the volume grid is the better answer. As can be seen, the behavior of "blob1" at the next finer level of discretization is as expected.

It can also be seen that errors stop decreasing at an L^∞ error of approximately $1e-6$. We attribute this to the severe ill-conditioning associated with our high order two-dimensional polynomial interpolation on a uniformly spaced lattice of points.

For the exponential function in Fig. 5 the orders of accuracy of the three schemes are also approximately 2, 4, and 6, respectively.

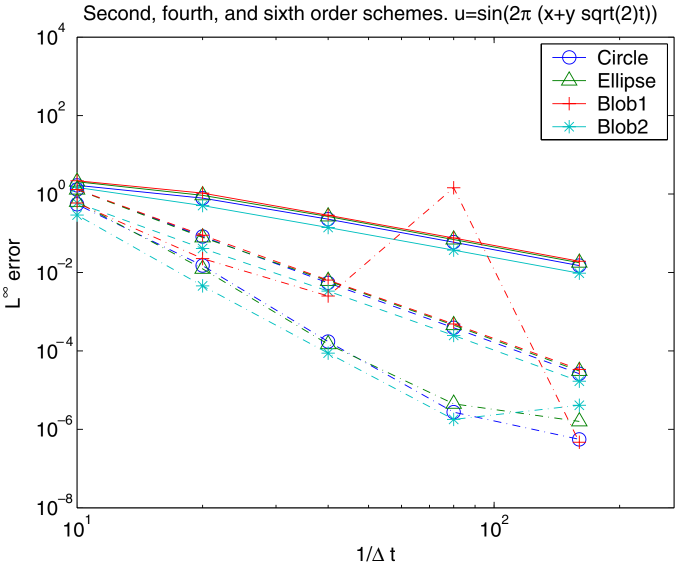


Fig. 4. Approximation errors at $t = 2$ of the second(solid), fourth(dashed), and sixth(dash-dot) order schemes using the test function $u = \sin 2\pi(x + y - \sqrt{2}t)$ in a variety of geometries. The data points correspond to meshes of dimension 20 × 20, 40 × 40, 80 × 80, 160 × 160, and 320 × 320.

The numerical results for test functions $(ax + by \pm \sqrt{a^2 + b^2}t)^m$, $m = 7$, and $\log(|ax + by \pm \sqrt{a^2 + b^2}t|)$ are similar to the sine and exponential test functions. For the sake of brevity we do not include them here.

To test the schemes against a weak instability which becomes apparent only after a very large number of time steps, we ran the fourth order scheme on the test function $u = \sin 2\pi(x + y - \sqrt{2}t)$ (because it oscillates

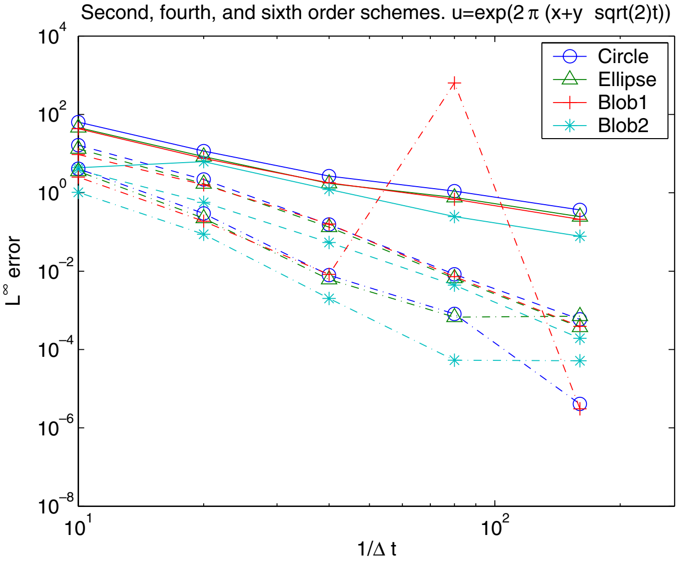


Fig. 5. Approximation errors at $t = 2$ of the second(solid), fourth(dashed), and sixth(dash-dot) order schemes using the test function $u = e^{2\pi(x+y-\sqrt{2}t)}$ in a variety of geometries. The data points correspond to meshes of dimension 20 × 20, 40 × 40, 80 × 80, 160 × 160, and 320 × 320.

