Inexact Interior-Point Method for Pde-Constrained Nonlinear Optimization

Marcus J. Grote, Johannes Huber, Drosos Kourounis, Olaf Schenk

Institute of Mathematics
University of Basel
Rheinsprung 21
CH - 4051 Basel
Switzerland

Preprint No. 2013-15
May, 2013

www.math.unibas.ch
INEXACT INTERIOR-POINT METHOD FOR PDE-CONSTRANGED
NONLINEAR OPTIMIZATION

MARCUS J. GROTE†, JOHANNES HUBER‡, DROSOS KOUROUNIS§, AND
OLAF SCHENK¶

Abstract. Starting from the inexact interior-point framework from Curtis et al. [Mathematical
Programming Series B, 32(6), pp. 3447-3475, 2012], we propose an effective reduced-space precondi-
tioner for the full Lagrangian Hessian matrix needed at each Newton iteration. Together they yield
a scalable, robust and highly parallel method for the numerical solution of large-scale nonconvex
PDE-constrained optimization problems with inequality constraints. Because it uses the full Hessian
matrix, modifying it whenever needed, the method is not only globally convergent, but also con-
verges fast locally. Our preconditioner is not tailored to any particular class of PDEs or constraints,
but instead judiciously exploits the sparsity structure of the Hessian. Numerical examples from
PDE-constrained optimal control, parameter estimation and full waveform inversion demonstrate
the robustness and efficiency of the method, even in the presence of active inequality constraints.

Key words. PDE-constrained optimization, nonconvex programming, inequality constraints,
inexact interior-point method, preconditioning, KKT system, optimal control, seismic imaging, full-
waveform inversion

AMS subject classifications. 49K20, 49J20, 65J22, 90C06, 90C51

DOI.

1. Introduction. Throughout science and engineering, mathematical models of
physical systems are extensively expressed as partial differential equations (PDEs).
As computational methods for the solution of PDEs reach a mature stage, much
interest in current research is shifting toward optimal design, optimal control, and
parameter estimation of systems governed by PDEs. Despite their seeming disparity,
all these problems fit the class of PDE-constrained optimization, where (unknown)
PDE parameters, $u$ – initial or boundary data, source terms, material coefficients,
or domain geometry – must be determined by minimizing an appropriate objective
functional

$$
\min_{y,u} F(y,u)
\text{ s.t. } A_k(y_k,u) = 0, \quad \text{for } k = 1, \ldots, N_E
u_- \leq u \leq u_+.
$$

Here $y = (y_1, y_2, \ldots, y_{N_E})$ denotes a collection of state variables $y_k$ that satisfy (pos-
sibly nonlinear) PDEs, each involving $u$. In optimal control, for instance, $u$ appears
as a source term in every PDE; hence when the differential operators $A_k$ are linear
in $y_k$, $y$ will also depend linearly on $u$. In parameter estimation, however, the partial

---

*Received by the editors May 15, 2013; accepted for publication (in revised form) July xx, 20xx;
published electronically November 26, 20xx. sisc/31-2/xxxxx.html
†Department of Mathematics and Computer Science, University of Basel, Rheinsprung 21, CH-
4051 Basel, Switzerland (marcus.grote@unibas.ch).
‡Department of Mathematics and Computer Science, University of Basel, Rheinsprung 21, CH-
4051 Basel, Switzerland (johannes.huber@unibas.ch).
§Institute of Computational Science, Universita della Svizzera italiana, CH-6900 Lugano, Switzer-
land (drosos.kourounis@usi.ch).
¶Institute of Computational Science, Universita della Svizzera italiana, CH-6900 Lugano, Switzerland
(olaf.schenk@usi.ch).
differential operator $\mathcal{A}_k$ itself also depends on the model parameters, $u$; hence even when $\mathcal{A}_k$ is linear in $y_k$, $y_k$ usually depends nonlinearly on $u$ yielding a nonconvex optimization problem. As a consequence, this inverse problem is often significantly more difficult to solve than the forward problem that consists of the mere computation of $y$ for a given $u$. It is often ill-posed, requires regularization, and yet can still have multiple local solutions. The inequality constraints in (1.1) not only permit the imposition of physical bounds on $u$, but also the inclusion of information available a priori. Thus, inequality bounds exclude unphysical solutions and prevent algorithms from becoming trapped too easily in a (false) local minimum.

Numerical methods for PDE-constrained nonlinear optimization with inequality constraints usually fall into one of two categories: active-set strategies and interior-point (IP) methods. Because of the combinatorial complexity of active-set algorithms, we opt for interior-point methods instead [7, 18, 17], which we apply to an appropriate finite-dimensional discrete version of (1.1).

Since the PDE constraints in (1.1) are only coupled through $u$, it may seem at first attractive to eliminate the state variables $y_1, \ldots, y_{N_E}$ via the PDE constraints. Known as reduced-space optimization, that approach capitalizes on well-established methods and software for solving the (nonlinear) PDEs [15, 16, 20, 27]. However, the resulting Hessian matrices needed by Newton's method are dense and cannot be formed explicitly. It requires the repeated exact solution of the state (and adjoint) equations, even during initial steps when the control variables are still far from their optimum, which is computationally inefficient. Moreover, the evaluation of the objective function during the line-search phase becomes as expensive as the solution of the full forward problem. Given those difficulties it is not surprising that most numerical methods for large-scale PDE-constrained optimization are based on gradients alone, thereby avoiding Hessian matrices entirely. Unlike Newton methods, however, gradient-based methods, such as nonlinear conjugate gradients or limited memory quasi-Newton methods [24], do not scale well with increasing problem size, suffering a reduction from quadratic to linear asymptotic convergence, and often fail to converge for ill-conditioned problems.

In contrast, full-space optimization [4, 5, 13, 25] treats at once both state and control as independent optimization variables that are coupled through equality or inequality constraints. Then the Hessian matrix, though very large, is also very sparse and therefore ideally suited for iterative solvers. For nonconvex problems, however, the projection of the Hessian onto the nullspace of the constraint Jacobian may not be positive definite; hence, the resulting search direction may point towards a saddle-point or a maximizer, instead of a minimizer. To enforce a direction of descent, the Hessian matrix must be appropriately modified. When a matrix factorization is explicitly available, the number of positive, negative and zero eigenvalues is easily obtained and can be used to modify the Hessian, if needed, and thus enforce a descent direction. Iterative methods, however, do not provide such information. As a consequence, most work on inexact Newton methods for large-scale PDE-constrained optimization usually avoids the indefiniteness of the Hessian either by neglecting part of it (Gauss-Newton approximation) and thus sacrificing speed of convergence [3], or simply by assuming convexity of the objective function [2].

Recently, Curtis et al. [9, 11] proposed a full-space inexact interior-point method that guarantees global convergence, while retaining the full Lagrangian Hessian information and thus the fast local convergence of Newton's method. The method was extended to inequality constraints in [12] and further enhanced for additional
efficiency in [10]. There, an algebraic multilevel preconditioner for general sparse indefinite linear systems was used within the inexact Newton step [6], and it led to significant speed-up over an algorithm based on direct factorization of the primal-dual system. Though encouraging, that approach does not extend to truly large-scale problems with multiple PDE constraints, because the preconditioner requires assembly and global reordering of the entire Hessian matrix; hence, it is actually restricted to shared-memory architectures.

Following [10], we present in Section 2 the inexact interior-point (IIP) framework and recall the salient points of the algorithm. Next in Section 3, we describe our reduced-space preconditioner (RSP) for the full Hessian matrix. Together they yield the RSP-IIP method, which we apply to a series of numerical examples in Section 4 to demonstrate its efficiency and robustness. Here we consider both PDE-constrained control and inverse problems, either in two or three space dimensions: all but one are nonconvex. Finally, we summarize our findings in Section 5.

