Optimization of Strength Threshold in Algebraic Multigrid for Immersed Interface Problems

Miranda J. Antonelli * Timothy P. Chartier †

May 13, 2003

Abstract

Computational complexity versus accuracy play a fundamental role in the efficacy of an iterative algorithm. In [1], AMG's robustness for interface problems that have been discretized using the methods described in [3] and in [7] for elliptic interface problems using the maximum principle preserving schemes was demonstrated. This paper conducts a parameter study on the strength threshold parameter that is a cornerstone to the AMG algorithm developed by Ruge and Stüben in [8]. The value of the strength threshold in the AMG algorithm directly relates to the speed of the method. This paper analyzes the performance of AMG on various interface problems and improves on the efficiency of AMG as compared to the results in [1].

1 Introduction

Algebraic Multigrid (AMG) is a variant of Geometric Multigrid (MG) that abstracts the multigrid process resulting in a black–box iterative method. AMG has been successfully applied to numerous applications including problems involving irregular and highly unstructured grids. The robustness and scalability of AMG were carefully studied in [6]. In [1], AMG was successfully applied to interface problems that have been discretized using the methods described in [3] and in [7] for elliptic interface problems using the maximum principle preserving schemes.

Fundamental to multigrid is the use of *multi*ple *grid*s. AMG (and MG) choose the coarse grid to be a subset of the fine grid. A rich coarse grid leads to a high degree of accuracy in the overall multigrid algorithm. Yet, it is important to balance the need for a rich coarse grid with low computational cost. In AMG, the size of the coarse grid (which directly affects the accuracy and complexity of the algorithm) is controlled by the strength threshold parameter, α (see Section 3 for more information on this parameter.) This paper carefully examines how to improve the efficiency of AMG on the interface problems examined in [1].

The paper is organized as follows. Section 2 reviews several fundamentals of multigrid emphasizing why the strength threshold parameter is a cornerstone to the AMG algorithm developed by Ruge and Stüben in [8]. Section 3 poses the test problem. Note this is identically the same problem studied in [1]. Section 4 discusses the input parameters (most notably α) that are varied in the numerical experiments. Section 5 develops the metrics that are used to measure the performance of the AMG algorithm. Section 6 contains the numerical results, as well as an analysis of the relative merits of different values for α. Finally, Section 7 summarizes the findings of this paper.

*Undergraduate student in the Department of Applied Computational Mathematical Sciences, University of Washington, Seattle, Washington 98195.

†Department of Mathematics, University of Washington, Seattle, Washington 98195. This work was supported by NSF VIGRE grant DMS-9810726.

2 AMG and Strength of Connection

Geometric and algebraic multigrid methods seek the solution $\mathbf{u} \in \mathbb{R}^n$ to the linear system

$$A\mathbf{u} = \mathbf{f}, \tag{1}$$

where A is an $n \times n$ matrix. The linear systems that will be studied in this paper arise from problems with internal interfaces that have been discretized with Cartesian grids using the methods described in [3] and in [7] for elliptic interface problems using the maximum principle preserving schemes.

Since the value of the strength threshold in the AMG algorithm directly relates to the speed of the algorithm to solve a linear system, the parameter study to follow will focused entirely on altering the strength threshold parameter, α, in AMG. Strength of connection plays a fundamental role in the construction of the necessary multigrid components and consequently, in the efficiency of the AMG algorithm. Since changes in α change the multigrid components, we take a moment for a brief discussion on the components of a multigrid algorithm.

2.1 Components of Algebraic Multigrid

In order to solve (1) efficiency, AMG algorithms must effectively and, by design, automatically construct all needed components of the multigrid iterative process. Using superscripts $1 \dots M$ to indicate level number, with 1 denoting the finest level ($A^1 = A$ and $\Omega^1 = \Omega$ for the finest grid), the required components are as follows:

1. Algebraic "Grids": $\Omega^1, \Omega^2, \dots, \Omega^M$.
2. Grid operators: $A^1, A^2, \dots, A^M$.
3. Grid transfer operators:
   Interpolation $I_{i+1}^i, i = 1, 2, \dots, M-1$,
   Restriction $I_i^{i+1}, i = 1, 2, \dots, M-1$.
4. Relaxation scheme for each level.

