# Technical Report

TR-2006-002

**Multilevel Algorithms for Large-Scale Interior Point Methods in Bound Constrained Optimization**

by

Michele Benzi, Eldad Haber, Lauren Hanson

Mathematics and Computer Science

**EMORY UNIVERSITY**

# MULTILEVEL ALGORITHMS FOR LARGE-SCALE INTERIOR POINT METHODS IN BOUND CONSTRAINED OPTIMIZATION

MICHELE BENZI*, ELDAD HABER†, AND LAUREN R. HANSON‡

**Abstract.** We develop and compare multilevel algorithms for solving bound constrained nonlinear variational problems via interior point methods. Several equivalent formulations of the linear systems arising at each iteration of the interior point method are compared from the point of view of conditioning and iterative solution. Furthermore, we show how a multilevel continuation strategy can be used to obtain good initial guesses ("hot starts") for each nonlinear iteration. A minimal surface problem is used to illustrate the various approaches.

**1. Introduction.** In this work we consider the solution of discrete optimization problems of the form

$$\min_u \quad J(u) \tag{1.1a}$$

$$\text{s.t.} \quad b \le u\,, \tag{1.1b}$$

where $u, b \in \mathbb{R}^n$. The objective function $J$ is assumed to be a discretization of a (generally nonlinear) functional $\mathcal{J}$ involving a differential operator defined on some function space. The optimization problem is assumed to be discretized on a "sufficiently fine" grid; the discrete problem (1.1) is the "fine grid problem," and we assume it is possible to generate coarser problems by some coarsening process (either rediscretization or Galerkin-type coarsening). For simplicity we assume that the underlying grid used to discretize the problem is uniform with spacing $h$ and that the discretization is of order $h^p$, for some $p \ge 1$; the uniformity assumption, however, is not essential and can be relaxed. As a motivation and illustration of this work, we consider the following model problem.

EXAMPLE 1. **Minimal surface problem**
*Let $\Omega$ be a bounded, open subset of $\mathbb{R}^2$, and let $b$ be a given function defined on $\Omega$. The minimal surface problem reads:*

$$\min_u \quad \mathcal{J}(u) = \int_\Omega \sqrt{(\operatorname{grad} u)^2 + 1}\; d\boldsymbol{x} \tag{1.2a}$$

$$\text{s.t.} \quad b \le u \tag{1.2b}$$

$$u|_{\partial\Omega} = 0. \tag{1.2c}$$

*Upon consistent discretization we obtain*

$$\min_{u_h} \quad J(u_h) = e^\top \left( \sqrt{A_1(D_1 u)^2 + A_2(D_2 u)^2 + 1} \right) \tag{1.3a}$$

$$\text{s.t.} \quad u_h \ge b_h\,, \tag{1.3b}$$

*Department of Mathematics and Computer Science, Emory University, Atlanta, Georgia 30322, USA (benzi@mathcs.emory.edu). The work of this author was supported in part by the National Science Foundation grant DMS-0511336.

†Department of Mathematics and Computer Science, Emory University, Atlanta, Georgia 30322, USA (haber@mathcs.emory.edu). The work of this author was supported in part by DOE under grant DEFG02- 05ER25696, and by NSF under grant CCF-0427094.

‡Department of Mathematics and Computer Science, Emory University, Atlanta, Georgia 30322, USA (lrhanso@mathcs.emory.edu).

*where $u_h$ is the cell-centered discretization of $u$, $D_j$ are central difference matrices that approximate the derivatives in the $j^{th}$ direction on the cell edges, $A_j$ is the averaging matrix that averages edge quantities into cell centers, and $e = [1, \ldots, 1]^\top$.*

While we focus here on the minimal surface problem, our goal is more general. Namely, we aim to develop efficient methods for nonlinear variational problems with bound constraints. For instance, functionals of the form

$$\mathcal{J}(u) = \mathcal{Q}(u) + \int_\Omega \sqrt{(\operatorname{grad} u)^2 + \varepsilon}\; d\boldsymbol{x}\,,$$

where $\mathcal{Q}$ is quadratic with sparse Hessian and $\varepsilon > 0$, are also of interest (see [1]). Problems of this kind arise in image processing (TV-based denoising). Also, our methods are not limited to two-dimensional problems. In the three-dimensional case, the discrete optimization problem can easily have several millions of unknowns and constraints. Standard optimization software is often unsuitable for such large-scale problems, and in many cases fails to produce the solution. Hence, there is a need for new optimization algorithms and software for the solution of such problems. Similar to many other large-scale problems which evolve from PDEs, it is difficult to develop a *general* toolbox for the solution of all problems. Nevertheless, it is possible to group some applications and develop numerical algorithms that can be very effective on a particular set of problems. Here, we consider problems with the following characteristics:

- A1 The Hessian $J_{uu}$ (or its approximations) can be thought of as a discretization of an elliptic PDE.
- A2 The constraints are simple bounds on the solution.

Such problems arise in many areas of science and engineering; see [11]. Assumptions A1-A2 above imply that the main contribution to the coefficient matrix of the linear system that is solved at each iteration of an interior point method comes from the objective function, and is to be found in the (1,1) block. Furthermore, the fact that $J_{uu}$ is a discretization of an elliptic operator suggests that we should be able to develop efficient multigrid methods for the unconstrained problem $J(u) = \min$.

Remark 1.1. *Throughout the paper we make the assumption that $J_{uu}$ (or its approximation used in the actual code) is symmetric and positive definite (SPD).*

Optimization techniques for the solution of problems with inequality constraints can be roughly divided into two main groups; active set methods and interior point methods (IP) [16, 21]. Currently, there is a lack of scalable iterative linear algebra techniques for IP methods. As a result, using IP methods can be inefficient. However, it is well known that the rate of convergence of IP methods is typically much better than that of active set methods [16, Chapter 14]. This motivates us to explore iterative methods and preconditioners for IP methods. In particular, we would like to use scalable techniques from numerical PDEs, such as multigrid and multilevel methods.

