Preprint
UCRL-JC-150259

Multigrid Methods for Nonlinear Problems: An Overview

Van Emden Henson

This article was submitted to
International Society for Optical Engineering
15th Annual Conference on Electronic Imaging
Santa Clara, California
January 20-24, 2003

U.S. Department of Energy

Lawrence
Livermore
National
Laboratory

December 23, 2002

MULTIGRID METHODS FOR NONLINEAR PROBLEMS: AN OVERVIEW

VAN EMDEN HENSON
CENTER FOR APPLIED SCIENTIFIC COMPUTING
LAWRENCE LIVERMORE NATIONAL LABORATORY

Abstract. Since their early application to elliptic partial differential equations, multigrid methods have been applied successfully to a large and growing class of problems, from elasticity and computational fluid dynamics to geodetics and molecular structures. Classical multigrid begins with a two-grid process. First, iterative relaxation is applied, whose effect is to smooth the error. Then a coarse-grid correction is applied, in which the smooth error is determined on a coarser grid. This error is interpolated to the fine grid and used to correct the fine-grid approximation. Applying this method recursively to solve the coarse-grid problem leads to multigrid.

The coarse-grid correction works because the residual equation is linear. But this is not the case for nonlinear problems, and different strategies must be employed. In this presentation we describe how to apply multigrid to nonlinear problems. There are two basic approaches. The first is to apply a linearization scheme, such as the Newton's method, and to employ multigrid for the solution of the Jacobian system in each iteration.

The second is to apply multigrid directly to the nonlinear problem by employing the so-called Full Approximation Scheme (FAS). In FAS a nonlinear iteration is applied to smooth the error. The full equation is solved on the coarse grid, after which the coarse-grid error is extracted from the solution. This correction is then interpolated and applied to the fine grid approximation.

We describe these methods in detail, and present numerical experiments that indicate the efficacy of them.

1. Introduction

Multigrid methods have become a common approach for solving many types of linear problems of the form $A\mathbf{u} = \mathbf{f}$. We denote by $\mathbf{v}$ an approximation to the exact solution $\mathbf{u}$ and by $\mathbf{e}$ the error, $\mathbf{e} = \mathbf{u} - \mathbf{v}$. Defining the residual to be $\mathbf{r} = \mathbf{f} - A\mathbf{v}$, we observe the critical relationship known as the residual equation, namely $A\mathbf{e} = \mathbf{r}$. In brief, multigrid is the recursive application of a two-grid process. First, an iterative method, such as Gauss-Seidel or Jacobi relaxation, is applied to the fine-grid problem. These iterations have the property that after relaxation the error will be smooth. This in turn means that the error can be accurately represented on a coarser grid (e.g., with every second gridpoint removed in each coordinate direction). Since the coarse grid is much smaller than the fine grid, it is much less expensive to work on the coarse grid. These facts permit the second part of the process, known as the coarse-grid correction. The fine-grid residual $\mathbf{r}^h$ is computed and restricted to the coarse grid $\mathbf{r}^{2h} = I_h^{2h}\mathbf{r}^h$, where it is used as the right-hand side of the coarse-grid residual equation $A^{2h}\mathbf{e}^{2h} = \mathbf{r}^{2h}$. This equation is solved, and the error thus determined is then interpolated back to the fine grid where it is used to correct the fine-grid approximation, $\mathbf{v}^h \leftarrow \mathbf{v}^h + I_{2h}^h \mathbf{e}^{2h}$. By recursively solving the coarse-grid equation with this two-grid process, a multigrid algorithm is defined.

Nonlinear problems necessarily must be solved using iterative methods, and for this reason it is natural to expect that multigrid ideas should be effective on these problems. Consider a system of nonlinear algebraic equations, $A(\mathbf{u}) = \mathbf{f}$, where $\mathbf{u}, \mathbf{f} \in \mathbf{R}^n$. Once again the error is given by $\mathbf{e} = \mathbf{u} - \mathbf{e}$ while now $\mathbf{r} = \mathbf{f} - A(\mathbf{v})$ is the residual. Subtracting the original equation from the residual, we obtain

$$A(\mathbf{u}) - A(\mathbf{v}) = \mathbf{r}. \tag{1.1}$$

Since A is nonlinear, in general $A(\mathbf{e}) \neq \mathbf{r}$, implying that for the nonlinear problem we can not determine the error by solving a simple linear equation on the coarse grid, as in standard multigrid. Instead, we must use (1.1) as the residual equation.

There are two basic approaches to using multigrid in the solution of (1.1). One is to use multigrid as the linear solver in a standard linearization, such as in Newton's method or Picard iteration. We briefly describe one such approach, Newton-multigrid, in the following section. The second approach, known as the Full Approximation Scheme (FAS), is to apply multigrid methodology directly to the original equation $A(\mathbf{u}) = \mathbf{f}$ and to base the coarse-grid correction on the nonlinear residual equation (1.1). We describe FAS in section 3. Finally, we demonstrate the usage of both FAS and Newton-multigrid on some simple model problems in section 5.

