USING INEXACT GRADIENTS IN A MULTILEVEL OPTIMIZATION ALGORITHM

ROBERT MICHAEL LEWIS* AND STEPHEN G. NASH†

Abstract. Many optimization algorithms require gradients of the model functions, but computing accurate gradients can be computationally expensive. We study the implications of using inexact gradients in the context of the multilevel optimization algorithm MG/Opt. MG/Opt recursively uses (typically cheaper) coarse models to obtain search directions for finer-level models. However, MG/Opt requires the gradient on the fine level to define the recursion. Our primary focus here is the impact of the gradient errors on the multilevel recursion. We analyze, partly through model problems, how MG/Opt is affected under various assumptions about the source of the error in the gradients, and demonstrate that in many cases the effect of the errors is benign. Computational experiments are included.

Key words. multilevel optimization, inexact gradient evaluation, optimization-based multigrid, nonlinear optimization

1. Introduction. We study the use of the multilevel optimization algorithm MG/Opt [9] applied to

$$\min f_h(x) \tag{1.1}$$

in the case where the gradients of f_h are computed inexactly. The problem (1.1) represents a hierarchy of models indexed by h, going from fine to coarse. MG/Opt recursively uses the coarse models to obtain search directions for finer-level models. This enables MG/Opt to compute solutions faster—sometimes dramatically so—than a traditional optimization algorithm [7].

Although gradients are used in many algorithms for (1.1) it can be expensive to compute accurate gradients if the optimization model is based on a complex numerical simulation (such as in [1]). Various authors have studied the implications of inexact gradients for a single optimization model [2, 3, 4, 8, 13]. They have demonstrated that convergence can be guaranteed if a trust-region optimization algorithm is used to solve (1.1), although the performance of the algorithm deteriorates as the errors in the gradient increase. If the savings from using an approximate gradient outweigh this deterioration, then the trade-off can be worthwhile.

Our goal is to study what happens when inexact gradients are used in the context of the multilevel optimization algorithm MG/Opt. MG/Opt is an optimization-based multigrid-type algorithm that uses a "surrogate" model—a coarse version of the problem (1.1)—to define a search direction for the fine-level model. The surrogate model is a shifted version of the coarse model that is defined using the value of the gradient on the fine level. We study here the impact of fine-level gradient errors on the surrogate model.

MG/Opt assumes the availability of a traditional optimization algorithm (referred to as Opt) that can be applied to an optimization problem on a specific level. Since the behavior of Opt is governed by the earlier research on this topic in the papers cited above, we focus on the issues that are unique to MG/Opt, i.e., the interactions of the models across levels.

*Department of Mathematics, College of William & Mary, Williamsburg, VA 23185. (rmlewi@wm.edu)

†Systems Engineering and Operations Research Department, George Mason University, Fairfax, VA 22030. (snash@gmu.edu).

In some cases, the underlying OPT algorithm may not be affected by gradient errors. For example, consider an engineering design problem with a small number of design variables but a potentially large number of state variables (perhaps corresponding to a PDE constraint). Then OPT could be a derivative-free algorithm, with a hierarchy of optimization models defined by refining the set of state variables.

To isolate the behavior of MG/OPT from the behavior of OPT we analyze only the effect of errors in the gradients used to form the surrogate models. (See Section 2.1 for a discussion of this issue.) We consider four possible sources of errors and analyze their impact on MG/OPT.

First we consider random errors, as was done in the computational tests in [4]. We demonstrate that the MG/OPT recursion can damp random errors.

There are many circumstances where the gradient errors would not be random. As a second possibility, suppose that a derivative-free method were used as the underlying optimization method OPT. Then a coarse finite-difference gradient could be obtained using values of the objective function at points in the pattern used by the optimization method. Here we show that the gradient errors will be dominated by higher-frequency components of the gradient. Since only a downdated gradient is used to construct the surrogate model, these errors would be filtered by MG/OPT.

The remaining two cases correspond to the situation where the errors in the gradient are confined to either the high or low frequency components. As a motivation for these two cases, suppose that (1.1) corresponds to a PDE-constrained optimization problem. If the PDE were time dependent, an approximate gradient could be obtained by solving the adjoint equation with a large time step. In this case the errors in the solution to the PDE would correspond to high-frequency components. Alternatively, if the PDE were linear and solved using an iterative scheme such as Gauss-Seidel, an approximate gradient could be obtained by solving the PDE with a small number of linear iterations. In this case the errors in the solution to the PDE would correspond to low-frequency components.

To analyze these two cases, we consider representative model problems, and show that the effects on the performance of MG/OPT are minimal, even when the errors in the gradient are large. We use model problems because they allow a detailed analysis of the properties of the models and the behavior of the algorithm. For general problems it is impossible to analyze the behavior of the reduced Hessian and other relevant quantities. Even though our model problems are simple (and even though they may not provide an exact correspondence to techniques applied to more realistic and challenging problems) we believe that they provide insight into the types of errors that might arise in the gradient of the optimization model, and on the effect of those errors on the MG/OPT recursion.

In a traditional multigrid algorithm applied to a linear PDE, the traditional optimization algorithm OPT corresponds to a smoother such as the Gauss-Seidel algorithm, and the gradient of the function f_h corresponds to the residual of the linear system. Thus an approximate gradient would correspond to an approximate residual. In the context of traditional multigrid the coefficient matrix is available, and the residual can be computed in a straightforward manner, so the issues that we discuss typically do not arise. However, in the context of optimization (particularly if derivative-free optimization methods are used) computation of accurate gradients can be a significant issue.

Here is an outline of the paper. In Section 2 we present the MG/OPT algorithm and summarize its convergence properties. Random errors are considered in Section

3, and truncation errors from coarse finite differencing in Section 4. The next two sections consider PDE-constrained model problems. In Section 5 the PDE is time-dependent, and an inexact gradient is obtained by taking a coarse time step. In Section 6 the PDE is solved by a linear iterative scheme, and an inexact gradient is obtained by performing a very small number of iterations of this scheme. Comments and conclusions are in Section 7.

2. The MG/OPT **Algorithm.** The description of the multilevel optimization algorithm MG/OPT used here is adapted from [10]. The algorithm explicitly refers to two adjacent levels in the hierarchy of models (1.1) denoted by h (fine) and H (coarse). It assumes the availability of appropriate update and downdate operators: I_h^H and I_h^H for the variables x_h. We also assume the availability of a convergent optimization algorithm OPT defined as a function of the form

$$x^+ \leftarrow \text{OPT}(f(\cdot), v, \bar{x}, k)$$

which applies k iterations of a convergent optimization algorithm to a shifted or "surrogate" problem

$$\min_x f(x) - v^T x$$

with initial guess $\bar{x}$ to obtain x^+. If the parameter k is omitted, the optimization algorithm continues to run until its termination criteria are satisfied. If OPT must cope with inexact gradients, then it should be a trust-region algorithm conforming to the theory in [3].