In the rest of this paper we discuss a multilevel approach for the solution of problem (1.1a)-(1.1b). In Section 2 we briefly review an inexact Newton method for the solution of the problem on a single grid. In Section 3 we describe multigrid and iterative solvers for the solution of the linear problems that arise at each outer iteration. In Section 4 we describe how to use grid sequencing with interior point methods. In particular, we describe how to adapt the log barrier parameter to the grid. In Section 5 we perform numerical experiments that compare the effectiveness of our methods. Finally, in Section 6 we summarize our findings.

**2. An Inexact Newton framework.** To solve the problem (1.1) we consider interior point methods with a log barrier term. We follow the work of [2] for the application of inexact Newton methods within interior point algorithms.

We begin by posing the problem as an equality-constrained optimization problem with a logarithmic barrier term:

$$\begin{aligned} \min \quad & J(u) - \mu e^\top \log(z)\,, \\ \text{s.t.} \quad & u - z = b\,, \end{aligned} \tag{2.1}$$

where $z$ is a slack variable, $\log(z)$ is the vector whose $i$th component is $\log(z_i)$, $\mu$ is the barrier parameter and $e = [1, \ldots, 1]^\top$.

Consider the Lagrangian

$$\mathcal{L}(u, z, \lambda) = J(u) - \mu e^\top \log(z) - \lambda^\top (u - z - b)\,,$$

where $\lambda$ is a vector of Lagrange multipliers. The Euler-Lagrange equations associated with the Lagrangian are:

$$r_1 := \mathcal{J}_u - \lambda = 0\,, \tag{2.2a}$$

$$r_2 := u - z - b = 0\,, \tag{2.2b}$$

$$r_3 := z \odot \lambda - \mu e = 0\,, \tag{2.2c}$$

where $\odot$ denotes the componentwise (Hadamard) product. Note that if we set the barrier parameter $\mu$ equal to zero we obtain the classical KKT conditions for the constrained optimization problem; see [16].

Using a Newton-type method for the solution of the system (2.2) we obtain a sequence of linear systems of the form

$$\begin{pmatrix} J_{uu} & -I & \\ -I & & I \\ & Z & \Lambda \end{pmatrix} \begin{pmatrix} \delta u \\ \delta\lambda \\ \delta z \end{pmatrix} = - \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix}\,, \tag{2.3}$$

where $Z = \operatorname{diag}(z)$ and $\Lambda = \operatorname{diag}(\lambda)$. Here $J_{uu}$ stands for the Hessian of $J$ or some approximation of it. In a truncated Newton method, each system is solved approximately rather than "exactly." The (nonlinear) solution is then updated using some globalization technique (e.g., a line search) so as to ensure positivity of $\lambda$ and $z$, and the next (outer) iteration is carried out. The overall procedure is briefly described in Algorithm 1 below. There are two key components that need to be

**Algorithm 1** Inexact-Newton method for constrained optimization:
$\boldsymbol{u} \leftarrow \text{INIC}(\boldsymbol{u}_0)$;

initialize $\{u, \lambda, z\} \leftarrow \{u_0, \lambda_0, z_0\}$ and the barrier parameter $\mu_0$
**while** true **do**
  compute $r_1, r_2, r_3$;
  approximately solve (2.3) with tolerance tol $\leq \eta \lambda^\top z$
  Use a weak line search to accept/reject step
  test for termination
  evaluate $\mu_{k+1}$
**end while**

addressed in order to make this algorithm an effective one. First, a good linear solver

is needed and second, a good strategy to evaluate $\mu$ at each iteration is required. These two components are far from trivial. The linear system is highly ill conditioned, nonsymmetric and indefinite, and the choice of $\mu$ has been the topic of several recent papers; see, e.g., [4, 15]. In the next section we discuss the solution of the linear system. Second, in Section 4 we show that it is possible to use a multilevel strategy where we solve increasingly finer problems. In other words, we successively refine the mesh as the outer IP iteration progresses. Unlike the inexact Newton method described above, we keep the log barrier parameter $\mu$ fixed for each optimization problem, and decrease $\mu$ in tandem with the mesh size.

**3. Multigrid methods for the linear system.** In this section we examine several alternative approaches to solving the linear systems arising at each step of the IP method.

**3.1. Alternative linear system formulations.** It is possible to reformulate the linear system (2.3) to be solved at each IP iteration in several ways. While mathematically equivalent, these forms are computationally quite different. Such formulations have been proposed before in the literature; here, however, we pay special attention to those problem features that are peculiar to PDE-type problems. First, it is obvious that we can simply solve the original system (2.3). The main advantage of this system is that as long as strict complementarity holds, vanishing $z_i$ and $\lambda_i$ do not have too much of an adverse effect on its conditioning. However, the disadvantage of this system is that it is large ($3n$-by-$3n$), highly coupled, and nonsymmetric. The following simple Lemma shows that the last difficulty can be circumvented, at least in principle, by an appropriate diagonal scaling.

LEMMA 3.1. *The coefficient matrix in* (2.3) *is similar to a real symmetric indefinite matrix, hence it has real spectrum. There are exactly* $2n$ *positive and* $n$ *negative eigenvalues.*

*Proof.* It is immediate to verify that

$$\begin{pmatrix} I & & \\ & I & \\ & & Z^{-\frac{1}{2}} \end{pmatrix} \begin{pmatrix} J_{uu} & -I & \\ -I & & I \\ & Z & \Lambda \end{pmatrix} \begin{pmatrix} I & & \\ & I & \\ & & Z^{\frac{1}{2}} \end{pmatrix} = \begin{pmatrix} J_{uu} & -I & \\ -I & & Z^{\frac{1}{2}} \\ & Z^{\frac{1}{2}} & \Lambda \end{pmatrix},$$

where we have used the fact that $Z^{-\frac{1}{2}}\Lambda Z^{\frac{1}{2}} = \Lambda$, since $Z$ and $\Lambda$ are both diagonal. Furthermore,

$$\begin{pmatrix} J_{uu} & -I & \\ -I & & Z^{\frac{1}{2}} \\ & Z^{\frac{1}{2}} & \Lambda \end{pmatrix} = \begin{pmatrix} I & & \\ & I & \\ -Z^{\frac{1}{2}} & -Z^{\frac{1}{2}}J_{uu} & I \end{pmatrix} \begin{pmatrix} J_{uu} & -I & \\ -I & & \\ & & S \end{pmatrix} \begin{pmatrix} I & & -Z^{\frac{1}{2}} \\ & I & -J_{uu}Z^{\frac{1}{2}} \\ & & I \end{pmatrix},$$

where $S = \Lambda + Z^{\frac{1}{2}}J_{uu}Z^{\frac{1}{2}}$. Hence, the original matrix is similar to a symmetric matrix which is congruent to a block diagonal matrix with blocks

$$\begin{pmatrix} J_{uu} & -I \\ -I & \end{pmatrix} \qquad \text{and} \qquad S = \Lambda + Z^{\frac{1}{2}}J_{uu}Z^{\frac{1}{2}}.$$

