Technical Report

TR-2009-018

Multilevel Algorithms for Large-Scale Interior Point Methods

by

Michele Benzi, Eldad Haber, Lauren Taralli

Mathematics and Computer Science

EMORY UNIVERSITY

MULTILEVEL ALGORITHMS FOR LARGE-SCALE INTERIOR POINT METHODS

MICHELE BENZI*, ELDAD HABER†, AND LAUREN TARALLI‡

Abstract. We develop and compare multilevel algorithms for solving 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. Some minimal surface and volume-constrained image registration problems are used to illustrate the various approaches.

1. Introduction. In this work we consider the solution of finite-dimensional constrained optimization problems of the form

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

$$\text{s.t.} \quad 0 \le c(u)\,, \tag{1.1b}$$

where $u \in \mathbb{R}^n$ and c denotes a differentiable mapping from $\mathbb{R}^n$ to $\mathbb{R}^m$ incorporating the constraints ($n \ge m$) and the objective function $J : \mathbb{R}^n \longrightarrow \mathbb{R}$ is assumed to be a discretization of a (generally nonlinear) functional $\mathcal{J}$ involving a differential operator defined on an appropriate Banach space $\mathcal{X}$. 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 two model problems.

EXAMPLE 1. **Minimal surface problem with an obstacle**. *Let Ω be a bounded, not necessarily simply connected, open subset of $\mathbb{R}^2$, and let $b = b(\boldsymbol{x}) \in \mathcal{X} := W_0^{1,1}(\Omega)$. The minimal surface problem reads:*

$$\min_{u\in\mathcal{X}} \quad \mathcal{J}(u) = \int_\Omega \sqrt{|\nabla u|^2 + 1}\; d\boldsymbol{x} \tag{1.2a}$$

$$\text{s.t.} \quad 0 \le u - b \tag{1.2b}$$

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

Upon consistent discretization we obtain

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

$$\text{s.t.} \quad u_h - b_h \ge 0\,, \tag{1.3b}$$

*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.

‡Quantitative Analytics Research Group, Standard & Poor's, New York, New York 10041, USA (lauren.taralli@gmail.com).

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

This problem has been intensively studied in literature, both theoretically and computationally; see, e.g., [13] for existence and uniqueness results. From the optimization point of view, this first example is a simple bound constrained minimization problem. Although interior point methods can be successfully applied to this problem, there already are highly effective methods; see, e.g., [1, 14, 25] and the references therein. For more general, possibly nonlinear constraints, this is not the case. As an illustration, consider the following example.

EXAMPLE 2. **Volume-constrained image registration**. *Let $\mathbf{u}$ be a vector function in $\mathcal{X} := H_0^1(\Omega) \times H_0^1(\Omega)$ where Ω is a bounded, open subset of $\mathbb{R}^2$ and let $b = b(\boldsymbol{x}) \in \mathcal{Y} = L^2(\Omega)$. In image registration one seeks a smooth $\mathbf{u}$ which solves the constrained variational problem*

$$\min_{\mathbf{u}\in\mathcal{X}} \quad \mathcal{J}(\mathbf{u}) = \int_\Omega \beta|\nabla\mathbf{u}|^2 + D(\mathbf{u})\; d\boldsymbol{x} \tag{1.4a}$$

$$\text{s.t.} \quad b \leq \nabla\cdot\,\mathbf{u} \tag{1.4b}$$

$$\mathbf{u}|_{\partial\Omega} = \mathbf{0}. \tag{1.4c}$$

Here D is a distance function between two images and $\beta > 0$ is a trade-off parameter; see e.g., [18, 17]). Upon consistent discretization we obtain

$$\min_{u_h} \quad J(u_h) = \beta u_h^\top A_h u_h + D_h(u_h) \tag{1.5a}$$

$$\text{s.t.} \quad B_h u_h - b_h \geq 0\,, \tag{1.5b}$$

where u_h is the nodal discretization of $\mathbf{u}$, A_h is a discretization of the vector Laplacian, and B_h is a discretization of the divergence.

While we focus here on these two model problems, our goal is more general. Namely, we aim to develop efficient methods for nonlinear variational problems with inequality constraints. 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 a 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 devise techniques that perform reasonably well for large classes of applications and to 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 a self-adjoint, elliptic partial differential operator.
- A2 The Jacobian of the constraints has full rank and represents a lower order differential operator than the Hessian of the objective function.

Such problems arise in many areas of science and engineering. 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). We note that this assumption is satisfied for the Hessians arising from Examples 1 and 2.*

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) [24, 30]. 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 usually better than that of active set methods; see [19] and [24, 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. Some convergence results are given in section 3. In section 4 we describe multigrid and iterative solvers for the solution of the linear problems that arise at each outer iteration. In section 5 we describe how to use multilevel continuation (grid sequencing) with interior point methods, in particular, how to adapt the log barrier parameter to the grid. In section 6 we perform numerical experiments that compare the effectiveness of our methods. Finally, in section 7 we summarize our findings.

2. An inexact Newton framework. To solve 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:

$$\begin{aligned} \min \quad & J(u) - \mu e^\top \log(z)\,, \\ \text{s.t.} \quad & c(u) - z = 0\,, \end{aligned} \tag{2.1}$$

where $c : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ (with $n \geq m$), $z \in \mathbb{R}^m$ is a slack variable, $\log(z)$ is the vector whose ith component is $\log(z_i)$, μ is the barrier parameter, and $e = [1, \ldots, 1]^\top \in \mathbb{R}^m$.

Consider the Lagrangian

$$\mathcal{L}(u, z, \lambda) = J(u) - \mu e^\top \log(z) - \lambda^\top (c(u) - z)\,, \tag{2.2}$$

where λ is a vector of Lagrange multipliers. The Euler–Lagrange equations associated with the Lagrangian are:

$$r_1 := J_u - c_u^\top \lambda = 0\,, \tag{2.3a}$$

$$r_2 := z - c(u) = 0\,, \tag{2.3b}$$

$$r_3 := z \odot \lambda - \mu e = 0\,, \tag{2.3c}$$

where $\odot$ denotes the componentwise (Hadamard) product, J_u is the Jacobian of J, and c_u is the Jacobian of c, which is assumed to have full rank; in the case of simple bound constraints (Example 1), it is $c_u \equiv I$. Note that if we set the barrier parameter μ equal to zero we obtain the classical KKT conditions for the constrained optimization problem; see [24].

Using a Newton-type method for the solution of system (2.3) we obtain a sequence of linear systems of the form

$$\begin{pmatrix} J_{uu} + T(u,\lambda) & -c_u^\top & 0 \\ -c_u & 0 & I_m \\ 0 & Z & \Lambda \end{pmatrix} \begin{pmatrix} \delta u \\ \delta\lambda \\ \delta z \end{pmatrix} = - \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix}, \tag{2.4}$$

where $Z = \mathrm{diag}(z)$ and $\Lambda = \mathrm{diag}(\lambda)$. Here J_{uu} stands for the Hessian of J or some approximation of it and T corresponds to the curvature of the constraints. As is often the case in large-scale problems, assuming that J_{uu} is SPD, we can drop T. This approximation is especially justified if T tends to give negative curvature. It is possible to show [16, 4] that this approximation corresponds to a Gauss–Newton method. From here on, we let $T(u,\lambda) = 0$ in (2.4).

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 λ and z, and the next (outer) iteration is carried out. The overall procedure is outlined in Algorithm 1 below.

