Title
Hybrid Multigrid for Adaptive Fourth Order Cut Cells:

Authors
Devendran, Dharshi
Graves, Daniel T.
Johansen, Hans

Publication Date
2014-11-01

Noname manuscript No.
(will be inserted by the editor)

A Hybrid Multigrid Algorithm for Poisson's equation using an Adaptive, Fourth Order Treatment of Cut Cells

D. Devendran · D. T. Graves · H. Johansen

Received: date / Accepted: date

Abstract We present a hybrid geometric-algebraic multigrid approach for solving Poisson's equation on domains with complex geometries. The discretization uses a novel fourth-order finite volume cut cell representation to discretize the Laplacian operator on a Cartesian mesh. This representation is based on a weighted least-squares fit to a cell-averaged discretization, which is used to provide a conservative and accurate framework for the multi-resolution discretization, despite the presence of cut cells. We use geometric multigrid coarsening with an algebraic multigrid bottom solver, so that the memory overhead of algebraic coarsening is avoided until the geometry becomes underresolved. With tuning, the hybrid approach has the simplicity of geometric multigrid while still retaining the robustness of algebraic multigrid. We investigate at what coarse level the transition should occur, and how the order of accuracy of the prolongation operator affects multigrid convergence rates. We also present some converged solutions as examples of how the use of adaptivity and a cell connectivity graph can affect performance in cases with underresolved geometries.

Keywords Cut cell · Adaptive Refinement · Multigrid · Higher Order

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research.

D. Devendran
Lawrence Berkeley National Laboratory
E-mail: pdevendran@lbl.gov

D. T. Graves
Lawrence Berkeley National Laboratory
E-mail: dtgraves@lbl.gov

H. Johansen
Lawrence Berkeley National Laboratory
E-mail: hjohansen@lbl.gov

1 Introduction

We are solving Poisson's equation with a charge density ρ, written as

$$\nabla \cdot (\nabla \phi) = \rho \tag{1}$$

for the potential ϕ. As a discretization, we use a fourth order, cell averaged, finite volume discretization in preparation ([12], with link to arXiv) for Cartesian domains with cut cells (see Figure 1). Because the resulting stencil is large, higher-order, and spatially varying, traditional algebraic multigrid approaches would require significant memory overhead, or risk the potential of poor multigrid performance resulting from aggressive algebraic coarsening approaches. Thus in this paper, we wish to investigate the factors that affect geometric multigrid performance of this algorithm and investigate under what circumstances it is advantageous to use algebraic multigrid.

The use of cut cells for representing complex geometries has been combined with multigrid solvers and adaptive mesh refinement previously, but usually at lower order. In [6], a first-order cut cell method was used for complex inclusions with small gaps; the authors demonstrated that extra care must be taken with the discretization near gaps to maintain good multigrid convergence. An efficient adaptive mesh first-order cut cell method for Poisson's equation for cosmological simulations was shown in [13]; they used a specialized prolongation operator to improve multigrid convergence. Another paper exploring prolongation operators for first-order cut-cells was [8]. There the authors achieved good multigrid convergence by enumerating prolongation and smoothing operators for each way the embedded boundary cut finite difference grid lines, and some benefit was seen with additional smoothings near the boundary. This Poisson problem routinely appears in projection methods for incompressible flows, such as [14], but again first-order treatments of boundary conditions limit the accuracy of the discretization. In a more complex example, in [4] the steady-state compressible flow was solved using multigrid, noting that thin boundary layers requiring piecewise linear prolongation, and careful tracking of connectivity on coarser levels. In many of these references, complicated geometries, adaptivity, and boundary conditions make multigrid convergence for cut cell discretizations difficult.

The basis for our work is in first-order finite volume Poisson solvers for embedded boundary calculations. The algorithm described in [17] is widely used in projection methods [16,20,19,11] and its performance has been investigated [21]. This algorithm is defined using pointwise data and reduces to the standard (five-point in two 2D, seven point in 3D) Laplacian discretization away from the embedded boundary. The current algorithm described in this paper uses a cell-averaged description of the data and away from the boundary, and has a much larger and denser stencil (see Figure 2). This stencil is higher-order, and therefore can have very different multigrid convergence properties.

In complex geometries we expect that, as geometric multigrid coarsens the domain, the operator evaluated on that coarsened domain can be a poor approximation to the finer grid operator. This can cause geometric multigrid

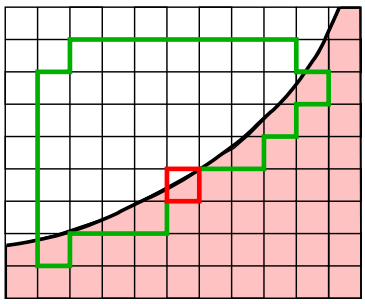

Fig. 1 Stencil support (in green) for a cut cell (in red).

