NUMERICAL LINEAR ALGEBRA WITH APPLICATIONS
Numer. Linear Algebra Appl. 2000; **7**:727–741

Interface preserving coarsening multigrid for elliptic problems with highly discontinuous coefficients

Wing Lok Wan*,†

SCCM Program, Gates Building 2B, Stanford University, Stanford, CA 94305-9025, U.S.A.

SUMMARY

We propose an interface preserving coarsening, a geometric technique for solving discontinuous coefficient partial differential equations by multigrid. It selects coarse grid points so that all the coarse grids are aligned with the interfaces for regular interface problems on structured grids, and so that the interfaces are resolved as much as possible for irregular interface problems. As a result, multigrid with linear interpolation is sufficient to obtain fast convergence. We show by one-dimensional and two-dimensional interface problems that the convergence rate of the resulting multigrid method is independent of the mesh size and the size of the jump at the interfaces. Copyright © 2000 John Wiley & Sons, Ltd.

KEY WORDS: coarsening; multigrid; interface; elliptic PDE; discontinuous coefficients

1. INTRODUCTION

Multigrid methods have been widely used for solving second order elliptic partial differential equations (PDEs). Their convergence rate is often independent of the mesh size, and such optimal convergence theory can be found, for instance, in References [1–8]. However, in practice, the convergence rate may depend on the PDE coefficients. Typically the convergence deteriorates as the coefficients become rougher. For example, if the coefficients have large jumps [4, 9–13], are highly oscillatory [14–16] or are anisotropic [4,17,18], standard multigrid methods will converge very slowly. Special techniques are needed to handle some of the problems. In this paper, we consider PDEs whose coefficients have jumps of several orders of magnitude:

$$\begin{aligned} -\nabla \cdot a\nabla u &= f \qquad \text{in } \Omega \\ u &= 0 \qquad \text{on } \partial\Omega \end{aligned} \tag{1}$$

* Correspondence to: Wing Lok Wan, SCCM Program, Gates Building 2B, Stanford University, Stanford, CA 94305-9025. U.S.A.
† E-mail: wan@sccm.stanford.edu

Contract/grant sponsor: US Department of Energy—Accelerated Strategic Computing Initiative (ASCI). Contract/grant sponsor: National Science Foundation; contract/grant number: contract ACR-0072112.

Received 20 December 1999
Revised 8 May 2000

where Ω is a polygon in $\mathbb{R}^d$, $d = 1, 2$, and the coefficient a is discontinuous. For ease of exposition, we assume that a is piecewise constant, i.e.

$$a(x) \equiv a_i \qquad x \in \Omega_i$$

where $\Omega_i \subset \Omega$, $\Omega_i \cap \Omega_j = \emptyset$ for $i \neq j$, and $\bar{\Omega} = \cup_i \overline{\Omega_i}$. Such problems arise in various applications such as flow in heterogeneous porous media [19], neutron transport [9] and biophysics [20]. Equation (1), or its variations, has also been used as a model problem in the multigrid literature [9–12,21] for solving discontinuous coefficient problems. The objective is to derive a multigrid method that is as insensitive to the jumps as possible.

A typical approach of improving the convergence is to define a sophisticated interpolation such as in black box multigrid [11,12]. The idea is to capture the discontinuous behaviour of the derivative of the solution along the interface, or, equivalently, to preserve the continuity of the flux across the interface [4,9,11,12]. However, these methods require a setup phase for the computation of the interpolation operator and extra storage for it. Besides, they usually only apply to structured grid problems. The robust energy-minimizing interpolation proposed by Wan, Chan and Smith [22] may apply to both structured and unstructured grids, but additional cost for the construction of the interpolation operator is still required. In this paper, we propose a coarsening scheme with which the simple linear interpolation suffices to produce fast multigrid convergence.

A successful multigrid method depends not just on interpolation but on all components as a whole. Our key observation is that coarsening can play a crucial role for interface problems. In scientific and engineering computations, we often have knowledge of the locations of the interfaces. Our idea, which is based on the technical report [23], is to select coarse grid points to resolve the shape of the interface in a certain sense to be described later. Intuitively speaking, experiences from the literature [4,9,21] indicated that the parts of the solution on the regions of different constant coefficients behave independently and are glued together through a Neumann boundary condition on the interfaces. Theoretically speaking, convergence analysis for interface problems [7,24,25] often requires that the discontinuities are preserved on all coarser grids. In view of these, we propose an interface preserving coarsening algorithm so that all the coarse grids are aligned with the interfaces for regular interface problems on structured grids, and that the interfaces are resolved as much as possible for irregular interface problems. Consequently, linear interpolation is sufficient to obtain fast multigrid convergence. In addition, we do not need to construct and store the interpolation operators for regular interfaces.

We remark that special coarsening techniques, for instance, semi-coarsening [17,18,26], are common for solving anisotropic coefficient PDEs. However, to the best of our knowledge, no coarsening strategy has been studied specifically for discontinuous coefficient problems. In algebraic multigrid (AMG) [27], which is applicable to a wide class of problems and, in particular, PDE problems with discontinuous coefficients, the coarsening is derived based on the notion of strong/weak coupling, which essentially measures the relative size of the off-diagonal entries. More precisely, node i is strongly connected to node j if

$$-A_{ij} \geq \theta \max_{k \neq i} \{-A_{ik}\}$$

where $0 < \theta \leq 1$ is a parameter. Hence, the geometric information of the interface is only exploited implicitly through the strong/weak coupling (i.e. relatively large matrix entries) occurring along the

interface nodes. Although the algebraic coarsening makes use of the abrupt changes of matrix entries arising from the interface, it does not intentionally align the coarse grid points with the interface as our proposed coarsening method does; see Section 4 for comparison. We also note that an algebraic interpolation is used in conjunction with the algebraic coarsening and hence the construction and storage of the interpolation operator is still needed.

