CORC Report 2007-07

A LINE SEARCH MULTIGRID METHOD FOR LARGE-SCALE CONVEX OPTIMIZATION [*]

ZAIWEN WEN [†] AND DONALD GOLDFARB [†]

July 3, 2007

Abstract. We present a line search multigrid method based on Nash's MG/OPT multilevel optimization approach for solving discretized versions of convex infinite dimensional optimization problems. Global convergence is proved under fairly minimal requirements on the minimization method used at all grid levels. In particular, our convergence proof does not require that these minimization, or so-called "smoothing" steps, which we interpret in the context of optimization, be taken at each grid level in contrast with multigrid algorithms for PDEs, which fail to converge without such steps. Preliminary numerical experiments show that our method is promising.

Key words. convex optimization, multigrid method, line search, global convergence

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

1. Introduction. Infinite dimensional optimization problems are a major source of large-scale finite dimensional optimization problems [13, 27]. Since it is not possible or very hard to obtain explicit solutions for these problems, they are usually solved numerically either by a "discretize-then-optimize" strategy or an "optimize-then-discretize" strategy. In this paper, we follow the first strategy and consider a class of problems whose discretized versions have the form:

$$(1.1) \qquad \min_{x_h \in \Omega_h} \quad f_h(x_h)$$

where h is an index used to specify the resolution or discretization of the optimization problem, x_h is a vector of dimension n_h and f_h is a real valued and twice continuously differentiable convex function on a domain $\Omega_h \in \mathbb{R}^{n_h}$.

Multigrid methods [9, 11, 12, 19, 25, 34, 35, 36] are iterative methods that were originally proposed for 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 for nonlinear PDEs have been well studied. One is the global linearization method [20, 34], which 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. The second is the local linearization method, such as the full approximation scheme (FAS) [10] and the closely related nonlinear multigrid method (NMGM) [19], in which the multigrid methodology is directly applied to the original system of nonlinear equations and its corresponding system of nonlinear residual equations. A combination of global and linearization is studied in [37] and a projection multilevel method is proposed for quasilinear elliptic PDEs in [23, 24, 26], where the system of nonlinear equations is reformulated as a least-squares problem.

Report Documentation Page		*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE **03 JUL 2007**	2. REPORT TYPE	3. DATES COVERED **00-00-2007 to 00-00-2007**
4. TITLE AND SUBTITLE **A Line Search Multigrid Method for Large-Scale Convex Optimization**		5a. CONTRACT NUMBER
		5b. GRANT NUMBER
		5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER
		5e. TASK NUMBER
		5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Columbia University ,Department of Computer Science,New York,NY,10027**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S)
		11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

We present a line search multigrid method based on Nash's MG/OPT multilevel optimization approach for solving discretized versions of convex infinite dimensional optimization problems. Global convergence is proved under fairly minimal requirements on the minimization method used at all grid levels. In particular, our convergence proof does not require that these minimization, or so-called "smoothing" steps, which we interpret in the context of optimization, be taken at each grid level in contrast with multigrid algorithms for PDEs, which fail to converge without such steps. Preliminary numerical experiments show that our method is promising.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**29**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

Multigrid methods for infinite dimensional optimization problems have also received considerable attention [1, 3, 4, 6, 7, 14, 33]. However, until recently the essential thrust of these methods was based on employing multigrid methods for solving the nonlinear equations derived from the optimality condition of problem (1.1). In a new approach, Nash [21] (see also [22, 29]) proposed a multigrid optimization framework for solving problem (1.1), where $f_h(x_h)$ is a convex function of x_h. A proof of the global convergence of Nash's method was given in [5]. This proof requires that at least one iteration of the optimization algorithm that is used at each level be performed either before going to a coarser level or after returning from a coarser level during a multigrid cycle. These iterations of the optimization algorithm are similar to prior smoothing or post smoothing steps in multigrid methods for PDEs. Expanding on Nash's approach, Gratton, Sartenaer and Toint [18, 16, 17] proposed a recursive trust region method that converges to a first-order optimal point without doing such smoothing steps at each multigrid cycle.

In this paper, we propose a line search multigrid optimization method that adopts some of the features of both Nash's method and the method of Gratton, Sartenaer and Toint. We show that the search direction generated by our multigrid approach is always a descent direction when the objective function $f_h(x_h)$ is convex. We interpret smoothing steps as steps in an optimization algorithm, and prove that our line search method does not require such steps at each multigrid cycle to guarantee global convergence in the convex case. We aslo prove that the convergence rate is at least R-linear. Smoothing steps can be Newton steps and we show that convergence is still guaranteed without solving the Newton systems exactly. If each Newton system is solved by the linear multigrid method, our algorithm can be viewed as a combination of the global linearization method and the FAS scheme for nonlinear PDEs.

This paper is organized as follows. In section 2, we briefly review multigrid methods for PDEs. In section 3, a multigrid method for solving unconstrained convex problems is developed. A proof of global convergence as well as a proof of R-linear convergence for uniformly convex problems are presented in section 4.1. Global convergence for general convex function is proved in section 4.2. In section 5, we discuss some techniques to enhance our mulitigrid method, including the full multigrid method, different ways to generate search directions, and other ways to do smoothing steps. Finally preliminary numerical results are given in Section 6.

We adopt the following notation in this paper: $f_{h,k} \equiv f_h(x_{h,k}), \nabla f_{h,k} \equiv \nabla f_h(x_{h,k})$. Here $x_{h,k}$ is a vector where the first subscript h 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_h, the subscript h either refers to the level of the multigrid, and thus x_h itself is a vector or it refers the fact that x_h is the hth component of the vector x. When it is not clear from the context, we will point out the specific meaning. We use letter H to denote the next coarsest level $h - 1$ from level h. N is reserved for the index of the finest level and N_0 for the coarsest level.

2. Multigrid Methods for PDEs. Consider solving the system of linear equations

(2.1)
$$A_h x_h = b_h,$$

where A_h is a symmetric positive definite matrix and h is the discretization level. Let B_h be an approximate inverse of A_h. Define R_h to be the restriction operator from level h to level H and P_h be the prolongation operator from level H to level h. As in standard multigrid methods, we assume that:

ASSUMPTION 2.1. *The prolongation operator P_h and the restriction operator R_h satisfy:*

$$(2.2) \qquad \sigma_h P_h = R_h^\top.$$

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

ASSUMPTION 2.2. *The coarser level matrix A_{h-1} relates to the finer level matrix A_h through $A_{h-1} = R_h A_h P_h$.*

Given an approximate solution $x_{h,k}$, a multigrid cycle [19, 35] for solving problem 2.1 can be stated as follows:

ALGORITHM 1. *Multigrid-cycle: $x_{h,k+1} = MGCYCLE(h, A_h, b_h, x_{h,k})$*

-PRE-SMOOTHING: Compute $\bar{x}_{h,k} = x_{h,k} + B_h(b_h - A_h x_{h,k})$.

-COARSE GRID CORRECTION:

    Compute the residual $\bar{r}_{h,k} = b_h - A_h \bar{x}_{h,k}$.

    Restrict the residual $\bar{r}_{h-1,k} = R_h \bar{r}_{h,k}$.

    -Solve the Coarse Grid Residual Equation $A_{h-1} e_{h-1,k} = \bar{r}_{h-1,k}$

    IF $h-1 = N_0$, solve $e_{h-1,k} = A_{h-1}^{-1} \bar{r}_{h-1,k}$,

    ELSE call $e_{h-1,k} = MGCYCLE(h-1, A_{h-1}, \bar{r}_{h-1,k}, 0)$.

    Interpolate the correction: $e_{h,k} = P_h e_{h-1,k}$.

    Compute the new approximation solution: $x_{h,k+1} = \bar{x}_{h,k} + e_{h,k}$.

As a result, we have the following iterative algorithm:

ALGORITHM 2. *Multigrid Algorithm $MG(A_h, b_h, \epsilon)$*

INITIALIZATION: LET $x_{h,0}$ AND ϵ BE GIVEN.

FOR $k = 0, 1, 2, \cdots$ UNTIL $\|A_h x_{h,k} - b_h\| \leq \epsilon$ DO

    Call $x_{h,k+1} = MGCYCLE(h, A_h, b_h, x_{h,k})$.

Since understanding the two-grid version of Algorithm 2 is sufficient for understanding the general algorithm, we only consider the two-grid algorithm here. From Algorithm 1, with $h - 1 = H$, we have

$$x_{h,k+1} = \bar{x}_{h,k} + P_h e_{H,k} = \bar{x}_{h,k} + P_h A_H^{-1} R_h \bar{r}_{h,k},$$

where $\bar{r}_{h,k} = b_h - A_h \bar{x}_{h,k}$ is the residual at the point $\bar{x}_{h,k}$. Note that the step $e_{H,k}$ involves A_H^{-1}. This is true for the two-grid version, where level H is the coarsest level. Assume that x_h^* is the solution of (2.1). Then at $\bar{x}_{h,k}$, the error is $\bar{x}_{h,k} - x_h^* = S_1(x_{h,k} - x_h^*)$ and the residual is $\bar{r}_{h,k} = S_2 r_{h,k}$, where $r_{h,k} = b_h - A_h x_{h,k}$, $S_1 = I_h - B_h A_h$ and $S_2 = I_h - A_h B_h$. Hence, the error at the new point $x_{h,k+1}$ is:

$$x_{h,k+1} - x_h^* = (I - P_h A_H^{-1} R_h A_h) S_1 (x_{h,k} - x_h^*),$$

and the residual at $x_{h,k+1}$ is

$$r_{h,k+1} = b_h - A_h x_{h,k+1} = (I - A_h P_h A_H^{-1} R_h) S_2 r_{h,k}.$$

Therefore, the two-grid multigrid algorithm converges uniformly if the spectral radius $\rho((I - A_h P_h A_H^{-1} R_h) S_2) < 1$. Note that the coarse grid correction alone will not lead to convergence, since usually $\rho(I - A_h P_h A_H^{-1} R_h) > 1$.

The smoothing steps of the two-grid multigrid algorithm smooth the residual $r_{h,k}$ on the fine level h and the coarse grid correction steps damp the error on the coarse level H. Different choices of B_h result in different iterative methods. Specifically, if

we decompose A_h as $A_h = D_h - L_h - U_h$, where D_h is the diagonal of A_h and $-L_h$ and $-U_h$ are the lower and upper triangular parts of A_h, respectively, then common choices for B_h are:

$$B_h = \begin{cases} D_h^{-1}, & Jacobi; \\ \omega D_h^{-1}, & \text{Damped Jacobi } (0 < \omega < 2/\rho(D_h^{-1}A_h)); \\ (D_h - L_h)^{-1}, & Gauss-Seidel; \\ \omega(D_h - L_h)^{-1}, & \text{SOR } (0 < \omega < 2). \end{cases}$$

Let us now study the idea of multigrid from the point view of optimization. Solving the system of linear equations (2.1) is equivalent to solving the strictly convex quadratic minimization problem:

$$(2.3) \qquad \min_{x_h} f_h(x_h) = \frac{1}{2}x_h^\top A_h x_h - b_h^\top x_h.$$

The reduction in the value of the objective function obtained by moving from $x_{h,k}$ to $x_{h,k+1}$ is

$$(2.4) \qquad f_h(x_{h,k}) - f_h(x_{h,k+1}) = [f_h(x_{h,k}) - f_h(\bar{x}_{h,k})] + [f_h(\bar{x}_{h,k}) - f_h(x_{h,k+1})],$$

where $\bar{x}_{h,k}$, the outcome of the presmoothing step is

$$\bar{x}_{h,k} = x_{h,k} + p_{h,k}, \quad p_{h,k} = -B_h \nabla f_{h,k},$$

and $\nabla f_{h,k}$ is the gradient of f_h at $x_{h,k}$. The reduction of the objective function value between $x_{h,k}$ and $\bar{x}_{h,k}$ is

$$(2.5) \quad f_h(x_{h,k}) - f_h(\bar{x}_{h,k}) = -(\frac{1}{2}p_{h,k}^\top A_h p_{h,k} + p_{h,k}^\top \nabla f_{h,k})$$

$$= -\frac{1}{2}(\nabla f_{h,k})^\top B_h^\top A_h B_h \nabla f_{h,k} + (\nabla f_{h,k})^\top B_h^\top \nabla f_{h,k}$$

$$= (\nabla f_{h,k})^\top B_h^\top (B_h^{-1} - \frac{1}{2}A_h)B_h \nabla f_{h,k}.$$

