

VARIABLE ORDER CASCADIC MULTIGRID ALGORITHMS

Craig C. Douglas* and Karen M. Walters†

*Departments of Computer Science and Mechanical Engineering and the Center for Computational Sciences University of Kentucky 325 McVey Hall - CCS Lexington, KY 40506-0045, USA e-mail: douglas@ccs.uky.edu, web pages: <http://www.ccs.uky.edu/~douglas> and <http://www.mgnet.org>

†Department of Mathematics University of Kentucky 715 Patterson Office Tower Lexington, KY 40506-0027, USA email: Karen.M.Walters.88@Alum.Dartmouth.Org, web page: <http://www.mgnet.org/~walters>

Key words: Multigrid, Cascadic, Iterative Method, PDE.

Abstract. *Cascadic multigrid algorithms have been shown by a number of researchers (e.g., Bornemann, Deufthard, Shaidurov, Shi, and many others) to be useful in a number of application areas. Cascadic multigrid differs from traditional one way multigrid methods by over solving the problem on coarse grids in order to reduce the required work on the finest grids. Since there is a geometric growth in the number of unknowns as the grids get finer and finer, useful work done on the coarse grids has a significant impact on the work estimate for the entire process. We take the approach that solving a higher order problem on some of the coarse grids will be even better than over solving the problem with a standard order discretization. We combine our approach with higher order, operator based interpolation to get an inexpensive, high order cascadic multigrid algorithm.*

1 INTRODUCTION

Multigrid is an approach to computational problem solving originally developed to solve boundary value problems arising in physical applications. A set of grid points is chosen in the domain of the problem. Using a discretization process, we get a system of algebraic equations that is associated with the chosen grid points and whose solution is an approximation to the continuous problem.

A sequence of related problems is generated on coarser and coarser grids, where the solution on a particular grid is a good approximation to the solution on the next finer grid. Alternately, a coarse grid problem is solved and then (adaptively or uniformly) refined until a fine enough grid is generated that adequately solves the problem.

A restriction operator maps an approximate solution on one level to the next coarser level. This can be, for example, a direct injection of points or a weighted average of each grid point with its nearest neighbors. An interpolation operator takes an approximate solution and maps it to the next finer grid.

The V cycle is the basic multigrid algorithm.

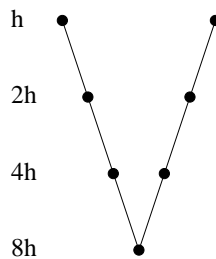


Figure 1: Four level V cycle

We start with a problem on the finest level. A smoother (e.g., Gauss-Seidel) or rougher (e.g., conjugate gradients) is used on this level to damp the high frequency components of the error. The residual is computed and projected to the next coarser level where the residual problem is solved with an initial guess of zero.

This process is repeated down to the coarsest level, where we compute the solution to the residual problem (typically with a direct solver). Then we correct the solution on the next finer grid by adding an interpolation of the coarse grid correction to the existing solution on this finer level. On the next finer level, we smooth the problem and interpolate our new correction up to the next finer level where we again correct the existing solution. This continues until we reach the finest level, where we smooth again to get our final solution.

The full multigrid cycle is composed of a series of nested V cycles.

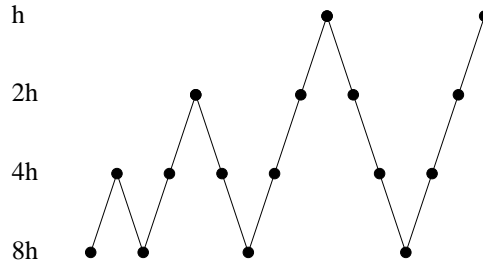


Figure 2: Four level full multigrid cycle

This algorithm begins on the coarsest level. The idea here is to get a good initial guess for a particular level before doing any processing on that level.

The cascadic multigrid algorithm differs from the full multigrid cycle in that it is a one way multigrid starting at the coarsest level and continuing to the finest level without ever going back for a correction.

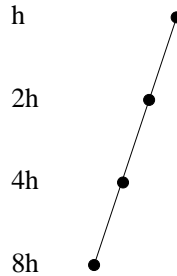


Figure 3: Four level Cascadic multigrid cycle

Cascadic multigrid evolved from the *Cascade Principle* of Deuffhard, Leinen, and Yserentant [5]. The main feature of this method is that the iteration is terminated by an adaptive control as soon as the algebraic error is considerably below the discretization error.

