SIAM J. OPTIM.
Vol. 20, No. 3, pp. 1478–1503

A LINE SEARCH MULTIGRID METHOD FOR LARGE-SCALE NONLINEAR OPTIMIZATION*

ZAIWEN WEN[†] AND DONALD GOLDFARB[†]

Abstract. We present a line search multigrid method for solving discretized versions of general unconstrained infinite-dimensional optimization problems. At each iteration on each level, the algorithm computes either a "direct search" direction on the current level or a "recursive search" direction from coarser level models. Introducing a new condition that must be satisfied by a backtracking line search procedure, the "recursive search" direction is guaranteed to be a descent direction. Global convergence is proved under fairly minimal requirements on the minimization method used at all grid levels. Using a limited memory BFGS quasi-Newton method to produce the "direct search" direction, preliminary numerical experiments show that our line search multigrid approach is promising.

Key words. convex and nonconvex optimization, multigrid/multilevel method, multiscale problems, line search, global convergence

AMS subject classifications. 65K05, 65N55, 90C06, 90C25

DOI. 10.1137/08071524X

1. Introduction.

Infinite-dimensional optimization problems are a major source of large-scale finite-dimensional optimization problems [14, 31]. Since in most cases it is not possible to obtain explicit solutions for these problems, they are usually solved numerically either by an "optimize-then-discretize" strategy or a "discretize-then-optimize" strategy. In an "optimize-then-discretize" approach, one first derives the optimization method in an infinite-dimensional space and then discretizes the objective functional and any subproblem that must be solved at each step of the method. In a "discretize-then-optimize" approach, the infinite-dimensional problem is first discretized to obtain a standard nonlinear programming problem in a finite-dimensional space, and then one solves this problem by a nonlinear programming algorithm. In this paper, we follow the "discretize-then-optimize" strategy and propose a new multigrid optimization approach to solve the discretized version of

$$\min_{\mathbf{x}\in\mathcal{V}} \quad \mathcal{F}(\mathbf{x}), \tag{1.1}$$

where $\mathcal{F}$ is a mapping from an infinite-dimensional space $\mathcal{V}$ to $\mathbb{R}$.

The computational cost of solving problem (1.1) using a very fine discretization directly is expensive. Fortunately, a hierarchy of discretized problems ranging from a fairly coarse discretization level to the fine discretization level can be constructed so that the corresponding solutions have similar structures and the problems on the coarser levels are easier to solve than those on the finer levels. The philosophy of multigrid algorithms is to use information from coarser levels to produce new trial points for problems on the finer grids. A simple multigrid method is the technique of mesh refinement [38], where the discretized problems are solved in turn from the coarsest level to the finest level and the starting point at each level other than the coarsest

*Received by the editors February 6, 2008; accepted for publication (in revised form) September 18, 2009; published electronically December 2, 2009. This research was supported in part by NSF grant DMS 06-06712, ONR grants N000140310514 and N00014-08-1-1118, and DOE grants DE-FG01-92ER-25126 and DE-FG02-08ER58562.
http://www.siam.org/journals/siopt/20-3/71524.html

[†]Department of Industrial Engineering and Operations Research, Columbia University, New York, NY 10027 (zw2109@columbia.edu, goldfarb@columbia.edu).

is obtained by prolongating the solution obtained at the previous (i.e., next coarser) level. Multigrid methods [8, 10, 11, 23, 29, 38, 40, 41] have been used very successfully to solve linear elliptic partial differential equations (PDEs). In this approach, coarser grid corrections are recursively imbedded in an iterative process, in combination with so-called relaxation or smoothing steps, to accelerate the convergence on the target grid. Several extensions of the multigrid approach to nonlinear PDEs have been proposed and extensively studied. One such extension, usually referred to as global linearization [24, 38], uses the multigrid method within Newton's method for nonlinear equations to solve the system of linear equations that provides the Newton step at each iteration. Another extension, referred to as local linearization, special cases of which are the full approximation scheme (FAS) [9, 38] and the closely related nonlinear multigrid method (NMGM) [23, 38], applies the multigrid methodology directly to the original system of nonlinear equations and its corresponding system of nonlinear residual equations. A combination of global and local linearization is studied in [42], and a projection multilevel method is proposed for quasi-linear elliptic PDEs in [27, 28, 30], where the system of nonlinear equations is reformulated as a least-squares problem.

Multigrid methods for infinite-dimensional optimization problems have also received considerable attention [1, 2, 3, 5, 6, 15, 37]. However, until recently the essential thrust of these methods has been based on employing multigrid methods for solving the nonlinear equations derived from the optimality conditions for either the original problem (1.1) or its discretized version. In a new approach, Nash (see [25, 26, 33]) proposed a multigrid optimization framework for solving convex infinite-dimensional optimization problems. A proof of the global convergence of Nash's method was given in [4]. This proof requires that at least one iteration of the optimization algorithm that is used at each level be performed either before going to or after returning from a coarser level during a multigrid cycle. These iterations are similar to prior smoothing or postsmoothing steps in multigrid methods for PDEs. Expanding on Nash's approach, Gratton, Sartenaer, and Toint [18, 19, 20] proposed a recursive trust region method that converges even for nonconvex problems to a first-order optimal point without doing such smoothing steps at each multigrid cycle. Alternative convergence results of the recursive trust region method under different assumptions were presented in [22].

In this paper, we propose an easily implementable line search multigrid optimization method under Nash's framework. Our algorithm depends on some *basic* iterative method, such as the steepest descent method, Newton's method, or a quasi-Newton method, and uses coarser grid steps recursively to accelerate the speed of the basic iterative scheme. The basic iterative method is used only when the coarser grid steps cannot satisfy certain criteria. By introducing an additional condition to a backtracking line search procedure, the step generated from the coarser levels is guaranteed to be a descent direction at the current level. We prove global convergence of our line search method without requiring it to take any "smoothing" steps in each multigrid cycle as multigrid algorithms do for PDEs. In our algorithm, smoothing steps are direct search steps of the basic method that are required to be taken before or after a recursive step. We also prove that the convergence rate is at least R-linear in the strictly convex case. Using a limited memory quasi-Newton method (L-BFGS) as the basic iterative scheme, our multigrid method is able to solve very large scale problems efficiently. Our use of L-BFGS is different from its use in the multisecant multigrid L-BFGS method in [21] since we do not derive the secant equation based on the multilevel structure.

This paper is organized as follows. In section 2.1, an easily implementable coarser level model is introduced and a line search multigrid optimization framework is proposed. Our line search procedure for the multigrid method for solving unconstrained nonconvex problems is developed in section 2.2. Proofs of global convergence and R-linear convergence for uniformly convex problems are presented in section 3.1. Global convergence for general nonconvex functions is proved in section 3.2. In section 4, we discuss some techniques to enhance our mulitigrid method, including different ways to generate direct search directions and the use of a so-called *full multigrid* approach. Finally, preliminary numerical results are given in section 5.

We adopt the following notation in this paper: $f_{\ell,k} := f_\ell(x_{\ell,k})$, and $\nabla f_{\ell,k} := \nabla f_\ell(x_{\ell,k})$. Here $x_{\ell,k}$ is a vector, where the first subscript ℓ denotes the discretization level of the multigrid and the second subscript k denotes the iteration count. If a vector has only one subscript, as, for example, x_ℓ, the subscript ℓ either refers to the level of the multigrid, and thus x_ℓ itself is a vector, or it refers the fact that x_ℓ is the ℓth component of the vector x. When it is not clear from the context, we will point out the specific meaning. N is reserved for the index of the finest level and N_0 for the coarsest level.

2. A line search multigrid method.

2.1. A framework for multigrid optimization. Let $\mathcal{V}_\ell$ be a standard finite element space for $\mathcal{V}$ with the basis $\{\phi_\ell^{(j)}\}_{j=1}^{n_\ell}$ at grid level ℓ, where n_ℓ is the dimension of $\mathcal{V}_\ell$. In our framework, for consecutive coarser levels, we choose nested spaces, so that $\mathcal{V}_{\mathrm{N}_0} \subset \cdots \subset \mathcal{V}_{\mathrm{N}-1} \subset \mathcal{V}_{\mathrm{N}} \subset \mathcal{V}$. Given $\mathbf{x}_\ell \in \mathcal{V}_\ell$, there exists a vector $x_\ell = (x_\ell^{(1)}, \ldots, x_\ell^{(n_\ell)})^\top \in \mathbb{R}^{n_\ell}$ such that $\mathbf{x}_\ell = \sum_{j=1}^{n_\ell} x_\ell^{(j)} \phi_\ell^{(j)}$. Defining the discrete functional f_ℓ as $f_\ell(x_\ell) := \mathcal{F}(\mathbf{x}_\ell)$, the discretized version of problem (1.1) on level ℓ is $\min_{x_\ell} f_\ell(x_\ell)$. The main goal of this paper is to design a multigrid method for the finest level problem:

$$\min_{x_\mathrm{N}} f_\mathrm{N}(x_\mathrm{N}). \tag{2.1}$$

Define R_ℓ to be the restriction operator from level ℓ to level $\ell-1$ and P_ℓ to be the prolongation operator from level $\ell-1$ to level ℓ. As in standard multigrid methods, we assume the following.

Assumption 2.1. The prolongation operator P_ℓ and the restriction operator R_ℓ satisfy

$$\sigma_\ell P_\ell = R_\ell^\top. \tag{2.2}$$

For simplicity, we take $\sigma_\ell = 1$, which does not affect our convergence analysis.

Line search algorithms iteratively generate a descent search direction and then search along this direction for a point at which the objective function is suitably reduced. Specifically, starting from the point $x_{\ell,k}$ on level ℓ, a line search algorithm determines the next point as $x_{\ell,k+1} = x_{\ell,k} + \alpha_{\ell,k} d_{\ell,k}$, where $d_{\ell,k}$ is the search direction and $\alpha_{\ell,k}$ is the step size. Our multigrid algorithm based on Nash's method [33, 25] alternates between two kinds of search directions, a *direct search* direction, which is generated on the current level, and a *recursive search* direction, which is generated from steps taken at coarser levels. The construction of these search directions depends on a *basic* iterative scheme, such as the steepest descent method, Newton's method, or a quasi-Newton method, on a single level. As we will see later, most of the computational cost of constructing a recursive direction comes from this basic iterative scheme. To ensure convergence and efficiency, some degree of coherence between the problem at each level and the problem at the next coarser level must be enforced. Hence, the objective function at the coarse level $\ell - 1$ is not simply the discretized function $f_{\ell-1}(x_{\ell-1})$, but rather

$$\psi_{\ell-1}(x_{\ell-1}) = f_{\ell-1}(x_{\ell-1}) - (v_{\ell-1})^\top x_{\ell-1}, \tag{2.3}$$

where $v_{\ell-1} = \nabla f_{\ell-1,0} - R_\ell g_{\ell,k}$ and we have used the notation that $g_{\ell,k} = \nabla \psi_{\ell,k} = \nabla \psi_\ell(x_{\ell,k})$. Furthermore, if we define $v_{\mathrm{N}} = 0$, the model (2.3) can be naturally extended to all levels since the uppermost level model problem is exactly problem (2.1). Also, since the function model $\psi_{\ell-1}$ depends on the point $x_{\ell,k}$ at the next finer level ℓ, it is different for different points. To simplify our notation, we omit this dependence of $\psi_{\ell-1}(\cdot)$'s on $x_{\ell,k}$, hopefully, without introducing any confusion. The same is true for all other quantities such as derivatives of the coarse level function $\psi_{\ell-1}$. Actually, the function (2.3) is a generalization of the coarse grid correction equation of the FAS scheme [38] in the context of optimization. This can be seen by noting the equivalence between the residual equation in the FAS scheme and the gradient of (2.3).

If a *recursive search* direction is chosen, we first move to the next coarser level $\ell - 1$ with an initial point $x_{\ell-1,0} = R_\ell x_{\ell,k}$. Next we compute the minimizer (or approximate minimizer) $x_{\ell-1,i^*}$ of the coarse level problem $\min_{x_{\ell-1}} \psi_{\ell-1}(x_{\ell-1})$, where $\psi_{\ell-1}$ is defined by (2.3) and the cumulative direction

$$\widetilde{d}_{\ell-1,i^*} = x_{\ell-1,i^*} - x_{\ell-1,0} = \sum_{i=0}^{i^*-1} \alpha_{\ell-1,i} d_{\ell-1,i}, \tag{2.4}$$

where $\alpha_{\ell-1,i}$ and $d_{\ell-1,i}$ are the step size and search direction, respectively, for the ith iteration on level $\ell - 1$. Here each search direction $d_{\ell-1,i}$ from $x_{\ell-1,i}$ to $x_{\ell-1,i+1}$ for $i = 0, \dots, i^* - 1$ is also computed recursively whenever possible. Then we prolongate the direction $\widetilde{d}_{\ell-1,i^*}$ on level $\ell-1$ back to level ℓ to obtain the recursive search direction

$$d_{\ell,k} = P_\ell \widetilde{d}_{\ell-1,i^*}. \tag{2.5}$$

The following lemma gives some properties of this recursive scheme.

