SIAM J. SCI. COMPUT.
Vol. 25, No. 6, pp. 1982–2003

A BOUNDARY CONDITION–CAPTURING MULTIGRID APPROACH TO IRREGULAR BOUNDARY PROBLEMS*

JUSTIN W. L. WAN† AND XU-DONG LIU‡

Abstract. We propose a geometric multigrid method for solving linear systems arising from irregular boundary problems involving multiple interfaces in two and three dimensions. In this method, we adopt a matrix-free approach; i.e., we do not form the fine grid matrix explicitly and we never form nor store the coarse grid matrices, as many other robust multigrid methods do. The main idea is to construct an accurate interpolation which captures the correct boundary conditions at the interfaces via a level set function. Numerical results are given to compare our multigrid method with black box and algebraic multigrid methods.

Key words. irregular boundary, boundary condition, multigrid, interpolation

AMS subject classifications. 15A09, 15A23, 65F10, 65F50, 65Y05

DOI. 10.1137/S1064827503428540

1. Introduction.

In many applications and simulations—for instance, in the simulation of epitaxial thin film growth using the island dynamics model [7, 9, 14] or water simulation modeled as incompressible two-phase flow [26]—one needs to solve second order elliptic partial differential equations (PDEs) of the form

$$-\nabla a(x)\nabla u = f \qquad \text{on } \Omega, \tag{1.1}$$

where $\overline{\Omega} = \cup_{i=1}^{m} \overline{\Omega}_i$ and $\{\Omega_i\}$ are disjoint subsets. Let $\Gamma_i = \partial\Omega_i$ be the (internal) interface. The model problem is to solve the PDE subject to boundary conditions on $\partial\Omega$ and coupling conditions on Γ_i, $i = 1, 2, \ldots, m$. We say the model problem is of *Dirichlet type* if a Dirichlet boundary condition is used on Γ_i, and of *interface type* if jump conditions are used instead. For instance, the epitaxial thin film growth simulation can be modeled as an irregular boundary problem of Dirichlet type, and water simulation modeled as interface type. We shall make these more precise in sections 2 and 3.

The challenge in solving irregular problems is that the shapes of the interfaces can be very complex and do not, in general, align with any Cartesian grid. One approach is to use finite element or finite volume discretizations together with unstructured triangulated grids [13, 18], which can easily adapt to complex geometries. However, the interfaces often move in each time step in a time-dependent calculation, e.g., water surfaces. In such a case, one needs to do regridding in every time step, which can be expensive.

An alternative approach is to use the level set formulation [22] to represent complex interfaces on an Eulerian grid, typically a Cartesian grid. No regridding is necessary. Instead, the level set function is updated, or evolved, in each time step according to the level set equation. In this paper, we adopt this approach for representing interfaces.

*Received by the editors May 28, 2003; accepted for publication (in revised form) October 30, 2003; published electronically May 25, 2004.
http://www.siam.org/journals/sisc/25-6/42854.html
†Department of Computer Science, University of Waterloo, Waterloo, ON, Canada N2L 3G1 (jwlwan@math.uwaterloo.ca). The work of this author was supported by the Natural Sciences and Engineering Research Council of Canada and was partially supported by NSF grant ACI-0072112.
‡Department of Mathematics, University of California at Santa Barbara, Santa Barbara, CA (xliu@math.ucsb.edu).

Discretization methods have been proposed to solve elliptic PDEs with an interface; see, for instance, [19] (see also the references therein for other methods). Regardless of which method is used, one needs to solve a linear system arising from the discretization, which is the most time-consuming part, in a time-dependent calculation. For example, in [19], preconditioned conjugate gradient (PCG) with incomplete Cholesky factorization as preconditioner is used to solve the linear system. Although PCG is more efficient than classical relaxation methods, as pointed out in [19], the rate of convergence still depends on the mesh size h. In practice, the convergence rate is often observed to be $O(h^{-1})$. Moreover, the complexity constant depends on PDE coefficients and geometry. Thus, for large problems and nonsmooth PDE coefficients on complex domains, many PCG iterations are required to obtain accurate solutions. Another preconditioner based on deflation can be found in [12], which shows improved results for discontinuous coefficient problems.

In this paper, we are interested in efficient iterative methods for solving linear systems arising from irregular boundary problems involving multiple (possibly moving) interfaces in two and three dimensions. In addition to the issue of mesh size dependence of convergence rate, memory storage for the fine grid matrix addresses another issue since the problem size in real applications can be in the order of millions, despite whether the fine grid matrix is sparse. In the following sections, we propose a geometric multigrid method which employs a matrix-free approach; i.e., we do not form the fine grid matrix explicitly. Specifically, we just need the discretization stencil at each grid point. Also, we never form nor store the coarse grid matrices, which is a common practice in many robust multigrid methods. What is necessary is the geometric information of the interface which is given implicitly through the level set functions. The main idea of our approach is to construct an interpolation which captures the correct boundary conditions at the interface. Thus, we allow jump discontinuity at the interface as well as multiple interfaces. To the best of our knowledge, the issue of multiple interfaces has not been addressed much in the multigrid literature.

Multigrid has been proved numerically and theoretically to be an efficient method for solving elliptic PDEs with smooth coefficients on simple geometries such as rectangular grids [5, 15, 28, 31]. The convergence rate is often independent of the mesh size, and typically is around 0.1 to 0.2—textbook multigrid efficiency. However, when the coefficients are not smooth—for instance, they have jump discontinuities of order 10^3 or higher—the convergence rate of standard multigrid can be very slow. Such problems often arise in practice; for instance, in water simulation modeled by incompressible two phase flow, the density ratio 1000:1 of water to air leads to a jump discontinuity of order 10^3.

Algebraic multigrid (AMG) [25] uses purely algebraic techniques to select coarse grid points based on the notion of strong connections and to define the interpolation operator based on algebraic smoothness. Because of its algebraic nature, geometry and dimensionality are not needed in the construction. Instead, the nonsmoothness of the PDE coefficients and the underlying geometry are exploited implicitly through the matrix entries by the matrix-dependent interpolation. However, for problems of Dirichlet type, we found that the AMG coarse grid matrices tend to be much denser than for the smooth coefficient case where a factor of 1/4 and 1/8 reduction in the number of nonzeros is usually observed on the coarse grid matrices in two and three dimensions, respectively. Moreover, since the coarse grid operators are generated algebraically by the Galerkin process, they have to be precomputed, and additional storage is required. However, we note that aggressive coarsening technique has been discussed in [28] to address the issue of denser AMG coarse grid matrices.

Another robust approach is the black box multigrid method [10, 11] (see also [2]). It assumes the matrix is formed by a five-point or nine-point discretization on a (logically) rectangular grid. But other than this assumption, the algorithm is completely algebraic. The coarse grid matrices are formed by the Galerkin process. To handle the nonsmooth coefficients, operator-dependent interpolation is constructed based on the discretization stencil information. The idea is to apply a robust one-dimensional (1D) interpolation (cf. section 2.2) to noncoarse grid points lying on coarse grid lines in two dimensions. To obtain a 1D three-point stencil from a two-dimensional (2D) five-point/seven-point stencil, a lumping technique is used. However, it does not take into account that an interface with zero boundary condition might exist, which can lead to poor interpolation weights. We shall explain this in more details in section 2.3. Because of the five-point or nine-point stencil structure of the fine grid matrix, all the coarse grid matrices, in general, have the nine-point stencil sparsity pattern. Thus the number of nonzeros of the coarse grid matrices is not influenced by the number or the shape of the interfaces, as opposed to the AMG approach. However, due to the use of the Galerkin process, it suffers from the same storage problem as AMG.

Other robust multigrid approaches include matrix-dependent [23, 24, 32], agglomeration [20, 29], and energy minimization [30]. Some of these methods are restricted to two dimensions only, and most of them use Galerkin coarsening, and hence extra storage is needed for the coarse grid operators. Besides, the numerical results presented are mainly for 2D problems, and PDEs involving multiple interfaces are not discussed. Recently, Adams and Li [1] proposed a 2D immersed interface multigrid method. Black box multigrid interpolation is used for grid points away from the interface, and Taylor expansion with a need of change of coordinate is used to derive interpolation weights for grid points near the interface. The three-dimensional (3D) case and multiple interfaces are not discussed, in which case the calculations would possibly be more involved. From their numerical results, the convergence of their multigrid method slows down with increasing jump sizes. Our proposed multigrid method maintains a more uniform convergence with respect to mesh size and the size of the jumps (cf. section 5).

We note that a capacitance matrix preconditioning approach has been studied in [3]. It shows a much improved convergence than PCG for one interface, i.e., $m = 1$. However, it deteriorates when m increases since the associated Schur complement problem becomes denser as more interfaces exist. Such a phenomenon commonly happens in the thin film simulation where small islands start to appear at different regions.