2. Interior-Point Framework. Numerical approximation of (1.1) leads to its discrete counterpart,

\[
\min_{y, u} F(y, u) = \frac{1}{2} \sum_{k=1}^{N_E} \|V y_k - \hat{y}_k\|^2 + \frac{\alpha}{2} R(u) \\
\text{s.t. } A_k(y_k, u) = 0, \quad \text{for } k = 1, \ldots, N_E \\
u_- \leq u \leq u_+, 
\]

where bold symbols denote finite-dimensional vector or matrix approximations of the corresponding functions or operators from the continuous formulation. The objective function now consists of two parts. The misfit term measures the deviation from some desired state or measurements \(\hat{y}_k\), where \(V\) is a projection or evaluation matrix. The Tikhonov regularization term \(R(u)\) with parameter \(\alpha\) is standard and removes ill-posedness.

Following a classical interior-point (IP) strategy, we now introduce slack variables \(s \in \mathbb{R}^{2n_u}, s \geq 0\) and a barrier term with parameter, \(\mu > 0\), for the inequality constraints. Then (2.1) is solved through a sequence of barrier subproblems

\[
\min_{y, u, s} \varphi(y, u, s; \mu) := \frac{1}{2} \sum_{k=1}^{N_E} \|V y_k - \hat{y}_k\|^2 + \frac{\alpha}{2} R(u) - \mu \sum_{i=1}^{2n_u} \log(s_i) \\
\text{s.t. } A_k(y_k, u) = 0, \quad \text{for } k = 1, \ldots, N_E, \\
\quad s = \begin{pmatrix} u - u_- \\ u_+ - u \end{pmatrix}
\]

for a sequence of barrier parameters \(\mu \downarrow 0\). If the objective function and the constraints are sufficiently smooth, the limit of the corresponding solutions of (2.2) satisfies first-order optimality conditions for (2.1) when the constraint Jacobian has full rank [24].

Solutions of (2.2) are critical points of the Lagrangian function

\[
L(y, u, s, \lambda) = \frac{1}{2} \sum_{k=1}^{N_E} \|V y_k - \hat{y}_k\|^2 + \frac{\alpha}{2} R(u) - \mu \sum_{i=1}^{2n_u} \log(s_i) \\
\quad \quad \quad \quad + \sum_{k=1}^{N_E} \lambda_k^T A_k(y_k, u) + \lambda^T \left( \begin{pmatrix} u - u_- \\ u_+ - u \end{pmatrix} - s \right),
\]
and thus satisfy the Karush-Kuhn-Tucker (KKT) conditions

\[
\nabla_y \lambda_k \left( y, u, s, \lambda \right) = V^\top (V y_k - \hat{y}_k) + J_{y_k}^\top \lambda_k = 0 \quad \text{for } k = 1, \ldots, N_E,
\]

\[
\nabla_u \lambda_k \left( y, u, s, \lambda \right) = \frac{\alpha}{2} \nabla_u R + \sum_{k=1}^{N_E} J_{ku}^T \lambda_k + \hat{I}^T \lambda_T = 0,
\]

(2.3)

\[
S \nabla_s \lambda_k \left( y, u, s, \lambda \right) = -\mu e - S \lambda_T = 0,
\]

\[
\nabla \lambda_c \left( y, u, s, \lambda \right) = A_k(y_k, u) = 0 \quad \text{for } k = 1, \ldots, N_E,
\]

\[
\nabla \lambda_z \left( y, u, s, \lambda \right) = \begin{pmatrix} u + u_+ - u \end{pmatrix} - s = 0,
\]

where pre-multiplication by \( S = \text{diag}(s) \) in the third equation leads to a primal-dual IP method [24]. Moreover, \( J_{y_k} = \nabla y_A \left( y_k, u \right) \) and \( J_{ku} = \nabla u A_k \left( y_k, u \right) \) denote the Jacobians of the \( k \)-th PDE constraint with respect to \( y_k \) and \( u \), respectively, \( \hat{I}^T = [I - I]^T \) denotes the Jacobians of the inequality constraints, and \( e \) is a vector with all entries equal to one. The interior-point method terminates, once the current iterate \( (y, u) \) satisfies (2.3) with \( \mu = 0 \) within a prescribed tolerance \( \varepsilon_{\text{tol}} \), that is

(2.4)

\[
\| \nabla L \|_\infty \leq \varepsilon_{\text{tol}}
\]

with \( \mu = 0 \).

At an iterate \( (y, u) \), we compute a primal-dual search direction \( (d_p, d_d) \), \( d_p = (d_y, d_u, d_s) \) and \( d_d = (d_{\lambda_c}, d_{\lambda_z}) \), which satisfies appropriate conditions for guaranteeing global convergence [10]. The computation of the search direction is based on Newton’s method applied to the KKT conditions (2.3) and hence involves the full Lagrangian Hessian, \( H \). To simplify notation, we now group the primal and dual variables as \( z = (y, u, s) \) and \( \lambda = (\lambda_c, \lambda_z) \), where \( \lambda_c \) denotes the vector of Lagrange multipliers \( \lambda_1, \ldots, \lambda_{N_E} \) associated with the PDE (equality) constraints. We further let

\[

\nabla_p L = \begin{pmatrix} \nabla_y L \\ \nabla_u L \\ \nabla_s L \end{pmatrix}, \quad c = \begin{pmatrix} \nabla \lambda_c L \\ \nabla \lambda_z L \end{pmatrix}, \quad J = \begin{pmatrix} J_y & J_u & 0 \\ 0 & I & -\Sigma \end{pmatrix},
\]

and \( \Sigma \) denote a positive definite diagonal scaling matrix for the slack variables.

For nonconvex problems, the projection of \( H \) onto the nullspace of the constraint Jacobian may not be positive definite; hence, the search direction \( (d_p, d_d) \) may point towards a saddle-point or a maximizer, instead of a minimizer. To enforce a direction of descent, we regularize \( H \) and modify the first three diagonal blocks of the Hessian by adding a multiple of the identity matrix \( \delta I \). Thus at each Newton step we actually solve the linear system

(2.5)

\[
\begin{pmatrix}
H_{yy} + \delta I & H_{yu} & 0 & J_y^\top & 0 \\
H_{yu}^\top & H_{uu} + \delta I & 0 & J_u^\top & \hat{I}^\top \\
0 & 0 & \Sigma(S^{-1} \lambda_T + \delta I) \Sigma & 0 & -\Sigma \\
J_y & J_u & 0 & 0 & 0 \\
0 & I & -\Sigma & 0 & 0
\end{pmatrix}
\begin{pmatrix}
d_y \\
d_u \\
d_s \\
d_{\lambda_c} \\
d_{\lambda_z}
\end{pmatrix} =
\begin{pmatrix}
\nabla y_A L \\
\nabla u_A L \\
\nabla s_A L \\
\nabla \lambda_c L \\
\nabla \lambda_z L
\end{pmatrix},
\]

where \( A_T = \text{diag}(\lambda_T) \) are the Lagrange multipliers associated with the inequality constraints. The linear system (2.5) is very large but also very sparse and thus well-suited for iterative solvers. Clearly an effective preconditioner is essential to keep the
number of iterations low. In Section 3, we shall propose a general purpose effective and parallel preconditioner.

2.1. Termination criteria. When we replace the direct solution of (2.5) by an iterative method, both error control and the handling of nonconvexity become non-trivial. Indeed if (2.5) is no longer solved exactly, how large an error will nonetheless guarantee global convergence? Moreover, when a matrix factorization is explicitly available, the number of positive, negative and zero eigenvalues is easily obtained and can be used to modify the Hessian, if needed, and thus enforce a descent direction. Iterative methods, however, do not provide such information. Answers to these two questions were given in [11, 9, 12], where a series of sufficient merit function approximation reduction termination tests (SMART tests, for short) were devised that allow inexact solutions of (2.5) and yet guarantee global convergence.