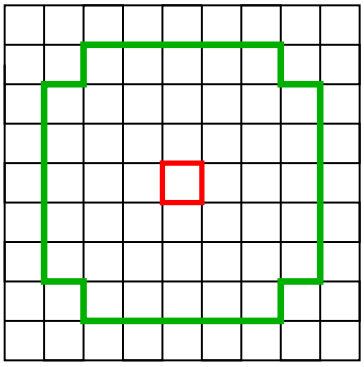

Fig. 2 Stencil support (in green) for a regular (uncut) cell (in red) in two dimensions.

convergence to deteriorate. If we can detect at what resolution geometric coarsening will fail to produce an adequately accurate operator, we can stop at this resolution and use algebraic multigrid as a bottom solver. In this way, we can retain the performance advantages of geometric multigrid, which include lower memory overhead and better parallel scaling. For higher order operators, the low order prolongation used in [17] can produce high frequency errors that require extra relaxation steps to achieve multigrid convergence.

For our geometric solvers, we use the Chombo software infrastructure [9, 10]. As our algebraic multigrid solver, we interface with PETSc [2,1,3].

1.1 Notation

The spatial discretization starts with rectangular control volumes on a Cartesian mesh in $1 \leq d \leq D$ dimensions, with mesh spacing h, and each grid cell indexed by a multi-index i. Given an irregular domain Ω, we define control volumes V_i defined on intersection of the Cartesian mesh with Ω. These cells may be "regular," meaning they do not contain any part of the domain boundary $\partial\Omega$, or otherwise "cut" by $\partial\Omega$. For a sufficiently small geometric feature, a cell may be cut into multiple "volumes," each of which is indexed by V_v, and

we use this notation generally to index any regular (V_i), cut cell, or multiple cut cell. The intersections of $\partial\Omega$ with the faces of each Cartesian grid volume are areas A, which again may be one or more faces connecting cells, including "regular" faces with half-indices $i \pm \frac{1}{2}$. We also define $A_{b,v}$ to be the surface of the boundary $\partial\Omega$ contained in volume V_v. We limit the pathologies of these intersections and small features by assuming that the boundary is resolved at the finest discretization, where it intersects a given cell face at most once.

Throughout this document, we use the following compact "multi-index" notation:

$$(\mathbf{x} - \bar{\mathbf{x}})^{\mathbf{p}} = \prod_{d=1}^{D} (x_d - \bar{x}_d)^{p_d}$$

$$\mathbf{p}! = \prod_{d=1}^{D} p_d!$$

Given a point in space $\bar{\mathbf{x}}$, and a D-dimensional integer vector $\mathbf{p}$, we define $m_v^{\mathbf{p}}(\bar{\mathbf{x}})$ to be the $\mathbf{p}^{th}$ moment of the volume V relative to the point $\bar{\mathbf{x}}$.

$$m_v^{\mathbf{p}}(\bar{\mathbf{x}}) = \int_V (\mathbf{x} - \bar{\mathbf{x}})^{\mathbf{p}} dV \tag{2}$$

The volume V_v of the cell v is simply $|V| = m_v^{\mathbf{z}}$, where $\mathbf{z}$ is the zero vector. We generate our geometric moments using the algorithm described in [18].

Given a sufficiently smooth function ψ, we can approximate ψ in the neighborhood of $\bar{\mathbf{x}}$ using a multi-dimensional Taylor expansion:

$$\psi(\mathbf{x}) = \sum_{|\mathbf{q}|<Q} \frac{1}{\mathbf{q}!} \psi^{(\mathbf{q})}(\bar{\mathbf{x}}) \, (\mathbf{x} - \bar{\mathbf{x}})^{\mathbf{q}} + O(h^Q), \tag{3}$$

with the multi-index partial derivative notation

$$\psi^{(\mathbf{q})} = \partial^{\mathbf{q}} \psi = \frac{\partial^{q_1}}{\partial x_1^{q_1}} \cdots \frac{\partial^{q_D}}{\partial x_D^{q_D}} \psi \,. \tag{4}$$

We express averages over volumes as

$$\langle \psi \rangle_v = \frac{1}{|V_v|} \int_{V_v} \psi \, dV$$

It is convenient to define a volume fraction κ that is inside Ω as the fraction of a full cell volume:

$$\kappa_v = h^{-D} |V_v| = h^{-D} m_v^{\mathbf{z}}. \tag{5}$$

We define a polynomial fit to the Poisson solution, ψ, using the cell-averaged form of the Taylor series (3)

$$\langle \psi \rangle_v = \frac{1}{m_v^{\mathbf{z}}} \sum_{|\mathbf{q}|<Q} c_{\mathbf{q}} m^{\mathbf{q}} \quad \text{or} \tag{6}$$

$$\Psi = M \, c \tag{7}$$

in vector form. Finally, if we choose the number stencil points $N_S > Q$ points, the system is over-determined and we can approximate the coefficients c using weighted least-squares (WLS),

$$c = (W\,M)^{\dagger} W \Phi\,, \tag{8}$$

where the choice of W is such that points further away contribute less to the lower-order coefficients. Specifically, the weights decrease as x^{-5} where x is the distance from the cell or face where the coefficients are needed. (see [12] for more details on the WLS approximation and stencils). With these coefficients, we create discrete operators for face-averaged gradients, prolongation, and other operators required in the multigrid algorithm for the discrete Laplacian operator, L, all of which use the same approach for incorporating partial cells and boundary conditions.