The reduction of the objective function values between $\bar{x}_{h,k}$ and $x_{h,k+1}$ is

$$(2.6) \qquad f_h(\bar{x}_{h,k}) - f_h(x_{h,k+1}) = -(\frac{1}{2}e_{h,k}^\top A_h e_{h,k} + e_{h,k}^\top \nabla \bar{f}_{h,k})$$

$$= -\frac{1}{2}(\nabla \bar{f}_{h,k})^\top P_h A_H^{-1} \underbrace{R_h A_h P_h}_{A_H} A_H^{-1} R_h \nabla \bar{f}_{h,k} + (\nabla \bar{f}_{h,k})^\top P_h A_H^{-1} R_h \nabla \bar{f}_{h,k}$$

$$= \frac{1}{2}(\nabla f_{h,k})^\top S_2^\top P_h A_H^{-1} R_h S_2 \nabla f_{h,k}$$

since $\nabla \bar{f}_{h,k} = \nabla f(\bar{x}_{h,k}) = S_2 \nabla f_{h,k}$. Combining (2.4), (2.5) and (2.6), we have that

$$(2.7) \qquad f_h(x_{h,k}) - f_h(x_{h,k+1}) = \frac{1}{2}(\nabla f_{h,k})^\top S_2^\top P_h A_H^{-1} R_h S_2 \nabla f_{h,k}$$

$$+ (\nabla f_{h,k})^\top B_h^\top (B_h^{-1} - \frac{1}{2}A_h)B_h \nabla f_{h,k}.$$

If B_h is the Gauss-Seidel operator, we have:

$$B_h^{-1} - \frac{1}{2}A_h = D_h - L_h - \frac{1}{2}A_h = \frac{1}{2}D_h - \frac{1}{2}(L_h - U_h),$$

where $u^\top(L_h - U_h)u = 0$ for $u \in \mathbb{R}^{n_h}$, as $L_h - U_h$ is antisymmetric. Hence,

$$(\nabla f_{h,k})^\top B_h^\top (B_h^{-1} - \frac{1}{2}A_h)B_h\nabla f_{h,k}$$

$$= \frac{1}{2}(\nabla f_{h,k})^\top B_h^\top D_h B_h \nabla f_{h,k} \geq \frac{1}{2}\lambda_{\min}(B_h^\top D_h B_h)\|\nabla f_{h,k}\|^2 > 0.$$

Therefore the reduction of the objective function value on each multigrid cycle is at least

$$(2.8) \qquad f_h(x_{h,k}) - f_h(x_{h,k+1}) \geq \beta_h\|\nabla f_{h,k}\|^2,$$

where the constant $\beta_h = \frac{1}{2}\lambda_{\min}(B_h^\top D_h B_h) > 0$ follows from the positive definiteness of A_h. Summing (2.8) over k from 0 to j, we have:

$$f_h(x_{h,0}) - f_h(x_{h,j+1}) \geq \sum_{k=0}^{j} \beta_h\|\nabla f_{h,k}\|^2.$$

Then taking the limit as j goes to $+\infty$, we obtain

$$\lim_{k\to+\infty} \nabla f_{h,k} = 0,$$

as $f_h(x_h)$ is bounded below.

Consider now the two-grid version of Algorithm 2 without pre-smoothing steps. Suppose we start from a point $x_{h,k}$ that satisfies:

$$(2.9) \qquad \|R_h\nabla f_{h,k}\| \geq \kappa\|\nabla f_{h,k}\|,$$

where κ is a constant. Then the gradient at $x_{h,k+1}$ is

$$\nabla f_{h,k+1} = (I - A_h P_h A_H^{-1} R_h)\nabla f_{h,k}.$$

Now using the fact that $A_H = R_h A_h P_h$,

$$R_h\nabla f_{h,k+1} = R_h\nabla f_{h,k} - R_h A_h P_h A_H^{-1} R_h\nabla f_{h,k} = 0,$$

which means that the next coarse grid correction $e_{H,k+1} = 0$. Therefore, the two grid algorithm can no longer make any progress at the point $x_{h,k+1}$ by taking steps in the coarse grid.

If we do a line search along the direction $e_{h,k} = -P_h A_H^{-1} R_h\nabla f_{h,k}$ to find the best point along that direction, i.e., we solve

$$\min_{\alpha\in\mathbb{R}} f_h(x_{h,k} + \alpha e_{h,k}),$$

we obtain

$$\alpha = \frac{-e_{h,k}^\top \nabla f_{h,k}}{e_{h,k}^\top A_h e_{h,k}} = \frac{(\nabla f_{h,k})^\top P_h A_H^{-1} R_h\nabla f_{h,k}}{(\nabla f_{h,k})^\top P_h A_H^{-1} \underbrace{R_h A_h P_h}_{A_H} A_H^{-1} R_h\nabla f_{h,k}} = 1.$$

Hence, using a step length of one along $e_{h,k}$ is optimal[1]. Moreover, we have from (2.6) and (2.9) that

$$f_h(x_{h,k}) - f_h(x_{h,k+1}) = \frac{1}{2}(\nabla f_{h,k})^\top P_h A_H^{-1} R_h\nabla f_{h,k} \geq \beta_{h,k}\|\nabla f_{h,k}\|^2,$$

---

[1]For the use of steplength optimization in linear multigrid methods see [31].

where $\beta_{h,k} = \frac{1}{2}\kappa^2/\lambda_{\max}(A_H)$. Although matrix $P_h A_H^{-1} R_h$ is not of full rank, the reduction of the objective function value is still bounded below by the square of the norm of the gradient multiplied by the positive constant $\beta_{h,k}$. Hence, the recursive step $e_{h,k}$ is a good step.

The two-grid algorithm can avoid a break down if it does pre-smoothing steps until a point $\bar{x}_{h,k}$ is generated that satisfies

$$(2.10) \qquad \|R_h \nabla \bar{f}_{h,k}\| \geq \kappa \|\nabla \bar{f}_{h,k}\|, \text{ and } f_h(\bar{x}_{h,k}) < f_h(x_{h,k}).$$

Then a recursive step can be taken and the sequence $\{x_{h,k}\}$ will converge to the optimal solution. The two-grid algorithm with traditional smoothing steps guarantees (2.10) and the smoothing steps also ensure that the norm of the gradient decreases, i.e., $\|\nabla f_{h,k+1}\| \leq \nu_h \|\nabla f_{h,k}\|$ for some constant $0 < \nu_h < 1$.

Generating the coarse grid correction step $e_{H,k}$ can also be viewed as solving a coarser level minimization problem that is closely related to the finer level problem (2.3). From the coarse grid residual equation, we have

$$A_H e_{H,k} = r_{H,k} = -R_h \nabla f_{h,k}$$
$$\Longleftrightarrow A_H(x_{H,k} + e_{H,k}) - b_H = A_H x_{H,k} - b_H - R_h \nabla f_{h,k}$$
$$\Longleftrightarrow A_H(x_{H,k} + e_{H,k}) = b_H + (\nabla f_{H,k} - R_h \nabla f_{h,k}).$$

(The above construction for the coarse grid residual equation is also what is done in the FAS [19, 34]). Hence $e_{H,k}$ is identical to $e_{H,k} = x_H^* - x_{H,k}$, where x_H^* is a minimizer of the problem

$$(2.11) \qquad \min_{x_H} \{\psi_H(x_H) \equiv f_H(x_H) - (v_H)^\top x_H\}$$

and $v_H = \nabla f_{H,k} - R_h \nabla f_{h,k}$. This interpretation provides a motivation for extending the multigrid Algorithm 2 to an algorithm for minimizing a general convex function.

3. A Multigrid Method for Unconstrained Convex Optimization. In this section, we develop a multigrid method for the uppermost finest level problem

$$(3.1) \qquad \min_{x_N} f_N(x_N).$$

Without loss of generality, we shall explain the basic idea underlying this method starting from the kth iteration $x_{h,k}$ at level h. Whenever possible, we will compute a search direction $d_{h,k}$ by resorting to problems on coarser levels recursively. If the current level is the coarsest level or the coarser level model is not a good choice, a direction $d_{h,k}$ will be computed directly on the current level h.

If a "*recursive search*" direction is chosen, we first move to the next coarsest level H with an initial point $x_{H,0} = R_h x_{h,k}$. Next we compute the minimizer (or approximate minimizer) x_{H,i^*} of the coarse level problem

$$\min \psi_H(x_H),$$

where ψ_H is an approximation of the original problem (1.1) on the coarse level H. The function ψ_H, which we will define below, depends on the point $x_{h,k}$ and the level h and will be different for different points. To simplify our notation, we will omit this dependence when referring to $\psi_H(\cdot)$ or its derivatives as this should not cause any

confusion. Then we project the direction $d_H^* = x_{H,i^*} - x_{H,0}$ on level H back to level h to obtain the recursive search direction

$$(3.2) \qquad d_{h,k} = P_h d_H^* = P_h \left(\sum_{i=0}^{i^*-1} \alpha_{H,i} d_{H,i} \right),$$

where $\alpha_{H,i}$ and $d_{H,i}$ are the step size and search direction, respectively, for the ith iteration on level H. Here each search direction $d_{H,i}$ from $x_{H,i}$ to $x_{H,i+1}$ for $i = 0, \cdots, i^* - 1$ is also computed recursively whenever possible.

If a "*direct search*" direction is chosen, $d_{h,k}$ is computed directly on level h. Many possibilities exist for how to compute such a direction. To illustrate our algorithm, we solve the Newton system:

$$(3.3) \qquad G_{h,k} d_{h,k} = -g_{h,k}$$

exactly or inexactly to obtain $d_{h,k}$, where we have used the notation $g_{h,k} = \nabla \psi_{h,k} = \nabla \psi_h(x_{h,k})$ and $G_{h,k} = \nabla^2 \psi_{h,k} = \nabla^2 \psi_h(x_{h,k})$. As stated above, we must use this direct search direction when the coarse level model is not appropriate. Specifically. we restrict the use of the recursive search direction at the point $x_{h,k}$ to the case where

$$(3.4) \qquad \|R_h g_{h,k}\| \geq \kappa \|g_{h,k}\|, \quad \|R_h g_{h,k}\| \geq \epsilon_h.$$

The reason for this is that $R_h g_{h,k}$ may be zero even though $g_{h,k}$ is not zero if $g_{h,k}$ lies in the null space of R_h; hence the current iterate appears to be a stationary point for ψ_H whereas it is not for ψ_h. These conditions are the same as those used in the multigrid algorithm proposed in [18, 16, 17].

Let us now define the coarse level approximation ψ_H explicitly. To ensure convergence and efficiency, the coarse level problem is not simply the discretized problem (1.1) for the coarse level H, but rather:

$$(3.5) \qquad \min_{x_H} \{\psi_H(x_H) \equiv f_H(x_H) - (v_H)^\top x_H\},$$

where $v_H = \nabla f_{H,0} - R_h g_{h,k}$. Furthermore, if we define $v_N = 0$, then model (3.5) can be naturally extended to all levels and the uppermost level model problem is exactly problem (3.1). Moreover, (3.5) enforces a certain coherence between the fine level problem ψ_h and the corresponding coarse level problem ψ_H.

LEMMA 3.1. *If we choose the recursive scheme to generate the direction* $d_{h,k} = P_h d_H^*$, *where the minimization on the coarse level H starts from the initial point* $x_{H,0} = R_h x_{h,k}$, *then the problems of the two consecutive levels h and H are first-order coherent in the sense that*

$$(3.6) \qquad g_{H,0} = R_h g_{h,k}, \quad (d_{h,k})^\top g_{h,k} = (d_H^*)^\top g_{H,0}.$$

Proof. The first part of (3.6) comes from the fact that

$$g_{H,0} = \nabla f_{H,0} - v_H = \nabla f_{H,0} - \nabla f_{H,0} + R_h g_{h,k} = R_h g_{h,k}.$$

This together with (3.2) and (2.2) and our assumption that $\sigma_h = 1$, implies that

$$(d_{h,k})^\top g_{h,k} = (P_h d_H^*)^\top g_{h,k} = (d_H^*)^\top R_h g_{h,k} = (d_H^*)^\top g_{H,0}.$$

☐

The following lemma shows, the recursive search direction $d_{h,k}$ is a descent direction for ψ_h at $x_{h,k}$ if $f_H(x_H)$ is a convex function.