Algorithm 1 Inexact-Newton method for constrained optimization:
$u \leftarrow \mathrm{INIC}(u_0)$;

initialize $\{u, \lambda, z\} \leftarrow \{u_0, \lambda_0, z_0\}$ and the barrier parameter μ_0; select $\eta > 0$
while true **do**
  compute r_1, r_2, r_3
  approximately solve (2.4) with tolerance $\mathrm{tol} \le \eta\lambda^\top z$ % (see [5])
  use an Armijo line search to accept/reject step
  test for termination
  evaluate μ_{k+1}
end while

There are two key components that need to be addressed in order to make this solution process an effective one. First, a good linear solver is needed and second, a good strategy to evaluate μ 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 μ has been the topic of several recent papers; see, e.g., [1, 6, 23]. In the next section we discuss properties of the discrete optimization problem. Next, we discuss the solution of the linear system. Subsequently, in section 5 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 μ fixed for each optimization problem, and decrease μ in tandem with the mesh size.

3. Convergence results. In this section we derive an estimate (bound) on the total number of Newton iterations in terms of the discretization parameter h. Consider the minimization problem

$$\min_{u \in \mathcal{X}} \quad \mathcal{J}(u) \tag{3.1a}$$

$$\text{s.t.} \quad c(u) \ge 0. \tag{3.1b}$$

Here $\mathcal{J}$ is a convex functional defined on the Banach space $\mathcal{X}$ and $c : \mathcal{X} \longrightarrow \mathcal{Y}$ represents an inequality constraint, where $\mathcal{Y}$ is an ordered Banach space. We also

assume that upon discretization, we obtain a discrete optimization problem of the form

$$\min_{u_h} \quad J(u_h) \tag{3.2a}$$

$$\text{s.t.} \quad c_h(u_h) \geq 0, \tag{3.2b}$$

where $J : \mathbb{R}^n \longrightarrow \mathbb{R}$ is convex and self-concordant [5], and each component c_i of $c_h : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ is convex. This conditon is often satisfied in practice; for instance, it holds for both examples (1.2)-(1.4). We consider a sequence of minimizers of the primal problem

$$\min \quad M_h(u_h, \mu_k) = J(u_h) - \mu_k e^\top \log(c_h(u_h)) \tag{3.3}$$

and drive μ_k to 0 at some rate. The following lemmas are proved in [24] and [5].

LEMMA 3.1. *For a fixed μ, let u_h be the minimizer of $M_h(u_h, \mu)$ and let u_h^0 be the minimizer of* (3.2). *Then*

$$J(u_h) - J(u_h^0) \leq \mu n.$$

The quantity μn is referred to as the *duality gap*. A direct byproduct of the duality gap is that it is easy to estimate the final μ that is needed to attain a prescribed accuracy.

LEMMA 3.2. *Assume that a prescribed accuracy $\varepsilon > 0$ is given, and that μ is decreased geometrically with a constant $\gamma > 1$, that is, $\mu_{k+1} = \gamma^{-1}\mu_k$. The total number $\mathcal{N}_b$ of barrier problems to be solved is*

$$\mathcal{N}_b \leq \frac{\log(\frac{n\mu_0}{\varepsilon})}{\log(\gamma)}. \tag{3.4}$$

Next, we need to estimate the number of Newton iterations needed per optimization problem. This is summarized in the following lemma.

LEMMA 3.3. *Assume that M_h in (3.3) is self-concordant. Then the number $\mathcal{N}_N$ of Newton iterations needed to minimize M_h within ε when starting at the minimizer of $M_h(u_h, \mu_{k-1})$ is bounded by*

$$\mathcal{N}_N \leq \frac{n(\gamma - 1 - \log(\gamma))}{\alpha_1} + \alpha_2, \tag{3.5}$$

where α_1 and α_2 are moderate constants.

The last piece in the puzzle is the transition between two grids. Once again, the following result holds under the self-concordance assumption.

LEMMA 3.4. *Let h and H, with $0 < h < H$, be given. For a fixed μ_k, let v_H^k be the minimizer of $M_H(u_H, \mu_k)$ and assume that*

$$M_H(v_H^k) = M(v^k) + \mathcal{O}(H^2).$$

Assume also that v_h^k is the minimizer of $M_h(u_h, \mu_k)$ and that

$$M_h(v_h^k) = M(v^k) + \mathcal{O}(h^2).$$

Let $\hat{v}_h^k$ be an interpolant of v_H^k and assume that

$$M_h(\hat{v}_h^k) = M_h(v_h^k) + \mathcal{O}(H^2).$$

Then the number $\mathcal{N}$ of Newton iterations needed for the minimization of M_h starting at $\hat{v}_h^k$ is bounded by

$$\mathcal{N}_N \leq \frac{H^{-2}}{\alpha_1} + \alpha_2,$$

where α_1 and α_2 are the constants in Lemma 3.3.

We note that the last result is rather unsatisfactory, since H is often of the form $H = ch$ for some constant $c > 1$, hence the upper bound grows like h^{-2}. Using these results, however, we can derive a better estimate for the total number of Newton iterations needed to attain a prescribed accuracy.

First, assuming that no multilevel strategy is used, the total number of Newton iterations is bounded by

$$\mathcal{N} = \mathcal{N}_N \mathcal{N}_b \leq \frac{\log(\frac{n\mu_0}{\varepsilon})}{\log(\gamma)} \left(\frac{n(\gamma - 1 - \log(\gamma))}{\alpha_1} + \alpha_2 \right).$$

To put this expression into the context of a discretized optimization problem we assume that we work in 2D and that $n = \frac{1}{h^2}$. We also assume that our discretization is $\mathcal{O}(h^2)$ accurate and therefore we choose $\varepsilon = h^2$. The foregoing bound becomes

$$\mathcal{N} \leq \frac{\log(\frac{\mu_0}{h^4})}{\log(\gamma)} \left(\frac{\gamma - 1 - \log(\gamma)}{h^2 \alpha_1} + \alpha_2 \right). \tag{3.6}$$

This bound is apparently still unsatisfactory, because the total number of Newton iterations grows like h^{-2}. It is possible to obtain a better bound by choosing γ as a function of h. So let $\gamma = 1 + h^p$ and recall that for h small,

$$\log(1 + h^p) = h^p - \frac{1}{2} h^{2p} + \mathcal{O}(h^{3p}).$$

We obtain

$$\mathcal{N} \leq \frac{\log(\mu_0) - 4\log(h)}{h^p} \left(\frac{h^{2p}}{2h^2 \alpha_1} + \alpha_2 \right) = (\log(\mu_0) - 4\log(h)) \left(\frac{h^{p-2}}{2\alpha_1} + \alpha_2 h^{-p} \right).$$

If we choose $p = 1$ we obtain the bound

$$\mathcal{N} \leq (\log(\mu_0) - 4\log(h)) \frac{1}{h} \left(\frac{1}{2\alpha_1} + \alpha_2 \right). \tag{3.7}$$

When using a multilevel strategy one typically starts with $\mu_0 = (2h)^4$. This gives a slightly better estimate:

$$\mathcal{N} \leq 4\log(2) \frac{1}{h} \left(\frac{1}{2\alpha_1} + \alpha_2 \right). \tag{3.8}$$

Hence, the total number of Newton iterations grows at worst like h^{-1}. Of course, the above estimate is just an upper bound, and better convergence rates may be observed in practice. In particular, it is not necessary to take $\mu_0 = (2h)^4$; see section 6.