1.2 AMR and Multigrid Algorithm

We solve equation (1) on an adaptive block-structured hierarchy. The approach extends cell-average geometric multigrid V-cycle, as described in [15], to include higher-order operators on cut-cells. The cut-cells grids are coarsened geometrically; that is, the coarse grid cell volumes and face areas are obtained by summing the relevant fine grid quantities. One of the unique properties of this cell-average approach is the restriction to coarse grids is defined as a simple volume-weighted average. We choose a level of coarsening at which to switch from geometric to algebraic multigrid, which uses a PETSC solver [2, 1,3] to solve the matrix formed from the cut-cell Laplacian operator on the coarsest level.

An outline of the algorithm is given in figure 3 for a single-level calculation. We repeat this V-cycle iteratively until the residual reaches solution tolerance, typically machine precision scaled by the condition number of the Laplacian matrix. Define n_s to be the number of iterations of the smoothing operator, $S(R)$. The restriction operator A (for "averaging"), and the prolongation operator P are defined in §1.3. N_{max} is the grid refinement at the finest level, N_{min} is the refinement at which we switch to algebraic multigrid.

For AMR calculations, the algorithm can somewhat more complicated if the refinement ratio is greater than two but is exactly the same if the refinement ratio is two (see [15] for details). All our refinement ratios are two.

1.3 Multigrid Operators

For multigrid applied to linear operators like (1), we may use a residual-correction formulation. For a given charge distribution ρ and an approximation to the solution ϕ^l we define a residual of iteration l to be $R^l = \rho - L(\phi^l)$, and then solve for a correction δ^l such that

$$L(\delta^l) = R^l$$

1. Descending leg of V-cycle.
   1a. Compute residual at finest level.
   $R_{N_{max}} = \rho - L(\phi_{N_{max}})$
   for $N = \{N_{max}, N_{max}/2, N_{max}/4...N_{min}\}$
      1b. Smooth the correction.
         $\delta_N = S(R)$
      1c. Recompute residual and restrict (average) to the next level.
         $R_N = R_N - L(\delta_N))$
         $R_{N/2} = A(R_N)$
2. Solve $L(\delta_{N_{min}}) = R_{N_{min}}$ using algebraic multigrid
3. Ascending leg of V-cycle
   for $N = \{2N_{min}, 4N_{min}...N_{max}\}$
      3a. Prolong correction from coarser level
         $\delta_N += P(\delta_{N/2})$
      3b. Smooth correction.
         $\delta_N = S(R)$
4. Correct the solution.
   $\phi_{N_{max}} += \delta_{N_{max}}$

Fig. 3 Algorithm for multigrid V-cycle. We repeat this iteratively until the residual reaches solution tolerance. S is the smoothing operator applied n_s times, A is the restriction (averaging) operator and P is the prolongation (interpolation) operator, N_{max} is the grid refinement at the finest level, N_{min} is the refinement at which we switch to algebraic multigrid.

and an improved solution $\phi^{l+1} = \phi^l + \delta^l$.

1.3.1 Smoothing

Our smoothing operator S is pointwise Gauss-Seidel relaxation with eight color ordering in three dimensions, four color ordering in two dimensions. See figure 5 for an illustration of the colors. This is required because of the stencil's corner coupling, but this coupling decays rapidly with distance, so 8- or 4-color approaches are sufficient but preferable to standard red-black ordering. We chose this over other techniques, such as block smoothers, because of the large stencil and variability near the boundary, which would reduce their effectiveness and increase memory overhead. Each GS smoothing step is performed through the whole domain for each color in succession and the correction is updated in place. Figure 4 illustrates the relaxation algorithm $\delta = S(R)$. The relaxation parameter $\lambda_\mathbf{i}$ is a function of space given by

$$\lambda_\mathbf{i} = 0.5/w_\mathbf{i}$$

where $w_\mathbf{i}$ is the diagonal coefficient of the Laplacian operator.

1.3.2 Restriction

We restrict the residual from finer to coarser levels using volume-weighted averaging. Given a coarser volume V_c which is created by the graph-connected coarsening of a set of fine volumes V_f, the restriction $R_{N/2} = A(R_N)$ operator

```
for (l = 1, n_s)
    for each color c ∈ C_D:
        Compute R = ρ − Lφ over whole domain.
        for (i ∈ Ω)
            if( ∀d ∈ {1, D}, (i_d mod (2)) == (c_d mod (2)))
                δ_i += λ_i(R_i − L(δ)_i)
```

Fig. 4 Algorithm for smoothing $\delta = S(R)$ at a given resolution. C_D is the set of colors for dimension D (see Figure 5). n_s is the number of smoothing steps.