2. Newton-Multigrid

Undoubtedly most important method for solving nonlinear equations is *Newton's Method.* Suppose we wish to solve the scalar equation $F(x) = 0$. By expanding F in a Taylor series about an initial guess x_0 we obtain

$$F(x_0 + s) = F(x_0) + sF'(x_0) + \frac{s^2}{2}F''(\xi)$$

for some ξ between x_0 and $x_0 + s$. If $x_0 + s$ were the solution, then neglecting the second-order term and solving for s we obtain an improved approximate solution given by $x_0 \leftarrow x_0 - F(x_0)/F'(x_0)$. This is easily turned into the Newton iteration:

$$x_j \leftarrow x_j - \frac{F(x_j)}{F'(x_j)}.$$

We can use (1.1) as a basis for a multigrid solver by applying Newton's method to the system of n nonlinear equations

$$\mathbf{A}(\mathbf{x}) \equiv \begin{bmatrix} A_1(x_1, x_2, \dots x_n) \\ A_2(x_1, x_2, \dots x_n) \\ \vdots \\ A_n(x_1, x_2, \dots x_n) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Letting $J(\mathbf{v})$ be the Jacobian matrix

$$J(\mathbf{v}) = \begin{bmatrix} \frac{\partial A_1}{\partial x_1} & \frac{\partial A_1}{\partial x_2} & \cdots & \frac{\partial A_1}{\partial x_n} \\ \frac{\partial A_2}{\partial x_1} & \frac{\partial A_2}{\partial x_2} & \cdots & \frac{\partial A_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial A_n}{\partial x_1} & \frac{\partial A_n}{\partial x_2} & \cdots & \frac{\partial A_n}{\partial x_n} \end{bmatrix}_{\mathbf{x}=\mathbf{v}},$$

and letting $\mathbf{u} = \mathbf{v} + \mathbf{e}$ be the exact solution, the Taylor series for this system is

$$\mathbf{A}(\mathbf{v}+\mathbf{e}) = \mathbf{A}(\mathbf{v}) + J(\mathbf{v})\mathbf{e} + \text{higher order terms.}$$

Neglecting the higher order terms and subtracting $\mathbf{A}(\mathbf{v})$ from both sides, we see that

$$J(\mathbf{v})\mathbf{e}, \quad = \quad \mathbf{r}, \tag{2.1}$$

which represents an approximation to the nonlinear system (1.1). It can be solved for $\mathbf{e}$ and the current approximation $\mathbf{v}$ can be updated by $\mathbf{v} \leftarrow \mathbf{v} + \mathbf{e}$. Iteration of this step is

$$\mathbf{v}_j \leftarrow \mathbf{v}_j + J^{-1}(\mathbf{v}_j)\left(f_j - \mathbf{A}(\mathbf{v}_j)\right), \tag{2.2}$$

a form of Newton's method. By using multigrid to solve linear system (2.1) at each step we obtain a combination of Newton's method for the outer iteration and multigrid for the (linear) inner iteration. This is called *Newton-multigrid.* We will present examples of this method in section 5.

3. The Full Approximation Scheme (FAS)

While Newton-MG is often an extremely effective method, it does not use multigrid ideas to treat the nonlinearity directly. To do this, we return to the residual equation (1.1) and use it to determine a coarse-grid correction. Suppose we have found an approximation, $\mathbf{v}^h$, to the original fine-grid problem

$$A^h(\mathbf{u}^h) = \mathbf{f}^h\,. \tag{3.1}$$

The coarse-grid version of (1.1) is

$$A^{2h}(\mathbf{v}^{2h} + \mathbf{e}^{2h}) - A^{2h}(\mathbf{v}^{2h}) = \mathbf{r}^{2h}. \tag{3.2}$$

The coarse-grid residual is chosen simply to be the restriction of the fine-grid residual

$$\mathbf{r}^{2h} = I_h^{2h}\mathbf{r}^h = I_h^{2h}(\mathbf{f}^h - A^h(\mathbf{v}^h))$$

and, similarly, it makes sense to restrict the fine-grid approximation using the same transfer operator; that is, $\mathbf{v}^{2h} = I_h^{2h}\mathbf{v}^h$.