In the following, section 2 will describe the proposed multigrid for Dirichlet-type problems. In section 3, a modified multigrid method is described for interface-type problems. An analysis is given in section 4 to justify the better accuracy of our interpolation. In section 5, the proposed multigrid method is compared with PCG, AMG, and blackbox multigrid methods in several numerical examples. Finally, concluding remarks are given in section 6.

2. Irregular boundary problem of Dirichlet type. In the simulation of the growth of epitaxial thin films using the island dynamics model [9, 14], growth is described by the creation and subsequent motion of island boundaries. For a continuous adatom-density $\rho(x, t)$, the diffusion equation is

$$\frac{\partial \rho}{\partial t} = \nabla \cdot (D \nabla \rho) + F - 2D\sigma_1 \rho^2, \tag{2.1}$$

where D is the diffusion coefficient, F is the flux of atoms to the surface, and σ_1 is the capture number for nucleation. The numerical boundary condition for ρ on a square grid is periodic. For each of the islands, if irreversible aggregation or growth is assumed, the boundary condition imposed on the island boundaries will be of Dirichlet type:

$$\rho = 0. \tag{2.2}$$

To track the large number of individual interfaces (island boundaries) that coalesce or are created by nucleation, a level set formulation has been used [9, 14]. In this formulation, the closed curves of the island boundary, $\Gamma = \cup_{i=1}^{m} \Gamma_i$, are represented as the zero level set of a function, $\phi(x,t)$, i.e.,

$$\Gamma = \{x : \phi(x,t) = 0\}. \tag{2.3}$$

The motion of the curves Γ are evolved by the level set equation

$$\frac{\partial \phi}{\partial t} + v_n |\nabla \phi| = 0, \tag{2.4}$$

where v_n is the normal velocity on the level set which can be given by a jump condition of ρ across island boundaries in the normal direction. To summarize, in each time step, perform the following computations:

1. Compute an approximation to the normal velocity, v_n.
2. Update ϕ by solving (2.4).
3. Solve the diffusion equation (2.1) for ρ, with the internal boundary condition (2.2). The location of the boundary is implicitly given by (2.3).

The diffusion equation can be discretized by a finite difference method [9, 19], which results in a symmetric linear system. In [9], PCG is used to solve the linear system, and an average of 50 PCG iterations is needed in each time step. Note that for the incomplete Cholesky factorization preconditioner, one needs extra storage for the incomplete factor. In this paper, the proposed multigrid solver typically requires much fewer iterations.

2.1. Boundary-capturing multigrid. Robust multigrid methods have been proposed for solving diffusion equations, for example, [2, 10, 16, 23, 25, 27, 29, 30, 32]; see also [8] for a recent survey. However, many of these methods focus primarily on the nonsmoothness of the PDE coefficient $a(x)$ in (1.1). In numerical experiments, typically only one single interface is considered and is usually aligned with the fine grid. Moreover, the Dirichlet boundary condition on the interface is seldom discussed. It is tempting to believe that Dirichlet boundary condition is easier to deal with since it is essentially just a Poisson problem defined on a domain with complex boundary shape, and classical multigrid should work well. However, since the Dirichlet boundary condition decouples the unknowns near the interface, if one ignores this fact and attempts to interpolate across the interface, the poor interpolation will slow down the multigrid convergence. We shall come back to this point in section 2.3.

The discretization of (1.1) by finite difference or finite element methods will lead to a linear system of the form

$$A^h u^h = f^h.$$

The multigrid algorithm consists of two basic steps: smoothing and coarse grid correction. For elliptic problems, relaxation methods—in particular, Gauss–Seidel—are

effective smoothers. We note that in each Gauss–Seidel iteration, the ith component of the solution is updated by using its neighboring values. Hence, the stencil information at the ith grid point is sufficient and the entire matrix A is not needed.

In the coarse grid correction step, the fine grid error is solved approximately on the coarse grid. The coarse grid matrix, A^H, is often obtained by the Galerkin process:

$$A^H = P^T A^h P,$$

where P is the interpolation operator. In general, A^H is not a submatrix of A^h, and hence one needs to store all the coarse grid matrices formed by the Galerkin coarsening. Employing the matrix-free multigrid approach, we obtain the coarse grid operators by direct discretization using an appropriate coarse grid representation of the level set function ϕ^H. More precisely,

$$\phi^H(x_i^H) = \phi^h(x_j^h),$$

where the fine grid point x_j^h coincides with the coarse grid point x_i^H. The construction of the interpolation operator, P, will be described in the following sections, and the restriction operator is defined as the transpose of P. V-cycle is to be used in the multigrid iteration.

2.2. One dimension. To obtain an efficient multigrid algorithm, we need to define the interpolation operator P^l appropriately, especially across the interfaces. When interpolating the coarse grid solution u^H, we do not want to interpolate solution values across the interface. The idea of treating regions where the problem behaves very differently has also been exploited in solving linear complementarity problems using multigrid methods [4, 21].

As shown in Figure 2.1, at the noncoarse grid point x_{2i-1}^h, if simple linear interpolation is used to approximate u_{2i-1}^h by

$$u_{2i-1}^h = \frac{1}{2}(u_{i-1}^H + u_i^H),$$

there will be a large approximation error to the fine grid solution since u_{i-1}^H essentially has nothing to do with the solution values on the right of Γ. The correct linear interpolation should use the location of Γ instead to approximate u_{2i-1}^h:

$$u_{2i-1}^h = \frac{x_{2i-1}^h - \Gamma}{x_{2i}^h - \Gamma} u_i^H. \tag{2.5}$$

(Note: $u^h(\Gamma) = 0$ due to the Dirichlet boundary condition.)

The location of Γ of the island boundary can be easily determined by the level set function $\phi(x)$. Recall that Γ is on the zero level set of ϕ, i.e., $\phi(\Gamma) = 0$. Interpolating the values of ϕ linearly, one obtains

$$\Gamma = x_{i-1}^H + \frac{|\phi(x_{i-1}^H)|}{|\phi(x_i^H)| + |\phi(x_{i-1}^H)|}.$$

We note that in multigrid literature, e.g., [15], a robust 1D matrix-dependent interpolation can be defined as

$$u_{2i-1}^h = -\frac{a u_{i-1}^H + b u_i^H}{c}, \tag{2.6}$$

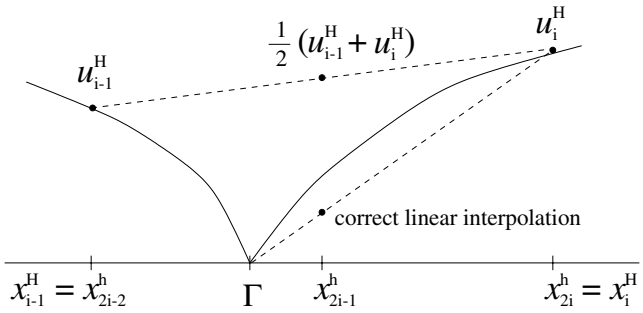

FIG. 2.1. *Linear interpolation for the Dirichlet-type problem. The solid line shows the coarse grid solution u^H. The two dashed lines show the linear interpolations applied at different points.*

where a, b, and c are the values of the three-point discretization stencil, $[a \;\; c \;\; b]$, at x^h_{2i-1}. This interpolation formula can be interpreted as solving a local PDE, energy minimization, preserving flux continuity, or Gaussian elimination; see a recent survey in [8]. If the null space of the underlying PDE consists of constant functions, then $c = -(a+b)$. In the case of constant coefficient PDE, i.e., $a = b$, then (2.6) will result in linear interpolation, which is well known to be optimal for multigrid. If constants are not in the null space, an alternative formula could be used,

$$u^h_{2i-1} = \frac{a u^H_{i-1} + b u^H_i}{a+b}, \tag{2.7}$$

so that linear interpolation can still be recovered in the constant coefficient case. Both black box multigrid and AMG use the interpolation formula of (2.6) in one dimension if standard coarsening were to be used.[1] Near the interface, since Γ decouples u^h_{2i-2} and u^h_{2i-1}, the discretization stencil at x^h_{2i-1} given by a finite element discretization would be $[0 \;\; 1/\theta + 1/h \;\; -1/h]$, where $\theta = x^h_{2i-1} - \Gamma$. The formulae of (2.6) and (2.7) would give the interpolation value at x^h_{2i-1} as

$$u^h_{2i-1} = \frac{\theta}{\theta + h} u^H_i \qquad \text{and} \qquad u^h_{2i-1} = u^H_i,$$

respectively. The interpolation given by (2.6) coincides with our geometric interpolation (2.5), whereas that by (2.7) results in piecewise constant interpolation, which violates the necessary order condition [6, 17] for multigrid. However, as we shall see in the next section, neither of these two formulae will give the correct interpolation in two dimensions.