Of central importance in those termination tests are the residual vectors

\[
\begin{pmatrix}
    r_d \\
    r_p
\end{pmatrix} = 
\begin{pmatrix}
    H_{pp} & J^\top \\
    J & 0
\end{pmatrix}
\begin{pmatrix}
    d_p \\
    d_d
\end{pmatrix} + \begin{pmatrix}
    \nabla_p L \\
    c
\end{pmatrix},
\]

where

\[
H_{pp} = 
\begin{pmatrix}
    H_{yy} + \delta I & H_{yu} \\
    H_{yu}^\top & H_{uu} + \delta I
\end{pmatrix} + 0
\]

as well as the primal-dual relative residuals of (2.5),

\[
\Psi := \frac{\begin{pmatrix}
    r_d \\
    r_p
\end{pmatrix}}{\begin{pmatrix}
    \nabla_p L \\
    c
\end{pmatrix}}.
\]

For convex problems the algorithm can focus exclusively on \(\Psi\), terminating the calculation of \((d_p, d_d)\) whenever this value is below a threshold. For nonconvex problems, however, the priority is to find solutions to (2.3) that correspond to minimizers, not saddle-points or maximizers. The methods developed in [11, 9, 12] therefore include additional conditions and procedures that enforce convergence toward minimizers of (2.1). These additional conditions involve a local model of the merit function \(\phi\), denoted as

\[
m(d_p; \mu, \pi) := \phi(z; \mu, \pi) + \nabla_z \phi(z)^\top d_p + \pi \|c + Jd_p\|
\]

and the reduction in this model yielded by \(d_p\), which is defined as

\[
\Delta m(d_p; \mu, \pi) := m(0; \mu, \pi) - m(d_p; \mu, \pi)
\]

\[
= -\nabla_z \phi(z)^\top d_p + \pi (\|c\| - \|c + Jd_p\|).
\]

Once the primal-dual search direction \((d_p, d_d)\) has been determined, we compute the scaled direction \(d_p = (d_y, d_u, \Sigma d_s)\) along which a line search is performed. The line search involves two conditions. First, to maintain positivity of the slacks, a step size \(\alpha^{\text{max}} \in (0, 1]\) satisfying

\[
s + \alpha^{\text{max}} \Sigma d_s \geq (1 - \eta_1) s
\]

is determined for a constant \(\eta_1 \in (0, 1)\); we use \(\eta_1 = \max(0.99, 1 - \mu)\), which converges to one as \(\mu \downarrow 0\) and thus preserves fast local convergence. Second, the algorithm
the last term in (2.10) is zero. Thus, a stationary point of (2.10) satisfies
where the maximum is meant componentwise. For any feasible PDE parameter
whether it solves the problem
Step 2, we check whether the current iterate is an infeasible stationary point, that
is
Finally for the dual variables, the step length \( \beta \) is set to the smallest value in \([\alpha, 1]\)
that leads to a dual infeasibility reduction at least as large as a full Newton step, that is
\( \beta = \min \left\{ \beta \in [\alpha, 1] \mid \| (c(z, \lambda + \beta d_d)) \| \leq \| (c(z, \lambda + d_d)) \| \right\} \).

The interior-point framework is summarized in Algorithm 1. In particular in
Step 2, we check whether the current iterate is an infeasible stationary point, that is,
whether it solves the problem
\[
\min_{x \in \mathbb{R}^n} \frac{1}{2} \sum_{k=1}^{N_E} \| A_k(y_k, u) \|^2 + \frac{1}{2} \| \max \left\{ \frac{u_- - u}{u - u_+}, 0 \right\} \|^2,
\]
where the maximum is meant componentwise. For any feasible PDE parameter \( u \),
the last term in (2.10) is zero. Thus, a stationary point of (2.10) satisfies
\[
(J_y, J_u)^\top A_k(y_k, u) = 0.
\]
Since \( J_y \) originates from a (well-posed) discretized and linearized PDE, it ought to be invertible; hence, \( A_k(y, u) = 0 \) and \( (y, u) \) is feasible. Therefore, the termination in Step 2 will never occur, as confirmed by our numerical experiments in Section 4.

3. Reduced-space preconditioning. At each Newton iteration, we must solve the very large but sparse linear system (2.5), where \( H_{yy} \) and \( H_{uu} \) are symmetric. Moreover, since the PDE constraints in (2.1) are only coupled through the control variable \( u \) and the objective function \( F \) is additive, the largest dominant block \( H_{yy} \) is in fact block diagonal,

\[
H_{yy} = \begin{pmatrix} V^T V + \sum_i (\lambda_i) \nabla_{yy}^2 (A_i) \vdots \cdots \vdots \nabla_{yy}^2 (A_N) \end{pmatrix},
\]

where \((A_k)_i\) is the \(i\)th component of \( A_k(y, u)\). Similarly, the constraint Jacobian \( J_y \) only consists of diagonal blocks,

\[
J_y = \begin{pmatrix} J_{y1} & \cdots & \cdots & J_{yN} \end{pmatrix}. 
\]

In fact if the partial differential operators \( A_k(y, u) \) in (1.1) are linear in \( y \), that is if \( A_k(y, u) = A_k(u) y \), then each \( J_{y_i} = A_k(u) \). Moreover, when all the PDEs are identical, all the diagonal blocks in \( H_{yy} \) or \( J_y \) coincide.

First, we permute the linear system (2.5) to swap the (2,2) and (4,4) blocks, and denote by \( D \) the resulting matrix, which we rewrite as

\[
D = \begin{pmatrix} Q & V \\ V^T & P \end{pmatrix},
\]

with \( \delta = 0 \) for simplicity, where

\[
Q = \begin{pmatrix} H_{yy} & J_y \\ J_y & 0 \end{pmatrix}, \quad P = \begin{pmatrix} \Sigma S^{-1} A \Sigma & 0 & -\Sigma \\ 0 & H_{uu} & I^T \\ -\Sigma & I & 0 \end{pmatrix}, \quad V = \begin{pmatrix} 0 & H_{yu} & 0 \\ 0 & J_u & 0 \end{pmatrix}.
\]

Typically the size of \( P \) is much smaller than the size of \( Q \). Indeed a very large number of PDE parameters can lead to unphysical optimal solutions [28, 22]. Then it is particularly attractive to pursue the solution of the original linear system by forming and solving first the Schur-complement system with respect to the (2,2) block \( P \) of \( D \), given by

\[
G = P - V^T Q^{-1} V.
\]

The particular structure of \( Q \) allows its inverse to be written explicitly as

\[
Q^{-1} = \begin{pmatrix} 0 & J_y^{-1} \\ J_y^{-1} & -J_y^{-1} H_{yy} J_y^{-1} \end{pmatrix}.
\]

Therefore the solution of linear systems with system matrix \( Q \) appears trivial provided that a robust and efficient solver for the inversion of \( J_y \) is available.
Hence, let the (permuted) linear system take the form
\[
\begin{pmatrix}
Q & V \\
V^\top & P
\end{pmatrix}
\begin{pmatrix}
d_{y\lambda_e} \\
d_{us}
\end{pmatrix} =
\begin{pmatrix}
w_1 \\
w_2
\end{pmatrix},
\]
where \(d_{y\lambda_e}\) stands for \((d_y^\top d_{\lambda_e}^\top)\) and \(d_{us}\) for \((d_s^\top, d_u^\top, d_{\lambda_e}^\top)\). First, we transform it to
\[
\begin{pmatrix}
Q & V \\
0 & P - V^\top Q^{-1} V
\end{pmatrix}
\begin{pmatrix}
d_{y\lambda_e} \\
d_{us}
\end{pmatrix} =
\begin{pmatrix}
w_1 \\
w_2 - V^\top Q^{-1} w_1
\end{pmatrix}
\]
which we then solve by using the following algorithm. Clearly we never explicitly form the dense Schur-complement \(G\) in the third step of Algorithm 2, but instead again use GMRES preconditioned by the matrix \(P\).

Algorithm 2 Reduced-Space Preconditioned GMRES Solver (RSP-GMRES)

1: Solve \(Q p_1 = w_1\)
2: Form \(p_2 := w_2 - V^\top p_1\)
3: Solve \(G d_{us} = p_2\), where \(G = P - V^\top Q^{-1} V\)
4: Form \(p_1 := w_1 - V d_{us}\)
5: Solve \(Q d_{y\lambda_e} = p_1\)

Since \(H_{yy}\) and \(J_y\) in (3.1)–(3.2) are both block-diagonal, the PDEs can be solved independently of one another and in parallel. Hence if every diagonal block is assigned to a separate processor, the total execution time becomes constant and independent of \(N_E\). When the PDEs \(A_k(y_k, u)\) are linear in \(y_k\) and the diagonal blocks \(J_{y_k}, k = 1, \ldots, N_E\) all the same, just a single block needs to be factorized yielding linear run-time complexity with respect to \(N_E\), even on a single processor.

