

Interior Methods For a Class of Elliptic Variational Inequalities

Randolph E. Bank, Philip E. Gill, and Roummel F. Marcia*

Department of Mathematics, University of California, San Diego, La Jolla, CA

Abstract. We consider the application of primal-dual interior methods to the optimization of systems arising in the finite-element discretization of a class of elliptic variational inequalities. These problems lead to very large (possibly non-convex) optimization problems with upper and lower bound constraints.

When interior methods are applied to the discretized problem, the resulting linear systems have the same zero/nonzero structure as the finite-element equations solved for the unconstrained case. This crucial property allows the interior method to exploit existing efficient, robust and scalable multilevel algorithms for the solution of partial differential equations (PDEs).

We illustrate some of these ideas in the context of the elliptic PDE package PLTMG.

1 Introduction

We consider a class of elliptic partial differential equation (PDEs) for which the solution is required to satisfy certain inequality constraints (for example, so-called obstacle problems in the class of elliptic variational inequalities [19,20,18,21,22]). The proposed method combines an adaptive finite-element method with a finite-dimensional primal-dual interior method for optimization.

As background for the problem to be discussed, we introduce the self-adjoint, positive-definite elliptic boundary value problem:

$$\begin{aligned} -\nabla \cdot (a \nabla u) + bu &= f \text{ for } x \in \Omega, \\ (a \nabla u) \cdot n &= g \text{ for } x \in \partial\Omega_1, \\ u &= 0 \text{ for } x \in \partial\Omega_2 \equiv \partial\Omega - \partial\Omega_1. \end{aligned} \tag{1}$$

Here, $a(x)$ and $b(x)$ are smooth functions with $a(x) > 0$ and $b(x) \geq 0$, but more generally we could let $a(x)$ and $b(x)$ be symmetric positive-definite and positive semidefinite matrix functions. It is assumed throughout that a unique solution exists (which will follow, for example, if $b > 0$ or $\partial\Omega_2$ is nonempty). The methods to be considered exploit the so-called weak formulation of problem (1) and hence require the definition of certain function spaces. Let

$$H \equiv \mathcal{H}_E^1(\Omega) = \left\{ u : \int_{\Omega} |\nabla u|^2 dx + \int_{\Omega} u^2 dx < \infty \text{ and } u = 0 \text{ on } \partial\Omega_2 \right\}.$$

For $u \in H$, the energy inner-product norm is $\|u\|^2 = a(u, u)$, where $a(u, v)$ denotes the operator

$$a(u, v) = \int_{\Omega} a(x) \nabla u \cdot \nabla v dx + \int_{\Omega} b(x) uv dx, \quad u, v \in H.$$

For $u, v \in \mathcal{L}^2(\Omega)$, the standard inner-product norm is $\|u\|^2 = (u, u)$, where $(u, v) = \int_{\Omega} uv dx$. Similarly, on the boundary we use the inner product $\langle u, v \rangle = \int_{\partial\Omega_1} uv ds$ for $u, v \in \mathcal{L}^2(\partial\Omega_1)$.

The Ritz variational formulation of (1) involves the minimization problem

$$\underset{u \in H}{\text{minimize}} \quad q(u) = a(u, u) - 2\{(f, u) + \langle g, u \rangle\}. \tag{2}$$

* Research supported by National Science Foundation grants DMS-9973276 and ACI-0082100.

Many nonlinear PDE's have a similar variational formulation. However, unlike (2), the objective function $q(u)$ is not necessarily quadratic in u .

Let $\mathcal{S} \subset H$ be the n dimensional space of continuous piecewise-linear polynomials corresponding to a triangulation \mathcal{T} of Ω . The finite-element approximation $u_h \in \mathcal{S}$ to $u \in H$ solves the minimization problem:

$$\underset{u_h \in \mathcal{S}}{\text{minimize}} \quad q(u_h). \quad (3)$$

Let $\{\phi_i\}_{i=1}^n$ denote the usual nodal basis for \mathcal{S} . Any $u_h \in \mathcal{S}$ can be written in the form $u_h = \sum_{i=1}^n U_i \phi_i$, which implies that (3) can be formulated as the finite-dimensional minimization problem

$$\underset{U \in \mathbb{R}^n}{\text{minimize}} \quad Q(U).$$

This problem is usually solved using Newton's method. Given an approximate solution U , a direction of improvement ΔU is computed from the Newton equations $\nabla^2 Q(U) \Delta U = -\nabla Q(U)$, where $\nabla Q \in \mathbb{R}^n$ is the gradient and $\nabla^2 Q \in \mathbb{R}^{n \times n}$ is the Hessian of Q . The new estimate is then $U + \alpha \Delta U$, where α is a step length used to enforce convergence. In the case of the linear PDE (1), the objective is quadratic, with

$$Q(U) = U^T A U - 2U^T F,$$

where A is the sparse symmetric positive-definite stiffness matrix with $A_{ij} = a(\phi_j, \phi_i)$, and $F_i = (f, \phi_i) + \langle g, \phi_i \rangle$. In this case, the optimal U can be computed from the single sparse symmetric system $AU = F$. (Matrices with similar nonzero structure arise in the nonlinear case, but the entries of the stiffness matrix generally depend upon the current value of U .) Such linear systems are solved using iterative methods. In particular, the finite-element code PLTMG [1] uses the conjugate-gradient method with a preconditioner based on the multilevel multigraph technique [2]. Multigraph preconditioners are related to hierarchical basis multigrid preconditioners, which have proved to be very robust for problems posed on a sequence of nonuniform, adaptively refined meshes. However, multigrid methods rely on the refinement structure generated through the adaptive refinement process. This limits their applicability when geometrically complex domains require many elements just for the definition of the domain, or when adaptivity is derived from moving mesh points rather than refinement. In these situations, it is possible to have fine, highly nonuniform meshes with no refinement history available to create a hierarchical basis. Multigraph methods overcome this limitation by creating a hierarchical basis when no natural geometry is present. The reader is referred to [2] for a detailed description of the multigraph preconditioner used in PLTMG. It suffices here to note that the preconditioner considered in this paper involves incomplete sparse factorizations of A and its restriction to a sequence of subspaces of decreasing dimension.

