

## A MULTILEVEL NONLINEAR METHOD\*

IRAD YAVNEH<sup>†</sup> AND GREGORY DARDYK<sup>†</sup>

**Abstract.** A multilevel nonlinear method (MNM) for the numerical solution of nonlinear partial differential equations is developed. The MNM algorithm is motivated and analyzed using a simplified model which retains the essential features of the new approach. It is thereby shown to combine the advantages of the two classical multigrid approaches to nonlinear problems. The analysis is supported by numerical tests for nonlinear differential equations in one and two dimensions.

**Key words.** multigrid, nonlinear partial differential equations, numerical methods for partial differential equations

**AMS subject classifications.** 65N55, 65N22, 65H10

**DOI.** 10.1137/040613809

**1. Introduction.** Many problems in the sciences and engineering are modeled by nonlinear partial differential equations (PDEs), usually requiring numerical solution. Most commonly, the PDE is approximated by a finite difference, element, or volume discretization on a grid, which results in a large but sparse nonlinear algebraic system of equations. Reliable and computationally efficient solvers for such systems are essential for achieving relevant and accurate simulations.

One of the most efficient techniques for solving such large systems is the multigrid iterative method (see, for example, the classical [1, 14, 15], the expository [5], and the comprehensive [25]). In this approach, a sequence of progressively coarser grids is used to accelerate the convergence of some basic iterative process on the target grid. The basic iterative method is referred to as the “relaxation” or “smoothing” operator, as its objective is to smooth the error in the current approximation relative to the computational grid. There are two general approaches for solving discretized *nonlinear* PDEs by multigrid methods. One is to perform a global linearization (GL), usually by Newton’s method or some inexact variant thereof, and solve the resulting linear system approximately by a linear multigrid algorithm; this is then repeated iteratively. The second approach, represented by the full approximation scheme (FAS) of [1, 2] and the closely related nonlinear multigrid method (NLMG) of [14], is to perform only local linearization (LL)—in the error-smoothing process. Convergence acceleration is then provided by nonlinear coarse-grid operators.

Over the years, variants of these two general categories have been proposed and developed, usually in an ad hoc framework of a particular problem. Some recent examples of developments and implementations of GL multigrid solvers appear in [16, 17, 23, 27, 28], while LL applications can be found in, e.g., [4, 7, 11, 12, 13, 18, 19, 20, 21]. For “friendly” problems, both approaches work well and the difference in efficiency is usually not large (e.g., [25, section 5.3]). However, for more difficult problems, the two approaches often exhibit distinct behavior, with GL holding an advantage in some situations and LL in others. We are therefore motivated to develop

---

\*Received by the editors August 24, 2004; accepted for publication (in revised form) June 27, 2005; published electronically March 3, 2006. This research was supported by the Israeli Science Foundation under grant 48/02.

<http://www.siam.org/journals/sisc/28-1/61380.html>

<sup>†</sup>Department of Computer Science, Technion—Israel Institute of Technology, Haifa 32000, Israel (irad@cs.technion.ac.il, gregoryd@cs.technion.ac.il).

methods which will be at least as good as the more suitable of these two approaches, and often better than both.

The algorithm we propose is first motivated and analyzed in section 2, employing a simplified scalar analogy, and then presented in detail in section 3. The algorithm has been applied recently in [6] for the problem of two-dimensional phase unwrapping. However, this was an application-focused paper comparing different methods, with no analysis and just a very brief description of the algorithm.

Multigrid methods for *linear* problems have been studied very extensively. For many problems, robust multigrid methods (such as the classical “algebraic” [3, 22], or “black-box” [8, 9] multigrid) are well known to be effective even in the presence of discontinuous coefficients and irregular domain boundaries. This represents an advantage for GL methods, where the problem solved per iteration is linear. LL approaches, on the other hand, generally require “direct” rediscrization of the nonlinear operator on the coarse grids, which is less robust. On the other hand, the *attraction basin* of efficient GL methods is usually small, which means that slow global search methods must often be applied before the fast GL method can be effective. This may represent a significant disadvantage.

**1.1. Classical approaches.** We first review two classical GL and LL methods. A two-grid framework is used, which is later generalized to a complete multigrid algorithm by recursion. Indices  $h$  and  $H$  denote fine- and coarse-grid operators and functions, respectively. The fine grid  $h$  is the target grid, while grid  $H$  is employed to accelerate convergence. We assume that, for *linear* fine-grid operators, we can construct *robust* coarse-grid approximations on grid  $H$ . By “robust” we mean that, given an appropriate smoother, the multigrid solver exhibits the typical fast and mesh-independent convergence.

Let the fine-grid problem be

$$(1) \quad N_h u_h = f_h,$$

where  $N_h$  is some discrete nonlinear matrix operator,  $u_h$  is the sought solution vector, and  $f_h$  is a given vector. Using iterative methods, it is assumed that we have some current approximation to the solution, denoted  $\tilde{u}_h$ , which we are trying to improve. It is convenient to subtract  $N_h \tilde{u}_h$  from both sides of (1), obtaining the defect equation

$$(2) \quad N_h u_h - N_h \tilde{u}_h = f_h - N_h \tilde{u}_h \equiv r_h.$$

Let  $K_h$  denote a linearization of  $N_h$ . For example, using Newton’s method we define  $K_h$  to be the Jacobian matrix of  $N_h$ , computed for the current approximation:

$$K_h = N'_h[\tilde{u}_h].$$

Then, Newton’s method approximates (2) by

$$(3) \quad K_h(u_h - \tilde{u}_h) \approx r_h.$$

Let  $K_H$  denote the coarse-grid approximation to the linear operator  $K_h$ , which is obtained using a robust linear coarse-grid correction method. Indeed, in our simple analysis below, it is assumed that this coarse-grid operator provides a “perfect” approximation of the linear fine-grid operator (as would be the case if total reduction is used). That is, the two-grid algorithm, with suitable error-smoothing, solves the linear fine-grid problem exactly. Let  $I_h^H$  and  $I_H^h$  denote appropriate fine-to-coarse

(restriction) and coarse-to-fine (prolongation) operators, respectively. Then, after we smooth the error in (3) with a suitable relaxation, we solve the following coarse-grid problem:

$$(4) \quad K_H(u_H - \bar{I}_h^H \tilde{u}_h) = I_h^H r_h,$$

where  $\bar{I}_h^H$  is some restriction operator<sup>1</sup> which may be different from  $I_h^H$ . Once  $u_H$  is computed, the correction is interpolated and added to  $\tilde{u}_h$ :

$$(5) \quad \tilde{u}_h^{new} = \tilde{u}_h + I_H^h(u_H - \bar{I}_h^H \tilde{u}_h).$$

We can optionally follow up with additional relaxation of (3), and this completes one Newton iteration.

Next, we review the classical FAS. First, the error in the fine-grid solution is smoothed using a suitable nonlinear relaxation method (which employs only local linearization). Let  $\hat{N}_H$  denote a nonlinear coarse-grid approximation to  $N_h$ , obtained by rediscrctizing the nonlinear problem on the coarse grid (normally using the same discretization as that of the fine grid; this is called direct discretization). The coarse-grid problem is then

$$(6) \quad \hat{N}_H u_H - \hat{N}_H \bar{I}_h^H \tilde{u}_h = I_h^H r_h.$$

The correction to the fine-grid approximation is added as in (5).

The two approaches are substantially distinct. In the GL method we approximate  $N_h$  by its linearization, which we then proceed to approximate on the coarse grid. Using suitable linear multigrid methods, the latter approximation is excellent, and the overall robustness of the method is therefore determined, loosely speaking, by the accuracy of the linearization, which depends in turn on the properties of the Jacobian and on the closeness of  $\tilde{u}_h$  to  $u_h$ . In the LL approach, on the other hand, we use a nonlinear approximation to  $N_h$  on the coarse grid. If this approximation is sufficiently accurate, the LL algorithm can yield fast convergence even with a rather poor initial guess for the fine-grid solution, thus having an advantage over the GL approach. However, if  $\hat{N}_H$  approximates  $N_h$  poorly, for example due to some nonsmoothness of the coefficients, then the method might not converge even if the initial error is very small and the nonlinearity very weak.

**2. A scalar analogy.** To motivate and analyze our approach, we present an analogy in the form of a simple scalar problem, which contains the essential elements we are studying. This analysis can be generalized to the multivariate vector-valued case, but this would increase the complication without yielding additional insight. Consider the problem of finding a real number  $u$  satisfying

$$(7) \quad N(u) = f,$$

where  $N$  is some twice continuously differentiable nonlinear real-valued function, and  $f$  is a constant which serves to help follow the analogy. For any given  $\rho > 0$ , denote  $D_\rho = \{z : |u - z| < \rho\}$ , and assume  $|N'(z)| \geq K > 0$  for all  $z \in D_\rho$ .

---

<sup>1</sup>Since this is a linear problem, the coarse-grid correction is independent of this transfer operator; the usual choice for linear problems is  $\bar{I}_h^H = 0$ , which yields the so-called correction scheme.