2.3. Higher dimensions. The 1D geometric interpolation can be extended to two dimensions by applying the 1D technique to each dimension. Consider a generic 2D coarse grid cell as shown in Figure 2.2(a). The values at the coarse grid point u^H_i, $i = 1, 3, 7, 9$, are given by the coarse grid correction process. For noncoarse grid points lying on the coarse grid lines (grid points 2, 4, 6, 8), we apply the 1D interpolation. For instance,

$$u^h_2 = \frac{x^h_2 - \Gamma}{x^h_3 - \Gamma} u^H_3.$$

[1] In AMG algorithms, algebraic coarsenings are used instead of standard coarsening.

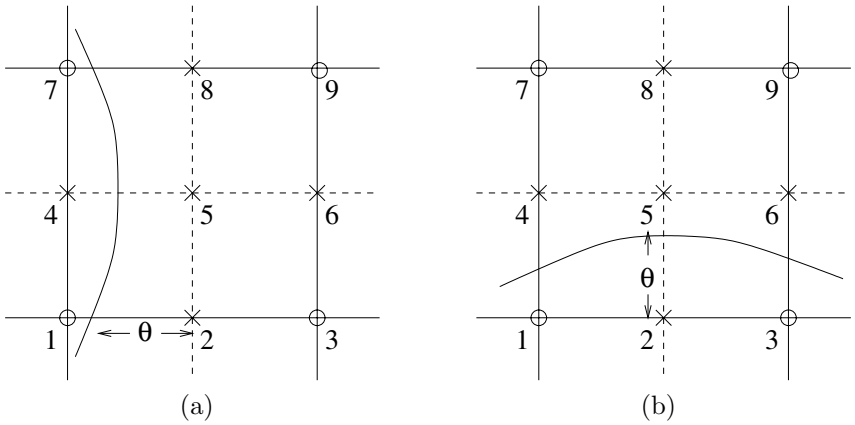

FIG. 2.2. *Boundary-capturing interpolation on a generic* 2D *coarse grid cell for the Dirichlet-type problem. Coarse and fine grid points are denoted by* 1, 3, 7, 9, *and* 2, 4, 5, 6, 8, *respectively. The interface is located* (a) *vertically towards the left,* (b) *horizontally towards the bottom.*

For the other noncoarse grid point (grid point 5), since the values of the neighboring grid points have been determined either by the coarse grid values or by the 1D interpolation, its value is obtained by solving a local PDE problem using the neighboring values as boundary values. Since this local problem consists of a single unknown (grid point 5) only, we need only the stencil information at this point. More precisely, suppose the discretization stencil at x_5^h is

$$\begin{bmatrix} a_{NW} & a_N & a_{NE} \\ a_W & a_C & a_E \\ a_{SW} & a_S & a_{SE} \end{bmatrix}.$$

Then the interpolation at x_5^h is given by

$$u_5^h = -\frac{a_{NW}u_7^H + a_N u_8^h + a_{NE}u_9^H + a_W u_4^h + a_E u_6^h + a_{SW}u_1^H + a_{SE}u_3^H}{a_C}. \tag{2.8}$$

It is in effect the same as applying one Gauss–Seidel sweep to the interior noncoarse grid points.

Black box vs. geometric interpolation. We remark that black box multigrid, unlike in one dimension, gives rise to a different 2D interpolation. Specifically, when interpolating the noncoarse grid points lying on coarse grid lines, for instance, grid point 2 in Figure 2.2(a), black box multigrid considers the five-point/nine-point stencil defined at that coarse grid point. A 1D three-point stencil is obtained by lumping the first and third rows of the stencil values to the center row. Then the robust interpolation is applied to this three-point stencil. For example, consider grid point 2 in Figure 2.2(a), (b) with two different locations of the interface Γ. Considering the fine grid, the five-point stencils are given by

$$\begin{bmatrix} 0 & -\frac{1}{h^2} & 0 \\ 0 & \frac{3}{h^2}+\frac{1}{\theta^2} & -\frac{1}{h^2} \\ 0 & -\frac{1}{h^2} & 0 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 0 & 0 & 0 \\ -\frac{1}{h^2} & \frac{3}{h^2}+\frac{1}{\theta^2} & -\frac{1}{h^2} \\ 0 & -\frac{1}{h^2} & 0 \end{bmatrix},$$

respectively, where θ is as indicated in the figure. (This finite difference discretization is used in [19].) Applying the lumping idea, the three-point stencils obtained are

$$[0,\ 1/h^2 + 1/\theta^2,\ -1/h^2] \qquad \text{and} \qquad [-1/h^2,\ 2/h^2 + 1/\theta^2,\ -1/h^2],$$

respectively. Using the robust interpolation formula (2.6), the resulting 1D interpolations at grid point 2 are then given by

$$u_2^h = \frac{\theta^2}{\theta^2 + h^2} u_3^H \qquad \text{and} \qquad u_2^h = \frac{\theta^2}{2\theta^2 + h^2} u_1^H + \frac{\theta^2}{2\theta^2 + h^2} u_3^H,$$

respectively. For the case as shown in Figure 2.2(a), where the interface is located between coarse grid points 1 and 3, it is reasonable to interpolate grid point 2 using the value only at grid point 3. However, we note that the interpolation weight is different from that of our geometric interpolation (2.5). For the case as shown in Figure 2.2(b), since the interface is not located between the coarse grid points 1 and 3, one would expect to use linear interpolation, which, unfortunately, is not the case here. Besides it seems no reason for the interpolation weights to be affected by θ, the vertical distance between grid point 2 and the interface. If the other 1D interpolation formula (2.7) is used instead, the corresponding interpolations at grid point 2 are, respectively,

$$u_2^h = u_3^h \qquad \text{and} \qquad u_2^h = \frac{1}{2} u_1^h + \frac{1}{2} u_3^h.$$

This time, the interpolation formula will give the correct linear interpolation for the case of Figure 2.2(b), but for the case of Figure 2.2(a) the result would be the first order piecewise constant interpolation, which violates the necessary order condition [6, 17] for multigrid. Hence, neither methods will lead to a robust 2D interpolation if the interface information is obtained purely algebraically via the discretization stencils. The numerical results in section 5 verify the slower convergence rate of black box multigrid.

Three dimensions. The extension of our interpolation operator to three dimensions is similar to that of two dimensions. Figure 2.3 shows a generic coarse grid cell; the corners of the cube represent eight coarse grid points. We first apply the 1D interpolation to the noncoarse grid points lying on the coarse grid lines. Second, for the noncoarse grid points lying on the six coarse grid planes, we solve a local 2D problem. However, the discretization stencil at those grid points is actually a 3D seven-point stencil. To obtain a 2D five-point stencil, we lump the stencil values above and below the plane on which the noncoarse grid point lies to the center stencil value. We note that the interface information has already been used in the interpolation of coarse grid edges, and we do not find that the lumping procedure causes any deterioration in convergence, contrary to the case of black box multigrid. Finally, the interpolation value at the noncoarse grid point right at the center of the cube is obtained by solving a local 3D problem as in two dimensions; see (2.8).

Remark. The restriction operator can be constructed in the same way as the interpolation operator; i.e., the restriction is applied to either side of the interface. It turns out that such a restriction will be different from the transpose of the interpolation constructed above. However, numerical results show that the resulting multigrid convergence is essentially the same as that using the transpose of the interpolation, and hence it is not discussed further here.

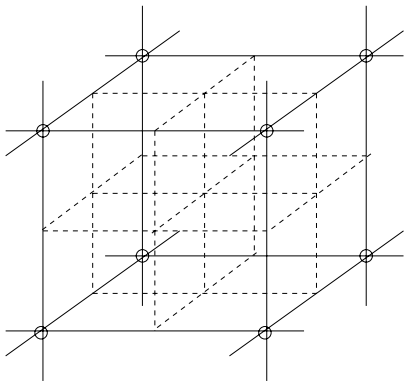

FIG. 2.3. *Boundary-capturing interpolation on a generic* 3D *coarse grid cell for the Dirichlet-type problem. The coarse grid points are located at the eight corners marked by circles and the rest are noncoarse grid points.*

2.4. Coarse representation of ϕ. On very coarse grids, the coarse level set functions fail to resolve the boundaries of small islands. Thus, the interpolation becomes less accurate on very coarse grids. One could use the level set function on the finest grid for all the other coarse grid calculations. However, it turns out that there is not much influence on the resulting multigrid convergence. Thus, we do not consider this issue here for the Dirichlet-type problems. We note, however, that the coarse representation of ϕ for the interface-type problems cannot be neglected.