LEMMA 2.2. *If the minimization on the coarse level $\ell - 1$ starts from the initial point $x_{\ell-1,0} = R_\ell x_{\ell,k}$ and stops at $x_{\ell-1,i^*}$, and the recursive direction is defined as $d_{\ell,k} = P_\ell \widetilde{d}_{\ell-1,i^*}$, where $\widetilde{d}_{\ell-1,i^*} = x_{\ell-1,i^*} - x_{\ell-1,0}$, then the problems of the two consecutive levels ℓ and $\ell - 1$ are first-order coherent in the sense that*

$$g_{\ell-1,0} = R_\ell g_{\ell,k}, \quad (d_{\ell,k})^\top g_{\ell,k} = (\widetilde{d}_{\ell-1,i^*})^\top g_{\ell-1,0}. \tag{2.6}$$

Suppose $f_{\ell-1}(x_{\ell-1})$ is a convex function and $\psi_{\ell-1}(x_{\ell-1,i^*}) < \psi_{\ell-1}(x_{\ell-1,0})$; then $d_{\ell,k}$ is a descent direction, that is, $(d_{\ell,k})^\top g_{\ell,k} < 0$. Moreover, the directional derivative $(d_{\ell,k})^\top g_{\ell,k}$ satisfies

$$-(d_{\ell,k})^\top g_{\ell,k} \geq \psi_{\ell-1,0} - \psi_{\ell-1,i^*}. \tag{2.7}$$

Proof. The first part of (2.6) comes from the fact that

$$g_{\ell-1,0} = \nabla f_{\ell-1,0} - v_{\ell-1} = \nabla f_{\ell-1,0} - \nabla f_{\ell-1,0} + R_\ell g_{\ell,k} = R_\ell g_{\ell,k}.$$

This, together with (2.5) and (2.2) and our assumption that $\sigma_\ell = 1$, implies that

$$(d_{\ell,k})^\top g_{\ell,k} = (P_\ell \widetilde{d}_{\ell-1,i^*})^\top g_{\ell,k} = (\widetilde{d}_{\ell-1,i^*})^\top R_\ell g_{\ell,k} = (\widetilde{d}_{\ell-1,i^*})^\top g_{\ell-1,0}.$$

If $f_{\ell-1}(x_{\ell-1})$ is convex, then so is $\psi_{\ell-1}(x_{\ell-1})$; hence

$$\psi_{\ell-1}(x_{\ell-1,i^*}) \geq \psi_{\ell-1}(x_{\ell-1,0}) + (x_{\ell-1,i^*} - x_{\ell-1,0})^\top g_{\ell-1,0}. \tag{2.8}$$

Hence we conclude from (2.4) that inequality (2.7) holds. Then from the fact that $\psi_{\ell-1}(x_{\ell-1,i^*}) < \psi_{\ell-1}(x_{\ell-1,0})$, it follows that $(d^*_{\ell-1})^\top g_{\ell-1,0} < 0$. □

Remark 2.3. Lemma 2.2 shows that the coarser model $\psi_{\ell-1}(x_{\ell-1})$ defined by (2.3) is a first-order approximation to the finer model $\psi_\ell(x_\ell)$. The property (2.6) is essential for our line search procedure to construct a descent direction from steps taken on coarser levels. It is also possible to design a second-order coherent model using information about the Hessian. The requirement (2.6) on the function $f_{\ell-1}(x_{\ell-1})$ is not restrictive; even $f_{\ell-1}(x_{\ell-1}) := 0$ is possible. However, we use the discretization $f_\ell(x_\ell)$ of the continuous function on each level ℓ, as this is a natural choice. Although Lemma 2.2 shows that the recursive direction $d_{\ell,k}$ defined by (2.5) is a descent direction for convex problems, this is not the case in general for nonconvex problems.

We now specify conditions for when to choose a direct search direction on the current level. Specifically, $d_{\ell,k}$ is computed directly on level ℓ if

$$\|R_\ell g_{\ell,k}\| < \kappa \|g_{\ell,k}\| \quad \text{or} \quad \|R_\ell g_{\ell,k}\| < \epsilon_\ell \tag{2.9}$$

holds, where $\kappa \in (0, \min(1, \min_\ell \|R_\ell\|))$ and $\epsilon_\ell \in (0,1)$ is the tolerance on the first-order optimality conditions on level ℓ. The reason for this is that $R_\ell g_{\ell,k}$ may be zero while $g_{\ell,k}$ is not if $g_{\ell,k}$ lies in the null space of R_ℓ; hence the current iterate appears to be a stationary point for $\psi_{\ell-1}$ whereas it is not for ψ_ℓ. These conditions were first used in the multigrid algorithm proposed in [19, 20, 18]. We also compute the search direction $d_{\ell,k}$ directly if the current point $x_{\ell,k}$ is very close to the point $\tilde{x}_\ell$ at which the last recursive step on level ℓ was initialized (i.e., if $\|x_{\ell,k} - \tilde{x}_\ell\| < \epsilon_x \|\tilde{x}_\ell\|$, where $\epsilon_x \in (0,1)$) as long as the algorithm has performed fewer than K_d consecutive direct search steps. The motivation for the first part of these conditions is that doing a new recursive step at this point will yield a result that is similar to what was obtained on the last recursive step.

Many unconstrained optimization algorithms can be used to compute a direct search direction. In particular, we are able to prove global convergence if this direction satisfies the following condition.

CONDITION 2.4. *The step direction $d_{\ell,k}$ satisfies*

$$\|d_{\ell,k}\| \leq \beta_{\mathcal{D}} \|g_{\ell,k}\| \quad \textit{and} \quad -(d_{\ell,k})^\top g_{\ell,k} \geq \eta_{\mathcal{D}} \|g_{\ell,k}\|^2, \tag{2.10}$$

where $\beta_{\mathcal{D}}$ and $\eta_{\mathcal{D}}$ are positive constants.

We state our line search multigrid method formally in Algorithm 1 below. We will present our line search procedure for choosing the step size $\alpha_{\ell,k}$ in subsection 2.2.

Remark 2.5. If a recursive step is taken, then the condition $\|R_\ell g_{\ell,k}\| \geq \kappa \|g_{\ell,k}\|$ always holds (see Step 3.2 in Algorithm 1). We note that the recursive routine $MLS(\cdot,\cdot,\cdot)$ terminates if either the norm of the gradient at the current iterate is smaller than some prescribed tolerance ϵ_ℓ for the level ℓ on which it is operating or the total number of iterations (direct or recursive) performed on that level exceeds some upper bound K. Routine $MLS(\cdot,\cdot,\cdot)$ is also terminated if the acceptable step size $\alpha_{\ell,k} < \xi$, where ξ is a small constant, since this is an indication that it may not continue to make significant progress on the current level.

Algorithm 1 $x_\ell = MLS(\ell, x_{\ell,0}, \tilde{g}_{\ell,0})$.

```
Step 1. Set $\kappa, \epsilon_\ell(\ell = \mathrm{N}_0, \dots, \mathrm{N}), \epsilon_x, \xi \in (0,1)$. Set integers K and $\mathrm{K}_d$ with $0 < \mathrm{K}_d <$ K. Set $q = 0$, $\tilde{x}_\ell = \inf$.
Step 2. IF $\ell < \mathrm{N}$, compute $v_\ell = \nabla f_{\ell,0} - \tilde{g}_{\ell,0}$ and set $g_{\ell,0} = \tilde{g}_{\ell,0}$;
        ELSE set $v_\ell = 0$ and compute $g_{\ell,0} = \nabla f_{\ell,0}$.
Step 3. FOR $k = 0, 1, 2, \dots$
    3.1. IF $\|g_{\ell,k}\| \le \epsilon_\ell$ or IF $\ell < \mathrm{N}$ and $k \ge \mathrm{K}$,
             RETURN solution $x_{\ell,k}$;
    3.2. IF $\ell = \mathrm{N}_0$ or $\|R_\ell g_{\ell,k}\| < \kappa\|g_{\ell,k}\|$ or $\|R_\ell g_{\ell,k}\| < \epsilon_\ell$ or (both $\|x_{\ell,k} - \tilde{x}_\ell\| < \epsilon_x\|\tilde{x}_\ell\|$ and $q < \mathrm{K}_d$)
             -Direct Search Direction Computation.
             Compute a descent search direction $d_{\ell,k}$ on the current level. Set $q = q + 1$.
         ELSE
             -Recursive Search Direction Computation.
             Call $x_{\ell-1,i^*} = MLS(\ell - 1, R_\ell x_{\ell,k}, R_\ell g_{\ell,k})$ to return a solution (or approximate solution) $x_{\ell-1,i^*}$ of "$\min_{x_{\ell-1}} \psi_{\ell-1}(x_{\ell-1})$."
             Compute $d_{\ell,k} = P_\ell \widetilde{d}_{\ell-1,i^*} = P_\ell(x_{\ell-1,i^*} - R_\ell x_{\ell,k})$. Set $q = 0$ and $\tilde{x}_\ell = x_{\ell,k}$.
    3.3. Call a line search procedure to obtain a step size $\alpha_{\ell,k}$.
    3.4. Set $x_{\ell,k+1} = x_{\ell,k} + \alpha_{\ell,k} d_{\ell,k}$. IF $\alpha_{\ell,k} \le \xi$ and $\ell < \mathrm{N}$, RETURN solution $x_{\ell,k+1}$.
```

2.2. A new line search procedure. Several approaches can be used to make sure that the recursive search direction is a descent direction. One method is to modify the coarse level function $\psi_{\ell-1}$ by adding an additional quadratic term $\lambda_{\ell-1}\|x_{\ell-1} - x_{\ell-1,0}\|_2^2$ so that the function

$$\widehat{\psi}_{\ell-1} = \psi_{\ell-1} + \lambda_{\ell-1}\|x_{\ell-1} - x_{\ell-1,0}\|_2^2 \tag{2.11}$$

is convex, where $\lambda_{\ell-1}$ is a sufficiently large parameter. Then solving the convex problem $\min_{x_{\ell-1}} \widehat{\psi}_{\ell-1}$ provides a descent direction. It is difficult, however, to choose an appropriate parameter $\lambda_{\ell-1}$. An alternative (and related) approach is the recursive trust region method proposed in [20, 18] in which each subproblem is solved subject to a trust region constraint. The trust region ball is shrunk iteratively until the recursive search direction provides a sufficient reduction in the objective function on the finer level ℓ. Essentially, this is equivalent to using (2.11) but with an automatic mechanism for determining an appropriate $\lambda_{\ell-1}$. This strategy may be expensive, especially on levels with a lot of variables, since the entire minimization sequence, as well as the computations on the coarser levels for recursive steps, will be discarded if the solution of the trust region subproblem does not yield a sufficient reduction in the model function.

We now describe our line search procedure based on a backtracking line search scheme [35, 36]. We choose two constants ρ_1 and ρ_2 such that $0 < \rho_1 < \frac{1}{2}$ and $1 - \rho_1 \le \rho_2 \le 1$. If ℓ is the finest level, we require that the step size $\alpha_{\ell,k}$ along $d_{\ell,k}$ satisfy the Armijo condition

$$\psi_\ell(x_{\ell,k} + \alpha_{\ell,k} d_{\ell,k}) \le \psi_{\ell,k} + \rho_1 \alpha_{\ell,k} (g_{\ell,k})^\top d_{\ell,k}. \tag{2.12}$$

If ℓ is any level other than the finest level, we require that the step size $\alpha_{\ell,k}$ along $d_{\ell,k}$ also satisfy the condition

$$\psi_\ell(x_{\ell,k} + \alpha_{\ell,k} d_{\ell,k}) > \psi_{\ell,0} + \rho_2 g_{\ell,0}^\top (x_{\ell,k} + \alpha_{\ell,k} d_{\ell,k} - x_{\ell,0}), \tag{2.13}$$

in addition to the Armijo condition (2.12).

Note that condition (2.13) is similar to the Goldstein rule if $k = 0$ and $\rho_2 = 1 - \rho_1$, i.e., on the first step in a minimization sequence at level ℓ. However, its use here is very different. In the Goldstein rule, the inequality is based on the starting point of the current step on level ℓ and ensures that the step is not too small. Here it is based on the initial point in the current minimization sequence on level ℓ (i.e., at the start of a recursive step taken at level $\ell+1$), and it ensures that the decrease in the first-order Taylor series approximation to $\psi_\ell(\cdot)$ obtained by taking the step $x_{\ell,k+1} - x_{\ell,0}$ is at least as great as $1/\rho_2$ times the decrease in $\psi_\ell(\cdot)$ (i.e., $\psi_{\ell,k+1} - \psi_{\ell,0}$). It then follows from the first-order coherence relation (2.6) that the above statement also holds on level $\ell + 1$. Specifically, the following lemma tells us that if condition (2.13) holds for all steps on level $\ell - 1$ during a recursive search step on level ℓ, then the latter is a descent step on level ℓ.

Lemma 2.6. *Suppose that a recursive search direction is computed by Algorithm* 1 *at $x_{\ell,k}$ on level ℓ. If conditions* (2.12) *and* (2.13) *hold on the coarse level $\ell - 1$ from iteration $i = 0$ to i^*, then the recursive step $d_{\ell,k}^{(i)} = P_\ell \widetilde{d}_{\ell-1,i}$, where $\widetilde{d}_{\ell-1,i} = x_{\ell-1,i} - x_{\ell-1,0}$, is a descent direction, and in particular,*

$$-(g_{\ell,k})^\top d_{\ell,k}^{(i)} > \rho_2^{-1}(\psi_{\ell-1,0} - \psi_{\ell-1,i}). \tag{2.14}$$

Proof. Since on level $\ell - 1$ condition (2.13) can be rewritten as

$$\psi_{\ell-1}(x_{\ell-1,i}) > \psi_{\ell-1,0} + \rho_2 g_{\ell-1,0}^\top \widetilde{d}_{\ell-1,i}, \tag{2.15}$$

and $g_{\ell-1,0}^\top \widetilde{d}_{\ell-1,i} = (R_\ell g_{\ell,k})^\top \widetilde{d}_{\ell-1,i} = g_{\ell,k}^\top P_\ell \widetilde{d}_{\ell-1,i}$, it follows that the direction $d_{\ell,k}^{(i)} = P_\ell \widetilde{d}_{\ell-1,i}$ satisfies (2.14). □

To select a step size $\alpha_{\ell,k}$ that satisfies these conditions, we use a traditional backtracking scheme in Algorithm 2.

```
Algorithm 2 Backtracking Line Search.
Step 1. Given $\alpha_\rho > 0$, $0 < \rho_1 < \frac{1}{2}$ and $1 - \rho_1 \le \rho_2 \le 1$ ($1 - \rho_1 \le \rho_2 < 1$ in the
        nonconvex case). Let $\alpha^{(0)} = \alpha_\rho$. Set $t = 0$.
Step 2. If ($\ell = N$ and condition (2.12) is satisfied) or if ($\ell < N$ and both conditions
        (2.13) and (2.12) are satisfied), Return $\alpha_{\ell,k} = \alpha^{(t)}$.
Step 3. Set $\alpha^{(t+1)} = \tau\alpha^{(t)}$, where $\tau \in (0, 1)$. Set $t = t + 1$ and go to Step 2.
```

The existence of a step size satisfying the Armijo condition (2.12) for a descent direction follows from the differentiability of ψ_ℓ and the fact that $d_{\ell,k}^\top g_{\ell,k} < 0$. Hence, the pure backtracking phase in Algorithm 2 when $\ell = \mathrm{N}$ is well defined. We now prove that there exists a step size that satisfies both conditions (2.12) and (2.13) when $\ell < \mathrm{N}$.