**2.1. FAS.** Denote the current approximation to  $u$  by  $u_n \in D_\rho$ , and the error by  $e_n = u_n - u$ . Suppose that there exists some nonlinear function  $\hat{N}$  which approximates  $N$  and which is easy to invert ( $\hat{N}$  corresponds to  $\hat{N}_H$  in our analogy). Then we can define an iteration process, which is analogous to FAS, by

$$(8) \quad \hat{N}(u_{n+1}) - \hat{N}(u_n) = f - N(u_n) \equiv r_n.$$

Suppose that  $\hat{N}$  is an  $O(\epsilon)$  approximation to  $N$ , where  $\epsilon$  is a positive small parameter. That is,

$$(9) \quad \hat{N}(z) \equiv N(z) + \epsilon\phi(z),$$

for some twice continuously differentiable function  $\phi$ . Substituting (9) into (8), we obtain after rearrangement

$$\epsilon[\phi(u_{n+1}) - \phi(u_n)] = f - N(u_{n+1}) \equiv r_{n+1}.$$

Using the mean-value theorem and the fact that  $f = N(u)$ , we get

$$\epsilon\phi'(\xi_1)(e_{n+1} - e_n) = -N'(\xi_2)e_{n+1},$$

for some  $\xi_1 \in (u_n, u_{n+1})$  and  $\xi_2 \in (u, u_{n+1})$ . Assume  $|\phi'| \leq P < K/2\epsilon$  in  $D_\rho$ . Then

$$(10) \quad \left| \frac{e_{n+1}}{e_n} \right| = \left| \frac{\epsilon\phi'(\xi_1)}{N'(\xi_2) + \epsilon\phi'(\xi_1)} \right| \leq \left| \frac{\epsilon P}{K - \epsilon P} \right| < 1.$$

Thus, the iteration converges to  $u$ , and the asymptotic convergence behavior is given by

$$(11) \quad \lim_{n \rightarrow \infty} \left| \frac{e_{n+1}}{e_n} \right| = \left| \frac{\epsilon\phi'(u)}{N'(u) + \epsilon\phi'(u)} \right| = \epsilon \left| \frac{\phi'(u)}{N'(u)} \right| + O(\epsilon^2) \quad \text{for } \epsilon \rightarrow 0.$$

**2.2. Newton's method.** The Newton iteration for this problem can be written as

$$(12) \quad N'(u_n)(u_{n+1} - u_n) = r_n.$$

For a sufficiently good initial approximation Newton's method converges. In particular, if  $|N''| \leq R$  in  $D_\rho$ , then

$$(13) \quad \left| \frac{e_{n+1}}{e_n} \right| \leq \left| \frac{e_n R}{2K} \right|,$$

and the iteration converges to  $u$  if  $|e_n| < 2K/R$ . The well-known quadratic asymptotic convergence behavior is given by

$$(14) \quad \lim_{n \rightarrow \infty} \left| \frac{e_{n+1}}{e_n^2} \right| = \left| \frac{N''(u)}{2N'(u)} \right|.$$

**2.3. Multilevel nonlinear method (MNM).** Equations (10)–(11) and (13)–(14) demonstrate the advantage and disadvantage of each approach. Newton's method converges extremely fast when we have a good initial approximation ( $|e_n|$  sufficiently small). FAS, on the other hand, will have a large attraction basin if  $\hat{N}$  is a good

approximation to  $N$ , though it will not be useful when  $\hat{N}$  approximates  $N$  poorly. It is generally far less dependent on a good initial approximation than Newton's method.

We now devise a method that has the advantages of both fast asymptotic convergence and a large attraction basin. First, we subtract  $N(u_n)$  from both sides of (7) to obtain the defect equation,

$$(15) \quad N(u) - N(u_n) = r_n.$$

The key idea is to split the left side of (15) into a relatively large linear part and a small nonlinear part and use the  $\hat{N}$  approximation only for the nonlinear part. Accordingly, we add and subtract  $N'(u_n)(u - u_n)$  in the left side of (15). This yields

$$(16) \quad N'(u_n)(u - u_n) + F[N(u), N(u_n), N'(u_n), u, u_n] = r_n,$$

where

$$(17) \quad F[N(u), N(u_n), N'(u_n), u, u_n] = N(u) - N(u_n) - N'(u_n)(u - u_n).$$

Note that (16) is equivalent to the original problem (7), but the nonlinear term  $F$  (which is nothing but the remainder term of a truncated Taylor series) is only of size  $O(|u - u_n|^2)$ , while the first term and the right side are  $O(|u - u_n|)$ . We use the nonlinear approximation only for the small nonlinear part. That is, we define the iteration by

$$(18) \quad N'(u_n)(u_{n+1} - u_n) + F[\hat{N}(u_{n+1}), \hat{N}(u_n), \hat{N}'(u_n), u_{n+1}, u_n] = r_n.$$

Leaving only  $u_{n+1}$  terms on the left side we obtain

$$(19) \quad N'(u_n)u_{n+1} + \hat{N}(u_{n+1}) - \hat{N}'(u_n)u_{n+1} = r_n + N'(u_n)u_n + \hat{N}(u_n) - \hat{N}'(u_n)u_n.$$

Assume that we can easily invert the left side of (19) and solve for  $u_{n+1}$ . To analyze the convergence behavior of this iteration, we first use (9) to obtain, after rearrangement,

$$(20) \quad \epsilon \{ [\phi(u_{n+1}) - \phi(u_n)] - \phi'(u_n)(u_{n+1} - u_n) \} = r_{n+1}.$$

Taylor series expansions now yield

$$(21) \quad \frac{\epsilon}{2} \phi''(\xi_1)(e_{n+1} - e_n)^2 = -e_{n+1}N'(\xi_2)$$

for some  $\xi_1 \in (u_n, u_{n+1})$  and  $\xi_2 \in (u_{n+1}, u)$ . Denote

$$\mu = -\frac{N'(\xi_2)}{\epsilon e_n \phi''(\xi_1)}.$$

(If the denominator here is zero, then we have immediate convergence by (21), so we exclude this case.) Solving the quadratic equation (21) for the error reduction factor we obtain two solutions, the relevant one being

$$(22) \quad \frac{e_{n+1}}{e_n} = \mu + 1 - \text{sign}(\mu) \sqrt{\mu^2 + 2\mu}.$$

TABLE 1

Comparison of convergence behavior for the scalar problem, under the assumptions  $|N'| \geq K > 0$ ,  $|N''| \leq R$ ,  $|\phi'| \leq P$ ,  $|\phi''| \leq Q$  in  $D_\rho$ .

Method	$\lim_{n \rightarrow \infty} \left  \frac{e_{n+1}}{e_n} \right $	$p$	Sufficient for convergence
FAS	$\left  \frac{\epsilon \phi'(u)}{N'(u)} \right $	1	$\left  \frac{2\epsilon P}{K} \right  < 1$
Newton	$\left  \frac{N''(u)}{2N'(u)} \right $	2	$\left  \frac{e_n R}{2K} \right  < 1$
MNM	$\left  \frac{\epsilon \phi''(u)}{2N'(u)} \right $	2	$\left  \frac{2\epsilon e_n Q}{K} \right  < 1$

(We assume here that we know how to choose the relevant solution when we solve (19) for  $u_{n+1}$ . In practice, this is not a problem when the iteration is relevant, since  $u_n$  provides a good initial approximation to the correct  $u_{n+1}$ .) A real solution to (22) exists if  $\mu \notin (-2, 0)$ . Furthermore,  $|e_{n+1}/e_n| < 1$  for  $\mu \notin [-2, 0]$ . That is, the error is reduced by the iteration essentially whenever (19) has a real solution. In particular, if  $|\phi''| \leq Q$  in  $D_\rho$ , then a sufficient condition for convergence of the new algorithm is

$$(23) \quad |\epsilon e_n| < \frac{K}{2Q}.$$

Furthermore, if the iteration converges to  $u$ , then the asymptotic convergence behavior is given by

$$(24) \quad \lim_{n \rightarrow \infty} \left| \frac{e_{n+1}}{e_n^2} \right| = \frac{\epsilon}{2} \left| \frac{\phi''(u)}{N'(u)} \right| + O(\epsilon^2) \quad \text{for } \epsilon \rightarrow 0.$$

We summarize the results of the scalar analysis in Table 1. These results are encouraging inasmuch as they show that the proposed method, which is analogous to the MNM described in detail below, enjoys both the quadratic convergence rate of Newton’s method and the large attraction basin suggested by the  $\epsilon$  coefficient. Indeed, the convergence rate of MNM is similar to the product of the convergence rates of FAS and Newton’s method.

