arXiv:1412.3057v1 [math.NA] 9 Dec 2014

ADAPTIVE FINITE DIFFERENCE METHODS FOR NONLINEAR ELLIPTIC AND PARABOLIC PARTIAL DIFFERENTIAL EQUATIONS WITH FREE BOUNDARIES

A. OBERMAN, AND I. ZWIERS

Abstract. Monotone finite difference methods provide stable convergent discretizations of a class of degenerate elliptic and parabolic Partial Differential Equations (PDEs). These methods are best suited to regular rectangular grids, which leads to low accuracy near curved boundaries or singularities of solutions. In this article we combine monotone finite difference methods with an adaptive grid refinement technique to produce a PDE discretization and solver with applied to a broad class of equations, in curved or unbounded domains which include free boundaries. The grid refinement is flexible and adaptive. The discretization is combined with a fast solution method, which incorporates asynchronous time stepping adapted to the spatial scale. The framework is validated on linear problems in curved and unbounded domains. Key applications include the obstacle problem and the one-phase Stefan free boundary problem.

1. Introduction

In this article we numerically approximate a class of nonlinear elliptic and parabolic PDEs using monotone finite difference methods. Finite difference methods are most easily implemented on regular, rectangular grids. In this article we combine the monotone finite difference methods with an adaptive quadtree grid, resulting in significantly improved accuracy near boundaries. The effectiveness of the method is demonstrated on the Laplace equation on curved and on unbounded domains. Key applications are the obstacle problem, and the Stefan Free Boundary problems.

Using the framework of nonlinear elliptic operators, we can combine the partial differential equation with the boundary conditions (or even free boundaries) into a *single* degenerate elliptic operator. This allows us to build adaptive discretizations and solvers using a unified framework, and to experiment with different grid adaptation strategies.

Adaptive finite difference methods have been used in a similar context in a variety of problems, but a not in a framework as general as this one. A review of data structures and implementation of sparse grids for Partial Differential Equations can be found in [BG04]. Many approaches using finite differences methods combine the popular level set method for tracking the boundary with a representation of the operator inside the boundary. A fourth order adaptive method for the heat equation and stefan equation can be found in [GF05]. Adaptive grids for the Stefan

Date: January 27, 2015.

Key words and phrases. finite difference methods; adaptive grids; elliptic partial differential equations; obstacle problem; free boundary problems; Stefan problem; monotone finite difference methods .

problem were used in [CMG09]. Adaptive grid refinement combined with a level set representation of the free boundary was used for the Poisson-Boltman system in [HG11, MTG11].

An advantage of the finite difference implementation and the viscosity solution framework is that the conditioning of the solvers does not break down as the equation becomes degenerate. For example, fast solvers for the degenerate elliptic Monge-Ampere equation have been built, where the Newton's method solver speed is (nearly) independent of the regularity of the solutions [FO11, FO13]. These problems were solved on a uniform grid using wide stencil finite difference schemes, but the later article extended the problem to Optimal Transportation boundary conditions, where the source domain is irregular, and the target domain is convex [BFO14]. However the anisotropy of the operator requires wide stencils for monotone discretizations, which are more challenging to implement on an adaptive grid.

In order to work with an adaptive grid, we need a refinement criteria. We take the point of view that the equation itself should provide this criteria. By writing the entire problem (including boundary conditions) as a *single degenerate elliptic operator* we are able to produce an effective refinement criteria.

The framework can be used for many purposes, including:

- Artificial Boundary Conditions for problems in an unbounded domain. We use coarse grids in the exterior, and choose to adapt based on either the residual of the boundary conditions or the distance from a reference point in the domain.
- Grid adaptation for PDEs on curved domains, using grid based discretizations.
- Obstacle problems or one phase free boundary problems such as the Stefan problem.
- Nonlinear iterative methods for stationary problems. On an adaptive grid, the iterations are asynchronous, so there the nonlinear CFL condition is locally determined.

1.1. **The framework of degenerate elliptic operators.** We consider the the class of degenerate elliptic equations [CIL92], which include first order equations, such as the eikonal equation, as well as fully nonlinear PDEs, such as the Monge-Ampere equation, and free boundary problems. Singularities can be present in the solutions to these equation, in particular at locations near the free boundary or where the equation changes types. For this reason, weak solutions, are needed, which are the viscosity solutions [CIL92]. The theory of viscosity solutions is by now well-established. To prove convergence of the schemes, we require that that uniqueness hold for the underlying PDE. In most cases, this is covered by the standard theory. Classical solutions of the Stefan problem arise only under limited conditions [DL05]. For the one phase Stefan problem, uniqueness of viscosity solutions is established in [Kim03].

Let Ω be a domain in $\mathbb{R}^n$, Du and D^2u denote the gradient and Hessian of u, respectively, and let $F(X, p, r, x)$ be a continuous real valued function defined on $\mathbb{S}^n \times \mathbb{R}^n \times \mathbb{R} \times \Omega$, $\mathbb{S}^n$ being the space of symmetric $n \times n$ matrices. Write

$$F[u](x) \equiv F(D^2u(x), Du(x), u(x), x).$$

Definition 1.1. *The operator F is* degenerate elliptic *if*

$$F(X,p,r,x) \leq F(Y,p,s,x) \text{ whenever } r \leq s \text{ and } Y \leq X,$$

where $Y \leq X$ means that $Y - X$ is a nonnegative definite symmetric matrix.

If the operator F is degenerate elliptic, then we say the Partial Differential Equation on the domain Ω

$$F[u](x) = 0, \quad \text{for } x \text{ in}$$

(along with, for example, Dirichlet boundary conditions, $u(x) = g(x)$, or x on $\partial\Omega$) is as well. The initial-boundary value problem for the

$$u_t(x,t) + F[u](x,t)$$

is called degenerate parabolic, when the operator F is degenerate elliptic.

Example 1. The obstacle problem, $\min(-u_{xx}, u - g(x)) = 0$, is degenerate elliptic. The Hamilton-Jacobi equation, $u_t - |u_x| = 0$, is degenerate parabolic.

1.2. **Elliptic finite difference methods.** The class of finite difference methods (or equations) we focus on are called *elliptic*, [Obe06]. They are a special class of monotone finite difference schemes which are automatically stable, and arise from a simple construction. Consistent elliptic schemes, since they are monotone and stable, converge, according to the theory presented in [BS91].

Finite difference equations can be defined on a general unstructured grid, regarded as a weighted, directed graph. In our case the adaptive finite difference grid has a natural data structure given by the quadtree, which is discussed below. But to define monotone schemes, we can consider the abstract setting. The unstructured grid on the domain Ω; is a directed graph consisting of a set of points, $x_i \in \Omega, i = 1, \dots N$, each endowed with a list of neighbors, $N(i) = (i_1, \dots, i_d)$. A *grid function* is a real-valued function defined on the grid, with values $u_i = u(x_i)$. The finite difference operator is represented at each grid point by

$$F^i[u] \equiv F^i\left(u_i, \frac{u_i - u_{i_1}}{|x_i - x_{i_1}|}, \dots, \frac{u_i - u_{i_d}}{|x_i - x_{i_d}|}\right), \quad i = 1, \dots N, \tag{1}$$

where $F^i(x, y_1, \dots, y_d)$ is a specified, usually nonlinear, function of its arguments. The list of finite differences in the above expression can be regarded as the gradient of the function on the graph. The notation

$$\nabla u(x_i) = \left(\frac{u_i - u_{i_1}}{|x_i - x_{i_1}|}, \dots, \frac{u_i - u_{i_d}}{|x_i - x_{i_d}|}\right)$$

was used in [MOS12], so that we can write

$$F^i[u] = F^i(u_i, \nabla u(x_i)).$$

This notation emphasizes the fact that a finite difference operator is local: it depends only on the value at the reference points, and the gradient of the function on the graph. (Second order finite differences come from combinations of first order differences; higher order differences are not needed). A *solution* is a grid function which satisfies $F[u] = 0$ (at all grid points). A *boundary* point can be identified as a grid point with no neighbours, so that Dirichlet boundary conditions can be imposed by setting $F^i[u] = u_i - g(x_i)$.