REMARK 3.5. *It is important to note that if we fix μ for all grids then, under suitable smoothness assumptions on the solution, the usual mesh independence theorem for Newton's method holds [8]. The growth in the number of iterations occurs because we demand more accurate results for a sequence of decreasing values of h. In practice, mesh-independence may also fail to occur when one of the field variables (here, the Lagrange multiplier) is discontinuous at the solution; see, for instance, the discussion in [20].*

4. 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.

4.1. Alternative linear system formulations. It is possible to reformulate the linear system (2.4) 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, we can simply try solving the original system (2.4). The main advantage of this system is that as long as strict complementarity holds, vanishing z_i and λ_i do not have too much of an adverse effect on its conditioning. However, the disadvantage of this system is that it is large and nonsymmetric. The following simple lemma shows that the last difficulty can be circumvented, at least in principle, by an appropriate diagonal scaling.

LEMMA 4.1. *The coefficient matrix in* (2.4) *is similar to a real symmetric indefinite matrix, hence it has real spectrum. There are exactly* $n+m$ *positive and* m *negative eigenvalues.*

Proof. It is immediate to verify that

$$\begin{pmatrix} I_n & 0 & 0 \\ 0 & I_m & 0 \\ 0 & 0 & Z^{-\frac{1}{2}} \end{pmatrix} \begin{pmatrix} J_{uu} & -c_u^\top & 0 \\ -c_u & 0 & I_m \\ 0 & Z & \Lambda \end{pmatrix} \begin{pmatrix} I_n & 0 & 0 \\ 0 & I_m & 0 \\ 0 & 0 & Z^{\frac{1}{2}} \end{pmatrix} = \begin{pmatrix} J_{uu} & -c_u^\top & 0 \\ -c_u & 0 & Z^{\frac{1}{2}} \\ 0 & Z^{\frac{1}{2}} & \Lambda \end{pmatrix},$$

where we have used the fact that $Z^{-\frac{1}{2}}\Lambda Z^{\frac{1}{2}} = \Lambda$, since Z and Λ are both diagonal. Furthermore, the assumptions on J_{uu} (SPD) and on c_u (full rank) ensure that the matrices

$$A = \begin{pmatrix} J_{uu} & -c_u^\top \\ -c_u & 0 \end{pmatrix} \qquad \text{and} \qquad c_u J_{uu}^{-1} c_u^\top$$

are nonsingular. Letting

$$\mathcal{L} = \begin{pmatrix} I_n & 0 & 0 \\ 0 & I_m & 0 \\ -Z^{\frac{1}{2}} \left(c_u J_{uu}^{-1} c_u^\top\right)^{-1} c_u J_{uu}^{-1} & -Z^{\frac{1}{2}} \left(c_u J_{uu}^{-1} c_u^\top\right)^{-1} & I_m \end{pmatrix},$$

we have that

$$\mathcal{L} \begin{pmatrix} J_{uu} & -c_u^\top & 0 \\ -c_u & 0 & Z^{\frac{1}{2}} \\ 0 & Z^{\frac{1}{2}} & \Lambda \end{pmatrix} \mathcal{L}^T = \begin{pmatrix} J_{uu} & -c_u^\top & 0 \\ -c_u & 0 & 0 \\ 0 & 0 & S \end{pmatrix},$$

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

$$\begin{pmatrix} J_{uu} & -c_u^\top \\ -c_u & 0 \end{pmatrix} \qquad \text{and} \qquad S = \Lambda + Z^{\frac{1}{2}} \left(c_u J_{uu}^{-1} c_u^\top\right)^{-1} Z^{\frac{1}{2}}.$$

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

The foregoing lemma implies that we can use, in principle, symmetric Krylov subspace methods such as MINRES [26] or SQMR [11] for solving the linear system arising at each IP step. These methods require less storage compared with a nonsymmetric solver like GMRES [27]. In practice, however, the diagonal scaling used in Lemma 4.1 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 (2.4). In section 4.2 we shall describe an effective multigrid method which can be used 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 δz from the original equations leads to the *partially reduced* system

$$\begin{pmatrix} J_{uu} & -c_u^\top \\ -c_u & -\Lambda^{-1}Z \end{pmatrix} \begin{pmatrix} \delta u \\ \delta\lambda \end{pmatrix} = - \begin{pmatrix} r_1 \\ r_2 - \Lambda^{-1} r_3 \end{pmatrix} . \tag{4.1}$$

Note that δz can be readily obtained once δu and $\delta\lambda$ have been computed. The system matrix in (4.1) is symmetric indefinite, with n positive and m negative eigenvalues. Most optimization algorithms further eliminate $\delta\lambda$ to obtain the *fully reduced* $(n \times n)$ system

$$\left(J_{uu} + c_u^\top \Lambda Z^{-1} c_u\right) \delta u = -r_1 + c_u^\top \Lambda Z^{-1} \left(r_2 - \Lambda^{-1} r_3\right) , \tag{4.2}$$

leading to a third formulation of the linear subproblem. Note that the coefficient matrix in (4.2) is SPD, hence we can use the preconditioned conjugate gradient (PCG) method to solve (4.2). Below we propose using one V-cycle of a multigrid method as a preconditioner. Once δu has been computed, both δz and $\delta\lambda$ are easily obtained.

Assuming strict complementarity holds, it is often observed that while the original system (2.4) can be relatively well conditioned (compared to the unconstrained problem), the reduced systems (4.1) and (4.2) are usually highly ill conditioned. The main reason for this problem is the complementarity condition (2.3c). This condition implies that as the outer IP iteration approaches the solution, either z_i or λ_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 [29]. However, ill-conditioning is a source of difficulties when considering large-scale problems, for which iterative methods are required in order to solve the linear systems that arise at each (outer) iteration. Typically, ill-conditioning causes slow convergence of the inner iterative linear solver.

There are three approaches to obtain a well conditioned system. First, and most simple, is to solve the original system (2.4) without the elimination of unknowns.

A second approach is to use partial elimination on (2.4). 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 tend to infinity while the entries of D_1 tend to 0. We can now reorder system

(4.1) as

$$\begin{pmatrix} J_{uu} & -(c_u)_1^\top & -(c_u)_2^\top \\ -(c_u)_1 & -D_1 & 0 \\ -(c_u)_2 & 0 & -D_2 \end{pmatrix} \begin{pmatrix} \delta u \\ \delta\lambda_1 \\ \delta\lambda_2 \end{pmatrix} = - \begin{pmatrix} r_1 \\ g_1 \\ g_2 \end{pmatrix}, \tag{4.3}$$

where $(c_u)_1$ and $(c_u)_2$ consist of different rows of the Jacobian c_u corresponding to the different variables, and g_1 and g_2 contain 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

$$\begin{pmatrix} J_{uu} + (c_u)_2^\top D_2^{-1} (c_u)_2 & -(c_u)_1^\top \\ -(c_u)_1 & -D_1 \end{pmatrix} \begin{pmatrix} \delta u \\ \delta\lambda_1 \end{pmatrix} = - \begin{pmatrix} r_1 - (c_u)_2^\top D_2^{-1} g_2 \\ g_1 \end{pmatrix}. \tag{4.4}$$

Although the system (4.4) does not suffer from the ill conditioning induced by small λ_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 [12].