The first matrix is indefinite with $n$ positive and $n$ negative eigenvalues, whereas the second one is SPD, hence it has $n$ positive eigenvalues. It follows that the original matrix has $2n$ positive and $n$ negative eigenvalues. ∎

The foregoing Lemma implies that we can use, in principle, symmetric Krylov subspace methods such as MINRES [18] or SQMR [9] for solving the linear system arising

at each IP step. These methods require less storage compared with a nonsymmetric solver like GMRES. In practice, however, the diagonal scaling used in the Lemma may not be advisable numerically. Furthermore, a nonsymmetric Krylov solver becomes necessary if the preconditioner used is nonsymmetric. We note that preconditioning is absolutely essential, since iterative methods without preconditioning are ineffective and tend to stagnate or converge very slowly when applied to (2.3). In Section 3.2 we shall describe an effective multigrid method that we use as a preconditioner for the solution of this system.

As an alternative to solving the original system (or its symmetrized version), it is also possible to reduce the size of the system and to make it symmetric by means of one step of block Gaussian elimination. Elimination of $\delta z$ from the original equations leads to the *partially reduced* system

$$\begin{pmatrix} J_{uu} & -I \\ -I & -\Lambda^{-1}Z \end{pmatrix} \begin{pmatrix} \delta u \\ \delta\lambda \end{pmatrix} = - \begin{pmatrix} r_1 \\ r_2 - \Lambda^{-1} r_3 \end{pmatrix} . \tag{3.1}$$

The system matrix in (3.1) is symmetric indefinite, with $n$ positive and $n$ negative eigenvalues. Most optimization algorithms further eliminate $\delta\lambda$ to obtain the *fully reduced* system

$$(J_{uu} + \Lambda Z^{-1})\delta u = -r_1 + \Lambda Z^{-1}(r_2 - \Lambda^{-1} r_3) , \tag{3.2}$$

leading to a third formulation of the linear subproblem. Note that the coefficient matrix in (3.2) is SPD, hence we can use the preconditioned conjugate gradient (PCG) method to solve (3.2). The system can also be solved using some multigrid method.

Assuming strict complementarity holds, it is often observed that while the original system (2.3) can be relatively well conditioned (compared to the unconstrained problem), the reduced systems (3.1) and (3.2) are usually highly ill conditioned. The main reason for this problem is the complementarity condition (2.2c). This condition implies that as the outer IP iteration approaches the solution, either $z_i$ or $\lambda_i$ are close to zero. In cases where strict complementarity holds, there is no redundancy, and either $z_i \approx 0$ and $\lambda_i \gg 0$ or vice versa. Therefore, near convergence (i.e., in the later IP iterations) the matrix $Z^{-1}\Lambda$ contains very large and very small entries, making the system highly ill conditioned. It has been advocated that the above ill conditioning is "benign" to the overall optimization routine [20]. However, in our experience, this statement can be far from true when considering large-scale problems, for which iterative methods are required in order to solve the linear systems that arise at each (outer) iteration.

There are three approaches to obtain a well conditioned system. First, and most simple, is to solve the original system (2.3) without the elimination of unknowns.

A second approach is to use partial elimination. Let $D = \Lambda^{-1}Z$. Then $D$ has very large and very small entries. Since $D$ is diagonal, it is easy to partition $D$ into two matrices $D_1$ and $D_2$ where $D_1$ has entries which are smaller than 1 (say) and $D_2$ has entries that are larger than 1. As we get closer to the solution, the entries of $D_2$ go to infinity while the entries of $D_1$ go to 0.

We can now reorder the system as

$$\begin{pmatrix} J_{uu} & -I_1^\top & -I_2^\top \\ -I_1 & -D_1 & \\ -I_2 & & -D_2 \end{pmatrix} \begin{pmatrix} \delta u \\ \delta\lambda_1 \\ \delta\lambda_2 \end{pmatrix} = - \begin{pmatrix} r_1 \\ g_1 \\ g_2 \end{pmatrix} , \tag{3.3}$$

where $I_1$ and $I_2$ are made up from different rows of the identity matrix that correspond to the different variables, and $g_1$ and $g_2$ are made up from the corresponding components of $r_2 - \Lambda^{-1} r_3$. Eliminating $\delta\lambda_2$, i.e., the unknowns associated with the large matrix entries, we obtain

$$\begin{pmatrix} J_{uu} + I_2^\top D_2^{-1} I_2 & -I_1^\top \\ -I_1 & -D_1 \end{pmatrix} \begin{pmatrix} \delta u \\ \delta\lambda_1 \end{pmatrix} = - \begin{pmatrix} r_1 - I_2^\top D_2^{-1} g_2 \\ g_1 \end{pmatrix}. \tag{3.4}$$

Although the system (3.4) does not suffer from the ill conditioning induced by small $\lambda_i$ or $z_i$, it is still indefinite and requires special methods for its solution. In this paper we experiment with block preconditioners for saddle point problems; see [3] for details. We mention that a similar partitioning of diagonal entries into small and large ones can be found, for instance, in [10].