3. Irregular boundary problem of interface type. This problem is often simply known as the interface problem. One example is the model of two-phase flow, such as simulations involving water [26], by the incompressible Navier–Stokes equations:

$$\begin{aligned} u_t + (u \cdot \nabla)u &= F + \frac{1}{\rho}(-\nabla p + \nabla \cdot (2\mu D) + \sigma\kappa\delta(d)n), \\ \nabla \cdot u &= 0, \end{aligned}$$

where u is the fluid velocity, ρ is the fluid density, μ is the fluid viscosity, D is the viscous stress tensor, and F is a body force. The surface tension is given by the last term where σ is the surface tension, κ is the curvature of the interface, d is the normal distance to the interface, δ is the Dirac delta function, and n is the unit outward normal. The pressure p satisfies the Poisson equation

$$-\nabla \cdot \frac{1}{\rho}\nabla p = f,$$

where the density ratio of water and air is 1 to 1000; i.e.,

$$\rho(x) = \begin{cases} 1000 & \text{if } x \text{ is in the region of water,} \\ 1 & \text{if } x \text{ is in the region of air.} \end{cases}$$

In the general case, jump conditions are given to specify the solution behavior near the interface, and they are usually written in the form

$$\begin{aligned} [u]_n &= \alpha, \\ [a\nabla u]_n &= \beta, \end{aligned} \tag{3.1}$$

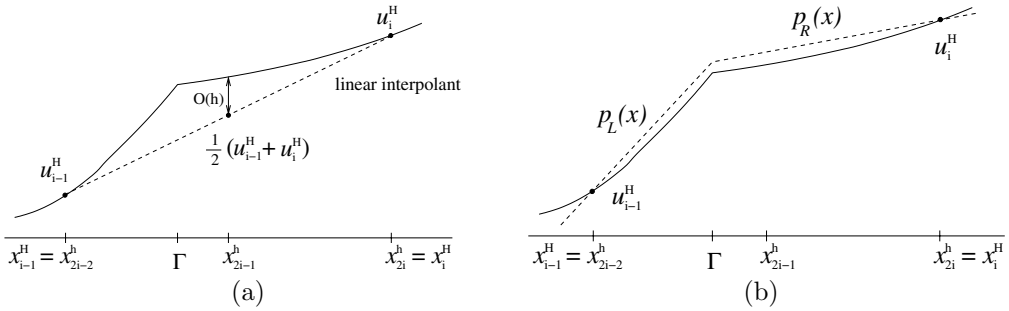

FIG. 3.1. *The solid line denotes the coarse grid solution u^H, and the dotted line the linear interpolant.* (a) *$O(h)$ approximation error results at the kink if linear interpolation is used.* (b) *Jump-preserving interpolation near interface with jump conditions*

where $[\cdot]_n$ denotes the jump in magnitude in the normal direction, and α and β are given functions. In contrast with the Dirichlet-type problem, in the numerical solution of interface-type problems, all the unknowns are coupled throughout the computational domain.

As in the previous section, we are interested in matrix-free multigrid preconditioning, and hence the coarse grid operators are obtained by direct discretization. The multigrid algorithm using V-cycle is also used here. However, the interpolation has to be modified to capture the appropriate jump conditions on the interface.

3.1. Jump-preserving interpolation. Since the model equation is linear, we can always consider the residual equation for the error. Without loss of generality, we assume $\alpha = \beta = 0$. Consequently, the solution is continuous except for its first derivative in the normal direction. Thus, the solution typically exhibits a kink at the interface location; see Figure 3.1(a). If linear interpolation is applied naively, an $O(h)$ error would result because of the kink, leading to poor coarse grid correction. Hence, we must incorporate the jump conditions in the construction of the interpolation operator to correctly capture the boundary effect.

We observe that away from the interface, the solution usually behaves smoothly and so linear interpolation will be sufficient; see Figure 3.1(b). Near the interface Γ, we let $p_L(x)$ be the linear interpolation polynomial on $[x_{i-1}^H, \Gamma]$, and we let $p_R(x)$ be the one on $[\Gamma, x_i^H]$, i.e.,

$$p_L(x) = u_{i-1}^H + s_L(x - x_{i-1}^H),$$
$$p_R(x) = u_i^H + s_R(x - x_i^H),$$

where s_L and s_R are the slopes of $p_L(x)$ and $p_R(x)$, respectively. Applying the jump conditions, we have

$$p_L(\Gamma) = p_R(\Gamma),$$
$$a_L s_L = a_R s_R,$$

where $a_L \equiv a(x_{i-1}^H)$ and $a_R \equiv a(x_i^H)$ are the diffusion coefficients on either side of Γ, respectively. With two equations and two unknowns, we can uniquely solve for s_L and s_R. Then the interpolated value at the noncoarse grid point x_{2i-1}^h is given by

$$u_{2i-1}^h = p_R(x_{2i-1}^h)$$

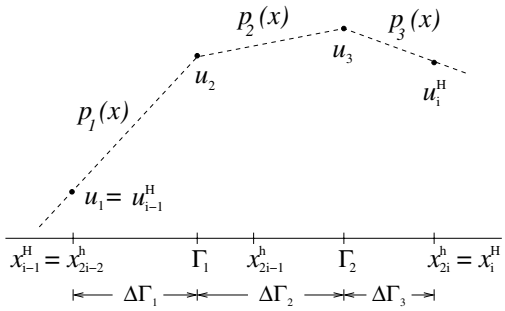

FIG. 3.2. *Jump-preserving interpolation near interface with multiple interfaces.*

if $x_{2i-1}^h > \Gamma$, and by

$$u_{2i-1}^h = p_L(x_{2i-1}^h)$$

if $x_{2i-1}^h < \Gamma$. Note that only the level set function defining the location of Γ and the diffusion is needed.

We remark that the robust 1D interpolation described in the previous section coincides with our geometric interpolation if the interface aligns with the fine grid.

The extension to higher dimensions is similar to the case of Dirichlet-type problems; we apply the 1D interpolation dimension-by-dimension, and hence it is not repeated here.

3.2. Multiple interfaces. On finer grids, there will be at most one interface point between two consecutive coarse grid points lying on the same coarse grid line. However, on coarser grids, it is possible to have more than one interface point, as shown in Figure 3.2. In the construction of the interpolation, we need to incorporate the jump conditions across all the interfaces between two consecutive coarse grid points.

We describe the construction of our interpolation for two interfaces; other cases are similar. Between two interfaces, we use linear interpolation as before. Thus, we have three linear interpolation polynomials with slopes s_1, s_2, and s_3, i.e.,

$$\begin{aligned} p_1(x) &= u_1 + s_1(x - x_{2i-2}^h), \\ p_2(x) &= u_2 + s_2(x - \Gamma_1), \\ p_3(x) &= u_3 + s_3(x - \Gamma_2), \end{aligned}$$

where $u_1 = u_{i-1}^H$ and u_2, u_3 are to be determined. For easy exposition, we assume the PDE coefficient is piecewise constant; i.e., $a(x)$=a_1, a_2, and a_3 on $[x_{2i-2}^h, \Gamma_1]$, $[\Gamma_1, \Gamma_2]$, and $[\Gamma_2, x_{2i}^h]$, respectively. Now, considering the jump conditions across the interfaces,

$$\begin{aligned} a_1 s_1 &= a_2 s_2, \\ a_2 s_2 &= a_3 s_3, \end{aligned}$$

we have $s_2 = (a_1/a_2)s_1$ and $s_3 = (a_1/a_3)s_1$. By the continuity conditions at the interfaces, we have

$$\begin{aligned} u_2 &= u_{i-1}^H + s_1 \Delta\Gamma_1, \\ u_3 &= u_2 + s_2 \Delta\Gamma_2, \\ u_i^H &= u_3 + s_3 \Delta\Gamma_3, \end{aligned}$$

where $\Delta\Gamma_1 = \Gamma_1 - x^h_{2i-2}$, $\Delta\Gamma_2 = \Gamma_2 - \Gamma_1$, and $\Delta\Gamma_3 = x^h_{2i} - \Gamma_2$. Summing up the equations and substituting s_2 and s_3 in terms of s_1, we obtain an equation for s_1:

$$u^H_i = u^H_{i-1} + \sum_{j=1}^{3} \frac{a_1 s_1}{a_j} \Delta\Gamma_j,$$

$$\Rightarrow \qquad s_1 = \frac{u^H_i - u^H_{i-1}}{a_1 \sum_{j=1}^{3} \frac{\Delta\Gamma_j}{a_j}}.$$

We can solve for the value of s_1 and hence s_2 and s_3. The interpolated value can then be given by one of the linear polynomials, depending on the location of x^h_{2i-1}.

4. Analysis. It is well known in the multigrid literature that fast multigrid convergence relies on accurate coarse grid correction, which in turn requires accurate interpolation between coarse and fine grids; see, e.g., [15, 31]. In this section, we present an analysis to show that our boundary-capturing and interface-preserving interpolations are more accurate than the naive linear interpolation, which results in first order accuracy only. More precisely, the proposed interpolations are both second order accurate.

4.1. Irregular boundary problem of Dirichlet type. We first consider the 1D case. If the Dirichlet boundary condition at the interface is ignored and standard linear interpolation is applied to two consecutive coarse grid points, as shown in Figure 2.1, the error can be very large since u^H_{i-1} would not provide any useful information in approximating u^h_{2i-1}. Specifically, we have the following result.