A third approach to obtain a well conditioned system is by a simple rescaling. Note that the system (4.1) is equivalent to the system

$$\begin{pmatrix} J_{uu} & c_u^\top \Lambda^{\frac{1}{2}} \\ \Lambda^{\frac{1}{2}} c_u & -Z \end{pmatrix} \begin{pmatrix} \delta u \\ \Lambda^{-\frac{1}{2}} \delta\lambda \end{pmatrix} = - \begin{pmatrix} r_1 \\ \Lambda^{\frac{1}{2}} c_u^\top \left(r_2 - \Lambda^{-1} r_3 \right) \end{pmatrix}, \tag{4.5}$$

which (assuming strict complementary) is also well conditioned. Note that the scaling preserves the symmetry of the system. On the other hand, recovering $\delta\lambda$ from the solution of (4.5) requires scaling the second component by $\Lambda^{\frac{1}{2}}$. Since $\Lambda^{\frac{1}{2}}$ will contain some large entries as the IP iteration approaches the solution, this may lead to severe amplification of any errors contained in the computed approximation to $\delta\lambda$, with adverse effects on the overall solution process. For this reason, we only consider system (4.4).

While the original system (2.4) and the partially reduced system (4.4) 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 for the other 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.

4.2. 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 (2.4). Next, we discuss a multigrid method for the fully reduced and ill conditioned system (4.2). 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 partially reduced system (4.4). 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 elliptic PDE. We mention that multigrid methods for PDE-related constrained

optimization problems have been developed by Bank et al. in [1]. See also the recent survey by Gräser and Kornhuber in [14] for work on multigrid methods for obstacle problems, although not in the framework of IP methods.

Before describing our multigrid method, we make a simple but important observation about the variables z and λ. Although these variables have an obvious algebraic interpretation, in our examples they also have interpretation as grid functions. Assuming for example a cell-centered finite volume discretization of $c(u) = u - b$ in Example 1, it is easy to see that both z and λ are defined at precisely the same spatial locations as the primal variable u. For more complicated constraints such as the ones in Example 2, however, one has also to consider the Jacobian of the constraint operator and its discretization. This matrix will generally be rectangular (non-square), hence u, z and λ will generally belong to different spaces. This will generally increase the complexity of a coupled multigrid approach, at least to some extent.

4.2.1. The h-ellipticity of the systems. Following [28] we say that a system $Au = b$ arising from the discretization of a PDE is *h-elliptic* if its symbol $\tilde{A}$ satisfies

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

where ρ is a constant independent of h and T^{high} is the range of high grid frequencies. For regular coarsening, $T^{\text{high}} := [-\pi,\pi)^2 \backslash [-\pi/2,\pi/2)^2$. For an effective multigrid method to be feasible, we need to have an h-elliptic system; see [28, page 121]. 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.

For problems with simple bound constraints on the unknowns, like the one in Example 1, h-ellipticity can be easily established as follows. Let $\mathcal{J}_{uu}$ denote the operator corresponding to the second variation of the functional $\mathcal{J}$. Assuming that the discretization J_{uu} of the differential operator $\mathcal{J}_{uu}$ is h-elliptic, we have the following result.

PROPOSITION 4.2. *Let $\widehat{\mathcal{J}}_{uu}(\theta)$ be the symbol of $\mathcal{J}_{uu}$ and assume that it is h-elliptic. Then the system* (2.4) *is also h-elliptic.*

Proof. The symbol of the system (2.4) is the matrix

$$\widehat{H}(\theta) = \begin{pmatrix} \widehat{\mathcal{J}}_{uu}(\theta) & -\widehat{c}_u^* & 0 \\ -\widehat{c}_u & 0 & 1 \\ 0 & \zeta & \eta \end{pmatrix}$$

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

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

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

$$\frac{\min_{\theta \in T^{\text{high}}} |\widehat{\mathcal{J}}_{uu}(\theta)|}{\max_{\theta \in [-\pi,\pi)} |\widehat{\mathcal{J}}_{uu}(\theta)|} \ge \rho > 0$$

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

$$\frac{\min_{\theta \in T^{\text{high}}} |\widehat{\mathcal{J}}_{uu}(\theta) + \eta\zeta^{-1} c_u^* c_u|}{\max_{\theta \in [-\pi,\pi)} |\widehat{\mathcal{J}}_{uu}(\theta) + \eta\zeta^{-1} c_u^* c_u|} > \rho\,.$$

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 λ and z vary in magnitude. However, the effectiveness of a multigrid scheme can be ascertained empirically by means of numerical experiments.

REMARK 4.3. *It is important to note that for simple model problems like Example 1, 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.*

4.2.2. Multigrid components. Here we give a brief discussion of the basic ingredients of the multigrid schemes used in the paper. We have chosen to use a simple geometric multigrid approach. This may not be the best choice for problems with jumping coefficients [28], but it is straightforward to implement. It is possible to derive more sophisticated (and more expensive) multigrid methods for a class of more difficult problems with jumping coefficients [21].

Smoothing: Since we are assuming that J_{uu} is h-elliptic, we can use pointwise relaxation for the smoothing of the reduced system (4.2). For the coupled system (2.4) 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, λ 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 (4.2) or the full system (2.4).

4.3. Solving the partially reduced system. While it is straightforward to develop multigrid methods for the systems (4.2) and (2.4), it is more difficult to develop such a method for system (4.4). 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 [10] (see also [3]) we consider a block preconditioner of the form

$$M = \begin{pmatrix} A & 0 \\ 0 & \widehat{S} \end{pmatrix} \tag{4.6}$$

where

$$\begin{aligned} A &\approx J_{uu} + (c_u)_2^\top D_2^{-1}(c_u)_2\,, \\ \widehat{S} &\approx (c_u)_1(J_{uu} + (c_u)_2^\top D_2^{-1}(c_u)_2)^{-1}(c_u)_1^\top + D_1\,. \end{aligned}$$

Here A need not be an explicitly formed matrix; rather, a prescription is given for computing (or approximating) 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.

For the minimal surface problem (Example 1) $(c_u)_1 = I_1$, which is a matrix consisting of a subset of rows of the identity matrix. In this case, 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

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

which has very low set-up and application costs. The above preconditioner is used within a MINRES iteration [26].

5. 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 μ 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 [7]).

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 μ 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 [22] and artificial compressibility is added to unstable discretizations of the Stokes problem [15].

Similar to [1], 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 5.1.

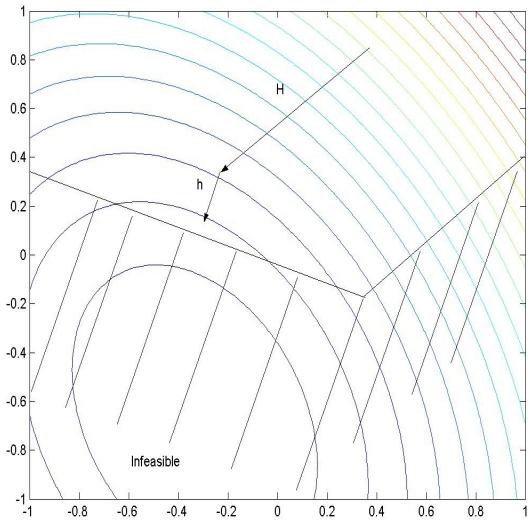


FIG. 5.1. *Advancement in the interior point as a function of the grid size.*

Using the theorem in [24, page 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 μ is $\mathcal{O}(h^p)$ we can modify the line search so as to guarantee that 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.

6. Numerical experiments. We have used two examples to demonstrate the concepts in this work. First, the minimal surface problem has been studied. This problem is well understood and there are many good algorithms to solve it: in particular, approaches for bound constrained optimization based on projected gradient/Newton method. Our focus here is not on competing with other algorithms, but on using this relatively simple problem as a test problem for our approach. The second example is the registration problem, where the constraint is no longer a simple bound. In this case simple projection methods do not work well, whereas interior point methods can be quite efficient.