Lemma 2.7. *Suppose $\ell < \mathrm{N}$ and $\psi_\ell(x_\ell)$ is continuously differentiable. Let $d_{\ell,k}$ be a descent direction at $x_{\ell,k}$ (i.e., $(g_{\ell,0})^\top d_{\ell,k} < 0$), and assume that ψ_ℓ is bounded from below along the ray $\{x_{\ell,k} + \alpha d_{\ell,k} \mid \alpha > 0\}$ for all $k \ge 0$. Then if $0 < \rho_1 < \frac{1}{2}$ and*

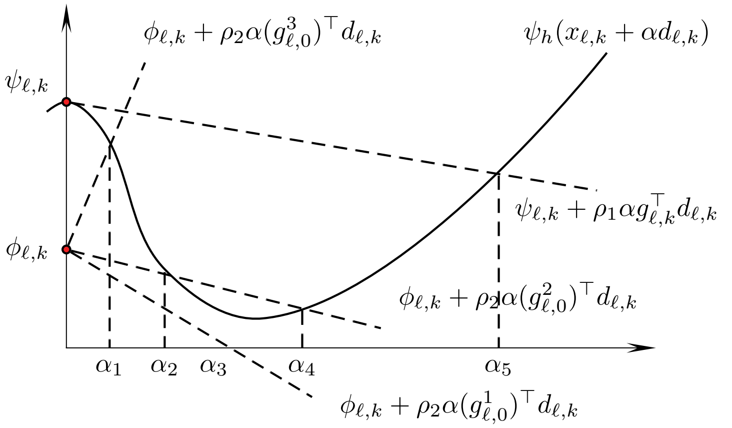

$\phi_{\ell,k} + \rho_2\alpha(g^3_{\ell,0})^\top d_{\ell,k}$
$\psi_h(x_{\ell,k} + \alpha d_{\ell,k})$
$\psi_{\ell,k}$
$\psi_{\ell,k} + \rho_1\alpha g^\top_{\ell,k} d_{\ell,k}$
$\phi_{\ell,k}$
$\phi_{\ell,k} + \rho_2\alpha(g^2_{\ell,0})^\top d_{\ell,k}$
α_1
α_2
α_3
α_4
α_5
$\phi_{\ell,k} + \rho_2\alpha(g^1_{\ell,0})^\top d_{\ell,k}$

FIG. 2.1. *An illustration of the line search procedure. If $g_{\ell,0} = g^1_{\ell,0}$, then the acceptable interval is $[0, \alpha_5)$. If $g_{\ell,0} = g^2_{\ell,0}$, then the acceptable interval is $[0, \alpha_2) \cup (\alpha_4, \alpha_5]$. If $g_{\ell,0} = g^3_{\ell,0}$, then the acceptable interval is $[0, \alpha_1)$.*

$1 - \rho_1 \leq \rho_2 \leq 1$, *there exist intervals of step lengths satisfying both conditions* (2.12) *and* (2.13) *for all* $k \geq 0$.

Proof. Since $y_k(\alpha) = \psi_\ell(x_{\ell,k} + \alpha d_{\ell,k})$ is bounded from below for all $\alpha > 0$ and since $0 < \rho_1 < \frac{1}{2}$, the line $t_k(\alpha) = \psi_{\ell,k} + \rho_1 \alpha g^\top_{\ell,k} d_{\ell,k}$ must intersect the graph of $y_k(\alpha)$ at least once, since for small enough $\alpha > 0$, $t_k(\alpha)$ lies above $y_k(\alpha)$, i.e., $t_k(\alpha) > y_k(\alpha)$. Let $\bar{\alpha} > 0$ be the smallest value of α where this occurs. Define the term

$$\phi_{\ell,k} \overset{def}{=} \psi_{\ell,0} + \rho_2 g^\top_{\ell,0}(x_{\ell,k} - x_{\ell,0}); \tag{2.16}$$

then condition (2.13) can be rewritten as

$$\psi_\ell(x_{\ell,k} + \alpha_{\ell,k} d_{\ell,k}) > \phi_{\ell,k} + \alpha_{\ell,k}\rho_2 g^\top_{\ell,0} d_{\ell,k}. \tag{2.17}$$

If $k = 0$, then $\psi_{\ell,k} = \phi_{\ell,k}$. The line $\hat{t}_0(\alpha) = \psi_{\ell,0} + \rho_2\alpha(g_{\ell,0})^\top d_{\ell,0}$ lies below the line $t_0(\alpha)$ since $\rho_1 < \frac{1}{2} < 1 - \rho_1 \leq \rho_2 \leq 1$ but above $y(\alpha)$ for small enough $\alpha > 0$. Therefore, the line $\hat{t}_0(\alpha)$ must intersect the graph of $y_0(\alpha)$ at least once. Let α' be the smallest value of α where this occurs. Clearly, $0 < \alpha' < \bar{\alpha}$ and $(\alpha', \bar{\alpha}]$ is an interval in which both conditions hold.

Suppose the lemma holds for iteration $k - 1$. We now prove that it also holds for iteration k. Since condition (2.13) is satisfied at iteration $(\ell, k-1)$, we obtain that

$$\begin{aligned} \psi_{\ell,k} &= \psi_\ell(x_{\ell,k-1} + \alpha_{\ell,k-1} d_{\ell,k-1}) \\ &> \psi_{\ell,0} + \rho_2 g^\top_{\ell,0}(x_{\ell,k-1} + \alpha_{\ell,k-1} d_{\ell,k-1} - x_{\ell,0}) \\ &= \phi_{\ell,k}, \end{aligned}$$

which implies that there is an interval $[0, \alpha''')$ with $\alpha''' > 0$ ($\alpha''' = \infty$ is possible), in which the line $\hat{t}_k(\alpha) = \phi_{\ell,k} + \alpha\rho_2 g^\top_{\ell,0} d_{\ell,k}$ lies below the graph of $y_k(\alpha)$ no matter what the slope $g^\top_{\ell,0} d_{\ell,k}$ of $\hat{t}_k(\alpha)$ is. If $\hat{t}_k(\alpha)$ intersects $y_k(\alpha)$ at least once with the smallest intersecting value $0 < \alpha'' < \alpha$, then $[0, \alpha'')$ is an interval in which both conditions hold. Otherwise, the line $\hat{t}_k(\alpha)$ lies below $y_k(\alpha)$ in $[0, \alpha)$, and this interval satisfies the requirement. Figure 2.1 illustrates several possible cases. ▯

We note that if step sizes at level $\ell - 1$ are chosen to satisfy commonly used line search conditions such as the Armijo–Wolfe conditions, the direction $d_{\ell,k}$ may not be a descent direction.

3. Convergence analysis.

Throughout this section, we define

$$\varpi := \max\{1, \max_{i=\mathrm{N}_0,\dots,\mathrm{N}} \|P_i\|, \max_{i=\mathrm{N}_0,\dots,\mathrm{N}} \|R_i\|\} < \infty \tag{3.1}$$

and adopt some concepts and notation from [19, 20, 18]. We shall refer to the kth iteration on level ℓ as iteration (ℓ, k). We define the iteration (ℓ, k) as the predecessor of a minimization sequence that consists of all successive iterations on and below level $\ell - 1$ until a return is made to level ℓ. For iteration (ℓ, k), we define the set

$$\mathcal{R}(\ell, k) := \{(j,t) \mid \text{iteration } (j,t) \text{ occurs within iteration } (\ell, k)\} \tag{3.2}$$

and define the deepest level in $\mathcal{R}(\ell, k)$ by

$$p(\ell, k) := \min_{(j,t)\in\mathcal{R}(\ell,k)} j. \tag{3.3}$$

We denote the subset of iterations $(j,t) \in \mathcal{R}(\ell, k)$ for which $d_{j,t}$ is a direct search direction by

$$\mathcal{D}(\ell, k) := \{(j,t) \in \mathcal{R}(\ell, k) \mid d_{j,t} \text{ is a direct search direction}\}. \tag{3.4}$$

We also denote the Hessian of ψ_ℓ by $G_{\ell,k} = \nabla^2 \psi_{\ell,k} = \nabla^2 \psi_\ell(x_{\ell,k})$.

3.1. Uniformly convex problems.

In this subsection, we consider Algorithm 1 with $\rho_2 = 1$ under the following.

Assumption 3.1. $f_\ell(x)$ is twice continuously differentiable and uniformly convex; that is, there exist constants $0 < \chi_\ell < M_\ell < \infty$ such that

$$\chi_\ell \|d\|_2^2 \le d^\top \nabla^2 f_\ell(x) d \le M_\ell \|d\|_2^2 \quad \text{for all } d \in \Re^{n_\ell} \tag{3.5}$$

for all $x \in \{x \mid f_\ell(x_\ell) \le f_\ell(x_{\ell,0})\}$. Moreover, let $\chi = \min_\ell\{\chi_\ell\}$, $M = \max_\ell\{M_\ell\}$.

Since $\psi_\ell(x)$ differs from $f_\ell(x)$ only by a linear term, Assumption 3.1 also holds for $\psi_\ell(x)$. The following lemma shows that when $\rho_2 = 1$, condition (2.13) or equivalently (2.15) is always satisfied under Assumption 3.1.

Lemma 3.2. *Suppose $\psi_\ell(x)$ satisfies Assumption* 3.1*. Then for any $x_{\ell,k} \neq x_{\ell,0}$, condition* (2.13) *is always satisfied and Algorithm* 2 *is the traditional Armijo backtracking line search procedure.*

Proof. Since $\psi_\ell(x)$ satisfies Assumption 3.1 and $x_\ell \neq x_{\ell,0}$,

$$\begin{aligned} \psi_\ell(x_\ell) &\ge \psi_{\ell,0} + g_{\ell,0}^\top (x_\ell - x_{\ell,0}) + \frac{\chi}{2}\|x_\ell - x_{\ell,0}\|^2 \\ &> \psi_{\ell,0} + g_{\ell,0}^\top (x_\ell - x_{\ell,0}), \end{aligned}$$

which implies that condition (2.15) is always satisfied. □

The following theorem shows that the step size generated by the backtracking line search procedure is bounded from below.

Theorem 3.3. *Suppose that $\psi_\ell \in C^1$, $\nabla\psi_\ell$ is Lipschitz continuous with Lipschitz constant L_ℓ, $\rho_1 \in (0,1)$, and $d_{\ell,k}$ is a descent direction at $x_{\ell,k}$. Then the Armijo condition* (2.12) *is satisfied for all $\alpha \in [0, \widehat{\alpha}]$, where $\widehat{\alpha} = \frac{2(\rho_1 - 1)d_{\ell,k}^\top g_{\ell,k}}{L_\ell \|d_{\ell,k}\|_2^2}$ and the step size generated by the backtracking algorithm, Algorithm* 2*, terminates with*

$$\min\left(\alpha_\rho, \frac{2\tau(\rho_1 - 1)d_{\ell,k}^\top g_{\ell,k}}{L_\ell \|d_{\ell,k}\|_2^2}\right) \le \alpha_{\ell,k} \le \alpha_\rho, \tag{3.6}$$

where τ is the step size reduction parameter in Algorithm 2.

Proof. 1. Since $\nabla\psi_\ell$ is Lipschitz continuous, it follows from Taylor's theorem (Theorem 1.2.22 in [36]) that

$$\psi_\ell(x_{\ell,k}+\alpha d_{\ell,k}) \le \psi_{\ell,k} + \alpha g_{\ell,k}^\top d_{\ell,k} + \frac{1}{2}L_\ell \alpha^2 \|d_{\ell,k}\|^2.$$

Then for all $\alpha \in [0,\widehat{\alpha}]$, we obtain

$$\psi_\ell(x_{\ell,k}+\alpha d_{\ell,k}) \le \psi_{\ell,k} + \alpha g_{\ell,k}^\top d_{\ell,k} + \alpha(\rho_1 - 1) g_{\ell,k}^\top d_{\ell,k},$$

which implies that the Armijo condition (2.12) is satisfied for such α.

2. Clearly, the initial step size α_ρ is an upper bound on the step size $\alpha_{\ell,k}$. Since according to Lemma 3.2 only the Armijo condition (2.12) plays a role in Algorithm 2, the line search will terminate as soon as $\alpha_{\ell,k} \le \widehat{\alpha}$. If the initial step size α_ρ satisfies the Armijo condition, then $\alpha_{\ell,k} = \alpha_\rho$. If not, there is an iteration, say the tth, such that $\alpha^{(t)} > \widehat{\alpha} \ge \alpha^{(t+1)}$. Then $\alpha_{\ell,k} = \alpha^{(t+1)} = \tau\alpha^{(t)} > \tau\widehat{\alpha}$. Combining these two cases gives the required result. □

The following lemmas give some useful properties of convex functions.

LEMMA 3.4 (see [36, Lemma 5.3.4]). *Suppose $f_\ell(x)$, and hence $\psi_\ell(x)$, satisfy Assumption* 3.1.

1. *If $\psi_\ell(y) \le \psi_\ell(x)$, then*

$$\|\nabla\psi_\ell(x)\| \ge \frac{\chi}{2}\|y - x\|. \tag{3.7}$$

2. *For all x,*

$$\frac{\chi}{2}\|x - x^*\|^2 \le \psi_\ell(x) - \psi_\ell(x^*) \le \frac{1}{\chi}\|\nabla\psi_\ell(x)\|^2, \tag{3.8}$$

where x^* is the unique minimizer of $\psi_\ell(x)$.

LEMMA 3.5 (see [36, Theorem 2.5.8]). *Suppose $\psi_\ell(x)$ satisfies Assumption* 3.1. *If α is a step size that satisfies the Armijo condition* (2.12) *along a descent direction d, then the decrease of $\psi_\ell(x)$ satisfies $\psi_\ell(x) - \psi_\ell(x+\alpha d) \ge c_1\|\alpha d\|^2$ with $c_1 = \frac{\rho_1\chi}{1+\sqrt{M/\chi}}$.*

We will also make use of the following inequality.