The purpose of this paper is to extend the multigraph finite-element method to the case where u is subject to inequality constraints. In this case, the continuous problem is

$$\underset{u \in H}{\text{minimize}} \quad q(u), \quad \text{subject to} \quad \underline{b} \leq u \leq \bar{b}. \quad (4)$$

If \mathcal{I} denotes the interpolation operator, then the finite-element formulation of this problem can be written as

$$\underset{u_h \in \mathcal{S}}{\text{minimize}} \quad q(u_h), \quad \text{subject to} \quad \mathcal{I}(\underline{b}) \leq u_h \leq \mathcal{I}(\bar{b}).$$

This leads to the finite-dimensional optimization problem

$$\underset{U \in \mathbb{R}^n}{\text{minimize}} \quad Q(U), \quad \text{subject to} \quad \underline{B} \leq U \leq \bar{B}, \quad (5)$$

where \underline{B} and \bar{B} define the expansions $\mathcal{I}(\underline{b}) = \sum_{i=1}^n \underline{B}_i \phi_i$, and $\mathcal{I}(\bar{b}) = \sum_{i=1}^n \bar{B}_i \phi_i$. Many methods have been proposed for solving problems in this form. These methods may be broadly categorized as active-set methods (see, e.g., [3,4,8,23]) and *interior methods* (see, e.g., [9,7,17,27]). As in the unconstrained case, the quadratic form of the objective can be exploited (see, e.g., [24,11]).

2 Solving the Finite-Dimensional Problem

In this section we focus on the solution of the finite-dimensional bound-constrained problem (5). Accordingly, we change our notation to that used in the optimization literature. In particular, we write the problem (5) in the form

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x) \quad \text{subject to} \quad x \geq 0, \quad (6)$$

where $f(x)$ is a general smooth nonlinear function of n variables x_1, x_2, \dots, x_n . For simplicity, we start with a problem with only nonnegativity constraints. Throughout, we will assume that f is twice-continuously differentiable, with gradient $\nabla f(x)$ and Hessian $\nabla^2 f(x)$ denoted by $g(x)$ and $H(x)$ respectively. Since the original continuous problems are usually convex, we will assume that $f(x)$ is a convex function. However, we emphasize that all the methods to be discussed can be extended to the nonconvex case.

The first-order necessary conditions for a solution of (6) are that there exist nonnegative x_i and z_i (Lagrange multipliers) such that $z_i = g_i(x)$ and $x_i z_i = 0$. In this section we adopt a convention common to the literature of interior-point methods of using upper-case letters to denote diagonal matrices whose diagonal consists of the components of the vector represented by the corresponding lower-case letter. With this convention, we can write the first-order conditions in the compact form

$$\begin{aligned} z &= g(x), & z &\geq 0, \\ Xz &= 0, & x &\geq 0. \end{aligned} \quad (7)$$

The pair (x, z) satisfies the property of strict complementarity if $x + z > 0$, i.e., one of the x_i and z_i is nonzero for each i . If both x_i and z_i are zero, a solution is said to be degenerate, or more precisely, dual degenerate. Second-order sufficient conditions are that (7) hold; x and z are strictly complementary and $XHX + Z$ is positive definite. At any point satisfying the first-order conditions, an active nonnegativity constraint (i.e., a variable on its lower bound of zero) causes the corresponding row and column of XHX to be zero.

Primal-dual interior methods are based on solving a system of $2n$ nonlinear equations that represent the first-order optimality conditions (7) with each condition $x_i z_i = 0$ perturbed by a scalar μ (known as the *barrier parameter*). For a given μ , the equations are written as $G^\mu(x, z) = 0$, where

$$G^\mu(x, z) = \begin{pmatrix} g(x) - z \\ Xz - \mu e \end{pmatrix}, \quad (8)$$

where e is the vector of ones. Let (x, z) be an *interior point*, i.e., $x > 0$ and $z > 0$. For a given μ , the equations $G^\mu(x, z) = 0$ are solved using a form of Newton's method in which x and z are maintained to be interior. Linearizing the perturbed conditions (8) at an interior point (x, z) gives

$$\begin{aligned} \Delta z &= g(x) + H(x)\Delta x - z \\ X\Delta z + Z\Delta x &= \mu e - Xz, \end{aligned}$$

where $Z = \text{diag}(z_1, z_2, \dots, z_n)$. This yields the linear system

$$\begin{pmatrix} H & -I \\ Z & X \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta z \end{pmatrix} = - \begin{pmatrix} g - z \\ X(z - \pi) \end{pmatrix}, \quad (9)$$

where π is the vector of *primal multipliers* such that $\pi = \mu X^{-1}e$. (The dependencies on x , z and μ have been suppressed for clarity.) If H is positive definite this system is nonsingular at all interior points (x, z) . If v denotes a combined $2n$ vector of unknowns (x, z) and $G^\mu(v)$ denotes the function $G^\mu(x, z)$, then (9) are the equations $G^\mu(v)' \Delta v = -G^\mu(v)$ for the Newton direction $\Delta v = (\Delta x, \Delta z)$. Applying block elimination to (9) gives Δx and Δz as

$$(H + X^{-1}Z)\Delta x = -(g - \pi) \quad \text{and} \quad \Delta z = g + H\Delta x - z.$$

Scaling this system with $X^{1/2}$ gives the solution

$$\Delta x = X^{1/2} \Delta \bar{x}, \quad \text{where } \Delta \bar{x} \text{ solves } (X^{1/2} H X^{1/2} + Z) \Delta \bar{x} = -X^{1/2}(g - \pi). \quad (10)$$

The crucial feature of this system is that $X^{1/2} H X^{1/2} + Z$ has the same dimension and sparsity pattern as H . If the second-order sufficient conditions hold, then for μ sufficiently small, a differentiable *trajectory* of solutions $(x(\mu), z(\mu))$ exists such that $(x(\mu), z(\mu)) \rightarrow (x^*, z^*)$ as $\mu \rightarrow 0^+$. Primal-dual interior methods attempt to follow this trajectory by finding an approximate solution of $G^\mu(x, z) = 0$ for a decreasing sequence of μ -values such that $\mu \rightarrow 0^+$. As the solution is approached, the scaled matrix $X^{1/2} H X^{1/2}$ converges to a row and column-scaled version of H in which the zero rows and columns correspond to the active bounds. It follows that if the problem is nondegenerate, the rows and columns of $X^{1/2} H X^{1/2} + Z$ corresponding to the active bounds are diagonal.