A third approach to obtain a well conditioned system is by a simple rescaling. Note that the system (3.1) is equivalent to the system

$$\begin{pmatrix} J_{uu} & \Lambda^{\frac{1}{2}} \\ \Lambda^{\frac{1}{2}} & -Z \end{pmatrix} \begin{pmatrix} \delta u \\ \Lambda^{-\frac{1}{2}} \delta\lambda \end{pmatrix} = - \begin{pmatrix} r_1 \\ \Lambda^{\frac{1}{2}} (r_2 - \Lambda^{-1} r_3) \end{pmatrix}, \tag{3.5}$$

which (assuming strict complementary) is also well conditioned. Note that the scaling preserves the symmetry of the system.

While the reformulated systems (2.3), (3.4) and (3.5) do not involve very large entries, they can still provide a challenge when the matrices arise from the discretization of partial differential operators. Although their conditioning is typically much better than the reduced systems, they can still be rather ill conditioned. The main difficulty is that, as a rule, the conditioning of the discretized PDEs deteriorates as the grid becomes finer. Therefore, even if the problem is well scaled it can still be ill conditioned. This observation motivates us to develop a multigrid method and preconditioners for the solution of the linear system.

**3.2. Multigrid.** While it is possible at least in principle to develop multigrid methods for each of the above linear systems, our strategy is to focus on three main systems. First, we develop a multigrid method for the fully coupled unreduced system (2.3). Next, we discuss a multigrid method for the fully reduced and ill conditioned system (3.2). This is by far the most commonly solved system for such problems and it is instructive to compare and observe the effect of the ill conditioning on multigrid methods. Finally, we make an attempt to develop preconditioners based on multigrid for the reduced system (3.4). A potential advantage of this approach is that we are able to work with a multigrid method for the $J_{uu}$ block which may be readily available for many problems where the Hessian is a discretization of a second-order PDE. We mention that multigrid methods for bound constrained optimization have been developed by Oosterlee in [17], but not in the framework of IP methods.

Before describing our multigrid method, we make a simple but important observation about the variables $z$ and $\lambda$. Although these variables have an obvious algebraic interpretation, they also have interpretation as grid functions. Assuming a cell-centered finite volume discretization, it is easy to see that in our case of simple bound constraints, both $z$ and $\lambda$ are defined at precisely the same spatial locations as the primal variable $u$. For more complicated constraints, however, one has to further consider the Jacobian of the constraint matrix and their discretization. This matrix will generally be rectangular (non-square), hence $u$, $z$ and $\lambda$ will generally belong to different spaces. This will generally increase the complexity of a coupled multigrid approach, at least to some extent.

**3.2.1. The $h$-ellipticity of the systems.** Following [19] we say that a system $Au = b$ is $h$-elliptic if its symbol $\tilde{A}$ satisfies

$$E_h(\tilde{A}(\theta)) := \frac{\min\{|\det(\tilde{A}(\theta))| : \theta \in T^{\text{high}}\}}{\max\{|\det(\tilde{A}(\theta))| : \pi/2 \le \theta \le \pi/2\}} \ge \rho > 0\,,$$

where $\rho$ is a constant independent of $h$ and $T^{\text{high}}$ is the range of high grid frequencies. For regular coarsening $T^{\text{high}} = \{\pi/2 \le |\theta| \le \pi\}$. For an effective multigrid method to be possible, we need to have an $h$-elliptic system; see [19]. The $h$-ellipticity property can be established only for smooth coefficients and therefore, in the context of IP methods, it offers only limited insight into the algorithm as we get close to the minimum. Nevertheless, it can be indicative of the behavior of our algorithm in the initial stages of the iteration.

Let $\mathcal{J}_{uu}$ denote the operator corresponding to the second variation of the functional $\mathcal{J}$. Assuming that the discretization of the differential operator $\mathcal{J}_{uu}$ is $h$-elliptic, we can easily establish the following result.

PROPOSITION 3.2. *Let $\widehat{\mathcal{J}}_{uu}(\theta)$ be the symbol of $\mathcal{J}_{uu}$ and assume that it is $h$-elliptic. Then the system* (2.3) *is also $h$-elliptic.*

*Proof.* The symbol of the system (2.3) is simply

$$\widehat{H}(\theta) = \begin{pmatrix} \widehat{\mathcal{J}}_{uu}(\theta) & -1 & \\ -1 & & 1 \\ & \zeta & \eta \end{pmatrix}$$

where $\zeta > 0$ and $\eta > 0$ are "frozen" coefficients. The absolute value of the determinant is therefore

$$|\det(\widehat{H})(\theta)| = |\zeta \widehat{\mathcal{J}}_{uu}(\theta) + \eta|\,.$$

Now, $h$-ellipticity of $J_{uu}$ implies that

$$\frac{\min_{\theta\in[\pi/2,\pi]} |\widehat{\mathcal{J}}_{uu}|}{\max_{\theta\in[0,\pi]} |\widehat{\mathcal{J}}_{uu}|} \ge \rho > 0$$

and that $\widehat{\mathcal{J}}_{uu}(\theta) > 0$ for $\theta \in [0, \pi]$. Therefore, we also have

$$\frac{\min_{\theta\in[\pi/2,\pi]} |\widehat{\mathcal{J}}_{uu} + \eta\zeta^{-1}|}{\max_{\theta\in[0,\pi]} |\widehat{\mathcal{J}}_{uu} + \eta\zeta^{-1}|} > \rho\,.$$

This completes the proof. □

The above Lemma implies that if we can derive an effective multigrid solver for systems with $J_{uu}$, we should also be able to derive an effective multigrid method for the systems that arise from the constrained problem. The only problem with this view is that we assumed that we freeze the coefficients. Such assumption is grossly incorrect close to the solution where $\lambda$ and $z$ vary in magnitude. However, the effectiveness of a multigrid scheme can be ascertained empirically by means of numerical experiments.

REMARK 3.3. *It is important to note that for the problems considered here, the $h$-ellipticity of the system immediately follows from the $h$-ellipticity of $J_{uu}$. However, for general constraints this is not as trivial. In fact, special attention is needed to make sure that the system is $h$-elliptic for more general constraints.*

**3.2.2. Multigrid components.** Here we give a very brief discussion of the basic ingredients of the multigrid schemes used in the paper. We have chosen to use simple geometric multigrid. Such multigrid may not be the best choice for problems with jumping coefficients [19] and therefore our algorithm works well only if $\mathcal{J}_{uu}$ has smoothly varying coefficients. Nevertheless, since we aim for simplicity of presentation we concentrate on a simple geometric multigrid cycle. It is possible to derive more sophisticated (and more expensive) multigrid methods for a class of more difficult problems with jumping coefficients [13].