Here is the MG/OPT algorithm: Given an initial estimate of the solution x_h^0 on the finest level, set $v_h = 0$. Choose non-negative integers k_1 and k_2 satisfying $k_1 + k_2 > 0$. Then for $j = 0, 1, \ldots$, set

$$x_h^{j+1} \leftarrow \text{MG/OPT}(f_h(\cdot), v_h, x_h^j),$$

where the function MG/OPT is defined as follows:

- *Coarse-level solve:* If on the coarsest level,

$$x_h^{j+1} \leftarrow \text{OPT}(f_h(\cdot), v_h, x_h^j).$$

  Otherwise,
- *Pre-smoothing:*

$$\bar{x}_h \leftarrow \text{OPT}(f_h(\cdot), v_h, x_h^j, k_1).$$

- *Recursion:*
  - Compute

$$\begin{aligned} \bar{x}_H &= I_h^H \bar{x}_h, \\ \bar{v}_H &= I_h^H v_h + \nabla f_H(\bar{x}_H) - I_h^H \nabla f_h(\bar{x}_h). \end{aligned}$$

  - Apply MG/OPT recursively to the surrogate model:

$$x_H^+ \leftarrow \text{MG/OPT}(f_H(\cdot), \bar{v}_H, \bar{x}_H).$$

  - Compute the search directions $e_H = x_H^+ - \bar{x}_H$ and $e_h = I_H^h e_H$.
  - Use a line search to determine $x_h^+ = \bar{x}_h + \alpha e_h$ satisfying $f_h(x_h^+) \le f_h(\bar{x}_h)$.

- *Post-smoothing:*

$$x_h^{j+1} \leftarrow \text{OPT}(f_h(\cdot), v_h, x_h^+, k_2).$$

The convergence properties of MG/OPT are analyzed in [10]. The assumptions made on the objective function are the same as the assumptions typically needed to prove convergence of the underlying optimization method OPT. With additional assumptions, it is possible to guarantee that the search direction e_h from the recursion step of MG/OPT is a descent direction for f_h at $\bar{x}_h$. These results assume that the gradients are accurate.

2.1. Effects of Gradient Errors on MG/OPT**.** As mentioned in the Introduction, since the behavior of OPT is assumed to be governed by the earlier research on this topic in the papers cited above, we focus on the interactions of the models across levels. In addition, we only consider errors in the gradient on the fine level. Here we justify this approach.

There are several justifications for focusing on the fine-level gradient. First, since the coarse model will typically be a cheaper or simpler model, it may be feasible to compute accurate gradients on the coarse level. On the coarse level, these might be analytical gradients, or just more accurate gradients made feasible by the lower computational costs on this level. This would be true if the errors in the gradient corresponded to coarse time stepping (see Section 5) or a truncated linear iteration (see Section 6). Thus in these cases the errors in the gradient on the fine level may dominate any errors on the coarse level.

Second, any errors in the coarse-level gradient could be considered as part of the analysis of the behavior of the optimization algorithm OPT applied on the coarse level. The surrogate model has the form

$$f_H(x) - v^T x, \tag{2.1}$$

where, if we focus on the case where there are only two levels in the model hierarchy,

$$v = \nabla f_H(\bar{x}_H) - I_h^H \nabla f_h(\bar{x}_h).$$

An error in the computation of $\nabla f_H(\bar{x}_H)$ could be interpreted as contributing to an error in the evaluation of $\nabla f_H(x)$. This perspective would be especially relevant if there were random errors in the gradient (see Section 3).

If a pattern search method were used on each level, and, for the purposes of the correction in (2.1), the gradient were estimated using coarse finite differencing (see Section 4) then the estimate of $\nabla f_H(\bar{x}_H)$ would have to be obtained from the initial pattern used on the coarse level. If the same pattern were used on the coarse and fine levels then the errors in the coarse and fine gradients would be of comparable magnitude. If we wish to ignore the error in the coarse-level gradient then a finer pattern (or a higher-order approximation formula) would have to be used at the initial point $\bar{x}_H$. Since the coarse model is typically cheaper to evaluate, and since the extra points would only be needed at the initial point, this may be reasonable. Also, since at later iterations of MG/OPT the initial guess $\bar{x}_H$ is often close to a solution, it is appropriate to start with a finer pattern.

There are also general reasons for focusing on the errors in the gradient on the fine level, and ignoring gradient errors on the coarse level.

In the case of inexact gradients, convergence theory for the underlying optimization method OPT can be found in [3]. It is straightforward to prove convergence for

MG/Opt with inexact gradients. If the underlying optimization algorithm Opt is guaranteed to converge in this case, then so is MG/Opt since in the worst case the results of the recursion step are discarded. This argument is formalized in [10], and is not changed in the case of inexact gradients.

Under appropriate assumptions, it is possible to prove that the MG/Opt search direction e_h is a descent direction. The theorem below extends a result in [10] to the case where the gradients are computed inexactly.

Theorem 2.1. *Assume that on all levels*

(a) $\nabla f_h(x_h)$ *is defined for all values of* x_h,

(b) the level set $S_h = \left\{x_h : f(x_h) \le f(x_h^0)\right\}$ *is compact, where* x_h^0 *is the initial guess of the solution of (1.1),*

(c) $\nabla^2 f_h(x_h)$ *is continuous for all choices of* $x_h \in S_h$, *and*

(d) $(I_H^h)^T = C_I I_h^H$ *for some constant* $C_I > 0$.

In addition, assume that $\nabla f_h(\bar{x}_h) \neq 0$. *Let* ξ *be the error in the computed value of* $I_h^H \nabla f_h(\bar{x}_h)$, *i.e.,*

$$I_h^H \nabla f_h(\bar{x}_h) = I_h^H \nabla \hat{f}_h(\bar{x}_h) + \xi$$

where $\hat{f}$ *refers to the computed value. Then the search direction* e_h *from the recursion step of MG/Opt will be a descent direction for* f_h *at* $\bar{x}_h$, *i.e.,*

$$e_h^T \nabla f_h(\bar{x}_h) < 0,$$

if

(e) $f_s(x_H^+) < f_s(\bar{x}_H)$, *and*

(f) $e_H^T \nabla^2 f_H(\bar{x}_H + \eta e_H) e_H > \xi^T e_H$ *for* $0 \le \eta \le 1$.