LEMMA 3.6. *Let $d_1, d_2, \ldots, d_k$ be vectors in $\mathbb{R}^n$. Then $\sum_{j=1}^k \|d_j\|^2 \ge \frac{1}{k}\|\sum_{j=1}^k d_j\|^2$.*

Proof. We prove this lemma by induction on k. The result is trivial if $k = 1$. Suppose the inequality is true for $k-1$; we now prove that it is also true for k:

$$\begin{aligned}\sum_{j=1}^k \|d_j\|^2 - \frac{1}{k}\left\|\sum_{j=1}^k d_j\right\|^2 &\ge \frac{1}{k-1}\left\|\sum_{j=1}^{k-1} d_j\right\|^2 + \|d_k\|^2 \\ &\quad - \frac{1}{k}\left(\left\|\sum_{j=1}^{k-1} d_j\right\|^2 + \|d_k\|^2 + 2\left(\sum_{j=1}^{k-1} d_j\right)^\top d_k\right) \\ &= \frac{1}{k}\left(\frac{1}{k-1}\left\|\sum_{j=1}^{k-1} d_j\right\|^2 + (k-1)\|d_k\|^2 - 2\left(\sum_{j=1}^{k-1} d_j\right)^\top d_k\right) \\ &\ge 0,\end{aligned}$$

where the last inequality comes from the Cauchy–Schwarz inequality and the fact that $\frac{1}{\epsilon}a^2 + \epsilon b^2 \ge 2ab$ for arbitrary scalars a and b and $\epsilon > 0$. This proves the lemma. □

We now derive a lower bound on the step size for any search direction.

LEMMA 3.7. *Suppose Condition* 2.4 *is satisfied by all direct search steps and Assumption* 3.1 *holds. Then the step size $\alpha_{j,t} \geq \alpha_{\mathcal{D}}$ for $(j,t) \in \mathcal{D}(\ell,k)$ and $\alpha_{j,t} \geq \alpha_{\mathcal{I}}$ for $(j,t) \in \mathcal{R}(\ell,k)\backslash\mathcal{D}(\ell,k)$, where*

$$\alpha_{\mathcal{D}} = \min\left(\alpha_\rho, \frac{2\tau(1-\rho_1)\eta_{\mathcal{D}}}{M\beta_{\mathcal{D}}^2}\right), \quad \alpha_{\mathcal{I}} = \min\left(\alpha_\rho, \frac{2\tau c_1(1-\rho_1)}{M\mathrm{K}\varpi^2}\right),$$

and K, *specified in Algorithm* 1, *is the maximum number of iterations of the minimization sequence at level $j-1$. Therefore,*

$$\alpha_{j,t} \geq \alpha^* = \min\{\alpha_{\mathcal{D}}, \alpha_{\mathcal{I}}\} \tag{3.9}$$

for any $(j,t) \in \mathcal{R}(\ell,k)$.

Proof. 1. Since ψ_ℓ satisfies Assumption 3.1, it follows from Theorem 3.3 that the step size

$$\alpha_{j,t} \geq \min\left(\alpha_\rho, \frac{2\tau(\rho_1-1)d_{j,t}^\top g_{j,t}}{M\|d_{j,t}\|_2^2}\right), \tag{3.10}$$

since the Lipschitz constant L_ℓ can be taken to be M.

2. If iteration $(j,t) \in \mathcal{D}(\ell,k)$ satisfies Condition 2.4, then it can be verified directly from (3.10) that $\alpha_{j,t} \geq \alpha_{\mathcal{D}}$.

3. Now consider iteration $(j,t) \in \mathcal{R}(\ell,k)\backslash\mathcal{D}(\ell,k)$. From inequality (2.7), it follows that

$$-(d_{j,t})^\top g_{j,t} \geq \psi_{j-1,0} - \psi_{j-1,i^*}.$$

Since the sequence $\{\psi_{j-1,i}\}$ is monotonically decreasing, the reduction of the function value satisfies

$$\psi_{j-1,0} - \psi_{j-1,i^*} \geq \sum_{k=0}^{i^*-1} \psi_{j-1,k} - \psi_{j-1,k+1}.$$

Since ψ_{j-1} is uniformly convex, it follows from Lemma 3.5 that

$$\psi_{j-1,k} - \psi_{j-1,k+1} \geq c_1\|\alpha_{j-1,k}d_{j-1,k}\|^2.$$

Using Lemma 3.6 and the fact that the total number of iterations at level $\ell-1$ is less than K, we have

$$-(d_{j,t})^\top g_{j,t} \geq c_1\frac{1}{i^*}\left\|\sum_{k=0}^{i^*-1}\alpha_{j-1,k}d_{j-1,k}\right\|^2 \geq \frac{c_1}{\mathrm{K}}\|d_{j-1}^*\|^2 \geq \frac{c_1}{\mathrm{K}\varpi^2}\|d_{j,t}\|^2, \tag{3.11}$$

where the last inequality comes from the fact that $d_{j,t}$ is a prolongation of d_{j-1}^* and

$$\|d_{j,t}\| = \|P_j d_{j-1}^*\| \leq \|P_j\|\,\|d_{j-1}^*\| \leq \varpi\|d_{j-1}^*\|.$$

Therefore, combining (3.10) and (3.11), we obtain $\alpha_{j,t} \geq \alpha_{\mathcal{I}}$, which completes the proof. □

Remark 3.8. We have shown in Lemma 2.2 that the recursive search direction is a descent direction. Therefore, for the convex case, the backtracking algorithm,

Algorithm 2, can be replaced by other line search procedures and Algorithm 1 still works. For example, the Armijo–Wolfe conditions require that $\alpha_{\ell,k}$ satisfy condition (2.12) as well as the curvature condition

$$(\nabla\psi_\ell(x_{\ell,k} + \alpha_{\ell,k} d_{\ell,k}))^\top d_{\ell,k} \geq \rho_2 (g_{\ell,k})^\top d_{\ell,k}, \tag{3.12}$$

where $0 < \rho_1 < \rho_2 < 1$ are the two controlling parameters. In this case, using the uniform convexity of ψ_ℓ, we have

$$\alpha_{j,t} M \|d_{j,t}\|^2 \geq (d_{j,t})^\top [\nabla\psi_\ell(x_{j,t} + \alpha_{j,t} d_{j,t}) - \nabla\psi_\ell(x_{j,t})] \geq -(1-\rho_2)(g_{j,t})^\top d_{j,t}$$

for any iteration $(j,t) \in \mathcal{R}(\ell,k)$. Hence, the step size $\alpha_{j,t}$ is bounded from below by

$$\alpha_{j,t} \geq (1-\rho_2) \frac{|(g_{j,t})^\top d_{j,t}|}{M\|d_{j,t}\|^2}. \tag{3.13}$$

Therefore, Lemma 3.7 holds with the constants $\alpha_{\mathcal{D}} = \frac{(1-\rho_2)\eta_{\mathcal{D}}}{M\beta_{\mathcal{D}}^2}$, $\alpha_{\mathcal{I}} = \frac{c_1(1-\rho_2)}{M\mathrm{K}\varpi^2}$.

The following lemma shows that if the direct search directions satisfy Condition 2.4, the recursive steps satisfy properties that will enable us to prove convergence of our multigrid method.

LEMMA 3.9. *Suppose iteration* $(j,t) \in \mathcal{R}(\ell,k)\backslash\mathcal{D}(\ell,k)$ *and Condition* 2.4 *is satisfied by all direct search directions. Let* p *be the deepest level in* $\mathcal{R}(j,t)$ *such that*

$$g_{p,0} = R_{p+1} g_{p+1,0} = \cdots = R_{p+1} \cdots R_j g_{j,t}. \tag{3.14}$$

Then under Assumption 3.1*, for any iteration* $(q,k) = (q,0)$*, where* $p < q < j$*, and for iteration* $(q,k) = (j,t)$*, we have*

$$\cos(\theta_{q,k}) \geq \delta_{q-p} \quad \text{and} \quad -(d_{q,k})^\top g_{q,k} \geq \eta_{q-p} \|g_{q,k}\|^2, \tag{3.15}$$

where $\theta_{q,k}$ *is the angle between* $d_{q,k}$ *and the steepest descent direction* $-g_{q,k}$*,* $\delta_{q-p} = \frac{\chi}{2}\eta_{q-p}$*, and* $\eta_i = \left(\alpha^* \rho_1 \kappa^2\right)^i \eta_{\mathcal{D}}$*.*

Proof. 1. We will prove (3.15) for $(q,k) = (q,0)$, where $p < q < j$, by induction on q. First, let us consider iteration $(p+1,0)$, which is computed recursively. From inequality (2.7), it follows that

$$-(d_{p+1,0})^\top g_{p+1,0} \geq \psi_{p,0} - \psi_{p,i^*} \geq \psi_{p,0} - \psi_{p,1} \geq -\alpha_{p,0}\rho_1 (d_{p,0})^\top g_{p,0}, \tag{3.16}$$

where the last inequality comes from the Armijo condition (2.12) for iteration $(p,0)$. Since $(p,0)$ is computed directly, $-(d_{p,0})^\top g_{p,0} \geq \eta_{\mathcal{D}} \|g_{p,0}\|_2^2$. From (3.14) and the first condition in (2.9), we obtain

$$\|g_{p,0}\|_2^2 = \|R_{p+1} g_{p+1,0}\|_2^2 \geq \kappa^2 \|g_{p+1,0}\|_2^2. \tag{3.17}$$

Combining all of these facts together, we get

$$-(d_{p+1,0})^\top g_{p+1,0} \geq \alpha^* \rho_1 \kappa^2 \eta_{\mathcal{D}} \|g_{p+1,0}\|_2^2,$$

which proves the second inequality of (3.15) for $q = p+1$. From Lemma 3.4, we obtain $\|g_{p+1,0}\|_2 \geq \frac{\chi}{2}\|d_{p+1,0}\|$, which completes the proof of the first inequality of (3.15).

Now, suppose (3.15) holds for $p < q < j-1$; we prove that (3.15) also holds for $q+1$. Similar to the case $q = p+1$, we have

$$\begin{aligned}-(d_{q+1,0})^\top g_{q+1,0} &\geq \psi_{q,0} - \psi_{q,i^*} \geq \psi_{q,0} - \psi_{q,1} \geq -\rho_1 \alpha_{q,0} (d_{q,0})^\top g_{q,0}\\ &\geq \rho_1 \alpha^* \left(\alpha^* \rho_1 \kappa^2\right)^{q-p} \eta_{\mathcal{D}} \|g_{q,0}\|^2\\ &\geq \rho_1 \alpha^* \left(\alpha^* \rho_1 \kappa^2\right)^{q-p} \eta_{\mathcal{D}} \kappa^2 \|g_{q+1,0}\|^2\\ &= \eta_{q+1-p} \|g_{q+1,0}\|^2,\end{aligned}$$

since relationship (3.17) also holds with p replaced by q. Again using Lemma 3.4, we obtain (3.15).

2. For iteration (j,t), inequality (3.15) holds by simply repeating, in an analogous fashion, the above proof:

$$\begin{aligned}-(d_{j,t})^\top g_{j,t} &\geq \psi_{j-1,0} - \psi_{j-1,i^*} \geq \psi_{j-1,0} - \psi_{j-t,1} \geq -\rho_1 \alpha_{j-1,0} (d_{j-1,0})^\top g_{j-1,0}\\ &\geq \rho_1 \alpha^* \left(\alpha^* \rho_1 \kappa^2\right)^{j-p-1} \eta_{\mathcal{D}} \|g_{j-1,0}\|^2\\ &\geq \eta_{j-p} \|g_{j,t}\|^2. \qquad \square\end{aligned}$$

We can now prove that the minimization sequence generated by Algorithm 1 on the finest level is globally convergent whereas the minimization sequences on all other coarser levels either are globally convergent or stop after at most K steps.

THEOREM 3.10. *Suppose Condition* 2.4 *is satisfied by all direct search directions. Then under Assumption* 3.1 *the iterative sequence* $\{x_{\mathrm{N},k}\}$ *generated by Algorithm* 1 *at the finest level converges to the unique minimizer of* $f_{\mathrm{N}}(x_{\mathrm{N}})$.

Proof. The step size $\alpha_{\mathrm{N},k}$ at the uppermost level is bounded from below by a constant $\alpha^* > 0$ from Lemma 3.7. From the Armijo condition (2.12), we have

$$\psi_{\mathrm{N},k} - \psi_{\mathrm{N},k+1} \geq -\alpha_{\mathrm{N},k} \rho_1 d_{\mathrm{N},k}^\top g_{\mathrm{N},k}.$$

Therefore, since by Assumption 3.1 $\psi(\cdot)$ is bounded from below, $\lim_{k\to\infty} d_{\mathrm{N},k}^\top g_{\mathrm{N},k} = 0$. From Lemma 3.9, we have

$$-d_{\mathrm{N},k}^\top g_{\mathrm{N},k} \geq \sigma \|g_{\mathrm{N},k}\|^2$$

for some constant σ. This shows that

$$\lim_{k\to\infty} \|\nabla f_{\mathrm{N}}(x_{\mathrm{N},k})\| = 0 \tag{3.18}$$

holds, since $\nabla f_{\mathrm{N}}(x_{\mathrm{N},k}) = g_{\mathrm{N},k}$ (recall that $v_{\mathrm{N}} = 0$). The uniqueness of the minimizer follows from the strict convexity of $f_{\mathrm{N}}(x_{\mathrm{N}})$ in Assumption 3.1. $\square$

We now prove R-linear convergence.