LEMMA 4.1. *Assume the interface Γ is located between x^h_{2i-2} and x^h_{2i-1}, as shown in Figure* 2.1. *Let $\tilde{u}^h_{2i-1}$ and u^h_{2i-1} be the approximate solution given by linear interpolation and the fine grid solution at the noncoarse grid point x^h_{2i-1}, respectively. Then*

$$\tilde{u}^h_{2i-1} - u^h_{2i-1} = \frac{u^+_{\Gamma,x} - u^-_{\Gamma,x}}{2} \nu h + O(h^2),$$

where $\nu h = \Gamma - x^h_{2i-2}$, and $u^+_{\Gamma,x}$ and $u^-_{\Gamma,x}$ are the first derivatives of the fine grid solution u^h at Γ from the right and from the left, respectively.

Proof. Taylor expanding u^H_{i-1}, u^H_i, and u^h_{2i-1} at Γ, we have

$$\begin{aligned} u^H_{i-1} &= u_\Gamma - \nu h u^-_{\Gamma,x} + O(h^2), \\ u^H_i &= u_\Gamma + (2-\nu) h u^+_{\Gamma,x} + O(h^2), \\ u^H_{2i-1} &= u_\Gamma + (1-\nu) h u^+_{\Gamma,x} + O(h^2), \end{aligned}$$

where u_Γ is the value of u^h at Γ. (Note: With the Dirichlet boundary condition, $u_\Gamma = 0$. But the result turns out to be independent of the value of u_Γ.) Then the error of linear interpolation is

$$\begin{aligned} \frac{1}{2}(u^H_{i-1} + u^H_i) - u^h_{2i-1} &= u_\Gamma - \frac{\nu}{2} h u^-_{\Gamma,x} + \frac{2-\nu}{2} h u^+_{\Gamma,x} - u_\Gamma - (1-\nu) h u^+_{\Gamma,x} + O(h^2) \\ &= \frac{u^+_{\Gamma,x} - u^-_{\Gamma,x}}{2} \nu h + O(h^2). \qquad \square \end{aligned}$$

We remark that only first order accuracy results, even if linear interpolation is used. Moreover, the constant of the first order term can be arbitrarily large, depending on the values of the first derivatives of u^h on both sides of Γ.

For the boundary-capturing interpolation, since we identify the location of Γ and then apply linear interpolation accordingly, we readily have the second order accuracy result by the standard Taylor expansion analysis.

LEMMA 4.2. *Let $\tilde{u}^h_{2i-1}$ be the approximate solution given by the boundary-capturing interpolation* (2.5). *Then*

$$\tilde{u}^h_{2i-1} - u^h_{2i-1} = O(h^2).$$

The boundary-capturing interpolation in higher dimensions is extended from one dimension. Thus, the analysis for higher-dimensional interpolations is essentially the same as that for the 1D case, and hence they are omitted here.

4.2. Irregular boundary problem of interface type. The analysis for interface problems is not as straightforward as it is for Dirichlet problems since we do not know the value of u^h at the interface, and we also have to take into account the jump conditions.

Consider Figure 3.1 in one dimension. We first show that linear interpolation is only first order accurate.

LEMMA 4.3. *Assume the PDE coefficient $a(x)$ in* (1.1) *is piecewise constant,*

$$a(x) = \begin{cases} a_L & \text{if } x < \Gamma, \\ a_R & \text{if } x > \Gamma. \end{cases}$$

Let $\tilde{u}^h_{2i-1}$ be the approximate solution given by linear interpolation. Then

$$\tilde{u}^h_{2i-1} - u^h_{2i-1} = \frac{\nu h}{2}\left(1 - \frac{a_R}{a_L}\right) u^+_{\Gamma,x} + O(h^2),$$

where $\nu h = \Gamma - x^h_{2i-2}$.

Proof. By a similar calculation as in Lemma 4.1, we have

$$\frac{1}{2}(u^H_{i-1} + u^H_i) - u^h_{2i-1} = \frac{\nu h}{2}(u^+_{\Gamma,x} - u^-_{\Gamma,x}) + O(h^2).$$

Using the jump condition

$$a_L u^-_{\Gamma,x} = a_R u^+_{\Gamma,x}, \tag{4.1}$$

the error formula becomes

$$\frac{1}{2}(u^H_{i-1} + u^H_i) - u^h_{2i-1} = \frac{\nu h}{2}\left(1 - \frac{a_R}{a_L}\right) u^+_{\Gamma,x} + O(h^2). \qquad \square$$

Hence, if $a_R/a_L \neq 1$, only first order accuracy results. Also, the error becomes bigger for a large jump in the coefficient.

The large error that occurred in linear interpolation is essentially due to not using the jump conditions. In the jump-preserving interpolation, we try to preserve the jump conditions, and as a result, second order accuracy can be restored.

LEMMA 4.4. *Let $\tilde{u}^h_{2i-1}$ be the approximate solution given by the jump-preserving interpolation. Then*

$$\tilde{u}^h_{2i-1} - u^h_{2i-1} = O(h^2).$$

Proof. By the construction of the jump-preserving interpolation described in section 3.1, it is straightforward to compute the formula for $\tilde{u}^h_{2i-1}$:

$$\tilde{u}^h_{2i-1} = u^H_i - \frac{a_L}{\mu a_L + \nu a_R}(u^H_i - u^H_{i-1}),$$

where $\mu = 2 - \nu$. Applying Taylor expansion on $u^H_i - u^H_{i-1}$ and using the jump condition (4.1), we obtain

$$\begin{aligned}\tilde{u}^h_{2i-1} &= u^H_i - \frac{a_L}{\mu a_L + \nu a_R}(\mu h u^+_{\Gamma,x} + \nu h u^-_{\Gamma,x}) + O(h^2)\\ &= u^H_i - \frac{1}{\mu a_L + \nu a_R}(\mu h a_L u^+_{\Gamma,x} + \nu h a_R u^+_{\Gamma,x}) + O(h^2)\\ &= u^H_i - h u^+_{\Gamma,x} + O(h^2).\end{aligned}$$

Now, using the Taylor expansions of u^H_i and u^h_{2i-1},

$$\begin{aligned}\tilde{u}^h_{2i-1} &= u_\Gamma + \mu h u^+_{\Gamma,x} - h u^+_{\Gamma,x} + O(h^2)\\ &= u^h_{2i-1} - (\mu - 1) h u^+_{\Gamma,x} + \mu h u^+_{\Gamma,x} - h u^+_{\Gamma,x} + O(h^2)\\ &= u^h_{2i-1} + O(h^2). \qquad \square\end{aligned}$$

The analysis for two and three dimensions is similar. So, we just present the 2D analysis for easy exposition. Consider the generic coarse grid cell, as shown in Figure 2.2. Since the analysis for the two interface locations (and other possible cases) is almost identical, we focus on the one on the left where the interface goes between grid points 1 and 2, 4 and 5, and 7 and 8.

THEOREM 4.5. *Assume the PDE coefficient $a(x)$ in (1.1) is piecewise constant,*

$$a(x) = \begin{cases} a_L & \text{if } x \text{ is on the left of } \Gamma,\\ a_R & \text{if } x \text{ is on the right of } \Gamma.\end{cases}$$

Let $\tilde{u}^h_j$ and u^h_j, $j = 2, 4, 5, 6, 8$ (cf. Figure 2.2), be the approximate solution given by jump-preserving interpolation and the fine grid solution values, respectively. Then

$$\tilde{u}^h_j - u^h_j = O(h^2).$$

Proof. Since the construction of the 2D jump-preserving interpolation is derived from the 1D interpolation, it is straightforward to show that the errors of approximation at the noncoarse grid points 2, 4, 6, and 8 are second order (cf. Lemma 4.4). Now, consider the interior noncoarse grid point 5. Using the ghost fluid method [19], the discretization stencil at grid point 5 is given by

$$\begin{bmatrix} 0 & -\frac{a_R}{h^2} & 0\\ -\frac{a}{h^2} & \frac{3a_R}{h^2} + \frac{a}{h^2} & -\frac{a_R}{h^2}\\ 0 & -\frac{a_R}{h^2} & 0\end{bmatrix},$$

where a is the harmonic average:

$$a = \frac{a_L a_R}{a_R \nu + a_L(1 - \nu)}. \tag{4.2}$$

Hence, the jump-preserving interpolation at grid point 5 (cf. (2.8)) leads to the approximation formula

$$\begin{aligned}\tilde{u}_5^h &= \frac{1}{\frac{3a_R}{h^2}+\frac{a}{h^2}}\left(\frac{a_R}{h^2}\tilde{u}_8^h+\frac{a_R}{h^2}\tilde{u}_6^h+\frac{a_R}{h^2}\tilde{u}_2^h+\frac{a}{h^2}\tilde{u}_4^h\right)\\ &= \frac{1}{3a_R+a}(a_R\tilde{u}_8^h+a_R\tilde{u}_6^h+a_R\tilde{u}_2^h+a\tilde{u}_4^h).\end{aligned}$$