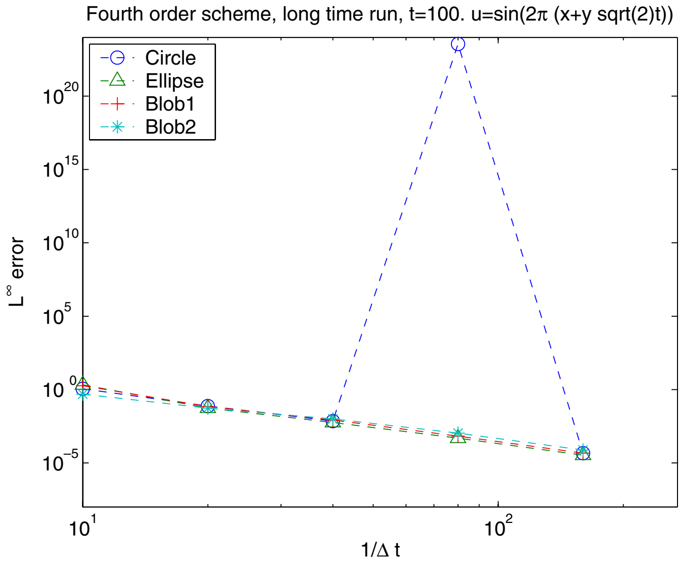


Fig. 6. Long time approximation errors at $t = 100$ of the fourth order scheme using the test function $u = \sin 2\pi(x + y - \sqrt{2}t)$ in a variety of geometries. The data points correspond to meshes of dimension 20 × 20, 40 × 40, 80 × 80, 160 × 160, and 320 × 320.

indefinitely) until $t = 100$. This corresponds to 16,000 time steps in the finest spatial discretization. The numerical results are plotted in Fig. 6. An instability occurred on the disk at the spatial discretization 160 × 160 which disappeared at the next finer discretization. In all the other three geometries, the scheme remained stable and converged with fourth order accuracy.

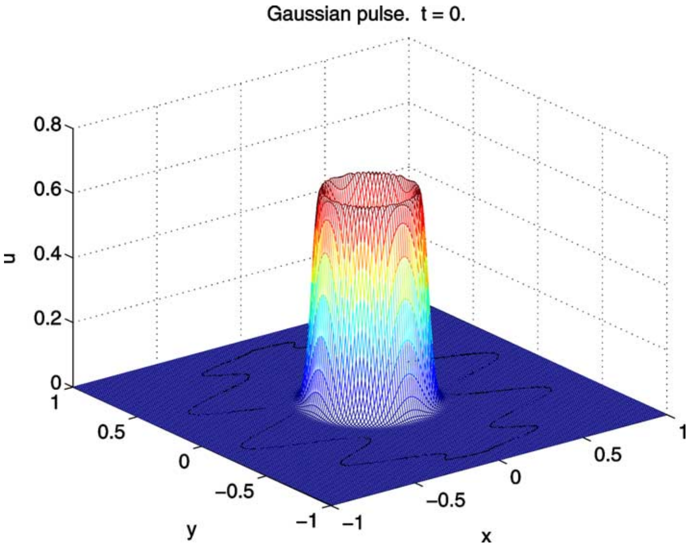
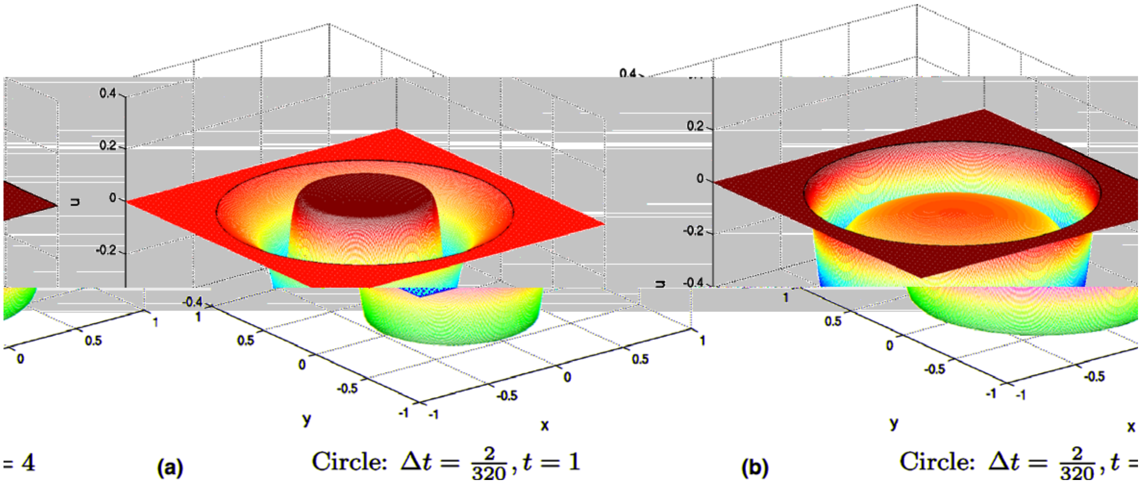