THEOREM 3.11. *Suppose Condition* 2.4 *is satisfied by all direct search directions. Assume that the iterative sequence* $\{x_{\mathrm{N},k}\}$ *generated by Algorithm* 1 *at the uppermost level converges to the unique minimizer* x_{N}^* *of* $f_{\mathrm{N}}(x_{\mathrm{N}})$ *and that Assumption* 3.1 *holds. Then the rate of convergence is at least R-linear.*

Proof. Again from Condition 2.4 and Lemma 3.9, we have

$$f_{\mathrm{N}}(x_{\mathrm{N},k+1}) - f_{\mathrm{N}}(x_{\mathrm{N},k}) \leq -\alpha^* \eta_{\mathrm{N}} \|\nabla f_{\mathrm{N}}(x_{\mathrm{N},k})\|^2. \tag{3.19}$$

From the second inequality of (3.8) in Lemma 3.4, we get

$$\|\nabla f_{\mathrm{N}}(x_{\mathrm{N},k})\|^2 \geq \chi \left(f_{\mathrm{N}}(x_{\mathrm{N},k}) - f_{\mathrm{N}}(x_{\mathrm{N}}^*)\right).$$

Hence,

$$f_{\mathrm{N}}(x_{\mathrm{N},k+1}) - f_{\mathrm{N}}(x_{\mathrm{N},k}) \le -\alpha^* \eta_{\mathrm{N}} \chi \left(f_{\mathrm{N}}(x_{\mathrm{N},k}) - f_{\mathrm{N}}(x_{\mathrm{N}}^*) \right),$$

where $0 < \alpha^* \eta_{\mathrm{N}} \chi < 1$ can be verified straightforwardly. By subtracting $f_{\mathrm{N}}(x_{\mathrm{N}}^*)$ from both sides of the above inequality, we have

$$f_{\mathrm{N}}(x_{\mathrm{N},k+1}) - f_{\mathrm{N}}(x_{\mathrm{N}}^*) \le (1 - \alpha^* \eta_{\mathrm{N}} \chi) \left(f_{\mathrm{N}}(x_{\mathrm{N},k}) - f_{\mathrm{N}}(x_{\mathrm{N}}^*) \right).$$

From the first inequality of (3.8) in Lemma 3.4, we obtain that $f_{\mathrm{N}}(x_{\mathrm{N},k}) - f_{\mathrm{N}}(x_{\mathrm{N}}^*) \ge \frac{\chi}{2} \|x_{\mathrm{N},k} - x_{\mathrm{N}}^*\|^2$. Hence,

$$\begin{aligned} \|x_{\mathrm{N},k} - x_{\mathrm{N}}^*\| &\le \sqrt{\frac{2}{\chi}} (f_{\mathrm{N}}(x_{\mathrm{N},k}) - f_{\mathrm{N}}(x_{\mathrm{N}}^*))^{\frac{1}{2}} \\ &\le \sqrt{\frac{2}{\chi}} (1 - \alpha^* \eta_{\mathrm{N}} \chi)^{\frac{1}{2}} (f_{\mathrm{N}}(x_{\mathrm{N},k-1}) - f_{\mathrm{N}}(x_{\mathrm{N}}^*))^{\frac{1}{2}} \\ &\le \sqrt{\frac{2}{\chi}} (1 - \alpha^* \eta_{\mathrm{N}} \chi)^{\frac{k}{2}} (f_{\mathrm{N}}(x_{\mathrm{N},0}) - f_{\mathrm{N}}(x_{\mathrm{N}}^*))^{\frac{1}{2}}. \qquad \square \end{aligned}$$

COROLLARY 3.12. *For any* $\epsilon > 0$, *after at most* $\tau = \frac{\log((f_{\mathrm{N}}(x_{\mathrm{N},0}) - f_{\mathrm{N}}(x_{\mathrm{N}}^*))/\epsilon)}{\log(1/c)}$ *iterations, where* $0 < c = 1 - \frac{\chi \alpha^* \eta_{\mathrm{N}}}{2} < 1$, *we have* $f_{\mathrm{N}}(x_{\mathrm{N},k}) - f_{\mathrm{N}}(x_{\mathrm{N}}^*) \le \epsilon$.

Proof. With the help of inequality (3.19) and from the standard convergence analysis for convex functions [7], we have the result immediately. $\square$

3.2. General nonconvex problems. In this subsection, we prove that Algorithm 1 is globally convergent when applied to general differentiable functions if the search parameter ρ_2 in (2.13) satisfies

$$1 - \rho_1 \le \rho_2 < 1, \tag{3.20}$$

and we replace Assumption 3.1 by the following.

Assumption 3.13.

1. The level set $\mathcal{D}_\ell = \{x_\ell : \psi_\ell(x_\ell) \le \psi_\ell(x_{\ell,0})\}$ is bounded.
2. The objective function ψ_ℓ is continuously differentiable and the gradient $\nabla \psi_\ell$ is Lipschitz continuous; i.e., there exists a constant $L > 0$ such that

$$\|\nabla \psi_\ell(x_\ell) - \nabla \psi_\ell(\tilde{x}_\ell)\| \le L \|x_\ell - \tilde{x}_\ell\| \quad \text{for all } x_\ell, \tilde{x}_\ell \in \mathcal{D}_\ell. \tag{3.21}$$

This assumption implies that there is a positive constant γ such that

$$\|\nabla \psi_\ell(x_\ell)\| \le \gamma \quad \text{for all } x_\ell \in \mathcal{D}_\ell. \tag{3.22}$$

The following lemma shows that the norm of the search direction is uniformly bounded from above.

LEMMA 3.14. *Suppose Assumption* 3.13 *holds and Condition* 2.4 *is satisfied by all direct search directions. Then, for all iterations* $(j, t) \in \mathcal{R}(\ell, k)$, *we have*

$$\|d_{j,t}\| \le \zeta_{j-p} \gamma \beta_{\mathcal{D}}, \tag{3.23}$$

where $\zeta_i = \max((\varpi \alpha_\rho \mathrm{K})^i, 1)$ *and* $p := p(\ell, k)$ *is the deepest level in* $\mathcal{R}(\ell, k)$ *defined by* (3.3). *Therefore,* $\|d_{j,t}\| \le \widetilde{\gamma} := \zeta_{\mathrm{N}-\mathrm{N}_0} \gamma \beta_{\mathcal{D}}$.

Proof. 1. If iteration $(j,t) \in \mathcal{D}(\ell,k)$, we obtain $\|d_{j,t}\| \le \beta_{\mathcal{D}}\|g_{j,t}\| \le \gamma\beta_{\mathcal{D}}$ from Condition 2.4 and (3.22). Hence, the inequality (3.23) is proved since $\zeta_{j-p} \ge 1$.

2. Now consider iteration $(j,t) \in \mathcal{R}(\ell,k)\backslash\mathcal{D}(\ell,k)$. We prove (3.23) by induction on the levels $q = p+1,\dots,j$. Since there is no recursive step on level p in the minimization sequence initialized by iteration $(p+1,t)$ for any t, the total number of iterations at level p is less than K, and the step size $\alpha_{p,k}$ generated by Algorithm 2 is always bounded from above by α_ρ, we obtain

$$\|d_{p+1,t}\| = \left\| P_{p+1}\left(\sum_{k=0}^{i^*-1} \alpha_{p,k} d_{p,k}\right)\right\| \le \varpi\alpha_\rho \sum_{k=0}^{i^*-1} \|d_{p,k}\| \le \varpi\alpha_\rho \mathrm{K}\gamma\beta_{\mathcal{D}} \le \zeta_1\gamma\beta_{\mathcal{D}},$$

which proves (3.23) for $q = p+1$. Suppose (3.23) is true for $q-1 \ge p$. As above, we have

$$\begin{aligned}\|d_{q,t}\| = \left\| P_{q}\left(\sum_{k=0}^{i^*-1} \alpha_{q-1,k} d_{q-1,k}\right)\right\| &\le \varpi\alpha_\rho \sum_{k=0}^{i^*-1} \|d_{q-1,k}\| \le \varpi\alpha_\rho \mathrm{K}\zeta_{q-1-p}\gamma\beta_{\mathcal{D}} \\ &\le \zeta_{q-p}\gamma\beta_{\mathcal{D}},\end{aligned}$$

since $\varpi\alpha_\rho\mathrm{K}\zeta_{q-1-p} \le \zeta_{q-p}$. This completes the proof. □

LEMMA 3.15. *Suppose Assumption* 3.13 *holds. The step size* $\alpha_{\ell,0}$ *on the first iteration of each minimization sequence on level* ℓ *is bounded from below by*

$$\alpha_{\ell,0} > \frac{-(1-\rho_2)g_{\ell,0}^\top d_{\ell,0}}{L\|d_{\ell,0}\|^2}. \tag{3.24}$$

If $d_{\ell,0}$ *is a direct search direction satisfying Condition* 2.4, *then*

$$\alpha_{\ell,0} > \alpha_{\mathcal{D}} = \frac{(1-\rho_2)\eta_{\mathcal{D}}}{L\beta_{\mathcal{D}}^2}. \tag{3.25}$$

Proof. Since condition (2.13) is satisfied at the first iteration,

$$\psi_\ell(x_{\ell,0} + \alpha_{\ell,0}d_{\ell,0}) > \psi_{\ell,0} + \rho_2\alpha_{\ell,0}g_{\ell,0}^\top d_{\ell,0}.$$

From the mean-value theorem, we have

$$\psi_\ell(x_{\ell,0} + \alpha_{\ell,0}d_{\ell,0}) - \psi_{\ell,0} = \alpha_{\ell,0}\nabla\psi_\ell(x_{\ell,0} + \theta d_{\ell,0})^\top d_{\ell,0},$$

where $\theta \in [0, \alpha_{\ell,0}]$. Combining these facts, we obtain $\nabla\psi_\ell(x_{\ell,0} + \theta d_{\ell,0})^\top d_{\ell,0} > \rho_2 g_{\ell,0}^\top d_{\ell,0}$; hence,

$$\nabla\psi_\ell(x_{\ell,0} + \theta d_{\ell,0})^\top d_{\ell,0} - g_{\ell,0}^\top d_{\ell,0} > (\rho_2 - 1)g_{\ell,0}^\top d_{\ell,0}.$$

Since $\nabla\psi_\ell$ is Lipschitz continuous by Assumption 3.13 and $g_{\ell,0} := \nabla\psi_\ell(x_{\ell,0})$, we obtain

$$\theta L\|d_{\ell,0}\|^2 > (\rho_2 - 1)g_{\ell,0}^\top d_{\ell,0}.$$

Therefore, $\alpha_{\ell,0} \ge \frac{(\rho_2-1)g_{\ell,0}^\top d_{\ell,0}}{L\|d_{\ell,0}\|^2}$, proving (3.24). Using the inequalities (2.10) of Condition 2.4 immediately gives (3.25). □

The following lemma shows that the directional derivative along a recursive search direction and the step size are bounded from below by the norm of the gradient raised to some finite power.

LEMMA 3.16. *Suppose iteration* $(j,t) \in \mathcal{R}(\ell,k)\backslash\mathcal{D}(\ell,k)$ *and Condition* 2.4 *is satisfied by all direct search directions and Assumption* 3.13 *holds. Let* p *be the deepest level in* $\mathcal{R}(j,t)$ *such that*

$$g_{p,0} = R_{p+1}g_{p+1,0} = \cdots = R_{p+1}\cdots R_j g_{j,t} \tag{3.26}$$

is satisfied. Then for any iteration $(q,k) = (q,0)$*, where* $p < q < j$*, and for iteration* $(q,k) = (j,t)$*, we have*

$$-d_{q,k}^{\top}g_{q,k} \geq \left(\frac{\rho_1(1-\rho_2)}{\rho_2 L}\right)^{(2^{i-1}-1)} \frac{(\rho_2^{-1}\rho_1\alpha_{\mathcal{D}}\eta_{\mathcal{D}}\kappa^{2(j-p)}\|g_{j,t}\|^2)^{2^{i-1}}}{\widetilde{\gamma}^{2^i-2}}, \tag{3.27}$$

and if $q < \mathrm{N}$,

$$\alpha_{q,k} \geq \left(\frac{\rho_1}{\rho_2}\right)^{(2^{i-1}-1)} \left(\frac{1-\rho_2}{L}\right)^{(2^{i-1})} \frac{(\rho_2^{-1}\rho_1\alpha_{\mathcal{D}}\eta_{\mathcal{D}}\kappa^{2(j-p)}\|g_{j,t}\|^2)^{2^{i-1}}}{\widetilde{\gamma}^{2^i}}, \tag{3.28}$$

and if $q = \mathrm{N}$,

$$\alpha_{q,k} \geq \min\left(\alpha_\rho, \left(\frac{\rho_1(1-\rho_2)}{\rho_2 L}\right)^{(2^{i-1}-1)} \frac{2\tau(1-\rho_1)(\rho_2^{-1}\rho_1\alpha_{\mathcal{D}}\eta_{\mathcal{D}}\kappa^{2(j-p)}\|g_{j,t}\|^2)^{2^{i-1}}}{L\widetilde{\gamma}^{2^i}}\right), \tag{3.29}$$

where $i = q - p$.

Proof. 1. We prove this lemma by induction on the level q. First, let us consider iteration $(q,k) := (p+1,0)$. From inequality (2.14) and Condition 2.4, it follows that

$$\begin{aligned}-d_{p+1,0}^{\top}g_{p+1,0} &\geq \rho_2^{-1}(\psi_{p,0} - \psi_{p,1}) \geq -\rho_2^{-1}\rho_1\alpha_{p,0}d_{p,0}^{\top}g_{p,0}\\ &\geq \rho_2^{-1}\rho_1\alpha_{\mathcal{D}}\eta_{\mathcal{D}}\|g_{p,0}\|^2\\ &\geq \rho_2^{-1}\rho_1\alpha_{\mathcal{D}}\eta_{\mathcal{D}}\kappa^{2(j-p)}\|g_{j,t}\|^2,\end{aligned}$$

which proves inequality (3.27). If $p+1 < \mathrm{N}$, the line search is the modified backtracking procedure. It follows from Lemma 3.15 that the step size $\alpha_{p+1,0}$ is bounded from below:

$$\alpha_{p+1,0} \geq (1-\rho_2)\frac{-d_{p+1,0}^{\top}g_{p+1,0}}{L\|d_{p+1,0}\|^2}.$$

From Lemma 3.14, we obtain

$$\alpha_{p+1,0} \geq (1-\rho_2)\frac{\rho_1\alpha_{\mathcal{D}}\eta_{\mathcal{D}}\kappa^{2(j-p)}\|g_{j,t}\|^2}{\rho_2 L\widetilde{\gamma}^2},$$

which proves inequality (3.28).

2. Now suppose inequalities (3.27) and (3.28) hold for $p < q < j-1$; we prove that they also hold for $q+1$. As in the case of $q = p+1$, we have

$$\begin{aligned}
-d_{q+1,0}^{\top} g_{q+1,0} &\geq \rho_2^{-1}(\psi_{q,0}-\psi_{q,1}) \geq -\rho_2^{-1}\rho_1 \alpha_{q,0} d_{q,0}^{\top} g_{q,0} \\
&\geq \frac{\rho_1}{\rho_2}\left(\frac{\rho_1}{\rho_2}\right)^{(2^{i-1}-1)}\left(\frac{1-\rho_2}{L}\right)^{(2^{i-1})} \frac{(\rho_2^{-1}\rho_1\alpha_{\mathcal{D}}\eta_{\mathcal{D}}\kappa^{2(j-p)}\|g_{j,t}\|^2)^{2^{i-1}}}{\widetilde{\gamma}^{2^i}} \\
&\quad \cdot \left(\frac{\rho_1(1-\rho_2)}{\rho_2 L}\right)^{(2^{i-1}-1)} \frac{(\rho_2^{-1}\rho_1\alpha_{\mathcal{D}}\eta_{\mathcal{D}}\kappa^{2(j-p)}\|g_{j,t}\|^2)^{2^{i-1}}}{\widetilde{\gamma}^{2^i-2}} \\
&= \left(\frac{\rho_1(1-\rho_2)}{\rho_2 L}\right)^{(2^{i}-1)} \frac{(\rho_2^{-1}\rho_1\alpha_{\mathcal{D}}\eta_{\mathcal{D}}\kappa^{2(j-p)}\|g_{j,t}\|^2)^{2^{i}}}{\widetilde{\gamma}^{2^{i+1}-2}}.
\end{aligned}$$

Again using Lemmas 3.14 and 3.15, we obtain inequality (3.28).