6.1. The minimal surface problem. Recall that in the minimal surface problem (Example 1) we seek a function u that lies above a function b (the "obstacle") and minimizes the energy functional defined in (1.2). In this section we present the results of numerical experiments with the minimal surface problem for a few different choices of the obstacle function b and domain Ω, described below.

- **Non-smooth obstacle:** This is a frequently used test problem; see, for instance, Example 17 in [9]. Here $\Omega = [0,1] \times [0,1]$, and the obstacle b, shown in Figure 6.1(a), is defined as

$$b(x_1, x_2) = \begin{cases} 1 & \text{if} \quad |x_1 - .5| < .25 \quad \text{and} \quad |x_2 - .5| < .25, \\ 0 & \text{otherwise} . \end{cases} \tag{6.1}$$

- **Smooth obstacle:** As in the non-smooth obstacle problem, let $\Omega = [0,1] \times [0,1]$. Define the obstacle b, shown in Figure 6.1(b), as

$$b(x_1, x_2) = \max\left\{3\exp\left(-5(x_1 - .5)^2 - 5(x_2 - .5)^2\right), 0\right\}. \tag{6.2}$$

- **Multiply connected domain:** Now consider a problem with a multiply connected domain. Let $\Omega^0 = [.25, .5] \times [.25, .5]$, let $\Omega^1 = [0,1] \times [0,1]$, and define the domain $\Omega = \Omega^1 \setminus \Omega^0$. Define the obstacle b, shown in Figure 6.1(c), as

$$b(x_1, x_2) = \max\left\{3\exp\left(-10(x_1 - .5)^2 - 100(x_2 - .8)^2, 0\right)\right\}. \tag{6.3}$$

For the discretization we use cell-centered finite differencing. For the simply connected domain $\Omega = [0,1] \times [0,1]$, we divide Ω into $n = N^2$ cells, each with width $h = 1/N$, and place the grid function u in cell centers. Next, we use central difference matrices D_1 and D_2, which approximate the derivatives on the cell edges in the x_1 and x_2 directions, respectively, to discretize the gradient ∇u. Finally, we define averaging matrices A_1 and A_2 to average cell edge quantities into cell centers. Then we can write the discrete formulation of the minimal surface problem in the form (1.3).

We can extend the cell-centered discretization to the multiply connected domain, condensing D_1 and D_2 to exclude points inside Ω^0, the "hole" in the domain. Note that, in the case of the multiply connected domain, the vectors u and b are of length less than N^2 to account for the area removed from the unit square domain.

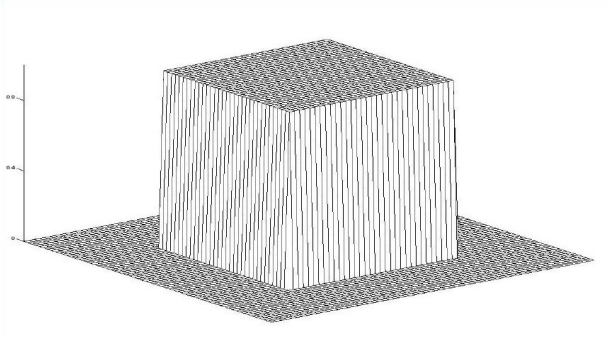
Next, the inexact Newton algorithm described in Algorithm 1 is used to solve the discrete constrained optimization problem. Note that, using the above finite differencing approach, it is possible to perform a direct computation of the Jacobian J_u and of the Hessian J_{uu}. Indeed, the Jacobian and the Hessian can be written by computing first

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

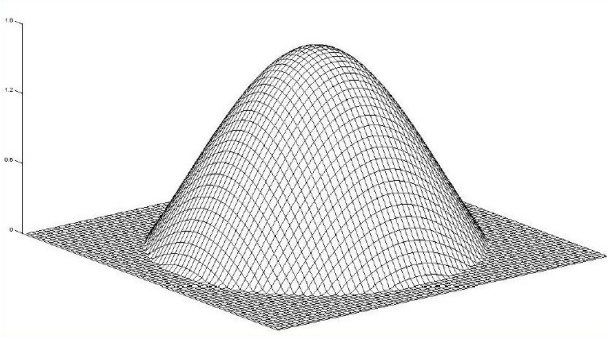
and then setting

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

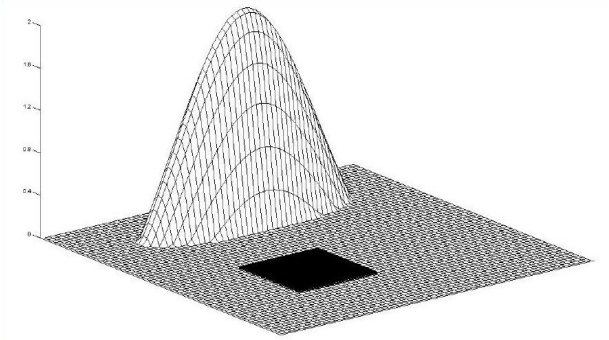
It is worth noting that the matrix J_{uu}^{+} is SPD everywhere and that it approximates a differential operator of the form $J_{uu}^{+} \approx -\nabla \cdot (\sigma(u)\nabla)$. Furthermore, note that $\sigma(u)$ may not vary smoothly if ∇u is large. Note also that J_{uu}^{-} tends to make the Hessian "less positive definite" (more nearly singular) 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.



(a) Non-smooth obstacle function



(b) Smooth obstacle function



(c) Multiply connected domain

FIG. 6.1. *Obstacle functions $b(x_1, x_2)$ used in numerical experiments*

6.1.1. Computational results. We begin by comparing the multilevel continuation approach with the one-grid approach, where the one-grid approach only computes the solution to the fine grid problem. Next, we experiment with each of

the three formulations (2.4), (4.2), and (4.4) of the linear system, using different solvers and preconditioners appropriate for each system. We examine the linear system solvers and preconditioners by testing their performance at each Newton (outer) iteration, using the fairly stringent stopping criterion of $\|r_k\|_2 < 10^{-8}\|r_0\|_2$, where r_k is the residual of the linear system (2.4), (4.2), or (4.4), at the k^{th} linear (inner) iteration.

6.1.2. Experiments with multilevel continuation. Consider the multilevel continuation approach which uses a coarse grid solution as an approximation to a fine grid solution. To illustrate the effects of multilevel continuation, consider the example of solving the problem with the non-smooth obstacle function defined in (6.1). We solve the minimal surface problem on a 16×16, 32×32, 64×64, and 128×128 grid, moving from a coarse level to a fine level once the coarse grid problem is solved to a relative tolerance of 10^{-3}. In other words, at the j-th Newton iteration, we calculate the 2-norm of the gradient of the Lagrangian (defined in (2.2)), say $\|R_j\|_2$, and we stop the coarse grid algorithm if $\|R_j\|_2 < 10^{-3}\|R_0\|_2$. The barrier parameter μ is set to $\mu(h) = h^p$, where p is the accuracy of the discretization, as described in section 5. Here, $p = 2$. We choose the feasible solution $u(x_1, x_2) = 5\sin(\pi x_1)\sin(\pi x_2)$ as the initial guess, and we solve the full system (4.1) using GMRES preconditioned with one V-cycle of the multigrid method described in section 4.2.

To demonstrate the results of multilevel continuation, Figure 6.2 displays the solution on different grids. Recall that multilevel methods use the interpolated solution on a coarser grid as the initial guess on a finer grid.