Proof. It is sufficient to consider only two levels. Let $f_s(x) = f_H(x) - v^T x$ be the surrogate model on the coarse level with

$$v = \nabla f_H(\bar{x}_H) - I_h^H \nabla \hat{f}_h(\bar{x}_h).$$

Since $f_s(x_H^+) < f_s(\bar{x}_H)$ and $x_H^+ = \bar{x}_H + e_H$, Taylor's theorem gives

$$f_s(\bar{x}_H) + e_H^T \nabla f_s(\bar{x}_H) + \frac{1}{2} e_H^T \nabla^2 f_s(\bar{x}_H + \eta e_H) e_H < f_s(\bar{x}_H)$$

for some $0 \le \eta \le 1$. Using the formulas for the surrogate model this simplifies to

$$e_H^T I_h^H \nabla \hat{f}_h(\bar{x}_h) + \frac{1}{2} e_H^T \nabla^2 f_H(\bar{x}_H + \eta e_H) e_H < 0.$$

Using the definition of C_I and the formula for the computed gradient we can rewrite this as

$$e_h^T \nabla f_h(\bar{x}_h) < C_I \left[-\frac{1}{2} e_H^T \nabla^2 f_H(\bar{x}_H + \eta e_H) e_H + \xi^T e_H \right].$$

The right-hand side will be negative if

$$e_H^T \nabla^2 f_H(\bar{x}_H + \eta e_H) e_H > \xi^T e_H.$$

This completes the proof. □

In the theorem the assumptions (a), (b), and (c) are used to guarantee convergence. Assumption (e) just says that Opt reduces the surrogate function on the

coarse level. Assumption (d) is the same as in [10] and (f) is a slight variant. There is an extended discussion in [10] of the reasons for the various assumptions, and of alternative results that do not assume that

$$e_H^T \nabla^2 f_H(\bar{x}_H + \eta e_H) e_H > 0.$$

This theorem only makes reference to errors in the gradient on the fine level. Gradient errors on the coarse level are not relevant to proving descent. This is further justification for our focus on errors in the gradient on the fine level.

3. Random Errors in the Gradient. We now examine (1.1) in the case where the errors in the gradient are independent and identically distributed (i.i.d.) random variables with bounded variance. We assume that the variables x_h are discrete approximations to some continuous parameter $X(\cdot)$. For convenience we assume that the variables x_h are defined on an interval or box, with uniform discretization. For example, x_h might correspond to the values of the solution at uniformly spaced points on the interval $[0, 1]$ or in the box $[-\pi, \pi] \times [-\pi, \pi]$.

We will examine the variance of the errors in the gradients. Recall that the variance of a linear combination of i.i.d. random variables $\{y_i\}$ satisfies

$$\mathrm{Var}\left(\sum_i a_i y_i\right) = \sum_i a_i^2 \mathrm{Var}(y_i).$$

We show that the variance of the errors in the gradients is reduced in the recursion step.

For simplicity, consider a two-level model where the gradients are computed accurately on the (cheaper) coarse level. Then, on the coarse level, the gradient errors from the fine level appear in the shift vector

$$\bar{v}_H = \nabla f_H(\bar{x}_H) - I_h^H \nabla f_h(\bar{x}_h).$$

(The formula for $\bar{v}_H$ in Section 2 includes $I_h^H v_h$, but that term is zero in a two-level model.)

Suppose that the downdate operator I_h^H is based on full weighting. Then, if $X(\cdot)$ is defined on an interval, I_h^H is defined by the stencil

$$\frac{1}{4}\begin{pmatrix} 1 & 2 & 1 \end{pmatrix}. \tag{3.1}$$

If all components of the error have variance σ^2 then a typical component of $I_h^H \nabla f_h$ will have variance

$$\left(\frac{1}{4}\right)^2 \left[1^2 + 2^2 + 1^2\right] \sigma^2 = \frac{3}{8}\sigma^2.$$

Thus the variance of the error is reduced by about 60%. Similar results would be obtained for other downdate operators.

If $X(\cdot)$ is defined on a rectangle in two dimensions, then the stencil for the downdate operator I_h^H based on full weighting is

$$\frac{1}{16}\begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}.$$

If all components of the error have variance σ^2 then a typical component of $I_h^H \nabla f_h$ will have variance

$$\left(\frac{1}{16}\right)^2 \left[4 \times 1^2 + 4 \times 2^2 + 4^2\right] \sigma^2 = \frac{9}{64}\sigma^2.$$

In this case the variance is reduced by about 86%.

Similarly, if $X(\cdot)$ is defined on a box in three dimensions, then a typical component of $I_h^H \nabla f_h$ will have variance

$$\frac{216}{4096}\sigma^2,$$

and the variance is reduced by about 95%.

These results suggest that the multigrid recursion will ameliorate the effect of random errors in the gradient on the fine level. We now report on numerical experiments that support this conclusion.

We consider two test problems. The first is a quadratic based on the one-dimensional discrete Laplacian described in [11]. Two levels of discretization are used: the fine level is based on 511 points, and the coarse level on 255 points. The second test problem is the minimal surface problem described in [9]. Again, two levels are used: the fine level is based on a discretization of 31×31 points, and the coarse level on 15×15 points. To isolate the effects of gradient errors on the recursion step of MG/OPT, errors were introduced into the fine-level gradient evaluation of $\nabla f_h(\bar{x}_h)$, but all other gradients were computed accurately. (The computational results in [4] show how errors in the gradient affect the performance of the underlying optimization algorithm OPT.) In cases where the search direction e_h from the MG/OPT recursion was not a descent direction, the results of the recursion step were discarded.

The gradients were perturbed with normally distributed random errors scaled to give a specified relative error. Computations were done with the relative error varying from 5% to 50%. For each setting, 20 separate runs were made with different random errors. The results are in Figures 3.1 and 3.2. In each figure the horizontal axis corresponds to the relative error in the perturbation, and the vertical axis to the number of equivalent fine-grid gradient computations necessary to obtain the solution to the specified tolerance. The three curves correspond to the minimum, average, and maximum number of gradient evaluations over the 20 runs.

As the figures demonstrate, with inexact gradients the algorithm typically takes longer to find the solution. But even with large errors in the gradient (up to 50% relative error) the deterioration in performance is not severe. If using inexact gradients leads to significant savings, then the trade-off may be worthwhile. Notice that there is a greater deterioration in performance for the one-dimensional Laplacian problem than for the two-dimensional minimal surface problem. This may be because random errors are damped more strongly in the two-dimensional case.

For the Laplacian test problem the search direction e_h was always a descent direction, regardless of the size of the gradient errors. For the minimal surface problem the search direction e_h was always a descent direction for relative errors of 10% or less. For larger relative errors, e_h failed to be a descent direction in 1–2 MG/OPT iterations per optimization run. Thus, even with large relative errors in the fine-level gradient, at the majority of MG/OPT iterations a descent direction was obtained.

4. Truncation Errors from Coarse Finite Differencing. Consider now the case where the gradient is estimated using coarse finite differencing. This might be

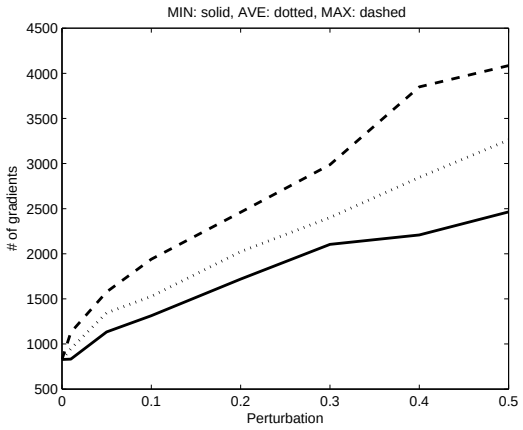

Fig. 3.1: Effect of gradient errors for the Laplacian test problem.