4. Numerical Results. To demonstrate the effectiveness of our inexact interior-point (IIP) method, we shall now apply it to four different PDE-constrained optimization problems. In the first two examples from optimal control, the control variable \(u\) appears on the right-hand side of the PDE constraint and attempts to steer the system’s state, \(y\), towards a desired state \(\hat{y}\). In the last two examples from parameter estimation, the spatially distributed model parameter \(u\) appears inside the differential operator itself, which typically leads to a nonconvex inverse problem. Here, additional knowledge about the true model \(\hat{u}\), such as inequality constraints or multiple measurements, not only reduce the number of false local minima, but also mitigates the effect of noise in the observations \(\hat{y}\).

In all examples, we follow the “discretize-then-optimize” approach, where the objective function and the PDE-constraints are first approximated numerically before applying our IIP method to the resulting finite-dimensional nonlinear optimization problem. To determine the new search direction at each Newton iteration, we apply to the (linear) KKT system (2.5) the reduced-space preconditioned (RSP) GMRES method, described in Section 3; in particular, prior to the GMRES iteration, the PDE block matrix \(J_y\) in (3.3) is factorized.

Once the GMRES iterate meets the desired tolerance, it is evaluated by the SMART tests. If the termination tests (see Algorithms 2 and 3 in [10] for details) accept the new search direction, the optimization method proceeds with the step length computation, see Step 5 of Algorithm 1. If a Hessian modification is required, the Hessian block of the KKT system is modified as in (2.5) and the RSP-GMRES method is
restarted. If none of the previous cases apply, the desired relative residual is reduced by a factor 10 while the RSP-GMRES iteration and SMART acceptance procedure are repeated. Due to (Lemma 3.2, [12]) this process will eventually terminate.

Both the IIP algorithm and the exact IP method, later used for the sake of comparison, have been implemented in the Ipopt open-source optimization package (rev. 2004). Unless noted otherwise, we always set the first desired relative residual to $10^{-2}$ and use default parameter values elsewhere for the inexact algorithm. The RSP-preconditioner is implemented in C++ and uses a linear solver based on an $LU$-factorization from the Pardiso software package (version 4.1.2); again we use default settings unless noted otherwise. The IIP algorithm stops once the optimality conditions (2.3) are satisfied within a tolerance of $10^{-8}$ in the maximum norm.

All comparisons were performed on an Intel Xeon architecture with 128 GB main memory using Intel's compiler version 10.1 under CentOS 5.8.

4.1. 2D Distributed Control. To illustrate the efficiency of our RSP-IIP approach, in particular with respect to the mesh size $h$, we now consider a PDE-constrained quadratic program, for which a multigrid based preconditioned projected conjugate gradient (PPCG) method was developed in [26]. Note that the PPCG algorithm requires a positive definite Hessian on the null-space of the Jacobian in (2.5); therefore, it cannot be applied to more difficult nonconvex problems, as we shall discuss in Sections 4.2–4.4.

Hence, we consider the following convex optimal control problem, either with a “corner load” or a “centered load” desired state $\hat{y}$:

$$
\min_{y,u} F(y,u) = \frac{1}{2} ||y - \hat{y}||_{L^2(\Omega)}^2 + \frac{\alpha}{2} ||u||_{L^2(\Omega)}^2,
$$

s.t. $\Delta y = u$ in $\Omega = (0,1)^2$,

$y = g$ on $\partial\Omega$,

with $\alpha = 0.02$. For the corner load problem we set $g = \hat{y}|_{\partial\Omega}$, where

$$
\hat{y}(x_1,x_2) = \begin{cases} 
(2x_1 - 1)^2(2x_2 - 1)^2 & \text{if } (x_1,x_2) \in [0, \frac{1}{2}]^2, \\
0 & \text{otherwise},
\end{cases}
$$

\end{figure}

\begin{figure}
\centering
\begin{subfigure}{0.3\textwidth}
\centering
\includegraphics[width=\textwidth]{optimal_control}
\caption{Optimal Control $u^*$}
\end{subfigure}
\begin{subfigure}{0.3\textwidth}
\centering
\includegraphics[width=\textwidth]{optimal_state}
\caption{Optimal State $y^*$}
\end{subfigure}
\begin{subfigure}{0.3\textwidth}
\centering
\includegraphics[width=\textwidth]{desired_state}
\caption{Desired State $\hat{y}$}
\end{subfigure}
\caption{Corner load problem: Optimal control $u^*$, optimal state $y^*$ and desired state $\hat{y}$ with $N_h = 64$}
\end{figure}

\footnotesize

\begin{itemize}
\item \[ \text{http://www.coin-or.org/Ipopt/} \]
\item \[ \text{http://www.pardiso-project.org/} \]
\end{itemize}
sequel. nonconvex problems, possibly with inequality constraints, as we shall discuss in the sequel. Recall, however, that the PPCG method cannot be applied to more general optimization problems and compare run-times and iteration counts with those from the RSP-GMRES, with varying tolerance $\tau$ and mesh size $N_h \times N_h$, leading to a KKT system of size $n$. All optimization problems were solved within a single (inexact) optimization step. In all cases, the number of iterations remains independent of $N_h$; hence, our RS preconditioner also exhibits optimal $h$-independent behavior. The run-times of both algorithms are comparable, while the iteration counts differ by at most two. Recall, however, that the PPCG method cannot be applied to more general nonconvex problems, possibly with inequality constraints, as we shall discuss in the sequel.

<table>
<thead>
<tr>
<th>$N_h$</th>
<th>$n$</th>
<th>PPCG</th>
<th>RSP-GMRES</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>98</td>
<td>0.05 (3)</td>
<td>0.01 (3)</td>
</tr>
<tr>
<td>16</td>
<td>450</td>
<td>0.05 (2)</td>
<td>0.02 (3)</td>
</tr>
<tr>
<td>32</td>
<td>1'922</td>
<td>0.07 (2)</td>
<td>0.05 (3)</td>
</tr>
<tr>
<td>64</td>
<td>7'938</td>
<td>0.17 (1)</td>
<td>0.24 (3)</td>
</tr>
<tr>
<td>128</td>
<td>32'258</td>
<td>0.73 (1)</td>
<td>1.30 (3)</td>
</tr>
<tr>
<td>256</td>
<td>130'050</td>
<td>4.41 (1)</td>
<td>6.33 (3)</td>
</tr>
<tr>
<td>512</td>
<td>522'242</td>
<td>23.0 (1)</td>
<td>30.2 (3)</td>
</tr>
</tbody>
</table>

Centered load problem: Run-time [s] and iteration count (in parenthesis) for PPCG and RSP-GMRES with varying tolerance $\tau$ and mesh size $N_h \times N_h$, leading to a KKT system of size $n$. All optimization problems were solved within a single inexact optimization step.