The step length is chosen using a standard interior-point backtracking line search (for more details see, e.g., [28]). Consider the calculation of new iterates $(x_{k+1}, z_{k+1}) = (x_k + \alpha_k \Delta x_k, z_k + \alpha_k \Delta z_k)$ at the k th iteration. First, an upper bound on the step is computed such that $\alpha_M = \min\{1, .99\tau\}$, where τ is the largest positive α such that $(x_k + \alpha \Delta x_k, z_k + \alpha \Delta z_k)$ is feasible. The step α_k is then the first member of the sequence $\{\gamma_c^j \alpha_M\}_{j=0}^\infty$ such that

$$\|G^\mu(x_k + \alpha_k \Delta x_k, z_k + \alpha_k \Delta z_k)\|_2 \leq (1 - \alpha_k \eta_s) \|G^\mu(x_k, z_k)\|_2,$$

for fixed scalars η_s ($0 < \eta_s < \frac{1}{2}$) and γ_c ($0 < \gamma_c < 1$) with typical values $\eta_s = \frac{1}{4}$ and $\gamma_c = \frac{1}{2}$. (Throughout this section, $\|\cdot\|_2$ refers to the usual vector two-norm.)

In the general case where f is not convex, a more sophisticated strategy must be used to guarantee convergence to a point satisfying the second-order necessary conditions for problem (6). The description of such strategies is beyond the scope of this paper. However, it is relatively straightforward to formulate methods that are not only provably convergent to second-order points, but are also able to exploit the properties of the multilevel iterative solver. One approach is to define line search or trust region methods that minimize the function

$$M^\mu(x, z) = f(x) - \mu \sum_{j=1}^n \left(\ln x_j + \ln \left(\frac{x_j z_j}{\mu} \right) + \left(\frac{\mu - x_j z_j}{\mu} \right) \right)$$

(see [13,14]). This function is well defined for all (x, z) such that $x_j > 0$ and $z_j > 0$ and has a local minimizer at a point (x, z) such that $G^\mu(x, z) = 0$. It can be arranged that algorithms for minimizing $M^\mu(x, z)$ solve a system that has identical structure to that of (10). However, the multilevel iterative solver must satisfy two requirements: the multigraph algorithm must always implicitly generate a positive-definite preconditioner, and the conjugate-gradient method must be modified to detect indefiniteness in the matrix of system (10) (see, e.g., [25,16]).

2.1 Treatment of upper and lower bounds

Now consider problem (6) with finite upper and lower bounds $b_l \leq x \leq b_u$. If $L = \text{diag}(b_l)$ and $U = \text{diag}(b_u)$, let $X_1 = X - L$ and $X_2 = X - U$. Assume that x is interior, so that $X_1 > 0$ and $X_2 < 0$. Let z_1 and z_2 denote estimates of the multipliers associated with the constraints $x \geq b_l$ and $x \leq b_u$ respectively. The perturbed optimality conditions are

$$\begin{aligned} g(x) - (z_1 + z_2) &= 0 \\ X_1(z_1 - \pi_1) &= 0 \\ X_2(z_2 - \pi_2) &= 0, \end{aligned}$$

with $\pi_1 = \mu X_1^{-1}e$ and $\pi_2 = \mu X_2^{-1}e$. Linearizing these conditions at a point (x, z_1, z_2) such that $b_l < x < b_u$, $z_1 > 0$ and $z_2 < 0$ gives the following system analogous to (9)

$$\begin{pmatrix} H & -I \\ \widehat{Z} & \widehat{X} \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta z \end{pmatrix} = - \begin{pmatrix} g - z \\ \widehat{X}(z - \pi) \end{pmatrix},$$

where $\widehat{X} = -X_1 X_2$, $\widehat{Z} = -(X_2 Z_1 + X_1 Z_2)$, $z = z_1 + z_2$ and $\pi = \pi_1 + \pi_2$.

It follows that Δx can be calculated as $\Delta x = \widehat{X}^{1/2} \Delta \bar{x}$, where $\Delta \bar{x}$ satisfies the system

$$(\widehat{X}^{1/2} H \widehat{X}^{1/2} + \widehat{Z}) \Delta \bar{x} = -\widehat{X}^{1/2} (g - \pi). \quad (11)$$

Again, the relevant matrix has the same dimension and sparsity pattern as H .

3 Interior Methods for the Variational Problem

Now we consider the application of the primal-dual interior method to a *sequence* of finite-dimensional optimization problems (5) defined in a finite-element discretization with adaptive mesh refinement.

3.1 Scaling the finite-element discretization

It is well known that problem scaling has a substantial effect on the efficiency of optimization methods. Scaling is particularly relevant when the problem is defined by an adaptive finite-element method. For a given mesh, as the solution is approached, the magnitude of each objective gradient element g_i depends on the support of its constituent nodal basis functions. This implies that the gradients (and hence the Lagrange multipliers) can vary widely in magnitude depending on the degree of refinement in a particular region.

Here we use a scaling that balances the magnitudes of the Lagrange multipliers and nonlinear equations as the mesh is refined. Let D be the positive diagonal matrix with entries $d_i = 2\|\phi_i\|^2$, where $\{\phi_i\}$ are the nodal basis functions. The $\{d_i\}$ form the diagonal elements of the so-called *lumped mass matrix* M , with entries $M_{ij} = (\phi_i, \phi_j)$. Note that M has the same sparsity pattern as the stiffness matrix A . The matrix D is used to replace the first-order optimality condition $g - z = 0$ in (7) by the scaled condition $g - Dz = 0$. Proceeding as in the previous section, we obtain Δx as

$$\Delta x = \widehat{X}^{1/2} \Delta \bar{x}, \text{ where } \Delta \bar{x} \text{ satisfies } (\widehat{X}^{1/2} H \widehat{X}^{1/2} + D\widehat{Z}) \Delta \bar{x} = -\widehat{X}^{1/2} (g - D\pi). \quad (12)$$

In the line search, a norm of the balanced vectors $g - Dz$ and $DX(z - \pi)$ is used to measure the proximity to the solution. In particular, the two-norm of G^μ (8) is replaced by the elliptic norm

$$\begin{aligned} \|G^\mu\|_{D^{-1}}^2 &\equiv \|g - Dz\|_{D^{-1}}^2 + \|DX(z - \pi)\|_{D^{-1}}^2 \\ &= (g - Dz)^T D^{-1} (g - Dz) + (z - \pi)^T X^T DX (z - \pi). \end{aligned}$$

Similarly, the variables are balanced using $\|v\|_D^2 \equiv \|x\|_D^2 + \|z\|_D^2 = x^T Dx + z^T Dz$.