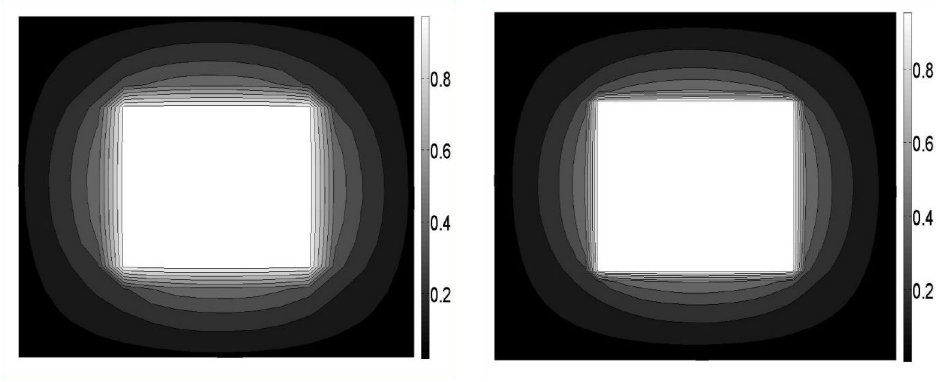
Table 6.1 displays Newton iteration counts on each grid using the multilevel continuation method described in section 5. Observe that a slight increase in Newton iterations occurs as we refine the grid. This is likely due to the fact that we are solving a slightly different optimization problem on each grid. In particular, we choose the barrier parameter μ to be dependent on the grid size h (to wit, $\mu = h^2$), causing the problem to change when the grid changes.

If we do not apply a multilevel continuation method, and instead solve the problem exclusively on the 128×128 grid, convergence cannot be achieved within 300 Newton iterations. It is clear that a multilevel method is crucial for this problem. The coarse grid problems are significantly less expensive to solve, and the residual on the first Newton iteration on the 128×128 grid after multilevel continuation is already less than 10^{-4}. Applying multilevel continuation, we have transformed a method that fails to compute a solution into a powerful method for this problem.

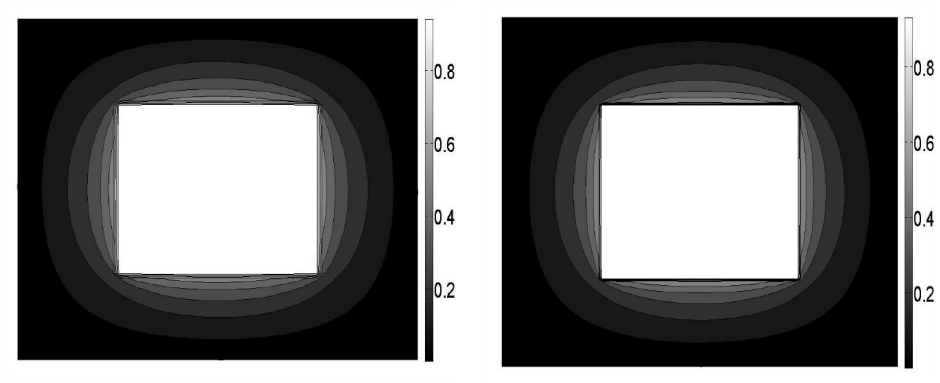
Figure 6.3 displays the 2-norm of the relative residual of the system (2.4) at each of the outer iterations for different levels. We see that by using coarse grids we are able to reduce the relative 2-norm of the residual of the fine grid to less than 10^{-4}.

Very similar results were obtained when using multilevel continuation on the two other minimal surface problems (smooth obstacle and multiply connected domain). The results are summarized in Table 6.1.

6.1.3. Experiments with the full system. Now we turn to techniques to solve the linear system arising at each Newton iteration. Consider solving the full system (2.4) at each Newton iteration in the process of solving the minimal surface problem in each of our three examples. We applied GMRES without restarting, preconditioned using one multigrid V-cycle. Recall that we choose to apply simple geometric multigrid, with block SGS as the smoother and linear prolongation and restriction. Also, we set the stopping criterion so that GMRES stops when the relative residual of the linear system (2.4) reaches a low tolerance of 10^{-8}. As a result, an



(a) 16×16 grid (b) 32×32 grid



(c) 64×64 grid (d) 128×128 grid

FIG. 6.2. *Solutions $u(x_1, x_2)$ at the final Newton iteration on different grids using multilevel continuation, non-smooth obstacle function*

exact Newton method is essentially carried out, ensuring in theory that that the outer convergence is quadratic. While we can expect quadratic outer convergence, setting a low inner tolerance leads to a more expensive linear system solve.

Table 6.1 displays GMRES iteration counts when solving the full system (2.4). We show GMRES iteration counts at the first Newton iteration, the last Newton iteration, and the average number of GMRES iteration counts per Newton iteration. All three examples, described in section 6, are included. Clearly, the multigrid preconditioner for the full system is an effective approach for solving the minimal surface problem. A slight increase in the number of iterations 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 $\nabla \cdot \sigma \nabla$ where, towards convergence, σ contains jumps. Thus, any purely geometric multigrid is expected to produce similar results.

6.1.4. Experiments with the fully reduced system. Next, consider solving the fully reduced system (4.2). In this case, the linear system we solve at each Newton

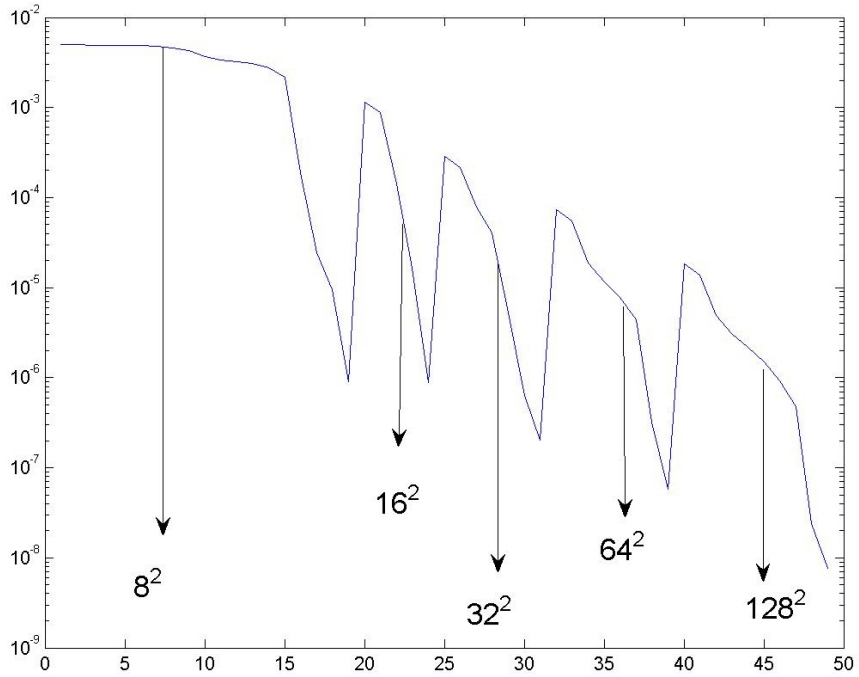


Fig. 6.3. *Relative residual at each outer iteration for different grid sizes.*

Example	Grid Size	Newton Iterations	GMRES Iterations (First)	(Last)	(Average)
Non-smooth obstacle function	16×16	7	5	5	5.0
	32×32	10	6	7	6.8
	64×64	13	8	10	9.3
	128×128	16	11	13	12.3
Smooth obstacle function	16×16	9	5	5	5.0
	32×32	13	7	6	6.2
	64×64	17	9	7	7.8
	128×128	20	10	10	10.2
Multiply connected domain	16×16	15	5	4	4.4
	32×32	16	7	5	5.6
	64×64	25	9	6	6.8
	128×128	26	14	8	9.9

Table 6.1

Newton and GMRES iteration counts for the three obstacle problems. Full GMRES with a multigrid preconditioner is used to solve the full system (2.4)*. Stopping tolerances are* 10^{-5} *for Newton and* 10^{-8} *for GMRES.*

iteration is SPD. We can therefore apply efficient solvers like the conjugate gradient method. In our experiments we use the preconditioned conjugate gradient method with one V-cycle of multigrid as the preconditioner. Once again, we choose simple geometric multigrid, with an SGS smoother and linear prolongation and restriction.