<table>
<thead>
<tr>
<th>$N_h$</th>
<th>$n$</th>
<th>PPCG</th>
<th>RSP-GMRES</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>98</td>
<td>0.05 (2)</td>
<td>0.02 (3)</td>
</tr>
<tr>
<td>16</td>
<td>450</td>
<td>0.05 (2)</td>
<td>0.05 (3)</td>
</tr>
<tr>
<td>32</td>
<td>1'922</td>
<td>0.07 (2)</td>
<td>0.05 (3)</td>
</tr>
<tr>
<td>64</td>
<td>7'938</td>
<td>0.22 (2)</td>
<td>0.24 (3)</td>
</tr>
<tr>
<td>128</td>
<td>32'258</td>
<td>0.99 (2)</td>
<td>1.30 (3)</td>
</tr>
<tr>
<td>256</td>
<td>130'050</td>
<td>5.54 (2)</td>
<td>6.35 (3)</td>
</tr>
<tr>
<td>512</td>
<td>522'242</td>
<td>27.2 (2)</td>
<td>30.3 (3)</td>
</tr>
</tbody>
</table>

whereas for the centered load problem we let $g = 0$ and

$$
\hat{y}(x_1, x_2) = \exp \left(-\frac{(x_1 - 0.5)^2 + (x_2 - 0.5)^2}{0.125^2}\right).
$$

The state and control variables $y, u$ are discretized on a regular $N_h \times N_h$ grid with $Q_1$ finite elements. Next, we apply the RSP-IIP method with $\mu_{init} = 10^{-11}$ to both problems and compare run-times and iteration counts with those from the MATLAB implementation of the PPCG algorithm in [26]. The KKT systems (2.5) are solved iteratively either with PPCG or RSP-GMRES until the relative residual has reached a desired tolerance $\tau = 10^{-6}$ or $10^{-12}$. Regardless of problem size, the IIP algorithm, though inexact, always found a solution within a single optimization step, thus demonstrating the remarkable accuracy of its search direction.

In Tables 4.1 and 4.2, we compare run-times and iteration counts (in parenthesis) of the PPCG and the RSP-IIP method to solve the KKT system for varying mesh size and tolerance. In all cases, the number of iterations remains independent of $N_h$; hence, our RS preconditioner also exhibits optimal $h$-independent behavior. The run-times of both algorithms are comparable, while the iteration counts differ by at most two. Recall, however, that the PPCG method cannot be applied to more general nonconvex problems, possibly with inequality constraints, as we shall discuss in the sequel.
4.2. 2D Boundary Control Problem. Next, to demonstrate the robustness of the RSP-IIP method with respect to increasing mesh size and number of PDE constraints, we consider a 2D boundary control problem (example 5.7, [23]) with multiple nonlinear PDEs and inequality constraints. Since the PPCG algorithm no longer applies here, we shall compare the RSP-IIP method to the exact IP method from [33] instead, where the KKT systems are solved by the sparse direct solver Pardiso [29]. By exploiting the sparsity structure of the KKT system, the RSP-IIP method achieves a significant speed-up over the exact IP method, even more so as the KKT system increases.

Hence, we consider the optimal boundary control problem:

\[
\begin{align*}
\min_{y,u} F(y,u) &= \frac{1}{2N_E} \sum_{k=1}^{N_E} ||y_k - \hat{y}_k||^2_{L^2(\Omega)} + \frac{\alpha}{2} ||u||^2_{L^2(\partial\Omega)}, \\
\text{s.t.} & -\Delta y_k - y_k + y_k^3 = 0 \quad \text{in } \Omega = (0,1)^2, \\
& \frac{\partial y_k}{\partial n} = ku \quad \text{on } \partial\Omega = (0,1)^2,
\end{align*}
\]

where

\[
\hat{y}_k = k(2 - x_1(x_1 - 1) + x_2(x_2 - 1)), \quad k = 1, \ldots, N_E,
\]

and \( \alpha = 0.01 \). Here (4.1a) describes the interaction of normalized quantum mechanical wave functions of electrons \( y_k \) in a superconductor on the basis of a simplified Ginzburg-Landau model [32]. For each \( k = 1, \ldots, N_E \) the corresponding PDE constraint (4.1a) is discretized with standard second-order finite-differences on a regular \( N_h \times N_h \) grid.

In Figure 2, the optimal control \( u^* \) and optimal state \( y^* \) are shown for a single PDE constraint, that is \( N_E = 1 \) and \( N_h = 100 \). Both are initialized as \( u \equiv 2.15 \) and \( y = \hat{y}_k \), respectively. Note that the bounds (4.1b) on \( u \) are partially active and the KKT system thus becomes increasingly ill-conditioned as \( \mu \to 0 \) because of the barrier term in (2.2).

Next, in Figure 3, we show the number of RSP-GMRES iterations ('\( \times \)') and the relative residual of each iterate accepted by the SMART test ('\( \checkmark \)') for varying \( N_E \) and
Fig. 3. 2D boundary control: number of RSP-GMRES iterations (×) and relative residual (○) at SMART acceptance for varying number of PDE constraints, $N_E$, and mesh sizes $N_h \times N_h$. Note the different scales on the left and right axes.

$N_h$. Independently of $N_E$ or the mesh size, $N_h \times N_h$, the RSP-IIP method always converges within 20 optimization steps, while the RSP-GMRES iterates satisfy the SMART test within only five iterations.

In Table 4.3 we list for varying $N_h$ and $N_E$ the size of the resulting KKT system $n$, the total run-time, and the number of optimization steps for the RSP-IIP and the exact IP method. The number of optimization steps needed by the RSP-IIP method exceeds at most by four that of the exact IP method; it barely increases as the KKT system grows one thousand fold. By taking advantage of the KKT system’s sparsity structure, the RSP-IIP method achieves a speedup of 36, and even more so with growing problem size. As the run-time per inexact interior-point iteration scales linearly with respect to the number of PDE constraints, $N_E$, while the number of optimization steps barely increases, we obtain essentially linear run-time complexity. Moreover, if the different diagonal blocks in $J_y$ and $J^T_y$ in (3.2), all independent of
In 2D, the source terms are given by
\[ \text{source}, \text{measurements } \hat{x} \text{ with } \tilde{x}, \]
with (4.2) being the estimation problem:

The key ingredient for its efficiency depends on the number of PDE constraints,
and the RSP-IIP algorithm with respect to the number of PDE constraints,
and practical applications often lead to multiple PDE constraints. Then, the (optimal) linear complexity achieved by the RSP-IIP algorithm with respect to the number of PDE constraints, \( N_E \), becomes a key ingredient for its efficiency.

Thus, we let \( \Omega = (0,1)^d \), where \( d = 2, 3 \), and consider the spatially distributed parameter estimation problem:

\[
\min_{y,u} F(y,u) = \frac{1}{2N_E} \sum_{k=1}^{N_E} \|v(y_k) - \hat{y}_k\|_2^2 + \frac{\alpha}{2} \left( V(u) + \beta \|u\|_2^2 \right),
\]

(4.2) subject to:
- \( -\nabla \cdot (\exp(u)\nabla y_k) = q_k \) in \( \Omega, k = 1, \ldots, N_E \)
- \( y_k = 0 \) on \( \partial\Omega \)
- \( -2 \leq u \leq 2 \)

Here, the true log conductivity is \( \hat{u} \) given by

\[ \hat{u}(x) = \exp \left( -\frac{||\tilde{x}_a - x||^2}{0.05} \right) - \exp \left( -\frac{||\tilde{x}_b - x||^2}{0.05} \right), \]

with \( \tilde{x}_a = (0.25, 0.25)^T, \tilde{x}_b = (0.75, 0.75)^T \) in two dimensions, or \( \tilde{x}_a = (0.25, 0.25, 0.25)^T, \tilde{x}_b = (0.75, 0.75, 0.75)^T \) in three dimensions, respectively. To generate the synthetic measurements \( \hat{y}_k, k = 1, \ldots, N_E \), we numerically solve the forward problem for a given source, \( q_k \), and add 1% of white noise to its solution.

In 2D, the source terms are given by

\[ q_{4l+1}(x) = \sin(\nu_1(l) 2\pi x_1) \sin(\nu_2(l) 2\pi x_2), \]
\[ q_{4l+2}(x) = \cos(\nu_1(l) 2\pi x_1) \sin(\nu_2(l) 2\pi x_2), \]
\[ q_{4l+3}(x) = \sin(\nu_1(l) 2\pi x_1) \cos(\nu_2(l) 2\pi x_2), \]
\[ q_{4l+4}(x) = \cos(\nu_1(l) 2\pi x_1) \cos(\nu_2(l) 2\pi x_2), \]

one another, were solved in parallel, the total execution time would essentially become constant and thus independent of \( N_E \).