**Smoothing:** Since we are assuming that $J_{uu}$ is $h$-elliptic, we can use pointwise relaxation for the smoothing of the reduced system (3.2). For the coupled system (2.3) we consider "box" smoothers in which unknowns associated with a particular cell are relaxed simultaneously. Here we have used Symmetric Gauss–Seidel (SGS) for the reduced system and block SGS for the full system.

**Prolongation/Restriction:** We have used simple linear prolongation for $u, \lambda$ and $z$. The restriction is chosen to be the transpose of the prolongation and we use Galerkin coarse grid operators throughout.

The above is combined to form a standard V-cycle for the solution of either the fully reduced system (3.2) or the full system (2.3).

**3.3. Solving the partially reduced system.** While it is straightforward to develop multigrid methods for the systems (3.2) and (2.3) it is difficult to develop such a method for system (3.1). The problem is that while $\delta\lambda$ can be easily interpreted as a grid function (with natural prolongation, restriction and coarsening) the variables $\delta\lambda_1$ involve only part of this set. Therefore, it is more difficult to construct such operations. We have chosen to consider algebraic preconditioners for this system where the multigrid V-cycle for the reduced system is part of the preconditioner.

Motivated by the work of [8] (see also [3]) we consider a block preconditioner of the form

$$M = \begin{pmatrix} A & \\ & \widehat{S} \end{pmatrix} \tag{3.6}$$

where

$$\begin{aligned} A &\approx J_{uu} + I_2^\top D_2^{-1} I_2 \,, \\ \widehat{S} &\approx I_1 (J_{uu} + I_2^\top D_2^{-1} I_2)^{-1} I_1^\top + D_1 \,. \end{aligned}$$

Here $A$ need not be an explicitly formed matrix; rather, a prescription is given for computing the action of the linear operator $A^{-1}$ on a given vector. For instance, we can use one or more V-cycles of the standard multigrid method previously discussed. It is less straightforward to find a good choice of the approximate Schur complement $\widehat{S}$ or of its inverse $\widehat{S}^{-1}$. Here we describe a possible approach. First, since the entries of $D_1$ are small, we drop them altogether from the approximation. Second, we note that the Moore–Penrose inverse of $I_1$ is just

$$I_1^\dagger = I_1^\top \,,$$

and therefore we choose $\widehat{S}^{-1}$ as

$$\widehat{S}^{-1} = I_1 (\mathcal{J}_{uu} + I_2^\top D_2^{-1} I_2) I_1^\top \,,$$

which has very low set-up and application costs. The above preconditioner is used within a MINRES iteration [18].

**4. Multilevel continuation.** In the previous sections we discussed the solution of the constrained optimization problem on a single grid; auxiliary coarse grids were used only in the solution of the linear subproblems arising at each outer IP iteration. However, a general framework for nonlinear PDE problems can and should involve solution on multiple grids. The advantage of such approach is that we are able to obtain a good initial guess for the fine grid problem by solving coarse grid problems. Since the coarse grid problems can be solved cheaply, it is possible to quickly find an approximate solution to the fine grid problem; using this approximate solution as an initial guess, only a few iterations are needed to achieve convergence. However, the use of interior point methods in a multilevel framework is difficult. The problem is that in order to solve the coarse grid problems accurately we need to decrease the barrier parameter $\mu$ to (approximately) 0. Moreover, using the coarse grid solution on a finer grid is difficult. The reason is that after interpolation the points in the interpolated solution are not always interior to the feasibility domain, or they may be very close to the boundary of the domain. Thus, it is hard to use information about the active set in a multilevel method. Such information is often referred to as a "hot start"; it is a crucial part of every active set nonlinear programming package (see for example [5]).

Nevertheless, PDE-related optimization problems are different from general constrained optimization problems. The main difference is that the objective functions that we need to minimize are obtained from the discretization of differential operators and as such they are inherently inaccurate (i.e., approximate). Thus, in the context of optimization with PDEs, we are willing to tolerate errors in the solution of the optimization problem that are up to $h^p$ where $h$ is the grid size and $p$ is the accuracy of our discretization. Therefore, for most PDE optimization problems there is no need to reduce $\mu$ beyond some threshold which depends on $h$. The idea of stabilizing a numerically ill conditioned problem by adding a small $\mathcal{O}(h^p)$ term is a fundamental principle in numerical analysis. For example, artificial viscosity is often added to unstable methods for hyperbolic PDEs [14] and artificial compressibility is added to unstable discretizations of the Stokes problem [12].

In this work we simply choose $\mu = \mu(h) = h^p$. We then envision a process where a grid continuation approach is used to solve the nonlinear problem. For a given (coarse) grid size $H$ we use $\mu = \mu(H)$ which is rather large and therefore terminate the iteration far away from the boundary. We then interpolate the solution to a finer grid $h$ and start the computation with $\mu(h)$, and get closer to the boundary. A sketch of the process is shown in Figure 4.1.

Using the theorem in [16, p. 514] we can see that the error in the solution is also of order $h^p$ and therefore the solution on each grid is acceptable within the discretization error. Using interior point methods in this fashion we get better conditioned problems to solve and we can expect to converge quickly. Furthermore, since $\mu$ is $\mathcal{O}(h^p)$ our points are all interior and we are able to use "hot starts" at each outer iteration. Another advantage of this approach is that at least for coarse grids, we avoid some of the notorious ill conditioning of the systems that arise from interior point methods.