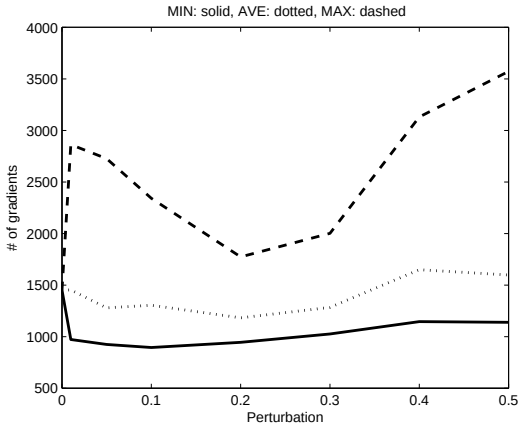

Fig. 3.2: Effect of gradient errors for the minimal surface test problem.

appropriate if a pattern search method were used as the underlying optimization method OPT. Then the values of the objective function computed by the pattern search method could be used to estimate the gradient in order to make the correction in (2.1) used in MG/OPT. If compass search were used as the pattern search method, then this would indeed be coarse finite differencing. (For a detailed discussion of pattern search methods, see [6].)

For simplicity we assume that compass search is used with a steplength parameter Δ that is the same for all search directions. (Similar remarks would apply if the gradient were estimated using a simplex gradient [5].) Then as part of the pattern search around a point $\bar{x}_h$, OPT will compute $f_h(\bar{x}_h)$ as well as $\{f_h(\bar{x}_h \pm \Delta e_i)\}_{i=1}^N$, where e_i is the i-th coordinate direction. The gradient can then be estimated using

$$(\nabla f_h(\bar{x}_h))_i \approx \frac{f_h(\bar{x} + \Delta e_i) - f_h(\bar{x} - \Delta e_i)}{2\Delta}.$$

Laplacian		*Surface*	
Δ	Change	Δ	Change
10^2	1.11	10^2	0.57
10^1	0.95	10^1	0.74
10^0	0.96	10^0	0.71
10^{-1}	1.04	10^{-1}	0.67
10^{-2}	0.92	10^{-2}	1.70
10^{-3}	1.14	10^{-3}	1.59

Table 4.1: Using coarse finite-differencing with parameter Δ. The Change column is the ratio of computational effort compared with the use of analytic gradients.

In general, central differencing damps high frequency components (e.g., numerical dissipation in finite difference schemes for PDE [12]). The range of frequencies that are damped increases with the magnitude of Δ, so if Δ is large then only low-frequency components of ∇f_h will be computed accurately. For a related discussion of this low-pass filtering effect in the context of pattern search, see [6].

In MG/Opt, the shift vector $\bar{v}_H$ is formed using the *downdated* gradient

$$I_h^H \nabla f_h(\bar{x}_h).$$

Since the downdated gradient represents the low-frequency components of the gradient, we would expect that coarse finite-differencing would have little effect on the performance of MG/Opt, as long as Δ were not excessively large. (For a particular problem it would be possible to be more explicit about the meaning of "excessively", based on the specific choices for the optimization models at the various levels.)

To confirm this analysis we applied coarse central differencing within MG/Opt to estimate the fine-level gradient used to construct the surrogate model on the coarse level. This idea was tested on the two problems described in the previous section. The results are in Table 4.1. The table lists the values of Δ that were used. The column labeled "Change" is the ratio of the computational effort with coarse finite differencing versus the effort with analytic gradients. Effort is measured by the number of equivalent fine-level gradient evaluations required to achieve convergence to a specified tolerance. The performance of MG/Opt was not significantly affected by the use of these approximations. In fact, in a number of cases the effort was less with approximate gradients than with analytic gradients.

In the computational experiments we used central differencing and not forward differencing. Forward differencing with coarse values of Δ can produced biased gradient estimates. Consider the Laplacian problem, a quadratic of the form

$$f_h(x) = \frac{1}{2} x^T A x - x^T b,$$

where A is the negative of the one-dimensional discrete Laplacian with the stencil

$$\frac{1}{h^2} \begin{pmatrix} -1 & 2 & -1 \end{pmatrix}.$$

Suppose that x is near the solution, so $Ax \approx b$. Let e_i be the i-th column of the

identity matrix. Then forward differencing produces the gradient estimate

$$\begin{aligned}\frac{\partial f_h(x)}{\partial x_i} &\approx \frac{f_h(x+\Delta e_i) - f_h(x)}{\Delta} \\ &= \frac{1}{2}\Delta e_i^T A e_i + e_i^T(Ax-b) \\ &\approx \frac{1}{2}\Delta e_i^T A e_i \\ &= \frac{\Delta}{h^2}.\end{aligned}$$

For a problem with $n = 511$ variables, $h = 1/512$ and $h^2 = 1/262,144$, so

$$\frac{\partial f_h(x)}{\partial x_i} \approx 262,144\Delta.$$

Thus forward differencing in this case produces a poor estimate of the gradient unless Δ is tiny. For this reason we used central differencing.

5. Errors from Coarse Time Stepping. We now consider the case of a PDE-constrained problem, where the PDE is time dependent. An approximate gradient can be computed by solving the adjoint equation with a coarse time step. We analyze the effect of errors of this type on the MG/Opt recursion.

We analyze this case via a model problem. Given $a(x)$, let $u[a]$ be the solution of the initial-value problem

$$\begin{aligned} u_t(x,t) + cu_x(x,t) &= 0 \\ u(x,0) &= a(x), \end{aligned} \tag{5.1}$$

and consider the optimization problem

$$\begin{aligned}\underset{a}{\text{minimize}}\, F(a) &= f(a, u[a]) \\ &= \frac{1}{2}\int\int_0^T \left[\alpha(u(x,t)-\phi(x,t))^2 + \beta(u_x(x,t)-\phi_x(x,t))^2\right] dx\, dt,\end{aligned} \tag{5.2}$$

where α, β are non-negative weights, ϕ is a prescribed target, and c is a constant. The approximate gradient is obtained by solving the PDE with a coarse time step. We analyze the effect of the corresponding errors on the behavior of MG/Opt.

This model problem is the advection problem considered in [7]. Suppose that the errors in the gradient result from computing the solution to the PDE using a coarse time step. These errors will not be random. Also, since high-frequency behavior will not be properly resolved, the magnitudes of the errors may be large, even if the numerical scheme accurately resolves the low-frequency components of the solution. Hence the analysis in [3] may not apply.

The discussion here makes use of the analysis in [7]. The reduced gradient of the model problem has the form

$$\nabla F(a) = \frac{du}{da}^* f_u$$

where f_u is the gradient of the objective with respect to the state variables u, and du/da is obtained from the PDE.