3.2 Mesh refinement and the choice of barrier parameter

The proposed algorithm has two levels of iteration: the multilevel computations associated with a given finite-element mesh, and the Newton iterations associated with the given barrier parameter μ . Here we link the choice of barrier parameter to the mesh size, the main idea being that it is appropriate to accurately solve the optimization problem (i.e., choose μ small) only in the final stages of the mesh refinement.

Consider the approximate solution of a sequence of bound-constrained optimization problems BC^j , each associated with a given level of mesh refinement. We associate a barrier parameter $\mu^{(j)}$ with each problem, where $\mu^{(j-1)} > \mu^{(j)}$. The initial point for problem BC^j is obtained by interpolating the continuous finite-element solution associated with the final point from BC^{j-1} . In this situation, the value of $\mu^{(j)}$ is crucial, since a poor choice may result in the interpolated finite-element solution from BC^{j-1} being far from the trajectory for BC^j .

From standard theory, the finite-element approximation $u_h \in \mathcal{S}$ has the error estimate $\|u_h - u^*\| = O(n^{-1/2})$, where u^* is the optimal solution. It follows that if n_j is the number of variables associated with BC^j , then the error in the finite-element approximation is reduced by a factor of $O((n_{j-1}/n_j)^{1/2})$ at each refinement. In practice, n is usually increased geometrically. In the numerical examples of Section 4, the n for adaptive refinement is increased by a factor of four, which implies that the error is reduced by a factor of two after each refinement.

Next we estimate the error in any point $v(\mu)$ on the primal-dual trajectory as an estimate of the optimal point v^* . If $dv(\mu)/d\mu$ exists and is bounded, then for μ sufficiently small, $\|v(\mu) - v^*\| = O(\mu)$ (see [12, Theorem 15], and [29, p. 8]). It follows that if $u_h(\mu) \in \mathcal{S}$ denotes the solution associated with barrier parameter μ , we would expect the total error in $u_h(\mu)$ to be $O(\mu) + O(n^{-1/2})$. This bound suggests the choice of initial barrier parameter $\mu^{(0)} = O(n_0^{-1/2})$, with subsequent values reduced at the same rate as the finite-element error estimate, i.e., $\mu^{(j)} = (n_{j-1}/n_j)^{1/2} \mu^{(j-1)}$.

3.3 Choosing the starting point for BC^j

The initial values of the x -variables for BC^j are found by interpolating the final values for BC^{j-1} . The initial z -variables may be defined in several ways. First, as in the primal case, the z values can be interpolated from the final iteration of problem BC^{j-1} . This choice will tend to give small initial residuals for the first set of equations $g - Dz = 0$ of the system $G^\mu(v) = 0$. However, the second set of equations $Xz - \mu e = 0$ will have a large residual since the refined z values are generally increased by the scaling with D . Another option is to use the complementarity conditions to define new dual variables from the interpolated primal values. In this case, $(z_1)_i = (\pi_1)_i$ and $(z_2)_i = (\pi_2)_i$, where π_1 and π_2 are the initial primal multipliers $\pi_1 = \mu X_1^{-1}e$ and $\pi_2 = \mu X_2^{-1}e$. Another possibility is to define z so that the residual for $g - Dz$ is small. For example, if $g_i > 0$, then we can set $(z_1)_i = g_i/d_i$ and $(z_2)_i = (\pi_2)_i/d_i$, where π_2 are the initial primal multipliers $\pi_2 = \mu X_2^{-1}e$. In this case, the residual for the equation $g_i - d_i z_i = 0$ is $g_i - d_i(z_1 + z_2)_i = (\pi_2)_i$.

Any one of these alternatives is reasonable when the “continuous” Lagrange multipliers are smooth. However, when the PDE solution is not differentiable, the continuous multipliers can resemble delta functions (see, e.g., the minimal surface problem of Section 4) and naive refinement techniques can lead to poor initial estimates of the multipliers. To deal with this problem, the multipliers are smoothed using the inverse Laplacian before being interpolated. The interpolated multipliers are then mapped back to the original space to initialize the optimization. Further details of the smoothing procedure are given in Section 3.4.

Adaptive mesh refinement introduces some algorithmic issues that are unique to interior methods. Adaptive refinement involves not only the introduction of new elements where

needed, but can also be combined with mesh regularization (i.e., the movement of elements) to further improve the quality of the mesh. Either of these procedures can cause the upper and lower bounds of the discretized problem to change, which implies that the approximate solution of BC^{j-1} may not be a good estimate of the solution for BC^j . In particular, the solution of BC^{j-1} may not even lie in the interior of the feasible region for problem BC^j . In this situation, it is necessary to redefine the initial point for BC^j so that it is interior. In the numerical results of Section 4 the initial point is forced to lie at a distance of at least $\nu\mu$ from their nearest bound, where $0 < \nu \leq 1$.

3.4 Lagrange multiplier smoothing

Let ζ denote the Lagrange multipliers associated with the continuous bound-constrained problem (4). In general, the natural space for ζ is $\mathcal{H}^{-1}(\Omega)$, the class of functions that are integrable against functions in $\mathcal{H}^1(\Omega)$ (for the precise definition, see, e.g., [6]). As $\mathcal{H}^{-1}(\Omega)$ functions need not be continuous and can exhibit δ -function-like behavior, interpolating in a region that includes a jump discontinuity can not fully capture the behavior of ζ . This lack of smoothness is problematic when ζ is approximated by a piecewise-linear function derived by linear interpolation from a coarse mesh. If ζ is poorly behaved, then the interpolant will invariably be a poor initial guess for the problem on the finer mesh.

An alternative approach is to smooth the coarse mesh multipliers before interpolation. For any $\zeta \in \mathcal{H}^{-1}(\Omega)$, the solution w of

$$-\Delta w = \zeta \quad \text{in } \Omega \tag{13}$$

is in $\mathcal{H}^1(\Omega)$ and it follows that the inverse Laplacian Δ^{-1} can be used as a “smoothing” map for the approximate multipliers. Given a multiplier function $\zeta = \zeta_c$ on the coarse mesh, the Poisson equation (13) is solved for $w = w_c$. The linear interpolant w_f of w_c is computed for the refined mesh, and the new set of Lagrange multipliers ζ_f is defined as $\zeta_f = -\Delta w_f$. Thus, the interpolation is actually applied to $\Delta^{-1}\zeta_c$, and not on ζ_c itself. The discontinuous behavior of ζ is better reflected in this interpolation.