Making these substitutions in the coarse-grid residual equation yields

$$A^{2h}(\underbrace{I_h^{2h}\mathbf{v}^h + \mathbf{e}^{2h}}_{\mathbf{u}^{2h}}) = \underbrace{A^{2h}(I_h^{2h}\mathbf{v}^h) + I_h^{2h}(\mathbf{f}^h - A^h(\mathbf{v}^h))}_{\mathbf{f}^{2h}}. \tag{3.3}$$

The right side of this nonlinear system is known, and the equation is of the same form as the original fine-grid equation (3.1). Assume we can find a solution to this system, which we denote $\mathbf{u}^{2h}$. The coarse-grid error can be extracted from the solution by $\mathbf{e}^{2h} = \mathbf{u}^{2h} - I_h^{2h}\mathbf{v}^h$, and can then be interpolated up to the fine grid and used to correct the fine-grid approximation $\mathbf{v}^h$:

$$\mathbf{v}^h \leftarrow \mathbf{v}^h + I_{2h}^h \mathbf{e}^{2h} \quad \text{or} \quad \mathbf{v}^h \leftarrow \mathbf{v}^h + I_{2h}^h(\mathbf{u}^{2h} - I_h^{2h}\mathbf{v}^h).$$

This scheme is the most commonly used nonlinear version of multigrid. Because the coarse-grid problem is solved for the full approximation, rather than the error $\mathbf{e}^{2h}$, the method is called the *full approximation scheme* (FAS). A two-grid version of this scheme is described as follows:

Full Approximation Scheme (FAS)

- Restrict the fine-grid approximation and its residual: $\mathbf{r}^{2h} = I_h^{2h}(\mathbf{f}^h - A^h(\mathbf{v}^h))$ and $\mathbf{v}^{2h} = I_h^{2h}\mathbf{v}^h$.
- Solve the coarse-grid problem $A^{2h}(\mathbf{u}^{2h}) = A^{2h}(\mathbf{v}^{2h}) + \mathbf{r}^{2h}$.
- Compute the coarse-grid approximation to the error: $\mathbf{e}^{2h} = \mathbf{u}^{2h} - \mathbf{v}^{2h}$.
- Interpolate the error approximation to the fine grid and correct the current fine-grid approximation: $\mathbf{v}^h \leftarrow \mathbf{v}^h + I_{2h}^h\mathbf{e}^{2h}$.

It is easy to see that if A is a linear operator, then FAS reduces directly to the linear coarse-grid correction scheme. Hence, FAS is in fact a generalization of the coarse-grid correction to nonlinear problems. It is also useful to observe that $\mathbf{v}^h$ is a fixed point of the FAS iteration if and only if it is an exact solution of the fine-grid problem.

We may also view FAS as a way to alter the coarse-grid equations so that their approximation properties are substantially enhanced. This may be seen if the FAS coarse-grid equation is written as

$$A^{2h}(\mathbf{u}^{2h}) = \mathbf{f}^{2h} + \boldsymbol{\tau}_h^{2h} \quad \text{where} \quad \boldsymbol{\tau}_h^{2h} = A^{2h}(I_h^{2h}\mathbf{v}^h) - I_h^{2h}A^h(\mathbf{v}^h)$$

defines the so-called *tau correction* $\boldsymbol{\tau}_h^{2h}$. Because $\boldsymbol{\tau}_h^{2h} \neq \mathbf{0}$ generally, the solution of the coarse-grid FAS equation, $\mathbf{u}^{2h}$, is not the same as the solution of the coarsened original equation $A^{2h}(\mathbf{u}^{2h}) = I_h^{2h}f^h$. In fact, as FAS processing advances, $\mathbf{u}^{2h}$ begins to achieve *accuracy* that compares to that of the solution on the finest grid, albeit at the *resolution* of grid $2h$.

4. Nonlinear Relaxation and Cycling

As described above, multigrid is the recursive application of a two-stage process performed on two grids. For nonlinear problems FAS plays the role that the coarse-grid correction plays in the linear case. For FAS to be effective in a multigrid method, it must be combined with a relaxation scheme to ensure that the error is smooth before extracting it from the coarse grid. Hence we require a relaxation method for nonlinear systems. Most linear relaxation schemes have analogs for nonlinear systems; one of the most frequently used methods is *nonlinear Gauss-Seidel* relaxation [5]. We can describe this algorithm just as in the linear case: for $1 \leq j \leq n$, solve $(A(\mathbf{v}))_j = \mathbf{f}_j$ for v_j. We can write this equivalently by letting $\boldsymbol{\epsilon}_j$ be the jth unit vector, so that nonlinear Gauss-Seidel amounts to finding an $s \in \mathbf{R}$ such that

$$(A(\mathbf{v} + s\boldsymbol{\epsilon}_j))_j = \mathbf{f}_j, \quad 1 \leq j \leq n.$$