**2.3.1. Backtracking.** Although we expect to see a large attraction basin of the MNM, compared to Newton’s method, there will be occasions where  $\epsilon$  is not sufficiently small to yield convergence for a given initial guess. A well-known method for ensuring global convergence of Newton’s method is backtracking [10]. While a Newton step does not ensure that the residual  $r_n$  will be reduced, the *direction* of the Newton step is always one of descent. Thus, if we *damp* the Newton correction by some sufficiently small positive scalar  $0 < \lambda < 1$ , we are assured that the residual will be reduced. Alternatively, instead of damping the correction step, we can damp the residual, by multiplying the right side of (12) by  $\lambda$ . Since this is a linear equation,

damping the right side is equivalent to damping the correction. We obtain from the damped-residual version of (12)

$$(25) \quad u_{n+1} - u_n = \frac{\lambda r_n}{N'(u_n)}.$$

Therefore we have, by the mean value theorem and (25),

$$(26) \quad \begin{aligned} r_{n+1} &= r_n - N'(\xi)(u_{n+1} - u_n) \\ &= r_n \left( 1 - \lambda \frac{N'(\xi)}{N'(u_n)} \right) \end{aligned}$$

for some  $\xi \in (u_n, u_{n+1})$ . By (25), (26), and the continuity of  $N'$  (and the fact that it is bounded away from zero in  $D_\rho$ ), there always exists a positive  $\lambda_{max}$  such that

$$(27) \quad \left| \frac{r_{n+1}}{r_n} \right| < 1 \quad \text{for every } \lambda \in (0, \lambda_{max}).$$

For MNM, damping the correction is no longer equivalent to damping the residual, since it is a nonlinear method. Indeed, as seen above, there might not even be a solution to the nonlinear iteration, hence no “direction” is defined. However, we show now that MNM also yields a “descent direction” if we backtrack using residual damping. This approach is in the spirit of the NLMG of [14], where residual damping is introduced in conjunction with a compensating overcorrection. Suppose that we multiply  $r_n$  in (16), (18), and (19) by  $\lambda \in (0, 1)$ . We then get from (19), denoting  $\delta = u_{n+1} - u_n$  and using a Taylor series expansion,

$$(28) \quad \frac{1}{2} \hat{N}''(\xi_1) \delta^2 + N'(u_n) \delta = \lambda r_n$$

for some  $\xi_1 \in (u_n, u_{n+1})$ . If  $\hat{N}''(\xi_1) = 0$ , then (25)–(26) hold. Otherwise,  $\delta$  is given by

$$(29) \quad \delta = \frac{N'(u_n)}{\hat{N}''(\xi_1)} \left( -1 \pm \sqrt{1 + 2\lambda r_n \hat{N}''(\xi_1) / N'(u_n)^2} \right).$$

Only the solution with the plus sign in front of the square root is relevant, and we assume that our nonlinear solver can find this solution. (In practice this is not a problem, as  $u_n$  provides an arbitrarily good initial guess for the correct  $u_{n+1}$  as  $\lambda$  is reduced.) We thus have

$$(30) \quad \delta = \frac{\lambda r_n}{N'(u_n)} + O\left(\lambda^2 r_n^2 \hat{N}''(\xi_1) / N'(u_n)^3\right), \quad \lambda r_n \rightarrow 0.$$

Now, similarly to (26),

$$(31) \quad \begin{aligned} r_{n+1} &= r_n - N'(\xi)(u_{n+1} - u_n) \\ &= r_n \left( 1 - \lambda \frac{N'(\xi)}{N'(u_n)} + O\left(\lambda^2 r_n^2 \hat{N}''(\xi_1) / N'(u_n)^3\right) \right) \quad \text{for } \lambda r_n \rightarrow 0, \end{aligned}$$

so here too we can always choose some sufficiently small positive  $\lambda_{max}$  such that (27) holds. Indeed, the residual reduction factor of damped MNM tends to that of the damped Newton step as  $\lambda$  tends to zero. This is expected, as MNM can be viewed as a nonlinear correction to Newton's method, with the correction tending to zero with the damping parameter (cf. section 3.2).

**3. MNM: A multilevel nonlinear method.** We return now to the actual problem and the multigrid algorithm. Let  $\hat{K}_H$  denote the linearization of  $\hat{N}_H$ . Then, analogously to (19), we define the coarse-grid problem as

$$(32) \quad N_H u_H = f_H,$$

with

$$(33) \quad N_H = K_H + \hat{N}_H - \hat{K}_H$$

and

$$(34) \quad f_H = I_h^H r_h + N_H \bar{I}_h^H \tilde{u}_h.$$

Note that  $K_H$  and  $\hat{K}_H$  are analogous to  $N'(u_n)$  and  $\hat{N}'(u_n)$ , respectively. That is, we are assuming that  $K_H$  is an excellent approximation to  $K_h$ , while  $\hat{K}_H$  might not be as good. This makes sense because, as noted above, there are well-known methods that provide excellent coarse-grid approximations to linear fine-grid operators, whereas direct discretization often does not when coefficients are nonsmooth.

**3.1. The MNM algorithm.** The multigrid algorithm, which we describe next, is obtained from the two-grid algorithm by the usual recursion. Specific details, such as the relaxation method, intergrid transfers, and algorithmic parameters, can vary. In sections 4.1 and 4.2 we specify the particular choices used in our numerical tests.

A hierarchy of grids is defined and numbered 0 to  $M$ , where grid 0 is the target grid, which is the finest one, and grid  $M$  is the coarsest. On each level (grid)  $j$  we solve approximately a problem of the form

$$(35) \quad N_j u_j + L_j u_j = f_j,$$

where  $u_j$  is the vector of unknowns,  $f_j$  is a given vector,  $N_j$ —like  $\hat{N}_H$  above—is a grid- $j$  discretization of  $N$  (as in the classical FAS algorithm), and  $L_j$  is a linear correction operator (similar to  $K_H - \hat{K}_H$  above). We specify the usual integer parameters,  $\nu_1$  and  $\nu_2$ , which are the number of relaxation sweeps carried out before and after the coarse-grid correction, respectively, and the cycle index  $\gamma$ , which specifies the number of recursive calls (1 for a so-called V cycle, 2 for a W cycle, and so on.)

Initially,  $u_0$  is set to some approximation (which is normally obtained using the so-called full multigrid (FMG) approach [1]), and  $L_0$  is the zero matrix of appropriate size. Then, the MNM cycle, described below, is called repeatedly for  $j = 0$ , until some convergence criterion is met.

```

function  $u_j = \text{MNM}(u_j, N_j, L_j, f_j, j, \gamma, \nu_1, \nu_2, M)$ 
If ( $j < M$ )      ** On every grid but the coarsest **
{
  Relax  $\nu_1$  times the equation,
     $N_j u_j + L_j u_j = f_j$ 
  Compute the residual,
     $r_j = f_j - (N_j u_j + L_j u_j)$ 
  Construct the linearized operator,
     $K_j = L_j + N'_j[u_j]$ 
  Initialize the coarse-grid solution,
     $u_{j+1} = \bar{I}_j^{j+1} u_j$ 
  Compute the robust coarsening of the linearized operator,
     $K_{j+1} = I_j^{j+1} K_j I_{j+1}^j$ 
  Compute  $L_{j+1}$  by subtracting off the linearization of  $N_{j+1}$ ,
     $L_{j+1} = K_{j+1} - N'_{j+1}[u_{j+1}]$ 
  Compute the coarse-grid right side,
     $f_{j+1} = I_j^{j+1} r_j + N_{j+1} u_{j+1} + L_{j+1} u_{j+1}$ 
  Call MNM recursively (repeat  $\gamma$  times),
     $u_{j+1} = \text{MNM}(u_{j+1}, N_{j+1}, L_{j+1}, f_{j+1}, j + 1, \gamma, \nu_1, \nu_2, M)$ 
}
else      ** On the coarsest grid **
{
  Solve approximately the equation,
     $N_M u_M + L_M u_M = f_M$ 
  Return  $u_M$ 
}
end if

Interpolate and add the correction,
   $u_j = u_j + I_{j+1}^j (u_{j+1} - \bar{I}_j^{j+1} u_j)$ 
Relax  $\nu_2$  times the equation,
   $N_j u_j + L_j u_j = f_j$ 
Return  $u_j$ 

```

The restriction and prolongation operators, denoted here by  $I_j^{j+1}$  and  $I_{j+1}^j$ , respectively, are chosen so as to yield a robust coarse-grid approximation of the linear operator,  $K_j$ . These can be any of several matrix-dependent transfer operators that have been developed over the years. In our two-dimensional (2D) tests we use the black-box multigrid method of [8]. For the restriction of  $u_j$  we use the operator  $\bar{I}_j^{j+1}$ , which can be simple injection or some local averaging. Observe that the combined operator,  $N_j + L_j$ , is nothing but an approximation on level  $j$  of the fine-grid operator,  $N_0$ . Thus, the relaxation methods used in the classical FAS algorithm, such as Gauss–Newton, are suitable here as well. It is important to note that, since we are combining two different approximations to the fine-grid operator (for example, when computing the linear correction,  $L_j$ ), they need to be scaled compatibly. We explain this issue and demonstrate the algorithm with an explicit simple example in the next subsection. (See also comments regarding scaling in section 4.1.)