### 4.3. 2D and 3D Parameter Estimation

We now consider two problems from groundwater modeling [14], the first in two and the second in three space dimensions, where the log conductivity, \( u(x) \), needs to be estimated from noisy measurements of the fluid pressure, \( y(x) \). Since the influence of noise can be further reduced by including measurements from multiple sources, practical applications often lead to multiple PDE constraints. Then, the (optimal) linear complexity achieved by the RSP-IIP algorithm with respect to the number of PDE constraints, \( N_E \), becomes a key ingredient for its efficiency.

Thus, we let \( \Omega = (0,1)^d, d = 2, 3 \), and consider the spatially distributed parameter estimation problem:

\[
\min_{y,u} F(y,u) = \frac{1}{2N_E} \sum_{k=1}^{N_E} \|v(y_k) - \hat{y}_k\|_2^2 + \frac{\alpha}{2} \left( V(u) + \beta \|u\|_2^2 \right),
\]

(4.2) subject to:
- \( -\nabla \cdot (\exp(u)\nabla y_k) = q_k \) in \( \Omega, k = 1, \ldots, N_E \)
- \( y_k = 0 \) on \( \partial\Omega \)
- \( -2 \leq u \leq 2 \)

Here, the true log conductivity is \( \hat{u}(x) \) given by

\[ \hat{u}(x) = \exp \left( -\frac{||\tilde{x}_a - x||^2}{0.05} \right) - \exp \left( -\frac{||\tilde{x}_b - x||^2}{0.05} \right), \]

with \( \tilde{x}_a = (0.25, 0.25)^T, \tilde{x}_b = (0.75, 0.75)^T \) in two dimensions, or \( \tilde{x}_a = (0.25, 0.25, 0.25)^T, \tilde{x}_b = (0.75, 0.75, 0.75)^T \) in three dimensions, respectively. To generate the synthetic measurements \( \hat{y}_k, k = 1, \ldots, N_E \), we numerically solve the forward problem for a given source, \( q_k \), and add 1% of white noise to its solution.

In 2D, the source terms are given by

\[ q_{4l+1}(x) = \sin(\nu_1(l) 2\pi x_1) \sin(\nu_2(l) 2\pi x_2), \]
\[ q_{4l+2}(x) = \cos(\nu_1(l) 2\pi x_1) \sin(\nu_2(l) 2\pi x_2), \]
\[ q_{4l+3}(x) = \sin(\nu_1(l) 2\pi x_1) \cos(\nu_2(l) 2\pi x_2), \]
\[ q_{4l+4}(x) = \cos(\nu_1(l) 2\pi x_1) \cos(\nu_2(l) 2\pi x_2), \]

\[ q_{4l+1}(x) = \sin(\nu_1(l) 2\pi x_1), \]
\[ q_{4l+2}(x) = \cos(\nu_1(l) 2\pi x_1), \]
\[ q_{4l+3}(x) = \sin(\nu_1(l) 2\pi x_1) \cos(\nu_2(l) 2\pi x_2), \]
\[ q_{4l+4}(x) = \cos(\nu_1(l) 2\pi x_1) \cos(\nu_2(l) 2\pi x_2), \]
Fig. 4. 2D log conductivity estimation: true model and optimal solution with \( N_E = 10 \) PDE constraints on a \( N_h \times N_h \) mesh with \( N_h = 100 \)

Table 4.4

2D log conductivity estimation: size of the KKT system, \( n \), number of optimization steps and total run-time for the RSP-IIP method for varying number of PDE constraints, \( N_E \), and mesh size \( N_h \times N_h \)

<table>
<thead>
<tr>
<th>( N_h \times N_h )</th>
<th>( N_E )</th>
<th>( n )</th>
<th># IIP steps</th>
<th>run-time [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>100 100</td>
<td>5</td>
<td>151’005</td>
<td>9</td>
<td>25</td>
</tr>
<tr>
<td>100 100</td>
<td>10</td>
<td>251’005</td>
<td>9</td>
<td>38</td>
</tr>
<tr>
<td>100 100</td>
<td>20</td>
<td>451’005</td>
<td>9</td>
<td>60</td>
</tr>
<tr>
<td>200 100</td>
<td>5</td>
<td>602’005</td>
<td>9</td>
<td>109</td>
</tr>
<tr>
<td>200 100</td>
<td>10</td>
<td>1’002’005</td>
<td>9</td>
<td>155</td>
</tr>
<tr>
<td>200 200</td>
<td>10</td>
<td>1’802’005</td>
<td>9</td>
<td>256</td>
</tr>
<tr>
<td>400 100</td>
<td>5</td>
<td>2’404’005</td>
<td>11</td>
<td>579</td>
</tr>
<tr>
<td>400 100</td>
<td>10</td>
<td>4’004’005</td>
<td>9</td>
<td>649</td>
</tr>
<tr>
<td>400 200</td>
<td>20</td>
<td>7’204’005</td>
<td>9</td>
<td>1’073</td>
</tr>
</tbody>
</table>

with \( \nu_1(1), \nu_2(1) = (1, 1), \nu_1(2), \nu_2(2) = (1, 2), \nu_1(3), \nu_2(3) = (2, 1), \nu_1(4), \nu_2(4) = (1, 3), \nu_1(5), \nu_2(5) = (2, 2) \) \ldots and similarly in 3D by triple products of trigonometric functions. For each \( k \), the corresponding pressure field \( y_k(x) \) is measured at 64 fixed locations irregularly distributed throughout \( \Omega \); those measurements are then collected in the vector \( v(y_k) \).

In (4.2) each elliptic PDE is discretized on a regular grid using \( Q^0 \) finite elements for the model variable, \( u \), and \( Q^1 \) finite elements for the state variable \( y_k \). As a measure of variation in the piecewise constant log conductivity coefficient, we include the penalty term

\[
V(u) = h^{2d} \int_{\mathcal{F}_h} \|u\|^2 ds,
\]

where \( \mathcal{F}_h \) denotes all inter-element boundaries and \( \|u\| \) denotes jumps across interfaces.

First, we consider the 2D case and apply our RSP-IIP method to (4.2) with \( \alpha = 10^{-7} \) and \( \beta = 0 \). As initial guess, we always set \( u \) identically to zero and \( y_k \) to the corresponding forward solutions. Figure 4 shows the optimal solution for \( N_E = 10 \) PDE constraints on an \( N_h \times N_h \) mesh with \( N_h = 100 \). Indeed, the optimal model approximately follows the true model, while the measurements’ misfits act as point sources in the adjoint problem for the Lagrange multiplier \( \lambda \). In Table 4.4 we list the number of optimization steps and total run-time for varying mesh size \( N_h \times N_h \) and number of PDE constraints, \( N_E \), thereby resulting in KKT systems up to \( n = 7.2 \cdot 10^6 \).
Inexact Interior-Point Method for PDE-Constrained Nonlinear Optimization

Fig. 5. 3D log conductivity estimation: Optimal model $u^*$ computed by the RSP-IIP method applied to (4.2) with $N_E = 6$ PDE constraints on a $25 \times 25 \times 25$ grid (left) and corresponding iso-surfaces at $[-0.45, -0.2, 0.2, 0.45]$ (right).