LEMMA 3.2. *Suppose $f_H(x_H)$ is a convex function. If we choose the recursive scheme to generate the direction $d_{h,k} = P_h d_H^*$, where the minimization on the coarse level H starts from the initial point $x_{H,0} = R_h x_{h,k}$ and stops at the point x_{H,i^*} with $\psi_H(x_{H,i^*}) < \psi_H(x_{H,0})$, then $d_{h,k}$ is a descent direction; that is $(d_{h,k})^\top g_{h,k} < 0$. Moreover, the directional derivative $(d_{h,k})^\top g_{h,k}$ satisfies*

$$(3.7) \qquad\qquad -(d_{h,k})^\top g_{h,k} \geq \psi_{H,0} - \psi_{H,i^*}.$$

Proof. Since $f_H(x_H)$ is convex, so is $\psi_H(x_H)$; hence

$$(3.8) \qquad\qquad \psi_H(x_{H,i^*}) \geq \psi_H(x_{H,0}) + (x_{H,i^*} - x_{H,0})^\top g_{H,0}.$$

Hence we conclude that inequality (3.7) holds, and from the fact that $\psi_H(x_{H,i^*}) < \psi_H(x_{H,0})$, it follows that $(d_H^*)^\top g_{H,0} < 0$. ☐

In our algorithm, we chose a step size $\alpha_{h,k}$ along the direction $d_{h,k}$ that satisfies the Armijo-Wolfe conditions

$$(3.9a) \qquad\qquad \psi_h(x_{h,k} + \alpha_{h,k} d_{h,k}) \leq \psi_{h,k} + \rho_1 \alpha_{h,k}(g_{h,k})^\top d_{h,k},$$
$$(3.9b) \qquad\quad (\nabla \psi_h(x_{h,k} + \alpha_{h,k} d_{h,k}))^\top d_{h,k} \geq \rho_2 (g_{h,k})^\top d_{h,k},$$

where $0 < \rho_1 < \rho_2 < 1$ are two controlling parameters. The smaller ρ_2 is, the stricter the line search is. To select a step size $\alpha_{h,k}$ to satisfy (3.9a) and (3.9b), we refer the reader to Algorithms 3.2 and 3.3 in [30], which are based on interpolation and/or bisection. For a more detailed description of these kind of strategies, see, for example [28].

Our multigrid algorithm stops when the norm of the gradient is smaller than a given tolerance , i.e. $\|g_{h,k}\| \leq \epsilon_h$. It also limits the number of iterations to at most K at all levels h other than the finest level $h = N$.

ALGORITHM 3. $x_h = MLS(h, x_{h,0}, \tilde{g}_{h,0})$

Step 1. IF $h < N$, compute $v_h = \nabla f_{h,0} - \tilde{g}_{h,0}$, set $g_{h,0} = \tilde{g}_{h,0}$;
    ELSE set $v_h = 0$ and compute $g_{h,0} = \nabla f_{h,0}$.
Step 2. FOR $k = 0, 1, 2, \cdots$
   2.1. IF $\|g_{h,k}\| \leq \epsilon_h$ or if $h < N$ and $k \geq K$,
     RETURN solution $x_{h,k}$;
   2.2. IF $h > N_0$ and $\|R_h g_{h,k}\| \geq \kappa \|g_{h,k}\|$ and $\|R_h g_{h,k}\| \geq \epsilon_h$
     -Recursive Search Direction Computation
     Call $x_{h-1,i^*} = MLS(h-1, R_h x_{h,k}, R_h g_{h,k})$ to return a solution (or approximate solution) x_{h-1,i^*} of "$\min_{x_{h-1}} \psi_{h-1}(x_{h-1})$".
     Compute $d_{h,k} = P_h(x_{h-1,i^*} - R_h x_{h,k}) = P_h d_{h-1}^*$.
    ELSE
     -Direct Search Direction Computation
     Solve $G_{h,k} d_{h,k} = -g_{h,k}$ exactly or inexactly to obtain $d_{h,k}$.
   2.3. Call line search to obtain a step size $\alpha_{h,k}$ that satisfies the Armijo-Wolfe conditions (3.9a) and (3.9b).
   2.4. Set $x_{h,k+1} = x_{h,k} + \alpha_{h,k} d_{h,k}$.

REMARK 3.3. *Algorithm 3 automatically chooses between the direct search direction and the recursive direction based on condition* (3.4). *Therefore it can be viewed as a combination of the global linearization method and the FAS scheme when applied to nonlinear PDEs.*

One element in Algorithm 3 that we have not fully specified is how to solve the Newton system (3.3) in the direct search direction computation. The most straightforward way is to solve (3.3) exactly using factorization methods [15]. However, doing this is very expensive on the finer levels. A very natural adaptive strategy is the following. Whenever we are on levels where the total number of variables is not too large and the corresponding Hessian is sparse, compute the Hessian and its Cholesky factorization directly, i.e., (3.3) is solved exactly; for all other cases, we only solve (3.3) to a certain accuracy by using a (preconditioned) conjugate gradient method [15] or the multigrid Algorithm 2.

4. Convergence Analysis. Throughout this section, we define

$$(4.1) \qquad \varpi \stackrel{def}{=} \max\{1, \max_{i=1,\cdots,N} \|P_i\|\} = \max\{1, \max_{i=1,\cdots,N} \|R_i\|\} < \infty,$$

and adopt some concepts and notation from [18, 16, 17].

1. We shall refer to the kth iteration on level h as iteration (h, k). We define the iteration (h, k) as the predecessor of a minimization sequence that consists of all successive iterations on level $h - 1$ until a return is made to level h. If iteration $(h - 1, l)$ is in this minimization sequence, we use the notation $(h, k) = \pi(h - 1, l)$ to indicate this.

2. For iteration (h, k), we define the set

$$(4.2) \qquad \mathcal{R}(h, k) \stackrel{def}{=} \{(j, l) \mid \text{iteration } (j, l) \text{ occurs within iteration } (h, k)\}$$

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

$$(4.3) \qquad p(h, k) \stackrel{def}{=} \min_{(j,l)\in\mathcal{R}(h,k)} j$$

3. We denote by $\mathcal{T}(h, k)$ the subset of iterations $(j, l) \in \mathcal{R}(h, k)$ in which $d_{j,l}$ is a direct search direction, i.e.,

$$(4.4) \qquad \mathcal{T}(h, k) \stackrel{def}{=} \{(j, l) \in \mathcal{R}(h, k) \mid d_{j,l} \text{ is a direct search direction }\}.$$

4.1. Uniformly Convex Problems. In this subsection, we assume

ASSUMPTION 4.1. $f_h(x)$ *is twice continuously differentiable and uniformly convex; that is, there exist constants* $0 < m_h < M_h < \infty$ *such that*

$$(4.5) \qquad m_h\|d\|_2^2 \leq d^\top \nabla^2 f_h(x)d \leq M_h\|d\|_2^2, \quad \forall d \in \Re^{n_h},$$

for all $x \in \{x \mid f_h(x_h) \leq f_h(x_{h,0})\}$. *Moreover, let* $m = \min_h\{m_h\}, M = \max_h\{M_h\}$.

Since $\psi_h(x)$ differs from $f_h(x)$ only by a linear term, Assumption 4.1 also holds for $\psi_h(x)$. In the following, we state some useful properties of convex functions.

LEMMA 4.2. *([32]: Lemma 5.3.4) Suppose* $f_h(x)$, *and hence* $\psi_h(x)$, *satisfy Assumption 4.1.*

1. *If $\psi_h(y) \leq \psi_h(x)$, then*

$$(4.6) \qquad\qquad \|\nabla\psi_h(x)\| \geq \frac{m}{2}\|y - x\|.$$

2. *For all x,*

$$(4.7) \qquad \frac{m}{2}\|x - x^*\|^2 \leq \psi_h(x) - \psi_h(x^*) \leq \frac{1}{m}\|\nabla\psi_h(x)\|^2,$$

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

LEMMA 4.3. *([32]: Theorem 2.5.8) Suppose $\psi_h(x)$ satisfies Assumption 4.1. If α is a step size that satisfies the Armijo condition (3.9a) along a descent direction d, then the decrease of $\psi_h(x)$ satisfies $\psi_h(x) - \psi_h(x+\alpha d) \geq c_1\|\alpha d\|^2$ with $c_1 = \frac{\rho_1 m}{1+\sqrt{M/m}}$.*

We will also make use of the following inequality.

LEMMA 4.4. *Let $d_1, d_2, \cdots, d_k$ be vectors in $\mathbb{R}^n$. Then $\sum_{j=1}^k \|d_j\|^2 \geq \frac{1}{k}\|\sum_{j=1}^k d_j\|^2$.*

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.

$$\sum_{j=1}^k \|d_j\|^2 - \frac{1}{k}\|\sum_{j=1}^k d_j\|^2$$

$$\geq \frac{1}{k-1}\|\sum_{j=1}^{k-1} d_j\|^2 + \|d_k\|^2 - \frac{1}{k}\left(\|\sum_{j=1}^{k-1} d_j\|^2 + \|d_k\|^2 + 2\left(\sum_{j=1}^{k-1} d_j\right)^\top d_k\right)$$

$$= \frac{1}{k}\left(\frac{1}{k-1}\|\sum_{j=1}^{k-1} d_j\|^2 + (k-1)\|d_k\|^2 - 2\left(\sum_{j=1}^{k-1} d_j\right)^\top d_k\right)$$

$$\geq 0,$$

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

Based on the curvature condition (3.9b) and the uniform convexity of ψ_h, we have

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

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

$$(4.8) \qquad\qquad \alpha_{j,l} \geq (1 - \rho_2)\frac{|(g_{j,l})^\top d_{j,l}|}{M\|d_{j,l}\|^2}.$$

We will now show that for certain search directions both $\alpha_{j,l}$ and $\cos(\theta_{j,l})$, where $\theta_{j,l}$ is the angle between $d_{j,l}$ and the steepest descent direction $-g_{j,l}$, are bounded away from zero. Therefore, for such choices, the minimization sequence generated by Algorithm 3 on the uppermost finest level is globally convergent whereas the minimization sequences on all other coarser levels are either globally convergent or stop after at most K steps.

Let us first consider the direct search direction. Specifically, we show that these particular choices for this direction satisfy:

CONDITION 4.5. *If iteration $(j, l) \in \mathcal{T}(h, k)$, the step direction $d_{j,l}$ and the step size $\alpha_{j,l}$ satisfy*

$$(4.9) \qquad \alpha_{j,l} \geq \alpha_{\mathcal{T}}, \ \|d_{j,l}\| \leq \beta_{\mathcal{T}} \|g_{j,l}\| \ and \ -(d_{j,l})^{\top} g_{j,l} \geq \eta_{\mathcal{T}} \|g_{j,l}\|^2,$$

where $\alpha_{\mathcal{T}}$, $\delta_{\mathcal{T}}$ and $\eta_{\mathcal{T}}$ are positive constants.

Note that the last two inequalities in (4.9) imply that $\cos(\theta_{j,l}) \geq \eta_{\mathcal{T}}/\beta_{\mathcal{T}}$. The steepest descent search direction $d_{j,l} = -g_{j,l}$ obviously satisfies Condition 4.5. The following lemma shows that the exact Newton step satisfies Condition 4.5.