Following the notation in [7], the adjoint of the map that takes the initial condition to the solution (as a function of x and t) is given by

$$\frac{du}{da}^* : w_m^n \to \sum_{n=1}^{N} \overline{g(\omega_m \Delta x)}^{\,n} \, w_m^n \, \Delta t.$$

Here $g(\cdot)$ is the amplification factor for the numerical scheme applied to the PDE. In [7] we used a backward-space, forward-time scheme. Here, because we are going to coarsen the time step and we are concerned about stability, we will use a backward-space, backward-time scheme. The amplification factor for that scheme is derived below.

Let $r(\cdot)$ be the residual in the objective function, i.e.,

$$r(x,t) = u(x,t) - \phi(x,t).$$

We define $\bar{\omega}_m = \omega_m \Delta x$. Then we get the following formula for the symbol of $\nabla F(a)$:

$$\left(\sum_{n=1}^{N} \overline{g(\bar{\omega}_m)}^{\,n} \, \Delta t \right) \sigma(\bar{\omega}_m) \hat{r}(\bar{\omega}_m), \tag{5.3}$$

where $\sigma(\omega) = \alpha + \beta |\theta(\omega)|^2$,

$$\theta(\omega) = i e^{i \Delta x \frac{\omega}{2}} \frac{2 \sin(\Delta x \frac{\omega}{2})}{\Delta x},$$

and α and β are the constants in our objective function.

The formula (5.3) includes the term $\hat{r}(\bar{\omega}_m)$. This residual term can take on any value, and depends on the particular values of the variables in the optimization model. The main focus here is on how the Fourier coefficients are affected by the algorithm. For this reason, we drop the term $\hat{r}(\bar{\omega}_m)$ in the following analysis.

5.1. Fourier analysis of the backward-time, backward-space scheme for the advection equation. We discretize (5.1) letting Δt be the time step and Δx be the spatial mesh spacing. Then the backward-time, backward-space scheme is

$$u_m^{n+1} = u_m^n - c \frac{\Delta t}{\Delta x} (u_m^{n+1} - u_{m-1}^{n+1}).$$

Define $\lambda = \Delta t / \Delta x$.

For the initial condition $u(x,0) = e^{i\omega x}$, posit a solution of the form

$$u_m^n = g^n e^{i \omega m \Delta x}.$$

Solving for g we obtain

$$\begin{aligned}
u_m^{n+1} &= u_m^n - c\lambda (u_m^{n+1} - u_{m-1}^{n+1}) \\
g^{n+1} e^{i\omega m \Delta x} &= g^n e^{i\omega m \Delta x} - c\lambda (g^{n+1} e^{i\omega m \Delta x} - g^{n+1} e^{i\omega (m-1) \Delta x}) \\
g e^{i\omega m \Delta x} &= e^{i\omega m \Delta x} - gc\lambda (e^{i\omega m \Delta x} - e^{i\omega (m-1) \Delta x}) \\
g e^{i\omega m \Delta x} &= e^{i\omega m \Delta x} - gc\lambda (1 - e^{-i\omega \Delta x}) e^{i\omega m \Delta x} \\
g &= 1 - gc\lambda (e^{i\omega \Delta x / 2} - e^{-i\omega \Delta x / 2}) e^{-i\omega \Delta x / 2}.
\end{aligned}$$

Let $\theta = \omega\Delta x/2$; then

$$e^{i\theta} - e^{-i\theta} = 2i\sin\theta.$$

Thus,

$$g = 1 - g\,2i\,c\lambda\sin\theta\,e^{-i\theta},$$

so

$$g = \frac{1}{1 + 2i\,c\lambda\sin\theta\,e^{-i\theta}}. \tag{5.4}$$

Note that

$$|\,g\,|^2 = \frac{1}{|\,1 + 2i\,c\lambda\sin\theta\,e^{-i\theta}\,|^2} = \frac{1}{1 + 4c^2\lambda^2\sin^2\theta}.$$

Thus $|\,g\,| \le 1$, so the scheme is unconditionally stable. Observe also that as $\Delta t/\Delta x$ increases, the magnitude $|\,g\,|$ decreases, indicating a smoothing effect since the amplitude of the sinusoid is damped.

Now suppose the initial condition is defined by the values $a = a_m$ at the points $m\Delta x$, $m \in \mathbb{Z}$. The Fourier transform of this grid function is

$$\hat{a}(\omega) = \frac{1}{\sqrt{2\pi}} \sum_{m=-\infty}^{\infty} e^{-im\omega\Delta x} a_m \Delta x.$$

At time step n the solution of the difference scheme is given by u_m^n, where the Fourier transform of the grid function u_m^n is given by

$$\hat{u}(\omega\Delta x, n\Delta t) = g(\omega\Delta x; \lambda)^n\,\hat{a}(\omega).$$

We have written $g(\omega\Delta x; \lambda)$ to indicate the dependence of the amplification factor on ω, Δx, and λ.

5.2. Effect on MG/Opt**.** In the recursion step of MG/Opt we use a shift that is based on the downdated gradient of the optimization model on the fine level. We are interested here in what happens if that gradient is only computed approximately. On the coarse level, only low-frequency components can be represented. So we are interested in the low-frequency components of the difference between the downdated gradients, one accurate and one approximate.

Following [14], page 113, the effect of the downdate (3.1) is to multiply the symbol in (5.4) by

$$\frac{1}{2}(1 + \cos\bar{\omega}),$$

where $\bar{\omega} = \omega\Delta x$.

Combining all of these ideas, we obtain that the Fourier symbol for the downdated gradient with the accurate difference scheme is

$$\frac{1}{2}(1 + \cos\bar{\omega}_m)\left(\sum_{n=1}^{N} \overline{g(\bar{\omega}_m)}^{\,n}\,\Delta t\right)\sigma(\bar{\omega}_m) \tag{5.5}$$

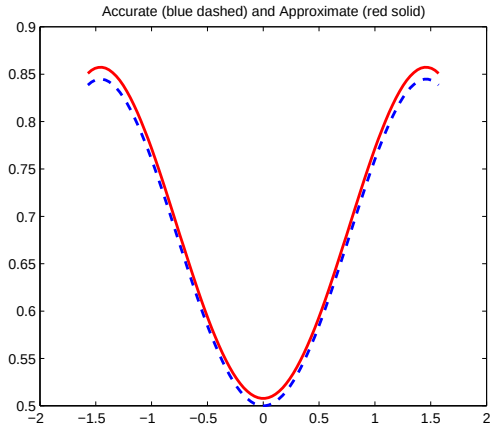

Fig. 5.1: Fourier Symbols for Accurate and Approximate Gradients

and the Fourier symbol for the downdated gradient with the approximate difference scheme based on a larger time step is

$$\frac{1}{2}(1+\cos\bar{\omega}_m)\left(\sum_{n=1}^{\tilde{N}}\overline{\tilde{g}(\bar{\omega}_m)}^{\,n}\,\Delta t\right)\sigma(\bar{\omega}_m) \tag{5.6}$$

It should be noted that the amplification factor $g(\cdot)$ depends on Δx, Δt, and the coefficient c in the PDE.