3. There are two cases for iteration (j,t). If $j < \mathrm{N}$, then inequalities (3.27) and (3.28) hold by simply repeating, in an analogous fashion, the above proof. If $j = \mathrm{N}$, inequality (3.27) still holds but inequality (3.28) has to be modified since the line search is now the traditional backtracking procedure. It follows from Theorem 3.3 that the step size satisfies

$$\alpha_{j,t} \geq \min\left(\alpha_\rho, \frac{2\tau(\rho_1-1)d_{j,t}^{\top} g_{j,t}}{L\|d_{j,t}\|_2^2}\right),$$

which provides the required result. □

Remark 3.17. The techniques for proving Lemmas 3.9 and 3.16 are similar; i.e., both depend only on the very first iteration of each minimization sequence on the coarser levels. While the step sizes can be bounded from below by a constant in the uniformly convex case, the step size of the first iteration of each minimization sequence on the coarser levels can be bounded from below only by the norm of gradient raised to some finite power in the general case.

We now establish the global convergence of Algorithm 1.

THEOREM 3.18. *Suppose Condition* 2.4 *is satisfied by all direct search directions and Assumption* 3.13 *holds. Then in Algorithm* 1 *at the uppermost level*

$$\lim_{k\to\infty} \|\nabla f_{\mathrm{N}}(x_{\mathrm{N},k})\| = 0.$$

Proof. From the same reasoning as in the proof of Theorem 3.10, we obtain

$$\lim_{k\to\infty} \alpha_{\mathrm{N},k} d_{\mathrm{N},k}^{\top} g_{\mathrm{N},k} = 0 \tag{3.30}$$

from the Armijo condition (2.12) and the boundedness of the function ψ_{N} on the level set $\mathcal{D}_{\mathrm{N}}$. From Condition 2.4 and Lemma 3.16, it follows that each iteration satisfies

$$-\alpha_{\mathrm{N},k} d_{\mathrm{N},k}^{\top} g_{\mathrm{N},k} \geq \sigma \|g_{\mathrm{N},k}\|^i, \tag{3.31}$$

where σ is a positive constant and the order i can be selected only from a finite set of integers $\{2^1, 2^2, \ldots, 2^{\mathrm{N}-\mathrm{N}_0+1}\}$, whether or not the direction $d_{\mathrm{N},k}$ is a direct search direction or a recursive search direction. Combining (3.30) and (3.31), we obtain $\lim_{k\to\infty}\|\nabla f_{\mathrm{N}}(x_{\mathrm{N},k})\| = 0$, since $\nabla f_{\mathrm{N}}(x_{\mathrm{N},k}) = g_{\mathrm{N},k}$ (recall that $v_{\mathrm{N}} = 0$). □

4. Practical issues.

In this section, we discuss different ways to generate direct search directions and how to apply the *full multigrid* method, which is used to enhance the performance of the multigrid method for solving PDEs, to our optimization context.

4.1. The direct search direction.

We first discuss the role that direct search steps play as "smoothers." The role of "smoothing" has been extensively discussed in multigrid methods for PDEs. Basically, smoothing steps smooth the residual on the fine level and the coarse grid correction steps damp the error on the coarse levels. In geometric multigrid methods, which fix the coarsening and adjust the smoother, an error $e_{\ell,k} := x_{\ell,k} - x^*_\ell$ (x^*_ℓ denotes the exact solution on level ℓ) is said to be "smooth" if it can be approximated on some predefined coarser level. In algebraic multigrid methods, which fix the smoother and adjust the coarsening, the error $e_{\ell,k}$ is said to be "smooth" if it is slow to converge with respect to the smoothing operator; i.e., it has to be approximated by means of a coarser level in order to speed up convergence (see section A.3 in [38] for a detailed discussion). A standard proof of convergence of multigrid methods for PDEs requires that the coarse grid correction and the smoothing operator cooperate with each other so that the spectral radius ρ of a certain matrix is strictly less than one. As a consequence, the error $e_{\ell,k}$ is reduced proportionally at the rate ρ, i.e., $\|e_{\ell,k}\| \leq \rho\|e_{\ell,k-1}\|$, for all iterates k. For example, some algebraic multigrid methods require that the smoothing operator S_ℓ produce a sufficient reduction in the error $e_{\ell,k}$; that is, $\|S_\ell e_{\ell,k}\|$ should be sufficiently smaller than $\|e_{\ell,k}\|$ (see the inequality (A.3.7) in [38]).

Our multigrid optimization framework follows the geometric multigrid framework. We refer to direct search steps in our method as "smoothing steps" if they must be taken before or after a recursive step. Such steps act as a kind of "smoothing" operator, although the term "smoothing" is misleading in the context of optimization. A local convergence analysis which includes proving that $\|e_{\ell,k}\| \leq \rho\|e_{\ell,k-1}\|$ for all iterates k sufficiently large enough might be possible for a carefully chosen direct search method under suitable assumptions. However, we focus only on the global convergence of our multigrid method in this paper, and our analysis does not depend on these "extra" smoothing steps. One natural and practical requirement for a "smoothing" step is that they result in a "sufficient" decrease in the objective function value [3, 39]. Since it is hard to provide a theoretical justification for the impact of smoothing steps on the performance of our algorithm on different levels, we instead provide empirical evidence of this impact by depicting the evolution of the objective function value and the norm of the gradient for the problems tested in section 5.

4.1.1. Newton-type step directions. Exact Newton steps as well as inexact Newton steps generated by the linear conjugate gradient (CG) method satisfy Condition 2.4 [39] for strictly convex problems. For nonconvex problems at iteration (ℓ, k), if the Hessian $G_{\ell,k}$ is not positive definite, a modified Newton method [35, 36] can be used to generate a descent direction. In particular, one can add a term E to the Hessian $G_{\ell,k}$ so that $\widetilde{G}_{\ell,k} = G_{\ell,k} + E_{\ell,k} \succ 0$. The additive term $E_{\ell,k}$ can be taken as a diagonal matrix $\mu_{\ell,k} I_\ell$, where $\mu_{\ell,k} > 0$ and I_ℓ is the identity matrix. If the term $E_{\ell,k}$ is chosen large enough so that the smallest eigenvalue of $\widetilde{G}_{\ell,k}$ is uniformly bounded from below and if we assume that the norm $\|\widetilde{G}_{\ell,k}\| \leq M$, i.e., $\widetilde{G}_{\ell,k}$ is uniformly bounded from above, then Condition 2.4 is satisfied. Another way of choosing $E_{\ell,k}$ is to use a modified Cholesky factorization approach [17, 16]. If the condition number number of $\widetilde{G}_{\ell,k}$ is uniformly bounded, i.e.,

$$\|\widetilde{G}_{\ell,k}\| \, \|\widetilde{G}^{-1}_{\ell,k}\| \leq \vartheta, \tag{4.1}$$

then $d_{\ell,k} = -\widetilde{G}^{-1}_{\ell,k} g_{\ell,k}$ is a descent direction and the angle $\theta_{\ell,k}$ between $d_{\ell,k}$ and the steepest descent direction $-g_{\ell,k}$ is bounded away from $\pi/2$ since

$$\cos(\theta_{\ell,k}) = \frac{|g^\top_{\ell,k} d_{\ell,k}|}{\|g_{\ell,k}\|_2 \, \|d_{\ell,k}\|_2} \geq \frac{1}{\vartheta}. \tag{4.2}$$

If (4.2) holds, we can also prove Theorem 3.18 even though Condition 2.4 might not hold. Assuming that Assumption 3.13 holds and $\|\widetilde{G}_{\ell,k}\| \leq M$, from the fact that $\|Ax\| \geq \|x\|/\|A^{-1}\|$ for any invertible matrix A, we obtain

$$-d_{\ell,k}^{\top} g_{\ell,k} = \|\widetilde{G}_{\ell,k}^{-\frac{1}{2}} g_{\ell,k}\|^2 \geq \|g_{\ell,k}\|^2/\|\widetilde{G}_{\ell,k}\| \geq \|g_{\ell,k}\|^2/M. \tag{4.3}$$

If the backtracking line search procedure is used, it follows from Theorem 3.3 that

$$-\alpha_{\ell,k} d_{\ell,k}^{\top} g_{\ell,k} \geq -\min\left(\alpha_\rho, \frac{2\tau(\rho_1 - 1) d_{\ell,k}^{\top} g_{\ell,k}}{L_\ell \|d_{\ell,k}\|_2^2}\right) d_{\ell,k}^{\top} g_{\ell,k}.$$

This, together with (4.2) and (4.3), gives

$$-\alpha_{\ell,k} d_{\ell,k}^{\top} g_{\ell,k} \geq \min\left(\frac{\alpha_\rho}{M}, \frac{2\tau(1-\rho_1)}{L_\ell \vartheta^2}\right) \|g_{\ell,k}\|^2.$$

Therefore, the first step in the proof of Lemma 3.16 can also go through except with different constants. Similar results can be obtained for the Goldstein rule.

4.1.2. Quasi-Newton step directions: L-BFGS.

Using a modified Newton method to compute a direct search direction is not appropriate if the Hessian cannot be computed at a reasonable cost. Quasi-Newton methods are also expensive since the approximation to the Hessian has to be stored explicitly. However, the limited memory BFGS (L-BFGS) method requires only a few vectors to represent this approximation implicitly. These vectors are updated using information from only the most recent iterations no matter whether they are direct search steps or recursive search steps. Therefore, the L-BFGS method can be seamlessly integrated into our multigrid method. For a description of the L-BFGS method based on a recursive formula and a compact representation of inverse BFGS matrices, we refer the reader to [35, 12].

According to the way coarser level problems (2.3) are constructed, all coarser level models on the same level differ only by the additive term $-v_\ell^{\top} x_\ell$. Since there may be more than one minimization sequence on a particular level, information stored for the L-BFGS method from a previous minimization sequence can be used to accelerate the convergence of the current minimization sequence. A minimization sequence ends whenever the method goes to the next finer level, since the additive term $-v_\ell^{\top} x_\ell$ changes when the method returns to the current level. Another strategy is to combine an inexact Newton method with the L-BFGS method. Specifically, the L-BFGS method can be used to build an approximation of the inverse of the Hessian at each iteration, and this approximate inverse Hessian can be provided as a preconditioner to the preconditioned linear conjugate gradient method in an inexact Newton method. Since Newton's method performs well when the iterates are close to an optimal solution, one can alternate between L-BFGS and the inexact Newton method, especially when L-BFGS stagnates. Further discussion of this kind of hybrid strategy can be found in [32].

4.2. Full multigrid method.

The basic multigrid method solves problem (2.1) by calling $x_{\mathrm{N},i^*} = MLS(\mathrm{N}, x_{\mathrm{N},0}, 0)$. Since starting from a good initial point usually reduces the total number of iterations required, the idea underlying the "full multigrid method" is the use of the multilevel approach itself to provide a good initial point. Suppose we start at a level $\ell = \mathrm{N}_0$ where the discretized problem is very easily solved.

Algorithm 1 is applied to the discretized problem at level ℓ to obtain a solution x_{ℓ,i^*}, and we interpolate this solution to the next finer level $\ell+1$ as an initial approximation. This process is repeated over and over until we reach the uppermost level. The detailed algorithm is described in Algorithm 3.

```
ALGORITHM 3 Full Multigrid Method FMLS.
Step 1. Set an initial approximation $x_{\mathrm{N}_0,0}$.
Step 2. FOR $\ell = \mathrm{N}_0, \mathrm{N}_0 + 1, \ldots, \mathrm{N}$
    2.1. Call $x_\ell = MLS(\ell, x_{\ell,0}, 0)$ starting with $x_{\ell,0}$; i.e., apply multigrid Algorithm
         1 to solve the discretized problem $\min_{x_\ell} f_\ell(x_\ell)$ on level $\ell$.
    2.2. If $\ell < \mathrm{N}$, prolongate $x_\ell$ to obtain an initial point $x_{\ell+1,0}$ on level $\ell + 1$.
```

5. Numerical tests. In this section, we demonstrate the effectiveness of our multigrid approach by comparing different versions of it with other methods such as Newton's method, limited memory BFGS, and mesh refinement on two problems. The standard L-BFGS method applied at the finest level without recourse to coarse level computations is denoted by L-BFGS. Similarly, Newton's method on the finest level using Cholesky factorization to solve the system of linear equations is denoted by NT-FACT. The mesh refinement technique where the discretized problems are solved in turn from the coarsest level to the finest level using the standard L-BFGS method is denoted by MR-LBFGS. Algorithm 1 using the steps of the L-BFGS method as the direct search direction is denoted by MLS-LBFGS. The full multigrid algorithm, Algorithm 3, using the L-BFGS method is denoted by FMLS-LBFGS. Similarly, the full multigrid algorithm, Algorithm 3, using Newton's method and the linear CG method or the linear multigrid method to solve the system of linear equations at each Newton step is denoted by FMLS-CG or FMLS-LMG, respectively. The variant FMLS-LMG can be viewed as an extension of the classical nonlinear multigrid method following a global linearization approach. Preliminary computational testing indicated that doing one smoothing step improved performance, although our convergence results do not require that smoothing steps be taken. (When we specify that a particular version of Algorithm 1 or 3 does k smoothing steps, we mean that before considering doing a recursive step, the algorithm first takes k direct search steps.)