**3.1.1. A concrete example.** By way of illustration, we now show an explicit example of the MNM cycle with two grids. Capital subscripts correspond to coarse-grid values and operators. Consider the one-dimensional (1D) nonlinear problem,

$$(36) \quad Nu = -u_{xx} - Ce^u = f,$$

with some appropriate boundary conditions, which are not important for this discussion. Here,  $C$  is a positive constant and  $f(x)$  is given. (With  $f = 0$  and zero Dirichlet boundary conditions, this is the classical Bratu problem.) Standard second-order finite-difference discretization on a grid of mesh size  $h$  produces a nonlinear system,

$$(37) \quad N_h u_h = f_h,$$

with the  $i$ th row of  $N_h u_h$  given by

$$(38) \quad (N_h u_h)_i = h^{-2} [-u_{i-1} + 2u_i - h^2 C e^{u_i} - u_{i+1}].$$

Here and below, we omit the  $h$  subscript when using subscripts to denote row number (equivalently, mesh point). Let  $\tilde{u}_h$  denote the current approximation to  $u_h$ . Employing Newton linearization, we obtain  $K_h = N'_h(\tilde{u}_h)$ , where the  $i$ th row of  $K_h u_h$  can be written as

$$(39) \quad (K_h u_h)_i = h^{-2} [-u_{i-1} + u_i(2 - \sigma_i) - u_{i+1}],$$

with

$$\sigma_i = h^2 C e^{\tilde{u}_i}.$$

Evidently,  $K_h$  approximates the linear operator,

$$K = -\partial_{xx} - C e^{\tilde{u}}.$$

Assume standard coarsening, with coarse mesh point  $I$  coinciding with fine mesh point  $i = 2I$ . As in FAS, we need to transfer the current approximation,  $\tilde{u}_h$ , to the coarse grid and construct the nonlinear coarse-grid operator,  $\hat{N}_H$ , where  $H = 2h$  denotes the coarse mesh interval. Assume we use injection for this transfer,

$$\tilde{U}_H = \bar{I}_h^H \tilde{u}_h,$$

with  $\tilde{U}_I = \tilde{u}_i$ . The operator  $\hat{N}_H$  is obtained by rediscrretization on the coarse grid. However, we introduce a multiplicative scaling, to be discussed below. The  $I$ th row of  $\hat{N}_H U_H$  is thus given by

$$(40) \quad (\hat{N}_H U_H)_I = B_I H^{-2} [-U_{I-1} + 2U_I - H^2 C e^{U_I} - U_{I+1}],$$

where  $B_I$  is the scaling. The linearization of  $\hat{N}_H$  is denoted by  $\hat{K}_H$ , with the  $I$ th row of  $\hat{K}_H U_H$  given by

$$(41) \quad (\hat{K}_H U_H)_I = B_I H^{-2} [-U_{I-1} + U_I(2 - \Sigma_I) - U_{I+1}],$$

with

$$\Sigma_I = H^2 C e^{\tilde{U}_I} = 4\sigma_i.$$

Next, we turn to the construction of the robust coarse-grid approximation to  $K_h$ . The appropriate prolongation operator (see discussion in section 4) is such that the interpolated function,  $v_h = I_H^h U_H$ , satisfies the homogeneous equation,  $(K_h v_h)_i = 0$ , at odd  $i$ :

$$(42) \quad (I_H^h U_H)_i = \begin{cases} U_{i/2} & \text{if } i \text{ is even,} \\ \frac{U_{(i-1)/2} + U_{(i+1)/2}}{2 - \sigma_i} & \text{if } i \text{ is odd.} \end{cases}$$

As the problem is symmetric, we choose the restriction to be the transpose of the prolongation. Therefore, the Galerkin coarsening rule yields the robust coarse-grid approximation to  $K_h$ :

$$K_H = (I_H^h)^T K_h I_H^h.$$

Observe that the restriction,  $I_H^h$ , appears as a factor both in  $K_H$  and in the right-hand side of the coarse-grid equation (since it is used to restrict the residuals). Hence, its scaling is unimportant with respect to  $K_H$ , since it cancels out. In contrast, when we construct  $\hat{K}_H$  by direct discretization, we are implicitly assuming that the rows of the restriction sum up to one (hence, that a constant function remains invariant when restricted to the coarse grid). Since in MNM we combine the different operators, we must therefore scale  $K_H$  and  $N_H$  for consistency by multiplying them by the row-sums of the restriction operator. For  $I = i/2$ , this yields

$$B_I = 1 + \frac{1}{2 - \sigma_{i-1}} + \frac{1}{2 - \sigma_{i+1}} = \frac{2 + \delta^- + \delta^+ - \sigma_i}{2 - \sigma_i},$$

where

$$\delta^\pm = \frac{2 - \sigma_i}{2 - \sigma_{i\pm 1}}.$$

For the Galerkin coarse-grid linear operator, an explicit computation yields

$$(43) \quad (K_H U_H)_I = B_I \left( \frac{4}{2 + \delta^- + \delta^+ - \sigma_i} \right) H^{-2} \{ -\delta^- U_{I-1} + [-\delta^- - \delta^+ + (2 - \sigma_i)^2] U_I - \delta^+ U_{I+1} \}.$$

The linear correction to  $\hat{N}_H$  is given by  $L_H = K_H - \hat{K}_H$ , and the complete MNM coarse-grid operator is given by

$$(44) \quad N_H = \hat{N}_H + L_H = N_H + K_H - \hat{K}_H,$$

with  $\hat{N}_H$ ,  $K_H$ , and  $\hat{K}_H$  given by (40), (43), and (41), respectively.

To get an impression of the character of  $L_H$ , observe that if  $u_h$  is smooth (that is, undergoes a relative change of order  $h$  from one grid point to the next), then  $\delta^\pm = 1 + O(h^3)$  (because  $\sigma_i$  itself is of order  $h^2$ ). Substituting 1 for  $\delta^\pm$ , we obtain by a straightforward computation

$$(L_H U_H)_I = B_I H^{-2} \frac{\sigma_i}{4 - \sigma_i} (-U_{I-1} + 2U_I - U_{I+1}).$$

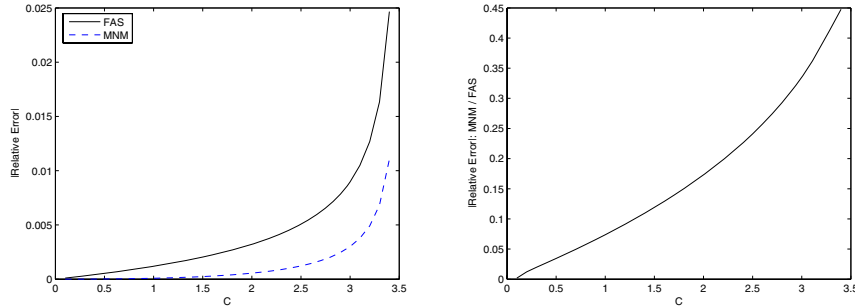


FIG. 1. The maximal relative difference between the fine-grid solution to the 1D Bratu problem and two coarse-grid solutions, one using  $\hat{N}_H$  (FAS) and the other using  $N_H$  (MNM). The relative differences appear on the left and their ratio on the right.

Note that the correction  $L_H$  increases somewhat the relative size of the Laplacian in the nonlinear operator. This turns out to compensate for the slight extra “stiffness” of the coarse grid compared to the fine grid. Thus, it improves the nonlinear coarse-grid approximation to the fine-grid operator, as demonstrated below. Observe that the linear correction is normally very small, since  $K_H$  and  $\hat{K}_H$  are both  $O(h^2)$  approximations to the scaled  $K$ . It becomes large only when  $\sigma_i \sim 1$ . Furthermore,  $L_H$  is “very slightly” nonsymmetric due to the spatially varying scaling factor,  $B_I$ . This may become significant on coarse grids. We may choose to retain the symmetry by fixing the scaling factor to be some constant nominal value.

To demonstrate the effect of the linear correction,  $L_H$ , we perform a simple numerical test. We set  $f$  and the boundary conditions to zero, and the domain to  $(0, 1)$ , and solve the problem on a grid of 16 mesh intervals—our fine grid—for a range of positive  $C$  values. It is well known that solutions for the differential problem exist only for  $C$  that is below some critical value, and the problem becomes more and more difficult as  $C$  approaches this value. Next, we approximate and solve the same problem on a coarse grid of eight intervals, once using  $\hat{N}_H$  (FAS) and once using  $N_H$  (MNM), employing the previously determined fine-grid solution to compute  $L_H$ . We compare in Figure 1 the maximal difference between the fine-grid solution and the two coarse-grid solutions. Evidently, the MNM coarse-grid operator provides a much better approximation. The relative gain in accuracy is very large when the parameter  $C$  is small (because the linearizations then yield excellent approximations), and it becomes smaller and smaller, but still quite significant, as the critical turning point is approached.

**3.1.2. Computational cost.** The number of operations per V cycle is proportional to the number of unknowns in all three algorithms—FAS, Newton, and MNM. The actual cost per cycle for each of the methods depends on the problem and on the implementation. FAS requires evaluating the nonlinear operator for every relaxation sweep and for residual computation. Newton’s method requires only one linearization per cycle, which is comparable in cost to an evaluation of the nonlinear operator, and the resulting linear operator can then be stored. On the other hand, after each such linearization, a setup of the robust linear solver with its matrix-dependent intergrid transfers and (Petrov–)Galerkin coarse-grid operators needs to be applied. The MNM algorithm is essentially a combination of the two, starting from the second-finest grid. An MNM cycle is therefore more expensive than either an FAS or a Newton iteration,

but it is less expensive than the sum of the two. In particular, on the finest grid, where normally most of the work is performed, the MNM and FAS algorithms are identical because there is no linear correction term.