4 Numerical Results

In this section we discuss the results from applying PLTMG to three problems. Each problem was solved using adaptive mesh refinement, starting from (unless otherwise indicated) a uniform 5×5 mesh. At each refinement the number of unknowns was increased by a factor of k_{ref} , where $k_{ref} = 4$. Two variants of PLTMG were used to solve each problem.

Method A. This method was designed to illustrate the influence of the initial interpolated solution and the change of barrier parameter. For each level of refinement, the finite-dimensional problem was first solved using a so-called *refinement step* in which the value of μ is inherited from the previous mesh. In this case, the initial point for the subproblem was the final solution on the previous mesh. The subproblem was then solved again in a *continuation step*, starting from the fine mesh solution, but with μ reduced by a factor of two. All times reported are the cumulative times to each point in the calculation. For the refinement steps, the reported figures include error estimation and mesh refinement times as well as time spent setting up and solving the variational inequality. In each case we report the number **Iter** of major Newton iterations to solve the problem at the given step.

Method B. In this method, the refinement and μ -reduction were performed simultaneously. This is the strategy recommended for practical computation, and is the basis of Algorithm 4.1 given below.

In all cases, the Newton iteration was stopped when the relative increment $\|\Delta v\|_D/\|v\|_D$ was reduced by a factor $\eta = 10^{-2}$ compared to its initial value. This test implies that at least two Newton steps are performed for each subproblem. A more stringent convergence criterion would result in a more accurate solution of the discrete variational inequality, but no improvement in the accuracy of the underlying continuous problem. Indeed, given the typical second-order convergence of the the piecewise linear discretization, 10^{-2} could be considered conservative.

For each problem, the initial point for the first subproblem was $\frac{1}{2}(b_l + b_u)$. The initial values for the multipliers z_1 and z_2 were the primal multipliers $\mu_0/(x - b_l)$ and $\mu_0/(x - b_u)$. The scale factor C_1 for the initial barrier parameter was chosen so that $\mu_0 = 1$. The parameter for shifting variables away from their bounds was $\nu = 1$. The line search parameters were $\gamma_c = \frac{1}{2}$ and $\eta_s = \frac{1}{4}$.

4.1 The algorithm

ALGORITHM 4.1. PRIMAL-DUAL METHOD.

Define n_{\max} ; η, γ_c ($0 < \eta, \gamma_c < 1$); η_s ($0 < \eta_s < \frac{1}{2}$); C_1 ($C_1 > 0$);

Choose n ($0 < n \leq n_{\max}$);

$\mu = C_1 n^{-1/2}$; $v = (x, z_1, z_2)$;

while $n \leq n_{\max}$ **do**

$tol = 0$; $first = \mathbf{true}$; $\Delta v = v$;

while $\|G^\mu(v)\|_{D^{-1}} > 0$ **and** $\|\Delta v\|_D/\|v\|_D > tol$ **do**

Compute $\Delta v = (\Delta x, \Delta z_1, \Delta z_2)$ from (12);

$\tau = \max\{\alpha : \alpha > 0, v + \alpha \Delta v \text{ is feasible}\}$;

$\alpha = \min\{1, .99\tau\}$;

while $\|G^\mu(v + \alpha \Delta v)\|_{D^{-1}} > (1 - \alpha \eta_s)\|G^\mu(v)\|_{D^{-1}}$ **do**

$\alpha \leftarrow \gamma_c \alpha$;

end do

$v \leftarrow v + \alpha \Delta v$;

if $first$ **then**

$tol = \eta \|\Delta v\|_D/\|v\|_D$; $first = \mathbf{false}$;

end if

end do

$n \leftarrow k_{ref} n$; Refine the mesh;

$\mu \leftarrow \mu/\sqrt{k_{ref}}$;

end do

All runs were made on an SGI Octane workstation with 512MB of RAM and two 250MHz R10000 processors (only one being used for each problem solution). The f90 compiler was used with `-n32 -O` options specifying 32-bit mode and full code optimization.

Example 1: Elliptic variational inequality. Consider the variational problem

$$\min_{u \in K} q(u) = \int_{\Omega} |\nabla u(x)|^2 - 2c(x)u(x) dx,$$

where $\Omega = [0, 1] \times [0, 1]$, $K = \{u \in \mathcal{H}_0^1(\Omega) : |u| \leq \frac{1}{4} - \frac{1}{10} \sin(\pi x_1) \sin(\pi x_2) \text{ for } x \in \Omega\}$, and $c(x)$ is chosen such that $-\Delta w = c$, with $w = \sin(3\pi x_1) \sin(3\pi x_2)$.

The results are summarized in Table 1. Columns 3 and 4 give the times and iterations for the separate refinement and continuations steps. Columns 5 and 6 give the details for Algorithm 4.1.

Table 1. Elliptic variational inequality.

n	μ	Method A		Method B	
		Iter	cpu secs	Iter	cpu secs
25	1	8	0.01	8	0.01
64	1	6	0.04		
64	2^{-1}	3	0.05	6	0.04
250	2^{-1}	8	0.19		
250	2^{-2}	3	0.25	9	0.19
1000	2^{-2}	12	1.05		
1000	2^{-3}	3	1.34	14	1.08
4000	2^{-3}	10	4.84		
4000	2^{-4}	3	6.34	15	5.74
16000	2^{-4}	10	24.37		
16000	2^{-5}	3	32.23	12	26.49
64000	2^{-5}	9	136.10		
64000	2^{-6}	3	189.75	11	144.47

Fig. 1 gives the final adaptive mesh and corresponding approximate solution. The results in Table 1 indicate that the refinement steps generally require more Newton iterations than the continuation steps. This mainly reflects the quality of the initial guesses for these problems, since the underlying discrete variational inequalities associated with each mesh are quite similar. To some extent, the additional iterations are needed because the adaptive refinement tends to introduce new grid points in areas where the solution is least accurate. This is shown quite clearly in Fig. 1, where we observe that the smallest elements resolve the interface between contact and non-contact zones, and the largest elements appear near the centers of the contact zones, where the solution is well-defined by the obstacle. The initial guess for refinement steps is interpolated from the previous mesh, and is clearly not as good as the fine-grid solution that serves as initial guess for the continuation steps.