For practical considerations, we terminated all algorithms if the iterations stagnated, that is,

$$\frac{\psi_{\ell,k} - \psi_{\ell,k+1}}{\max(|\psi_{\ell,k}|, |\psi_{\ell,k+1}|, 1)} \leq 10^{-14} \quad \text{or} \quad (\|x_{\ell,k} - x_{\ell,k+1}\| < 10^{-9} \text{ and } \ell = \mathrm{N}). \tag{5.1}$$

The initial point in all algorithms was taken to be the zero vector. For the multigrid algorithm, Algorithm 1, we set

$$\begin{aligned} &\kappa = 10^{-1}, \quad \epsilon_\ell = 10^{-5}/5^{\mathrm{N}-\ell}, \quad \epsilon_x = 10^{-1}, \quad \xi = 10^{-16}, \\ &\mathrm{K_d} = 5, \quad \rho_1 = 10^{-3}, \quad \rho_2 = 1 - \rho_1, \end{aligned}$$

and $\mathrm{K} = 10$ on all levels other than the finest. The line search method for choosing a step size was adapted from Algorithm A6.3.1 in [13], which is based on interpolation combined with backtracking. The upper bound on the number of gradient and step difference pairs stored by the L-BFGS method was set to 5. We terminated the linear CG method in FMLS-CG once the norm of the residual was less than $10^{-3}\|g_{\ell,k}\|$.

We used cubic interpolation in Step 2.2 of Algorithm 3. All codes were written in MATLAB (Release 7.3.0), and all experiments were performed on a Dell Precision 670 workstation with an Intel Xeon 3.4GHZ CPU and 6GB of RAM.

5.1. Nonlinear PDE. Consider the variational problem

$$\begin{cases} \min\limits_{u} \ \mathcal{F}(u(x,y)) = \displaystyle\int_{\Omega} \frac{1}{2}|\nabla u(x,y)|^2 - \lambda(u(x,y)e^{u(x,y)} - e^{u(x,y)}) \\ \qquad\qquad\qquad\qquad\qquad - \gamma(x,y)u(x,y)\,dx\,dy \\ \quad \text{such that } \ u = 0 \text{ on } \partial\Omega, \end{cases} \tag{5.2}$$

where $\lambda = 10$, $\Omega = [0,1] \times [0,1]$, and

$$\gamma = \Big(\big(9\pi^2 + \lambda e^{((x^2-x^3)\sin(3\pi y))}\big)\big(x^2 - x^3\big) + 6x - 2\Big)\sin(3\pi y).$$

The variational problem (5.2) corresponds to the nonlinear PDE $-\Delta u + \lambda u e^u = \gamma$ in Ω, given $u = 0$ on $\partial\Omega$ [24]. We discretized Ω at level ℓ as a square grid:

$$\Omega_\ell = \{(i,j) \overset{def}{=} (x_i, y_j) \mid x_i = i\omega_\ell^x,\ y_j = j\omega_\ell^y,\ i = 0,1,\ldots,n_\ell^x;\ j = 0,1,\ldots,n_\ell^y\}, \tag{5.3}$$

where we took $n_\ell^x = n_\ell^y = 2^\ell$ yielding a mesh with widths $\omega_\ell^x = 1/n_\ell^x$ and $\omega_\ell^y = 1/n_\ell^y$, and we discretized the term ∇u in objective functional using the forward finite difference operator δu. Hence, the discretized version of $\mathcal{F}$ in (5.2) was

$$f = \omega_\ell^x \omega_\ell^y \sum_{i,j=0,\ldots,n_\ell^x-1} \frac{1}{2}\|(\delta u)_{i,j}\|^2 - \lambda e^{u_{i,j}}(u_{i,j} - 1) - \gamma_{i,j}u_{i,j}.$$

In our test problems the grid spacing was set to 2^{-3} at the coarsest level $\ell = 3$ and to 2^{-10} at the finest level $\ell = 10$, which gave 9×9 and 1024×1024 grids, respectively. We used the nine-point prolongation

$$\begin{aligned} &(P_\ell u_{\ell-1})(i,j) \\ &= \begin{cases} u_{\ell-1}(\frac{i}{2}, \frac{j}{2}), & i = 0:2:n_\ell^x, j = 0:2:n_\ell^y, \\ \frac{1}{2}u_{\ell-1}(\frac{i}{2}, \frac{j+1}{2} - 1) + \frac{1}{2}u_{\ell-1}(\frac{i}{2}, \frac{j+1}{2}), & i = 0:2:n_\ell^x, j = 1:2:n_\ell^y - 1, \\ \frac{1}{2}u_{\ell-1}(\frac{i+1}{2} - 1, \frac{j}{2}) + \frac{1}{2}u_{\ell-1}(\frac{i+1}{2}, \frac{j}{2}), & i = 1:2:n_\ell^x - 1, j = 0:2:n_\ell^y, \\ \frac{1}{4}u_{\ell-1}(\frac{i+1}{2} - 1, \frac{j+1}{2} - 1) + \frac{1}{4}u_{\ell-1}(\frac{i+1}{2} - 1, \frac{j+1}{2}) & \\ \quad + \frac{1}{4}u_{\ell-1}(\frac{i+1}{2}, \frac{j+1}{2} - 1) + \frac{1}{4}u_{\ell-1}(\frac{i+1}{2}, \frac{j+1}{2}), & i = 1:2:n_\ell^x - 1, j = 1:2:n_\ell^y - 1, \end{cases} \end{aligned}$$

and $R_\ell = \frac{1}{4}P_\ell^\top$ similar to multigrid methods for PDEs [23, 38, 40].

In Tables 5.1–5.2, we summarize the computational costs of the various methods on this and one other problem. We use "ℓ" to indicate the level, and "nfe" and "nge" to denote the total numbers of function and gradient evaluations at that level, respectively. We define a cycle as the iterations between two consecutive recursive steps and denote the total number of cycles on each level by "nv." We also report the total CPU time measured in seconds and the accuracy attained, which is measured by the Euclidean norm of the gradient $\|g^*\|$ at the final iteration.

From Table 5.1, we can see that L-BFGS is not efficient. For example, it terminated after 1018 function evaluations and 1986.93 seconds with $\|g^*\| = 9.6\text{e-}5$ on level 10. For MLS-LBFGS, the numbers of function/gradient evaluations on the finest

TABLE 5.1
Summary of computational costs for problem (5.2).

NT-FACT on level 10					L-BFGS on level 10				
ℓ	nfe	nge	$\|\|g^*\|\|$	CPU	ℓ	nfe	nge	$\|\|g^*\|\|$	CPU
10	6	4	4.3e-08	154.19	10	1018	1001	9.6e-05	1986.93

MR-LBFGS			MLS-LBFGS									FMLS-LBFGS						FMLS-LMG			FMLS-CG		
Level 10			Level 8			Level 9			Level 10			Level 8			Level 10			Level 10			Level 10		
ℓ	nfe	nge	nfe	nge	nv	nfe	nge	nv	nfe	nge	nv	nfe	nge	nv	nfe	nge	nv	nfe	nge	nv	nfe	nge	nv
3	18	17	78	71	0	32	28	0	21	20	0	74	70	0	74	70	0	23	13	0	50	28	0
4	32	29	116	99	17	61	53	9	67	53	6	49	40	7	49	40	7	6	4	1	44	25	3
5	43	40	123	100	17	103	87	12	109	95	15	27	23	3	27	23	3	3	2	0	20	12	2
6	42	40	96	77	11	118	95	18	190	154	23	17	15	1	17	15	1	3	2	0	6	4	1
7	47	46	50	38	7	90	72	11	194	152	27	6	5	1	6	5	1	3	2	0	7	4	0
8	1	1	23	18	4	51	33	7	123	88	17	1	1	0	1	1	0	1	1	0	1	1	0
9	1	1				21	16	3	74	44	8				1	1	0	1	1	0	1	1	0
10	1	1							25	18	4				1	1	0	1	1	0	1	1	0
$\|\|g^*\|\|$	3.1e-06		5.26e-06			9.76e-06			9.68e-06			8.8e-06			3.0e-06			2.9e-06			3.1e-06		
CPU	2.42		4.69			16.19			82.89			0.70			1.61			1.53			2.44		

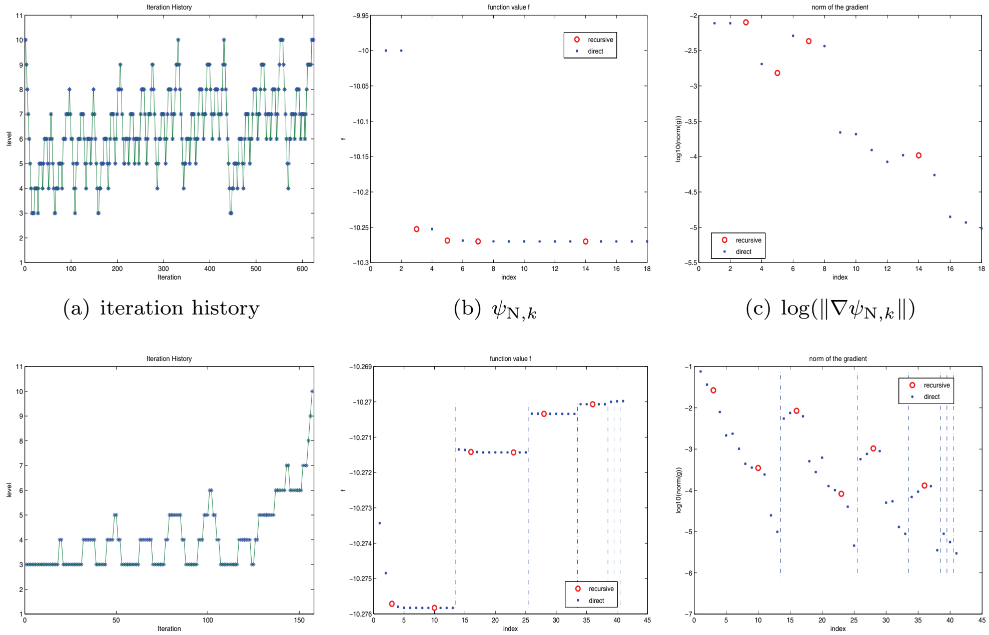
TABLE 5.2
Summary of computational costs for problem (5.4).

L-BFGS on level 10				
ℓ	nfe	nge	$\|\|g^*\|\|$	CPU
10	191	169	3.8e-02	662.79

	MR-LBFGS		MLS-LBFGS									FMLS-LBFGS								
	Level 10		Level 8			Level 9			Level 10			Level 8			Level 9			Level 10		
ℓ	nfe	nge	nfe	nge	nv	nfe	nge	nv	nfe	nge	nv	nfe	nge	nv	nfe	nge	nv	nfe	nge	nv
3	13	12	28	24	0	11	8	0	7	4	0	86	79	0	86	79	0	86	79	0
4	43	37	70	54	9	35	28	3	32	23	2	140	116	16	142	118	16	142	118	16
5	157	152	118	97	13	82	70	9	67	57	9	137	115	17	143	120	18	143	120	18
6	406	391	122	95	16	128	106	16	121	102	16	107	87	14	123	101	16	126	103	16
7	426	417	93	65	10	162	132	22	159	134	20	80	63	10	111	87	14	126	101	15
8	239	228	54	43	6	140	101	17	188	146	22	38	30	4	73	56	9	101	79	14
9	172	168				78	67	10	139	103	17				28	20	3	58	42	7
10	32	29							101	76	11							22	15	2
$\|\|g^*\|\|$	1.3e-02		7.41e-04			1.23e-03			5.00e-04			1.7e-04			7.0e-04			1.1e-03		
CPU	347.77		15.00			94.13			436.26			13.38			38.85			122.03		

level are smaller than those on the coarser levels and they are almost the same when MLS-LBFGS is applied to levels 8, 9, and 10, respectively. However, MLS-LBFGS took a lot of iterations on the coarser levels. One reason is that the recursive steps cannot provide sufficient improvements since the first-order model is not good when the solution is far away from the optimal solution. By using MLS-LBFGS as an approach to obtain a better initial point, the full multigrid algorithm FMLS-LBFGS performed better. The results in Table 5.1 for FMLS-LBFGS applied to level 10 also include all information when FMLS-LBFGS was applied to level 9 since once FMLS-LBFGS reached level 10, it never returned to any coarser level. FMLS-LBFGS required fewer function and gradient evaluations on the finer levels than MR-LBFGS. This is most obvious on level 7, on which FMLS-LBFGS exhibited an approximately 8-fold improvement in terms of the number of function and gradient evaluations over MR-LBFGS. Therefore, FMLS-LBFGS consumed less CPU time than MR-LBFGS, even though FMLS-LBFGS took more iterations on the coarser levels than MR-LBFGS.

To illustrate the multilevel behavior of the MLS-LBFGS method, we plot the level versus iteration history for it in Figure 5.1(a). To see the performance of L-BFGS as a direct search direction and as a smoothing approach, we show the evolutions of the objective function values in Figure 5.1(b) and the norm of the gradients using a base 10 logarithmic scale in Figure 5.1(c). Similar plots for FMLS-LBFGS are



(a) iteration history (b) $\psi_{\mathrm{N},k}$ (c) $\log(\|\nabla\psi_{\mathrm{N},k}\|)$

(d) iteration history (e) $\psi_{\ell,k}$, $\ell = 4, \dots, 10$ (f) $\log(\|\nabla\psi_{\ell,k}\|)$, $\ell = 4, \dots, 10$

FIG. 5.1. *Performance plots for problem* (5.2) *running at the finest level* $\ell = 10$. *The first row corresponds to MLS-LBFGS and the second row corresponds to FMLS-LBFGS. The objective function values are depicted in* (b) *and* (e). *The* log*'s (base* 10*) of the norms of the gradients are depicted in* (c) *and* (f). *Recursive steps are marked by o and direct search steps are marked by* $*$.

depicted in Figures 5.1(d), (e), and (f). In particular, for function values or gradients, we plot all of the sequences from level 4 to level 10 in one figure and differentiate different sequences on different levels by dashed vertical lines.