**5. Numerical experiments.** We now return to the minimal surface problem. We consider Example 17 from [7]. Here the domain is $\Omega = [0,1] \times [0,1]$ and the obstacle function $b$ is defined as follows:

$$b(x,y) = \begin{cases} 1 & \text{if} \quad |x-.5| < .25 \quad \text{and} \quad |y-.5| < .25\,, \\ 0 & \text{otherwise}\,. \end{cases} \tag{5.1}$$

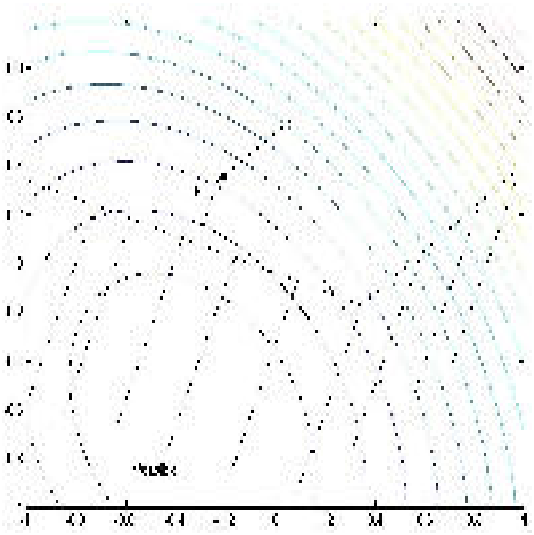

FIG. 4.1. *Advancement in the interior point as a function of the grid size.*

First, note that the gradient and the Hessian can be written by computing first

$$\begin{aligned}
\sigma &= \left(\sqrt{A_1(D_1u)^2 + A_2(D_2u)^2 + 1}\right)^{-1}, \\
\widehat{\sigma} &= \left(\sqrt{A_1(D_1u)^2 + A_2(D_2u)^2 + 1}\right)^{-3}, \\
J_{uu}^{+} &= D_1^{\top}\,\mathrm{diag}(A_1^{\top}\sigma)D_1 \;+\; D_2^{\top}\,\mathrm{diag}(A_2^{\top}\sigma)D_2\,, \\
J_{uu}^{-} &= D_1^{\top}\,\mathrm{diag}(D_1u)A_1^{\top}\,\mathrm{diag}(\widehat{\sigma})\,(A_1\,\mathrm{diag}(D_1u)D_1 + A_2\,\mathrm{diag}(D_2u)D_2) + \\
&\quad D_2^{\top}\,\mathrm{diag}(D_2u)A_2^{\top}\,\mathrm{diag}(\widehat{\sigma})\,(A_1\,\mathrm{diag}(D_1u)D_1 + A_2\,\mathrm{diag}(D_2u)D_2)\;,
\end{aligned}$$

and then setting

$$\begin{aligned}
J_u &= J_{uu}^{+}u\,, \\
J_{uu} &= J_{uu}^{+} - J_{uu}^{-}\,.
\end{aligned}$$

It is worth noting that the matrix $J_{uu}^{+}$ is SPD for every $u$ and that it approximates a differential operator of the form

$$J_{uu}^{+} \approx \mathrm{div}\;\sigma(u)\;\mathrm{grad}\,.$$

Furthermore, note that $\sigma(u)$ may not vary smoothly if $\mathrm{grad}\,u$ is large. Note also that $J_{uu}^{-}$ may have an adverse effect on the positivity of the Hessian far from the solution. This is the reason why most algorithms for solving problems with similar operators ignore this term. Nevertheless, since we use a multilevel approach, we are able to stay close to the minimum and use the full Hessian in our experiments.

We now use the gradient and Hessian in our numerical experiments. We divide the numerical experiment section into four parts. First, we experiment with the multilevel approach and compare it to a one-grid approach, where we do not use coarser grids in order to obtain an approximate solution to the fine grid problem. Second, we experiment with different preconditioners. This includes a multigrid preconditioners for the unreduced system (2.3), a block diagonal preconditioner for the partially

| Level | Iterations |
|---|---|
| $2^4$ | 5 |
| $2^5$ | 7 |
| $2^6$ | 8 |
| $2^7$ | 10 |

TABLE 5.1
*Number of inexact Newton steps for different grids.*

reduced system (3.4), and a multigrid preconditioner for the fully reduced system (3.2).

For each of the systems we first test the linear solver by itself on linear systems arising in both the initial and final stages of the IP iteration, using a fairly stringent stopping criterion ($\|r_k\|_2 < 10^{-8}\|r_0\|_2$, where $r_k$ is the computed residual at the $k$th iteration).

**5.1. Experiments with multilevel continuation.** In our first experiment we use a multilevel approach to obtain an approximate solution to the fine grid problem. We set $\mu = \mu(h)$ and solve each coarse grid problem to a relative tolerance of $10^{-3}$. Each of the optimization problems was solved using an inexact Newton method. The linear system that is solved is (2.3) and we use a multigrid preconditioner with GMRES as the Krylov subspace method. The multigrid iteration is discussed in the next subsection. The number of iterations for each level is presented in Table 5.1. The solutions obtained at the end of the IP iterations is shown in Fig. 5.1.

As can be seen from the table, the iteration count is similar for all grids. Only a very slight increase is observed when going from coarse grids to very fine grids. Recall that for Newton's method, mesh independence properties have been established [6]. We suspect that this (rather small) increase in iterations is due to the dependence of $\mu$ in $h$ which makes each optimization problem somewhat different. In contrast, starting from the feasible solution $u(x, y) = 5\sin(\pi x)\sin(\pi y)$, we were unable to achieve convergence even after 300 iterations (at which point the algorithm had to be terminated).

This experiment demonstrates that using a multilevel approach here is crucial. It can result in an order of magnitude computational saving and, most important, it succeeds where more standard approaches fail.

Figure 5.1 displays the solution obtained on different grids and Figure 5.2 displays the 2-norm of the relative residual of the system (2.3) at each of the outer iterations for different levels. We see that by using coarse grids we are able to reduce the residual of the fine grid to less than $10^{-4}$.

**5.2. Experiments with the full system.** In this subsection we describe the results of solving the full system (2.3) using GMRES with a multigrid preconditioner. Since the systems change their character between the initial and the final stages of the algorithm, we experiment with the systems obtained at each stage of the optimization problem. This is done by changing the inner tolerance at each Newton iteration to $10^{-8}$. Thus, we use a full Newton method which has (at least in principle) quadratic outer convergence at the price of accurate solution of the linear system. The results are presented in Table 5.2.