We now define degenerate elliptic operators.

Definition 1.2. *The finite difference operator F is* degenerate elliptic *if each component $F^i(x, y_1, \ldots, y_d)$ is nondecreasing in each variable.*

We emphasize that the scheme is a nondecreasing function of u_i and the differences $u_i - u_j$ for neighbours j of i.

1.3. **Boundary conditions and far field boundary conditions.** Here we show how to include boundary conditions on more general (non-rectangular) domains, as well as far field boundary conditions. These boundary conditions are combined along with the elliptic PDE operator into a single (possibly discontinuous) elliptic operator, which is combined with a refinement criteria to perform the grid adaptation.

Example 2. Consider the Poisson equation $-\Delta u = f$, with f supported on the unit ball and the far field boundary condition $u \to 0$, as $\|x\| \to \infty$. An adaptive grid allows us to capture fine details for x near 0 while reducing computational effort in the far field. The artificial boundary condition,

$$\partial_r u + \frac{u}{r} \approx 0, \qquad \text{for } r \gg 0,$$

approximates the solution with accuracy $\mathcal{O}\left(\frac{1}{r^3}\right)$ [BGT82].

Example 3. Consider the Dirichlet problem for the domain $\Omega \subset B = [0,1]^2$ in $\mathbb{R}^2$,

$$-\Delta u(x) = f(x), \qquad \text{for } x \in \Omega$$

along with boundary conditions

$$u = g, \qquad \text{for } x \text{ on } \partial\Omega$$

Define the (discontinuous in x) degenerate elliptic operator,

$$F^{bc}[u] = \chi_\Omega(x)\left(-\Delta u(x) + f(x)\right) + \chi_{\Omega^c}(x)(u(x) - g(x))$$

where χ_S is the characteristic function of the set S. Then we can solve $F^{bc}[u] = 0$ on B and recover the solution of the Dirichlet problem inside Ω.

We can impose other (for example Neumann or Robin boundary conditions), by replacing the second term with

$$\chi_{\Omega^c}(x) H(Du(x), x) = 0,$$

where $H(Du(x), x)$ is itself a first order degenerate elliptic operator.

1.4. **Including free boundaries in a single degenerate elliptic operator.** The obstacle problem can be formulated as a variational inequality [KS00, Glo84], which is naturally discretized using finite element methods [BHR77], and solved using a multigrid method [GK09].

Our approach of adaptive finite difference methods is natural for the obstacle problem, using a formulation of the problem as a degenerate elliptic PDE, however there are far fewer works which use this approach. Our framework leads to a simple, effective finite difference method which achieves good results using adaptive grids.

We describe here how to write a free boundary problem as a single degenerate elliptic operator.

Example 4. The obstacle problem, for a given obstacle function $g(x)$, which requires that $u(x) \geq g(x)$ and that

$$-\Delta u(x) = 0, \quad \text{for } x \text{ in } \{u(x) > g(x)\}$$

can be written as a single elliptic equation

$$F^{obs}[u] = \min(-\Delta u, u - g) = 0$$

This example generalizes to double obstacle problems (using a maximum as well as a minimum), as well as obstacle problems involving nonlinear PDEs which replace the Laplacian [Obe06].

Example 5. The evolution of the one-phase Stefan problem in two dimensions,

$$\begin{cases} u_t - \Delta u = 0 & \text{in } \{u > 0\} \\ u_t - |Du|^2 = 0 & \text{on } \partial\{u = 0\} \end{cases}$$

can be represented by the degenerate elliptic operator,

$$u_t + F^{Stef}[u] = 0$$

with

$$F^{Stef}[u] = \begin{cases} -\Delta u, & \text{in } \{u > 0\} \\ \min(-\Delta u, -|Du|^2) & \text{in } \{u \leq 0\}. \end{cases}$$

More general one phase free boundary problems can be represented as a single operator on the extended domain, as in [Obe06]. Here we have extended from the free boundary to a larger domain, and we solve for the extended operator in the whole domain.

2. Adaptive grid

Our adaptive grid is implemented using a quadtree representation [DBVKOS00, Chapter 14: Quadtrees]. Conceptually, the domain is divided into rectangular regions such that the sidelength of each neighbouring rectangle is either twice, half, or the same as its neighbour. The collection of all vertices are the grid nodes for computing the unknown function. Internally the quadtree is represented as a sparse matrix where the indices of non-zero entries represent coordinates on a fixed ultra-fine grid.

Our implementation with with sparse matrices in MATLAB. The tool is modular, and the inputs are simple: the discretization of the operator, F, and an additional operator, G, used as the refinement criteria, which can be intrinsic (simply setting $G = F$), or defined by the user. In addition, if Newton's method is to be used as a solver, the formal Jacobian of the operator is needed, DF.

2.1. Quadtree construction.

A quadtree is uniquely determined by a list of coordinates and corresponding maximum length scales. Either there is a node at each coordinate with all neighbours within the specified distance, or the coordinate must lie within a rectangle no larger than indicated.

To discretize the Laplacian, we impose an additional 'scale-padding' constraints depending on the aspect ratio of the physical domain. Dangling nodes, vertices with three neighbours, occur midway along the shared edge of two equal-sized rectangles one of which is subdivided. The scale-padding constraint in x specifies the minimum number of equal-sized rectangles that must exist to both the left and

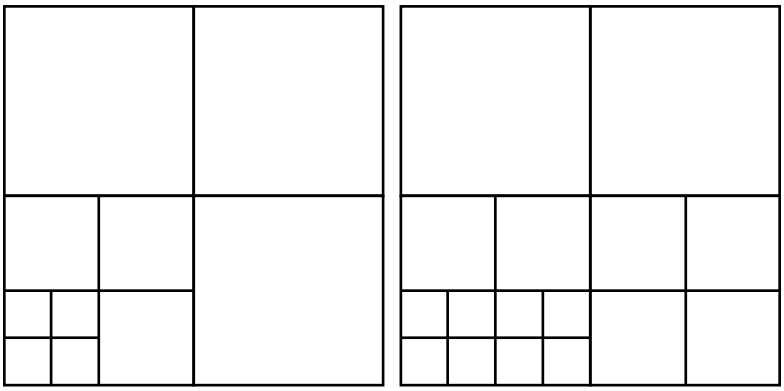

FIGURE 1. (a) Sample quadtree, consistent with scale-padding constraints $\text{pad}_x = 1$ and $\text{pad}_y = 1$. (b) Refinement of (a) to be consistent with $\text{pad}_x = 2$

right of a dangling node. Figure 1 illustrates a pair of quadtrees, the latter refined to observe the scale-padding constraints 2 in x and 1 in y.