LEMMA 4.6. *If iteration $(j, l) \in \mathcal{T}(h, k)$ and the step direction $d_{j,l}$ satisfies $G_{j,l} d_{j,l} = -g_{j,l}$ exactly, then Condition 4.5 is satisfied with parameters*

$$(4.10) \qquad \alpha_{\mathcal{T}} = (1 - \rho_2)\frac{m^2}{M^2}, \ \beta_{\mathcal{T}} = \frac{1}{m} \ and \ \eta_{\mathcal{T}} = \frac{1}{M},$$

Proof. Since the step $d_{j,l}$ satisfies $G_{j,l} d_{j,l} = -g_{j,l}$ exactly and Assumption 4.1 holds, we have

$$-(d_{j,l})^{\top} g_{j,l} = |(g_{j,l})^{\top} G_{j,l}^{-1} g_{j,l}| \geq M^{-1} \|g_{j,l}\|_2^2$$

and

$$\|d_{j,l}\|_2^2 = (g_{j,l})^{\top} G_{j,l}^{-2} g_{j,l} \leq m^{-2} \|g_{j,l}\|_2^2,$$

which together with (4.8) yields (4.9) with parameters (4.10). □

Now consider the inexact Newton step generated by the conjugate gradient method (CG) method. Given initial values $s_0 = 0$, $r_0 = g$ and $p_0 = -r_0$, the CG method for solving the system linear equations $Gd = -g$ generates

$$s_{i+1} = s_i + \alpha_i p_i, \quad r_{i+1} = r_i + \alpha_i G p_i, \quad p_{i+1} = -r_{i+1} + \beta_{i+1} p_i, \ for \ i = 0, 1, \cdots,$$

where $\alpha_i = \frac{r_i^{\top} r_i}{p_i^{\top} G p_i}$ and $\beta_{i+1} = \frac{r_{i+1}^{\top} r_{i+1}}{r_i^{\top} r_i}$. The solution is set to $d = s_i$ when a certain accuracy is achieved. The CG method is invariant under an orthogonal transformation [38]. Define $\bar{g} = Q^{\top} g$ and $\bar{G} = Q^{\top} G Q$, where Q is any given orthogonal matrix. Let (s_i, r_i, p_i) and $(\bar{s}_i, \bar{r}_i, \bar{p}_i)$ be iterates generated by the CG method applied to $Gd = -g$ and $\bar{G}\bar{d} = -\bar{g}$, respectively. Then, it can be shown that $\bar{s}_i = Q^{\top} s_i$, $\bar{r}_i = Q^{\top} r_i$ and $\bar{p}_i = Q^{\top} p_i$. In particular, for any given $g \in \mathbb{R}^{n_j}$ and any symmetric matrix G, there exists an orthogonal matrix Q such that $Q^{\top} g$ is parallel to the first coordinate direction and $Q^{\top} G Q$ is a tridiagonal matrix. Specifically, we have

$$(4.11) \qquad \bar{g} = \|g\| e_1^{\top}, \quad \bar{G} = \begin{pmatrix} u_1 & v_1 & 0 & \cdots & 0 & 0 \\ v_1 & u_2 & v_2 & \cdots & 0 & 0 \\ 0 & v_2 & u_3 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & u_{n_j-1} & v_{n_j-1} \\ 0 & 0 & 0 & \cdots & v_{n_j-1} & v_{n_j} \end{pmatrix},$$

where e_i is a vector whose ith element is one and all other elements are zero. We also denote the submatrix of the first i rows and i columns of $\bar{G}$ by $\bar{G}^i$. Then Lemma 3 in

[38] shows us that

$$(4.12) \qquad \bar{s}_{i+1} = -\|g\| \begin{pmatrix} (\bar{G}^i)^{-1} e_1 \\ 0 \end{pmatrix}, \quad \bar{r}_{i+1} = (-1)^i e_{i+1} \|g\| \frac{\prod_{l=1}^{i} v_l}{Det(\bar{G}^i)}.$$

Using the facts mentioned above, the following lemma shows that the inexact Newton step generated by the CG method satisfies Condition 4.5.

LEMMA 4.7. *If iteration $(j,l) \in \mathcal{T}(h,k)$ and the step $d_{j,l}$ is generated by the conjugate gradient method, then Condition 4.5 is satisfied with parameters (4.10).*

Proof. For ease of notation, we temporarily drop the subscripts (j,l). Since the CG method is invariant under an orthogonal transformation, we can analyze its behaviour when applied to the system of equations $\bar{G}\bar{d} = -\bar{g}$, where $\bar{G}$ and $\bar{g}$ have the form given in (4.11). Because an orthogonal transformation does not change the eigenvalues of a matrix, Assumption 4.1 holds for $\bar{G}$. The well-known interlacing eigenvalue theorem for bordered matrices also implies that Assumption 4.1 holds for all submatrices $\bar{G}^i$. From the relationship between (g_i, s_i) and $(\bar{g}_i, \bar{s}_i)$ and the fact (4.12), it follows that

$$-g^\top s_{i+1} = -(\bar{g})^\top \bar{s}_{i+1} = \|g\|^2 e_1^\top (\bar{G}^i)^{-1} e_1 \geq M^{-1} \|g\|^2.$$

In addition,

$$\|s_{i+1}\| = \|\bar{s}_{i+1}\| = \|g\| \, \|(\bar{G}^i)^{-1} e_1\| \leq m^{-1} \|g\|.$$

Therefore, since $d = s_{i+1}$ when the algorithm exits, we obtain (4.9) with parameters (4.10) similar to Lemma 4.6. $\square$

The following lemmas show that the recursive steps satisfy properties that enable us to prove convergence of our multigrid method if the direct search directions satisfy Condition 4.5.

LEMMA 4.8. *Suppose iteration $(j,l) \in \mathcal{R}(h,k) \backslash \mathcal{T}(h,k)$ and Condition 4.5 is satisfied by all direct search steps. Then the step size $\alpha_{j,l}$ is bounded below:*

$$(4.13) \qquad \alpha_{j,l} \geq \alpha_{\mathcal{I}} = \frac{c_1(1-\rho_2)}{MK\varpi^2},$$

where K, defined in Algorithm 3, is the maximum number of iterations of the minimization sequence at level $j-1$. Therefore, $\alpha_{j,l} \geq \alpha^* = \min\{\alpha_{\mathcal{T}}, \alpha_{\mathcal{I}}\}$ for any $(j,l) \in \mathcal{R}(h,k)$.

Proof. From the inequality (3.7), it follows that

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

Since the minimization sequence is monotonically decreasing, the reductions of the function value satisfy

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

Since ψ_{j-1} is uniformly convex, it follows from Lemma 4.3 that

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

Using Lemma 4.4 and the fact that the total number of iterations at level $h-1$ is less than K, we have

$$(4.14) \qquad -(d_{j,l})^\top g_{j,l} \geq c_1 \frac{1}{i^*} \| \sum_{k=0}^{i^*-1} \alpha_{j-1,k} d_{j-1,k} \|^2 \geq \frac{c_1}{K} \| d_{j-1}^* \|^2 \geq \frac{c_1}{K\varpi^2} \| d_{j,l} \|^2,$$

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

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

Therefore, combining (4.8) and (4.14), we obtain

$$\alpha_{j,l} \geq \frac{c_1(1-\rho_2)}{MK\varpi^2},$$

which completes the proof. $\square$

LEMMA 4.9. *Suppose iteration* $(j,l) \in \mathcal{R}(h,k)\backslash \mathcal{T}(h,k)$ *and Condition 4.5 is satisfied by all direct search steps. Let p be the deepest level in $\mathcal{R}(j,l)$ such that*

$$(4.15) \qquad g_{p,0} = R_{p+1} g_{p+1,0} = \cdots = R_{p+1} \cdots R_j g_{j,l}.$$

Then for any iteration $(q,k) = (q,0)$, where $p < q < j$, and for iteration $(q,k) = (j,l)$, we have

$$(4.16) \qquad \cos(\theta_{q,k}) \geq \delta_{q-p} \ and \ -(d_{q,k})^\top g_{q,k} \geq \eta_{q-p} \| g_{q,k} \|^2,$$

where $\delta_{q-p} = \frac{m}{2} \eta_{q-p}$ and $\eta_i = \left(\alpha^* \rho_1 \kappa^2 \right)^i \eta_{\mathcal{T}}$.

Proof. 1. We will prove (4.16) 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 (3.7), it follows that

$$(4.17) \qquad -(d_{p+1,0})^\top g_{p+1,0} \geq \psi_{p,0} - \psi_{p,i^*} \geq \psi_{p,0} - \psi_{p,1} \geq -\alpha_{p,0} \rho_1 (d_{p,0})^\top g_{p,0},$$

where the last inequality comes from the Armijo condition (3.9a) for iteration $(p,0)$. Since $(p,0)$ is computed directly,

$$-(d_{p,0})^\top g_{p,0} \geq \eta_{\mathcal{T}} \| g_{p,0} \|_2^2.$$

From (4.15) and the first condition in (3.4), we obtain

$$(4.18) \qquad \| g_{p,0} \|_2^2 = \| R_{p+1} g_{p+1,0} \|_2^2 \geq \kappa^2 \| g_{p+1,0} \|_2^2.$$

Combining all of these facts together, we get

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

which proves the second inequality of (4.16) for $q = p+1$. From Lemma 4.2, we obtain

$$\| g_{p+1,0} \|_2 \geq \frac{m}{2} \| d_{p+1,0} \|,$$

which completes the proof of the first inequality of (4.16).

Now, suppose (4.16) holds for $p < q < j - 1$; we prove that (4.16) also holds for $q + 1$. Similar to the case $q = p + 1$, we have

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

since relationship (4.18) also holds with p replaced by q. Using Lemma 4.2 again, we obtain (4.16).

2. For iteration (j, l), inequality (4.16) holds by simply repeating, in an analogous fashion, the above proof:

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

□

We can now prove the global convergence of Algorithm 3.

THEOREM 4.10. *Suppose Condition 4.5 is satisfied by all direct search steps. Then the iterative sequence $\{x_{N,k}\}$ generated by Algorithm 3 at the uppermost level converges to the unique minimizer of $f_N(x_N)$.*

Proof. The step size $\alpha_{N,k}$ at the uppermost level is bounded from below by a constant $\alpha^* > 0$ from Lemma 4.8. From the Armijo condition (3.9a), we have

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

Therefore, since by Assumption 4.1 $\psi(\cdot)$ is bounded below, $\lim_{k \to \infty} d_{N,k}^\top g_{N,k} = 0$. From Lemma 4.9, we have

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

for some constant σ. This shows that

$$
(4.19) \qquad \lim_{k \to \infty} \|\nabla f_N(x_{N,k})\| = 0
$$

holds, since $\nabla f_N(x_{N,k}) = g_{N,k}$ (recall that $v_N = 0$). The uniqueness of the minimizer is from the strict convexity of $f_N(x_N)$ in Assumption 4.1. □

We now prove R-linear convergence.

THEOREM 4.11. *Suppose Condition 4.5 is satisfied by all direct search steps. Assume that the iterative sequence $\{x_{N,k}\}$ generated by Algorithm 3 at the uppermost level converges to the unique minimizer $\{x_N^*\}$ of $f_N(x_N)$. Then the rate of convergence is at least R-linear.*

Proof. Again from Condition 4.5 and Lemma 4.9, we have

$$
(4.20) \qquad f_N(x_{N,k+1}) - f_N(x_{N,k}) \leq -\alpha^* \eta_N \|\nabla f_N(x_{N,k})\|^2.
$$

From the second inequality of (4.7) in Lemma 4.2, we get

$$
\|\nabla f_N(x_{N,k})\|^2 \geq m \left(f_N(x_{N,k}) - f_N(x_N^*)\right).
$$

Hence

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

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

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

From the first inequality of (4.7) in Lemma 4.2, we obtain that

$$f_{\mathrm{N}}(x_{\mathrm{N},k}) - f_{\mathrm{N}}(x_{\mathrm{N}}^*) \geq \frac{m}{2} \|x_{\mathrm{N},k} - x_{\mathrm{N}}^*\|^2.$$

Hence

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

$\square$

COROLLARY 4.12. *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 $0 < c = 1 - \frac{m\alpha^* \eta_{\mathrm{N}}}{2} < 1$, we have $f_{\mathrm{N}}(x_{\mathrm{N},k}) - f_{\mathrm{N}}(x_{\mathrm{N}}^*) \leq \epsilon$.

Proof. With the help of inequality (4.20) and from the standard convergence analysis for convex functions [8], we have the result immediately. $\square$

4.2. Relaxing the Uniform Convexity Assumption. In this subsection, we prove the global convergence of Algorithm 3 for general convex functions. Let us replace Assumption 4.1 by the following assumption.