Given that AMG requires only A and $\mathbf{f}$ in (1), the grid points and grid itself are algebraic structures associated with A rather than a geometric entity. More specifically, in the linear system (1), AMG's grid points correspond simply to the indices of the vector $\mathbf{u}$. The grid, per se, in AMG is defined as the undirected adjacency graph of the matrix A.

With MG, a serious attempt is made for the error left by relaxation to be geometrically smooth. With AMG, the approach differs in part as there is no physical grid or sense of geometry. Instead, AMG uses, not surprisingly, an algebraic definition of smoothness.

2.1.1 Algebraically Smooth Error and Strong Connections

AMG's effectiveness is dependent on its ability to design coarse-grid correction to dampen components of the error that remain after relaxation. These error components are not rapidly reduced by relaxation and are collectively called *algebraically smooth error*. AMG formulates algebraically smooth error as that which produces relatively small residuals, written loosely as

$$A^h\mathbf{e}^h \approx 0 \Rightarrow a_{ii}e_i^h \approx -\sum_{j \neq i} a_{ij}e_j^h, \tag{2}$$

where a_{ij} are entries of A^h, $\mathbf{e}$ is the error and the indices are identified with grid points. Note that if a_{ij} is relatively small, then e_i^h can be substituted for e_j^h and (2) would still hold.

In a sense, relatively small a_{ij} means that e_i^h does not depend critically on the value of e_j^h. In this case, we say loosely that point i does not depend strongly on point j, or that point i is *weakly connected* to point j. Alternatively, for k such that a_{ik} is relatively large, the value of e_i^h does depend

on the value of e_k^h, and we say that point i is *strongly connected* to point k. This leads to the following definition of the set of strong connections of point i:

$$S_i := \{j \neq i : -a_{ij} \geq \alpha \max_{k \neq i}(-a_{ik})\}. \tag{3}$$

In practice, α is usually set to 0.25. The purpose of this research is to determine if 0.25 produces optimal performance for AMG on interface problems.

The following heuristic motivates the choice of the coarse grid and the interpolation operator:

H1: *Algebraically smooth error varies slowest in the direction of strong dependence.*

Note that **H1** relies on the definition of strong dependence. Again, strong dependence is determined by S_i, which is directly effected by the magnitude of α. In AMG, the composition of a coarse grid is tightly bound to the definition of strong connection. For more details on the coarse-grid selection algorithm, see [8] or [1].

Also, the effectiveness of the interpolation operator I_{2h}^h, which defines the ith component of $I_{2h}^h \mathbf{e}^{2h}$, is coupled with the definition of strong connection. To see this, note that $I_{2h}^h \mathbf{e}^{2h}$, the approximation to $\mathbf{e}^h$ in (2), is defined as

$$(I_{2h}^h \mathbf{e}^{2h})_i = \begin{cases} e_i^{2h} & \text{if } i \in C, \\ \sum_{j \in C_i} w_{ij} e_j^{2h} & \text{if } i \in F, \end{cases} \tag{4}$$

where $C_i = S_i \cap C$. Here again, S_i plays a fundamental role in the construction of a multigrid component.

Thus, it is no surprise that changing α (in (3)) can dramatically alter the AMG algorithm with the potential of improving or deteriorating the effectiveness of the method. The goal of this paper is to determine if more optimal choices of α (in contrast to the choice of $\alpha = 0.25$) are possible for the interface problems studied in [1].

3 The Test Problem

The numerical results to follow focus on the specific body of problems studied by Adams and Chartier in their recent work on multigrid methods on interface problems found in [2] and [1]. More specifically, this paper attempts to improve on the results of AMG in [1].