The error of the approximation is then given by

$$\begin{aligned}\tilde{u}_5^h-u_5^h &= \frac{1}{3a_R+a}(a_R\tilde{u}_8^h+a_R\tilde{u}_6^h+a_R\tilde{u}_2^h+a\tilde{u}_4^h-3a_Ru_5^h-au_5^h)\\ &= \frac{1}{3a_R+a}[a_R(\tilde{u}_8^h-u_5^h)+a_R(\tilde{u}_6^h-u_5^h)+a_R(\tilde{u}_2^h-u_5^h)\\ &\qquad +a(\tilde{u}_4^h-u_5^h)].\end{aligned}\tag{4.3}$$

Note that $\tilde{u}_8^h-u_5^h$ can be written as

$$\begin{aligned}\tilde{u}_8^h-u_5^h &= (\tilde{u}_8^h-u_8^h)+(u_8^h-u_\Gamma)-(u_5^h-u_\Gamma)\\ &= (u_8^h-u_\Gamma)-(u_5^h-u_\Gamma)+O(h^2),\end{aligned}\tag{4.4}$$

since the jump-preserving interpolation of $\tilde{u}_8^h$ is second order accurate. Similar expressions hold for $\tilde{u}_6^h-u_5^h$, etc. Now, consider the Taylor expansions of u_8^h, u_2^h, u_6^h, u_4^h, and u_5^h at u_Γ, where u_Γ is the value of the fine grid solution u^h at the intersection of the interface and the grid line between grid point 4 and grid point 5:

$$\begin{aligned}u_8^h &= u_\Gamma+(1-\nu)hu_{\Gamma,x}^+ + hu_{\Gamma,y}+O(h^2),\\ u_2^h &= u_\Gamma+(1-\nu)hu_{\Gamma,x}^+ - hu_{\Gamma,y}+O(h^2),\\ u_6^h &= u_\Gamma+(2-\nu)hu_{\Gamma,x}^+ +O(h^2),\\ u_4^h &= u_\Gamma-\nu hu_{\Gamma,x}^- +O(h^2),\\ u_5^h &= u_\Gamma+(1-\nu)hu_{\Gamma,x}^+ +O(h^2),\end{aligned}\tag{4.5}$$

where νh = distance between grid point 4 and the interface. Substituting (4.4) and (4.5) into (4.3) and using (4.2), we have

$$\begin{aligned}&\tilde{u}_5^h-u_5^h\\ &= \frac{1}{3a_R+a}[a_R((1-\nu)hu_{\Gamma,x}^+ + hu_{\Gamma,y})+a_R(2-\nu)hu_{\Gamma,x}^+ + a_R((1-\nu)hu_{\Gamma,x}^+ - hu_{\Gamma,y})\\ &\qquad +a(-\nu hu_{\Gamma,x}^-)-(3a_R+a)((1-\nu)hu_{\Gamma,x}^+)]+O(h^2)\\ &= \frac{1}{3a_R+a}[a_Rhu_{\Gamma,x}^+ - a(\nu hu_{\Gamma,x}^- +(1-\nu)hu_{\Gamma,x}^+)]+O(h^2)\\ &= \frac{1}{3a_R+a}\left[a_Rhu_{\Gamma,x}^+ - \frac{\nu ha_Ra_Lu_{\Gamma,x}^-}{a_r\nu+a_L(1-\nu)}-\frac{(1-\nu)ha_La_Ru_{\Gamma,x}^+}{a_r\nu+a_L(1-\nu)}\right]+O(h^2).\end{aligned}$$

With the ghost fluid discretization [19, p. 163], the fine grid solution satisfies the following jump condition at the interface:

$$a_Lu_{\Gamma,x}^- = a_Ru_{\Gamma,x}^+.$$

As a result, the approximation error of $\tilde{u}_5^h$ becomes

$$\tilde{u}_5^h - u_5^h = \frac{1}{3a_R + a}\left[a_R h u_{\Gamma,x}^+ - \frac{\nu h a_R^2 u_{\Gamma,x}^+}{a_r\nu + a_L(1-\nu)} - \frac{(1-\nu)h a_L a_R u_{\Gamma,x}^+}{a_r\nu + a_L(1-\nu)}\right] + O(h^2)$$
$$= O(h^2). \qquad \square$$

5. Numerical results. In the following, we compare the convergence results of our geometric multigrid methods with PCG, AMG, and black box multigrid[2] for different irregular boundary problems. We remark that the purpose of the comparison is to demonstrate the effectiveness of the proposed method. Both AMG and black box multigrid are designed for problems in which geometric multigrid approach does not apply, for instance, problems with complex geometries and lack of geometric information. For all the multigrid methods, two pre- and two post-Gauss–Seidel smoothings are used. The multigrid iterations are stopped when the relative residual norm is less than 10^{-6}. The multigrid V-cycle is used.

We note that the AMG coarsening in general is different from the standard coarsening. Thus the number of coarse grid points as well as the number of nonzero entries in the coarse grid matrices can be quite different from the black box and our multigrid methods. To fairly compare the three methods, we report the work units used by the multigrid methods in place of the number of multigrid cycles. Here, one work unit is equal to the total amount of floating point operations involved in one V-cycle with standard coarsening and coarse grid matrices obtained from direct discretization. In other words, one V-cycle of the geometric multigrid is one work unit. The work unit of PCG is normalized by equating one Gauss–Seidel smoothing to one matrix vector multiplication. The work for one incomplete Cholesky preconditioning step is also considered as one matrix vector multiplication. The work unit of AMG is obtained from comparing the total number of nonzeros in all the coarse grid matrices with that from the geometric multigrid.

Example 1. We first compare the iterative methods by a 2D irregular boundary problem of Dirichlet type. The PDE coefficient $a(x) = 1$, and the source term is $f(x) = 1$. The interfaces are circles and hence do not align with any coarse grids (as well as the finest grid). The interfaces and sample solutions are shown in Figure 5.1.

Table 5.1 shows the convergence results of different methods. As noted in section 1, the convergence rate of PCG deteriorates with decreasing mesh size; the number of work units indeed doubles as h is reduced by half. Among the multigrid methods, black box multigrid (BMG) requires more work units to obtain convergence since the interpolation for the noncoarse grid points lying on coarse grid lines does not correctly capture the Dirichlet boundary condition, as explained in section 2.3. Our geometric multigrid (GMG) and AMG show about the same efficiency. However, we note that our geometric multigrid does not require the storage of the coarse grid matrices, whereas AMG has to store all the coarse grid matrices. We shall address the memory issue in the next example.

For all the multigrid methods, the convergence deteriorates slightly when the mesh size h decreases, but it improves when the number of interfaces increases. This is due to the Dirichlet boundary condition; each interface defines an independent problem, and hence Gauss–Seidel smoothing becomes more like a direct solver.

Example 2. We next compare the multigrid methods to a 3D irregular boundary problem of Dirichlet type. The interfaces are arrays of spheres; see Figure 5.2. The

[2]The AMG and black box multigrid codes are obtained from MGNet (www.mgnet.org).

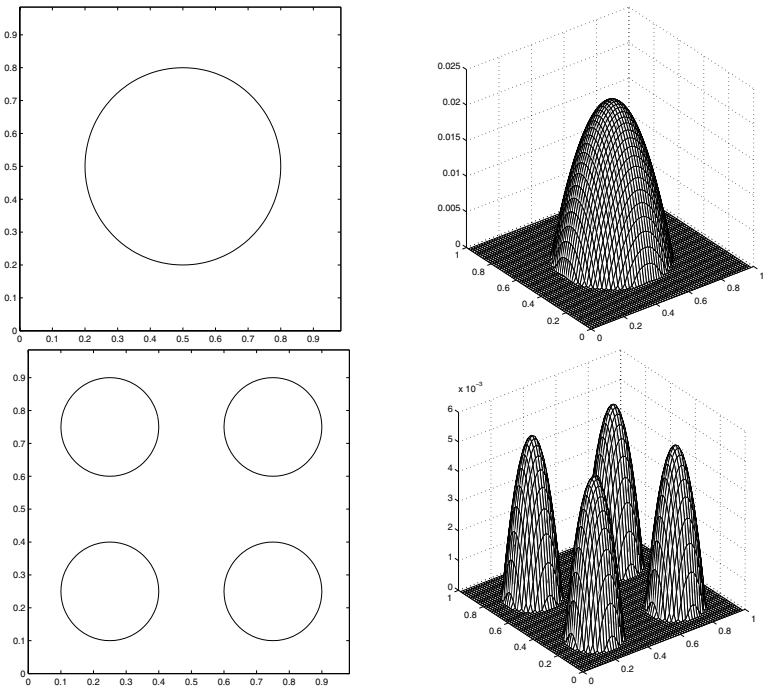

FIG. 5.1. *Circle interfaces (left) and their corresponding solutions (right) to the irregular boundary problem of Dirichlet type.*

TABLE 5.1

Comparison by work units of our geometric multigrid method with other methods for a 2D *problem of Dirichlet type. The interfaces are different arrays of circles. The numbers in parentheses are iteration counts.*

h	Circle array	PCG	GMG	AMG	BMG
1/16	1×1	4 (14)	7 (7)	7 (5)	11 (9)
1/32	1×1	8 (22)	8 (8)	8 (5)	15 (13)
	2×2	7 (18)	7 (7)	8 (5)	14 (12)
1/64	1×1	15 (41)	10 (10)	10 (6)	18 (15)
	2×2	13 (33)	8 (8)	8 (5)	19 (16)
	4×4	8 (20)	7 (7)	8 (5)	15 (13)
1/128	1×1	31 (81)	11 (11)	10 (6)	20 (17)
	2×2	24 (63)	10 (10)	10 (6)	20 (17)
	4×4	14 (37)	8 (8)	8 (5)	20 (17)
	6×6	10 (26)	8 (8)	8 (5)	18 (15)

convergence results are shown in Table 5.2. For 3D problems, our geometric multigrid is much more efficient than PCG and AMG. The convergence results for black box multigrid are not available since the code at MGNet is for two dimensions only. PCG shows a similar convergence behavior to that in two dimensions.