Example 2: Elastic-plastic torsion. This problem concerns an infinitely long cylindrical bar made up of an isotropic elastic perfectly plastic material. Starting from a zero-stress initial state, an increasing torsion moment is applied to the bar. The elastic-plastic torsion problem [15, p. 41-46] can be formulated in terms of the cross-section Ω of the cylinder and the torsion angle constant c per unit length. The stress field is the solution of the variational problem

$$\min_{u \in K} q(u) = \int_{\Omega} |\nabla u(x)|^2 - 2cu(x) dx,$$

where $\Omega = [0, 1] \times [0, 1]$, $K = \{u \in \mathcal{H}_0^1(\Omega) : |u| \leq \text{dist}(x, \partial\Omega) \text{ for } x \in \Omega\}$, and $\text{dist}(x, \partial\Omega)$ denotes the distance from x to the boundary of Ω . The symbol $\mathcal{H}_0^1(\Omega)$ denotes the space of functions with gradients in $L^2(\Omega)$ that vanish on the boundary of Ω . If there is no constraint, then u will satisfy the boundary-value problem

$$-\Delta u = c \text{ for } x \in \Omega, \text{ and } u = 0 \text{ for } x \in \partial\Omega.$$

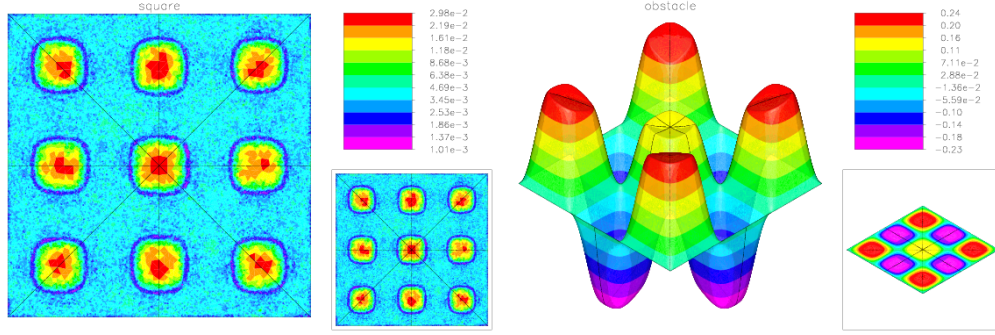


Fig. 1. Elliptic variational inequality. On the left is an plot of the element size in the adaptive mesh with $n = 64000$ vertices. On the right is the corresponding solution.

In all the numerical results, the constant $c = 5$ was chosen to be consistent with the elastic-plastic torsion problem in the COPS test set [5]. We repeated the same experiments as in the first example, beginning with an identical 5×5 mesh. Although subsequent meshes have the same numbers of mesh points, adaptive refinement made the meshes quite different. The results of the calculation are summarized in Table 2, and the final adaptive mesh and solution are shown in Fig. 2.

Table 2. Elastic-plastic torsion problem.

n	μ	Method A		Method B	
		Iter	cpu secs	Iter	cpu secs
25	1	5	0.01	5	0.01
64	1	4	0.03		
64	2^{-1}	3	0.04	4	0.03
250	2^{-1}	4	0.12		
250	2^{-2}	3	0.17	4	0.11
1000	2^{-2}	5	0.60		
1000	2^{-3}	3	0.85	6	0.57
4000	2^{-3}	5	2.93		
4000	2^{-4}	3	4.31	6	2.92
16000	2^{-4}	5	15.43		
16000	2^{-5}	3	23.34	5	14.45
64000	2^{-5}	5	95.48		
64000	2^{-6}	4	161.47	5	89.72

Generally, this was an easier problem with a smoother solution than our first example. This is reflected in a smaller numbers of Newton iterations needed for the refinement steps. The final adaptive mesh is much more uniform, although, once again, the largest elements appear in the interior of the contact zone where the solution is well-defined by the obstacle.

Example 3: Minimal surface with an obstacle. This problem involves finding the least amount of total energy to maintain, and thus enclose a given area/volume with as little perimeter/surface area as possible. This formulation is identical to the one in the

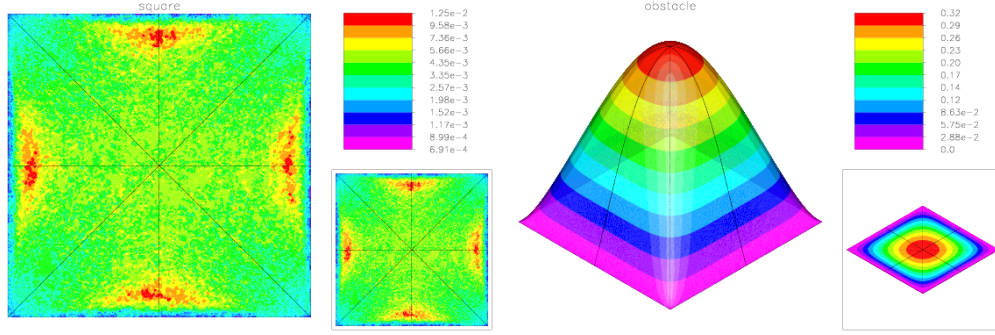


Fig. 2. Elastic-plastic torsion problem. On the left is a plot of the element size in the adaptive mesh with $n = 64000$ vertices. On the right is the corresponding solution.

COPS test set [10]. The aim is to minimize the area function $q: K \rightarrow \mathbb{R}$, such that

$$q(u) = \int_{\Omega} (1 + |\nabla u(x)|^2)^{1/2} dx, \quad (14)$$

where $\Omega = [0, 1] \times [0, 1]$, $K = \{u \in \mathcal{H}^1(\Omega) : u(x) = u_{\Omega}(x) \text{ for } x \in \partial\Omega, u(x) \geq u_L(x) \text{ for } x \in \Omega\}$. The function $u_{\Omega}: \partial\Omega \rightarrow \mathbb{R}$ defines the boundary data, and the function $u_L: \Omega \rightarrow \mathbb{R}$ defines the obstacle. It is assumed that $u_L \leq u_{\Omega}$ on the boundary. If no obstacle is present, the minimal surface satisfies the boundary-value problem

$$\operatorname{div}((1 + |\nabla u|^2)^{-1/2} \nabla u) = 0 \text{ for } x \in \Omega, \text{ and } u = u_{\Omega} \text{ for } x \in \partial\Omega$$

(see, e.g., [26, p. 174]). In the case considered here, the boundary constraint u_{Ω} is defined such that

$$u_{\Omega} = \begin{cases} 1 - (2x_1 - 1)^2, & \text{if } x_2 = 0, 1; \\ 0, & \text{otherwise.} \end{cases}$$

The obstacle is

$$u_L = \begin{cases} 1, & \text{if } |x_1 - \frac{1}{2}| \leq \frac{1}{4} \text{ and } |x_2 - \frac{1}{2}| \leq \frac{1}{4}; \\ 0, & \text{otherwise.} \end{cases}$$

To make u_L continuous, we added a small band around the center box where u_L varied linearly from 0 to 1. This is illustrated in Fig. 3.