ASSUMPTION 4.13. ,
1. *The level set $\mathcal{D}_h = \{x_h : \psi_h(x_h) \leq \psi_h(x_{h,0})\}$ is bounded.*
2. *The objective function ψ_h is convex and continuously differentiable, and there exists a constant $L > 0$ such that*

$$(4.21) \qquad \|\nabla \psi_h(x_h) - \nabla \psi_h(\tilde{x}_h)\| \leq L \|x_h - \tilde{x}_h\|, \text{ for all } x_h, \tilde{x}_h \in \mathcal{D}_h.$$

3. *The Hessian matrix is bounded*

$$(4.22) \qquad\qquad\qquad \|G_h(x_h)\| \leq M$$

for all x_h in the level set $\mathcal{D}_h$.

This assumption implies that there is a constant γ such that

$$(4.23) \qquad\qquad \|\nabla \psi_h(x_h)\| \leq \gamma, \text{ for all } x_h \in \mathcal{D}_h.$$

Let us first consider the direct search direction. Since the Hessian $G_{h,k}$ is only positive semi-definite, the direction $d_{h,k}$ generated by solving the system of equations

$G_{h,k}d_{h,k} = -g_{h,k}$ may not satisfy Condition 4.5. One strategy is to add a small positive multiple of the identity matrix I_h to $G_{h,k}$ so that $\widetilde{G}_{h,k} = G_{h,k} + mI_h \succeq mI_h$ for some constant $m > 0$. Then $d_{h,k}$ can be computed by solving the modified Newton system

$$(4.24) \qquad\qquad\qquad \widetilde{G}_{h,k}d_{h,k} = -g_{h,k}.$$

As in the proofs of Lemmas 4.6 and 4.7, we can easily show that the exact Newton step and the inexact Newton step generated by the conjugate gradient method satisfies Condition 4.5. We also make the following assumption:

ASSUMPTION 4.14. *The step size $a_{h,k}$ is bounded from above for any iteration (h, k); i.e., $\alpha_{h,k} \leq \widetilde{\alpha}$.*

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

LEMMA 4.15. *Suppose Condition 4.5 is satisfied by all direct search steps. Then $\|d_{j,l}\| \leq \widetilde{\gamma}$ for all iterations $(j, l) \in \mathcal{R}(h, k)$.*

Proof. 1. If iteration $(j, l) \in \mathcal{T}(h, k)$, we obtain

$$\|d_{j,l}\| \leq \beta_{\mathcal{T}}\|g_{j,l}\| \leq \gamma\beta_{\mathcal{T}}$$

from Condition 4.5 and the fact (4.23).

2. Now consider iteration $(j, l) \in \mathcal{R}(h, k) \backslash \mathcal{T}(h, k)$. Let p be the deepest level in $\mathcal{R}(j, l)$ such that

$$(4.25) \qquad\qquad g_{p,0} = R_{p+1}g_{p+1,0} = \cdots = R_{p+1}\cdots R_j g_{j,l}.$$

We prove this part by induction on levels. Since p is the coarsest level and the total number of iterations at level p is less than K, we obtain

$$\|d_{p+1,l}\| = \|P_{p+1}(\sum_{k=0}^{i^*-1} \alpha_{p,k}d_{p,k})\| \leq \varpi\widetilde{\alpha}\sum_{k=0}^{i^*-1}\|d_{p,k}\| \leq \varpi\widetilde{\alpha}\mathrm{K}\gamma\beta_{\mathcal{T}}$$

which proves the case $q = p + 1$. Suppose the lemma is true for level $q - 1$, we prove that it is also true for level q. Similarly, we have

$$\|d_{q,l}\| = \|P_q(\sum_{k=0}^{i^*-1} \alpha_{q-1,k}d_{q-1,k})\| \leq \varpi\widetilde{\alpha}\sum_{k=0}^{i^*-1}\|d_{q-1,k}\| \leq \varpi\widetilde{\alpha}\mathrm{K}\widetilde{\gamma}.$$

For ease of notation, we still denote the right hand side by $\widetilde{\gamma}$, which is finite and bounded from above since there is only a finite number of levels and the number of iterations of each minimization sequence on the coarser levels is at most K. This completes the proof. $\square$

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 power.

LEMMA 4.16. *Suppose iteration $(j, l) \in \mathcal{R}(h, k) \backslash \mathcal{T}(h, k)$ and Condition 4.5 is satisfied by all direct search steps. Let p be the deepest level in $\mathcal{R}(j, l)$ such that (4.25) satisfies. Then for any iteration $(q, k) = (q, 0)$, where $p < q < j$, and for iteration $(q, k) = (j, l)$, we have*

$$(4.26) \qquad -d_{q,k}^\top g_{q,k} \geq \left(\frac{\rho_1(1-\rho_2)}{L}\right)^{(2^{i-1}-1)} \frac{(\rho_1\alpha_{\mathcal{T}}\eta_{\mathcal{T}}\kappa^{2(j-p)}\|g_{j,l}\|^2)^{2^{i-1}}}{\widetilde{\gamma}^{2^i-2}},$$

$$(4.27) \qquad \alpha_{q,k} \geq \rho_1^{(2^{i-1}-1)}\left(\frac{1-\rho_2}{L}\right)^{(2^{i-1})} \frac{(\rho_1\alpha_{\mathcal{T}}\eta_{\mathcal{T}}\kappa^{2(j-p)}\|g_{j,l}\|^2)^{2^{i-1}}}{\widetilde{\gamma}^{2^i}},$$

where $i = q - p$.

Proof. 1. We prove this lemma by induction on level q. First, let us consider iteration $(p+1, 0)$. From inequality (3.7) and Condition 4.5, it follows that

$$-d_{p+1,0}^{\top} g_{p+1,0} \geq \psi_{p,0} - \psi_{p,1} \geq -\rho_1 \alpha_{p,0} d_{p,0}^{\top} g_{p,0}$$
$$\geq \rho_1 \alpha_{\mathcal{T}} \eta_{\mathcal{T}} \|g_{p,0}\|^2$$
$$\geq \rho_1 \alpha_{\mathcal{T}} \eta_{\mathcal{T}} \kappa^{2(j-p)} \|g_{j,l}\|^2$$

which proves inequality (4.26). Based on the curvature condition (3.9b) and the Lipschitz continuity of $\nabla \psi_h$, 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 4.15, we obtain

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

which proves inequality (4.27). Now suppose inequalities (4.26) and (4.27) 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

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

Using Lemma 4.15 again, we obtain inequality (4.27).

2. For iteration (j, l), inequalities (4.26) and (4.27) hold by simply repeating, in an analogous fashion, the above proof. □

Now we establish the global convergence of Algorithm 3.

THEOREM 4.17. *Suppose Condition 4.5 is satisfied by all direct search steps. Then in Algorithm 3 at the uppermost level*

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

Proof. The proof is by contradiction. Assume that $\|g_{N,k}\|$ is bounded away from zero; that is, there is a constant $\epsilon > 0$ such that

(4.28) $\|\nabla f_{\mathrm{N},k}\| = \|g_{\mathrm{N},k}\| \geq \epsilon > 0,$ for all k sufficiently large.

From Condition 4.5 and Lemma 4.16, it follows that each iteration satisfies

(4.29) $-\alpha_{\mathrm{N},k} d_{\mathrm{N},k}^{\top} g_{\mathrm{N},k} \geq \sigma \|g_{\mathrm{N},k}\|^i,$

where σ is a positive constant and the order i can only be selected from a finite set of integers $\{2^1, 2^2, \cdots, 2^{N-N_0+1}\}$, whether or not the direction $d_{N,k}$ is a direct search direction or a recursive search direction. Let B_j be the subset of indices k such that inequality (4.29) is satisfied with power $i = 2^j$. From the Armijo condition (3.9a), we have

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

Summing over k and taking limits, we obtain

$$\infty > \psi_{N,0} - \psi_{N,\infty} \geq -\sum_{k=0}^{\infty} \alpha_{N,k} d_{N,k}^\top g_{N,k} \geq \sum_{j=1}^{N-N_0} \sum_{k \in B_j} \sigma \|g_{N,k}\|^{2^j}.$$

Since at least one index set B_l is infinite, we have

$$\sum_{k \in B_l} \sigma \|g_{N,k}\|^{2^j} \geq \sum_{k \in B_l} \sigma \epsilon^{2^l} = \infty,$$

which is a contradiction. ☐

REMARK 4.18. *Theorem 4.17 still holds if the bound of the step size for direct search directions in Condition 4.5 is relaxed to $\alpha_{j,l} \geq \alpha_{\mathcal{T}} \|g_{j,l}\|^2$.*

Theorem 4.17 shows that there exists a subsequence of $\{x_{N,k}\}$ converging to a stationary point x_N^* of problem (3.1). However, since f_N is convex and the sequence $f_N(x_{N,k})$ converges, every accumulation point of $\{x_{N,k}\}$ is a global optimal solution of problem (3.1).

COROLLARY 4.19. *Suppose Condition 4.5 is satisfied by all direct search steps. Let $\{x_{N,k}\}$ be the sequence generated by Algorithm 3 at the uppermost level. Then the whole sequence $\{\nabla f_{N,k}\}$ converges to zero and every accumulation point of $\{x_{N,k}\}$ is a global optimal solution of problem (3.1).*

5. Practical Issues. In this section, we discuss some other components of the multigrid method, including different ways to generate search directions, different strategies for doing smoothing steps and ways to define prolongation and restriction operators. Finally, the full multigrid method, which is used to enhance the performance of the multigrid method for solving PDEs, is extended to our optimization context.

5.1. Direct Search Directions. In our multigrid algorithm, the direct search direction $d_{h,k}$ can be computed by minimization algorithms (combined with a proper line search scheme) other than Newton's method or the steepest descent method. These methods can be applied to compute $d_{h,k}$ without any difficulty because they are totally independent of any information from previous iterates $x_{h,0}, \cdots, x_{h,k-1}$. History dependent methods, such as nonlinear conjugate gradient methods and quasi-Newton methods, need to be carefully tailored here since $d_{h,k-1}$ may be computed recursively. One simple strategy is to start from scratch every time these methods are called after a recursive step. But how to utilize these incompatible recursive steps $d_{h,i}$ needs more study.

5.2. Smoothing Steps. Traditionally, the use of smoothing steps is motivated by their ability to smooth errors. Generally speaking, multigrid methods [19, 35] for PDEs will not converge without pre- and post-smoothing. While we can prove convergence of Algorithm 3 without using smoothing steps after each recursive step,

we computationally explore the possibility of improving the efficiency of Algorithm 3 by incorporating smoothing steps in section 6.

For quadratic functions, we only require smoothing steps to satisfy condition (2.10) to guarantee global convergence. For general convex functions, we require Condition 4.5 to be satisfied. Adding smoothing steps in the framework of Algorithm 3 is fairly easy. One simple strategy is to do some smoothing direct search steps after a recursive step. When doing smoothing, we don't want to spend too much time, especially for the expensive problems at the finer levels. Several iterations of the steepest descent method or a conjugate gradient method or a limited memory BFGS method are appropriate in this case. However, in our preliminary computational tests, we used a Newton's Method as a smoother but only solved the Newton equation inexactly.

5.3. Prolongations and Restrictions. The mechanics of prolongation and restriction are similar to their counterparts in multigrid methods for PDEs [19, 34, 35]. For example, the nine-point prolongation P_h is often represented by the stencil

$$P_h \stackrel{def}{=} \begin{bmatrix} 1/4 & 1/2 & 1/4 \\ 1/2 & 1 & 1/2 \\ 1/4 & 1/2 & 1/4 \end{bmatrix}.$$

We require that the restriction R_h be compatible with the prolongation P_h in the sense that $\sigma_h P_h = R_h$. First, define the discretized version of the continuous L_2 inner product at level h

(5.1) $$\langle u_h, v_h \rangle_h = \omega_h^x \omega_h^y \sum_{(i,j) \in \Omega_h} u_h(i,j) \, v_h(i,j),$$

where the scaling factor $\omega_h^x \omega_h^y$ [19, 34] allows us to compare the grid functions on different grids and the corresponding continuous function on Ω_h. Let $u_h = P_h u_H$, then

$$\langle u_h, v_h \rangle_h = \langle P_h u_H, v_h \rangle_h = \langle u_H, P_h^* v_h \rangle_H,$$

where P_h^* is the adjoint of P_h with respect to the inner product (5.1). Therefore, restriction can be defined as

$$R_h = P_h^*.$$

For the nine point prolongation, we have $R_h = \frac{1}{4} P_h^\top$ where $P_h^\top$ is the transpose of P_h.