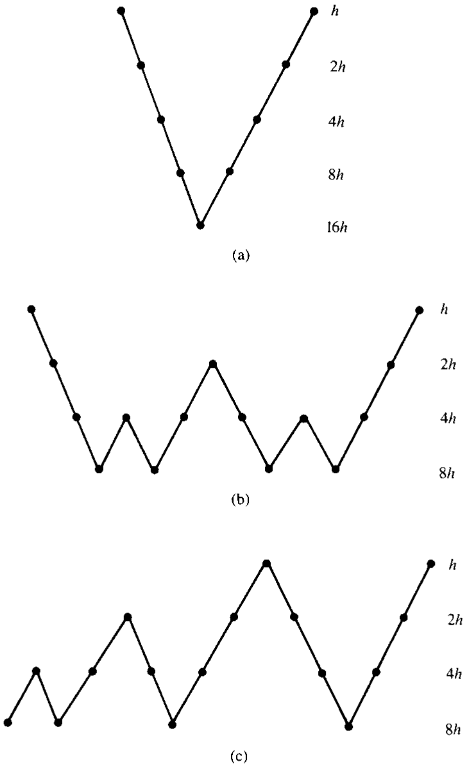

FIGURE 1. *Schedule of grids for (a) V-cycle, (b) W-cycle, and (c) FMG scheme, all on four levels.*

Typically, this is a nonlinear scalar equation in the scalar s, and the scalar Newton's method can be used to solve it efficiently [5]. The jth component is then updated by $\mathbf{v} \leftarrow \mathbf{v} + s\boldsymbol{\epsilon}_j$. Updating all n components sequentially constitutes one iteration sweep of nonlinear Gauss-Seidel.

The two-stage process of *a)* nonlinear relaxation followed by *b)* FAS correction is exactly analogous to the relaxation/coarse-grid correction that comprises the two-grid scheme underlying conventional multigrid. Hence it is natural to ask at this point, "How do we solve the coarse-grid problem $\mathbf{A}^{2h}(\mathbf{u}^{2h}) = \mathbf{f}^{2h} + \boldsymbol{\tau}_h^{2h}$?" Nonlinear multigrid results by recursively applying the same nonlinear relaxation/FAS correction algorithm to solve this problem. Thus, FAS, like its linear counterparts, is usually implemented as a V-cycle or W-cycle scheme (Figure 1).

Another important technique is the use of full multigrid (FMG) for obtaining a good initial guess the fine-grid problem [3], in which for any given grid level, the problem is first solved on the next coarser grid to provide and interpolated to the given grid to provide a good initial guess. The solution process begins on the coarsest grid (Figure 1(c)). Convergence of nonlinear iteration depends even more critically on a good initial guess, since the closer the fine-grid initial guess is to the solution, the more linear the fine-grid problem appears, and the more effective the solver will be. Using FMG for nonlinear problems typically ensures that the interpolant $I_{2h}^h \mathbf{u}^{2h}$ lies in the basin of attraction of the fine-grid solver. Thus, we can expect one FMG cycle to provide good accuracy whether we use Newton-multigrid or FAS V-cycles on each new level. If greater accuracy is required, the FMG cycle is followed by successive V-cycles.

5. Some Numerical Examples

We demonstrate the application of FAS and Newton-multigrid with a performance study of these solvers applied to the two-dimensional nonlinear problem

$$\begin{aligned} -\Delta u(x,y) + \gamma\, u(x,y)\, e^{u(x,y)} &= f(x,y) \qquad \text{in } \Omega, \\ u(x,y) &= 0 \qquad\quad\;\; \text{on } \partial\Omega, \end{aligned}$$

where Ω is the unit square $[0,1] \times [0,1]$. Note that for $\gamma = 0$, this problem reduces to the Poisson problem. We discretize this equation on uniform grids in both x- and y-directions; grid spacings of $h = 1/64$ and $h = 1/128$ are used. We apply a standard second-order finite difference operator to approximate the Laplacian, and the equation for the unknown $u_{i,j}$ at $(x_i, y_j) = (ih, jh)$ is

$$h^{-2}(4u_{i,j} - u_{i-1,j} - u_{i+1,j} - u_{i,j-1} - u_{i,j+1}) + \gamma\, u_{i,j}\, e^{u_{i,j}} = f_{i,j}, \tag{5.1}$$

for $1 < i,j < n$. To enforce the boundary condition, we set

$$u_{0,j} = u_{N,j} = u_{i,0} = u_{i,N} = 0$$

wherever these terms appear in the equations.

We consider first the implementation of an FAS solver for this problem. We must select intergrid transfer operators; the easiest and most natural for this problem are to use a linear interpolation operator and a full-weighting restriction operator. The linear interpolation operator can be defined by $\mathbf{v}^h = I_{2h}^h \mathbf{v}^{2h}$, with components of $\mathbf{v}^h$ given, for $0 \le i,j \le \frac{n}{2} - 1$, by

$$\begin{aligned} v_{2i,2j}^h &= v_{ij}^{2h}, \\ v_{2i+1,2j}^h &= \frac{1}{2}\big(v_{ij}^{2h} + v_{i+1,j}^{2h}\big), \\ v_{2i,2j+1}^h &= \frac{1}{2}\big(v_{ij}^{2h} + v_{i,j+1}^{2h}\big), \\ v_{2i+1,2j+1}^h &= \frac{1}{4}\big(v_{ij}^{2h} + v_{i+1,j}^{2h} + v_{i,j+1}^{2h} + v_{i+1,j+1}^{2h}\big). \end{aligned}$$