Typically we will double the time step, using $2\Delta t$ instead of Δt as the time step. Then $\tilde{N} = N/2$. In that case, we are interested in the values of (5.5) and (5.6) for

$$-\frac{\pi}{2} \le \bar{\omega}_m \le \frac{\pi}{2}.$$

We ran a simple Matlab experiment to compute the values for the two formulas. The results are in Figure 5.1. As can be seen, there is little difference between the two; this is not surprising in view of the smoothing effect of the term σ in (5.3). This suggests that using the approximate gradient will have little effect on the performance of MG/Opt.

When MG/Opt is applied to this problem, the results show virtually no change when coarse time stepping is used to compute the gradient. This provides additional evidence that the gradient errors that result from coarse time stepping do not have a major impact on the multilevel recursion.

6. Errors from Truncated Linear Iteration. We now consider another PDE-constrained problem where the PDE is solved by a linear iterative method. An approximate gradient is obtained by truncating the linear iteration after a small number of iterations. Again we illustrate this case via a model problem.

The model problem is a Dirichlet-to-Neumann map for the Laplacian on a rectangle in two dimensions. The Laplacian is defined on the region

$$\Omega = \{(x_1, x_2)\,|\,-\pi \le x_1 \le \pi,\, 0 \le x_2 \le 2\pi\}\,.$$

Let $\Gamma = \{(x_1, 0) \,|\, 0 \le x_1 \le \pi\}$ be the bottom portion of the boundary of Ω, and let u be the solution of

$$\begin{aligned} \Delta u(x_1, x_2) &= 0 \qquad \text{in } \Omega \\ u(x_1, x_2) &= 0 \qquad \text{on } \partial\Omega \setminus \Gamma \\ u(x_1, 0) &= a(x_1). \end{aligned}$$

The optimization model is

$$\min_a F(a) = \frac{1}{2} \int_0^\pi \left(\frac{\partial u}{\partial x_2}(x_1, 0) - \phi(x_1) \right)^2 dx_1,$$

where ϕ is a prescribed target function. We discretize u uniformly in the directions x_1 and x_2 with mesh spacing h, and use the same discretization in the x_1 direction for a. The discretization is indexed from 0 to $N-1$ in each direction, where N is odd. The standard five-point stencil is used to discretize the Laplacian.

This model problem was discussed in [7]. Here we extend that analysis to consider the effect of inexact gradient computations. The inexact gradient will be computed by applying the Gauss-Seidel method to the Laplacian, and terminating after a small number of iterations. Formally, the gradient g is computed by solving

$$\Delta w(x_1, x_2) = 0 \qquad \text{in } \Omega \tag{6.1}$$

$$w(x_1, x_2) = 0 \qquad \text{on } \partial\Omega \setminus \Gamma \tag{6.2}$$

$$w(x_1, 0) = r(x_1) = \frac{\partial u}{\partial x_2}(x_1, 0) - \phi(x_1), \tag{6.3}$$

and setting $g(x_1) = \partial_{x_2} w(x_1, 0)$.

We apply Fourier analysis to the numerical calculation of the gradient, and examine the frequency dependence of the amplification factors. We focus on this computation since the other steps in forming the gradient are computed exactly.

Consider the numerical computation of the g_m, the approximation of the values of g at the points $x_1 = mh$. We must first solve the boundary value problem (6.1)–(6.3) to obtain discretized values $w_{m,n}$ for w. Suppose we solve the discretized version of (6.1)–(6.3) using Gauss-Seidel. If $E^{(0)}_{m,n}$ is initial the error in the estimate of $w_{m,n}$ computed using Gauss-Seidel, then $E^{(0)}_{m,n}$ will have homogeneous boundary conditions and may be written in the form

$$E^{(0)}_{m,n} = \sum_{k_1,k_2=-M}^{M} \nu_{k_1,k_2} e^{i2\pi \frac{k_1}{N} m} e^{i2\pi \frac{k_2}{N} n},$$

where $M = \lfloor N/2 \rfloor$.

For a single iteration of Gauss-Seidel with lexicographic ordering of the variables, the amplification factor is [14]

$$\gamma(\theta_1, \theta_2) = \frac{e^{i\theta_1} + e^{i\theta_2}}{4 - e^{-i\theta_1} - e^{-i\theta_2}}.$$

That is, if we apply the smoothing operator S_h that results from the Gauss-Seidel iteration to functions of the form $\phi(\theta_1, \theta_2, x_1, x_2) = e^{i\theta_1 x_1/h} e^{i\theta_2 x_2/h}$, then we obtain

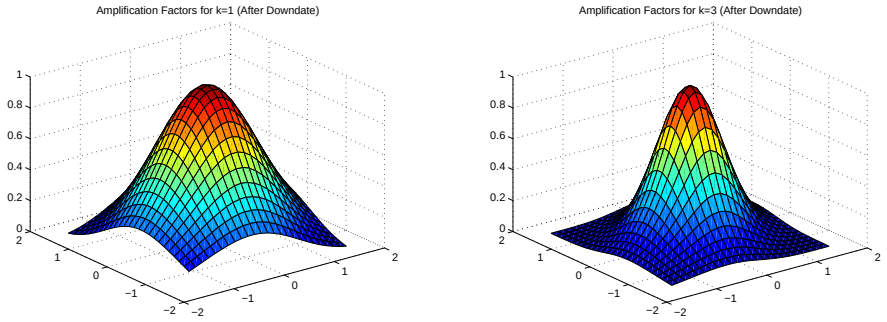

Fig. 6.1: Amplification Factors (left: $k = 1$, right: $k = 3$)

$S_h\phi(\theta_1, \theta_2, x_1, x_2) = \gamma(\theta_1, \theta_2)\phi(\theta_1, \theta_2, x_1, x_2)$. If we apply k iterations of Gauss-Seidel the resulting amplification factor is

$$\psi^{(k)}(\theta_1, \theta_2) = \gamma(\theta_1, \theta_2)^k$$

for $-\pi \le \theta_1, \theta_2 \le \pi$. Plots of $\psi^{(k)}(\theta_1, \theta_2)$ for $k = 1$ and $k = 3$ are shown in Figure 6.1. For $(\theta_1, \theta_2) \approx (0, 0)$ we have $\psi(\theta_1, \theta_2) \approx 1$, so the corresponding low-frequency errors in the error E are not damped. This is potentially discouraging, but it does not tell the full story.