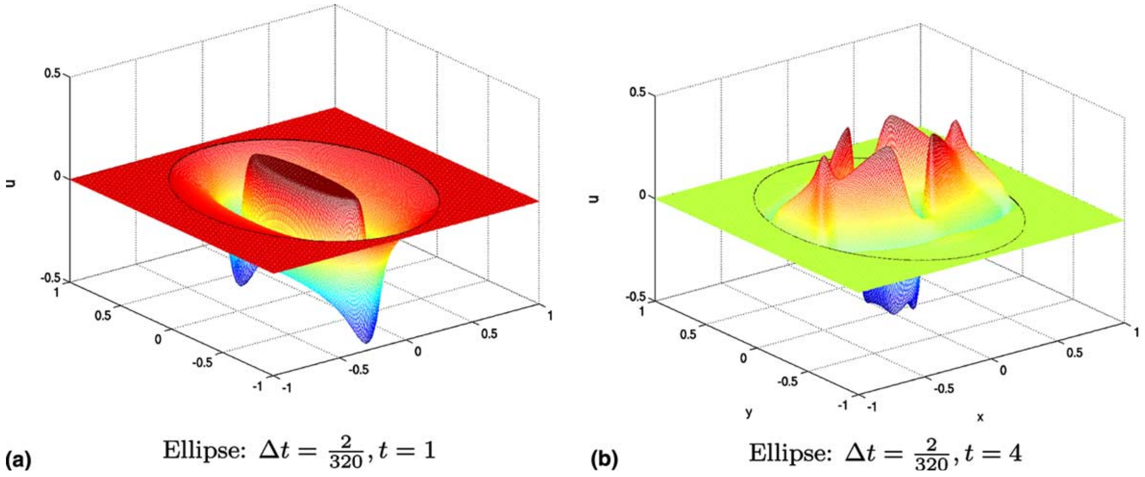
Fig. 7. A Gaussian (in time) source at the origin is turned quickly on and off before time 0 to give this initial data.



Circle: $\Delta t = \frac{2}{320}, t = 1$

Circle: $\Delta t = \frac{2}{320}, t =$

Fig. 8. Solution in a disk from the initial data in Fig. 7 subject to zero Dirichlet boundary conditions. A fourth order accurate scheme and an 320 × 320 mesh were used.



Ellipse: $\Delta t = \frac{2}{320}, t = 1$

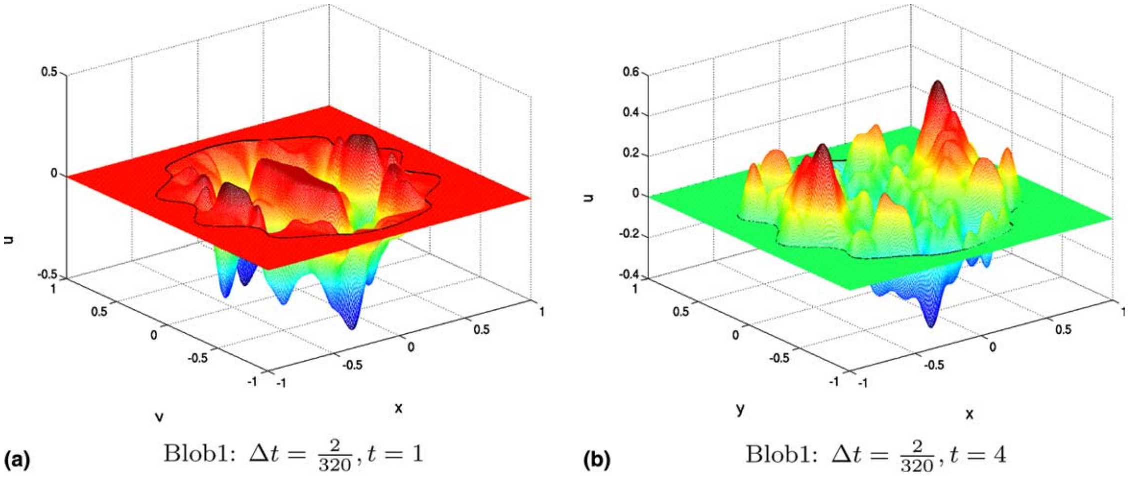
Ellipse: $\Delta t = \frac{2}{320}, t = 4$