Identification of surfaces, corners, etc. has been used by Adams and Demmel [28] in their node-nested multigrid method for solving three-dimensional unstructured grid finite element problems. In the node-nested approach [29], the coarse grids are constructed by retetrahedralization using a set of coarse grid points, and hence geometric details on the boundary are lost on the very coarse grids. Knowledge of surfaces and corners is useful in selecting coarse grid points to maintain the shape of the outer boundary during retetrahedralization. Thus, their primary focus is on the shape of the computational domain—whereas ours is on the location of the internal boundary formed by the interface. Besides, in their approach the coarse grids are formed by retetrahedralization where the geometric features such as surfaces and corners are still well-defined. In our approach, the coarse grids (coarse subspaces) are obtained by the Galerkin process and the geometric features become fuzzy.

We also remark that much work has been done on domain decomposition for discontinuous coefficient problems; see, for example References [6,30]. For many of these methods, the subdomains are naturally divided by the interfaces. Thus the true interfaces coincide with the computational subdomain interfaces. In multigrid, however, we do not usually have large subdomains and hence the interfaces do not generally align with the coarse grids.

Another approach for discontinuous coefficient problems is to use conjugate gradient preconditioned by diagonal scaling [31,32]. It is very simple and yet quite effective. Graham and Hagger [21] analysed such an approach theoretically, and extended the analysis to additive Schwarz preconditioners. They showed that the largest eigenvalues of the preconditioned system are bounded and that only a finite number of small eigenvalues approach to zero as the size of the jump increases. Thus the preconditioned conjugate gradient method converges within a number of iterations that grows only logarithmically with the size of the jump. In their analysis, the interfaces are not assumed to align with the coarse grids. Another related approach by Ashby *et al*. [19] is to use multigrid preconditioned conjugate gradients.

In Section 2, we first discuss the issues of coarsening for discontinuous coefficient problems and explain why sometimes it is mandatory to select coarse grid points in a special way. We illustrate the idea in one dimension in Section 3.1. In two dimensions, we discuss the regular and irregular interface case separately in Sections 3.2.1 and 3.2.2, respectively. The extension of the algorithms to three dimensions is possible but is not discussed in the present paper. In Section 4, the effectiveness of the interface preserving coarsening is demonstrated by one-dimensional and two-dimensional examples. Finally, concluding remarks are made in Section 5.

2. FAILURE OF MULTIGRID WITH STANDARD COARSENING

As mentioned in the previous section, robust interpolations have been the key for improving multigrid convergence for discontinuous coefficient problems. However, in some circumstances, especially in unstructured grid computations, robust interpolation is not sufficient, and hence it is mandatory to employ special coarsening strategy.

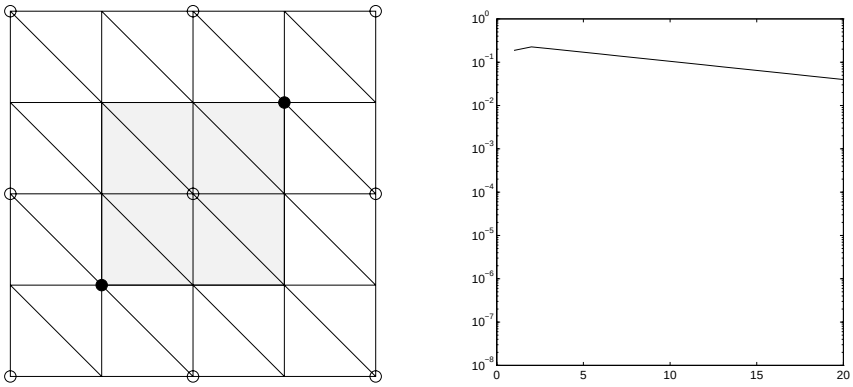

Figure 1. (a) A 5×5 grid with a jump in coefficient indicated by the shaded area. Coarse grid points are denoted by $\circ$. (b) Standard multigrid convergence.

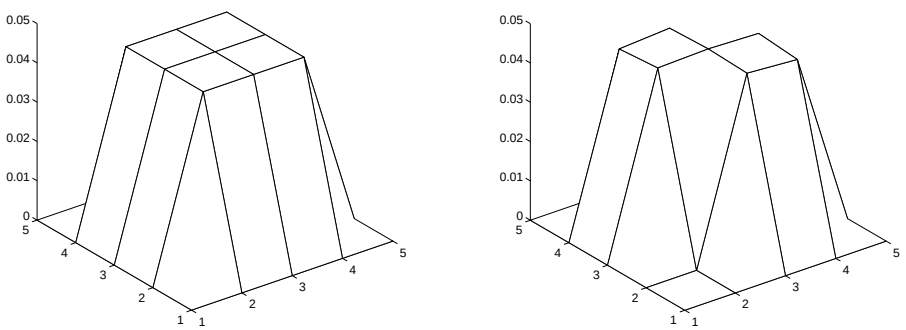

Figure 2. (a) Error after smoothing. (b) Error given by interpolation with standard coarsening.