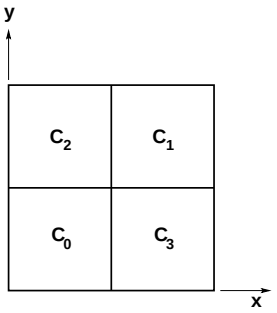

Fig. 5 Colors for GS relaxation. In two dimensions $C = \{(0,0),(1,1),(0,1),(1,0)\}$.

at $\mathbf{i}$ takes the form

$$R_{N/2,c} = A(R_N)_c = \frac{\sum_{v \in V_f} \kappa_v R_v}{\sum_{v \in V_f} \kappa_v}.$$

Because we are using cell averaged quantities, this restriction is exact if the fine values are exact, and the coarsened volume is defined as the sum of the fine. As mentioned in [22] (§2.7-8), multigrid for cell-average operators does not use the same restriction approach as nodal or Galerkin operators, but can be as effective with the correct combination of smoothing and higher-order prolongation operators.

1.3.3 Prolongation

Given a coarse volume V_c which is created by the graph coarsening of a set of fine volumes $\{v : V_{v,f} \in V_c\}$, our prolongation operator produces the increment to the fine correction interpolated from the coarse correction ($\delta_{f,N} += P(\delta_{N/2})_f$). Previous investigations (for example [21]) use piecewise-constant prolongation ($P(\delta_{N/2})_f = \delta_{N/2}$), which still works well because high-frequency errors from this interpolation are quickly damped by the smoother, S. However for higher order operators, the accuracy of the prolongation scheme can affect multigrid convergence [7,22].

To try to quantify this effect for our cut-cell discretization, suppose we wish to have a prolongation operator of accuracy $O(h^Q)$. Using equation (8),

we express the correction at a coarse cell as local polynomial with coefficients C. Integrating over the volume, we get

$$\langle \delta \rangle_v = \frac{1}{m_v^{\mathbf{z}}} \sum_{|\mathbf{q}|<Q} c_{\mathbf{q}} m_v^{\mathbf{q}} \tag{9}$$

Using all the coarse volumes within our stencil (see Figure 1), we make a system of equations using equation (9) as each element of the system. All the moments are relative to a common $\mathbf{x}_0$ (the center of the target cell). This whole process is identical to how the polynomial coefficients are calculated from (8) in [12]. We then use these coefficients, along with the moments for each fine cell to calculate $\langle \delta \rangle_f$.

1.3.4 Coarse-fine boundaries

At coarse-fine AMR boundaries, a similar operator is used to set "ghost cell" values that are needed by the fine-grid operator. Because these take into consideration both the change in grid spacing and the cut cell boundary, the domain boundary may cross refinement levels. To maintain conservation between levels, coarse cell face fluxes are defined as the sum of fine-grid fluxes, as in [15]; coarse-grid residuals are updated with these fine flux values when going down the V-cycle in geometric multigrid.

2 Results

We define our multigrid convergence rate C at V-cycle iteration l to be the ratio between the former cycle's residual max-norm and the current cycle's:

$$C^l = \frac{||R||^l}{||R||^{l-1}}.$$

where $||R||^l$ is the L_∞ norm of the residual at iteration l. We define C^{eff} to be the convergence rate into which multigrid settles; this settling typically takes 3-7 cycles. We investigate how convergence rate and time to solution are changed by three solver parameters: the order of prolongation, the number of relaxation iterations, and the coarsest resolution of the V-cycle. We also study the performance implications of using adaptive meshes. Considering the interactions between these parameter choices, for each geometry and resolution, we only explore a few of these combinations. The present sample problems are intended to isolate and illustrate the individual parameter choice effects, without having to exhaustively explore the parameter space.

Prolong Order	C^{eff}	Time to solution
0	1./2.2	1.22
1	1./4.2	0.67
2	1./3.8	0.68
3	1./4.4	0.66
4	1./4.4	0.60
AMG	-	2.68

Table 1 Multigrid convergence rates for the regular grid of 128x128. Two smoothings were used, and we switched to algebraic multigrid when the V-cycle reached 16x16. Using only algebraic multigrid takes 2.68 seconds to solve.

Prolong Order	C^{eff}	Time to solution
0	1./7.7	1.02
1	1./10.4	0.90
2	1./11.5	0.76
3	1./11.8	0.75
4	1./11.8	0.74
AMG	-	2.47

Table 2 Multigrid convergence rates for the domain $\{(x, y) \in [0, 1]^2 : (x-0.5)^2+(y-0.5)^2 \geq 0.21^2\}$. For the V-cycle, eight smoothings were used and the V-cycle was stopped at 16x16. Using only algebraic multigrid, the time to solution is 2.47 seconds.