These problems stem from the elliptic equation

$$\nabla \cdot (\beta(x,y) \nabla u(x,y)) = f(x,y), \tag{5}$$

where the interface is the circle $x^2 + y^2 = \frac{1}{4}$ within the computation domain $-1 \leq x, y \leq 1$. The value of β is

$$\beta(x,y) = \begin{cases} x^2 + y^2 + 1, & \text{if } x^2 + y^2 \leq \frac{1}{4}, \\ b & \text{if } x^2 + y^2 > \frac{1}{4}, \end{cases}$$

and the source term is

$$f(x,y) = 8(x^2 + y^2) + 4.$$

The true solution is

$$u(x,y) = \begin{cases} r^2, & \text{if } r \leq \frac{1}{2} \\ (1 - \frac{1}{8b} - \frac{1}{b})/4 + (\frac{r^4}{2} + r^2)/b + C \log(2r)/b, & \text{if } r > \frac{1}{2}, \end{cases}$$

which can allow for more insightful analysis of our numerics.

This problem has variable and discontinuous coefficients with the following jump conditions at the interface:

$$[u] = 0, \quad [\beta u_n] = 2C, \quad [\beta] = b - 5/4, \quad [f] = 0$$

Note that smaller values of b relate to more difficult problems since the jump in the normal derivative of the solution is given by

$$[u_n] = (2C + 5/4)/b - 1,$$

and the ratio of the β values at the interface is $1.25/b$. Finally, Dirichlet boundary conditions are imposed on the problem.

4 Input and Comparison Variables

Three parameters will be varied in the numerical tests. The most obvious is α. The problem parameter b, and the initial grid size, as measured by n will also be varied in order to test scalability on problems of increasing difficulty and refinement, respectively.

4.1 Strength Threshold Coefficient

The strength threshold coefficient α plays a fundamental role in coarse-grid selection and interpolation. Denser coarse grids result from larger α values. Denser coarse grids correlate to denser coarse-grid operators and higher computational complexity. Yet, sparse coarse grids eventually correlate to poor convergence rates.

The numerics in [1] used $\alpha = 0.25$ within AMG for the immersed interface problems. This value was chosen arbitrarily, in a certain sense, as a compromise between these two competing effects. The numerics even with this value reflected an efficient and robust application of AMG. Yet, as we will see, a closer study of this parameter can yield improved efficiency of the iterative process.

4.2 Other Independent Variables

In addition to α, two other parameters are varied. Unlike α, these parameters result in changes to the underlying test problem, as opposed to variations in the algorithm.

The first of these values, b, is part of the test problem itself. Specifically, it changes the behavior of the β term in (5). It is important to recall that this problem becomes more difficult as b (and thus β) decreases toward zero, as explained in Section 3.

The second independent variable, n, is a measure of the fineness of our initial grid. A small n represents a coarser grid (not to be confused with coarse grids in the multigrid process) and a smaller problem. A larger n corresponds to a more refined grid and generally a more difficult computational problem. The size of the problem is proportional to n^2, not n itself, meaning that setting n to 512 translates to a problem that is 256 times larger than a problem where n equals 32. For smaller values of b, problems with smaller values of n are not sufficient to capture the intricacies of parameter differences. This is mainly due to the speed at which AMG solves such small problems for a large range of α values. Consequently, problems with smaller b values will have fewer trials in the numerical tests in Section 6.

5 Output and Calculated Variables

5.1 Convergence Factor and Complexity

Five different output variables were measured from AMG. Each measures the efficiency of AMG from a different perspective. The first measure is the number of cycles (more specifically, V–cycles) necessary to convergence to a solution. The number of V–cycles is closely intertwined with the second measured variable, convergence factor. Convergence factor, f, is the ratio $(\|\mathbf{s}_k\|_2/\|\mathbf{s}_{k-1}\|_2)$ for the last iteration k where $\mathbf{s}_k = D_o^{-1}\mathbf{f} - A_s\mathbf{v}_k$, also called the scaled residual in [1]. (See [1] for a more detailed discussion

regarding $\mathbf{s}$.) Note that if $f = 0.1$, then the residual at k is one tenth the residual at step $k-1$. This is an effective measure of the efficacy of the cycling.