**3.1.3. Backtracking.** We introduce an optional backtracking procedure into the basic MNM algorithm, both for the local (scalar, as we use nonlinear pointwise Gauss–Seidel relaxation) nonlinear problems encountered in the relaxation, and for the coarse-grid correction. In the local problems we damp the Newton-step corrections in the classical manner, while for the coarse-grid corrections we backtrack via residual damping as described in subsection 2.3.1. We basically use the classical backtracking procedure of [10] with the following changes: (1) We do not compute the condition number and redefine the problem for poorly conditioned problems; (2) We limit the number of attempts to reduce the residual by backtracking; if unsuccessful, the correction is simply discarded. The reason for these changes is that we do not want to spend much effort to reduce the residual of coarse-grid problems when this is unlikely to contribute much to the convergence of the fine-grid problem. The greatest difficulties occur on the coarser grids (because, in our analogy,  $\epsilon$  is larger there), and it is often not cost-effective to insist on residual reduction on the very coarse grids, where there might not even be a solution in some cases. As the fine-grid residual is reduced, the convergence on all the grids normally becomes much easier, consistent with our analysis.

The backtracking procedure naturally adds to the cost of the algorithm, but this increase is rarely substantial. In the relaxation, the main increase normally comes from the need to recompute the residuals of the local problems to check at which of the equations (if any) backtracking is required. Usually, there are few such locations, if any, and the backtracking procedure is implemented only for those equations. For the coarse-grid correction process (which backtracks by residual damping), there is a potential for a greater increase in cost, because if the coarse-grid correction fails to reduce the residual norm then we discard it and recompute the coarse-grid correction with a damped residual, etc. As we shall show, the actual increase in the operation count in our tests was small.

The measure we use to describe the actual complexity of the MNM algorithm with backtracking is the *effective cycle index*,  $\bar{\gamma}$ . Recall that the cycle index  $\gamma$  is the number of recursive calls to function MNM in the standard algorithm (with no backtracking). Hence, the number of visits to grid  $j$  is  $\gamma^j$ . Due to the backtracking, the number of visits may be larger, resulting in an effective cycle index that is larger than the nominal one. Suppose that the actual total number of relaxation sweeps performed on grid  $j$  in a particular computation, divided by the number of sweeps carried out on the finest grid, is  $k_j$ , and that the number of variables on grid  $j$  is  $n_j$ . Then the effective cycle index is naturally defined implicitly as the only positive solution of

$$(45) \quad \sum_{j=0}^M n_j \bar{\gamma}^j = \sum_{j=0}^M k_j n_j.$$

In particular, if  $n_j/n_{j+1} \approx \tau > 1$ , where the coarsening ratio  $\tau$  is independent of  $j$ , then

$$(46) \quad \bar{\gamma} \approx \frac{\sum_{j=0}^{M-1} k_{j+1} \tau^{-j}}{\sum_{j=0}^M k_j \tau^{-j}}.$$

In this case, the bound (2.4.14) of [25, p. 51] for the total computational work per cycle holds approximately: if the fine-grid work is  $Cn_0$ , then the total work is approximately

$$(47) \quad W \approx \frac{\tau C n_0}{\tau - \bar{\gamma}} \quad \text{for } \bar{\gamma} < \tau.$$

In a parallel computing environment, the standard backtracking procedure is relatively more troubling, since it is done sequentially. In this case we would prefer to compute in parallel several coarse-grid solutions—with different residual-damping parameters ( $\lambda = 1, 1/2, 1/4, \dots$ , for example)—and choose the correction with the largest damping parameter which still yields a reduction of the residual norm, discarding the rest.

**3.2. Adaptive MNM.** An interesting alternative (or addition) to backtracking, is a more general adaptive procedure. Consider the MNM coarse-grid operator  $N_H$  of (33). Two points of view suggest themselves: (a) The first term,  $K_H$ , is the main term, while the second and third terms comprise a nonlinear correction for the defect which results from approximating  $N_h$  by the usual Newton linearization; (b) The second term,  $\hat{N}_H$ , is the main term, while the first and third terms form a linearized correction for the defect resulting from the nonrobustness of the FAS approximation. Normally, these corrections improve the coarse-grid approximation. Suppose, however, that there is somewhere a discontinuity in a coefficient. This may show up as a large linear correction,  $K_H - \hat{K}_H$ , because  $\hat{K}_H$  is no longer a good approximation to  $K_h$ . This indicates that  $\hat{N}_H$  itself is a poor approximation to  $N_h$ , and therefore we might be better off not using the FAS contribution. We should then damp (perhaps completely) the *nonlinear* correction, reducing to Newton’s method. On the other hand, suppose that due to some nonlinear near-singularity the linearization provides a poor approximation. This would show up as a large nonlinear correction. We might then be better off using FAS in this case, and we should damp the *linear* correction. Accordingly, we introduce two nonnegative parameters,  $a$  and  $b$ , and redefine  $N_H$  as

$$(48) \quad N_H = aK_H + b\hat{N}_H + (1 - a - b)\hat{K}_H.$$

We thus obtain MNM for  $a = b = 1$ , Newton’s method for  $a = 1, b = 0$ , and FAS for  $a = 0, b = 1$ . In fact,  $a$  and  $b$  can be diagonal matrices instead of scalars, so that the damping is chosen locally.

Although the details are quite different, this use of parameters to obtain favorable “compromises” between different types of coarse-grid operators is in the spirit of the nonlinear multigrid approach of [14]. There, the coarse-grid operator varies from that of FAS—when the (multiplicative) parameter is equal to one, to an operator that is linearized around the initial coarse-grid approximate solution—when it is equal to zero. This effect is somewhat similar to what we obtain in the present approach as we vary  $(a, b)$  from  $(0, 1)$  to  $(1, 0)$  if we use a fixed-point linearization.

We applied a version of adaptive MNM in [6], fixing  $b = 1$  and optimizing  $a \in [0, 1]$  over the convergence factor at very coarse grids. This approach proved quite useful, but we expect that a more refined analysis will allow us to yield still greater dividends from the adaptive approach. We do not pursue this further here; however, in section 4.2 we demonstrate the potential of this approach by manually selecting appropriate values of  $a$  and  $b$  for problems which prove to be difficult for the standard MNM, even with backtracking.

#### 4. Numerical tests.

**4.1. 1D tests.** We first test our algorithm on the following nonlinear 1D diffusion problem:

$$(49) \quad \begin{aligned} Nu \equiv -[g(x, u, u_x)u_x]_x &= f(x), & x \in (0, 1), \\ u(0) &= u_\ell, \\ u(1) &= u_r, \end{aligned}$$

where  $u$  is the unknown function, and  $g, f, u_\ell, u_r$  are given. Discretizing by second-order accurate finite differences on a uniform grid  $x_h = \{x_i\}_{i=0}^n$ , with mesh size  $= 1/n$ , we obtain the nonlinear system

$$(50) \quad N_h u_h = f_h,$$

where the  $i$ th equation in the system is given by

$$(51) \quad -\frac{g_{i+1/2}(x_h, u_{i+1}, u_i)[u_{i+1} - u_i] - g_{i-1/2}(x_h, u_i, u_{i-1})[u_i - u_{i-1}]}{h^2} = f_i, \\ i = 1, \dots, n-1.$$

Here,  $u_i$  and  $f_i$  are the  $i$ th components of  $u_h$  and  $f_h$ , respectively, and  $g_{i\pm 1/2}$  is some discrete approximation to  $g$  midway between grid points  $i$  and  $i \pm 1$ . The boundary conditions yield  $u_0 = u_\ell$  and  $u_n = u_r$ . In our test problems we define  $g$  to correspond to the so-called van Genuchten curve modeling hydraulic conductivity of unsaturated soils [26]. This problem, in two dimensions, has been studied extensively by Starke in [23, 24] employing a least-squares finite elements approach with, in the latter paper, a GL multilevel approach, and by Korsawe and Starke in [18], where three LL methods (variants of FAS) are compared.

Given constant parameters  $k_s$ ,  $\alpha$ , and  $p$ , we have

$$(52) \quad g = g(u) = \begin{cases} 1, & u \geq 0, \\ k_s \psi(|u|), & u < 0, \end{cases}$$

where

$$(53) \quad \psi(\theta) = [1 + (\alpha\theta)^p]^{-\frac{q}{2}} \left( 1 - \frac{(\alpha\theta)^{p-1}}{[1 + (\alpha\theta)^p]^q} \right)^2,$$

with  $q = 1 - 1/p$ . To represent  $g_{i\pm 1/2}$  we compute  $g(u_i)$  and  $g(u_{i\pm 1})$  and use simple averaging. For a given domain and boundary conditions, this problem generally becomes more difficult for smaller  $p$  and larger  $\alpha$ . For  $p < 2$ ,  $g'$  fails to satisfy a Lipschitz condition at  $u = 0$ , and  $g(u)$  actually tends to a step function for  $p \rightarrow 1^+$ . For  $\alpha$  that is not small compared to one, the solution exhibits a thin boundary layer near  $x = 0$  (in our example), which narrows quickly as  $\alpha$  grows. The boundary conditions are  $u_\ell = -2$ ,  $u_r = 1$ , and we set  $f = 0$ . The initial guess for the fine-grid solution is the linear profile which satisfies the boundary conditions. We perform  $V$  cycles until the  $\ell_2$  residual norm is reduced to at most  $10^{-8}$  times its initial value. We backtrack if the coarse-grid correction plus the (single) postrelaxation sweep has failed to reduce the  $\ell_2$  norm of the residual. In this case, we discard the coarse-grid correction, damp the residual according to the strategy of [10], and compute a new coarse-grid correction.