Consider a square interface on a 5×5 triangular mesh as shown in Figure 1(a). The circles denote the coarse grid points chosen by standard coarsening. The jump in coefficient is indicated by the shaded area, and zero Dirichlet boundary condition is used. Since it is a triangular mesh, it is natural to interpolate the non-coarse grid points by the two (and only two) connected coarse grid points. The robust energy-minimizing interpolation described in [22] is used along with Gauss–Seidel smoothing. Other types of robust interpolation can be used as well. Slow multigrid convergence is shown in Figure 1(b). The key observation is that the interpolated values at the two non-coarse grid points marked by $\bullet$ are far from the true values of the errors. Figures 2(a) and 2(b) show the error after smoothing and the error given by interpolation. Owing to the large jump in coefficient, the error has larger values in the region with large coefficient and smaller values in the region with small coefficient. Suppose the Dirichlet boundary condition is used. Then the interpolated values at the two corners will be zero since their connected coarse grid points are on the boundary. In fact, even if other more sophisticated interpolations were used, the interpolated values had to be zero as long as they were interpolated only by the two connected coarse grid points on the boundary and the interpolation method approximates

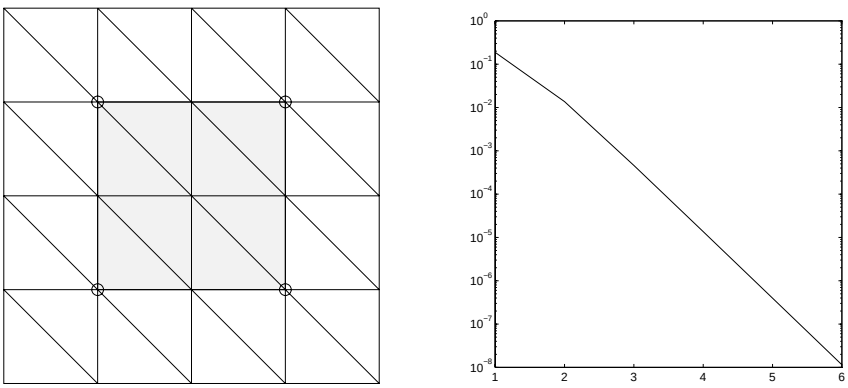

Figure 3. (a) Non-standard coarsening, denoted by ○, at the four corners of the square interface. (b) Standard multigrid convergence.

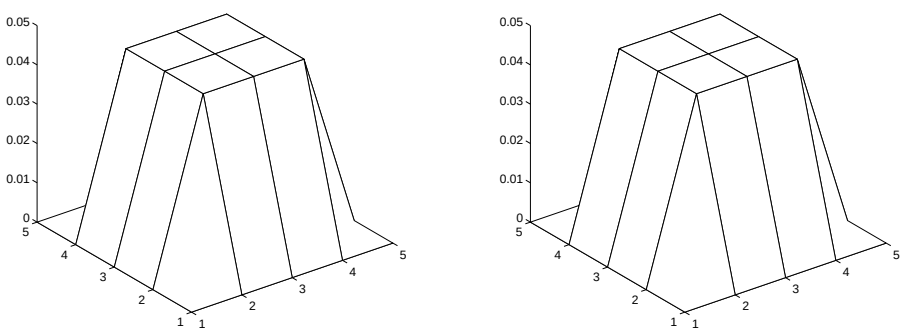

Figure 4. (a) Error after smoothing. (b) Error given by interpolation with non-standard coarsening.

constants exactly. Consequently, the approximation of the error on the coarse grid is very poor, which leads to slow multigrid convergence.

Based on this observation, a natural remedy is to select coarse grid points so that the previous situation will not happen. For instance, if we choose the coarse grid points shown in Figure 3(a), along with linear interpolation and Gauss–Seidel smoothing, we obtain the usual rapid multigrid convergence (Figure 3(b)). As shown in Figures 4(a) and 4(b), the true error and the interpolated error are essentially the same, and hence the fine grid errors can be corrected accurately by the coarse grid. In conclusion, a special coarsening is crucial to maintain fast multigrid convergence.

3. INTERFACE PRESERVING COARSENING

Standard multigrid converges slowly when the jumps differ by orders of magnitudes due to the inaccurate approximation given by linear interpolation across the discontinuities if the coarse grids do not align

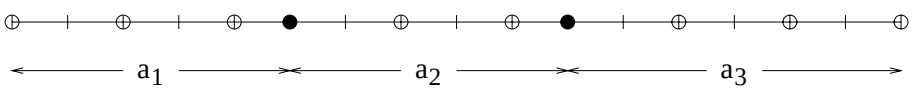

Figure 5. One-dimensional interface preserving coarsening without Step 2 (cf. Algorithm 1). The regions of different coefficient are denoted by a_1, a_2 and a_3. Coarse grid points are denoted by $\circ$ and the coarse grid points at the interfaces are denoted by $\bullet$.

with the interfaces. In the next sections, we describe an interface preserving coarsening technique in one and two dimensions that aligns the coarse grids with interfaces.

3.1. One dimension

The idea of the interface preserving coarsening is to select coarse grid points so that the interfaces are aligned with all the coarse grids. In one dimension, we simply assign the points at the interfaces to be the coarse grid points. After that, we perform standard coarsening for the remaining points; we select every other point as a coarse grid point (Figure 5). The idea can easily be extended recursively to coarser grids. The algorithm is described mathematically as follows.

Let $N^k = \{n_1^k, \ldots, n_{m_k}^k\}$ be the set of fine grid points on level k. Let N_C^k be the set of coarse grid points and $N_F^k = N^k \backslash N_C^k$ be the set of non-coarse grid points. The algorithm consists of the following three steps; see Algorithm 1.

Algorithm 1. 1D interface preserving coarsening

Step 1. Set $N_C^k = \{$ *interface points* $\} \equiv \{n_{i_1}^k, n_{i_2}^k, \ldots, n_{i_p}^k\}$.
Assume $n_{i_0}^k = n_1^k$ *and* $n_{i_{p+1}}^k = n_{m_k}^k$.
Step 2. (optional) Set $N_F^k = \{$ *neighbours of* $j : j \in N_C^k\}$.
Step 3. for $l = 0$ *to* p
for $j = n_{i_l}^k$ *to* $n_{i_{l+1}}^k$
if $j \notin N_F^k \cup N_C^k$ *then*
$N_C^k = N_C^k \cup \{j\}, \quad N_F^k = N_F^k \cup \{$ *neighbours of* $j\}$
end if
end for
end for

Step 1 selects the interface points as coarse grid points. Step 2 ensures that no two coarse grid points are adjacent to each other unless they are both interface points. Thus the number of coarse grid points is not more than half the number of fine grid points. The difference between applying and not applying Step 2 is shown in Figures 6 and 5, respectively. The former has fewer coarse grid points but more overlaps between basis functions and the opposite for the latter. Our default is not to apply Step 2. Step 3 performs the standard coarsening between the selected coarse grid points from Step 1.