From the Table we conclude that our multigrid method is effective for all the systems obtained in the IP process. A slight increase in the number of iterations

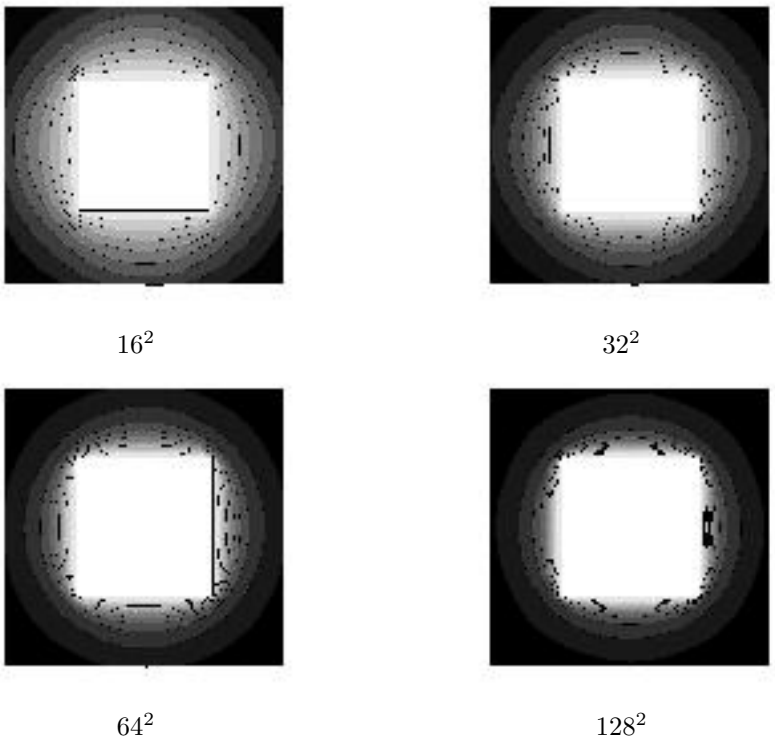


Fig. 5.1. *Solutions $u(x, y)$ at the final iterations of a multilevel interior point method.*

| IP Iteration Number | $16 \times 16$ | $32 \times 32$ | $64 \times 64$ | $128 \times 128$ |
|---|---|---|---|---|
| 1 | 6 | 8 | 9 | 13 |
| 2 | 6 | 8 | 9 | 13 |
| 3 | 5 | 8 | 8 | 12 |
| 4 | 4 | 6 | 8 | 9 |
| 5 | 4 | 6 | 7 | 9 |
| 6 | - | 6 | 7 | 8 |
| 7 | - | 5 | 7 | 8 |
| 8 | - | - | - | 7 |

Table 5.2
*Iteration counts for solving full system (2.3) using GMRES with multigrid preconditioner.*

is observed; however, for this particular problem this is expected. Notice that the matrix $J_{uu}$ is a discretization of an operator of the form $\operatorname{div} \sigma \operatorname{grad}$ where, towards convergence, $\sigma$ contains jumps. Thus, any purely geometric multigrid is expected to produce similar results.

**5.3. Experiments with the partially reduced system.** We made a few attempts to solve the partially reduced system (3.4), but we met with some difficulties. We cannot expect fast convergence for any solver of (3.4) without preconditioning, and as expected, MINRES without a preconditioner does not converge to a tolerance of $10^{-8}$ within 1000 iterations for each IP iteration on the $128 \times 128$ grid. The per-

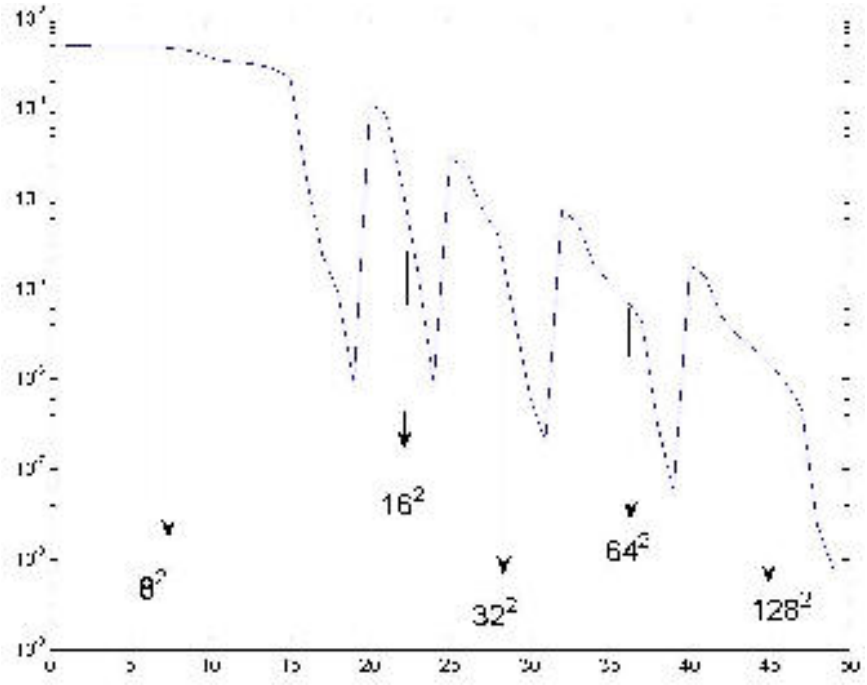


FIG. 5.2. *Relative residual at each outer iteration for different grid sizes.*

formance of the block diagonal preconditioner proposed in section 3.3 was found to be disappointing. The convergence of preconditioned GMRES was quite slow, even when we used exact solves for the (1,1) block. Clearly, a better approximation for the Schur complement is needed in order to make this approach viable.