5.4. Full Multigrid Method. The basic multigrid method solves problem (3.1) by calling $x_{N,i^*} = MLS(N, x_{N,0}, 0)$. Since starting from a good initial point usually reduces the total number 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 N_0 where the discretized problem is very easily solved. We interpolate this solution to the next finer level $h + 1$ as an initial approximation. Thereafter, Algorithm 3 is applied to the discretized problem at level $h + 1$. This process is then repeated over and over until we reach the uppermost level. The detailed algorithm is as follows.

ALGORITHM 4. *Full Multigrid Method FMLS*

Step 1. For $h = N_0 < N$, set initial approximation x_h

FIG. 5.1. *An illustration of the full multigrid Algorithm 4*
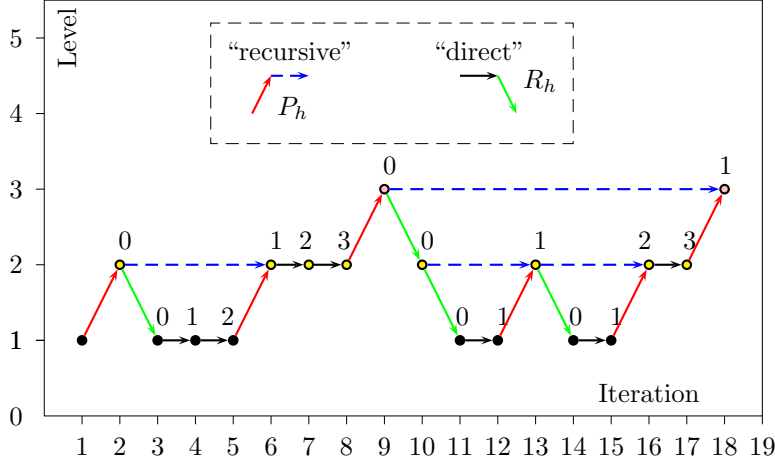
Step 2. FOR $k = 0, 1, 2, \cdots, N - 1$

2.1. Interpolate x_h to the next finer working level $x_{h+1,0} = P_h x_h$.

2.2. Call $x_{h+1} = MLS(h+1, x_{h+1,0}, 0)$ starting with $x_{h+1,0}$, i.e., apply multigrid Algorithm 3 to solve the discretized problem on level $h + 1$

$$\min_{x_{h+1}} f_{h+1}(x_{h+1})$$

To illustrate the full multigrid Algorithm 4, we simulate some steps of the algorithm running on a problem from level 1 to level 3 and show the relationship between the level and the iteration history in Figure 5.1. The x-axis denotes the index of the iteration in the whole minimization procedure whereas the y-axis denotes the level at which the minimization procedure takes places at a given iteration. We denote a direct search direction by $\rightarrow$, a recursive search direction by $\dashrightarrow$, a prolongation operation by $\nearrow$ and a restriction operation by $\searrow$. We mark each iteration in each minimization sequence by a circle and also record its order in the corresponding minimization sequence. For example, there are two minimization sequences on level 2 that we applied to functions that differ by only a linear term. The first minimization sequence on level 2 first resorts to the coarse level model and initializes a minimization sequence on that level to compute a recursive step. This recursive step direction is marked by $\dashrightarrow$ from the point 0 to the point 1 at level 2. At the point 1, a direct search direction has to be called to compute the point 2 since the switching condition (3.4) fails. It is possible that the switching condition (3.4) still fails at point the 2, then another direct search direction has to be computed. This procedure continues until the convergence condition is satisfied and the algorithm interpolates the solution to level 3 as an initial point.

6. Numerical Tests.

6.1. Test Problems and Discretization Issues. In this section, we apply our multigrid approach to infinite dimensional unconstrained minimization problems of the form

$$(6.1) \qquad \min_{u \in U} \mathcal{F}(u) = \int_{\Omega} \mathcal{L}(\nabla u, u, x) \; dx$$

Name	Direct Search Direction	Smoothing Steps
NEWTON	PCG	0
FMLS0-PCG	PCG	0
FMLS1-PCG	PCG	1
FMLS0-MG	MG	0
FMLS1-MG	MG	1

subject to proper boundary conditions, where U is the functional space wherein u resides. Such problems arise in a wide range of applications, such as variational formulations of nonlinear PDEs, image processing problems, optimal control problems and inverse problems.

For the sake of simplicity, only the finite difference case and only the two dimensional case of a simple domain $\Omega = [0,1] \times [0,1]$ with Dirichlet boundary conditions are treated. Neumman conditions and other boundary conditions can be handled in a similar way by first eliminating the boundary nodes. We discretize Ω at level h as a square grid

$$\Omega_h = \{(i,j) \stackrel{def}{=} (x_i, y_j) \mid x_i = i\omega_h^x, y_j = j\omega_h^y, i = 0, 1, \cdots, n_h^x; j = 0, 1, \cdots, n_h^y\},$$

where the mesh size $\omega_h^x = 1/n_h^x$ and $\omega_h^y = 1/n_h^y$ and we take $n_h^x = n_h^y = 2^h$ for the sake of simplicity. Then the objective functional is discretized as

$$(6.2) \qquad F(u) = \frac{1}{2} \sum_{i=0}^{n_h^x-1} \sum_{j=0}^{n_h^y-1} \mathcal{L}(\delta_x^+ u_{i,j}, \delta_y^+ u_{i,j}, u_{i,j}) + \mathcal{L}(\delta_x^- u_{i,j}, \delta_y^- u_{i,j}, u_{i,j}).$$

where $\delta_x^+ u_{i,j}, \delta_y^+ u_{i,j}$ and $\delta_x^- u_{i,j}, \delta_y^- u_{i,j}$ are, respectively, the forward and backward finite differences with respect to x and y.

6.2. Performance of the Multigrid Methods. We mainly focus on the performance of the full multigrid Algorithm 4 with zero or one smoothing step. When we specify that a particular version of Algorithm 3 does k smoothing steps, we mean that before considering doing a recursive step, the algorithm first takes k direct search steps. It may take additional direct search steps if the test for doing a recursive step is not met. The direct search directions on the coarsest level are obtained by factorizing the Hessian. The direct search directions on other levels are computed by a preconditioned conjugate gradient method (PCG) using an incomplete Cholesky factorization pre-conditioner. We denote the version of the method that does not use smoothing steps by "FMLS0-PCG" and the version that uses one smoothing step by "FMLS1-PCG". If the direct search directions on all but the coarsest level are computed by the multigrid Algorithm 2 (MG) for solving the linear equations, we denote these methods by "FMLS0-MG" and "FMLS1-MG", respectively. (See Table 6.1 for a summary of these methods). We construct the matrices A_h for the various levels so that Assumption 2.2 is satisfied for the linear multigrid method. We also give results obtained using Newton's Method with PCG on the finest level for a comparison.

In our test problems the grid spacing is set to 2^{-3} at the coarsest level and to 2^{-8} at the finest level. This gives a 257×257 grid on the finest level. How we computed the gradient and the Hessian is explained in the Appendix. The initial point in Algorithm 4 is taken to be the zero vector. For the multigrid Algorithm 3, we set

$$\kappa = 10^{-4}, \quad \epsilon_h = 10^{-4}, \quad K = 20, \quad \rho_1 = 0.01, \quad \rho_2 = 0.2.$$

TABLE 6.2
Summary of computational costs for Problem 6.3

	FMLS0-PCG						FMLS1-PCG					
n_h	9^2	17^2	33^2	65^2	129^2	257^2	9^2	17^2	33^2	65^2	129^2	257^2
nls	2	4	2	1	1	1	0	1	1	1	1	1
nge	4	8	4	2	2	2	0	2	2	2	2	2
nhe	$2^\dagger$	4	2	4	5	6	0	2	3	4	5	6
$\|g_N^*\|_2$	6.197382e-07						6.196731e-07					
CPU	6.787156						6.736728					

	FMLS0-MG						FMLS1-MG					
n_h	9^2	17^2	33^2	65^2	129^2	257^2	9^2	17^2	33^2	65^2	129^2	257^2
nls	3	6	3	1	1	1	1	3	2	1	1	1
nge	6	11	5	2	2	2	2	5	3	2	2	2
nvc	$3^\dagger$	6	4	2	2	2	$1^\dagger$	4	4	2	2	2
$\|g_N^*\|_2$	1.028328e-05						1.028328e-05					
CPU	3.565317						3.328619					

Newton					
n_h	nls	nge	nhe	$\|g_N^*\|_2$	CPU
257^2	1	2	20	1.074113e-06	8.10042

Note that these parameters could also be set adaptively for each level.

The algorithm described above has been implemented in MATALAB (Release 7.3.0); the line search method is adapted from the code "DCSRCH" [28] with an initial step size of one. All experiments were run on a Dell Precision 670 workstation with an Intel xeon(TM) 3.4GHZ CPU and 6GB of RAM.

6.2.1. Nonlinear PDE 1. We study the nonlinear PDE [2]

$$(6.3) \qquad \begin{aligned} -\Delta u - u^2 &= f(x) && \text{in} \quad \Omega, \\ u &= 0 && \text{on} \quad \partial\Omega, \end{aligned}$$

where $f(x) = x^6$ and $\Omega = [0,1] \times [0,1]$. The corresponding variational problem is

$$\min \mathcal{F}(u) = \int_\Omega \frac{1}{2}|\nabla u|^2 - \frac{1}{3}u^3 - f\ u\ dx.$$

The stopping tolerance for PCG and MG is 10^{-3}.

In Tables 6.2-6.5, we summarize the computational costs of the various methods on this and the other problems. Specifically, n_h denotes the number of variables on level "h"; "nls", and "nge" denote the total number of line searches and the total number of gradient evaluations at that level, respectively; "nhe" denotes the total number of Hessian evaluations on the coarsest level (marked with †) and the total number of matrix-vector products on other levels if a PCG is used; for all levels other than the coarsest, "nvc" denotes the total number of multigrid cycles on that level; on the coarsest level, "nvc" means the total number of Hessian evaluations (marked with †). Since the total number of objective function evaluations is equal to "nge" in our implementation, we do not include it in the table. Both PCG methods work well for this problem because on the finer levels, only one Newton's step is needed.

From Table 6.2, we can see that the counts "nls", "nge" and "nhe" at the coarse levels for method "FMLS0-PCG" are larger than those at the finer levels. This shows that most of the function value, gradient and Hessian evaluations occur mainly on the coarser levels. Table 6.2 also gives the total CPU time measured in seconds and the accuracy attained, which is measured by the 2-norm $\|g_N^*\|_2$ of the gradient at

FIG. 6.1. *Iteration history for Problem 6.3. Left: "FMLS0-PCG"; Right: "FMLS1-PCG"*
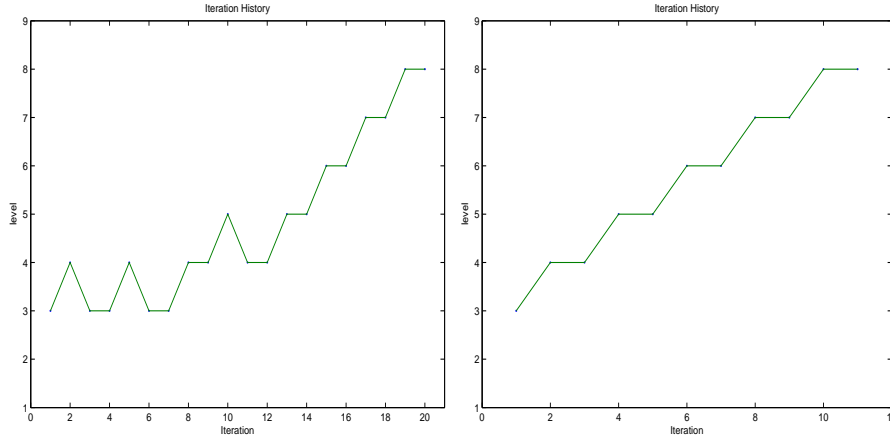
the final iteration. Both PCG methods achieve very good accuracy. It is interesting to note that the difference between CPU times is very small. This confirmed the property of multilevel methods that their cost does not increase very much if there are more function, gradient and Hessian evaluations at the coarser levels. We can observe similar behavior for method "FMLS0-MG" and method "FMLS1-MG". All of these methods only take one Newton step at the finest level.