The other multigrid components are standard. We use Gauss–Seidel as smoother and linear interpolation. We remark that the coordinate information of the computational points is needed to perform the linear interpolation owing to the non-uniformity in spacing of the coarse grid points.

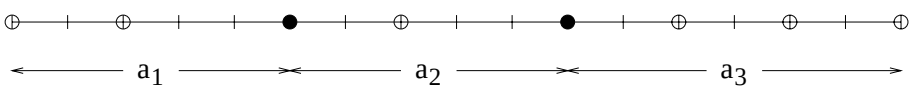

Figure 6. One-dimensional interface preserving coarsening with optional Step 2 (cf. Algorithm 1) in effect. No two consecutive points are coarse grid points.

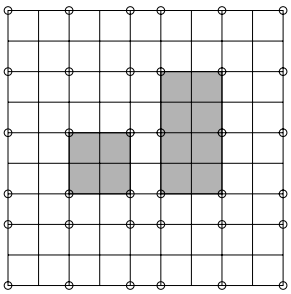

Figure 7. Tensor product interface preserving coarsening for a regular two-dimensional interface. Coarse grid points are denoted by ○.

3.2. Two dimensions

We describe separately the algorithm for the regular and irregular interface. For the former, we may apply the one-dimensional technique. For the latter, we need another algorithm to maintain low complexity while still being able to resolve the interfaces.

3.2.1. Regular interfaces. Suppose the grids are regular tensor product grids, and the regions of different coefficients, Ω_i, are also formed by tensor products. In other words, let

$$\begin{aligned} X &= \{x : x \text{ is the } x\text{-coordinate of an interface point}\} \\ Y &= \{y : y \text{ is the } y\text{-coordinate of an interface point}\} \end{aligned}$$

Then $\Omega_i = [x_1, x_2] \times [y_1, y_2]$ for some $x_1, x_2 \in X$ and $y_1, y_2 \in Y$. Figure 7 shows an example of a regular interface problem.

Algorithm 2. 2D regular interface preserving coarsening

Let Ω^k be the set of grid points on level k.
Step 1. Set $X^k = \{x : (x, y) \in \Omega^k\}$, $Y^k = \{y : (x, y) \in \Omega^k\}$.
Step 2. Set $N^k_{\mathrm{C}x}$ = coarse grid points obtained from 1D coarsening on X^k.
Step 3. Set $N^k_{\mathrm{C}y}$ = coarse grid points obtained from 1D coarsening on Y^k.
Step 4. Set $N^k_{\mathrm{C}} = \{(x, y) : x \in N^k_{\mathrm{C}x},\ y \in N^k_{\mathrm{C}y}\}$.

The two-dimensional regular interface preserving coarsening is obtained by a 'tensor product' of the one-dimensional algorithm; see Algorithm 2. A point $x \in X^k$ is an interface point if there exist a region Ω_i such that $\Omega_i = [x, \tilde{x}] \times [y, \tilde{y}]$ for some $\tilde{x} \in X^k$ and $y, \tilde{y} \in Y^k$. The set of coarse grid points, $N^k_{\mathrm{C}x}$, in X^k is obtained from applying Algorithm 1 to X^k. Similar coarsening procedure is performed in Y^k.

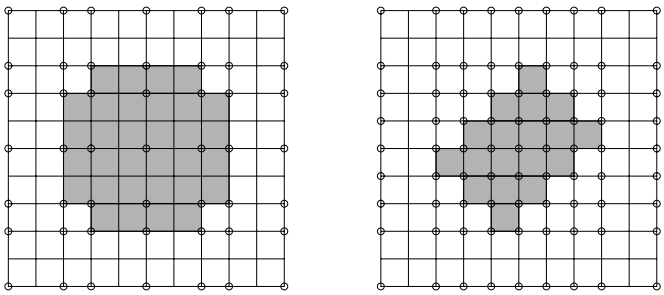

Figure 8. Tensor product interface preserving coarsening for (a) a less irregular interface, (b) very irregular interface. Coarse grid points are denoted by ○.

The final set of coarse grid points, N_{C}^k, is formed by taking the tensor products of the points in $N_{\mathrm{C}x}^k$ and $N_{\mathrm{C}y}^k$. A result of the regular interface preserving coarsening is illustrated in Figure 7. We note that an extension of the algorithm to three dimensions is straightforward.

3.2.2. Irregular interfaces. We can still apply the regular techniques if the interfaces are not too irregular. For example, for the interface shown in Figure 8(a), we may apply Algorithm 2 to obtain a coarsening of the grid points; we collect the x and y coordinates and determine their interface points as before, and then coarsen along the x- and y-directions independently. Figure 8(a) shows the resulting coarse grid points. For less regular interfaces, however, this tensor product procedure may create an excessive number of coarse grid points to preserve the interfaces; see Figure 8(b). Consequently, the overall computational cost may increase.

In general, we want to resolve the shape of the interface while maintaining the coarse-to-fine grid point ratio to be 1:4 as closely as possible. The idea is as follows. We first identify the points where the interfaces are located. Then we apply standard coarsening to these points and then to the remaining points. We convert some of the non-coarse grid points near the interfaces to coarse grid points so that the shapes of the interfaces are better resolved.