The final three variables measure complexities within the algorithm. A-Complexity, C_A, measures operator complexity as the ratio of the non-zero elements contained in the operators on all levels to number of non-zero elements in the operator on the finest grid. Grid complexity is measured by the Ω-Complexity, C_Ω, which is the ratio of the number of grid points on all levels to the number of grid points on the finest grid. Finally, S-Complexity, C_S, measures space (memory) complexity, which is measured by the ratio of the storage used by AMG on all levels to the storage required only for the problem on the finest grid.

Let n_Ω^i and n_S^i denote the number of grid points and amount of storage necessary on level i, respectively. Let n_L^i denote the number of non-zero entries in the level i operator A^i. Then alternatively,

$$C_A = \frac{\left(\sum_{i=1}^{n_l} n_L^i\right)}{n_L^1}, \quad C_\Omega = \frac{\left(\sum_{i=1}^{n_l} n_\Omega^i\right)}{n_\Omega^1}, \quad C_S = \frac{\left(\sum_{i=1}^{n_l} n_S^i\right)}{n_S^1}.$$

Clearly, C_A, C_Ω and C_S are always greater than 1.

5.2 Effective Convergence: ρ Factors

The three complexities are combined with f to create our comparison metrics. We obviously would like to have an algorithm with low computational overhead (low A, Ω and S complexities), but efficiency in terms of accuracy (low number of V–cycles and low convergence factor) is also desirable. These two effects are combined into a new variable, the effective convergence ρ. We calculate three different ρ values, one for each of the complexities, by taking:

$$\rho_A = f^{(1/C_A)} \quad \rho_\Omega = f^{(1/C_\Omega)} \quad \rho_S = f^{(1/C_S)}$$

Since the various complexities will always be greater than one, the smaller the effective convergence (regardless of the associated complexity) the more desirable is the associated algorithm.

6 Numerical Results

This section gives numerical results of the AMG method using AMG1R6 written by Ruge, Stüben and Hempel with version date 1997. Each experiment chooses $\mathbf{u}^0 = \mathbf{0}$ on the interior and sets the boundary nodes to their value as given in the Dirichlet boundary conditions. AMG is supplied with only the fine-grid matrix and right-hand side.

AMG uses a V(2,2) cycle, that is, two pre- and post- Gauss-Seidel smoothing steps. The stopping tolerance is $||\mathbf{s}_k||_2 < 10^{-6}$ for all problems. The AMG method was also run to enforce the coarsest grid be no smaller than 64 unknowns in order to match numerics contained in [1].

6.1 Comparing the Strength Thresholds for Individual Problems

This section focuses on numerical results of α comparisons for each trial (that is, for each b and n combination.) Each of the three ρ factors provides a slightly different perspective on the complexity of AMG as applied to the problem. Still, all the measures lead to similar conclusions. In all, 11 different trials, spanning five different values for b were run, and in all but two, the ideal α was the same for $\rho_A, \rho_\Omega,$ and ρ_S.

For each complexity, the corresponding table lists the ideal α, its performance (ρ and number of iterations) as well as the performance of $\alpha = 0.25$. Also, Δ denotes the percent improvement in ρ for the ideal as compared to $\alpha = 0.25$.

b	n	Ideal α	Ideal Cycles	0.25 Cycles	Ideal ρ_A	0.25 ρ_A	Δ
1.25	32	NA [1]	NA	3	NA	0.1193	NA
1.25	64	0.41	3	3	0.1244	0.1265	1.7%
1.25	128	0.41	3	3	0.1190	0.1230	3.2%
1.25	256	0.41	3	3	0.1145	0.1193	4.0%
0.10	64	0.13	4	6	0.2859	0.4080	29.9%
0.10	128	0.15	4	6	0.2934	0.4700	37.6%
0.10	256	0.15	4	6	0.3884	0.4659	16.6%
0.005	128	0.14	9	9	0.5395	0.5550	2.8%
0.005	256	0.15	8	10	0.5462	0.5719	4.5%
0.0005	128	**0.25**	NA	13	NA	0.5853	0.0%
0.0005	256	0.22	11	12	0.5689	0.6016	5.4%
0.00005	256	0.11	13	15	0.5807	0.6233	6.8%

Table 1: Operator Complexity (C_A) Results

Several observations are readily evident from Table 1. First, in only one instance was $\alpha = 0.25$ the ideal value. This suggests that a different value of α may indeed serve as a better default for interface problems.