Example	Grid Size	Newton Iterations	PCG Iterations (First)	(Last)	(Average)
Non-smooth obstacle function	16×16	7	5	5	5.0
	32×32	10	7	7	7.0
	64×64	13	10	10	10.0
	128×128	16	14	13	13.2
Smooth obstacle function	16×16	9	6	5	5.3
	32×32	13	7	7	6.9
	64×64	17	9	9	8.8
	128×128	20	11	11	10.5
Multiply connected domain	16×16	15	6	5	5.3
	32×32	16	7	6	6.6
	64×64	25	8	7	7.4
	128×128	26	11	8	9.1

TABLE 6.2

Newton and PCG iteration counts for the three obstacle problems. PCG with a multigrid preconditioner is used to solve the full system (4.2). *Stopping tolerances are* 10^{-5} *for Newton and* 10^{-8} *for PCG.*

We still set the linear solver tolerance to 10^{-8}, which (in theory) guarantees quadratic Newton convergence at the expense of accurate linear system solves.

Table 6.2 displays PCG iteration counts when solving the fully reduced system (4.2) for each of the three examples. Like before, we show PCG iteration counts at the first Newton iteration and the last Newton iteration, as well as the average number of PCG iteration counts per Newton iteration. As in the case of the full system, the simple geometric multigrid solver proves to be an effective method for the minimal surface problem. We still experience an increase in the linear solver iteration count as the grid is refined, which can again be explained by the jumping coefficients in the Hessian J_{uu}.

Comparing Tables 6.1 and 6.2, we can see that the iteration counts for the linear system solver are similar in both approaches. However, experimentation indicates that the fully reduced system approach is significantly faster (by an order of about 10) than the full system approach in terms of CPU time for the minimal surface problem examples. While rigorous testing with highly optimized code has not been performed, we expect the fully reduced system approach to be the optimal approach in terms of iteration counts and CPU time.

6.1.5. Experiments with the partially reduced system. Finally, consider solving the partially reduced system (4.4) on the problem with the non-smooth obstacle function, (6.1). Observe that the coefficient matrix is symmetric and indefinite. A natural choice of solver for systems of this type is the Krylov subspace method MINRES [26]. Attempting MINRES on (4.4) without preconditioning we cannot obtain convergence to a relative linear system residual tolerance of 10^{-8} within 1000 iterations on the 128×128 grid. Hence, preconditioning is essential in the solution of such systems. We performed some experiments with the block preconditioner (4.6) proposed in section 4.3. This approach did not perform satisfactorily. We experienced slow convergence of the Krylov solver using (4.6) as the preconditioner, even with exact solves for the $(1,1)$-block. This implies that we have a poor approximation with

the $(2,2)$-block, which involves the Schur complement. A better approximation to the Schur complement is clearly needed in order to make this approach competitive with the other approaches based on solving the full system and the fully reduced system.

Interestingly, a rather crude technique produced the most successful results for the partially reduced system. We performed a symmetric diagonal scaling of the coefficient matrix in (4.4), followed by an incomplete LU factorization with a drop tolerance of 10^{-2}, applied as a preconditioner for GMRES. For the final Newton iteration on the 128×128 grid, this preconditioned GMRES approach converged to a tolerance of 10^{-3} in 17 iterations, and converged to a tolerance of 10^{-8} in 57 iterations. Unfortunately, approaches involving incomplete LU factorizations do not scale well as $h \to 0$, and therefore this approach for solving the partially reduced system cannot be recommended for larger problems or higher dimensional domains.

As in the experiments with the non-smooth obstacle example, the partially reduced method did not fare well on the problems with the smooth obstacle function or the multiply connected domain. Very similar iteration counts and output occurred with all three example minimal surface problems.

Examining all of our numerical results, we can draw some conclusions about solution techniques for the minimal surface problem and similar ones. First, a multilevel approach can and should be applied to bound constrained optimization problems involving objective functions with differential operators. We are able to avoid the problem that arises from multilevel continuation in interior point methods by allowing the barrier parameter μ to be dependent on the accuracy of the discretization. In this way, we achieve significant cost reduction when using coarser grid information on a finer grid problem.

The linear system solver experiments led to conclusions about the different approaches considered in this work. We obtained low iteration counts and good scalability when we applied GMRES to the full system (2.4) with a multigrid preconditioner, as well as when we applied PCG to the fully reduced system (4.2) with a multigrid preconditioner. We conclude from experimentation that the fully reduced system approach is optimal in terms of both iteration counts and CPU time. We have yet to find a preconditioning technique to solve the partially reduced system (4.4) efficiently.

6.2. Volume-constrained registration. In the registration problem (1.5) the unknown quantity is a vector field $\mathbf{u} = [u_1, u_2]$. It is important to note that the problem is akin to the Stokes problem and therefore a similar discretization is required [28]. To this end we use a staggered orthogonal grid for the components of $\mathbf{u}$ where u_1 is associated with the x edges of each cell and u_2 is associated with the y edges of each cell, as shown in Figure 6.4 (see [17] for a detailed discussion of the discretization). This leads to the compact discretization of the constraint

$$b \leq B_h u_h,$$

where B_h is the discrete equivalent of the divergence operator. Similarly, the gradient is discretized by short differences and we obtain the discrete problem (1.5). Since the problem is nonlinear due to the term $D_h(u_h)$, we build a linear approximation to D_h and study the constrained quadratic problem with a random starting vector. A random starting vector generates a rather erratic right-hand side to the system and thus truly puts our iterative linear solver to the test. In our test problem, $\beta = 10^{-3}$. The choice of η, μ, and the stopping tolerances are the same as in the case of the obstacle problem.

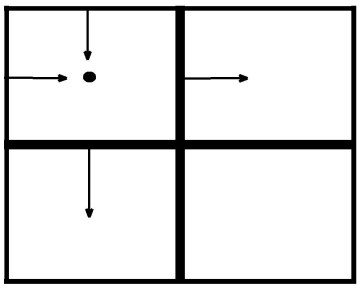


FIG. 6.4. *The staggered grid used for the discretization of the problem. The u variables are on the faces of each cell while the Lagrange multipliers and the slack variables are in cell-centers.*

Grid Size	Newton Iterations	GMRES Iterations (First)	(Last)	(Average)
16×16	42	3	3	3.0
32×32	32	4	4	3.7
64×64	19	6	5	5.9
128×128	21	7	6	6.8

TABLE 6.3

Iteration counts for the volume-constrained problem on different grids. Full GMRES with multigrid preconditioner is applied to system (2.4).