2.1 Prolongation Order Effects

First, we demonstrate that a higher-order prolongation operator can improve multigrid convergence for problems with embedded boundaries.

To show this, we baseline the effect of prolongation order in the absence of embedded boundaries. We solve Poisson's equation on $[0, 1]^2$ with homogeneous Dirichlet boundary conditions and a uniform charge density $\rho = 1$. Table 1 shows multigrid convergence rates and time to solution as the prolongation order P is varied between 0 and 4 for a regular single-level grid of size 128×128. For the V-cycle, the number of smoothings was set to two, and algebraic multigrid was used once the V-cycle had coarsened the grid to $h = 1/16$. Although convergence rates are not ideal, linear prolongation ($P = 1$) shows much better rates than piecewise-constant prolongation ($P = 0$) for this higher-order stencil.

Improvement due to higher-order prolongation ($P > 1$) is much more modest; this is in agreement with our experience with lower-order finite volume operators as in [11]. These higher convergence rates also translate to better solution times as P increases.

Next we solve the above problem (Poisson's equation with homogeneous Dirichlet boundary conditions and the uniform charge density $\rho = 1$) on the domain $\{(x, y) \in [0, 1]^2 : (x - 0.5)^2 + (y - 0.5)^2 \geq 0.21^2\}$ to investigate t effect of prolongation order for an example embedded boundary calculation. The grid is single-level and of size 128×128. Eight smoothings were used for

Number of smoothings	C^{eff}	Time to solution
2	1./1.8	1.08
3	1./2.5	0.81
4	1./3.4	0.76
5	1./4.6	0.74
6	1./6.3	0.71
7	1./8.6	0.68
8	1./11.7	0.74
AMG	-	2.38

Table 3 Multigrid convergence rates for the domain $\{(x,y) \in [0,1]^2 : (x-0.5)^2+(y-0.5)^2 \geq 0.21^2\}$. For the V-cycle, fourth-order prolongation was used, and the V-cycle was stopped at 16x16. The AMG time to solution is 2.38

the V-cycle, and algebraic multigrid was used when the V-cycle hit a grid refinement of $h = 1/16$.

Table 2 shows the multigrid convergence rates and time to solution for this example for prolongation orders of $P = 0...4$. The convergence rates behave similarly to the regular grid case: increasing the order P from 0 to 1 greatly improves the convergence rate (and hence the time to solution), but the convergence rates only improve slightly as P is increased further.

2.2 Relaxation effects

Now we investigate how much increasing the number of smoothings n_s during relaxation improves multigrid convergence. This is a well-known result for grids without embedded boundaries. Here, we verify this fact for our higher-order method with embedded boundaries.

Similar to the prolongation study, we solve Poisson's equation on the domain $\{(x,y) \in [0,1]^2 : (x-0.5)^2 + (y-0.5)^2 \geq 0.21^2\}$. with homogeneous Dirichlet boundary conditions and the uniform charge density $\rho = 1$. Algebraic multigrid is used when the V-cycle reaches a refinement of $h = 1/16$. In this study, we use a fourth-order prolongation operator.

Table 3 shows how multigrid convergence improves with increased n_s. Time to solution, however, initially decreases with n_s due to higher convergence rate but starts to increase because of the extra work required per V-cycle. In all cases, the time to solution was less than half the time to solution of algebraic multigrid (AMG).

2.3 V-Cycle depth effects

We now investigate how the resolution of the coarsest level in the V-cycle (N_{min} in Figure 3) influences multigrid convergence. Previous experience suggests that the grid size at the coarsest resolution must be fine enough to minimally resolve the geometric features of the domain for multigrid to converge.

N_{min}	ζ	C^{eff}	Time to solution
16	0.4	DNC	–
32	0.8	DNC	–
64	1.6	1./5.4	5.28
128	3.2	1./5.4	6.59
AMG	-	-	8.62

Table 4 Typical multigrid convergence rates with geometry described in Figure 6 on a grid of 256x256. "DNC" signifies that multigrid did not converge. The space between boundaries, δ, is fixed at $\delta = 0.025$. The parameters are fixed at $r = 0.225$ and $W = 1$. The AMG time to solution is 8.62. Eight smoothing iterations and fourth-order prolongation were used.

Our geometric description for this example is shown in Figure 6 (the domain is the region outside of the circles). The domain width W equals 1. The charge density is set to $\rho = 1$, and homogeneous Dirichlet boundary conditions are used on the boundaries. The grid is single-level with size 256×256.

To get multigrid convergence, the distance δ between the boundaries must be resolvable by the coarsest domain grid spacing $h_{min} = 1/N_{min}$. We demonstrate this through two studies.

In our first study, we fix $\delta = 0.025$, and vary N_{min}, the size of the coarsest level in the V-cycle. In the second study, we fix $N_{min} = 16$ and vary δ (by varying the radius r of the circles). In both studies, the V-cycle uses a fourth-order prolongation operator with 8 smoothings.