To built a quadtree over a virtual ultra-fine grid of $2^N + 1$ by $2^N + 1$, we build a list of squares the quadtree must contain.

(1) List all requested or required squares size $2^k + 1$ by $2^k + 1$.
(2) Add siblings to the list. Fill to the edge, if needed to prevent a dangling node too close to the boundary.
(3) List all parents. Expand list to satisfy scale-padding constraints.
(4) Use grandparents to ensure the expanded list of parents includes all of their siblings. This is the list of all required squares size $2^{k+1} + 1$ by $2^{k+1} + 1$.

Given M requested coordinate-length scale pairs, this procedure is $\mathcal{O}(NM \log M)$.

2.2. **Adaptivity.** Any refinement scheme based on position and the local value of the function and its derivatives may be specified. A "refinement criteria" operator is computed at all current grid points and compared with a "refinement threshold". The new grid must be refined to the finest available spacing at all nodes where the criteria exceeds the threshold.

For better control, the user may supply a non-decreasing array of refinement threshold values. Where the criteria exceeds the k^{th} threshold the new grid is refined to at least the k^{th}-finest scale.

The user may also supply a padding parameter, above that required for discretization of the Laplacian at dangling nodes. It is often convenient to give a simple refinement criteria and a large padding parameter.

All adaptive grids must include the fixed initial quadtree, which is determined by the placement of the initial data. The initial data is treated as scattered and linearly interpolated onto the smallest quadtree for which every supplied data-point appears as a node. We assume the placement of initial data implies the minimally acceptable spatial resolution.

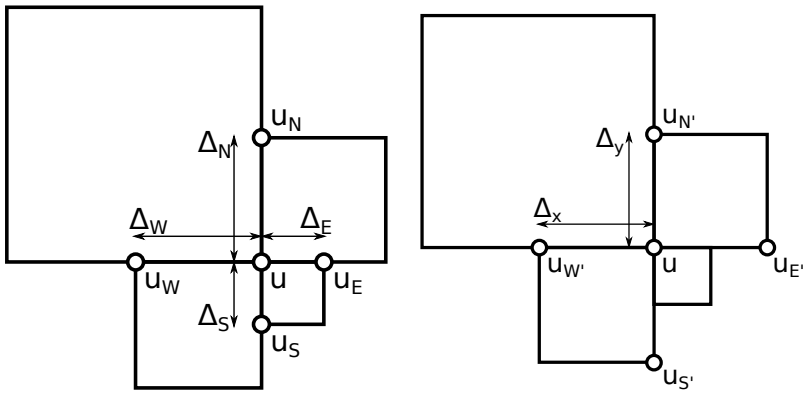

FIGURE 2. One of six possible configurations at a regular node. (a) The stencil for the discretization of first-derivatives. (b) The stencil for the Laplacian discretization.

3. Discretization on the adaptive grid

As mentioned above, degenerate elliptic schemes are easily built from the upwind schemes for derivatives and Laplacian. Our solvers expect the user to specify their operators in terms of these building blocks. For convenience, the discretization discussed here is kept in a black box.

We will consider regular nodes, dangling nodes, and boundary nodes separately. Dangling nodes are those with only three neighbours and occur midway along the edge of a rectangle that adjoins two half-size rectangles.

A regular node is the shared vertex of four rectangles. Consider the nearest neighbours u_E, u_W, u_N and u_S at distances $\Delta_E, \Delta_W, \Delta_N$ and Δ_S respectively, as in Figure 2. The standard upwind discretizations are:

$$\partial_x u \approx \frac{u - u_W}{\Delta_W} \quad \text{and} \quad -\partial_x u \approx \frac{u - u_E}{\Delta_E}, \tag{2}$$

both accurate to first order. For the Laplacian operator we identify the nearest pairs of equidistant opposing nodes, $u_{E'}$ and $u_{W'}$, and $u_{N'}$ and $u_{S'}$, as in Figure 2. The standard discretization for $\partial_x^2 u$,

$$-\partial_x^2 u \approx \frac{2u - u_{E'} - u_{W'}}{2(\Delta_x)^2}, \tag{3}$$

is accurate to second order. Discretization using only nearest-neighbours is only accurate to first order.

For a dangling node, as in Figure 3, we use the farther vertices of the larger square to interpolate a value for the unknown function directly opposite, then discretize as at a regular node. In the illustrated situation we would use,

$$\partial_x u \approx \frac{u - \frac{u_{NE} + u_{SE}}{2}}{\Delta_x}. \tag{4}$$

This is a monotone discretization, accurate to first order since u_N and u_S are equidistant. Second derivatives are more difficult. There is no general upwind discretization for $\partial_x^2 u$ for the dangling node illustrated in Figure 3.

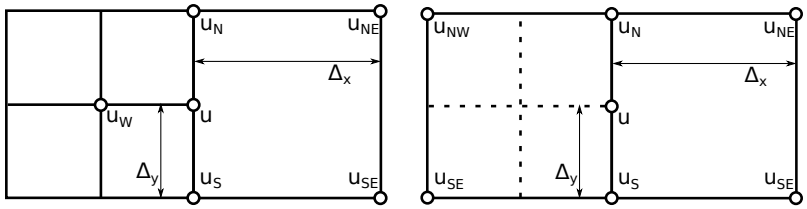

FIGURE 3. (a) A dangling node in the x variable, showing the stencil for first-derivative discretization. (b) The stencil for the Laplacian discretization at a dangling node, provided $\Delta y \le \Delta x$.

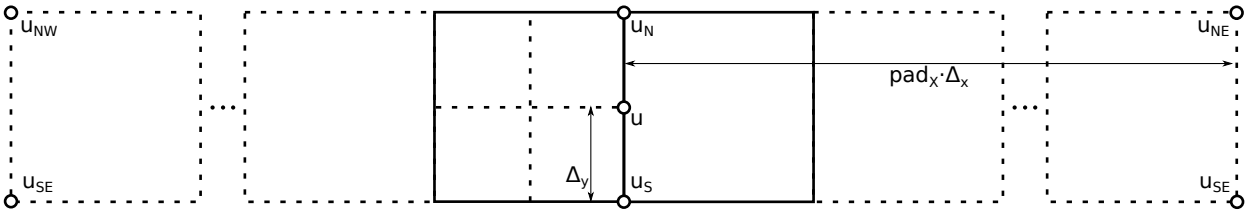

FIGURE 4. The extended stencil for laplacian discretization at a dangling node, when $\Delta y \not\le \Delta x$.