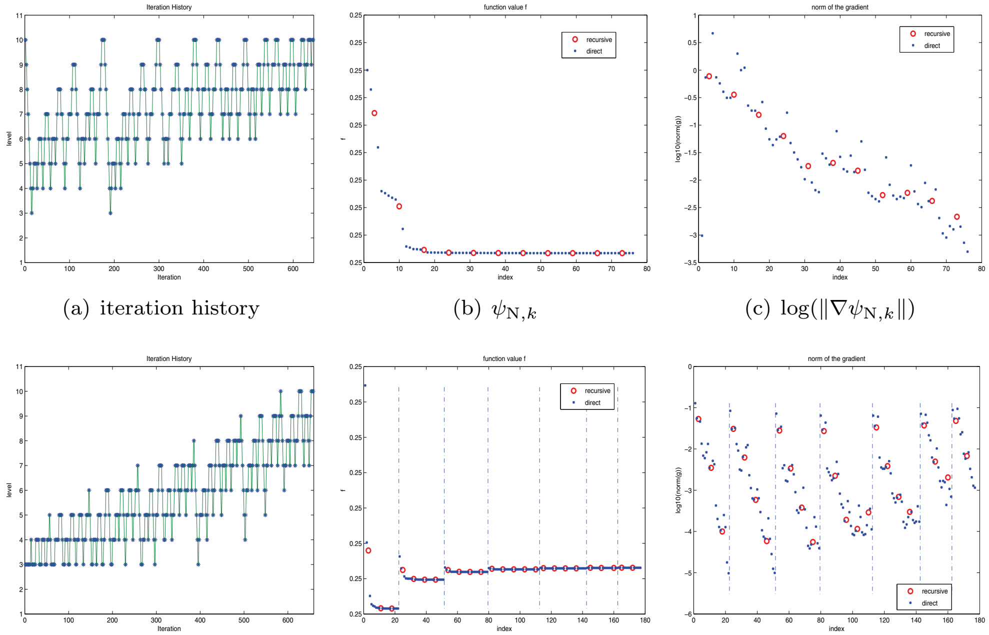
5.2. A nonconvex variational problem. Consider the variational problem

$$(5.4)\qquad \begin{cases} \displaystyle\min_{u,\gamma} \quad \mathcal{F}(u,\gamma) = \int_\Omega \frac{1}{1000}\gamma(x,y)^2 + (u(x,y) - u_0(x,y))^2 \\ \qquad\qquad\qquad\qquad + (\Delta u(x,y) - \gamma(x,y)u(x,y))^2\, dx\, dy \\ \text{such that} \quad u(x,y) = 0 \text{ and } \gamma(x,y) = 0 \text{ on } \partial\Omega, \end{cases}$$

where $\Omega = [0,1] \times [0,1]$, and $u_0(x,y) = \sin(6\pi x)\sin(2\pi y)$. We discretized the domain Ω at level ℓ according to (5.3), and the term Δu was discretized using the standard nine-point finite difference operator $\delta^2 u$. Hence, the discretized version of $\mathcal{F}$ in (5.4) was

$$f = \omega_\ell^x \omega_\ell^y \sum_{i,j=0,\dots,n_\ell^x} \frac{1}{1000}\gamma_{i,j}^2 + (u_{i,j} - (u_0)_{i,j})^2 + \left((\delta^2 u)_{i,j} - \gamma_{i,j}u_{i,j}\right)^2,$$

where $u_{i,j} = 0$ for $(i,j) \notin \Omega_\ell$. The derivatives with respect to $u_{i,j}$ contain terms with coefficient $1/(\omega_\ell^x)^2$ whose magnitude is of order 10^7 when $\ell = 10$ (since $\omega_\ell^x = \omega_\ell^y$). Hence, numerical difficulties can arise since a small change of u and γ can lead to a large change in the gradient of f resulting in very small steps being taken.



(a) iteration history

(b) $\psi_{\mathrm{N},k}$

(c) $\log(\|\nabla\psi_{\mathrm{N},k}\|)$

(d) iteration history

(e) $\psi_{\ell,k}$, $\ell = 4, \ldots, 10$

(f) $\log(\|\nabla\psi_{\ell,k}\|)$, $\ell = 4, \ldots, 10$

Fig. 5.2. *Performance plots for problem* (5.4) *running at the finest level* $\ell = 10$. *The first row corresponds to MLS-LBFGS and the second row corresponds to FMLS-LBFGS. The objective function values are depicted in* (b) *and* (e). *The* log*'s (base* 10*) of the norms of the gradients are depicted in* (c) *and* (f). *Recursive steps are marked by o and direct search steps are marked by* $*$.

Since problem (5.4) is nonconvex, we did not test Newton's method NT-FACT and the multigrid methods FMLS-LMG and FMLS-CG. From Table 5.2, we can see that MR-LBFGS, MLS-LBFGS, and FMLS-LBFGS are faster and more accurate than L-BFGS applied to level 10. The termination rule (5.1) was activated in all of these cases, which partly illustrates the ill-posedness of problem (5.4). FMLS-LBFGS required fewer function and gradient evaluations on the finer levels than MR-LBFGS. This is most obvious on level 9, on which FMLS-LBFGS exhibited an approximately 6-fold improvement in terms of the number of function and gradient evaluations over MR-LBFGS. Therefore, FMLS-LBFGS consumed less CPU time than MR-LBFGS. Finally, the level versus iteration history, the evolution of the objective function values, and the evolution of the norms of the gradients in a base 10 logarithmic scale for MLS-LBFGS and FMLS-LBFGS are depicted in Figure 5.2.

Remark 5.1. A proper discretization scheme is critical for robustness and efficiency in solving infinite-dimensional minimization problems. For example, while the analytical solution of a one-dimensional convection-diffusion equation is smooth, numerical difficulties can arise since the central finite difference of this equation can lead to a highly oscillating discretized solution (section 7.1 in [38]). Suppose that the minima $\{x_\ell\}$ of the discretizations of problem (1.1) converge. A discretization scheme might not be suitable if the restriction of x_ℓ is not a good approximation of $x_{\ell-1}$ or the prolongation of $x_{\ell-1}$ is not close to x_ℓ, in particular, on a fine level ℓ.

Remark 5.2. It is well known that L-BFGS can converge slowly on highly ill-conditioned problems (section 9.1 in [35]). The computational results for problems

(5.2) and (5.4) show that the performance of L-BFGS is improved when it is incorporated within our multigrid framework. Figures 5.1 and 5.2 show that the gradient norm is not monotone decreasing. However, the gradient norm is usually substantially reduced within a few steps after a large increase. Further discussion on the behavior of the gradient norm in the steepest descent method and L-BFGS can be found in [34].

6. Discussion. In this paper, we present a new line search multigrid algorithm for general nonconvex unconstrained problems. The algorithm takes as many recursive steps as possible to accelerate the overall computational speed. By imposing a new condition on a modified backtracking line search procedure, the recursive step is guaranteed to be a descent direction. Our multigrid algorithm has been implemented using the limited memory BFGS method to compute direct search directions. Although this method has not yet been shown to converge in theory, it exhibits excellent computational efficiency. Our future work includes developing a direct search direction that is able to utilize historical information more effectively and extending our algorithmic framework to optimization problems with constraints.

Acknowledgments. We want to thank Philippe L. Toint for his comments on the manuscript and for stimulating discussions on multigrid optimization. The authors are grateful to two anonymous referees for their detailed and valuable comments and suggestions.

REFERENCES

[1] U. M. Ascher and E. Haber, *A multigrid method for distributed parameter estimation problems*, Electron. Trans. Numer. Anal., 15 (2003), pp. 1–17.

[2] M. Benzi, E. Haber, and L. Hanson, *Multilevel Algorithms for Large-Scale Interior Point Methods in Bound Constrained Optimization*, Technical Rep., Department of Mathematics and Computer Science, Emory University, Atlanta, GA, 2006.

[3] A. Borzì, *Multilevel Methods in Optimization with Partial Differential Equations*, Lecture notes, Insitut für Mathematik und Wissenschaftliches Rechnen, Karl-Franzens-Universität Graz, Graz, Austria.

[4] A. Borzì, *On the convergence of the MG/OPT method*, PAMM, 5 (2005), pp. 735–736.

[5] A. Borzì and K. Kunisch, *A multigrid scheme for elliptic constrained optimal control problems*, Comput. Optim. Appl., 31 (2005), pp. 309–333.

[6] A. Borzì and K. Kunisch, *A globalization strategy for the multigrid solution of elliptic optimal control problems*, Optim. Methods Softw., 21 (2006), pp. 445–459.

[7] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, UK, 2004.

[8] J. H. Bramble, *Multigrid Methods*, Pitman Res. Notes Math. Ser. 294, Longman Scientific & Technical, Harlow, UK, 1993.

[9] A. Brandt, *Multi-level adaptive solutions to boundary-value problems*, Math. Comp., 31 (1977), pp. 333–390.

[10] A. Brandt, *Multigrid Techniques:* 1984 *Guide with Applications to Fluid Dynamics*, GMD-Stud. 85, Gesellschaft für Mathematik und Datenverarbeitung mbH, St. Augustin, Germany, 1984.

[11] W. L. Briggs, V. E. Henson, and S. F. McCormick, *A Multigrid Tutorial*, 2nd ed., SIAM, Philadelphia, 2000.

[12] R. H. Byrd, J. Nocedal, and R. B. Schnabel, *Representations of quasi-Newton matrices and their use in limited memory methods*, Math. Programming, 63 (1994), pp. 129–156.

[13] J. E. Dennis, Jr., and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice Hall Ser. Comput. Math., Prentice–Hall, Englewood Cliffs, NJ, 1983.

[14] E. D. Dolan, J. J. Moré, and T. S. Munson, *Benchmarking Optimization Software with Cops* 3.0, Technical Rep., Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, France, 2004.

[15] T. Dreyer, B. Maar, and V. Schulz, *Multigrid optimization in applications*, J. Comput. Appl. Math., 120 (2000), pp. 67–84.

[16] A. Forsgren, P. E. Gill, and W. Murray, *Computing modified Newton directions using a partial Cholesky factorization*, SIAM J. Sci. Comput., 16 (1994), pp. 139–150.

[17] P. E. Gill and W. Murray, *Newton-type methods for unconstrained and linearly constrained optimization*, Math. Programming, 7 (1974), pp. 311–350.

[18] S. Gratton, M. Mouffe, A. Sartenaer, Ph. L. Toint, and D. Tomanos, *Numerical experience with a recursive trust-region method for multilevel nonlinear optimization*, Optim. Methods Softw., to appear.

[19] S. Gratton, A. Sartenaer, and P. Toint, *Second-order convergence properties of trust-region methods using incomplete curvature information, with an application to multigrid optimization*, J. Comput. Math., 24 (2006), pp. 676–692.

[20] S. Gratton, A. Sartenaer, and P. L. Toint, *Recursive trust-region methods for multiscale nonlinear optimization*, SIAM J. Optim., 19 (2008), pp. 414–444.

[21] S. Gratton and Ph. L. Toint, *Approximate invariant subspaces and quasi-Newton optimization methods*, Optim. Methods Softw., to appear.

[22] C. Gross and R. Krause, *On the convergence of recursive trust-region methods for multiscale nonlinear optimization and applications to nonlinear mechanics*, SIAM J. Numer. Anal., 47 (2009), pp. 3044–3069.

[23] W. Hackbusch, *Multigrid Methods and Applications*, Springer Ser. Comput. Math. 4, Springer-Verlag, Berlin, 1985.

[24] V. E. Henson, *Multigrid methods nonlinear problems: An overview*, in Computational Imaging, Proc. SPIE 5016, C. A. Bouman and R. L. Stevenson, eds., SPIE, Bellingham, WA, 2003, pp. 36–48.

[25] R. M. Lewis and S. G. Nash, *Model problems for the multigrid optimization of systems governed by differential equations*, SIAM J. Sci. Comput., 26 (2005), pp. 1811–1837.

[26] R. M. Lewis and S. G. Nash, *Factors affecting the performance of optimization-based multigrid methods*, in Multiscale Optimization Methods and Applications, Nonconvex Optim. Appl. 82, Springer, New York, 2006, pp. 151–172.

[27] T. A. Manteuffel, S. F. McCormick, and O. Röhrle, *Projection multilevel methods for quasilinear elliptic partial differential equations: Theoretical results*, SIAM J. Numer. Anal., 44 (2006), pp. 139–152.

[28] T. A. Manteuffel, S. F. McCormick, O. Röhrle, and J. Ruge, *Projection multilevel methods for quasilinear elliptic partial differential equations: Numerical results*, SIAM J. Numer. Anal., 44 (2006), pp. 120–138.

[29] S. F. McCormick, ed., *Multigrid Methods*, Frontiers Appl. Math. 3, SIAM, Philadelphia, 1987.

[30] S. F. McCormick, *Projection multilevel methods for quasi-linear PDEs: V-cycle theory*, Multiscale Model. Simul., 4 (2006), pp. 1339–1348.

[31] H. D. Mittelmann, *Decision Tree for Optimization Software*, http://plato.asu.edu/guide.html.

[32] J. L. Morales and J. Nocedal, *Enriched methods for large-scale unconstrained optimization*, Comput. Optim. Appl., 21 (2002), pp. 143–154.

[33] S. G. Nash, *A multigrid approach to discretized optimization problems*, Optim. Methods Softw., 14 (2000), pp. 99–116.

[34] J. Nocedal, A. Sartenaer, and C. Zhu, *On the behavior of the gradient norm in the steepest descent method*, Comput. Optim. Appl., 22 (2002), pp. 5–35.

[35] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed., Springer Ser. Oper. Res. Financ. Eng., Springer, New York, 2006.

[36] W. Sun and Y.-X. Yuan, *Optimization Theory and Methods: Nonlinear Programming*, Springer Optim. Appl. 1, Springer, New York, 2006.

[37] X.-C. Tai and J. Xu, *Global and uniform convergence of subspace correction methods for some convex optimization problems*, Math. Comp., 71 (2002), pp. 105–124.

[38] U. Trottenberg, C. W. Oosterlee, and A. Schüller, *Multigrid*, Academic Press, San Diego, CA, 2001.

[39] Z. Wen and D. Goldfarb, *A Line Search Multigrid Method for Large-Scale Convex Optimization*, Technical Rep., Department of IEOR, Columbia University, New York, NY, 2007.

[40] P. Wesseling, *An Introduction to Multigrid Methods*, Pure Appl. Math. (N. Y.), John Wiley & Sons, Chichester, 1992.

[41] J. Xu, *An introduction to multilevel methods*, in Wavelets, Multilevel Methods and Elliptic PDEs (Leicester, 1996), Numer. Math. Sci. Comput., Oxford University Press, New York, 1997, pp. 213–302.

[42] I. Yavneh and G. Dardyk, *A multilevel nonlinear method*, SIAM J. Sci. Comput., 28 (2006), pp. 24–46.