It is interesting to note that the iteration counts for AMG are often less than 10, and yet the work units are usually more than three times higher. This is because the number of nonzero entries in the coarse grid matrices is much greater than those obtained from standard coarsening, especially in three dimensions. As a result, more work is needed to perform Gauss–Seidel smoothing on each coarse grid. To illustrate the point, we report typical numbers of nonzeros in the coarse grid matrices obtained

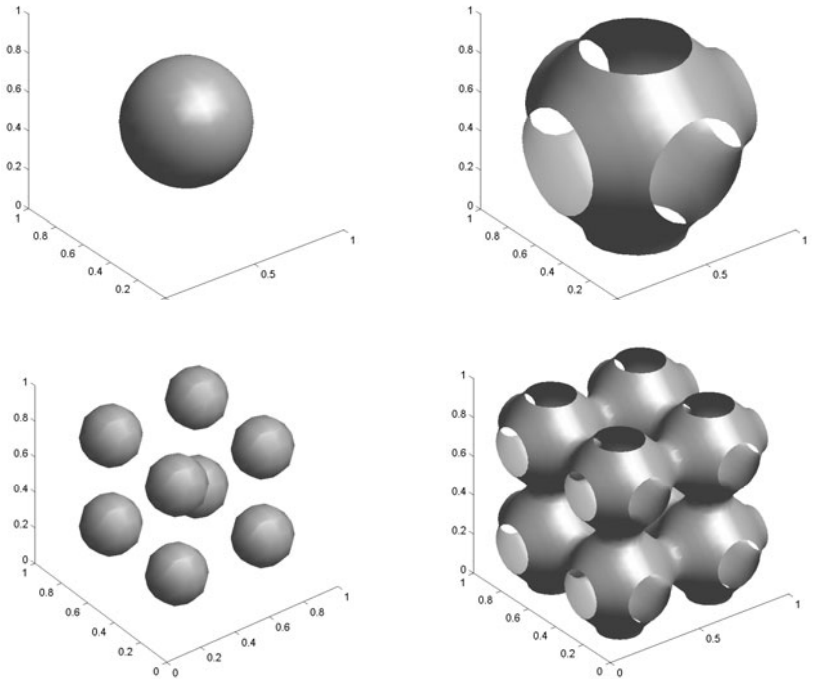

FIG. 5.2. *Sphere interfaces (left) and their corresponding solutions (right, a contour plot) to the irregular problem of Dirichlet type.*

TABLE 5.2

Comparison by work units of our geometric multigrid method with other methods for a 3D *problem of Dirichlet type. The interfaces are different arrays of spheres. Numbers in parentheses are iteration counts.*

h	Sphere array	PCG	GMG	AMG
1/8	$1 \times 1 \times 1$	4 (10)	7 (7)	9 (5)
1/16	$1 \times 1 \times 1$	8 (19)	7 (7)	14 (5)
	$2 \times 2 \times 2$	6 (14)	9 (9)	16 (5)
1/32	$1 \times 1 \times 1$	16 (36)	8 (8)	23 (7)
	$2 \times 2 \times 2$	11 (26)	8 (8)	17 (5)
	$4 \times 4 \times 4$	8 (17)	9 (9)	18 (5)
1/64	$1 \times 1 \times 1$	> 100	10 (10)	40 (12)
	$2 \times 2 \times 2$	93 (209)	8 (8)	23 (7)
	$4 \times 4 \times 4$	30 (67)	8 (8)	15 (5)

from AMG coarsening[3] and standard coarsening as shown in Table 5.3. For our geometric multigrid, a reduction of a factor of 8 results since direct discretization is used. For AMG, the reduction factor is less than 2; in fact, the number of nonzeros even increases when it goes from the finest to the first coarse grids. Moreover, the total number of nonzeros increases with the number of interfaces. Apart from taking more flops, AMG also needs to store all the coarse grid matrices, which amounts to almost four times that of the fine grid matrix. Note that our geometric multigrid

[3]In [28, p. 487], an operator complexity is used which is a ratio of the total number of nonzeros in all the coarse and fine grid matrices to the number of nonzeros in the fine grid matrix.

TABLE 5.3
*Number of nonzero entries (thousands) in the coarse grid matrices on different coarse grids (*1 *to* 4 *for the geometric multigrid method and* 1 *to* 6 *for AMG),* $h = 1/32$.

	GMG				AMG					
Sphere array	1	2	3	4	1	2	3	4	5	6
$1 \times 1 \times 1$	226	28	3	0.4	226	303	118	85	70	34
$2 \times 2 \times 2$	223	27	3	0.4	223	295	126	89	75	39
$3 \times 3 \times 3$	219	26	3	0.4	219	286	133	102	85	47
$4 \times 4 \times 4$	213	25	3	0.4	213	284	134	121	87	46

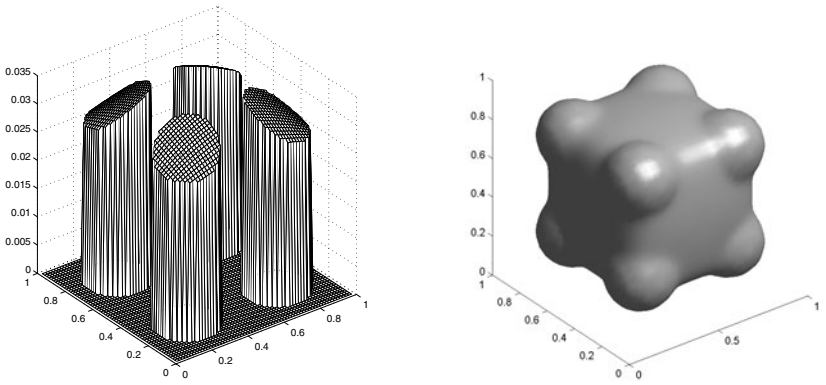

FIG. 5.3. *Solutions to the irregular boundary problem of interface type in (left) two dimensions and (right) three dimensions (a contour plot).*

method does not store any of the coarse grid matrices. We remark that an aggressive coarsening strategy [28, Appendix A] for AMG has recently been proposed to address the memory issue.

Example 3. We now compare the multigrid methods to a 2D irregular problem of interface type. A similar example is also used in [1] (with only one circle interface). Here, the interfaces are arrays of circles, and the jump conditions are as in (3.1), where $\alpha = \beta = 0$, and

$$a(x) = \begin{cases} a^- & \text{if } \phi(x) < 0, \\ a^+ & \text{if } \phi(x) > 0. \end{cases}$$

In the numerical experiments, we fix $a^- = 1$ and vary a^+ from 1 to 10^6. A solution where there are 4 circle interfaces are shown in Figure 5.3. The convergence results are given in Table 5.4. The first row indicates the values of a^+ chosen. This time, black box multigrid is more efficient than Dirichlet-type problems since no unknowns are decoupled. Its convergence is comparable to our geometric multigrid and insensitive to jump size but deteriorates when the mesh size decreases or when the number of interfaces increases. AMG requires more work than the other two methods, which is again due to denser coarse grid matrices. Also, the convergence of AMG deteriorates with the number of interfaces. PCG requires a lot more work units for interface problems, and convergence is slowed down by the mesh size, the size of the jump, and the number of interfaces.

Example 4. Finally, we compare the multigrid methods to a 3D irregular boundary problem of interface type. The interfaces are arrays of spheres, and the jump

TABLE 5.4
Comparison by work units of our geometric multigrid method with other methods for a 2D *problem of interface type. The interfaces are different arrays of circles.* $a^+ = 1, 10^3, 10^6$.