For error smoothing we use Gauss-Seidel relaxation in red-black ordering, whereby we first relax at the even-indexed points and then at the odd-indexed points. To

describe the prolongation operator, let the  $n_j$  by  $n_j$  matrix,  $K_j$ , denote the linearized operator on grid  $j$ . Then the prolongation operator is a matrix  $P$  (denoted  $I_{j+1}^j$  above) of size  $n_j$  by  $n_{j+1}$ , with

$$(54) \quad P_{i,k} = \begin{cases} 1 & \text{if } i = 2k, \\ -\frac{K_{i,i\mp 1}}{K_{i,i}} & \text{if } i = 2k \pm 1, \\ 0 & \text{otherwise.} \end{cases}$$

The restriction operator (denoted  $I_j^{j+1}$  above) is a matrix  $R$  of size  $n_{j+1}$  by  $n_j$ , whose  $(j, i)$ th element is given by (54), with  $K_j$  replaced by its transpose. With these choices and the construction of the coarse-grid operator by Petrov–Galerkin coarsening,  $K_j$  remains a tridiagonal matrix on all the grids. Furthermore, for linear problems, the multigrid cycle is an exact solver (equivalent to total reduction), provided that *either*  $\nu_1$  or  $\nu_2$  is nonzero and the problem on the coarsest grid is solved exactly. This is a well-known result of the fact that, after we relax on the fine grid at the odd-indexed points—leaving there zero residuals—the remaining error is in the range of the prolongation defined by (54).

As explained in subsection 3.1.1, it is important to scale the different coarse-grid approximations which are added/subtracted to be compatible. The usual FAS algorithm assumes that rows of the restriction matrix sum up to 1. Therefore, in the MNM algorithm of section 3.1, the correct scaling of  $N_{j+1}$  (hence also  $N_{j+1}'$ ) is by left-multiplication by a diagonal matrix whose elements are the row-sums of the restriction matrix,  $I_j^{j+1}$ . Such a rescaling can result in the loss of symmetry. We choose to simply scale by the constant 2, which is correct for the standard full-weighting, and is an excellent approximation on all the grids for the actual restriction.

In all our tests the finest grid is 256 intervals. For the nonlinear algorithms (MNM and FAS) we use seven multigrid levels, coarsening down to a grid of three interior variables. On the coarsest grid we relax 10 times. We use  $\gamma = 1$  (V cycle) and  $\nu_1 = \nu_2 = 1$ . The FAS algorithm is the standard one (except that we also allow backtracking), with injection for  $\bar{I}_h^H$  (local averaging is sometimes better for FAS, but not dramatically so in our tests).

We show results for physical parameters  $\alpha = 0.5, 1.0$ , and  $p = 2.5, 1.8$ , in Figures 2, 3, 4, and 5, with NWT denoting Newton’s method. In Figure 2, it is evident that MNM indeed behaves as predicted by the analysis. Its asymptotic convergence rate is similar to that of NWT, but it gets there much sooner. In the moderate case (top left),  $\alpha = 0.5$ ,  $p = 2.5$ , all three methods perform well. For  $\alpha = 1.0$ ,  $p = 2.5$  (bottom left) there is a rather thin boundary layer near  $x = 0$ , as seen in Figure 3. This nonsmooth solution causes FAS convergence to slow down significantly, while the other two methods remain effective. For  $\alpha = 0.5$ ,  $p = 1.8$  (top right), on the other hand, the solution is rather smooth but there is a strong singularity in the derivative of  $g$  at  $u = 0$  (see Figure 5). As a result, the linearization yields a very poor approximation at that point, and NWT stagnates, but FAS is hardly affected. MNM has a slower start, but soon regains its fast convergence. Here, it is evident that MNM could gain much from an adaptive procedure as outlined in section 3.2, and implemented in [6]. For  $\alpha = 1.0$ ,  $p = 1.8$  (bottom right), there is both a boundary layer in  $u$  and a singularity of  $g'$ , hence both FAS and NWT suffer, yet MNM remains very effective. In more extremely singular cases (smaller  $p$ ), FAS becomes better than MNM due to the extreme poorness of the linear approximation. In such cases, adaptive MNM would be highly advantageous.

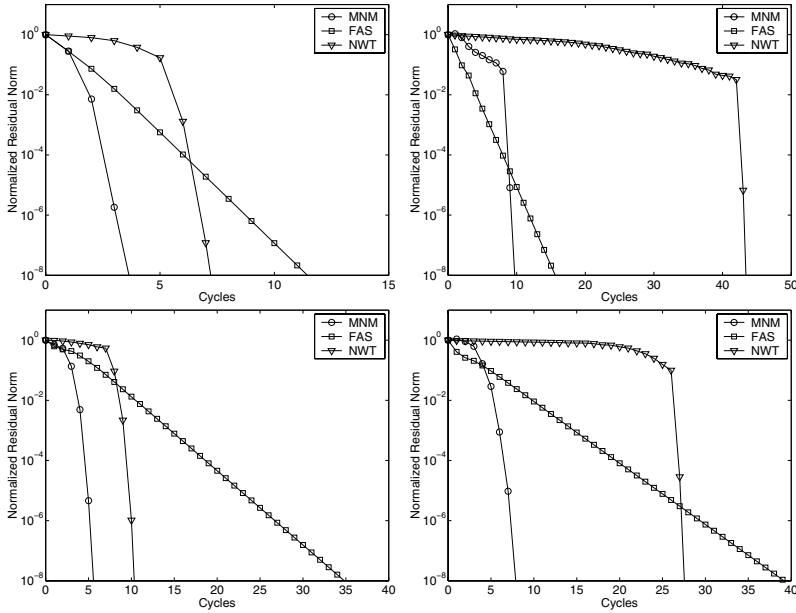


FIG. 2. Residual norm convergence histories, normalized by the initial value, are shown for  $\alpha = 0.5$ ,  $p = 2.5$  (top left);  $\alpha = 0.5$ ,  $p = 1.8$  (top right);  $\alpha = 1.0$ ,  $p = 2.5$  (bottom left);  $\alpha = 1.0$ ,  $p = 1.8$  (bottom right).

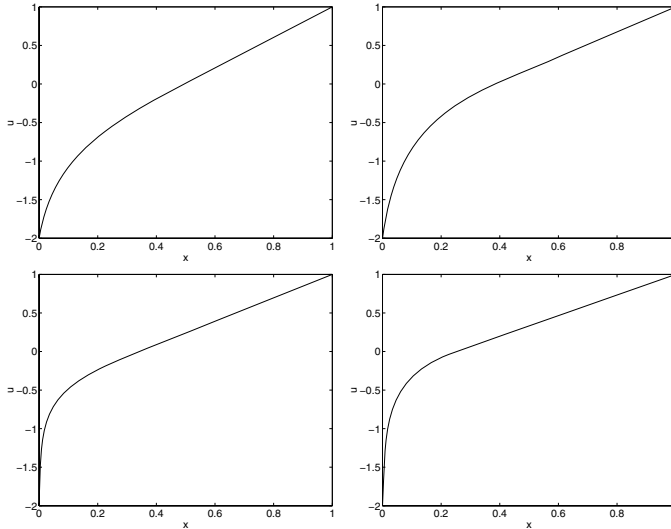


FIG. 3. The numerical solution is shown for  $\alpha = 0.5$ ,  $p = 2.5$  (top left);  $\alpha = 0.5$ ,  $p = 1.8$  (top right);  $\alpha = 1.0$ ,  $p = 2.5$  (bottom left);  $\alpha = 1.0$ ,  $p = 1.8$  (bottom right).

In the examples where  $p = 2.5$ , MNM used no backtracking, so  $\bar{\gamma} = \gamma = 1$ . For  $\alpha = 1.0$ ,  $p = 1.8$ , we obtained  $\bar{\gamma} = 1.11$ , indicating an average increase of about 12% in the number of floating-point operations, compared to  $V$  cycles, by (47). For  $\alpha = 0.5$ ,  $p = 1.8$ , we observed  $\bar{\gamma} = 1.19$ , implying a 23% average increase in computational cost, but there were also two cases of backtracking on the finest grid, so the overall work for MNM is about 35% more than is suggested by Figure 2.