Table 4.5

<table>
<thead>
<tr>
<th>$N_h$</th>
<th>$N_E$</th>
<th>$n$</th>
<th># IIP steps</th>
<th>run-time [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>4</td>
<td>110'305</td>
<td>11</td>
<td>59</td>
</tr>
<tr>
<td>20</td>
<td>8</td>
<td>174'305</td>
<td>11</td>
<td>80</td>
</tr>
<tr>
<td>20</td>
<td>16</td>
<td>302'305</td>
<td>11</td>
<td>129</td>
</tr>
<tr>
<td>20</td>
<td>32</td>
<td>558'305</td>
<td>9</td>
<td>202</td>
</tr>
<tr>
<td>40</td>
<td>4</td>
<td>856'605</td>
<td>15</td>
<td>1'452</td>
</tr>
<tr>
<td>40</td>
<td>8</td>
<td>1'368'605</td>
<td>12</td>
<td>1'345</td>
</tr>
<tr>
<td>40</td>
<td>16</td>
<td>2'392'605</td>
<td>12</td>
<td>1'913</td>
</tr>
<tr>
<td>40</td>
<td>32</td>
<td>4'440'605</td>
<td>12</td>
<td>3'102</td>
</tr>
<tr>
<td>60</td>
<td>4</td>
<td>2'862'905</td>
<td>15</td>
<td>10'180</td>
</tr>
<tr>
<td>60</td>
<td>8</td>
<td>4'590'905</td>
<td>14</td>
<td>11'491</td>
</tr>
<tr>
<td>60</td>
<td>16</td>
<td>8'046'905</td>
<td>14</td>
<td>14'331</td>
</tr>
<tr>
<td>60</td>
<td>32</td>
<td>14'958'905</td>
<td>13</td>
<td>18'595</td>
</tr>
</tbody>
</table>

in size. Remarkably, every problem was solved within merely 11 optimization steps, independently of $N_h$ and $N_E$, while the total run-time increases only linearly with $N_E$. Since the PDE in (4.2) is linear in the state variable $y$, as is often the case, the sub-matrices $J_{yk}$ in (3.2) are all identical; hence, the RSP-GMRES solver only needs to factorize a single PDE block-matrix, such as $J_y$.

Next, we consider the 3D case, and apply the RSP-IIP method to (4.2) with $\alpha = 10^{-7}$ and $\beta = 1$. Figure 5 shows slices of a typical optimal model $u^*$ for $N_E = 6$ PDE constraints on an $N_h \times N_h \times N_h$ mesh with $N_h = 25$, which also exhibits a maximum about $(0.25, 0.25, 0.25)$ and a minimum about $(0.75, 0.75, 0.75)$.

In Table 4.5 we list the number of optimization steps and total run-time for varying $N_E$ and mesh sizes $N_h \times N_h \times N_h$. Again, all problems were solved within merely 15 optimization steps, which barely increases with growing mesh size. Moreover, the RSP-IIP method efficiently takes advantage of the added information from
Fig. 6. 3D log conductivity estimation: average run-time per optimization step of the RSP-IIP method as a function of the number of PDE constraints, $N_E$, for an $N_h^3$ mesh with $N_h = 40, 60$.

Fig. 7. 3D log conductivity estimation: number of RSP-GMRES iterations ($\times$) and relative residual ($\circ$) at SMART acceptance for varying number of PDE constraints, $N_E$, and mesh sizes $N_h \times N_h \times N_h$. Note the different scales on the left and right axes.
increasingly many PDE constraints, as the number of optimization steps actually decreases for increasing $N_E$. In Figure 6, we again observe the linear average run-time complexity of the RSP-IIP method with respect to $N_E$ for two different $N_h \times N_h \times N_h$ meshes. Remarkably, the total run-time is even sub-linear due to the mild decrease in the number of optimization steps with increasing $N_E$. In Fig. 7 we also follow the evolution of the number of RSP-GMRES iterations (‘×’) and the relative residual of the accepted search direction (‘◦’) at each optimization step. The relative residual of the accepted iterate, never below $10^{-6}$, typically lies above $10^{-2}$. Hence, the SMART tests do not demand increasingly accurate solutions from the RSP-GMRES solver during the optimization process. Although the number of RSP-GMRES iterations typically increases in the course of any optimization run, all KKT systems were solved within 25 RSP-GMRES iterations, independently of $N_h$ or $N_E$.

### 4.4. Geophysical Imaging.

Full waveform seismic inversion leads to some of the most challenging nonlinear PDE-constrained optimization problems. Here we consider the Marmousi model [8, 31, 34], a standard benchmark in seismic imaging. It corresponds to a vertical slice through the Cuanza basin in Angola [19] delimited by the two dimensional domain $\Omega = (0.4 \text{ km}, 6.4 \text{ km}) \times (0 \text{ km}, 1.6 \text{ km})$.

Given measurements, $\hat{y}_k, k = 1, \ldots, N_E$ from $N_E = 11$ seismic events (“shots”), we shall attempt to reconstruct the true velocity profile $\hat{u}(x)$ in $\Omega$ – see Figure 8 (a). The measurements are obtained for each point source $\delta_k$ by recording the amplitude of the corresponding (complex-valued) wave field, $y_k$, at 367 observation points located at the surface and the two lateral boundaries (vertical wells); those values are then collected in the vector $v(y_k)$. Hence, we consider the PDE-constrained optimization problem:

$$\min_{y,u} F(y, u) = \frac{1}{2} \sum_{k=1}^{N_E} ||v(y_k) - \hat{y}_k||_2^2 + \frac{\alpha}{2} TV(u),$$

$$\text{s.t.} -\nabla \cdot (u^2 \nabla y_k) - \omega^2 y_k = \delta_k \quad \text{in} \ \Omega,$$

$$\frac{\partial y_k}{\partial n} = i \frac{\omega}{u} y_k \quad \text{on} \ \partial \Omega,$$

$$u_- \leq u \leq u_+.$$  \hspace{1cm} (4.3)

Here $TV(u) = \int_{\Omega} \sqrt{|\nabla u|^2 + \varepsilon} \, dx$ denotes regularization by total variation with $\varepsilon = 18.75$ and $\alpha = 10^{-4}$. Each wave field, $y_k$, satisfies the Helmholtz equation (forward problem) for a given velocity profile $u(x)$ and frequency $\omega > 0$. At the boundary of the computational domain, $\partial \Omega$, we impose a first-order Sommerfeld-like absorbing boundary condition. To generate the synthetic data $\hat{y}_k$, we first solve the forward problem on a fine mesh with mesh size $h = 8 \text{ m}$ using $Q^1$ elements for $u$ and $Q^2$ elements for $y_k$; then we add 1% white noise to the recorded values. The finite element discretization was implemented using \texttt{libMesh} [21] and \texttt{PETSc} [1].

Since the problem is nonconvex, any local optimization method may converge to a false solution. Prior knowledge, in practice often available as inequality constraints, reduces the search space and thereby prevents the algorithm from becoming trapped too easily in a (false) local minimum. Here, the upper and lower bounds $u_+, u_-$ are determined from $\hat{u}$ by local mean and maximum, or minimum filtering [30] respectively, followed by 10% further expansion of the feasible interval – see Figure 8 (c) and (d).
As the number of extrema increases with frequency, we further reduce the risk of ending up in a (false) local minimum through frequency stepping: starting at lower frequency \( \omega \), we progressively increase \( \omega \) while initializing each optimization run from the previous optimal model obtained at lower frequency. In all cases we set \( \varepsilon_{\text{tol}} = 10^{-4} \) in (2.4), as smaller values did not yield any further improvement in the reconstruction. Moreover, we discretize both, \( u \) and \( y_k \) with \( Q^1 \) finite elements on a coarser mesh, thus avoiding any potential “inverse crime”.

Starting at the lowest frequency, \( \omega = 20 \), we now apply the RSP-IIP method to (4.3) with mesh size \( h = 40 \text{ m} \) for \( u \) and \( h = 20 \text{ m} \) for \( y_k \). We initialize the algorithm with \( \mu = 10^{-3} \) and \( u = (u_+ + u_-)/2 \) – see Figure 8. After only 10 optimization steps, the RSP-IIP method converges to the optimal model \( u^* \), displayed in Figure 9; the imaginary part of one typical wave field, \( y_2 \), is also shown there. At such low frequency, the wave length is still large and hence unable to detect smaller features in the medium – see also Figure 10. The total run-time and problem size are summarized in Table 4.6. In Figure 11, we follow the number of RSP-GMRES iterations required to solve the \( n \times n \) system with \( n = 1.1 \cdot 10^6 \).

Next, we let \( \omega = 40 \) and initialize the RSP-IIP algorithm with the optimal point from the previous run at \( \omega = 20 \). Again, we observe in Figure 11 an increase in the number of RSP-GMRES iterations during the optimization process. Although the problem size and the number of optimization steps are identical, the overall larger number of RSP-GMRES iterations results in a slight increase in total run-time – see Table 4.6.