The above procedures can be described mathematically as follows. For simplicity, we describe the two-regions case only. Let Ω^+, Ω^- be disjoint open subsets of Ω such that $\Omega = \overline{\Omega^+} \cup \Omega^-$. Let $\Gamma = \partial\Omega^+$ be the interface and $\Gamma \cap \partial\Omega = \emptyset$. Let $a(x, y) \equiv a^+$ in Ω^+ and $a(x, y) \equiv a^-$ in Ω^- and $a^- \ll a^+$. The case of multiple interfaces is treated similarly and hence is omitted. The two-dimensional irregular interface preserving coarsening is given in Algorithm 3.

Algorithm 3. 2D irregular interface preserving coarsening

Step 1. Determine the set of fine grid points N^+ in Ω^+, fine grid points N^- in Ω^-, and the set of interface points $I = \overline{\Omega^+} \cap \overline{\Omega^-}$.
Step 2. Full coarsening on I.
Step 3. Full coarsening on N^+.
Step 4. Full coarsening on N^-.
Step 5. Change a non-coarse grid point x_i^h to a coarse grid point if any of the following conditions is satisfied:

(i) $x_i^h \in N^-$ and no coarse grid point $x_j^H \in N^-$ is connected to x_i^h;

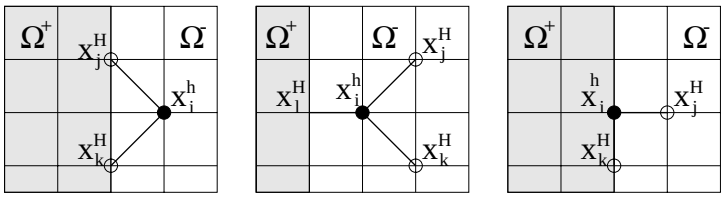

Figure 9. (a) A non-coarse grid $x_i^h \in \Omega^-$ has no coarse grid neighbours in Ω^-. (b) A non-coarse grid $x_i^h \in \Omega^-$ has a neighbour $x_l^h \in \Omega^+$ that is not a coarse grid. (b) A non-coarse grid $x_i^h \in \Omega^+$ is connected to one coarse grid $x_k^H \in \Omega^+$ only.

(ii) $x_i^h \in N^-$ *has neighbours in* N^+ *but none of them is a coarse grid point;*
(iii) $x_i^h \in N^+$ *is connected to one coarse grid point in* N^+*;*
(iv) x_i^h *is connected to one coarse grid point only.*

Note. During interpolation, coarse grid points in N^- *do not interpolate the non-coarse grid points in* N^+.

Remark 1
Step 1. Notice that the diagonal entries of the stiffness matrix $\mathcal{A}$ are

$$\mathcal{A}_{ii}^h = a^+ \int_\Omega |\nabla \phi_i^h|^2 \, \mathrm{d}x$$

for all $x_i^h \in N^+$. If $a^+ \gg a^-$, the set of points in N^+ can be easily identified by the large diagonal entries of $\mathcal{A}_{ii}^h$.

Steps 2, 3, 4. In full/standard coarsening, if a fine grid point is selected as a coarse grid point, its non-coarse grid point neighbours will not be selected as coarse grid points.

Step 5. The coarsening steps 2–4 may give rise to bad connectivities of non-coarse grid points to coarse grid points, which can lead to poor interpolation since non-coarse grid points are interpolated only by their neighboring coarse grid points. Criteria (i)–(iv) are used to eliminate such unwanted situations. (i) As shown in Figure 9(a), if $x_i^h \in \Omega^-$ is connected to coarse grids (e.g. x_j^H, x_k^H) in Ω^+ only, the interpolated value can be poor since the solution values in Ω^+ are usually much larger than those in Ω^-. (ii) If $x_i^h \in \Omega^-$ has neighbours in Ω^+, it must be close to the interface and hence the interpolated values given by the coarse grid neighbors (e.g. x_j^H, x_k^H) in Ω^- may underestimate the true value; see Figure 9(b). (iii) Since the values in Ω^+ including the boundary $\partial\Omega^+$ are typically larger, non-coarse grid points in Ω^+ are only interpolated by coarse grid points in Ω^+. (iv) Finally, if x_i^h connected to one coarse grid point x_j^H only, the interpolated value will be the same as the value at x_j^H if the interpolation operator preserves constants. In other words, it is just piecewise constant interpolation which is not accurate enough for solving second order PDEs.

The note at the end of Algorithm 3 requires that, during interpolation, the coarse grid values on Ω^- do not interpolate the non-coarse grid points in Ω^+. However, the opposite is permissible. This is because steps 2–5 guarantee that the non-coarse grid points in Ω^+ are connected to at least two coarse grid

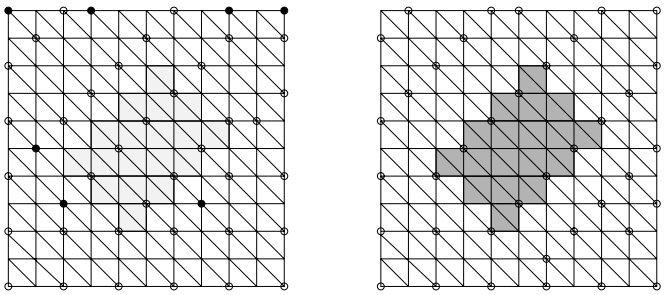

Figure 10. Irregular interface preserving coarsening for an irregular interface. Coarse grid points are denoted by ○.

points in Ω^+ while the non-coarse grid points in Ω^- are allowed to connect to one coarse grid point in Ω^+ and another one in Ω^-.

Figure 10 is obtained by applying Algorithm 3 to the irregular interface shown in Figure 7. Here, we consider a triangular mesh. Compared with Figure 8(b), we need many fewer coarse grid points. Moreover, within each constant coefficient region, the coarse grid points are selected as if in standard coarsening.