Fig. 9. Solution in an ellipse from the initial data in Fig. 7 subject to zero Dirichlet boundary conditions. A fourth order accurate scheme and an 320 × 320 mesh were used.

Finally, we ran the fourth order accurate scheme in the four test geometries using the initial data shown in Fig. 7, subject to zero Dirichlet boundary conditions. The profile in Fig. 7 is obtained by evaluating the function:

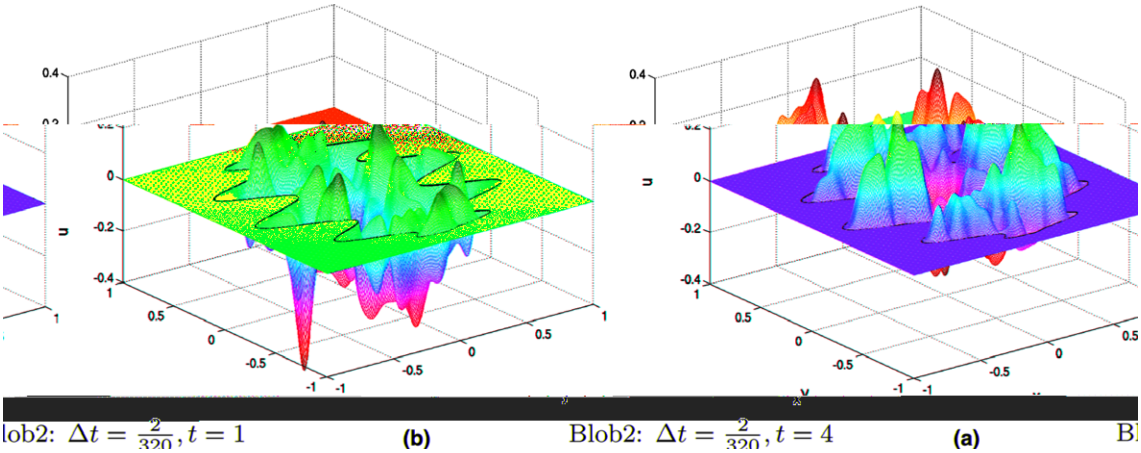
$$u(\mathbf{x},t) = \int_{-\infty}^{t} \frac{1}{\sqrt{(t-\tau)^2 - |\mathbf{x}|^2}} \mathrm{e}^{-(\tau-\tau_0)^2/\sigma} \, \mathrm{d}\tau, \tag{24}$$

at $t = 0$. We evaluate formula (24) at $t = \Delta t$ also in order to initialize the marching scheme. The formula corresponds to a Dirac δ-function source at the origin. By choosing $\tau_0 < 0$ and σ sufficiently small, the source is effectively shut off by the time $t = 0$ and $u(\mathbf{x},t)$ satisfies the homogeneous wave equation with an



(a) Blob1: $\Delta t = \frac{2}{320}, t = 1$ **(b)** Blob1: $\Delta t = \frac{2}{320}, t = 4$

Fig. 10. Solution in the non-convex shape “blob1” from the initial data in Fig. 7 subject to zero Dirichlet boundary conditions. A fourth order accurate scheme and an 320 × 320 mesh were used.



lob2: $\Delta t = \frac{2}{320}, t = 1$ **(b)** Blob2: $\Delta t = \frac{2}{320}, t = 4$ **(a)** Bl

Fig. 11. Solution in the non-convex shape “blob2” from the initial data in Fig. 7 subject to zero Dirichlet boundary conditions. A fourth order accurate scheme and an 320 × 320 mesh were used.

exponentially small error. At later times, since we enforce zero Dirichlet conditions on the domain boundary ∂D, the computed solution is of course no longer equal to the wave function (24).