Finally, we let \( \omega = 60, \mu = 10^{-5} \), and choose a finer mesh width \( h = 20 \text{ m} \) for the model parameter, too. As a consequence, the KKT system barely increases, whereas the KKT Schur complement now increases by a factor four. Nonetheless, the RSP-IIP method converges in only 14 steps, while the number of RSP-GMRES iterations never exceeds 300 iterations. The reconstructed model, shown in Figure 9, now reproduces the sharp transitions in the velocity field from the true model, with even smallest details revealed in Figure 10.

For comparison we now also apply the exact IP method to (4.3) with \( \omega = 40 \). The linear systems are solved using the direct solver Pardiso with 4 OpenMP threads.
Table 4.6

Geophysical imaging: time frequency, ω, problem size, size of the KKT system, n, number of inexact interior-point iterations, and run-times.

<table>
<thead>
<tr>
<th>ω</th>
<th># States</th>
<th># Model Param.</th>
<th>n</th>
<th># IIP steps</th>
<th>run-time [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>5.36 · 10^5</td>
<td>6.19 · 10^7</td>
<td>1.10 · 10^6</td>
<td>10</td>
<td>825</td>
</tr>
<tr>
<td>40</td>
<td>5.36 · 10^5</td>
<td>6.19 · 10^3</td>
<td>1.10 · 10^6</td>
<td>10</td>
<td>1’001</td>
</tr>
<tr>
<td>60</td>
<td>5.36 · 10^5</td>
<td>2.44 · 10^4</td>
<td>1.19 · 10^6</td>
<td>14</td>
<td>1’734</td>
</tr>
</tbody>
</table>

Optimal reconstructed profile $u^*_ω$

![Image of reconstructed profiles for ω = 20, 40, and 60](image)

Im{y2}

![Image of imaginary part of y2 for ω = 20, 40, and 60](image)

Fig. 9. Geophysical imaging: initial model for ω = 20, and reconstructed optimal model for ω = 20, 40, and 60 (left); Imaginary part of the $y_2$ for ω = 20, 40, and 60 (right). Compare with true model in Figure 8.

Fig. 10. Geophysical imaging: vertical cross-section of the true wave speed $u^*$, its reconstruction $u^*$ and lower and upper bounds $u_-$ and $u_+$ at $x_1 = 4.0$ at increasingly higher frequencies $ω = 20, 40$ and 60.
The exact IP method converges in 9 optimization steps, instead of 10 for the inexact IP method, yet its total run-time of 14’993 s now results in a fifteenth fold increase, despite the multi-threaded hardware! This illustrates that the SMART tests efficiently control the inexactness, thus leading to very few additional optimization steps, while the RSP-GMRES solver takes advantage of the sparsity structure of the KKT systems.

5. Concluding Remarks. Starting from the inexact interior-point (IIP) framework from [10, 12], we have devised an effective reduced-space preconditioner (RSP) for the iterative solution of the KKT system needed at each Newton step. Together they yield a scalable and robust numerical method for the solution of large-scale non-convex PDE-constrained optimization problems with inequality constraints. Because it uses the full Hessian matrix, modifying it whenever needed, the RSP-IIP method is not only globally convergent, but also converges fast locally.

The RSP preconditioner is not tailored to any particular class of PDEs or constraints, but instead judiciously exploits the structure of the KKT system. It efficiently handles any number of PDE constraints, $N_E$, a key advantage for inverse problems where the influence of noisy data is often reduced by including data from multiple sources. Moreover, our numerical experiments show that the number of optimization steps remains independent of $N_E$. Similarly, both the number of inner RSP-GMRES iterations and of outer Newton iterations remains essentially independent of the mesh size $h$. As a consequence, we repeatedly observe (optimal) linear run-time complexity with respect to $N_E$ or to $h$, even in the presence of active inequality constraints.

The number of optimization steps needed by the RSP-IIP method is always comparable to that of the corresponding exact IP approach, where the KKT system is explicitly factorized. Yet by exploiting the sparsity structure of the KKT system (2.5), the RSP-IIP method achieves a significant speed-up over the exact IP method, even more so as the size of the KKT system increases. Since the two dominant blocks of the KKT system (2.5) are both block-diagonal, the PDEs can be solved independently of one another and in parallel. Hence if every forward solve is assigned to a separate processor, the total execution time becomes independent of $N_E$. 

![Graph](image_url) Fig. 11. Geophysical imaging: number of RSP-GMRES iterations ($\times$) and relative residual ($\circ$) at SMART acceptance for $\omega = 20, 40$ and 60. Note the different scales on the left and right axes.
Acknowledgement. The authors would like to thank Tyronne Rees for the PCG Matlab code used in Section 5.1, and Ahmed Sameh, Saied Faisal and Alicia Klinvex for useful comments and suggestions.

REFERENCES


<table>
<thead>
<tr>
<th>No.</th>
<th>Author</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>2013-01</td>
<td>H. Harbrecht, M. Peters</td>
<td><em>Comparison of Fast Boundary Element Methods on Parametric Surfaces</em></td>
</tr>
<tr>
<td>2013-02</td>
<td>V. Bosser, A. Surroca</td>
<td><em>Elliptic logarithms, diophantine approximation and the Birch and Swinnerton-Dyer conjecture</em></td>
</tr>
<tr>
<td>2013-03</td>
<td>A. Surroca Ortiz</td>
<td><em>Unpublished Talk: On some conjectures on the Mordell-Weil and the Tate-Shafarevich groups of an abelian variety</em></td>
</tr>
<tr>
<td>2013-04</td>
<td>V. Bosser, A. Surroca</td>
<td><em>Upper bound for the height of S-integral points on elliptic curves</em></td>
</tr>
<tr>
<td>2013-05</td>
<td>Jérémy Blanc, Jean-Philippe Furter, Pierre-Marie Poloni</td>
<td><em>Extension of Automorphisms of Rational Smooth Affine Curves</em></td>
</tr>
<tr>
<td>2013-06</td>
<td>Rupert L. Frank, Enno Lenzmann, Luis Silvestre</td>
<td><em>Uniqueness of Radial Solutions for the Fractional Laplacian</em></td>
</tr>
<tr>
<td>2013-07</td>
<td>Michael Griebel, Helmut Harbrecht</td>
<td><em>On the convergence of the combination technique</em></td>
</tr>
<tr>
<td>2013-08</td>
<td>Gianluca Crippa, Carlotta Donadello, Laura V. Spinolo</td>
<td><em>Initial-Boundary Value Problems for Continuity Equations with BV Coefficients</em></td>
</tr>
<tr>
<td>2013-09</td>
<td>Gianluca Crippa, Carlotta Donadello, Laura V. Spinolo</td>
<td><em>A Note on the Initial-Boundary Value Problem for Continuity Equations with Rough Coefficients</em></td>
</tr>
<tr>
<td>2013-10</td>
<td>Gianluca Crippa</td>
<td><em>Ordinary Differential Equations and Singular Integrals</em></td>
</tr>
<tr>
<td>2013-12</td>
<td>L. Graff, J. Fender, H. Harbrecht, M. Zimmermann</td>
<td><em>Key Parameters in High-Dimensional Systems with Uncertainty</em></td>
</tr>
<tr>
<td>2013-13</td>
<td>Jérémy Blanc, Immanuel Stampfli</td>
<td><em>Automorphisms of the Plane Preserving a Curve</em></td>
</tr>
<tr>
<td>No.</td>
<td>Author</td>
<td>Title</td>
</tr>
<tr>
<td>-------</td>
<td>-------------------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>2013-14</td>
<td>Jérémie Blanc, Jung Kyu Canci</td>
<td>Moduli Spaces of Quadratic Rational Maps with a Marked Periodic Point of Small Order</td>
</tr>
<tr>
<td>2013-15</td>
<td>Marcus J. Grote, Johannes Huber, Drosos Kourounis, Olaf Schenk</td>
<td>Inexact Interior-Point Method for Pde-Constrained Nonlinear Optimization</td>
</tr>
</tbody>
</table>