A prime feature of cascadic multigrid is that more smoothing iterations are done on the coarser levels in the hope that fewer smoothing iterations are needed on the finer levels, i.e., we *over solve* the problem on the coarser levels. This does not make sense from the approximation theory viewpoint since we can really only solve up to the truncation error on these levels, which is normally all that multigrid does.

The motivation for cascadic multigrid comes from numerical linear algebra, not approximation theory, however. Once the continuous problem is discretized on all of the levels, we are only going to run the smoother on each level until the residual norm is reduced to some (fixed) size (e.g., 10^{-4} or 10^{-6}). We over solve in the hope that we get a better initial guess when we interpolate to the finer levels.

This actually works some of the time though it still depends on the approximation properties of the solution. For example, we could have higher derivatives blowing up, or we could have the case where the linear algebra system on a level is not related closely enough to the system on the previous level, in which case our approximation will not be good enough. However, for well behaved solutions, cascadic multigrid works well in some very difficult applications.

Shaidurov [7] showed that the cascadic conjugate gradient method (CCGM) is accurate with optimal complexity for elliptic problems in two dimensions and for quasi-uniform triangulations. Note that optimality here is with respect to the energy norm. However, Bornemann and Krause [2] showed that cascadic multigrid cannot be optimal with respect to the L^2 norm.

Bornemann and Deuffhard [1] simplified and extended Shaidurov's results. They showed that the three dimensional problem is accurate with optimal complexity for any choice of energy reducing smoother (e.g., symmetric Gauss-Seidel, SSOR, damped Jacobi, or conjugate gradients). They also showed that the two dimensional problem has optimal complexity using conjugate gradients, and is nearly optimal for the standard iterative smoothers.

Braess and Dahmen [3] looked at the Stokes problem. They discovered that special treatment is needed for the transfer operators. They obtained optimality for saddle point problems. They also found that their technique is useful for nonconforming finite elements.

Shi and Xu [9] showed that the previous results for the second order elliptic problem are valid for other conforming and nonconforming elements. For the plate bending problem, they showed that conjugate gradients gives only nearly optimal complexity, and that other smoothers cannot be used at all.

Shi and Xu [8] also studied parabolic problems. Note that multigrid is usually used for large time steps. For small time steps (i.e., $t \leq Ch^2$, where h is the mesh size), a standard iterative method guarantees good convergence for the discrete system. They showed that two and three dimensional problems were accurate with optimal complexity for conjugate gradients and traditional iterative methods. A one dimensional problem is nearly optimal for conjugate gradients and is optimal for traditional methods when the time step is restricted to not be too large.

In this paper, we use a cascadic multigrid algorithm that uses a higher order discretization on the coarser levels to accelerate the convergence. At some point, there is a switch to the standard (i.e., lower) method on the finer levels.

One reason for switching to a lower order method on the finer grids is that it is cheaper to use than the higher order method and we want to minimize the cost on the finer grids. Not only is the smoother more expensive, but the interpolation process is, too.

Another reason is that, in a number of cases, we are given a problem which may involve some post processing on the finest level that requires a particular order discretization. For example, in a combustion simulation or CFD application, an auxiliary problem may be solved to recover the actual solution.

If we use a fourth order method, we may have problems with continuity constraints on the data. For example, if there are sharp changes along a front, the high order method could result in a Gibb's effect, which completely destroys the value of the solution.

2 ALGORITHMS

In this section, we will define the algorithms that will be used in the numerical experiments' section. The notation used in the algorithms is as follows:

- j the current level number: $j = 1, \dots, l$.
- z the approximate solution.
- $E^{(j)}$ the energy norm on level j in the cascadic multigrid algorithms.
- E_2 the energy norm on level 2 computed in the nested iteration V cycle algorithm.

Note that while the algorithms in this paper are specifically tailored to $O(h^2)$ and $O(h^4)$ discretizations, they can be generalized (and have been elsewhere) to any $O(h^p)$ and $O(h^q)$ methods where $q > p$.

Algorithm 1 computes the energy norm of an approximate solution after a two level full multigrid cycle. For the sake of this paper, the V cycle uses an $O(h^2)$ discretization and linear interpolation. The smoother is symmetric Gauss-Seidel. The energy norm computed here will be used to compute upper bounds on the energy norm on each level of a cascadic multigrid problem using $O(h^4)$ discretization and fourth order interpolation and another using $O(h^2)$ discretization and linear interpolation. The goal is to have the norm on the finest level bound by the V cycle energy norm divided by $4^{(\text{number of levels}-1)}$. Ultimately, this bound will be used in an algorithm in which we switch from the $O(h^4)$ discretization to a $O(h^2)$ discretization.