The ratio

$$\zeta = \frac{\delta}{h_{min}}$$

measures how well the geometry is resolved on the coarsest level of the V-cycle. When $\zeta < 1$, the geometry is not well-resolved. Table 4 shows the multigrid convergence rates and time to solution in the first study. Table 5 shows the multigrid convergence rates and time to solution in the second study. These tables verify that multigrid doesn't converge for $\zeta < 1$, but has reasonable convergence for larger ζ. Moreover, multigrid convergence improves when the geometry is better resolved. For this geometry and discretization, this indicates that a minimum grid spacing of the features is required for geometric multigrid to converge.

Table 5 also compares the time to solution for algebraic multigrid versus our hybrid multigrid method. These results suggest that sometimes algebraic multigrid may reduce the time to solution slightly.

2.4 Adaptivity and Performance

In this section, we present some converged solutions as examples of how adaptivity can affect multigrid performance. The geometry and charge distribution are described in figure 7. Homogeneous Dirichlet boundary conditions are prescribed on the boundaries.

δ	ζ	C^{eff}	Time to solution	AMG time to solution
0.025	0.4	DNC	–	8.54
0.0625	1.0	DNC	–	10.33
0.1	1.6	1./13.3	7.90	9.25
0.15	2.4	1./13.3	9.62	7.95
0.2	3.2	1./34.7	10.24	8.77

Table 5 Typical multigrid convergence rates with geometry described in Figure 6 on a grid of 256x256. "DNC" signifies that multigrid did not converge. The space between boundaries is denoted δ. $N_{min} = 16$, 8 smoothings were used and fourth-order prolongation was used.

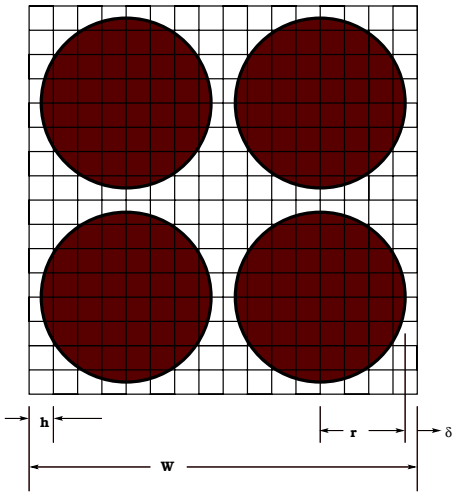

Fig. 6 Geometric description for V-cycle depth investigation.

In our study, we fix the finest level of the AMR grid to be 512×512, and we vary N_c, the resolution of the coarsest AMR level. In each example, the V-cycle uses fourth order prolongation, eight smoothings, and algebraic multigrid is used when the V-cycle coarsened resolution is $h = 1/16$.

Table 6 shows the multigrid convergence rates and time to solution as we vary N_c. The convergence rates are constant between the examples, suggesting that the number of levels do not affect the convergence rate. The time to solution for one and two levels is much higher than that for three or four levels. Surprisingly, the time to solution for the run with $N_c = 64$ is higher than the time to solution for the run with $N_c = 128$. This is because the finest grid when $N_c = 64$ covers more of the domain than the finest grid for $N_c = 128$ (see Figures 8 and 9). We use the grid generation algorithm described in [5]. This algorithm is complex and the number of points it produces is difficult to predetermine.

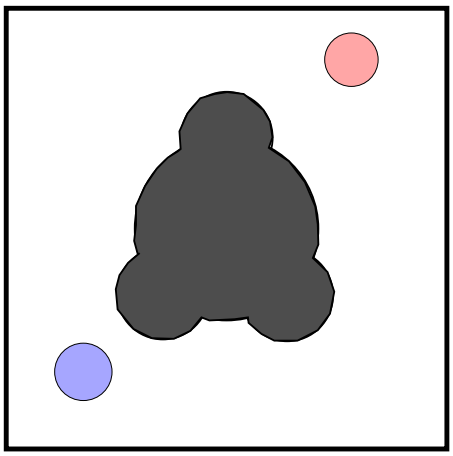

Fig. 7 Geometric description for the AMR investigation.The computational domain is the region in white. The embedded boundary is the intersection of four circles. One circle has center (1, 1) and radius 0.2. The other three circles have centers (1.1732, 0.9), (0.8268, 0.9), and (1, 1.2), and all three have radii of 0.1. The charge density consists of two circular uniformly distributed charges (shown in red and blue). The positive charge distribution (in red) has center (1.75, 1.75), radius 0.1, and a total charge of 10. The negative charge distribution (in blue) has center (0.2, 0.2), radius 0.1, and a total charge of -10.