Table 6.2 shows that different inexact solvers for the direct search direction computation lead to quite different results. MG performs better than PCG in this example in terms of CPU time. Although they are not that different in terms of "nls" and "nge", they are different in terms of the operations involving the Hessian (however, it is hard to compare the two quantities "nhe" and "nvc" directly).

To illustrate the multilevel behavior of methods "FMLS0-PCG" and "FMLS1-PCG" we plot the level versus iteration history for them in Figure 6.1, which is a simplified version of Figure 5.1. We can see that a recursive step is not always performed; it is only performed if it is necessary. Although "nls" indicates that there are iterations on the coarsest level, Figure 6.1 shows that the multigrid method never returns to the coarsest level once the minimization procedure is running on the finer levels.

6.2.2. Nonlinear PDE 2. We study the nonlinear PDE [20]:

$$
\begin{aligned}
-\Delta u + \lambda u e^u = f & \quad \text{in} \quad \Omega, \\
u = 0 & \quad \text{on} \quad \partial\Omega,
\end{aligned}
$$
(6.4)

where

$$
f = \left(9\pi^2 + \lambda e^{((x^2 - x^3)\sin(3\pi y))}(x^2 - x^3) + 6x - 2\right)\sin(3\pi y),
$$

$\lambda = 10$, $\Omega = [0,1] \times [0,1]$ and the exact solution is $u = (x^2 - x^3)\sin(3\pi y)$. The corresponding variational problem is

$$
\min \mathcal{F}(u) = \int_\Omega \frac{1}{2}|\nabla u|^2 - \lambda(ue^u - e^u) \, dx.
$$

TABLE 6.3
Summary of computational costs for Problem 6.4

	FMLS0-PCG						FMLS1-PCG					
n_h	9^2	17^2	33^2	65^2	129^2	257^2	9^2	17^2	33^2	65^2	129^2	257^2
nls	5	6	3	2	1	1	1	3	2	1	1	1
nge	9	12	6	4	2	2	2	5	3	2	2	2
nhe	$5^\dagger$	4	5	2	3	3	$1^\dagger$	4	4	2	3	3
$\|g_N^*\|_2$	1.303290e-05						1.362962e-05					
CPU	6.211418						6.172631					
	FMLS0-MG						FMLS1-MG					
n_h	9^2	17^2	33^2	65^2	129^2	257^2	9^2	17^2	33^2	65^2	129^2	257^2
nls	5	8	5	3	1	1	1	5	3	2	1	1
nge	9	15	9	5	2	2	2	8	5	3	2	2
nvc	$5^\dagger$	9	6	4	2	2	$1^\dagger$	9	4	4	2	2
$\|g_N^*\|_2$	2.898190e-05						2.898191e-05					
CPU	3.091660						2.855589					

Newton					
n_h	nls	nge	nhe	$\|g_N^*\|_2$	CPU
257^2	2	3	32	3.133664e-07	14.39141

FIG. 6.2. *Iteration history for Problem 6.4. Left: "FMLS0-PCG"; Right: "FMLS1-PCG"*
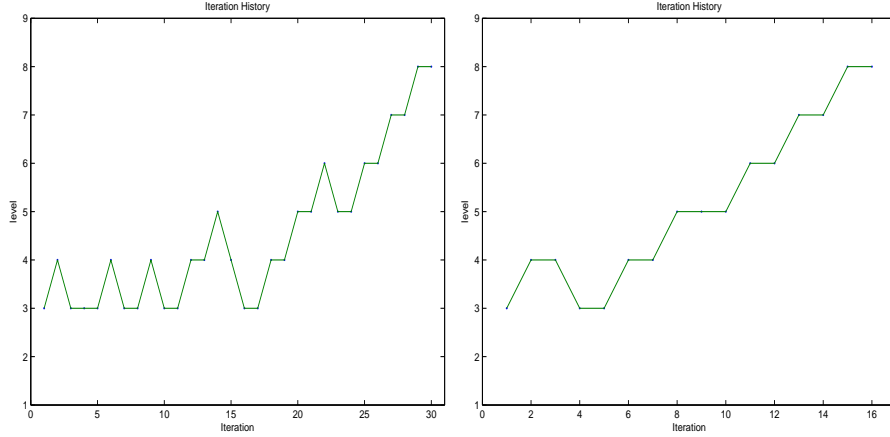
The stopping tolerance for PCG and MG is 10^{-3}. From Table 6.3, both "FMLS0-PCG" and "FMLS1-PCG" work well because both methods only need one Newton step at the finest level. But "FMLS0-PCG" spends more time at the coarse level especially at the beginning looking for a good initial point. Both methods achieve very good accuracy and the difference between CPU times is also very small. From Figure 6.2, we see that the multigrid method also never returns to the coarsest level once the minimization procedure is running on levels close to the finest level. We can observe similar behaviors for method "FMLS0-MG" and method "FMLS1-MG". MG performs better than PCG in this example in terms of CPU time.

6.2.3. Minimal Surface Problem. Consider the minimal surface problem

$$
(6.5) \quad
\begin{aligned}
\min \quad & f(u) = \int_\Omega \sqrt{1 + \|\nabla u(x)\|^2}\ \mathrm{dx} \\
\text{s.t.} \quad & u(x) \in K = \{u \in H^1(\Omega) : u(x) = u_\Omega(x) \text{ for } x \in \partial\Omega\},
\end{aligned}
$$

TABLE 6.4
Summary of computation costs for Problem 6.5 with boundary u^1_Ω

	FMLS0-PCG						FMLS1-PCG					
n_h	9^2	17^2	33^2	65^2	129^2	257^2	9^2	17^2	33^2	65^2	129^2	257^2
nls	8	11	3	2	1	1	2	5	1	1	1	1
nge	15	23	6	4	2	2	4	10	2	2	2	2
nhe	$8^\dagger$	9	5	4	4	3	$2^\dagger$	7	3	3	4	3
$\|g^*_N\|_2$	2.261061e-06						1.999010e-06					
CPU	7.068329						6.931453					
	FMLS0-MG						FMLS1-MG					
n_h	9^2	17^2	33^2	65^2	129^2	257^2	9^2	17^2	33^2	65^2	129^2	257^2
nls	8	13	5	3	1	1	2	6	3	2	1	1
nge	15	26	9	5	2	2	4	12	5	3	2	2
nvc	$8^\dagger$	18	6	4	2	2	$2^\dagger$	17	4	4	2	2
$\|g^*_N\|_2$	1.811500e-05						1.811409e-05					
CPU	3.982122						3.809665					

Newton					
n_h	nls	nge	nhe	$\|g^*_N\|_2$	CPU
257^2	5	16	73	2.136139e-06	38.81010

where $\Omega = [0,1] \times [0,1]$. We will test two sets of boundary data [17, 29]:

$$\text{(Surf1):} \quad u^1_\Omega(x) = \begin{cases} x(1-x), & y = 0, 1, \\ 0, & otherwise, \end{cases}$$

$$\text{(Surf2):} \quad u^2_\Omega(x) = \begin{cases} -sin(2\pi y), & x = 0, \\ sin(2\pi y), & x = 1, \\ sin(2\pi x), & y = 0, \\ -sin(2\pi x), & y = 1. \end{cases}$$

The stopping tolerance for PCG and MG is 10^{-3} for boundary condition $u^1_\Omega(x)$ while it is $10^{-1}\|g_{h,k}\|$ for boundary condition $u^2_\Omega(x)$.

From Table 6.4, both "FMLS0-PCG" and "FMLS1-PCG" work well for the boundary condition $u^1_\Omega(x)$. But "FMLS0-PCG" spends more time at the coarser levels, especially at the beginning looking for a good initial point. Both methods achieve very good accuracy and the difference between CPU times is also very small. From Figure 6.3, we see that the multigrid method also never returns to the coarsest level once the minimization procedure is running on levels close to the finest level. We can observe similar behaviors for method "FMLS0-MG" and method "FMLS1-MG". MG performs better than PCG in this example in terms of CPU time.

From Table 6.5, we see that "FMLS0-PCG" and "FMLS1-PCG" also work well, but they require recursive steps quite often for the boundary condition $u^2_\Omega(x)$. The difference between the achieved accuracy is small but the difference between CPU time is not. They behave like two level methods as we can see in Figure 6.4. This fact is of great advantage because the bounds estimated in the convergence analysis are much smaller in this case. Therefore, we can still anticipate a fast convergence rate. However, "FMLS0-MG" and "FMLS1-MG" do not work well in this case. One reason perhaps is that the linear multigrid method we implemented can not handle the nonlinearity well in this case.

7. Discussion. The multigrid Algorithm 3 provides an automatic way to alternate between recursive steps and direct search steps. Usually, different inexact solvers

TABLE 6.5
Summary of computation costs for Problem 6.5 with boundary u_Ω^2

	FMLS0-PCG						FMLS1-PCG					
n_h	9^2	17^2	33^2	65^2	129^2	257^2	9^2	17^2	33^2	65^2	129^2	257^2
nls	16	32	23	16	12	7	3	9	6	5	4	3
nge	31	62	46	33	25	17	5	15	12	8	7	7
nhe	$16^\dagger$	47	48	54	98	26	$3^\dagger$	20	13	23	31	24
$\|g_N^*\|_2$	1.081637e-05						3.840665e-05					
CPU	29.53286						20.78162					
	FMLS0-MG						FMLS1-MG					
n_h	9^2	17^2	33^2	65^2	129^2	257^2	9^2	17^2	33^2	65^2	129^2	257^2
nls	17	37	29	20	18	19	3	10	9	9	10	13
nge	33	71	55	41	40	41	5	18	16	19	22	31
nvc	$17^\dagger$	56	55	72	115	103	$3^\dagger$	25	27	55	83	78
$\|g_N^*\|_2$	1.838892e-05						5.439764e-05					
CPU	72.68670						54.00278					

Newton					
n_h	nls	nge	nhe	$\|g_N^*\|_2$	CPU
257^2	6	12	91	4.671913e-05	47.45587

FIG. 6.3. *Iteration history for Problem 6.5 with boundary u_Ω^1. Left: "FMLS0-PCG"; Right: "FMLS1-PCG"*
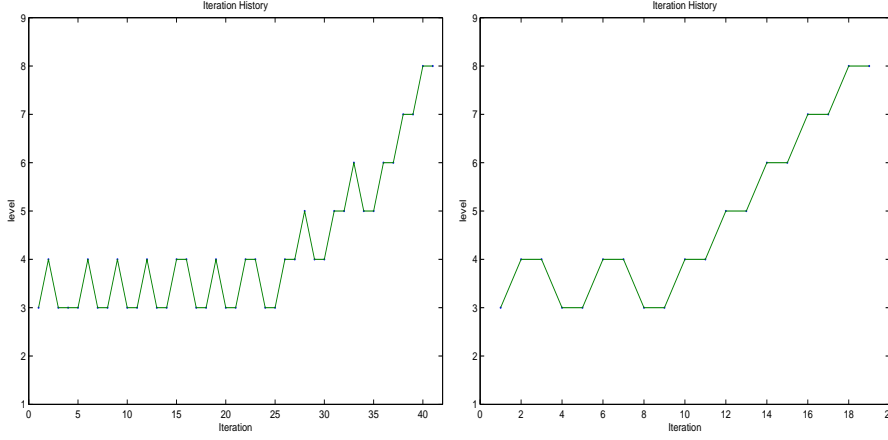

for the direct step leads to quite different results. Our algorithm can be viewed as a combination of the global linearization method and the FAS scheme if the direct search step is solved by the linear multigrid method. Applying the multigrid framework from the point view of optimization provides us with new opportunities for designing efficient algorithms.