Algorithm 1 Two level nested iteration V cycle to determine energy norm of $O(h^2)$ method

1. Solve directly on level 1
 2. Interpolate to level 2
 3. Perform a multigrid V cycle:
 - (a) Smooth once on z
 - (b) Compute the residual r and project to level 1
 - (c) Solve the residual problem with initial guess $e = 0$
 - (d) Correct on level 2: $z = z + e$ and smooth once more
 4. Compute energy norm E_2 on level 2
-

Algorithm 2 performs the cascadic multigrid method for an $O(h^4)$ method. On the coarsest level, we solve the problem directly using an $O(h^4)$ discretization. This solution is interpolated to the next finer level using a fourth order interpolation method due to Hyman [6], which is similar to doing interpolation plus a smoothing. Next we smooth using symmetric Gauss-Seidel until the energy norm is less than 1/4 that of the energy norm E_2 computed in Algorithm 1. This is repeated on all levels with bound $E_2/(4^{\text{level}-1})$. The initial guess on the finer grids is the fourth order interpolation of the approximation from the previous grid.

Algorithm 2 Cascadic multigrid using $O(h^4)$ method

1. If $j = 1$
 - (a) Solve directly
 - (b) Interpolate to $j = 2$ using fourth order interpolation
 2. If $j > 1$
 - (a) Smooth on z until $E^{(j)} < E_2/4^{(j-1)}$
 - (b) Interpolate z to level $j + 1$ using fourth order interpolation
-

The steps in Algorithm 3 are identical to those of Algorithm 2 except now the discretization is $O(h^2)$ and the interpolation is linear.

Algorithm 3 Cascadic multigrid using $O(h^2)$ method

1. If $j = 1$
 - (a) Solve directly
 - (b) Interpolate to $j = 2$ using linear interpolation
 2. If $j > 1$
 - (a) Smooth on z until $E^{(j)} < E_2/4^{(j-1)}$
 - (b) Interpolate z to level $j + 1$ using linear interpolation
-

Algorithm 4 performs the cascadic multigrid method using $O(h^4)$ discretization and fourth order interpolation up to a specified level k , at which point the discretization and interpolation are changed to $O(h^2)$ and linear, respectively.

Algorithm 4 Cascadic multigrid switching discretization at level k

1. If $j = 1$
 - (a) Solve directly using $O(h^4)$ method
 - (b) Interpolate to $j = 2$
2. Else if $j = 2, \dots, k$
 - (a) Smooth until $E^{(j)} < E_2/4^{(j-1)}$ using $O(h^4)$ method
 - (b) Interpolate to level $j + 1$ using fourth order interpolation
3. Else if $j = k + 1, \dots, L - 1$
 - (a) Smooth once using $O(h^2)$ method
 - (b) Interpolate to level $j + 1$ using linear interpolation

3 NUMERICAL EXPERIMENTS

Two problems were considered, a two dimensional problem and a three dimensional problem. The formulas given below are for the two dimensional case. Similar formulas apply to the three dimensional problem.

The problem to be solved is Poisson's equation with Dirichlet boundary conditions.

$$\begin{cases} -u_{xx} - u_{yy} = f(x, y) & \text{in } D^2, \\ u = 0 & \text{on } \partial D^2. \end{cases}$$

The interpolation used in the $O(h^2)$ algorithm is the standard bilinear interpolation.

$$\begin{aligned} u_{2i,2j}^{(l)} &= u_{i,j}^{(l-1)} \\ u_{2i+1,2j}^{(l)} &= \frac{1}{2}(u_{i,j}^{(l-1)} + u_{i+1,j}^{(l-1)}) \\ u_{2i,2j+1}^{(l)} &= \frac{1}{2}(u_{i,j}^{(l-1)} + u_{i,j+1}^{(l-1)}) \\ u_{2i+1,2j+1}^{(l)} &= \frac{1}{4}(u_{i,j}^{(l-1)} + u_{i+1,j}^{(l-1)} + u_{i,j+1}^{(l-1)} + u_{i+1,j+1}^{(l-1)}) \end{aligned}$$

The fourth order interpolation used is Hyman's Local Inversion Method [6]. First the coarse points are injected and the unknown fine points are solved for locally as with a smoother. We know the solution at the points (x_{2i}, y_{2j}) , thus these points are injected into the finer grid. For the points (x_{2i+1}, y_{2j+1}) , we rotate the standard five-point difference approximation by 90° . These points are called \times points because they use the surrounding

corner points to approximate the central point. They are calculated as follows:

$$u_{2i+1,2j+1} = \frac{1}{4}(u_{2i,2j} + u_{2i+2,2j} + u_{2i,2j+2} + u_{2i+2,2j+2} + 2h^2 f_{2i+1,2j+1}) + O(h^4).$$

The points (x_{2i}, y_{2j+1}) and (x_{2i+1}, y_{2j}) use the standard five-point difference scheme, called the + scheme. The calculation of these points involves the points at which the solution is known and the \times points:

$$u_{k,l} = \frac{1}{4}(u_{k-1,l} + u_{k+1,l} + u_{k,l-1} + u_{k,l+1} + h^2 f_{k,l}) + O(h^4),$$

where $k = 2i + 1, l = 2j$ or $k = 2i, l = 2j + 1$.