N_c	C^{eff}	Time to solution
64	1./1.9	4.77
128	1./1.9	3.96
256	1./1.9	11.05
512	1./1.9	37.15

Table 6 Multigrid convergence rates for the AMR example described in §2.4. The finest AMR level is 512x512. The base resolution is $N_c x N_c$. The V-cycle used 8 smoothings, and the V-cycle switched to algebraic multigrid at 16x16. We used fourth order prolongation.

3 Conclusions

We have presented an adaptive mesh multigrid algorithm for solving Poisson's equation using a fourth-order Cartesian grid, embedded boundary discretization. We show that higher order prolongation operators and more smoothings can both help multigrid convergence, especially when portions of the boundary are under-resolved on coarser geometric multigrid levels. We also show that the V-cycle convergence is best when the transition from geometric multigrid to an algebraic coarsening approach is done at a resolution that captures significant boundary features.

Immediate future work will include other operators, such as Helmholtz and variable-coefficient elliptic operators. An extension of this work to include jumps (as in [11]) on the interior of the domain will be important for equations in conservation form spanning multiple materials. Further optimization of the algorithm may include smoothing operators that focus additional effort

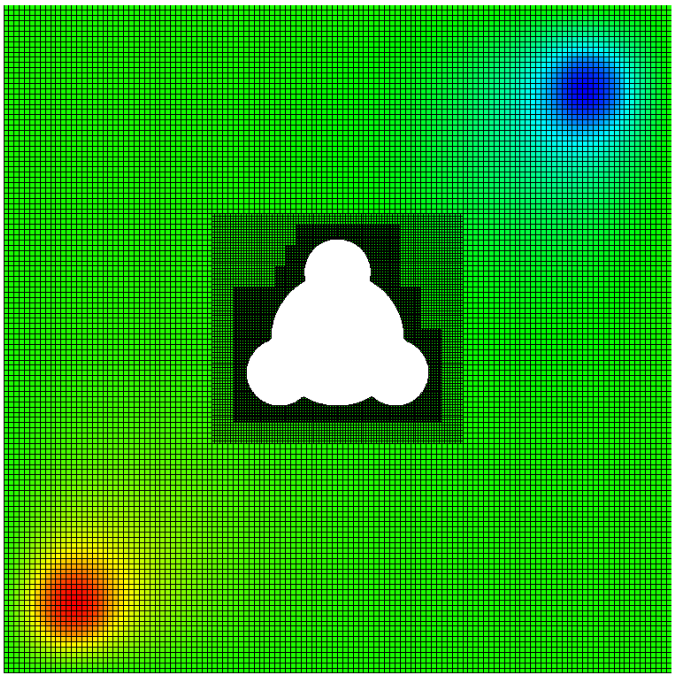

Fig. 8 Grid configuration and solution for the adaptivity study when the coarsest AMR level $N_c = 128$.

near the domain boundary, but use inexpensive regular grid smoothers in the interior of the domain.

References

1. Balay, S., Abhyankar, S., Adams, M.F., Brown, J., Brune, P., Buschelman, K., Eijkhout, V., Gropp, W.D., Kaushik, D., Knepley, M.G., McInnes, L.C., Rupp, K., Smith, B.F., Zhang, H.: PETSc users manual. Tech. Rep. ANL-95/11 - Revision 3.5, Argonne National Laboratory (2014). URL http://www.mcs.anl.gov/petsc
2. Balay, S., Abhyankar, S., Adams, M.F., Brown, J., Brune, P., Buschelman, K., Eijkhout, V., Gropp, W.D., Kaushik, D., Knepley, M.G., McInnes, L.C., Rupp, K., Smith, B.F., Zhang, H.: PETSc Web page (2014). URL http://www.mcs.anl.gov/petsc
3. Balay, S., Gropp, W.D., McInnes, L.C., Smith, B.F.: Efficient management of parallelism in object oriented numerical software libraries. In: E. Arge, A.M. Bruaset, H.P. Langtangen (eds.) Modern Software Tools in Scientific Computing, pp. 163–202. Birkhäuser Press (1997)
4. Berger, M.J., Aftosmis, M.J., Allmaras, S.: Progress towards a cartesian cut-cell method for viscous compressible flow. AIAA Paper 2012 **1301** (2012)
5. Berger, M.J., Rigoutsos, I.: An algorithm for point clustering and grid generation. IEEE Transactions Systems, Man, and Cybernetics **21**(5), 1278–1286 (1991)
6. Botto, L.: A geometric multigrid poisson solver for domains containing solid inclusions. Computer Physics Communications **184**(3), 1033 – 1044 (2013)
7. Briggs, W.L.: A Multigrid Tutorial. SIAM, Philadelphia, PA (1987)