3.2.3. Complexity issue. Resolving the interface, in general, increases the total number of coarse grid points, which in turn increases the overall computational cost. However, if the interface forms a simple piecewise smooth curve, the increase in the number of coarse grid points is at most the total number of grid points on the interface, which is only $O(1/h)$ compared to $O(1/h^2)$ total number of coarse grid points. Suppose the complexity, η_0, of one V-cycle multigrid with standard coarsening is estimated to be

$$\eta_0 = O\left(\frac{1}{h^2} + \frac{1}{4h^2} + \frac{1}{16h^2} + \cdots\right) = O\left(\frac{4}{3h^2}\right).$$

Then the complexity η_1 of one V-cycle multigrid with our special coarsening is

$$\begin{aligned}\eta_1 &= O\left(\frac{1}{h^2} + \left(\frac{1}{4h^2} + \frac{1}{h}\right) + \left(\frac{1}{4}\left(\frac{1}{4h^2} + \frac{1}{h}\right) + \frac{1}{2h}\right)\right.\\ &\quad \left. + \left(\frac{1}{4}\left(\frac{1}{4}\left(\frac{1}{4h^2} + \frac{1}{h}\right) + \frac{1}{2h}\right) + \frac{1}{4h}\right) + \cdots\right)\\ &= O\left(\frac{4}{3h^2} + \frac{8}{3h}\right)\end{aligned}$$

Thus the extra amount of work due to the increase in the number of coarse grid points is asymptotically small compared with the standard one.

Table I. Convergence of multigrid methods for a 1D discontinuous coefficient problem. Method 1 uses interface preserving coarsening and linear interpolation. Method 2 uses standard coarsening and flux preserving interpolation.

h	MG Method 1	MG Method 2	Standard MG
1/32	6	6	16
1/64	6	6	17
1/128	6	6	20
1/256	6	6	24

4. NUMERICAL RESULTS

We demonstrate the effectiveness of the interface preserving coarsening by three examples. The multigrid settings are standard. We apply two pre- and post- Gauss–Seidel smoothings. Linear interpolation is used for the one-dimensional problem and the two-dimensional regular interface problem. For the irregular interface problem, since the tensor product grid structure is destroyed after one level of coarsening, we have to use unstructured grid multigrid techniques for the subsequent coarser grids. We could use sophisticated interpolation such as that of Reference [22]. However, since we do no need to interpolate values across the interface we can use a simpler and more naive approach. In particular, we use the graph distance to define the interpolation weighting, which resembles linear interpolation on Cartesian grids. A more precise definition is described in Example 3. The Galerkin method is used to construct the coarse grid matrices. The multigrid iteration was terminated when the relative residual norm was less than 10^{-6}.

Example 1. We compare the convergence results of the multigrid method with one-dimensional interface preserving coarsening and linear interpolation and with that of the standard coarsening and flux preserving interpolation [4]. The model equation is

$$\begin{aligned} -\frac{\mathrm{d}}{\mathrm{d}x}a(x)\frac{\mathrm{d}}{\mathrm{d}x}u(x) &= 1 \qquad \text{in } (0,1), \\ u &= 0 \qquad \text{at } x = 0 \text{ and } x = 1 \end{aligned}$$

where

$$a(x) = \begin{cases} 10^4 & \text{if } x \le 1/4 + h \\ 1 & \text{if } 1/4 + h < x \le 1/2 + h \\ 10^2 & \text{if } x > 1/2 + h \end{cases}$$

Here, h is the size of the fine grid. The coefficient $a(x)$ is so designed that the interfaces are not aligned with any standard coarse grids. The convergence results of the two multigrid methods are shown in Table I. There is no difference in performance for between these two methods. Specifically, both convergence results are independent of the mesh size h and the jump in the coefficient. Thus, once the coarse grid points are aligned with the interfaces, we do not need a sophisticated interpolation to capture the jumps in the derivative of the approximate solution.

Table II. Convergence of multigrid methods for a 2D discontinuous coefficient problem. Method 1 uses interface preserving coarsening and linear interpolation. Method 2 uses AMG coarsening and AMG interpolation. The numbers 10, 10^2, 10^4 indicate the values of a^+, and $*$ denotes convergence more than 100 iterations.

	MG Method 1			MG Method 2			Standard MG		
h	10	10^2	10^4	10	10^2	10^4	10	10^2	10^4
1/16	5	5	6	7	7	7	14	$*$	$*$
1/32	5	6	6	7	7	8	14	$*$	$*$
1/64	6	6	6	8	9	9	14	$*$	$*$
1/128	6	6	6	8	9	10	14	$*$	$*$

Example 2. We show the effectiveness of the two-dimensional tensor product interface preserving coarsening for regular interfaces. The problem is a modification of Example I in Reference [9]:

$$-\nabla \cdot a(x, y)\nabla u = 1$$

where

$$a(x, y) = \begin{cases} a^+ & 0.25 - h \leq x \leq 0.75 - h \quad \text{and} \quad 0.25 - h \leq y \leq 0.75 - h \\ a^- & \text{otherwise} \end{cases}$$

We fix $a^- = 1$ and vary a^+ from 10 to 10^4. The square interface is shifted by one fine grid point so that the interface does not align with any coarse grid if standard coarsening were used. Finite element method with bilinear elements is used to discretize the equation. In this example, we compare our method with AMG [33] which utilizes strong/weak coupling for algebraic coarsening instead of interface location. Convergence results of the multigrid method with interface preserving interface and linear interpolation are compared with those of AMG coarsening and AMG interpolation in Table II. We see that the performance of the proposed method is essentially the same as in the one-dimensional case and compares favourably with the AMG method. Here, we do not need a setup phase for the construction and storage of a sophisticated interpolation operator.