The projection used in the V cycle is the full weighted restriction:

$$\begin{aligned} u_{i,j}^{(j-1)} &= \frac{1}{16}[u_{2i-1,2j-1}^{(j)} + u_{2i-1,2j+1}^{(j)} + u_{2i+1,2j-1}^{(j)} + u_{2i+1,2j+1}^{(j)} \\ &\quad + 2(u_{2i,2j-1}^{(j)} + u_{2i,2j+1}^{(j)} + u_{2i-1,2j}^{(j)} + u_{2i+1,2j}^{(j)}) + 4u_{2i,2j}^{(j)}]. \end{aligned}$$

The second order discretization involves the standard five-point formula:

$$4u_{i,j} - u_{i-1,j} - u_{i+1,j} - u_{i,j-1} - u_{i,j+1} = h^2 f_{i,j}$$

The fourth order discretization uses the nine-point formula known as *Mehrstellenverfahren* [4]:

$$\begin{aligned} 20u_{i,j} - 4(u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1}) - (u_{i-1,j-1} + u_{i-1,j+1} + u_{i+1,j-1} + u_{i+1,j+1}) \\ = \frac{h^2}{2}(8f_{i,j} + f_{i-1,j} + f_{i+1,j} + f_{i,j-1} + f_{i,j+1}) \end{aligned}$$

We give a comparison of the lower and higher order methods for Poisson's problem in two dimensions for a model problem that has the solution $e^{xy} \sin(\pi x) \sin(\pi y)$. The number of iterations on each level is given for the $O(h^2)$ discretization with linear interpolation and the $O(h^4)$ discretization with fourth order interpolation. The coarsest level is level 1 where the number of grid points in each direction is equal to seven.

Method	Level			
	2	3	4	5
Lower order	10	6	4	3
Higher order	4	2	1	1

Table 1: Comparisons for the two dimensional model problem

Next we give a comparison for Poisson's problem in three dimensions for a model problem that has the solution $e^{xyz} \sin(\pi x) \sin(\pi y) \sin(\pi z)$.

Method	Level		
	2	3	4
Lower order	18	6	3
Higher order	6	1	1

Table 2: Comparisons for the three dimensional model problem

4 CONCLUSIONS

In this paper, we developed a mixed order cascadic multigrid cycle. The higher order methods used on the coarser grids give us a better approximation to use on the finer grids once we switch to the lower order method.

Experiments show that the method has great promise for elliptic problems in two and three space dimensions. Theory, which is contained in another paper, shows that the experiments are not a fluke.

REFERENCES

- [1] F. A. Bornemann and P. Deuffhard. The cascadic multigrid method for elliptic problems. *Numer. Math.*, 75:135–152, 1996.
- [2] F. A. Bornemann and R. Krause. Classical and cascadic multigrid - a methodical comparison. In *Proceedings of the Ninth International Conference on Domain Decomposition*, pages 64–71, Bergen, Norway, 1998. Domain Decomposition Press.
- [3] D. Braess and W. Dahmen. A cascadic multigrid algorithm for the Stokes equation. *Numer. Math.*, 82:179–191, 1999.
- [4] L. Collatz. *The Numerical Treatment of Differential Equations*. Springer-Verlag, Berlin, 1960.
- [5] P. Deuffhard, P. Leinen, and H. Yserentant. Concepts of an adaptive hierarchical finite element code. *Impact Comput. Sci. Eng.*, 1:3–35, 1989.
- [6] J. M. Hyman. Mesh refinement and local inversion of elliptic differential equations. *J. Comput. Phys.*, 23:124–134, 1977.
- [7] V. V. Shaidurov. Some estimates of the rate of convergence for the cascadic conjugate-gradient method. *Comput. Math. Appl.*, 31:161–171, 1996.
- [8] Z. Shi and X. Xu. Cascadic multigrid method for parabolic problems. Preprint, 1999.
- [9] Z. Shi and X. Xu. Cascadic multigrid method for elliptic problems. *East-West J. Numer. Math.*, 3:199–209, 1999.