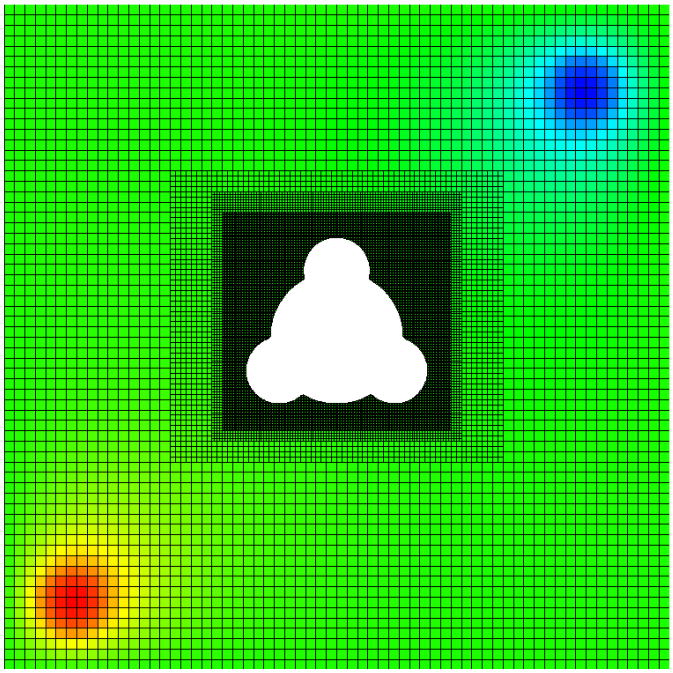

Fig. 9 Grid configuration and solution for the adaptivity study when the coarsest AMR level $N_c = 64$.

8. Coco, A., Russo, G.: Finite-difference ghost-point multigrid methods on cartesian grids for elliptic problems in arbitrary domains. Journal of Computational Physics **241**(0), 464 – 501 (2013)
9. Colella, P., Graves, D.T., Ligocki, T.J., Martin, D.F., Modiano, D., Serafini, D.B., Straalen, B.V.: Chombo Software Package for AMR Applications - Design Document. Tech. Rep. LBNL-6616E, LBNL (2014). URL http://chombo.lbl.gov/
10. Colella, P., Graves, D.T., Ligocki, T.J., Miller, G., Modiano, D., Schwartz, P., Straalen, B.V., Pillod, J., Trebotich, D., Barad, M.: Ebchombo software package for Cartesian grid, embedded boundary application. Tech. Rep. LBNL-6615E, LBNL (2014). URL http://chombo.lbl.gov/
11. Crockett, R., Colella, P., Graves, D.: A Cartesian grid embedded boundary method for solving the Poisson and heat equations with discontinuous coefficients in three dimensions. Journal of Computational Physics **230(7)**, 2451–2469 (2010)
12. Devendran, P., Johansen, H., Graves, D.T.: A higher-order finite-volume discretization method for Poisson's equation in cut-cell geometries. arXiv, to be submitted (2015). URL http://arxiv.org/pdf/1411.4283v1.pdf
13. Guillet, T., Teyssier, R.: A simple multigrid scheme for solving the poisson equation with arbitrary domain boundaries. Journal of Computational Physics **230**(12), 4756 – 4771 (2011)
14. Ma, Z.H., Qian, L., Causon, D.M., Gu, H.B., Mingham, C.G.: A cartesian ghost-cell multigrid poisson solver for incompressible flows. International Journal for Numerical Methods in Engineering **85**(2), 230–246 (2011)
15. Martin, D.F., Cartwright, K.L.: Solving Poisson's equation using adaptive mesh refinement. Technical Report UCB/ERI M96/66 UC Berkeley (1996)
16. Miller, G.H., Trebotich, D.: An embedded boundary method for the Navier-Stokes equations on a time-dependent domain. Communications in Applied Mathematics and Computational Science **7**, 1–31 (2012)

17. Schwartz, P., Barad, M., Colella, P., Ligocki, T.: A Cartesian grid embedded boundary method for the heat equation and Poisson's equation in three dimensions. Journal of Computational Physics **211**(2), 531–550 (2006)
18. Schwartz, P., Percelay, J., Ligocki, T., Johansen, H., Graves, D.T., Devendran, P., Colella, P., Ateljevich, E.: High accuracy embedded boundary grid generation using the divergence theorem (2014). URL http://crd.lbl.gov/groups-depts/ANAG/publications/. Submitted to CAMCoS
19. Steefel, C., Molins, S., Trebotich, D.: Pore scale processes associated with subsurface co2 injection and sequestration. Reviews in Mineralogy and Geochemistry (2013)
20. Trebotich, D., Graves, D.T.: An adaptive Cartesian grid embedded boundary method for the incompressible Navier-Stokes equations. Submitted to CAMCoS (2014). URL http://crd.lbl.gov/groups-depts/ANAG/publications/
21. Trebotich, D., Straalen, B.V., Graves, D.T., Colella, P.: Performance of embedded boundary methods for CFD with complex geometry. presented at SciDAC 2008 (2008)
22. Trottenberg, U., Oosterlee, C., Schuller, A.: Multigrid. Academic Press (2000). ISBN: 978-0-12-701070-0