Appendix. Computing Derivatives Efficiently . For the sake of simplicity, we only consider a discretization of the objective functional (6.1) using forward finite differences

$$(A.1) \qquad F(u) = \omega_h^x \omega_h^y \sum_{i=0}^{n_h^x-1} \sum_{j=0}^{n_h^y-1} \mathcal{L}(\delta_x^+ u_{i,j}, \delta_y^+ u_{i,j}, u_{i,j}) = \omega_h^x \omega_h^y \sum_{i=0}^{n_h^x-1} \sum_{j=0}^{n_h^y-1} L_{i,j}^+,$$

FIG. 6.4. *Iteration history for Problem 6.5 with boundary u_Ω^2. Left: "FMLS0-PCG"; Right: "FMLS1-PCG"*
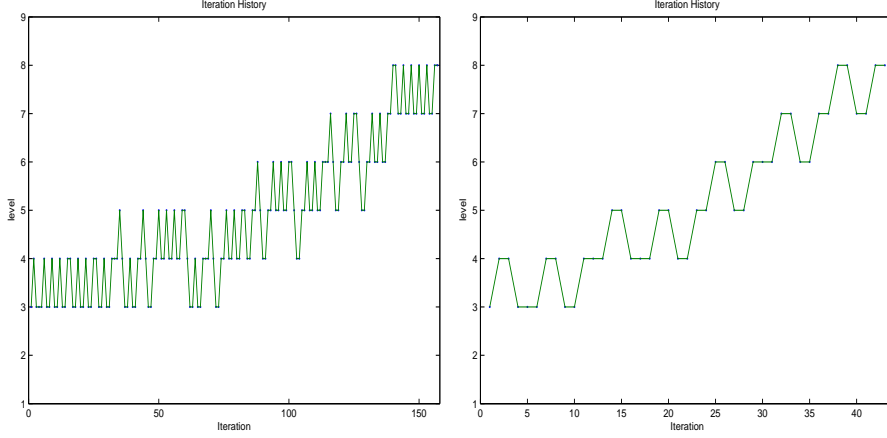
where $L_{i,j}^+ = L(\delta_x^+ u_{i,j}, \delta_y^+ u_{i,j}, u_{i,j})$. Then, the gradient of $F(u)$ with respect to interior points are

$$
\begin{aligned}
(A.2) \quad & \nabla F(1 : n_h^x - 1, 1 : n_h^y - 1) \\
& = \left(g_x^+(0 : n_h^x - 2, 1 : n_h^y - 1) - g_x^+(1 : n_h^x - 1, 1 : n_h^y - 1) \right) \\
& \quad + \left(g_y^+(1 : n_h^x - 1, 0 : n_h^y - 2) - g_y^+(1 : n_h^x - 1, 1 : n_h^y - 1) \right) \\
& \quad + g_u(1 : n_h^x - 1, 1 : n_h^y - 1),
\end{aligned}
$$

where

$$
g_x^+ = \omega_h^y \frac{\partial L(\delta_x^+ u, \delta_y^+ u, u)}{\partial(\delta_x^+ u)}, \ \ g_y^+ = \omega_h^x \frac{\partial L(\delta_x^+ u, \delta_y^+ u, u)}{\partial(\delta_y^+ u)}, \ \ g_u = \omega_h^x \omega_h^y \frac{\partial L(\delta_x^+ u, \delta_y^+ u, u)}{\partial u}.
$$

Similarly, we collect partial derivatives $\frac{\partial^2 L_{i,j}^+}{\partial(\delta_x^+ u_{i,j})^2}$, $\frac{\partial^2 L_{i,j}^+}{\partial(\delta_x^+ u_{i,j})\partial(\delta_y^+ u_{i,j})}$, $\frac{\partial^2 L_{i,j}^+}{\partial(\delta_y^+ u_{i,j})^2}$ and $\frac{\partial^2 L_{i,j}^+}{\partial u_{i,j}^2}$ into the matrices

$$
\begin{aligned}
G_{xx}^+ &= \frac{\omega_h^y}{\omega_h^x} \frac{\partial^2 L(\delta_x^+ u, \delta_y^+ u, u)}{\partial(\delta_x^+ u)^2}, & G_{xy}^+ &= \frac{\partial^2 L(\delta_x^+ u, \delta_y^+ u, u)}{\partial(\delta_x^+ u)\partial(\delta_y^+ u)}, \\
G_{yy}^+ &= \frac{\omega_h^x}{\omega_h^y} \frac{\partial^2 L(\delta_x^+ u, \delta_y^+ u, u)}{\partial(\delta_y^+ u)^2}, & G_{uu} &= \omega_h^x \omega_h^y \frac{\partial^2 L(\delta_x^+ u, \delta_y^+ u, u)}{\partial u_{i,j}^2}.
\end{aligned}
$$

Then, by using the notation in the Matlab language and using the function "spdiags" which creates sparse matrix, the lower triangular part G of the Hessian $\nabla^2 F(u)$ for the interior points can be computed as follows.

ALGORITHM 5. *Compute the lower triangular part of Hessian $\nabla^2 F$*

$$G_1 = (G_{xx}^+(0:n_h^x-2,1:n_h^y-1) + G_{xx}^+(1:n_h^x-1,1:n_h^y-1))$$
$$+(G_{yy}^+(1:n_h^x-1,0:n_h^y-2) + G_{yy}^+(1:n_h^x-1,1:n_h^y-1))$$
$$+2G_{xy}^+(1:n_h^x-1,1:n_h^y-1) + G_{uu}(1:n_h^x-1,1:n_h^y-1);$$
$$G_2 = -G_{xx}^+(1:n_h^x-1,1:n_h^y-1) - G_{xy}^+(1:n_h^x-1,1:n_h^y-1);$$
$$G_3 = G_{xy}^+(1:n_h^x-1,1:n_h^y-1);$$
$$G_4 = -G_{xy}^+(1:n_h^x-1,1:n_h^y-1) - G_{yy}^+(1:n_h^x-1,1:n_h^y-1);$$
$$G_1 = G_1(:); \quad G_2(n_h^x-1,:) = 0; \quad G_2 = G_2(:);$$
$$G_4 = G_4(:); \quad G_3(n_h^x-1,:) = 0; \quad G_3 = G_3(:); \quad G_3 = [0; G_3(1:end-1)]$$
$$G = spdiags([G_4, \ G_3, \ G_2, \ G_1], [-(n_h^x-2), \ -(n_h^x-3), \ -1, \ 0], (n_h^x-2)^2, (n_h^y-2)^2);$$

Matrix-vector products involving the Hessian can be computed as

$$[\nabla^2 F(u)] \, v = \omega_h^x \omega_h^y \sum_{i=0}^{n_h^x-1} \sum_{j=0}^{n_h^y-1} [\nabla^2 L_{i,j}^+] \, v.$$

REMARK A.1. *For many PDE constrained problems, one can eliminate the state variable and transform the problem into an unconstrained one. The gradient and matrix-vector products involving the Hessian can then be computed by the adjoint method. When the objective function is in the form of least squares or it involves a Total Variation regularization term, one can also approximate the Hessian by a Gauss-Newton approximation.*

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 (electronic). Tenth Copper Mountain Conference on Multigrid Methods (Copper Mountain, CO, 2001).

[2] S. BALAY, K. BUSCHELMAN, V. EIJKHOUT, W. D. GROPP, D. KAUSHIK, M. G. KNEPLEY, L. C. MCINNES, B. F. SMITH, AND H. ZHANG, *PETSc users manual*, Tech. Rep. ANL-95/11 - Revision 2.1.5, Argonne National Laboratory, 2004.

[3] M. BENZI, E. HABER, AND L. HANSON, *Multilevel algorithms for large-scale interior point methods in bound constrained optimization*, tech. rep., 2006.

[4] A. BORZÌ, *Multilevel methods in optimization with partial differential equations*. Lecture notes, Insitut für Mathematik und Wissenschaftliches Rechnen, Karl-Franzens-Universität Graz.

[5] ———, *On the convergence of the mg/opt method*, in Proceedings GAMM Annual Meeting, no. 5, 2005, pp. 735–736.

[6] A. BORZÌ AND K. KUNISCH, *A multigrid scheme for elliptic constrained optimal control problems*, Comput. Optim. Appl., 31 (2005), pp. 309–333.

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

[8] S. BOYD AND L. VANDENBERGHE, *Convex optimization*, Cambridge University Press, Cambridge, 2004.

[9] J. H. BRAMBLE, *Multigrid methods*, vol. 294 of Pitman Research Notes in Mathematics Series, Longman Scientific & Technical, Harlow, 1993.

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

[11] ———, *Multigrid techniques: 1984 guide with applications to fluid dynamics*, vol. 85 of GMD-Studien [GMD Studies], Gesellschaft für Mathematik und Datenverarbeitung mbH, St. Augustin, 1984.

[12] W. L. BRIGGS, V. E. HENSON, AND S. F. MCCORMICK, *A multigrid tutorial*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, second ed., 2000.

[13] E. D. Dolan, J. J. Moré, and T. S. Munson, *Benchmarking optimization software with cops 3.0*, tech. rep., Mathematics and Computer Science Division, Argonne National Laboratory, February 2004.

[14] T. Dreyer, B. Maar, and V. Schulz, *Multigrid optimization in applications*, J. Comput. Appl. Math., 120 (2000), pp. 67–84. SQP-based direct discretization methods for practical optimal control problems.

[15] G. H. Golub and C. F. Van Loan, *Matrix computations*, Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, Baltimore, MD, third ed., 1996.

[16] S. Gratton, A. Sartenaer, and P. Toint, *Recursive trust-region methods for multilevel nonlinear optimization (part i): Global convergence and complexity*, tech. rep., Dept of Mathematics, FUNDP, Namur (B), 2004.

[17] ———, *Numerical experience with a recursive trust-region method for multilevel nonlinear optimization*, tech. rep., Dept of Mathematics, FUNDP, Namur (B), 2006.

[18] ———, *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.

[19] W. Hackbusch, *Multigrid methods and applications*, vol. 4 of Springer Series in Computational Mathematics, Springer-Verlag, Berlin, 1985.

[20] V. E. Henson, *Multigrid methods nonlinear problems: an overview*, in Computational Imaging. Edited by Bouman, Charles A.; Stevenson, Robert L. Proceedings of the SPIE, Volume 5016, pp. 36-48 (2003)., C. A. Bouman and R. L. Stevenson, eds., June 2003, pp. 36–48.

[21] 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.

[22] ———, *Factors affecting the performance of optimization-based multigrid methods*, in Multiscale optimization methods and applications, vol. 82 of Nonconvex Optim. Appl., Springer, New York, 2006, pp. 151–172.

[23] 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 (electronic).

[24] 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 (electronic).

[25] S. F. McCormick, ed., *Multigrid methods*, vol. 3 of Frontiers in Applied Mathematics, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1987.

[26] ———, *Projection multilevel methods for quasi-linear PDEs: V-cycle theory*, Multiscale Model. Simul., 4 (2005), pp. 1339–1348 (electronic).

[27] H. D. Mittelmann, *Decision tree for optimization software.* http://plato.asu.edu/guide.html.

[28] J. J. Moré and D. J. Thuente, *Line search algorithms with guaranteed sufficient decrease*, ACM Trans. Math. Software, 20 (1994), pp. 286–307.

[29] S. G. Nash, *A multigrid approach to discretized optimization problems*, Optim. Methods Softw., 14 (2000), pp. 99–116. International Conference on Nonlinear Programming and Variational Inequalities (Hong Kong, 1998).

[30] J. Nocedal and S. J. Wright, *Numerical optimization*, Springer Series in Operations Research and Financial Engineering, Springer, New York, second ed., 2006.

[31] A. Reusken, *Steplength optimization and linear multigrid methods*, Numer. Math., 58 (1991), pp. 819–838.

[32] W. Sun and Y.-X. Yuan, *Optimization theory and methods*, vol. 1 of Springer Optimization and Its Applications, Springer, New York, 2006. Nonlinear programming.

[33] 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 (electronic).

[34] U. Trottenberg, C. W. Oosterlee, and A. Schüller, *Multigrid*, Academic Press Inc., San Diego, CA, 2001. With contributions by A. Brandt, P. Oswald and K. Stüben.

[35] P. Wesseling, *An introduction to multigrid methods*, Pure and Applied Mathematics (New York), John Wiley & Sons Ltd., Chichester, 1992.

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

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

[38] Y. Yuan, *On the truncated conjugate gradient method*, Math. Program., 87 (2000), pp. 561–573.