Second, although not ideal, $\alpha = 0.25$ was highly inefficient only for $b = 0.10$. Also, the ideal α is above and below 0.25 depending on the problem. This suggests the possibility that there may be a critical point in the b values, with respect to the ideal α.

Tables 2 and 3 for C_Ω and C_S follow. The conclusions from the tables are essentially the same as with Table 1. We pause only to note that the percentage gains for C_Ω values were consistently higher than for C_A and C_S, suggesting that the ideal α would be especially valuable if grid complexity is the emphasis.

b	n	Ideal α	Ideal Cycles	0.25 Cycles	Ideal ρ_Ω	0.25 ρ_Ω	Δ
1.25	32	NA	NA	3	NA	0.0518	NA
1.25	64	0.41	3	3	0.0580	0.0590	1.7%
1.25	128	0.41	3	3	0.0572	0.0595	3.8%
1.25	256	0.41	3	3	0.0556	0.0586	5.0%
0.1	64	0.13	4	6	0.1557	0.2876	45.8%
0.1	128	0.15	4	6	0.1709	0.3496	51.1%
0.1	256	0.15	4	6	0.2645	0.3541	25.3%
0.005	128	0.14	9	9	0.4095	0.4331	5.5%
0.005	256	0.15	8	10	0.4184	0.4615	9.3%
0.0005	128	0.25	-	13	-	0.4737	0.0%
0.0005	256	0.22	11	12	0.4472	0.4802	6.9%
0.00005	256	0.16	13	15	0.4592	0.5067	9.4%

Table 2: Grid Complexity (C_Ω) Results

b	n	Ideal α	Ideal Cycles	0.25 Cycles	Ideal ρ_S	0.25 ρ_S	Δ
1.25	32	NA	NA	3	NA	0.2503	NA
1.25	64	0.41	3	3	0.2078	0.2135	2.7%
1.25	128	0.41	3	3	0.1914	0.1969	2.8%
1.25	256	0.42	3	3	0.1829	0.1896	3.5%
0.1	64	0.13	4	6	0.3675	0.5216	29.5%
0.1	128	0.15	4	6	0.3744	0.5501	31.9%
0.1	256	0.15	4	6	0.4692	0.5470	14.2%
0.005	128	0.14	9	9	0.6083	0.6253	2.7%
0.005	256	0.15	8	10	0.6132	0.6414	4.4%
0.0005	128	0.25	-	13	-	0.6536	0.0%
0.0005	256	0.22	11	12	0.6354	0.6611	3.9%
0.00005	256	0.11	13	15	0.6452	0.6807	5.2%

Table 3: Space Complexity (C_S) Results

6.2 Effect of Strength Threshold on Larger b Values

For larger b (recall smaller b correlates to more difficult problems), the choice of α had only a small impact on the effective convergence. In Table 4, no value between 0.10 and 0.45 degrades performance by more than 6% in any category for $b = 1.25$ and $n = 128$.

Note that the ideal in this run occurred at $\alpha = 0.41$, but there was a second minima in the α versus ρ plots at $\alpha = 0.17$, corresponding to only a 0.9% decrease in efficiency. These two facts prevent the relatively high ideal α found for $b = 1.25$ from exerting a large influence on our conclusions regarding an optimal α across problem types in Section 6.3

Also note the rapid degradation of inefficiency as α increases beyond the ideal value. This can be seen in all of the problem types.