The full-weighting restriction operator produces at a coarse-grid point a value that is just an average of the values at the corresponding fine-grid point and its eight nearest neighbors:

$$\begin{aligned} v_{ij}^{2h} = \frac{1}{16}\big[& v_{2i-1,2j-1}^h + v_{2i-1,2j+1}^h + v_{2i+1,2j-1}^h + v_{2i+1,2j+1}^h \\ & + 2\big(v_{2i,2j-1}^h + v_{2i,2j+1}^h + v_{2i-1,2j}^h + v_{2i+1,2j}^h\big) \\ & + 4v_{2i,2j}^h\big], \end{aligned}$$

for $1 \le i, j \le \frac{n}{2} - 1$.

Coarse-grid versions of the nonlinear operator are obtained in the same manner as the fine-grid operator. That is, we employ discretization (5.1) with the appropriate grid spacing ($2h$, $4h, \ldots$) and index limits. Hence we may examine the individual component equations of the FAS correction (3.3). Once a fine-grid approximation $\mathbf{v}^h$ has been obtained and $\mathbf{v}^{2h} = I_h^{2h}\mathbf{v}^h$ and $\mathbf{r}^{2h} = I_h^{2h}(\mathbf{f}^h - \mathbf{A}^h(\mathbf{u}^h))$ are computed, the individual component equations of (3.3) appear as

$$\begin{aligned}4h^{-2}(4u_{i,j}^{2h} - u_{i-1,j}^{2h} - u_{i+1,j}^{2h} - u_{i,j-1}^{2h} - u_{i,j+1}^{2h}) + \gamma\, u_{i,j}^{2h}\, e^{u_{i,j}^{2h}} \\ = 4h^{-2}(4v_{i,j}^{2h} - v_{i-1,j}^{2h} - v_{i+1,j}^{2h} - v_{i,j-1}^{2h} - v_{i,j+1}^{2h}) + \gamma\, v_{i,j}^{2h}\, e^{v_{i,j}^{2h}} + r_{i,j}^{2h}.\end{aligned} \tag{5.2}$$

for $1 \le i, j \le \frac{n}{2} - 1$. Note that the left side of this equation is exactly the same form as (5.1) and represents the nonlinear operator; the right side carries the effect of the nonlinearity in the full approximation mode.

We employ nonlinear Gauss-Seidel as the relaxation method. Because the individual component equations of the system are nonlinear, the relaxation uses scalar Newton's method to solve the (i, j)th equation for $u_{i,j}$:

$$u_{i,j} \leftarrow u_{i,j} - \frac{h^{-2}(4u_{i,j} - u_{i-1,j} - u_{i+1,j} - u_{i,j-1} - u_{i,j+1}) + \gamma\, u_{i,j}\, e^{u_{i,j}} - f_{i,j}}{4h^{-2} + \gamma\,(1 + u_{i,j})\, e^{u_{i,j}}}. \tag{5.3}$$

An important observation is that while the nonlinear Gauss-Seidel is defined as *solving* the (i, j)th equation for the (i, j)th unknown, in practice it is not necessary to solve this equation to any real accuracy. The reason for this is that the relaxation is not being used as a solver; rather, its purpose is to smooth the error, and this is accomplished with very little effort. Although no fixed rule exists to determine how many Newton steps are required in each nonlinear Gauss-Seidel sweep, a small number often suffices (we use a single Newton step for each equation in these experiments). It should be noted that the number of Newton steps used on the scalar problem can have a significant impact on both the effectiveness and the cost of the FAS algorithm.

At the "bottom" of the V-cycle is the coarsest-grid problem. For linear multigrid it is usually simple to solve the system on the coarsest grid exactly, because it requires either the solution of one linear equation in one unknown or the inversion of a very small matrix. Here, however, we use as the coarsest grid a 3×3 grid with a single interior point, and the equation to be solved for $u_{1,1}$ is

$$16u_{1,1} + \gamma\, u_{1,1}\, e^{u_{1,1}} = f_{1,1}.$$

Because this equation is nonlinear, the "solve" on the coarsest grid requires the use of Newton's method. All experiments presented here use a 3×3 grid as the coarsest grid, and in all cases it is determined experimentally that a single Newton step is sufficient.

For these experiments we let the exact solution to the problem be

$$u(x, y) = (x - x^2)(y - y^2),$$

which results in the source term

$$f(x,y) = 2((x-x^2)+(y-y^2)) \;+\; \gamma\,(x-x^2)(y-y^2)\,e^{(x-x^2)(y-y^2)}.$$

It should be observed that the for this particular solution, a second-degree polynomial, there is no discretization error, which facilitates examination of the results.