Other attempts to use preconditioners for saddle point systems, such as (generalized) constraint preconditioning and HSS preconditioning (see, for example, [3] for a description of these techniques) also proved futile. Interestingly, reasonable results where obtained by a symmetric diagonal (Jacobi) scaling of the matrix in (3.4) followed by an incomplete LU factorization using a drop tolerance. For the linear system arising from the last IP iteration on a $128 \times 128$ grid, preconditioned GMRES converged in 57 iterations for a relative stopping criterion of $10^{-8}$, and in 17 iterations for a relative stopping criterion of $10^{-3}$. The drop tolerance used was $\tau = 10^{-2}$, which resulted in a fairly sparse preconditioner. However, ILU-preconditioned GMRES is unlikely to scale well for large problems, especially in 3D applications.

A few experiments on the related system (3.5) gave very similar results. In summary, we cannot recommend the partially reduced approach on the basis of our results, although better results may well be possible using different techniques.

**5.4. Experiments with the fully reduced system.** Next, we consider the fully reduced system (3.2). Numerical experiments were ran using several variations on the conjugate gradient method. Our results focus on the employment of one V-cycle of multigrid as a preconditioner for the conjugate gradient method. Once again we revert to a full Newton method for the solution of the optimization problem and set the tolerance of the inner iteration to $10^{-8}$. Table 5.4 displays iteration counts for each solver at each IP iteration and the last IP iteration for several grid sizes in

| IP Iteration Number | $16 \times 16$ | $32 \times 32$ | $64 \times 64$ | $128 \times 128$ |
|---|---|---|---|---|
| 1 | 10 | 16 | 22 | 36 |
| 2 | 9 | 14 | 23 | 34 |
| 3 | 8 | 14 | 16 | 20 |
| 4 | 7 | 10 | 13 | 16 |
| 5 | 6 | 9 | 12 | 14 |
| 6 | - | - | 11 | 13 |

Table 5.3
*Iteration counts for solving system (3.2) using PCG.*

order to show the scaling of the method.

Clearly, solving system (3.2) with conjugate gradient preconditioned with multigrid is effective. Similar to the full system, the number of iterations increases slightly as we approach the solution. This is explained again by the jumping coefficients in the continuous operator.

**6. Conclusions.** We have explored a new multilevel approach for variational problems with bound inequality constraints and compared a number of techniques for the solution of the linear systems that arise at each interior point iteration. We have demonstrated that an appropriate multilevel continuation in $\mu$ can dramatically improve the results of an interior point algorithm. We have also demonstrated that a multigrid method for the full system can be effectively used for the solution of the problem. Comparing different strategies for the solution, we observe that the multigrid method for the unreduced problem was the most effective method while a multigrid method for the fully reduced problem requires slightly more iterations. A preconditioner for the partially reduced system did not prove to be as efficient and further work is required in order to obtain better approximations to the Schur complement.

REFERENCES

[1] U. Ascher, E. Haber and H. Huang, *On effective methods for implicit piecewise smooth surface recovery*, Technical Report TR-2004-013-A, Department of Mathematics and Computer Science, Emory University, Atlanta, GA. To appear in SIAM J. Sci. Comp.

[2] V. Baryamureeba and T. Steihaug, *On the convergence of an inexact primal-dual interior point method for linear programming*, Technical Report No. 188, Department of Informatics, University of Bergen, 2000.

[3] M. Benzi, G. H. Golub and J. Liesen, *Numerical solution of saddle point problems*, Acta Numerica, 14 (2005), pp. 1–137.

[4] R. Byrd, J. Nocedal and R. Waltz, *Steering Penalty Methods*, Tech. Report, Optimization Technology Center, 2005.

[5] R. Byrd, M. E. Hribar, and J. Nocedal , *An interior point algorithm for large scale nonlinear programming*, SIAM J. Optim., 9 (2000), pp. 877-900.

[6] P. Deuflhard and F. Potra, *Asymptotic mesh independence of Newton–Galerkin methods via a refined Mysovskii theorem*, SIAM J. Numer. Anal, 29 (1992), pp. 1395–1412.

[7] D. E. Dolan and J. J. Moré, *Benchmarking Optimization Software with COPS*, Tech. Report ANL/MCS-246, Argonne National Laboratory, Argonne, IL, November 2000.

[8] H. C. Elman, D. J. Silvester and A. J. Wathen, *Finite Elements and Fast Iterative Solvers*, Numerical Mathematics and Scientific Computation, Oxford University Press, Oxford, UK, 2005.

[9] R. W. Freund and N. M. Nachtigal, *Software for simplified Lanczos and QMR algorithms*, Appl. Numer. Math., 19 (1994), pp. 319–341.

[10] P. E. Gill, W. Murray, D. B. Ponceleón, and M. A. Saunders, *Preconditioners for indefinite systems arising in optimization*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 292–311.
[11] R. Glowinski, *Numerical Methods for Nonlinear Variational Problems*, Springer Series in Computational Physics, New York/Berlin/Heidelberg/Tokyo, 1984.
[12] P. M. Gresho and R. L. Sani, *Incompressible Flow and the Finite Element Method. Advection-Diffusion and Isothermal Laminar Flow*, J. Wiley, Chichester, UK, 1998.
[13] S. Knapek, *Matrix-dependent multigrid homogenization for diffusion problems*, SIAM J. Sci. Comp., 20 (1999), pp. 515–533.
[14] R. J. LeVeque, *Numerical Methods for Conservation Laws*, Birkhäuser–Verlag, Basel, 1990.
[15] J. Nocedal, A. Waechter and R. Waltz, *Adaptive Barrier Update Strategies for Nonlinear Interior Methods*, Tech. Report, IBM Watson Research Center, 2005.
[16] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, New York, 1999.
[17] C. W. Oosterlee, *On multigrid for linear complementarity problems with applications to American-style options*, Elecr. Trans. Numer. Anal., 15 (2003), pp. 165–185.
[18] C. C. Paige and M. A. Saunders, *Solution of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal., 12 (1975), pp. 617–629.
[19] U. Trottenberg, C. W. Oosterlee, and A. Schuller, *Multigrid*, Academic Press, New York, 2001.
[20] M. H. Wright, *Ill-conditioning and computational error in interior methods for nonlinear programming*, SIAM J. Optim. 9 (1999), pp. 84–111.
[21] S. J. Wright, *Primal-Dual Interior Point Methods*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1997.