At dangling nodes, we choose to discretize the laplacian using an I-shaped stencil as in Figure 3. The following expansions are accurate to second-order:

$$2u - \frac{u_{NW} + u_{SW} + u_{NE} + u_{SE}}{2} = -(\Delta y)^2 \partial_y^2 u - (\Delta x)^2 \partial_x^2 u + \mathcal{O}(\Delta x^4 + \Delta y^4)$$
$$2u - (u_N + u_S) = -(\Delta y)^2 \partial_y^2 u + \mathcal{O}(\Delta y^4)$$

Our discretized Laplacian is then:

$$-\partial_y^2 u - \partial_x^2 u \approx \frac{1}{(\Delta x)^2}\left(2u - \frac{u_{NW} + u_{SW} + u_{NE} + u_{SE}}{2}\right) + \left(\frac{1}{(\Delta y)^2} - \frac{1}{(\Delta x)^2}\right)(2u - (u_N + u_S)). \tag{5}$$

This is a monotone discretization provided $\Delta y \le \Delta x$. Should $\Delta y > \Delta x$ we take a wider I-shaped stencil, as in Figure 4. To ensure $\Delta y \le \mathrm{pad}_x \Delta x$, (or, $\Delta x \le \mathrm{pad}_y \Delta y$, for the other type of dangling node,) we choose,

$$\mathrm{pad}_x = \left\lceil \frac{L_y}{2L_x} \right\rceil. \quad \text{and} \quad \mathrm{pad}_y = \left\lceil \frac{L_x}{2L_y} \right\rceil.$$

These values reflect the aspect ratio of the domain in physical variables. The 'scale-padding' constraints when building the quadtree guarantee this wider stencil is available at all dangling nodes.

3.1. **Boundary nodes.** At boundary nodes where Dirichlet boundary conditions are not provided, we implement generic Robin boundary conditions:

$$A(x)u'(x) + B(x)u(x) = C(x)$$

The functions A, B, C should be provided for each edge of the domain. User-friendly shortcuts for Neumann or Dirichlet conditions are provided.

Where $A(x) = 0$, $u(x)$ is specified. The node is considered inactive and is not updated by means of a logical mask. Where $A(x) \neq 0$, we use the boundary conditions to determine the outward derivative and the regular discretization for the inward derivative. We discretize the second derivative by weighting the outward and inward derivatives equally, as in (3).

4. Solvers

4.1. **Static solver.** We apply Newton's method. The formal Jacobian of the operator by its variable arguments is required. The proper Jacobian with respect to $u(x)$ is formed internally. It is sparse with the number of nonzero elements the same order as the number of nodes. We expect Gaussian elimination in $\mathcal{O}(M \log M)$ time, where M is the number of nodes. This makes Newton's method very fast.

For further efficiency, we seek a solution at coarse scales before allowing refinement to finer scales. Starting with the initial quadtree we iterate Newton's method until a stopping condition is reached. We then allow refinement up to the second-coarsest scale present in the initial quadtree as called for by the refinement criteria and threshold. We then seek another solution and repeat the process until all scales are allowed.

Facility for stopping criteria and thresholds are similar to those for refinement. When allowing refinement to the k^{th}-finest scale, iteration of Newton's method continues until the stopping criteria is less than the k^{th} stopping threshold at all nodes. The default stopping criteria is the L^∞ norm of the degenerate elliptic operator.

Note that when seeking a solution over a fixed multi-scale grid it is more efficient to define the multiscale grid through the refinement criteria and provide initial data only on a coarse grid.

4.2. **Time-dependent solver.** Before time evolution, we apply the refinement scheme to the linear interpolation of the initial data and interate to ensure infill of any coarse regions of the initial quadtree that are nevertheless of interest.

Nodes are separated according to the distance to their nearest neighbour. A nonlinear CFL condition is required to determine a characteristic time-scale for each group. The time-scales differ by powers of two. We list each group according to the inverse ratio of their characteristic time-scales and the coarsest time-scale. The result is randomly permuted before each time-step to produce a visitation schedule. To evolve by one time-step, all nodes in each group are simultaneously updated according to the visitation schedule.

This scheme is optimal in the sense that each group-update operation is of the order of the number of nodes in the group, and that no more updates occur than required by the nonlinear CFL condition.

5. Computational Examples

In this section we present numerical results, which show the validity and performance of the method, and allow for us to demonstrate the effectiveness of different refinement strategies.

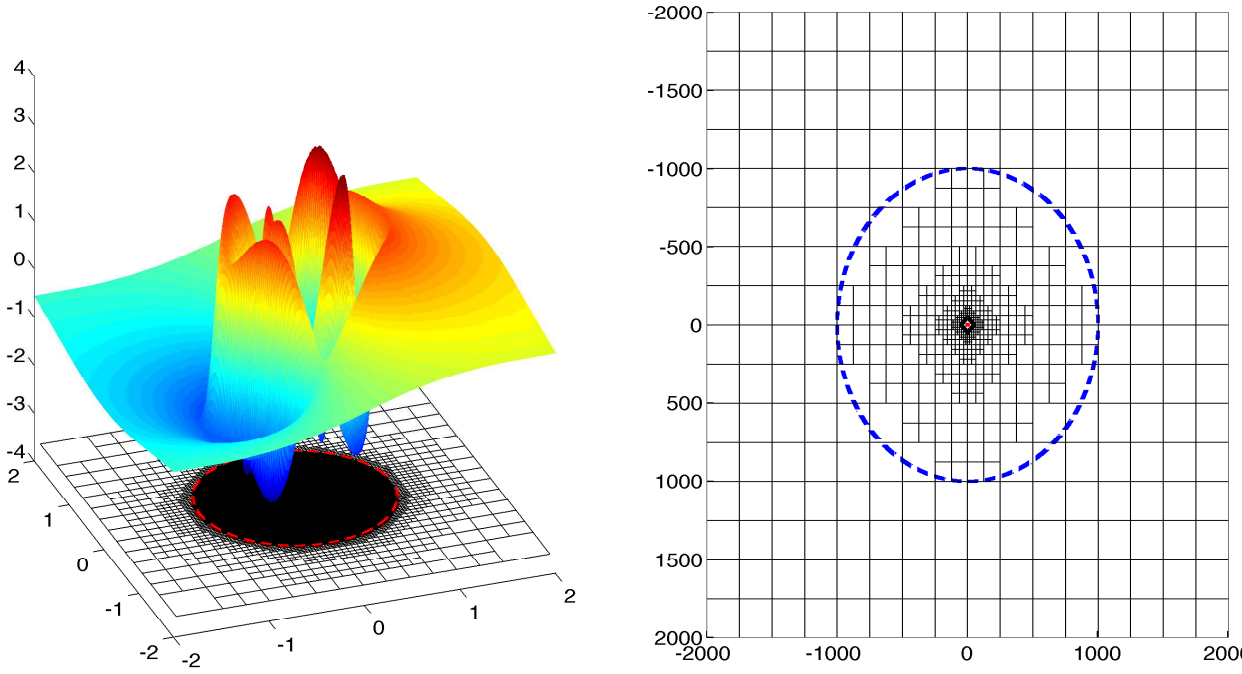

FIGURE 5. (a) Detail of the solution of the Poisson equation with $f(x)$ supported on for $r \leq 1$ (indicated by the red dashed circle). (b) The corresponding adapted grid. Artificial boundary conditions are applied for $r > 10^3$ (indicated by the blue dashed circle).

5.1. **Artificial boundary conditions.** We are in the setting of Example 2: the Poisson equation $-\Delta u = f$, with f supported on the unit ball and $u \to 0$, as $\|x\| \to \infty$. Set

$$f(r,\theta) = r(1-r)^+ \sin(5\pi r)\cos(3\theta)$$

and impose the artificial boundary condition, $\partial_r u + \frac{u}{r} = 0$ at the boundary of the computational domain. Set the domain to be a square domain with side length 2×10^3. The refinement criteria: the grid should be finest for $r < 1$.