We first examine the effectiveness of an FAS V-cycle solver for various choices of γ, from the linear case ($\gamma = 0$) through cases in which the nonlinear part of the operator dominates ($\gamma = 10,000$). Each test is posed on a 127×127 interior grid, and cycling continues until the norm of the residual is smaller than 10^{-10}. We report the average convergence rate and the number of V-cycles required to achieve the desired tolerance. The results appear in Table 1.

	γ					
	0	1	10	100	1000	10000
Convergence Factor	.136	.135	.124	.098	.072	.039
Number of FAS cycles	12	12	11	11	10	8

TABLE 1. *FAS performance for the problem* $-\Delta u + \gamma\,u\,e^u = f$, *discretized on grid with* 127×127 *interior points.*

This particular problem becomes *easier* to solve as the nonlinear term becomes more dominant, which is, in general, an uncommon effect in a nonlinear problem. Since the nonlinear term involves only $u_{i,j}$ and none of its neighboring values, as γ becomes larger the problem becomes more local in nature. As this occurs, the relaxation sweeps become more effective, being local in nature. This phenomenon is analogous to increasing diagonal dominance in the linear case. This type of behavior would not be expected for other types of nonlinearity.

We can compare the performance of the FAS solver Newton's method for this problem. The Jacobian matrix for this nonlinear problem is a block tridiagonal system

$$J(\mathbf{u}) = \begin{bmatrix} J_1 & B & & & & \\ B & J_2 & B & & & \\ & B & J_3 & B & & \\ & & \ddots & \ddots & \ddots & \\ & & & B & J_{N-2} & B \\ & & & & B & J_{N-1} \end{bmatrix},$$

where each of the block matrices is $(n-1) \times (n-1)$. The off-diagonal blocks B are all $-1/h^2$ times the identity matrix. The diagonal blocks are tridiagonal, with the constant value $-1/h^2$ on the super- and sub-diagonals. The diagonal entries of J_j, corresponding to the grid locations $(x_i, y_j) = (ih, jh)$ for fixed j and $1 \le i \le n-1$, are given by

$$(J_j)_{i,i} = \frac{4}{h^2} + \gamma\,u_{i,j}\,e^{u_{i,j}}.$$

For this experiment, we use a banded LU decomposition to Jacobian matrix at each Newton step. Because $J(\mathbf{u})$ is sparse and narrow-banded, so are the factors L and U and doth the decomposition and the solve can be computed quite efficiently. Table 2 gives the results of applying Newton's method to the problem for various choices of γ.

	γ					
	0	1	10	100	1000	10000
Convergence Factor	2.6e-13	3.9e-5	7.4e-5	3.2e-4	1.9e-4	1.2e-4
Newton Iterations	1	3	3	4	4	4

TABLE 2. *Performance of Newton's method for the problem* $-\Delta u + \gamma\, u\, e^u = f$.

Tables 1 and 2 reveal that Newton's method converges much faster than does FAS. This should not be surprising, as Newton's method converges quadratically, while multigrid is generally linear in its convergence. However, this comparison fails to consider the computational costs of the methods, which we will discuss shortly.

Before doing so, however, we consider the use of multigrid V-cycles as a an method for inverting the Jacobian system in the Newton iteration. If we solve the Jacobian system to the same level of accuracy by V-cycles as was done with the LU decomposition, the results should be essentially the same as in Table 2. However, it may be effective to solve the Jacobian system only approximately. This technique is known as the *inexact Newton's method*[4]. Performing inexact Newton by using only a few V-cycles is a method we denote *Newton-MG*.

The following experiment is designed to compare the performance of FAS, Newton's methods, and Newton-MG. Consider the same operator as before, with $\gamma = 10$, and let the exact solution be $u(x, y) = (x^2 - x^3)\sin(3\pi y)$. Unlike in the previous experiments, this solution results in a nontrivial discretization error. The resulting source term is

$$-\Delta u + \gamma u e^u = \left(\left(9\pi^2 + \gamma e^{(x^2-x^3)\sin(3\pi y)}\right)(x^2 - x^3) + 6x - 2\right)\sin(3\pi y). \tag{5.4}$$

The problem is discretized with $n = 128$ so that the interior grid is 127 × 127.

Table 3 shows the performance on and the computational costs of FAS, Newton (with a direct LU-based solve), and Newton-MG applied to (5.4), resulting in the nonlinear system (5.1). The iteration is stopped when the residual norm is less than 10^{-10}. The column labeled *outer iterations* lists how many V-cycles (for FAS) or Newton steps (for Newton and Newton-MG) are required to achieve the desired tolerance. For Newton-MG, we vary the number of V-cycles used to approximately solve the Jacobian system. The column labeled *inner iterations* gives the number of V-cycles used in the approximate inversion of the Jacobian system. The last column, labeled *Megaflops*, is the number of millions of floating-point operations expended by the methods. For this example, using these performance measurements, it appears that both of the multigrid-based methods are much more efficient than Newton's method using a direct solver. Furthermore, Newton-MG compares well to FAS when the number of inner MG iterations is properly tuned.