Let $\theta(\ell) = 2\pi\ell/N$. After k iterations of Gauss-Seidel the error $E^k_{m,n}$ in $w_{m,n}$ is

$$E^{(k)}_{m,n} = \sum_{k_1,k_2=-M}^{M} \psi^{(k)}(\theta(k_1), \theta(k_2))\ \nu_{k_1,k_2}\ e^{i\theta(k_1)m} e^{i\theta(k_2)n}.$$

The error $e^{(k)}_m$ in the gradient is given by the discrete normal derivative of $E^{(k)}_{m,n}$ on Γ:

$$e^{(k)}_m = \sum_{k_1=-M}^{M} \left(\sum_{k_2=-M}^{M} \psi^{(k)}(\theta(k_1), \theta(k_2))\ \nu_{k_1,k_2}\ \frac{e^{i\theta(k_2)} - 1}{h} \right) e^{i\theta(k_1)m}. \tag{6.4}$$

If we define

$$\sigma_{k_1} = \sum_{k_2=-M}^{M} \psi^{(k)}(\theta(k_1), \theta(k_2))\ \nu_{k_1,k_2}\ \frac{e^{i\theta(k_2)} - 1}{h},$$

then

$$e^{(k)}_m = \sum_{k_1=-M}^{M} \sigma_{k_1} e^{i\theta(k_1)m}.$$

The expression (6.4) is a mix of frequency terms in the x_1 direction and the x_2 derivative $x_2 = 0$. In contrast, the amplification factors in Figure 6.1 correspond to

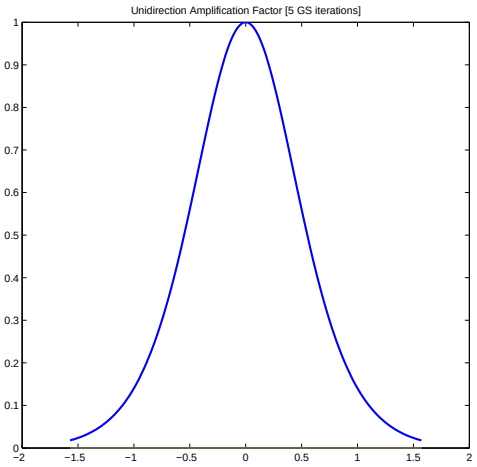

Fig. 6.2: Reduced Amplification Factor $\Psi(k_1)$

frequencies in both directions. Since

$$
\begin{aligned}
|\,\sigma_{k_1}\,| &\leq \sum_{k_2=-M}^{M} \left|\,\psi^{(k)}(\theta(k_1),\theta(k_2))\,\right| \;|\,\nu_{k_1,k_2}\,| \;\left|\,\frac{e^{i\theta(k_2)}-1}{h}\,\right| \\
&\leq \max_{-M\leq k_2<M} \left|\,\psi^{(k)}(\theta(k_1),\theta(k_2))\,\right| \sum_{k_2=-M}^{M} |\,\nu_{k_1,k_2}\,| \;\left|\,\frac{e^{i\theta(k_2)}-1}{h}\,\right|,
\end{aligned}
$$

we can obtain an bound on how errors in the gradient are damped as a function of k_1 and the number of Gauss-Seidel iterations k by computing

$$
\Psi^{(k)}(k_1) = \max_{-M\leq k_2\leq M} \left|\,\psi^{(k)}(\theta(k_1),\theta(k_2))\,\right|.
$$

A plot of this function corresponding to the use of $k = 5$ Gauss-Seidel iterations is given in Figure 6.2.

If MG/Opt is applied to a hierarchy of models then at each level we would expect MG/Opt to resolve components of the solution corresponding to high frequencies on that level. As Figure 6.2 shows, if $|\theta(k_1)| \approx \pi/2$ then $\Psi(k_1)$ will be small. These are the high frequencies on the next coarser grid. (The effect is even more pronounced if additional Gauss-Seidel iterations are performed.) Thus we can expect that the high-frequency components of the gradient on the next coarser level will be computed to relatively high accuracy, even if the low-frequency components of the gradient have large errors.

To confirm this analysis, we used MG/Opt to solve this model problem with a five-level hierarchy where the interval $[0, \pi]$ was discretized with $h = 1/N$ for $N = 257, 129, 65, 33, 17$. We compared the results for an exact gradient with the results for an inexact gradient with $k = 1$ iteration of Gauss-Seidel applied to "solve" the constraint for u given a. The results are in Figures 6.3 and 6.4. We report on three algorithms: OPT (the traditional optimization algorithm), MG (MG/Opt),

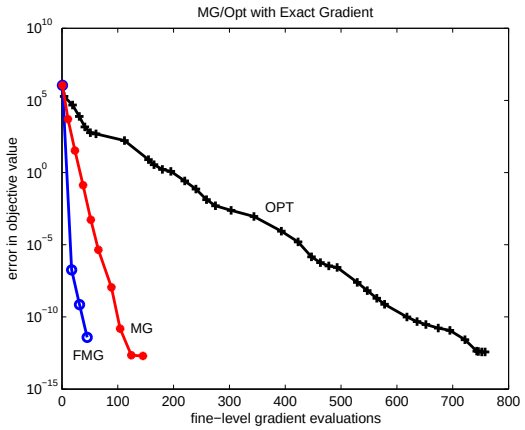

Fig. 6.3: Dirichlet-to-Neumann map: exact gradients

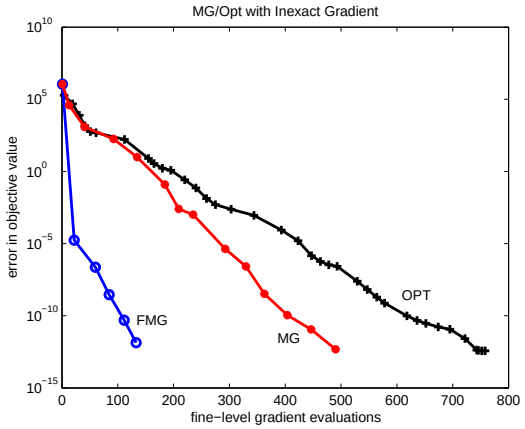

Fig. 6.4: Dirichlet-to-Neumann map: inexact gradients

and FMG (MG/Opt with full multigrid initialization). There is some deterioration as a result of the inexact gradient, but even so there is a significant improvement over using traditional optimization.

In Figure 6.5 we plot the downdated values of the exact and approximate gradients where the original gradient had $N = 257$ components and the downdated gradient had $N = 129$ components. In Figure 6.6 is a plot of the Fourier transform of the error in the downdated gradient, with the coefficients ordered so the lowest frequencies are at the center. As expected, the error in the downdated gradient consists primarily of low-frequency components, and the error diminishes rapidly as the frequency increases. We computed the errors in the downdated gradient when running MG/Opt. The errors in the gradient were significant, as much as 15 times as large as the gradient.