To illustrate the difference between the interface preserving coarsening and AMG coarsening, we show the two set of coarse grid points in Figure 11. The coarse grid points of the first and second coarse grids are denoted by $\circ$ and $\square$, respectively. We see that the interface preserving coarsening resolves the shape of the square interface nicely by the coarse grid points whereas AMG coarsening roughly resolves the square shape by a set of irregularly placed coarse grid points of the first grid and only barely identifies the square interface by the coarse grid points of the second grid.

Example 3. Finally, we present the results of an irregular interface problem. The star-shaped interface is shown in Figure 12(a), and the irregular interface coarsening on the 16×16 triangular mesh is shown in Figure 12(b). Since the coefficient is assumed to be constant on each triangle, a wrinkleyness results on the interface. We can see that more coarse grid points are selected near the interface and the remaining parts are coarsened as in full coarsening.

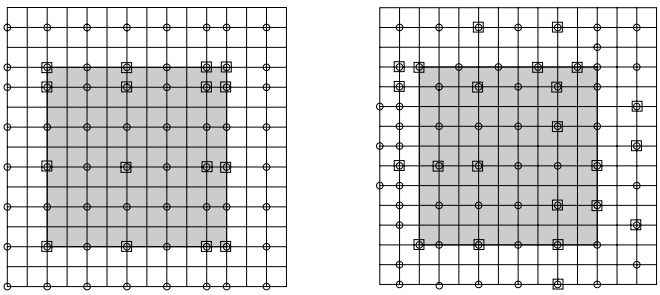

Figure 11. The coarse grid points of the first and second coarse grid are denoted by ○ and □, respectively. (a) Interface preserving coarsening. (b) AMG coarsening.

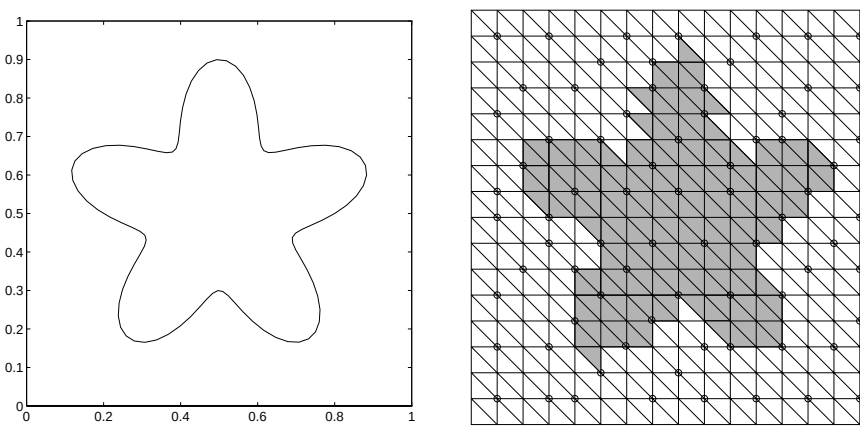

Figure 12. (a) A star-shaped interface is formed by: (x, y), $x = 0.5 + r\cos\theta$, $y = 0.5 + r\sin\theta$ where $r = 0.3 + 0.1\sin(5\theta)$, $0 \leq \theta \leq 2\pi$. (b) The star-shaped interface in (a) on a 16×16 triangular mesh.

In this example, both multigrid methods are used as a preconditioner for the conjugate gradient method. Four, five, six and seven grids are used for 16×16, 32×32, 64×64 and 128×128 grids, respectively. In this irregular interface case, we cannot use the usual linear interpolation since the Cartesian grid structure is destroyed after one level of coarsening. We use the graph distance to define the interpolation weighting, which mimics linear interpolation on Cartesian grids. More precisely, if a non-coarse grid point is connected to K coarse grid points, then the interpolation weighting from each coarse grid point is $1/K$. Since the interpolation is defined by the connectivity graph of the coarse grid matrices, we need neither construct nor store the interpolation operators, as opposed to AMG.

The convergence results of the multigrid method using the irregular interface preserving coarsening and AMG are given in Table III. We see that the convergence rates of both methods are independent of the jump and the mesh size. Apparently, AMG takes fewer iterations. However, we note that the coarse-to-fine grid point ratio of AMG coarsening is between 1:2 and 1:3, whereas the interface preserving coarsening maintains a ratio of between 1:3 and 1:4. Hence, AMG takes about 25 per cent more coarse grid points and the number of non-zeros in the coarse grid matrices are also 2 to 4 times denser than ours. If we measure the efficiency by the total number of flops, both methods are essentially the same.

Table III. Convergence of multigrid methods for the star-shaped interface problem. Method 1 uses interface preserving coarsening and linear interpolation. Method 2 uses AMG coarsening and AMG interpolation. The numbers 10, 10^2, 10^4 indicate the values of a^+.

	MG Method 1			MG Method 2		
h	10	10^2	10^4	10	10^2	10^4
1/16	7	7	7	5	5	5
1/32	9	9	10	5	5	6
1/64	11	11	12	6	6	6
1/128	13	14	14	8	8	8

But notice that the proposed method does not require the setup phase for the construction and storage of the interpolation operators.

5. CONCLUDING REMARKS

We have demonstrated numerically that coarsening can be an efficient alternative for robust interpolation to solve discontinuous coefficient problems using multigrid. In fact, we have shown by an example that special coarsening is sometimes mandatory, no matter what the interpolation is. We have shown that multigrid with the proposed interface preserving coarsening and simple linear interpolation is an effective solution method for discontinuous coefficient problems on structured grids. In general, we need to combine the interface preserving coarsening and an unstructured grid interpolation for irregular interface problems on general domains, and we have shown by an example how this can be done.

ACKNOWLEDGEMENTS

The author would like to thank Tony Chan for his constructive comments and suggestions throughout the development of this paper.