α	C_A	C_Ω	C_O
0.10	4.58%	5.73%	3.82%
0.15	1.48%	1.78%	1.09%
0.20	3.04%	3.70%	2.56%
0.25	3.28%	3.98%	2.85%
0.30	3.44%	4.22%	2.90%
0.35	2.65%	3.04%	2.57%
0.40	0.59%	0.79%	0.46%
0.45	0.39%	0.33%	0.56%
0.50	18.33%	19.57%	11.56%

Table 4: $b = 1.25$, $n = 128$

6.3 Optimum α for over the Immersed Interface Problems

Section 6.1 confirms that our original premise was correct: AMG can be made more efficient for interface problems with a careful choice of α. However, AMG is designed to be a black–box linear solver. The type of internal optimization of Section 6.1, while insightful and potentially useful for users interested in interface problems, clearly has its limitations and reduces the black–box nature of

AMG. Therefore, this section searches for a single value of α that performs efficiently over the body of interface problems under consideration.

The effectiveness of an α is taken to be the average of the effective convergence over all the problems (ranging over b values.) Since more n values exist for larger b values, each group is weighted to give equal strength to each of the tested b values.

Table 5 shows the percentage increase in ρ_A from ideal in each b and n combination for selected α values, as well as the (weighted) average increase. This average gives an expectation of how far from ideal a particular run will be if any single α is chosen. Obviously, the optimal is having this value as close to zero as possible. Note that the values for $\alpha = 0.45, 0.50$ are missing in the last column due to the exceedingly large number of cycles necessary to solve the problem.

	b:	1.25	1.25	0.10	0.10	0.005	0.005	0.0005	0.0005	0.00005
	n:	128	256	128	256	128	256	128	256	256
α	Ave. Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ
0.10	10.46%	4.58%	3.13%	57.82%	18.85%	3.75%	4.73%	7.64%	0.39%	1.86%
0.11	5.63%	4.58%	3.18%	9.85%	16.53%	6.43%	5.44%	8.02%	1.88%	0.18%
0.12	7.56%	4.54%	3.12%	35.85%	14.41%	5.13%	2.03%	5.77%	0.86%	1.94%
0.13	7.71%	4.57%	3.15%	33.79%	12.35%	8.53%	2.13%	7.63%	2.73%	1.09%
0.14	6.61%	1.48%	1.38%	51.30%	0.00%	0.00%	1.55%	7.11%	1.84%	0.71%
0.15	**3.23%**	1.48%	1.38%	0.00%	15.57%	1.84%	0.00%	9.31%	2.27%	0.21%
0.16	8.42%	2.61%	3.42%	47.94%	14.74%	1.72%	2.52%	6.81%	4.40%	0.00%
0.17	8.88%	0.98%	0.71%	45.58%	14.02%	2.51%	4.57%	8.33%	4.66%	3.74%
0.18	8.78%	2.89%	3.60%	46.71%	12.24%	3.44%	3.56%	7.21%	4.61%	1.77%
0.19	8.87%	2.88%	3.59%	51.75%	7.87%	4.70%	1.46%	8.06%	6.76%	0.82%
0.20	9.90%	3.04%	3.74%	53.06%	15.72%	3.60%	4.43%	5.13%	5.30%	2.51%
0.25	11.56%	3.28%	4.17%	60.21%	19.95%	2.88%	4.71%	0.00%	5.74%	7.33%
0.30	14.45%	3.44%	4.58%	61.12%	26.72%	6.19%	10.55%	6.18%	9.52%	8.11%
0.35	16.01%	2.65%	2.61%	71.52%	37.53%	9.41%	13.10%	9.56%	6.25%	3.73%
0.40	16.61%	0.59%	0.48%	70.89%	39.67%	16.82%	14.26%	10.23%	5.25%	3.95%
0.45	24.79%	0.39%	0.36%	88.68%	46.75%	18.47%	18.34%	12.19%	13.16%	
0.50	138.85%	381.95%	469.46%	111.04%	61.24%	28.64%	20.32%	20.66%	17.49%	

Table 5: Optimizing α for Operator Complexity

Table 5 shows a clear standout: $\alpha = 0.15$. It represents an average 8.1% increase in efficiency over $\alpha = 0.25$. It fails to outperform $\alpha = 0.25$ in only one case ($b = 0.0005$ and $n = 128$), where it increases ρ_A by only 10%.