The errors in the downdated gradient will result in large low-frequency errors in the search direction e_h from the multigrid recursion. We would expect these low-frequency errors to result in a direction that was either too long or pointed in the

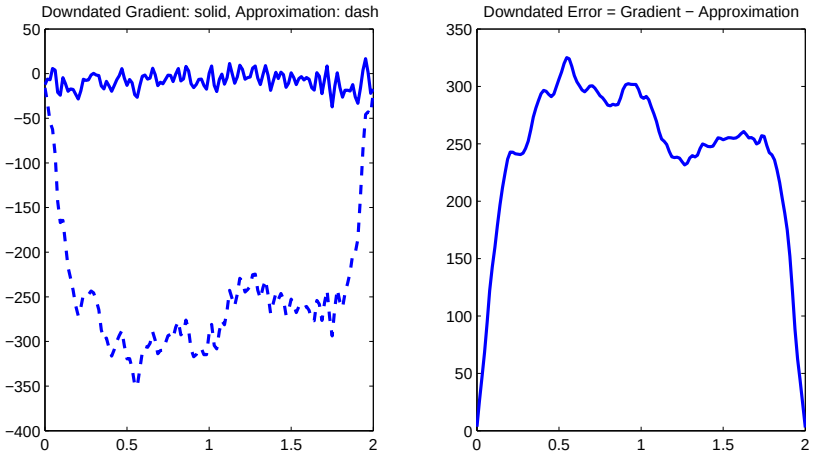

Fig. 6.5: Dirichlet-to-Neumann map: error in gradient

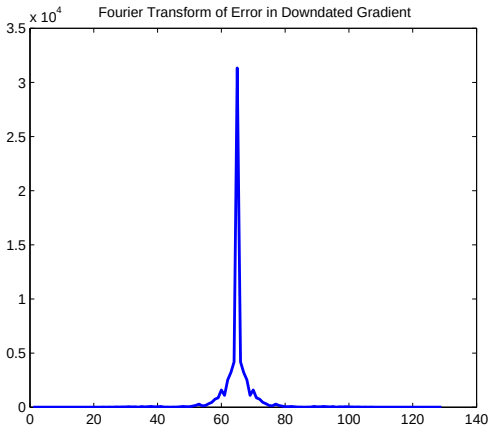

Fig. 6.6: Dirichlet-to-Neumann map: FFT of gradient error

wrong direction. And this is what happened. We can see this by looking more closely at the effect of gradient errors on the line search in the recursion step of MG/Opt, where we compute $x_h^+ = \bar{x}_h + \alpha e_h$. When the exact gradient was used, e_h was always a descent direction, and a step of $\alpha = 1$ was accepted at almost all iterations (indicating that the search direction e_h was well scaled). When an inexact gradient was used, we observed that $\alpha < 1$ at almost every iteration, and on a number of occasions the search direction e_h was not a descent direction.

When full-multigrid initialization is used, we start on the coarsest level and solve the problem there. Then we bootstrap up to the finest level to get an initial point to use in a multigrid V-cycle. Doing this tends to resolve the low-frequency components of the solution during the initialization phase. If this were done, the errors in the

low-frequency components of the solution would already be small, so it ought not to be too serious if the recursion did not resolve these components well. In fact, this is what occurred. There was little deterioration in the performance of the full-multigrid version of MG/Opt.

When using this approach to computing an approximate gradient, we strongly recommend using full-multigrid initialization to compensate for the deficiencies of the approximation.

7. Comments and Conclusions. We have analyzed the impact of errors in fine-level gradients on the coarse-level surrogate model used by the MG/Opt, an optimization-based multilevel method. Four sources of error have been considered. We have shown that if the gradient errors are random then they are damped by standard downdate operators used in multilevel algorithms. If the gradient errors arise from coarse finite-differencing then the errors are dominated by high-frequency terms, which are also damped by the downdate operators.

For PDE-constrained optimization models we considered two other sources of error. If the gradient is obtained by using the adjoint of the PDE we analyzed the impact of using coarse time-stepping. We considered a model problem and demonstrated that the errors in the downdated gradient were negligible. This is because the errors in the gradient are dominated by high-frequency errors which cannot be represented on a coarser grid. This principle will apply in more general settings even if the details of our analysis are specific to a model problem.

When solving the PDE with an iterative method, we also considered using a small number of iterations to estimate the gradient. We again considered a model problem. In this case, however, the errors are dominated by low-frequency terms. This causes the MG/Opt search direction to be poorly scaled, i.e., a step length of $\alpha = 1$ is not accepted and a smaller step length must be used. Even so, MG/Opt can improve significantly on the performance of the underlying single-level optimization algorithm. Using full multigrid initialization can greatly reduce the effects of the low-frequency errors. As with the previous case, the details of the analysis are specific to this model problem, but the principles underlying the analysis have broader applicability.

8. Acknowledgements. The research was supported by the U.S. Department of Energy under Award DE-SC-0001691. We thank Paul Boggs and David Gay for their helpful comments.

REFERENCES

[1] P. T. Boggs, D. M. Gay, S. Griffiths, R. M. Lewis, K. R. Long, S. Nash, and R. H. Nilson, *Optimization algorithms for hierarchical problems, with application to nanoporous materials*, Tech. Report 2011-9036, Sandia National Laboratories, Albuquerque NM and Livermore CA, December 2011.

[2] P. T. Boggs and J. John E. Dennis, *A stability analysis for perturbed nonlinear iterative methods*, Mathematics of Computationa, 30 (1976), pp. 199–215.

[3] R. G. Carter, *On the global convergence of trust region algorithms using inexact gradient information*, SIAM Journal on Numerical Analysis, 28 (1991), pp. 251–265.

[4] ———, *Numerical experience with a class of algorithms for nonlinear optimization using inexact function and gradient information*, SIAM Journal on Scientific Computing, 14 (1993), pp. 368–388.

[5] A. L. Custódio, J. J. E. Dennis, and L. N. Vicente, *Using simplex gradients of nonsmooth functions in direct search methods*, IMA Journal of Numerical Analysis, 28 (2008), pp. 770–784.

[6] T. G. Kolda, R. M. Lewis, and V. Torczon, *Optimization by direct search: New perspectives on some classical and modern methods*, SIAM Review, 45 (2003), pp. 385–482.

[7] R. M. Lewis and S. G. Nash, *Model problems for the multigrid optimization of systems governed by differential equations*, SIAM Journal on Scientific Computing, 26 (2005), pp. 1811–1837.

[8] J. J. Moré, *Recent developments in algorithms and software for trust region methods*, in Mathematical Programming: State of the Art, A. Bachem, M. Grötschel, and B. Korte, eds., Berlin, 1983, Spring-Verlag, pp. 258–287.

[9] S. G. Nash, *A multigrid approach to discretized optimization problems*, Journal of Computational and Applied Mathematics, 14 (2000), pp. 99–116.

[10] ———, *Convergence and descent properties for a class of multilevel optimization algorithms*. `http://www.optimization-online.org/DB_HTML/ 2010/04/2598.html`, 2010.

[11] S. G. Nash and R. M. Lewis, *Assessing the performance of an optimization-based multilevel method*, Optimization Methods and Software, 26 (2011), pp. 695–719.

[12] J. C. Strikwerda, *Finite Difference Schemes and Partial Differential Equations*, Wadsworth & Brooks/Cole, 1989.

[13] P. L. Toint, *Global convergence of a class of trust region methods for nonconvex minimization in Hilbert space*, IMA J. Numerical Analysis, 8 (1988), pp. 231–252.

[14] U. Trottenberg, C. Oosterlee, and A. Schüller, *Multigrid*, Academic Press, London, 2001.