A detailed view of the solution in the near field can be see in Figure 5(a). The layout of the grid at large scale can be seen in Figure 5(b). Table 1 outlines the allocation of computing resources and nodes by region. Broadly speaking, the adaptive grid allows us to compute on a very large domain, with a computational cost on the order of (within a couple multiples of) restricting to the unit square.

5.2. **Irregular domains.** Consider a problem of the type Example 3, where the Dirichlet problem is posed on an irregular domain Ω, contained in a rectangle. For the Poisson equation with Dirichlet BCs we use the operator F^{bc} The refinement criterion used was based on the combination of residual of the operator and the

Region	Relative Area	Time Spent	Nodes on Final Grid
$r < 1$	$\ll 0.001\%$	38.5%	78.3%
$1 < r < 10$	0.002%	13.3%	15.0%
$10 < r < 10^3$	19.6%	26.6%	5.2%
$10^3 < r$	80.4%	21.6%	1.5%

TABLE 1. Resource use compared to detail achieved, by domain region, for the example solution of the Poisson equation with artificial BCs. Artificial BCs were applied for $r > 10^3$.

proximity to the boundary. An example of what can be accomplished is shown in Figure 6. Notice that this leads to a maximal refinement in two blobs near the boundary, near local extreme points of the solution (red, where the solution value is near 60 and blue, where the solution value is near -100), while other areas near the boundary have a relatively coarse grid (yellow, where the solution is near 0).

To impose Neumann or Robin BCs, we apply the boundary conditions for grid points near the boundary, and further away, simply impose $u = 0$. As an example, Figure 7 presents results for the Laplace equation with homogeneous Dirichlet boundary conditions on the boundary of the unit square, combined with inhomogeneous Neumann boundary conditions

$$\frac{du}{dn} = 1, \quad \text{for } x \text{ on the boundary of a punctured circle inside the domain.}$$

The adaptive grid was determined by the slope of the solution combined with proximity to the interior boundary.

5.3. **Obstacle problems.** We are in the setting of the obstacle problem, Example 4, represented by the operator F^{obs}.

The obstacle $g(r, \theta) = r^2 \cos^2(\theta)$, multiplied by a factor of $2\sin^2(\pi y)$ for $x < 0$, and by $\exp(-r)$ for $r > \frac{1}{4}$. The obstacle and solution are shown in Figure 8.

The problem was solved using different refinement criteria, defined as follows. The contact contour determined in all three cases was virtually indistinguishable.

- As a baseline method, we used a simple predetermined (non-adaptive) grid criteria. The the finest grid resolution is specified by the distance to the local maxima x_1, x_2, x_3 of the obstacle (since the contact set is unknown).

$$G(x) \text{ determined by } (\text{dist}(x, \{x_1, x_2, x_3\}))$$

In this case, the solution was found using nonlinear multigrid, allowing a progressively finer grid each time the residual drops below a threshold.

- The free boundary-determined grid criteria specifies the finest grid resolution at nodes where both terms $F[u]$ and $u - g$ are close to zero.

$$G^T[u] \text{ determined by } \min(|\Delta u|, |u - g|)$$

The resulting grid provides the most refinement near the boundary of the contact set, as seen in Figure 9.

- We also chose to refine the grid at nodes where the absolute value of operator $F^{obs}[u] = \min(-\Delta[u], u - g(x))$ exceeds a threshold,

$$G^F[u] \text{ determined by } |F^{obs}[u]|.$$

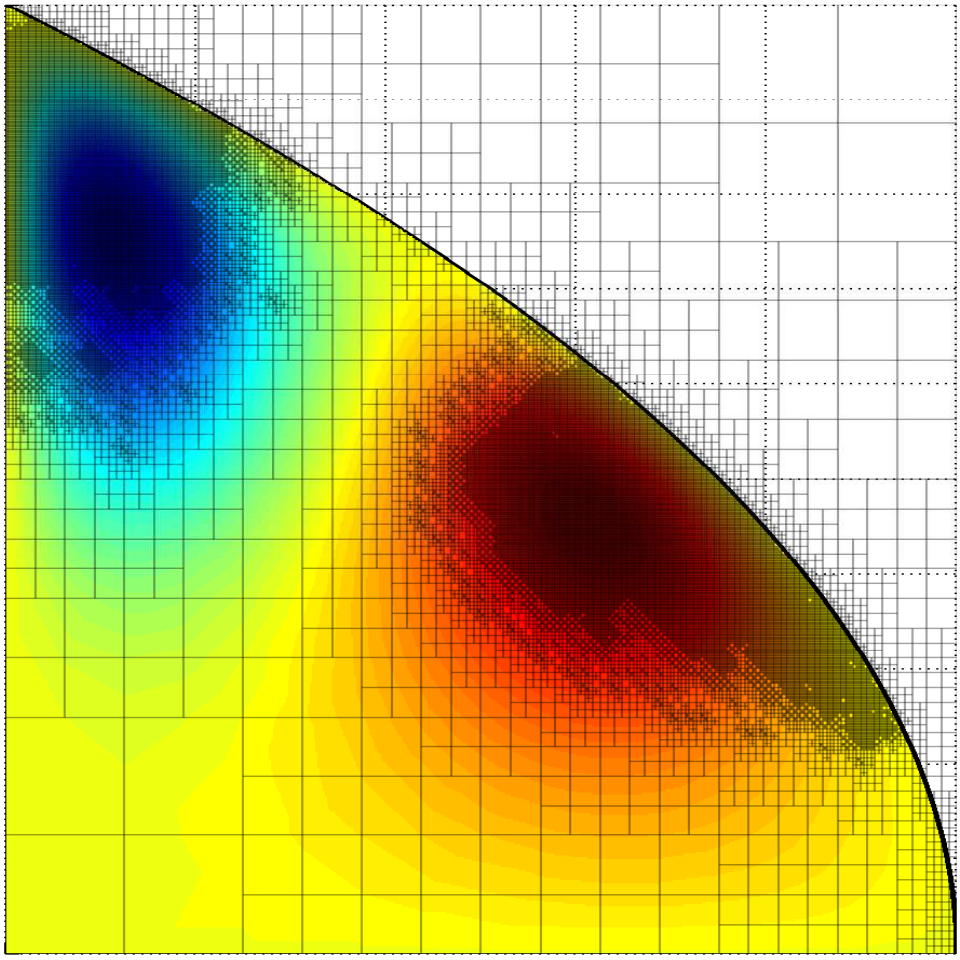

FIGURE 6. Solution of a Poisson equation on a curved domain with Dirichlet boundary conditions.

Notice in this case that the scaling of the two terms are different: the Laplacian scales like $1/h^2$ while the obstacle term has no scaling in h. The resulting grid is very similar to the previous one, but with more refinement inside the contact set (which corresponds to capturing details of $g(x)$) , see Figure 9. More Newton iterates were performed for the operator-determined grid, however these are often performed when the grid is still coarse resulting in overall slightly better performance.

The relative performance of the different refinement methods is given in Table 2.

5.4. **Stefan free boundary problems.** We are now in the context of Example 5, where the Stefan problem is represented by the single operator F^{Stef}.