Analysis of Tables 6 and 7 coincides with the ρ_A analysis. That is, $\alpha = 0.15$ is again a superior value, producing a 14.1% improvement for grid complexity and a 6.4% improvement in space complexity.

Note also that Table 6 bears out the prediction in Section 6.1, namely that the improvement is more pronounced in the grid complexity than in the space or operator complexities.

7 Conclusions

We have compared a range of strength thresholds in AMG for the body immersed interface problems examined in [1]. These tests show that a decrease in the strength threshold to $\alpha = 0.15$ is superior to the value of $\alpha = 0.25$. This change produces as much as a 6-14% improvement in efficiency, depending on the measure (grid, operator or space). This new α is superior in a majority of cases and competitive in all cases tested.

	b:	1.25	1.25	0.10	0.10	0.005	0.005	0.0005	0.0005	0.00005
	n:	128	256	128	256	128	256	128	256	256
α	Ave. Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ
0.10	14.9%	5.7%	3.9%	91.3%	25.9%	4.7%	6.1%	6.7%	0.2%	2.4%
0.11	7.3%	5.7%	4.0%	14.7%	24.6%	7.2%	6.7%	8.0%	2.1%	0.0%
0.12	10.4%	5.7%	3.9%	57.7%	20.4%	4.8%	2.0%	4.4%	0.8%	2.2%
0.13	10.5%	5.7%	3.9%	54.0%	15.1%	11.2%	2.4%	7.4%	3.2%	1.0%
0.14	9.7%	1.8%	1.7%	82.0%	0.0%	0.0%	2.0%	6.4%	2.0%	0.6%
0.15	4.5%	1.8%	1.7%	0.0%	24.1%	4.0%	0.0%	10.1%	3.1%	0.2%
0.16	12.8%	3.1%	4.4%	78.9%	23.1%	3.6%	3.8%	6.3%	4.8%	0.0%
0.17	13.7%	1.0%	0.7%	75.6%	22.9%	4.7%	6.4%	8.4%	5.6%	5.9%
0.18	14.1%	3.5%	4.6%	81.6%	20.4%	6.5%	5.4%	7.6%	5.5%	3.1%
0.19	14.4%	3.5%	4.6%	89.5%	15.1%	7.8%	2.3%	8.8%	8.9%	1.7%
0.20	16.0%	3.7%	4.8%	90.0%	25.8%	7.5%	6.9%	5.4%	7.1%	4.4%
0.25	19.2%	4.0%	5.3%	104.6%	33.9%	5.8%	10.3%	0.0%	7.4%	10.3%
0.30	24.0%	4.2%	5.9%	108.4%	43.7%	12.7%	17.0%	8.9%	14.0%	12.5%
0.35	27.8%	3.0%	3.2%	126.0%	61.0%	17.6%	21.5%	14.4%	12.7%	9.3%
0.40	29.1%	0.8%	0.6%	125.8%	64.5%	29.1%	25.2%	15.2%	11.0%	9.3%
0.45	42.1%	0.3%	0.3%	157.9%	75.3%	31.6%	31.7%	17.7%	22.4%	
0.50	243.2%	667.8%	870.9%	189.4%	92.3%	43.1%	30.5%	26.8%	25.2%	

Table 6: Optimizing α for Grid Complexity

References

[1] L. Adams and T. Chartier, *A Comparison of Algebraic Multigrid and Geometric Immersed Interface Multigrid Methods for Interface Problems*, SIAM Journal on Scientific Computing, submitted.

[2] ——, *New geometric immersed interface multigrid solvers*, SIAM Journal on Scientific Computing, submitted.