REFERENCES

1. Bramble J. *Multigrid Methods*. Longman Scientific & Technical, UK, 1993.
2. Bramble J, Pasciak J, Wang J, Xu J. Convergence estimates for multigrid algorithms without regularity assumptions. *Mathematics of Computing* 1991; **57**:23–45.
3. Bramble J, Pasciak J, Wang J, Xu J. Convergence estimates for product iterative methods with applications to domain decomposition and multigrid. *Mathematics of Computing* 1991; **57**:1–21.
4. Hackbusch W. *Multi-grid Methods and Applications*. Springer-Verlag, Berlin, 1985.
5. McCormick S. *Multigrid Methods*. SIAM, Philadelphia, PA, 1988.
6. Smith B, Bjorstad P, Gropp W. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press: Cambridge, UK, 1996.
7. Xu J. Theory of multilevel methods. PhD thesis, Cornell University, Ithaca, NY, 1989.

8. Xu J. Iterative methods by space decomposition and subspace correction. *SIAM Review*, 1992; **34**:581–613.
9. Alcouffe RE, Brandt A, Dendy JE, Jr, Painter JW. The multi-grid method for the diffusion equation with strongly discontinuous coefficients. *SIAM Journal on Scientific and Statistical Computing* 1981; **2**(4):430–454.
10. Behie A, Forsyth PA. Multi-grid solution of three-dimensional problems with discontinuous coefficients. *Applied Mathematics Computing* 1983; **13**:229–240.
11. Dendy JE, Jr. Black box multigrid. *Journal of Computational Physics* 1982; **48**:366–386.
12. Dendy JE, Jr. Black box multigrid for nonsymmetric problems. *Applied Mathematics Computing* 1983; **13**:261–283.
13. Kettler R, Meijerink JA. A multigrid method and a combined multigrid-conjugate gradient method for elliptic problems with strongly discontinuous coefficients in general domains. Shell Publ. 604, KSEPL, Rijswijk, 1981.
14. Engquist B, Luo E. Convergence of a multigrid method for elliptic equations with highly oscillatory coefficients. *SIAM Journal on Numerical Analysis* 1997; **34**:2254–2273.
15. Luo E. Multigrid method for elliptic equation with oscillatory coefficients. PhD thesis, Department of Mathematics, UCLA, Los Angeles, 1993.
16. Ta'asan S. Multigrid methods for highly oscillatory problems. PhD thesis, Weizmann Institute of Science, Rehovat, Israel, 1984.
17. Dendy JE, Jr, Ida MP, Rutledge JM. A semi-coarsening multigrid algorithm for SIMD machines. *SIAM Journal on Scientific and Statistical Computing* 1992; **13**:1460–1469.
18. Smith RA, Weiser A. Semicoarsening multigrid on a hypercube. *SIAM Journal on Scientific and Statistical Computing* 1992; **13**:1314–1329.
19. Ashby SF, Falgout RD, Smith SG, Fogwell TW. Multigrid preconditioned conjugate gradients for the numerical simulation of groundwater flow on the Cray T3D. Technical Report, Lawrence Livermore National Laboratory, Livermore, CA, 1996.
20. Holst MJ. Multilevel methods for the Poisson–Boltzmann equation. PhD thesis, University of Illinois, Urbana-Champaign, 1993.
21. Graham IG, Hagger MJ. Unstructured additive schwarz-cg method for elliptic problems with highly discontinuous coefficients. In *Proceedings of the Ninth Domain Decomposition Conference*, 1996.
22. Wan WL, Chan TF, Smith B. An energy-minimizing interpolation for robust multigrid. Technical Report 98-6, Department of Mathematics, UCLA, February 1998.
23. Wan WL. Interface preserving coarsening multigrid for elliptic problems with highly discontinuous coefficients. Technical Report CAM 98-27, Department of Mathematics, UCLA, May 1999.
24. Bramble J, Xu J. Some estimates for a weighted l^2 projection. *Mathematics of Computation* 1991; **56**:463–476.
25. Dryja M, Sarkis M, Widlund O. Multilevel Schwarz methods for elliptic problems with discontinuous coefficients in three dimensions. *Numerische Mathematik* 1996; **72**:313–348.
26. Dendy JE, Jr, McCormick SF, Ruge JW, Russell TF, Schaffer S. Multigrid methods for three-dimensional petroleum reservoir simulation. In *10th Symposium on Reservoir Simulation*, Houston, TX. 6–8 February 1989. Society of Petroleum Engineers (SPE), 19–25.
27. Ruge JW, Stüben K. Algebraic multigrid. In *Multigrid Methods*, McCormick S (ed.). SIAM, 1987; 73–130.
28. Adams M, Demmel J. Parallel multigrid solver algorithms and implementations for 3D unstructured finite element problem. In *ACM/IEEE Proceedings of SC99: High Performance Networking and Computing*. Portland, Oregon, November 1999.
29. Chan TF, Smith B, Zou J. Multigrid and domain decomposition methods for unstructured meshes. In *Procedings of the Third International Conference on Advances in Numerical Methods and Applications*, Sofia, Bulgaria, Dimov IT, Sendov Bl and Vassilevski P (eds). World Scientific, 1994; 53–62.
30. Chan TF, Mathew TP. Domain decomposition algorithms. In *Acta Numerica*. Cambridge University Press, Cambridge, UK, 1994; 61–143.
31. Greenbaum A, Li C, Chao H. Parallizing PCG algorithms. *Computational Physics Communications* 1989; **53**:295–309.
32. Ramage A, Wathen A. On preconditioning for finite-element equations on irregular grids. *SIAM Journal on Matrix Analysis and Applications* 1994; **15**:909–921.
33. GMD software. `http://mgnet.ccs.uky.edu/mgnet/Codes/gmd`.