We take initial data corresponding to a function with three local maxima. To test the effectiveness of different grid adaptation strategies we solve the equation using:

- a uniform coarse grid,

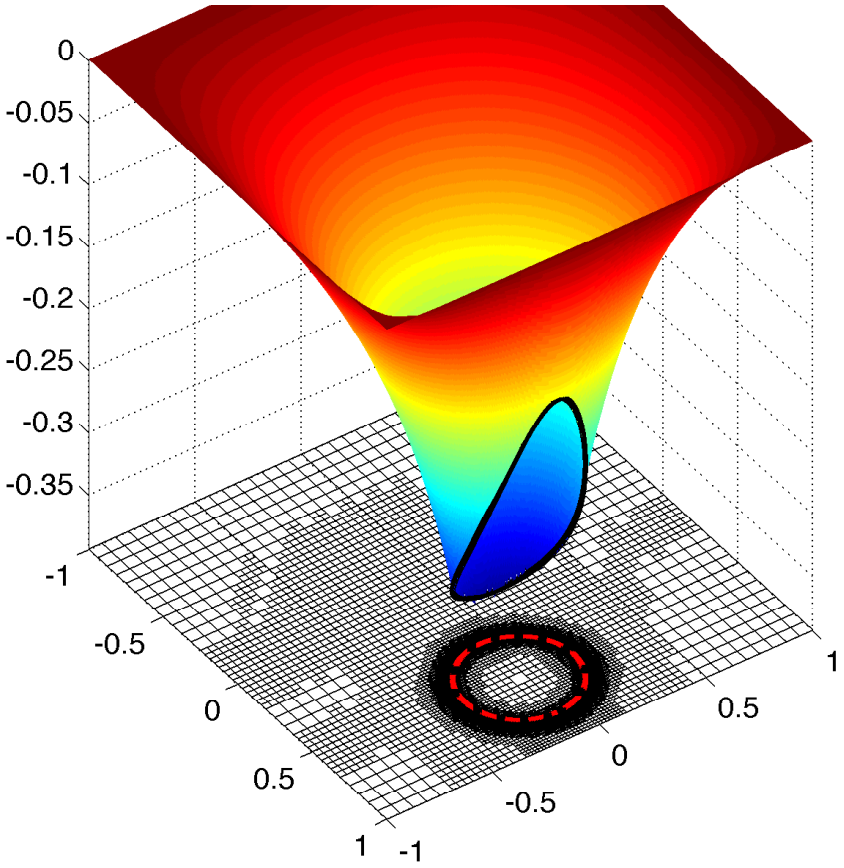

FIGURE 7. Solution of Laplace equation on a punctured domain, with inhomogeneous Neumann boundary conditions on the red circle.

Grid Type	Max nodes	Runtime	Iterates	
			with < 5000 nodes	total
Predetermined	22148	6.00s	21	59
Boundary	15156	5.70s	25	67
Operator	15967	5.54s	33	71

TABLE 2. Comparison of adaptive grids for the obstacle problem.

- a uniform fine grid,
- adapting the grid according to the size of the operator, $G^F[u] = |F^{Stef[u]}|$,
- adapting the grid according to the size of both terms in the operator $G^T[u] = \min(|\Delta u|, |\nabla u|^2)$.

The reason for choosing the term adapted refinement G^T comes from assuming that most of the accuracy of the solution comes from the accuracy of the free boundary: the term G^T refines near the free boundary. The operator-adapted

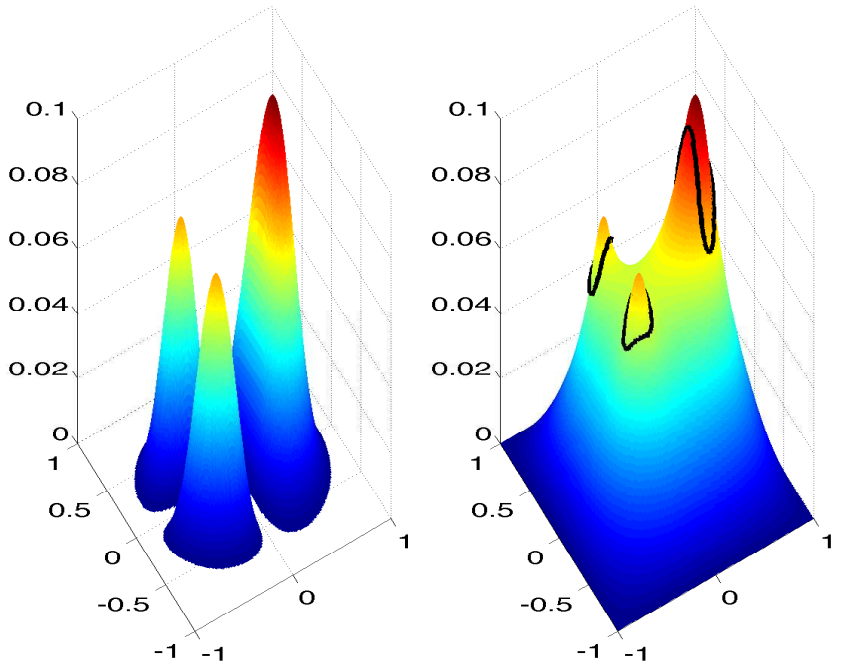

FIGURE 8. (left) Obstacle. (right) solution of the obstacle problem (with contact contour in black).

refinement is intrinsic, it also scales correctly in terms of the grid, since both terms are order $1/h^2$.

Figure 10 shows the observed computational complexity and outlines the L^∞ proximity of the three methods to the fine-grid solution. By this metric, using G^F, the operator-adaptive grid, is superior to using G^T, the both terms adapted grid. On closer inspection, Figure 11 shows the G^T grid is finest near $\partial\{u = 0\}$, as desired, but that the solution is evolving slowly in this region. The operator F^{Stef} is relatively small near the boundary. By comparing the implied boundary-curves, Figure 12 demonstrates clearly that the operator-adapted grid is a better strategy, because the accuracy of the location of the free boundary is significantly better in this case.

6. Conclusions

We introduced a general framework for bringing adaptive grid and solvers to bear on a class of degenerate elliptic and parabolic Partial Differential Equations, which allows for the incorporation of free boundary problems, irregular and unbounded domains, along with adaptive grid refinement. We have demonstrated the significant improvement of solution accuracy and solution time, as compared to methods on regular grids.