	Circle	PCG			GMG			AMG			BMG		
h	array	1	10^3	10^6	1	10^3	10^6	1	10^3	10^6	1	10^3	10^6
1/16	1×1	6	7	9	5	6	7	7	11	9	4	6	6
	2×2	6	10	13	5	9	9	7	9	9	4	7	7
1/32	1×1	10	12	14	5	6	7	7	9	8	5	6	6
	2×2	10	16	21	5	6	6	7	14	14	5	7	7
1/64	1×1	16	22	27	5	6	7	7	9	9	5	7	6
	2×2	16	30	41	5	7	7	7	13	13	5	9	9
1/128	1×1	33	44	52	5	6	8	7	9	10	5	10	7
	2×2	33	56	80	5	6	8	7	12	12	5	11	12

TABLE 5.5
Comparison by work units of our geometric multigrid method with other methods for a 3D *problem of interface type. The interfaces are different arrays of spheres.* $a^+ = 10^2, 10^4, 10^6$.

	Sphere	PCG			GMG			AMG		
h	array	10^2	10^4	10^6	10^2	10^4	10^6	10^2	10^4	10^6
1/8	$1 \times 1 \times 1$	5	6	7	8	8	8	12	16	17
	$2 \times 2 \times 2$	7	9	12	6	6	6	12	15	15
1/16	$1 \times 1 \times 1$	9	10	12	9	10	10	18	21	21
	$2 \times 2 \times 2$	12	17	22	8	8	8	40	32	32
1/32	$1 \times 1 \times 1$	16	20	24	10	11	11	20	24	>100
	$2 \times 2 \times 2$	24	32	42	9	9	9	77	42	42
1/64	$1 \times 1 \times 1$	30	38	42	8	10	11	19	29	26
	$2 \times 2 \times 2$	46	67	84	10	12	12	31	45	> 100
	$4 \times 4 \times 4$	58	89	98	9	11	11	> 100	> 100	> 100

conditions are as in Example 3. A solution where there are 4 sphere interfaces is shown in Figure 5.3. Table 5.5 shows the convergence results of PCG, the geometric multigrid method, and AMG. As in Example 3, our geometric multigrid is relatively insensitive to the mesh size and the size of the jump. The convergence of AMG and PCG, on the other hand, deteriorates with the number of interfaces and decreasing mesh size.

6. Conclusions. We have proposed an efficient geometric multigrid method for solving irregular boundary problems in two and three dimensions. The (fine and) coarse grid matrices are obtained from direct discretization, and hence no extra memory storage is needed, as opposed to the Galerkin approach used by AMG and black box multigrid. It is assumed that the interface location is captured by a level set function. As such, we have defined a second order accurate interpolation by capturing the appropriate boundary conditions at the interfaces. Theoretical analysis has been presented to justify the accuracy of our interpolation. Numerical results for problems with multiple interfaces in multiple dimensions have been given to compare our multigrid method with AMG and black box multigrid, which have been shown in the literature to be efficient multigrid solvers for interface problems with a single interface. We have shown that even without extra storage for coarse grid matrices, the convergence of our multigrid method is insensitive to the mesh size, the size of

the jump, and the number of interfaces, whereas the convergence of the other two multigrid methods shows a poorer convergence in some cases.

REFERENCES

[1] L. Adams and Z. Li, *The immersed interface/multigrid methods for interface problems*, SIAM J. Sci. Comput., 24 (2002), pp. 463–479.

[2] R. E. Alcouffe, A. Brandt, J. E. Dendy, Jr., and J. W. Painter, *The multigrid method for the diffusion equation with strongly discontinuous coefficients*, SIAM J. Sci. Statist. Comput., 2 (1981), pp. 430–454.

[3] C. R. Anderson, *Capacitance Matrix Methods for the Solution of Elliptic Equations Defined by Level Set Functions*, Technical Report CAM 00-15, Department of Mathematics, UCLA, 2000.

[4] A. Brandt and C. W. Cryer, *Multigrid algorithms for the solution of linear complementarity problems arising from free boundary problems*, SIAM J. Sci. Statist. Comput., 4 (1983), pp. 655–684.

[5] A. Brandt, *Multi-level adaptive solutions to boundary-value problems*, Math. Comp., 31 (1977), pp. 333–390.

[6] A. Brandt, *Guide to multigrid development*, in Multigrid Methods, W. Hackbusch and U. Trottenberg, eds., Lectures Notes in Math. 960, Springer-Verlag, Berlin, 1982, pp. 220–312.

[7] R. E. Caflisch, M. F. Gyure, B. Merriman, S. Osher, C. Ratsch, D. D. Vvedensky, and J. J. Zinck, *Island dynamics and the level set method for epitaxial growth*, Appl. Math. Lett., 12 (1999), pp. 13–22.

[8] T. F. Chan and W. L. Wan, *Robust multigrid methods for elliptic linear systems*, J. Comput. Appl. Math., 123 (2000), pp. 323–352.

[9] S. Chen, B. Merriman, M. Kang, R. E. Caflisch, C. Ratsch, L.-T. Cheng, M. Gyure, R. Fedkiw, C. Anderson, and S. Osher, *A level set method for thin film epitaxial growth*, J. Comput. Phys., 167 (2001), pp. 475–500.

[10] J. E. Dendy, Jr., *Black box multigrid*, J. Comput. Phys., 48 (1982), pp. 366–386.

[11] J. E. Dendy, Jr., *Black box multigrid for nonsymmetric problems*, Appl. Math. Comput., 13 (1983), pp. 261–283.

[12] J. Frank and C. Vuik, *On the construction of deflation-based preconditioners*, SIAM J. Sci. Comput., 23 (2001), pp. 442–462.

[13] I. G. Graham and M. J. Hagger, *Unstructured additive Schwarz–conjugate gradient method for elliptic problems with highly discontinuous coefficients*, SIAM J. Sci. Comput., 20 (1999), pp. 2041–2066.

[14] M. F. Gyure, C. Ratsch, B. Merriman, R. E. Caflisch, S. Osher, J. J. Zinck, and D. D. Vvedensky, *Level set methods for the simulation of epitaxial phenomena*, Phys. Rev. E, 58 (1998), pp. R6927–R6930.

[15] W. Hackbusch, *Multi-grid Methods and Applications*, Springer-Verlag, Berlin, 1985.

[16] W. Hackbusch, *The frequency decomposition multigrid method, part* I: *Application to anisotropic equations*, Numer. Math., 56 (1989), pp. 229–245.

[17] P. W. Hemker, *On the order of prolongations and restrictions in multigrid procedures*, J. Comput. Appl. Math., 32 (1990), pp. 423–429.

[18] B. Koobus, M. H. Lallemand, and A. Derieux, *Unstructured Volume-Agglomeration MG: Solution of the Poisson Equation*, Technical Report 1946, INRIA, Sophia Antipolis, France, 1993.

[19] X. Liu, R. Fedkiw, and M. Kang, *A boundary condition capturing method for Poisson's equation on irregular domains*, J. Comput. Phys., 160 (2000), pp. 151–178.

[20] J. Mandel, M. Brezina, and P. Vaněk, *Energy optimization of algebraic multigrid bases*, Computing, 62 (1999), pp. 205–228.

[21] C. W. Oosterlee, *On multigrid for linear complementarity problems with application to American-style options*, Electron. Trans. Numer. Anal., 15 (2003), pp. 165–185.

[22] S. Osher and J. A. Sethian, *Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulation*, J. Comput. Phys., 79 (1988), pp. 12–49.

[23] A. Reusken, *Multigrid with matrix-dependent transfer operators for a singular perturbation problem*, Computing, 50 (1993), pp. 199–211.

[24] A. Reusken, *Multigrid with matrix-dependent transfer operators for convection-diffusion problems*, in Multigrid Methods, Vol. IV, Internat. Ser. Numer. Math. 116, Birkhäuser, Basel, 1994, pp. 269–280.

[25] J. W. Ruge and K. Stüben, *Algebraic multigrid*, in Multigrid Methods, S. McCormick, ed.,

SIAM, Philadelphia, 1987, pp. 73–130.
[26] M. Sussman, P. Smereka, and S. Osher, *A level set approach for computing solutions to incompressible two-phase flow*, J. Comput. Phys., 114 (1994), pp. 146–159.
[27] W.-P. Tang and W. L. Wan, *Sparse approximate inverse smoother for multigrid*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1236–1252.
[28] U. Trottenbery, C. Oosterlee, and A. Schüller, *Multigrid*, Academic Press, New York, 2001.
[29] P. Vaněk, J. Mandel, and M. Brezina, *Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems*, Computing, 56 (1996), pp. 179–196.
[30] W. L. Wan, T. F. Chan, and B. Smith, *An energy-minimizing interpolation for robust multigrid methods*, SIAM J. Sci. Comput., 21 (2000), pp. 1632–1649.
[31] J. Xu, *Iterative methods by space decomposition and subspace correction*, SIAM Rev., 34 (1992), pp. 581–613.
[32] P. M. De Zeeuw, *Matrix-dependent prolongations and restrictions in a blackbox multigrid solver*, J. Comput. Appl. Math., 33 (1990), pp. 1–27.