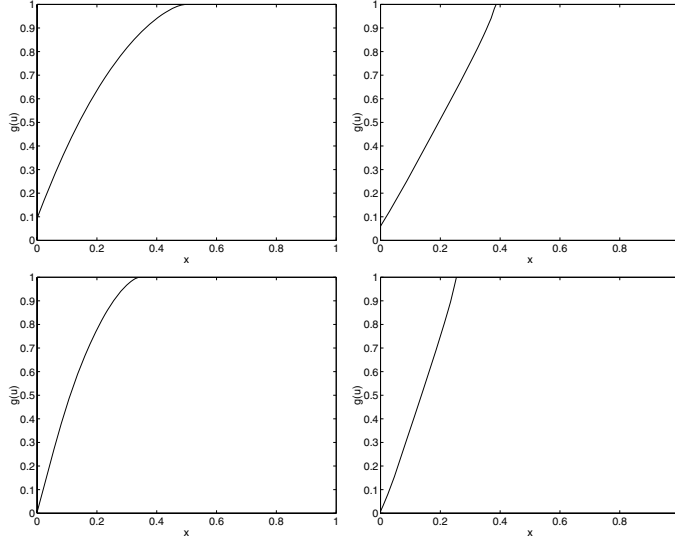


FIG. 4. The diffusion coefficient  $g$ , computed for the numerical solutions at gridpoints, is shown for  $\alpha = 0.5$ ,  $p = 2.5$  (top left);  $\alpha = 0.5$ ,  $p = 1.8$  (top right);  $\alpha = 1.0$ ,  $p = 2.5$  (bottom left);  $\alpha = 1.0$ ,  $p = 1.8$  (bottom right).

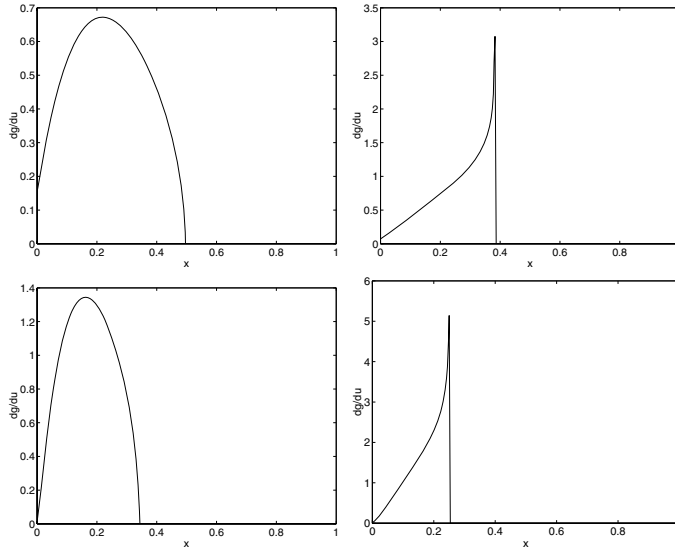


FIG. 5. The derivative  $g'(u)$ , computed for the numerical solutions at gridpoints, is shown for  $\alpha = 0.5$ ,  $p = 2.5$  (top left);  $\alpha = 0.5$ ,  $p = 1.8$  (top right);  $\alpha = 1.0$ ,  $p = 2.5$  (bottom left);  $\alpha = 1.0$ ,  $p = 1.8$  (bottom right).

**4.2. 2D tests.** We test the 2D version of the problem of section 4.1,<sup>2</sup>

$$(55) \quad N(u) \equiv -\nabla \cdot (g(u)\nabla u) = f(x, y), \quad x, y \in (0, 1) \times (0, 1),$$

again with Dirichlet boundary conditions, where  $u$  is the unknown function,  $g$  is given by (52), and  $f$  is a given function (zero in our tests). Discretizing by second-order

<sup>2</sup>See more 2D tests, including discontinuous coefficients, in [6].

accurate finite volumes on a uniform grid with mesh size  $h$ , we obtain the nonlinear system

$$(56) \quad N_h u_h = f_h,$$

where the  $i$ th equation in the system is given by

$$(57) \quad \begin{aligned} & \frac{g_{i+1/2,j}(u_{i+1,j}, u_{i,j})[u_{i+1,j} - u_{i,j}] - g_{i-1/2,j}(u_{i,j}, u_{i-1,j})[u_{i,j} - u_{i-1,j}]}{h^2} \\ & - \frac{g_{i,j+1/2}(u_{i,j+1}, u_{i,j})[u_{i,j+1} - u_{i,j}] - g_{i,j-1/2}(u_{i,j}, u_{i,j-1})[u_{i,j} - u_{i,j-1}]}{h^2} \\ & = f_{i,j} \quad i = 1, \dots, n-1; j = 1, \dots, n-1. \end{aligned}$$

As in the 1D problem,  $u_{i,j}$  and  $f_{i,j}$  are the components of  $u_h$  and  $f_h$ , respectively, while  $g_{i\pm 1/2,j}$  and  $g_{i,j\pm 1/2}$  are averaged approximations to  $g$  midway between neighboring grid points:  $(i, j)$  and  $(i \pm 1, j)$  or  $(i, j)$  and  $(i, j \pm 1)$ , respectively. The boundary conditions are described below.

The tests were performed with three cases of boundary conditions. The initial guesses for the three cases are shown in Figure 6. In the first case the boundary conditions were defined to be the boundaries of an inclined linear plane, with its upper side at  $u = 1$  and its lower side at  $u = -2$ . In the second case the boundary conditions were set to  $u = -2$  along two adjacent sides, while the boundaries along the other two sides were set to linear profiles with  $u = -2$  at their lower points and  $u = 1$  at their common top point. In the third case the boundary values were set to  $u = -1$  and  $u = 1$  along two opposite sides, while along the other sides the values had the shape of a sine function, varying from  $-1$  to  $1$ . The right-hand side function was set to  $f = 0$  in all the examples.

We perform V cycles until the  $\ell_2$  residual norm is reduced by a factor of  $10^{-8}$ . We backtrack if the coarse-grid correction plus the (single) postrelaxation sweep fails to reduce the  $\ell_2$  norm of the residual. In this case, we discard the coarse-grid correction, damp the residual according to the strategy of [10], and compute a new coarse-grid correction.

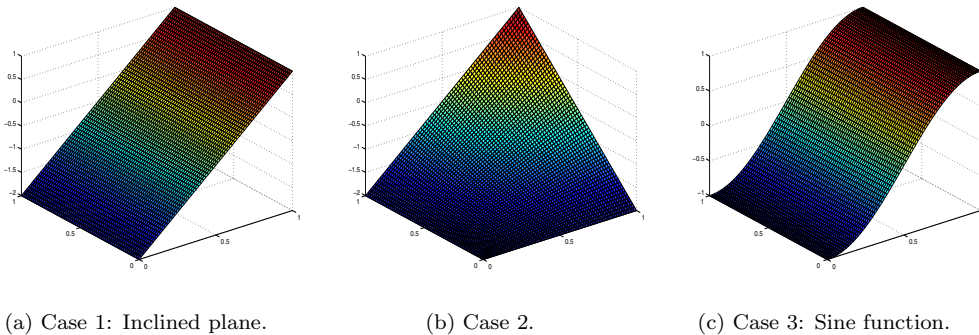


FIG. 6. *Boundary conditions and initial guesses.*

TABLE 2

Test results for Case 1. The \* superscript denotes that backtracking was required. For this case, setting  $a = 0.3$ ,  $b = 0.5$  yielded for MNM an average convergence factor of 0.44 with no backtracking.

$P \setminus \alpha$	.5	.75	1
1.5	MNM = 0.18 FAS = 0.26 NLMG = 0.25 FP = 0.50	MNM = 0.26 FAS = 0.35 NLMG = 0.35 FP = 0.64	MNM = 0.97* FAS = 0.64 NLMG = 0.64 FP = 0.77
2	MNM = 0.13 FAS = 0.23 NLMG = 0.20 FP = 0.39	MNM = 0.25 FAS = 0.33 NLMG = 0.31 FP = 0.58	MNM = 0.38 FAS = 0.41 NLMG = 0.40 FP = 0.73
2.5	MNM = 0.12 FAS = 0.21 NLMG = 0.16 FP = 0.35	MNM = 0.25 FAS = 0.33 NLMG = 0.26 FP = 0.58	MNM = 0.41 FAS = 0.42 NLMG = 0.40 FP = 0.76

For error-smoothing we use a nonlinear Gauss–Seidel relaxation in red-black ordering. The local nonlinear problems of the relaxation are solved approximately by one Newton iteration. We use Dendy’s classical black-box multigrid [8] to construct prolongation and restriction operators. Alternatives can of course be found in the literature, all essentially approximate generalizations of (54) to two dimensions.

In all our tests the finest grid has 64 intervals in each direction, while the coarsest grid has 4 intervals in each direction, yielding 5 multigrid levels, coarsening down to a grid of 3 by 3 interior variables. On the coarsest grid we relax 5 times. We use  $\gamma = 1$  (V cycle) and  $\nu_1 = \nu_2 = 1$ . For the linearization in the GL and MNM algorithms, we use a fixed-point (FP) approach (sometimes called “lagged diffusion” or “principal-part linearization”), that is, the operator is linearized by freezing the diffusion coefficient,  $g$ . This approach was chosen rather than NWT, because the linear operators obtained using NWT are nonsymmetric and therefore require a robust solver for nonsymmetric linear problems. We will consider such solvers in the future.

In these tests we also compute results using the NLMG of Hackbusch [14], with the parameter optimized manually. These must of course be at least as good as those of FAS, since the latter is a special case where the parameter is set to 1. With a few exceptions, the results are not significantly superior to FAS, even though the parameter is optimized.