Method	No. outer iterations	No. inner iterations	Megaflops
Newton	3	–	1660.6
Newton-MG	3	20	56.4
Newton-MG	4	10	38.5
Newton-MG	5	5	25.1
Newton-MG	10	2	22.3
Newton-MG	19	1	24.6
FAS	11	–	27.1

Table 3. *Comparison of FAS, Newton, and Newton-multigrid methods for the problem* $-\Delta u + \gamma\, u\, e^u = f$ *on a* 127×127 *grid. In all cases, a zero initial guess is used.*

While the convergence properties may be representative for a fairly broad class of nonlinear problems, we must emphasize that several factors can affect the operation counts dramatically. For example, they may depend heavily on the character of the nonlinearity, the details of implementation, the computer architecture, and the programming language. It should also be remembered that we have not accounted for the cost of evaluating the nonlinear function, which is highly problem-dependent, and, for many problems, may be the most costly part of the process. It does seem fairly clear, however, that for large problems the multigrid-based methods will outperform Newton's method using direct solvers. Further, it is apparent that, if Newton-MG is to be competitive with FAS, only a few V-cycles should be used in solving the Jacobian system. In the end, there may be very little difference in the performance between the two carefully designed multigrid-based schemes. The choice may depend largely on convenience and other available algorithm features (for example, τ-extrapolation in FAS [2]).

We should consider one further option: combining the Newton and FAS methods with a full multigrid (FMG) scheme [2, 3]. In FMG, for each new fine grid, the initial guess is obtained by first solving the nonlinear problem on the next coarser grid. For linear multigrid problems, a single FMG cycle suffices to reduce the error *to the level of discretization.* One indication that this is the case is that after a single FMG cycle continued V-cycles have the effect of further decreasing the residual (as the linear systems are solved with increasing accuracy) but the norm of the error no longer decreases.

Table 4 shows the results of applying the FMG-FAS combination to (5.4). Norms are shown, for the residual and the error, after one FMG-FAS cycle and after each of eight subsequent FAS V-cycles. Both the FMG cycle and the FAS V-cycles are performed using nonlinear Gauss-Seidel. The norm of the error is reduced to 2.0×10^{-5} by the FMG-FAS cycle alone. Further FAS V-cycling does not reduce the error, indicating that it has reached the level of discretization. However, subsequent FAS V-cycles do reduce the residual norm further until it reaches the prescribed tolerance of 10^{-10}. The column labeled *Mflops* in the table gives the cumulative number of floating point operations after each stage of the computation.

Cycle	$\|\|\mathbf{r}\|\|$	ratio	$\|\|\mathbf{e}\|\|$	Mflops
FMG-FAS	1.07e-2		2.00e-5	3.1
FAS V 1	6.81e-4	0.064	2.44e-5	5.4
FAS V 2	5.03e-5	0.074	2.49e-5	7.6
FAS V 3	3.89e-6	0.077	2.49e-5	9.9
FAS V 4	3.25e-7	0.083	2.49e-5	12.2
FAS V 5	2.98e-8	0.092	2.49e-5	14.4
FAS V 6	2.94e-9	0.099	2.49e-5	16.7
FAS V 7	3.01e-10	0.102	2.49e-5	18.9
FAS V 8	3.16e-11	0.105	2.49e-5	21.2

TABLE 4. *Performance of the FMG-FAS cycle, followed by eight FAS V-cycles, on* $-\Delta u + \gamma u e^u = f$, *with* $\gamma = 10$. *The grid size is* 127×127. *Note that one FMG-FAS cycle reduces the error to the level of discretization, and that subsequent FAS V-cycles further reduce the residual norm quickly to the prescribed tolerance of* 10^{-10}.

Analogous results for FMG with a Newton solver applied to (5.4) are shown in Table 5. Again we use FMG, applying one step of Newton-MG on each level in the FMG process. Each Newton-MG step starts with an initial guess from the next coarser grid and uses one V-cycle. Norms are shown for the residual and error after one FMG-Newton-MG cycle followed by subsequent Newton-MG cycles on the fine grid. The results are very similar to those for FMG-FAS in Table 4. Observe that the norm of the actual error is reduced to the level of discretization by one FMG-Newton cycle. Subsequent Newton-MG cycles do, however, continue to reduce the norm of the residual effectively to below the prescribed tolerance.

Both of these methods reduce the error to the level of discretization in one FMG cycle. The flop count indicates that the methods are similar in cost, with the FMG-Newton-MG cycle somewhat less expensive (2.4 Mflops) than the FMG-FAS (3.1 Mflops). However, the individual Newton-MG steps on the fine grid, although cheaper, are not quite as effective as FAS V-cycles for reducing the residual norm. Indeed, if the goal is to reduce the residual norm to 10^{-10}, it is somewhat less expensive (21.2 vs. 26.2 Mflops) to use FAS than Newton-MG. Again, however, we note that comparative measures of efficiency depend critically on specific implementation details, computing environment, and problem characteristics. The major conclusion to be reached is that both methods are very efficient and robust.