[3] L. Adams and Z. Li, *The immersed interface/multigrid methods for interface problems*, SIAM Journal on Scientific Computing, 24 (2002), pp. 463–470.

[4] M. J. Antonelli, *Optimization of Strength Threshold in Algebraic Multigrid for Immersed Interface Problems*, Senior thesis, University of Washington, 2003.

[5] W. Briggs, V. Henson, and S. McCormick, *A Multigrid tutorial 2nd ed.*, SIAM, Philadelphia, 2000.

[6] J. Cleary, R. Falgout, V. Henson, J. Jones, T. Manteuffel, S. McCormick, G. Miranda, and J. Ruge, *Robustness and scalability of algebraic multigrid*, SIAM Journal on Scientific Computing, 21 (2000), pp. 1886–1908.

[7] Z. Li and K. Ito, *Maximum principle preserving schemes for interface problem with discontinuous coefficients*, SIAM Journal on Scientific Computing, 23 (2001), pp. 339–361.

[8] J. Ruge and K. Stüben, *Algebraic multigrid*, in Multigrid Methods, S. McCormick, ed., vol. 3 of Frontiers in Applied Mathematics, SIAM, Philadelphia, 1987, ch. 4, pp. 73–130.

[9] K. Stüben, *Algebraic multigrid (AMG): An introduction with applications*, Gesellschaft für Mathematik und Datenveranbeitung, Nr. 70 (1999).

	b:	1.25	1.25	0.10	0.10	0.005	0.005	0.0005	0.0005	0.00005
	n:	128	256	128	256	128	256	128	256	256
α	Ave. Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ
0.10	8.0%	3.8%	2.9%	43.7%	14.4%	2.9%	3.7%	5.3%	0.1%	1.4%
0.11	4.3%	3.8%	3.0%	7.4%	13.1%	4.8%	4.2%	5.6%	1.3%	0.1%
0.12	5.9%	3.8%	2.9%	28.2%	11.3%	3.7%	1.5%	3.9%	0.5%	1.5%
0.13	6.0%	3.8%	2.9%	26.4%	9.2%	6.6%	1.5%	5.6%	1.9%	0.8%
0.14	5.0%	1.1%	1.5%	39.2%	0.0%	0.0%	1.2%	5.1%	1.3%	0.5%
0.15	2.6%	1.1%	1.5%	0.0%	12.6%	1.8%	0.0%	6.7%	1.7%	0.1%
0.16	6.6%	2.2%	2.9%	36.9%	12.0%	1.7%	2.0%	4.9%	2.9%	0.0%
0.17	7.0%	0.9%	1.1%	35.3%	11.5%	2.3%	3.3%	6.0%	3.4%	3.1%
0.18	7.0%	2.4%	3.1%	36.8%	10.2%	3.1%	2.6%	5.3%	3.1%	1.5%
0.19	7.1%	2.4%	3.1%	40.6%	7.1%	3.9%	1.2%	6.0%	4.8%	0.8%
0.20	7.9%	2.6%	3.3%	41.4%	13.0%	3.3%	3.7%	3.8%	3.8%	2.2%
0.25	9.2%	2.8%	3.6%	46.9%	16.6%	2.8%	4.6%	0.0%	4.0%	5.5%
0.30	11.6%	2.9%	4.1%	47.6%	21.7%	5.8%	8.5%	5.0%	7.3%	6.3%
0.35	13.0%	2.6%	2.5%	55.2%	29.9%	8.2%	10.7%	7.7%	5.7%	3.8%
0.40	13.5%	0.5%	0.8%	54.9%	31.4%	14.1%	12.1%	8.2%	4.9%	4.0%
0.45	19.6%	0.6%	0.8%	67.5%	36.3%	15.5%	15.3%	9.8%	10.9%	
0.50	88.7%	231.2%	286.1%	80.4%	45.6%	21.9%	15.9%	15.2%	13.4%	

Table 7: Optimizing α for Space Complexity

[10] ——, *A review of algebraic multigrid*, Gesellschaft für Mathematik und Datenveranbeitung, Nr. 69 (1999).