Table 3 shows the results of the computation. Because the initial mesh had $n = 685$ vertices, the computation was started with $\mu = .25$ in order to ensure that the refined meshes correspond to those of the previous examples. However, this problem differs from the previous ones in some other significant respects. The definition of the initial coarse mesh provides a convenient framework in which to define u_L (see Fig. 3). However, the use of this mesh also means that the boundary of the central square where u_L is nearly discontinuous is automatically and exactly incorporated into each of the refined meshes. The fact that this (anticipated) interface is described exactly in the mesh is, of course, beneficial to the adaptive meshing procedure. The solution and refined mesh are shown in Fig. 4.

Another difference we observed in this problem is that the Lagrange multipliers are quite rough. In Fig. 5, we show both the Lagrange multipliers and the smoothed Lagrange multipliers used for interpolation. It is clear that the multipliers exhibit a δ -function-like behavior on the boundary of the contact zone, particularly at the corners of the square. The smoothed multipliers are visibly much nicer. We also note that the number of Newton iterations for the refinement and continuation steps is roughly comparable for this problem.

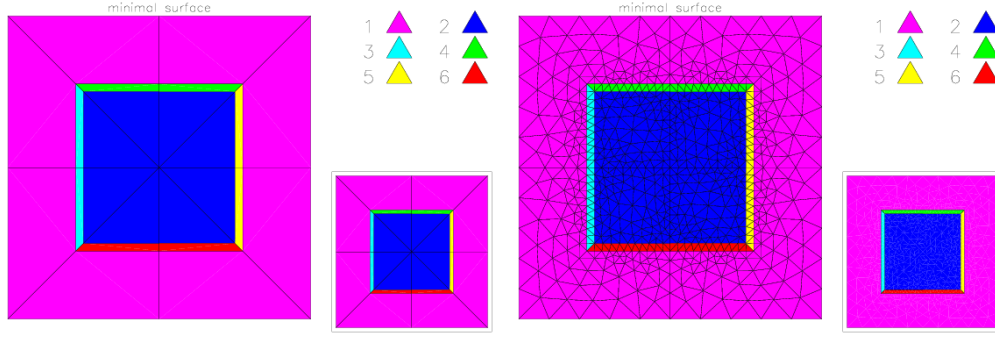


Fig. 3. Minimal surface problem: On the left, we illustrate the skeleton corresponding to the definition of u_L . On the right, we show the corresponding initial mesh with 685 mesh points.

As before, this behavior mostly reflects the quality of the initial guesses. For this problem, the roughness of the Lagrange multipliers is probably responsible for the additional Newton iterations needed for the continuation steps, in comparison with previous examples.

Table 3. Minimal surface problem.

n	μ	Method A		Method B	
		Iter	cpu secs	Iter	cpu secs
685	2^{-2}	11	0.52	11	0.52
1000	2^{-2}	7	1.09		
1000	2^{-3}	3	1.41	5	1.00
4000	2^{-3}	14	6.10		
4000	2^{-4}	3	7.70	9	4.31
16000	2^{-4}	7	21.86		
16000	2^{-5}	6	36.27	5	16.27
64000	2^{-5}	5	109.61		
64000	2^{-6}	7	199.06	5	92.11

Finally, Table 4 gives the cpu seconds required for some of the principal routines of the algorithm. All initialization and preprocessing, including the computation of the sparsity pattern of H and the setup for the data structures for each level, is done in subroutine `mginit`. The multigraph incomplete LU factorization is computed in subroutine `mgilu`. These factors are used in subroutine `mg` to solve the main system (11). Subroutine `tpick` performs the line search and computes required directional derivatives. Subroutine `linsys` computes the finite element Jacobian and Newton residual. The computation of the smoothed Lagrange multipliers is done in subroutine `smlm`.

Acknowledgements

We thank Michael Saunders for many helpful comments on the performance and implementation of iterative solvers for interior methods.

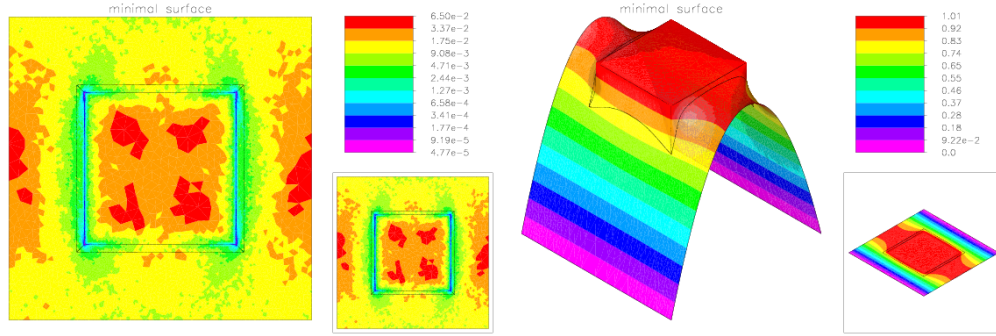


Fig. 4. Minimal surface problem. On the left is a plot of the element size in the adaptive mesh with $n = 64000$ vertices. On the right is the corresponding solution.

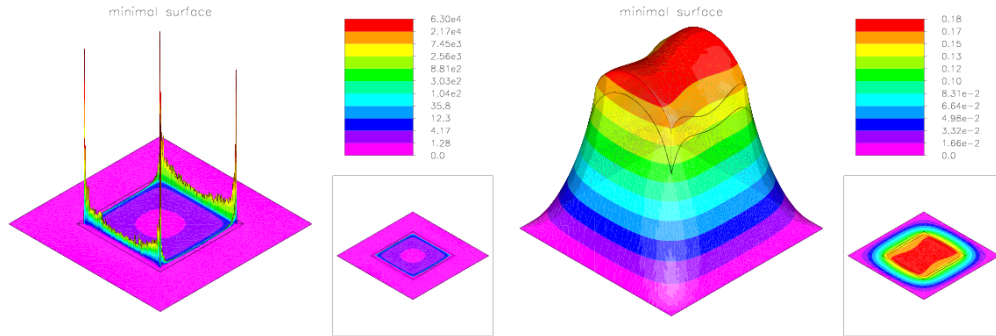


Fig. 5. Minimal surface problem. On the left is a plot of the Lagrange multipliers for the mesh with $n = 64000$ vertices. On the right is a plot of the smoothed Lagrange multipliers.