6. Conclusions

We have presented this material very much in an overview or tutorial style, and have not discussed particular applications. Both Newton-MG and FAS are in widespread use for many

Cycle	$\|\|\mathbf{r}\|\|$	ratio	$\|\|\mathbf{e}\|\|$	Mflops
FMG-Newton-MG	1.06e-002		2.50e-005	2.4
Newton-MG 1	6.72e-004	0.063	2.49e-005	4.1
Newton-MG 2	5.12e-005	0.076	2.49e-005	5.8
Newton-MG 3	6.30e-006	0.123	2.49e-005	7.5
Newton-MG 4	1.68e-006	0.267	2.49e-005	9.2
Newton-MG 5	5.30e-007	0.315	2.49e-005	10.9
Newton-MG 6	1.69e-007	0.319	2.49e-005	12.6
Newton-MG 7	5.39e-008	0.319	2.49e-005	14.3
Newton-MG 8	1.72e-008	0.319	2.49e-005	16.0
Newton-MG 9	5.50e-009	0.319	2.49e-005	17.7
Newton-MG 10	1.76e-009	0.319	2.49e-005	19.4
Newton-MG 11	5.61e-010	0.319	2.49e-005	21.1
Newton-MG 12	1.79e-010	0.319	2.49e-005	22.8
Newton-MG 13	5.71e-011	0.319	2.49e-005	24.5

Table 5. *Performance of the FMG-Newton-MG cycle, followed by 13 Newton-MG steps, on* $-\Delta u + \gamma u e^u = f$*, with* $\gamma = 10$*. The grid size is* 127×127*. Note that one FMG-Newton-MG cycle reduces the error to the level of discretization, and that subsequent Newton-MG steps on the fine grid further reduce the residual error to the prescribed tolerance of* 10^{-10}*.*

application areas, including computational fluid dynamics, radiation transport, image reconstruction, and porous media flow, among others. Still other techniques, including hybrids of various linear and nonlinear methods, are known or are under development. Current research also includes applying FAS to unconventional multigrid, such as algebraic multigrid (AMG).

Two major factors determine what sort of nonlinear multigrid solver is applied. First, one must determine how the nonlinear operator can be represented on the coarse grid. The most common approach is to apply the same discretization method with a coarse spacing as was used to discretize the fine grid problem. If this is unfeasible of overly expensive, however, other choices must be considered if FAS is to be used. Sometimes it is possible to use the so-called Galerkin method, in which the action of the coarse-grid operator is given by $A^{2h}(u^{2h}) = I_h^{2h} A^h(I_{2h}^h u^{2h})$. The second factor is the relative expense of evaluating the nonlinear function versus the cost of forming the Jacobian matrix. In many applications evaluating the nonlinear function is a very expensive computational procedure; indeed, it can be the most costly part of the FAS algorithm. If the Jacobian matrix can be evaluated with relative ease, Newton-MG may be preferable. These determinations are highly dependent on the nature of the specific problem at hand.

For any nonlinear method, it is well-known that the iteration is much more reliable and much faster if the initial guess is good. Hence, it is well worth the effort to obtain the highest quality initial guess possible. As a result, nested iteration, the method that obtains a good initial guess by first solving the problem on a coarser grid, should be employed. Doing this on all levels is the heart of the FMG approach. Whether FAS, Newton-MG, or some other nonlinear solver is obtained, using the FMG scheme will improve both speed of convergence and robustness.

7. Acknowledgements

This discussion follows closely that of Chapter 6 of the book *A Multigrid Tutorial, Second Edition*, published by the Society for Industrial and Applied Mathematics and co-authored by Bill Briggs, Steve McCormick,and the author of this paper [3]. The numerical examples are taken directly from that work. This has been done with the gracious permission of the authors and the publisher.

This work was performed under the auspices of the U. S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract W-7405-Eng-48.

The author's contact information is: Van Emden Henson, Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, L-560, PO Box 808, Livermore, CA, 94551, email:*vhenson@llnl.gov.*

References

[1] A. Brandt, *Generally highly accurate algebraic coarsening*, Elec. Trans. Num. Anal., 10 (2000), pp. 1–20.

[2] A. Brandt, *Multigrid techniques: 1984 guide with applications to fluid dynamics*, GMD-studien Nr. 85, Gesellschaft für Mathematik und Datenverarbeitung, St. Agustin, 1984.

[3] W. L. Briggs, V. E. Henson, and S. F. McCormick,*A Multigrid Tutorial, Second Edition*, SIAM Books, Philadelphia, 2000.

[4] J.E. Dennis and R.B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, SIAM Books, Philadelphia, 1996.

[5] J.M. Ortega and W.C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, San Diego, 1970.