The results are presented in Tables 2, 3, and 4 for the three cases of boundary conditions. We show average convergence factors for physical parameters  $\alpha = 0.5, 1.0, 1.5$ , and  $p = 1.5, 2.0, 2.5$ , with FP denoting fixed-point GL. As in the 1D case, for a given domain and boundary conditions, the problem generally becomes more difficult for smaller  $p$  and larger  $\alpha$ . The MNM algorithm yielded the fastest convergence in every case but two (the upper-right terms in Tables 2 and 4). In these cases (only), backtracking was required, and the results were still quite poor. (Recall that the analysis for backtracking was performed for Newton’s linearization, not FP.) For these two cases where MNM failed, we recomputed the results using the “adaptive” approach

TABLE 3  
Test results for Case 2.

$P \setminus \alpha$	.5	.75	1
1.5	MNM = 0.12 FAS = 0.18 NLMG = 0.18 FP = 0.42	MNM = 0.16 FAS = 0.22 NLMG = 0.22 FP = 0.53	MNM = 0.19 FAS = 0.30 NLMG = 0.30 FP = 0.64
2	MNM = 0.11 FAS = 0.18 NLMG = 0.15 FP = 0.34	MNM = 0.16 FAS = 0.26 NLMG = 0.26 FP = 0.51	MNM = 0.27 FAS = 0.35 NLMG = 0.35 FP = 0.66
2.5	MNM = 0.11 FAS = 0.18 NLMG = 0.16 FP = 0.33	MNM = 0.20 FAS = 0.30 NLMG = 0.29 FP = 0.55	MNM = 0.34 FAS = 0.40 NLMG = 0.40 FP = 0.74

TABLE 4

Test results for Case 3. The \* superscript denotes that backtracking was required. For this case, setting  $a = 0.2$ ,  $b = 0.4$  yielded for MNM an average convergence factor of 0.37 with no backtracking.

$P \setminus \alpha$	.5	.75	1
1.5	MNM = 0.15 FAS = 0.21 NLMG = 0.17 FP = 0.39	MNM = 0.24 FAS = 0.31 NLMG = 0.22 FP = 0.50	MNM = 0.94* FAS = 0.94* NLMG = 0.44 FP = 0.58
2	MNM = 0.10 FAS = 0.15 NLMG = 0.13 FP = 0.20	MNM = 0.11 FAS = 0.19 NLMG = 0.15 FP = 0.31	MNM = 0.13 FAS = 0.25 NLMG = 0.22 FP = 0.43
2.5	MNM = 0.10 FAS = 0.14 NLMG = 0.12 FP = 0.13	MNM = 0.10 FAS = 0.16 NLMG = 0.15 FP = 0.24	MNM = 0.11 FAS = 0.21 NLMG = 0.17 FP = 0.38

of section 3.2, with manually chosen parameters:  $a = 0.3$ ,  $b = 0.5$  yielded an average convergence factor of 0.44 for Case 1 with  $p = 1.5$ ,  $\alpha = 1$ , while  $a = 0.2$ ,  $b = 0.4$  yielded an average convergence factor of 0.37 for Case 3 with the same values for  $p$  and  $\alpha$ . This demonstrates the potential efficiency of adaptive MNM.

**5. Discussion.** MNM eliminates the compromise between global and local linearization, as we approximate the full nonlinear operator on the coarse grid, but we use a robust approximation for a large part of it. The new method exhibits better behavior than both its predecessors, consistent with the prediction of the scalar

analysis. The development of a two- (or higher-) dimensional solver employing Newton linearization is expected to yield better results, provided that a robust linear solver for nonsymmetric linear problems is used. The (partial) *adaptive* MNM approach was used in [6], where it proved to be effective, but it can no doubt be improved significantly by deriving appropriate criteria to replace the generic optimization. In the tests above, we demonstrate that optimizing both parameters of the adaptive MNM may be quite efficient, while ensuring that MNM will be at least as good as the better of the two competitors, as they are each a special case of adaptive MNM.

## REFERENCES

- [1] A. BRANDT, *Multi-level adaptive solutions to boundary-value problems*, Math. Comp., 31 (1977), pp. 333–390.
- [2] A. BRANDT, *Guide to multigrid development*, in Multigrid Methods, W. Hackbusch and U. Trottenberg, eds., Springer-Verlag, Berlin, 1982, pp. 220–312.
- [3] A. BRANDT, S. F. MCCORMICK, AND J. W. RUGE, *Algebraic multigrid (AMG) for sparse matrix equations*, in Sparsity and Its Applications, D. J. Evans, ed., Cambridge University Press, Cambridge, UK, 1984, pp. 257–284.
- [4] A. BRANDT AND I. YAVNEH, *On multigrid solution of high-Reynolds incompressible entering flows*, J. Comput. Phys., 101 (1992), pp. 151–164.
- [5] W. L. BRIGGS, V. E. HENSON, AND S. F. MCCORMICK, *A Multigrid Tutorial*, 2nd ed., SIAM, Philadelphia, 2000.
- [6] G. DARDYK AND I. YAVNEH, *A multigrid approach to two-dimensional phase unwrapping*, Numer. Linear Algebra Appl., 11 (2004), pp. 241–259.
- [7] P. M. DE ZEEUW, *Nonlinear multigrid applied to a one-dimensional stationary semiconductor model*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 512–530.
- [8] J. E. DENDY, JR., *Black box multigrid*, J. Comput. Phys., 48 (1982), pp. 366–386.
- [9] J. E. DENDY, JR., *Black box multigrid for nonsymmetric problems*, Appl. Math. Comput., 13 (1983), pp. 261–284.
- [10] J. E. DENNIS, JR., AND R. B. SCHNABEL, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice Hall Series in Computational Mathematics, Prentice-Hall, Englewood Cliffs, NJ, 1983.
- [11] D. DRIKAKIS, O. P. ILIEV, AND D. P. VASSILEVA, *A nonlinear multigrid method for the three-dimensional incompressible Navier-Stokes equations*, J. Comput. Phys., 146 (1998), pp. 301–321.
- [12] B. EPSTEIN, A. AVERBUCH, AND I. YAVNEH, *An accurate ENO driven multigrid method applied to 3D turbulent transonic flows*, J. Comput. Phys., 168 (2001), pp. 316–338.
- [13] P. GERLINGER, H. MÖBUS, AND D. BRÜGGEMAN, *An implicit multigrid method for turbulent combustion*, J. Comput. Phys., 167 (2001), pp. 247–276.
- [14] W. HACKBUSCH, *Multi-grid Methods and Applications*, Springer-Verlag, Berlin, 1985.
- [15] W. HACKBUSCH AND U. TROTTEBERG, EDs., *Multigrid Methods*, Springer-Verlag, Berlin, 1982.
- [16] M. J. HOLST, R. E. KOZACK, F. SAIED, AND S. SUBRAMANIAM, *Treatment of electrostatic effects in proteins – multigrid-based Newton iterative method for solution of the full nonlinear Poisson-Boltzmann equation*, Proteins: Structure Function and Genetics, 18 (1994), pp. 231–245.
- [17] M. J. HOLST AND F. SAIED, *Numerical solution of the nonlinear Poisson-Boltzmann equation – developing more robust and efficient methods*, J. Comput. Chem., 16 (1995), pp. 337–364.
- [18] J. KORSAAWE AND G. STARKE, *Multilevel projection methods for nonlinear least-squares finite element computations*, Electron. Trans. Numer. Anal., 21 (2000), pp. 1869–1885.
- [19] M. H. LALLEMAND AND B. KOREN, *Iterative defect correction and multigrid accelerated explicit time stepping schemes for the steady Euler equations*, SIAM J. Sci. Comput., 14 (1993), pp. 953–970.
- [20] J. MOLENAAR, *Adaptive multigrid applied to a bipolar-transistor problem*, Appl. Numer. Math., 17 (1995), pp. 61–83.
- [21] J. W. RUGE, Y. LI, S. MCCORMICK, A. BRANDT, AND J. R. BATES, *A nonlinear multigrid solver for a semi-Lagrangian potential vorticity-based shallow-water model on the sphere*, SIAM J. Sci. Comput., 21 (2000), pp. 2381–2395.
- [22] J. W. RUGE AND K. STÜBEN, *Algebraic multigrid (AMG)*, in Multigrid Methods, Frontiers Appl. Math. 3, S. F. McCormick, ed., SIAM, Philadelphia, 1987, pp. 73–130.

- [23] G. STARKE, *Gauss-Newton multilevel methods for least-squares finite element computations of variably saturated subsurface flow*, *Computing*, 64 (2000), pp. 323–338.
- [24] G. STARKE, *Least-squares mixed finite element solution of variably saturated subsurface flow problems*, *SIAM J. Sci. Comput.*, 21 (2000), pp. 1869–1885.
- [25] U. TROTTEBERG, C. OOSTERLEE, AND A. SCHÜLLER, *Multigrid*, Academic Press, London, San Diego, 2001.
- [26] M. TH. VAN GENUCHTEN, *A closed-form equation for predicting the hydraulic conductivity of unsaturated soils*, *Soil Sci. Soc. Am. J.*, 44 (1980), pp. 892–898.
- [27] C. R. VOGEL AND M. E. OMAN, *Iterative methods for total variation denoising*, *SIAM J. Sci. Comput.*, 17 (1996), pp. 227–238.
- [28] S. ZENG AND P. WESSELING, *Multigrid solution of the incompressible Navier–Stokes equations in general coordinates*, *SIAM J. Numer. Anal.*, 31 (1994), pp. 1764–1784.