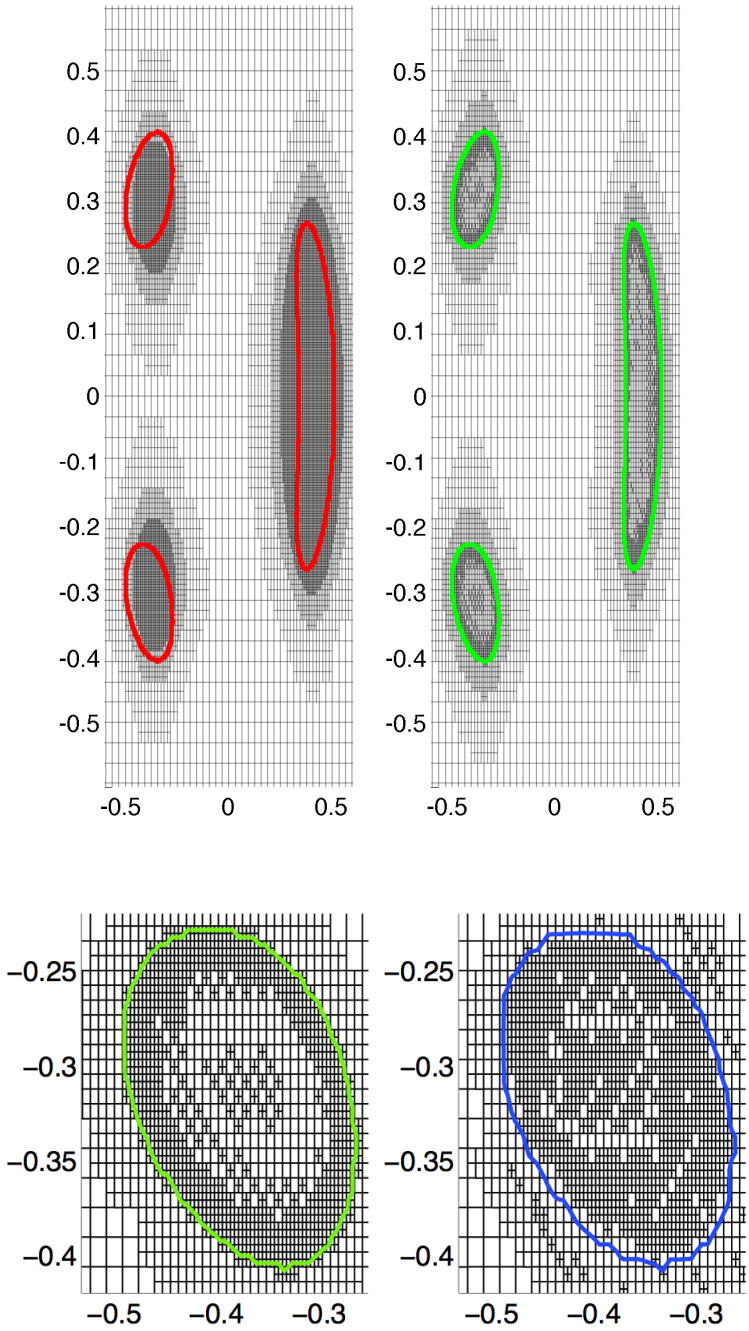

FIGURE 9. (top) Detail of predetermined grid (left) and boundary-determined grid (right), with contact contour. (bottom) Zoomed in detail of boundary-determined grid (left) and operator-determined grid (right), with contact contour.

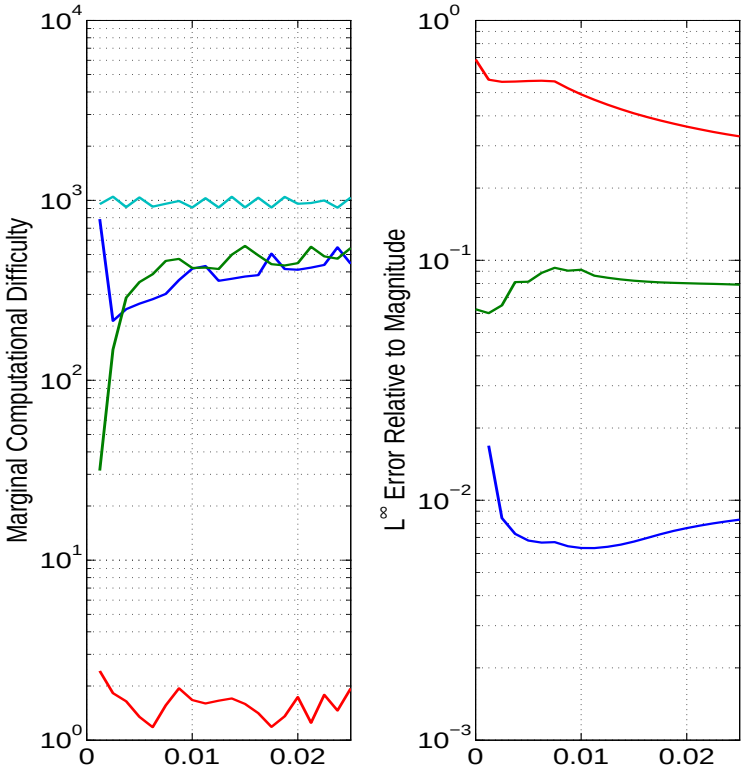

FIGURE 10. Performance of the uniform-coarse grid (red), boundary-adaptive grid (green), operator-adaptive grid (blue) & uniform-fine grid (light blue) when solving the sample Stefan equation.

The adaptive grid overcomes the low accuracy of the finite difference method where curved boundaries are involved. This includes the free boundaries which arise in the obstacle problem, or the Stefan problem. By incorporating the boundary conditions etc into the operator, we can define a global residual, which included errors from the geometry as well as form the error. Using this criteria we developed a grid refinement criteria which resulted in improvements over the other methods.

Other applications of the framework which are easily implemented include:

- First order equations, such as the eikonal equation in either bounded or unbounded domains.
- Visibility problems, with refinement near essential small features of the obstructions
- Optimal or stochastic control problems with refinement at switching regions, so long as the second order operator can be discretized on the grid.
- One phase free boundary problems, such as Hele-Shaw.

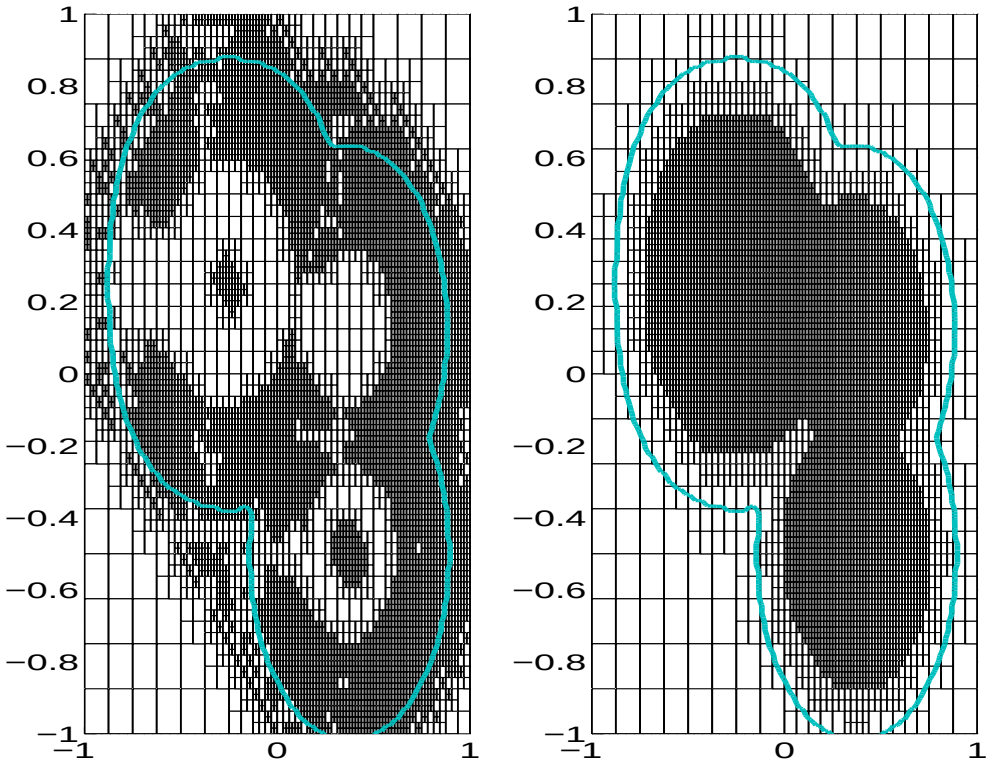

FIGURE 11. Detail of the term-adapted (left) and operator-adapted (right) grids with the boundary contour of the uniform-fine grid solution of the sample Stefan equation at $t = 0.005$.