To efficiently solve the linear system at each iteration we use a multigrid method. Developing a multigrid method for the reduced problems presents a serious challenge. This is because after some of the unknowns are eliminated, the remaining unknowns can end up on a regular mesh with "holes." Thus, for this application the full system (2.4) is not only better conditioned, it is also easier to deal with from a multigrid perspective. Hence, we only report results for (2.4).

As for any multigrid algorithm, we require a smoother for the problem. It is important to note that since the unknowns do not reside in the same location, simple, point-wise smoothing is not advised [28]. Therefore, similar to many applications in Computational Fluid Dynamics (CFD), we use a Vanka-like smoother where each cell in the computational grid is relaxed sequentially [28]. Consider the cell in Figure 6.4. This cell has 6 variables associated to it. In a Vanka-like smoother we relax all these unknowns simultaneously by solving a 6×6 linear system. We use simple linear interpolation and Galerkin coarsening.

In Table 6.3 we record the number of iterations needed for each grid. One can see that the number of Newton iterations stays bounded (it actually decreases as the mesh is refined), while the multigrid solver shows a very slow growth in the number of iterations as $h \to 0$. The combined effect is an essentially mesh-independent behavior of the solver.

Since $\mathbf{u}$ is a vector field we plot in Figure 6.5 the constraint, div$\mathbf{u}$ (which is a scalar field), on each of the meshes.

7. Conclusions. We have explored a new multilevel approach for variational problems with inequality constraints in the context of interior point methods and

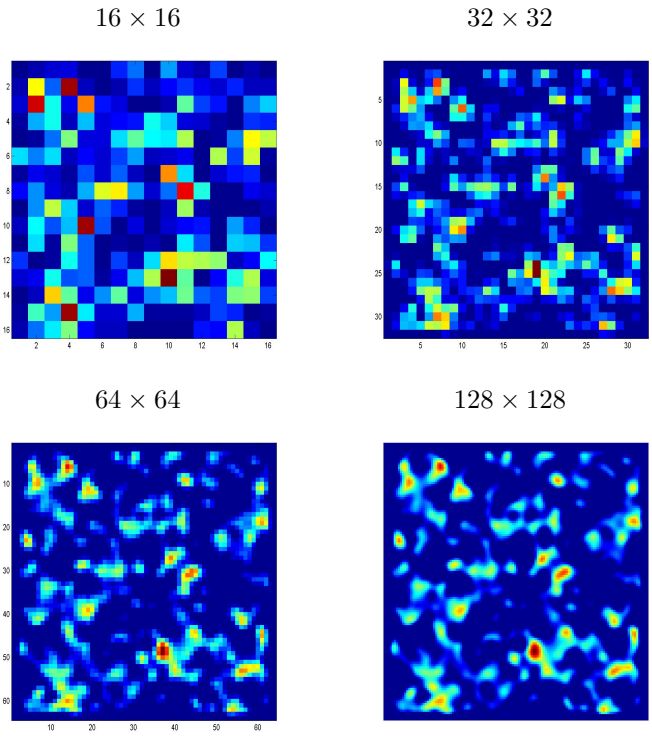


Fig. 6.5. *The divergence of* $\mathbf{u}$ *for different mesh sizes.*

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 μ can dramatically improve the results of an interior point algorithm.

For the simpler case of bound constraints (as in the minimal surface problems), solving the fully reduced system with a multigrid-preconditioned conjugate gradient method proved to be the most efficient approach in terms of CPU time. The second best approach was solving the full system (2.4) with GMRES preconditioned with a multigrid method with symmetric block Gauss–Seidel ("box") smoothing. 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.

For the case of more complicated constraints (as in the volume-constrained registration problem), excellent results were obtained solving the full linear system (2.4) with GMRES preconditioned by a multigrid method with Vanka-like smoothing. This is appropriate since the primal variable in this problem is the discretization of a vector field, akin to the velocity field in CFD. On the other hand, the development of efficient multigrid solvers for the reduced systems remains a challenge.

REFERENCES

[1] R. E. Bank, P. E. Gill, and R. F. Marcia, *Interior methods for a class of elliptic variational inequalities*, in *Large-Scale PDE-Constrained Optimization* (Santa Fe, New Mexico, 2001), Vol. 30 of Lecture Notes in Computational Science and Engineering, Springer, Berlin, pp. 218–235, 2003.

[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] G. Biros and O. Ghattas, *Parallel Lagrange-Krylov-Newton-Schur methods for PDE-constrained optimization. Parts I-II*, SIAM J. Sci. Comput., 27 (2005), pp. 687–738.
[5] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, UK, 2004.
[6] R. Byrd, J. Nocedal, and R. Waltz, *Steering penalty methods*, Optimization Methods & Software, 23 (2008), pp. 197–213.
[7] 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.
[8] 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.
[9] D. E. Dolan and J. J. Moré, *Benchmarking Optimization Software with COPS*, Tech. Report ANL/MCS-246, Argonne National Laboratory, Argonne, IL, November 2000.
[10] 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.
[11] R. W. Freund and N. M. Nachtigal, *Software for simplified Lanczos and QMR algorithms*, Appl. Numer. Math., 19 (1994), pp. 319–341.
[12] 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.
[13] R. Glowinski, *Numerical Methods for Nonlinear Variational Problems*, Springer Series in Computational Physics, New York/Berlin/Heidelberg/Tokyo, 1984.
[14] C. Gräser and R. Kornhuber, *Multigrid methods for obstacle problems*, J. Comput. Math., 27 (2009), pp. 1–44.
[15] 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.
[16] E. Haber, U. M. Ascher, and D. Oldenburg, *On optimization techniques for solving nonlinear inverse problems*, Inverse Problems, 16 (2000), pp. 1263–1280.
[17] E. Haber and J. Modersitzki, *Numerical methods for volume preserving image registration*, Inverse Problems, 20 (2004), pp. 1621–1638.
[18] S. Haker, L. Zhu, A. Tannenbaum, and S. Angenent, *Optimal mass transport for registration and warping*, Int. J. Computer Vision, 60 (2004), pp. 225–240.
[19] M. Heinkenschloss, *Projected sequential quadratic programming methods*, SIAM J. Optim., 6 (1996), pp. 373–417.
[20] M. Hintermüller and M. Ulbrich, *A mesh independence result for semismooth Newton methods*, Math. Progr., 101 (2004), pp. 151–184.
[21] S. Knapek, *Matrix-dependent multigrid homogenization for diffusion problems*, SIAM J. Sci. Comp., 20 (1999), pp. 515–533.
[22] R. J. LeVeque, *Numerical Methods for Conservation Laws*, Birkhäuser–Verlag, Basel, 1990.
[23] J. Nocedal, A. Waechter, and R. Waltz, *Adaptive barrier update strategies for nonlinear interior methods*, SIAM J. Optim., 19 (2009), pp. 1674–1693.
[24] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, New York, 1999.
[25] C. W. Oosterlee, *On multigrid for linear complementarity problems with applications to American-style options*, Elecr. Trans. Numer. Anal., 15 (2003), pp. 165–185.
[26] C. C. Paige and M. A. Saunders, *Solution of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal., 12 (1975), pp. 617–629.
[27] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 7 (1986), pp. 856–869.
[28] U. Trottenberg, C. W. Oosterlee, and A. Schuller, *Multigrid*, Academic Press, New York, 2001.
[29] M. H. Wright, *Ill-conditioning and computational error in interior methods for nonlinear programming*, SIAM J. Optim., 9 (1999), pp. 84–111.
[30] S. J. Wright, *Primal-Dual Interior Point Methods*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1997.