Table 4. Minimal surface problem. Breakdown of the computation time. The first set of times are for explicit refinement and continuation steps.

Task	cpu secs	cpu secs
Mesh generation	19.19	15.17
Subroutine mginit	20.40	10.35
Subroutine mgilu	31.16	14.01
Subroutine mg	26.77	10.07
Subroutine linsys	50.19	17.89
Subroutine tpick	15.43	5.94
Subroutine smlm	31.41	16.29
Other	4.51	2.39

References

1. R. E. BANK, *PLTMG: A Software Package for Solving Elliptic Partial Differential Equations, Users' Guide 8.0*, Software, Environments and Tools, Vol. 5, SIAM, Philadelphia, 1998.
2. R. E. BANK AND K. R. SMITH, *The incomplete factorization multigraph algorithm*, SIAM J. Sci. Comput., 20 (1999), pp. 1349–1364 (electronic).
3. D. P. BERTSEKAS, *On the Goldstein-Levitin-Polyak gradient projection method*, IEEE Trans. Automatic Control, AC-21 (1976), pp. 174–184.
4. ———, *Projected Newton methods for optimization problems with simple constraints*, SIAM J. Control Optim., 20 (1982), pp. 221–246.
5. A. BONDARENKO, D. BORTZ, AND J. J. MORÉ, *COPS: Large-scale nonlinearly constrained optimization problems*, Technical Report ANL/MCS-TM-237, Mathematics and Computer Science division, Argonne National Laboratory, Argonne, IL, 1998. Revised October 1999.
6. D. BRAESS, *Finite elements*, Cambridge University Press, Cambridge, second ed., 2001. Theory, fast solvers, and applications in solid mechanics, Translated from the 1992 German edition by Larry L. Schumaker.
7. M. A. BRANCH, T. F. COLEMAN, AND Y. LI, *A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems*, SIAM J. Sci. Comput., 21 (1999), pp. 1–23 (electronic).
8. T. D. CHOI AND C. T. KELLEY, *Estimates for the Nash-Sofer preconditioner for the reduced Hessian for some elliptic variational inequalities*, SIAM J. Optim., 9 (1999), pp. 327–341.
9. T. F. COLEMAN AND Y. LI, *An interior trust region approach for nonlinear minimization subject to bounds*, SIAM J. Optim., 6 (1996), pp. 418–445.
10. E. D. DOLAN AND J. J. MORÉ, *Benchmarking optimization software with performance profiles*, Math. Program., 91 (2002), pp. 201–213.
11. Z. DOSTÁL, *Box constrained quadratic programming with proportioning and projections*, SIAM J. Optim., 7 (1997), pp. 871–887.
12. A. V. FIACCO AND G. P. MCCORMICK, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, Classics in Applied Mathematics, SIAM, Philadelphia, 1990. ISBN 0-89871-254-8.
13. A. FORSGREN AND P. E. GILL, *Primal-dual interior methods for nonconvex nonlinear programming*, SIAM J. Optim., 8 (1998), pp. 1132–1152. (Electronic).
14. E. M. GERTZ AND P. E. GILL, *A primal-dual trust region algorithm for nonlinear programming*, Report to appear, Department of Mathematics, University of California, San Diego, 2002.
15. R. GLOWINSKI, *Numerical methods for nonlinear variational problems*, Springer-Verlag, New York, 1984. ISBN 0-387-12434-9.
16. N. I. M. GOULD, S. LUCIDI, M. ROMA, AND P. L. TOINT, *Solving the trust-region subproblem using the Lanczos method*, SIAM J. Optim., 9 (1999), pp. 504–525 (electronic).
17. M. HEINKENSCHLOSS, M. ULBRICH, AND S. ULBRICH, *Superlinear and quadratic convergence of affine-scaling interior-point Newton methods for problems with simple bounds without strict complementarity assumption*, Math. Program., 86 (1999), pp. 615–635.

18. R. H. W. HOPPE AND R. KORNHUBER, *Adaptive multilevel methods for obstacle problems*, SIAM J. Numer. Anal., 31 (1994), pp. 301–323.
19. D. KINDERLEHRER AND G. STAMPACCHIA, *An introduction to variational inequalities and their applications*, Academic Press Inc. [Harcourt Brace Jovanovich Publishers], New York, 1980.
20. ———, *An introduction to variational inequalities and their applications*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000. Reprint of the 1980 original.
21. R. KORNHUBER, *Monotone multigrid methods for elliptic variational inequalities. I*, Numer. Math., 69 (1994), pp. 167–184.
22. ———, *Monotone multigrid methods for elliptic variational inequalities. II*, Numer. Math., 72 (1996), pp. 481–499.
23. C.-J. LIN AND J. J. MORÉ, *Newton's method for large bound-constrained optimization problems*, SIAM J. Optim., 9 (1999), pp. 1100–1127 (electronic). Dedicated to John E. Dennis, Jr., on his 60th birthday.
24. J. J. MORÉ AND G. TORALDO, *On the solution of large quadratic programming problems with bound constraints*, SIAM J. Optim., 1 (1991), pp. 93–113.
25. S. G. NASH, *Newton-type minimization via the Lánczos method*, SIAM J. Numer. Anal., 21 (1984), pp. 770–788.
26. J. RAUCH, *Partial differential equations*, Springer-Verlag, New York, 1991.
27. M. ULBRICH, S. ULBRICH, AND M. HEINKENSCHLOSS, *Global convergence of trust-region interior-point algorithms for infinite-dimensional nonconvex minimization subject to pointwise bounds*, SIAM J. Control Optim., 37 (1999), pp. 731–764 (electronic).
28. S. J. WRIGHT, *Primal-dual interior-point methods*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997. ISBN 0-89871-382-X.
29. ———, *On the convergence of the Newton/log-barrier method*, Technical report MCS-P681-0897, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, 1997, Revised April, 2000.