Figs. 8–11 show the results of the simulation in the four geometries at $t = 1$ and 4. The volume discretization used was 320 × 320 and the number of boundary points for the four geometries are 875(circle), 966(ellipse), 1394(“blob1”), and 2844(“blob2”).

7. Conclusions

We have presented a new explicit numerical method for time-domain wave propagation in two dimensions. The method is based on an exact integral evolution formula [3]. Unlike most finite difference or finite element formulations, it appears to be able to achieve high order accuracy in complex geometry without the need to take excessively small time steps.

References

[1] M.J. Aftosmis, M. Berger, J.E. Melton, Robust and efficient cartesian mesh generation using for component-based geometry, AIAA Journal 36 (6) (1998) 952–960.

[2] A.S. Almgren, J.B. Bell, P. Colella, T. Marthaler, A cartesian mesh method for the incompressible Euler equations in complex geometries, SIAM Journal of Scientific Computing 142 (1) (1997) 1–46.

[3] B. Alpert, L. Greengard, T. Hagstrom, An integral evolution formula for the wave equation, Journal of Computational Physics 162 (2) (2000) 536–543.

[4] B. Alpert, L. Greengard, T. Hagstrom, Rapid evaluation of nonreflecting boundary kernels for time-domain wave propagation, SIAM Journal of Numerical Analysis 37 (4) (2000) 1138–1164 (electronic).

[5] M.J. Berger, R.J. LeVeque, A rotated difference scheme for cartesian grids in complex geometries, in: Proceedings of the AIAA 10th Computational Fluid Dynamics Conference, Honolulu, Hawaii, 1991, pp. 1–9, AIAA Paper CP-91-1602.

[6] W.J. Coirier, K.G. Powell, An accuracy assessment of Cartesian-mesh approaches for the Euler equations, Journal of Computational Physics 117 (1) (1995) 121–131.

[7] A. Ditkowski, K. Dridi, J.S. Hesthaven, Convergent cartesian grid methods for Maxwell's equations in complex geometries, Journal of Computational Physics 170 (1) (2001) 39–80, June.

[8] A.A. Ergin, B. Shanker, E. Michielssen, Fast evaluation of three dimensional transient wave fields using diagonal translation operators, Journal of Computational Physics 146 (1) (1998) 157–180.

[9] R.B. Guenther, J.W. Lee, Partial Differential Equations of Mathematical Physics and Integral Equations, Prentice Hall, Inglewood Cliffs, NJ, 1988.

[10] D. Jiao, M.Y. Lu, E. Michielssen, J.M. Jin, A fast time-domain finite element-boundary integral method method for electromagnetic analysis, IEEE Transactions of Antennuation Propagation 49 (10) (2001) 1453–1461.

[11] H.-O. Kreiss, N. Anders Petersson, J. Yström, Difference approximations for the second order wave equation, SIAM Journal of Numerical Analysis 40 (5) (2002) 1940–1967 (electronic).

[12] R.J. LeVeque, A large time step generalization of Godunov's method for systems of conservation laws, SIAM Journal of Numerical Analysis 22 (6) (1985) 1051–1073.

[13] J.-R. Li, L. Greengard, Strongly consistent marching schemes for the wave equation, Journal of Computational Physics 188 (1) (2003) 194–208, June.

[14] M.Y. Lu, J.G. Wang, A.A. Ergin, E. Michielssen, Fast evaluation of two dimensional transient wave fields, Journal of Computational Physics 158 (2) (2000) 161–185.

[15] J.J. Quirk, An alternative to unstructured grids for computing gas dynamic flows around arbitrarily complex two-dimensional bodies, Computers and Fluids 23 (1) (1994) 42–125, January.

[16] R.D. Richtmyer, K.W. Morton, Difference Methods for Initial-value Problems, Interscience Publishers and John Wiley & Sons, Inc, New York, London, Sydney, 1967.

[17] J.C. Strikwerda, Finite Difference Schemes and Partial Differential Equations, Wadsworth & Brooks/Cole Advanced Books & Software, Pacific Grove, CA, 1989.

[18] J. Visher, S. Wandzura, A. White, Stable, high-order discretization for evolution of the wave equation in $1+1$ dimensions, J. Comput. Phys. 194 (2) (2004) 395–408.