REFERENCES

[BFO14] Jean-David Benamou, Brittany D. Froese, and Adam M. Oberman. Numerical solution of the optimal transportation problem using the Monge-Ampère equation. *J. Comput. Phys.*, 260:107–126, 2014.

[BG04] Hans-Joachim Bungartz and Michael Griebel. Sparse grids. *Acta numerica*, 13:147–269, 2004.

[BGT82] Alvin Bayliss, Max Gunzburger, and Eli Turkel. Boundary conditions for the numerical solution of elliptic equations in exterior regions. *SIAM Journal on Applied Mathematics*, 42(2):430–451, 1982.

[BHR77] Franco Brezzi, William W Hager, and Pierre-Arnaud Raviart. Error estimates for the finite element solution of variational inequalities. *Numerische Mathematik*, 28(4):431–443, 1977.

[BS91] Guy Barles and Panagiotis E. Souganidis. Convergence of approximation schemes for fully nonlinear second order equations. *Asymptotic Anal.*, 4(3):271–283, 1991.

[CIL92] Michael G. Crandall, Hitoshi Ishii, and Pierre-Louis Lions. User's guide to viscosity solutions of second order partial differential equations. *Bull. Amer. Math. Soc. (N.S.)*, 27(1):1–67, 1992.

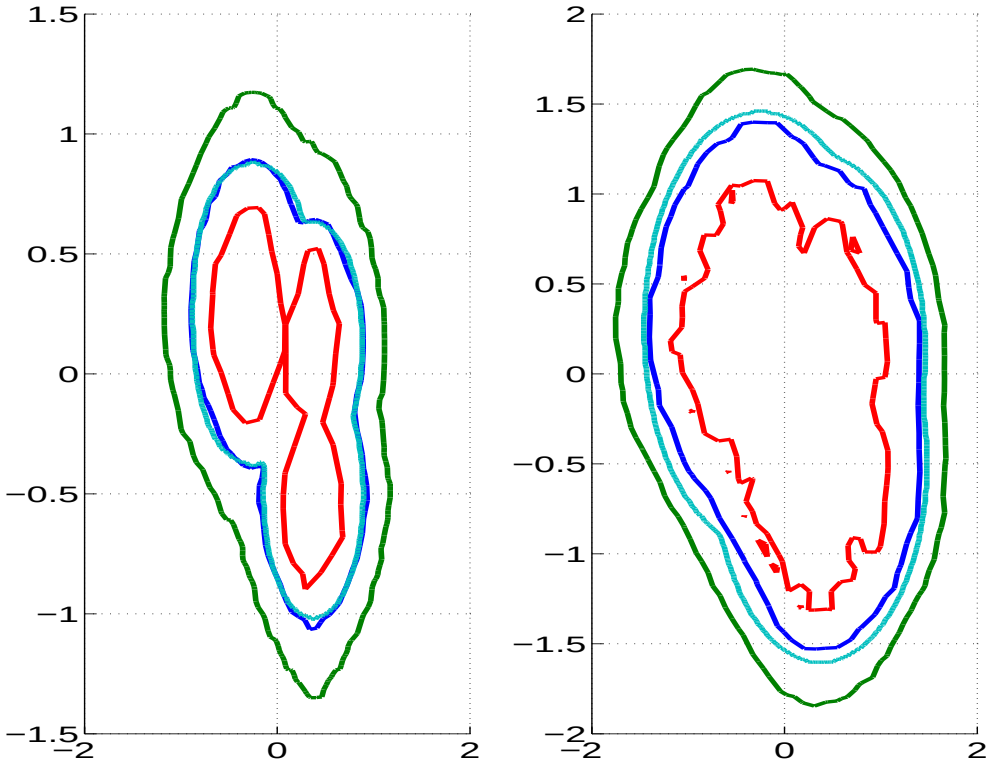

FIGURE 12. Curves $\partial\{u = 0\}$ of the sample Stefan problem at $t = 0.005$ (left) and $t = 0.025$ (right), due to solution with a uniform-coarse grid (red), term-adapted grid (green), operator-adapted grid (blue) & uniform-fine grid (light blue).

[CMG09] Han Chen, Chohong Min, and Frédéric Gibou. A numerical scheme for the stefan problem on adaptive cartesian grids with supralinear convergence rate. *Journal of Computational Physics*, 228(16):5803–5818, 2009.

[DBVKOS00] Mark De Berg, Marc Van Kreveld, Mark Overmars, and Otfried Cheong Schwarzkopf. *Computational geometry*. Springer, 2000.

[DL05] P. Daskalopoulos and Ki-Ahm Lee. All time smooth solutions of the one-phase stefan problem and the hele-shaw flow. *Communications in Partial Differential Equations*, 29(1-2):71–89, 2005.

[FO11] Brittany D. Froese and Adam M. Oberman. Fast finite difference solvers for singular solutions of the elliptic Monge-Ampère equation. *J. Comput. Phys.*, 230(3):818–834, 2011.

[FO13] Brittany D Froese and Adam M Oberman. Convergent filtered schemes for the monge–ampère partial differential equation. *SIAM Journal on Numerical Analysis*, 51(1):423–444, 2013.

[GF05] Frédéric Gibou and Ronald Fedkiw. A fourth order accurate discretization for the laplace and heat equations on arbitrary domains, with applications to the stefan problem. *Journal of Computational Physics*, 202(2):577–601, 2005.

[GK09] Carsten Gräser and Ralf Kornhuber. Multigrid methods for obstacle problems. *Journal of Computational Mathematics*, 27(1):1–44, 2009.

[Glo84] Roland Glowinski. *Numerical methods for nonlinear variational problems*, volume 4. Springer, 1984.

[HG11] Ásdís Helgadóttir and Frédéric Gibou. A poisson–boltzmann solver on irregular domains with neumann or robin boundary conditions on non-graded adaptive grid. *Journal of Computational Physics*, 230(10):3830–3848, 2011.

[Kim03] Inwon C Kim. Uniqueness and existence results on the hele-shaw and the stefan problems. *Archive for rational mechanics and analysis*, 168(4):299–328, 2003.

[KS00] David Kinderlehrer and Guido Stampacchia. *An introduction to variational inequalities and their applications*, volume 31. Siam, 2000.

[MOS12] Juan J Manfredi, Adam M Oberman, and Alex P Svirodov. Nonlinear elliptic partial differential equations and p-harmonic functions on graphs. *arXiv preprint arXiv:1212.0834*, 2012.

[MTG11] Mohammad Mirzadeh, Maxime Theillard, and Frédéric Gibou. A second-order discretization of the nonlinear poisson–boltzmann equation over irregular geometries using non-graded adaptive cartesian grids. *Journal of Computational Physics*, 230(5):2125–2140, 2011.

[Obe06] Adam M. Oberman. Convergent difference schemes for degenerate elliptic and parabolic equations: Hamilton-Jacobi equations and free boundary problems. *SIAM J. Numer. Anal.*, 44(2):879–895 (electronic), 2006.