INDUSTRIAL AND SYSTEMS ENGINEERING

ISE

Complexity Analysis of Regularization Methods for Implicitly Constrained Least-Squares

AKWUM ONWUNTA[1] AND CLÉMENT W. ROYER[2]

[1]Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA, USA

[2]LAMSADE, CNRS, Université Paris Dauphine-PSL, Place du Maréchal de Lattre de Tassigny, 75016 Paris, France

ISE Technical Report 23T-022

LEHIGH UNIVERSITY

Complexity analysis of regularization methods for implicitly constrained least-squares

Akwum Onwunta* Clément W. Royer†

September 13, 2023

Abstract

Optimization problems constrained by partial differential equations (PDEs) naturally arise in scientific computing, as those constraints often model physical systems or the simulation thereof. In an implicitly constrained approach, the constraints are incorporated into the objective through a reduced formulation. To this end, a numerical procedure is typically applied to solve the constraint system, and efficient numerical routines with quantifiable cost have long been developed. Meanwhile, the field of complexity in optimization, that estimates the cost of an optimization algorithm, has received significant attention in the literature, with most of the focus being on unconstrained or explicitly constrained problems.

In this paper, we analyze an algorithmic framework based on quadratic regularization for implicitly constrained nonlinear least squares. By leveraging adjoint formulations, we can quantify the worst-case cost of our method to reach an approximate stationary point of the optimization problem. Our definition of such points exploits the least-squares structure of the objective, leading to an efficient implementation. Numerical experiments conducted on PDE-constrained optimization problems demonstrate the efficiency of the proposed framework.

1 Introduction

PDE-constrained optimization problems arise in various scientific and engineering fields where the objective is to find the optimal distribution of a given quantity while satisfying physical or mathematical laws described by PDEs, such as heat conduction or electromagnetic waves [1, 18, 13, 20]. Similar constrained formulations have also received recent interest from the machine learning community, as they opened new possibility for building neural network architectures [17]. A popular approach to handle PDE constraints is the so-called reduced formulation, in which the constraints are incorporated into the objective and become *implicit*. By properly accounting for the presence of these constraints while computing derivatives, it becomes possible to generalize unconstrained optimization techniques to the implicitly constrained setting [18].

*Department of Industrial and Systems Engineering, Lehigh University, 200 West Packer Avenue, Bethlehem, PA 18015-1582, USA (ako221@lehigh.edu).

†LAMSADE, CNRS, Université Paris Dauphine-PSL, Place du Maréchal de Lattre de Tassigny, 75016 Paris, France (clement.royer@lamsade.dauphine.fr). Funding for this author's research was partially provided by PGMO under the grant OCEAN and by Agence Nationale de la Recherche through program ANR-19-P3IA-0001 (PRAIRIE 3IA Institute).

Algorithms designed for implicitly-constrained optimization in scientific computing are typically based on careful problem representation, that allow for the use of linear algebra routines in very high dimensions. The cost of the associated operations in terms of floating-point number calculations or memory use is often at the core of an efficient implementation. Nevertheless, the analysis of these methods usually does not account for the cost of the optimization routines themselves, and rather provides asymptotic convergence results. Although these guarantees certify that a given method is capable of reaching a solution of the problem, they do not quantify how fast an algorithm can be at satisfying a desired stopping criterion, or its performance under a constrained computational budget.

Providing such guarantees is the idea behind *worst-case complexity analysis*, a technique that has gained significant traction in the optimization community over the past decade, especially in the nonconvex setting [6]. A complexity bound characterizes the worst-case performance of a given optimization scheme according to a performance metric (e.g., number of iterations, derivative evaluations, etc) and a stopping criterion (e.g., approximate optimality, predefined budget, etc). Recent progress in the area has switched from designing optimization techniques with complexity guarantees in mind to studying popular algorithmic frameworks through the prism of complexity, with several results focusing on the least-squares setting [3, 4, 5, 10]. Despite this connection to practical considerations, complexity guarantees have yet to be fully explored, especially in the context of implicitly-constrained optimization.

In this paper, we study an algorithmic framework for least-squares problems with implicit constraints. Our approach leverages the particular structure of the objective in order to compute derivatives, and encompasses popular algorithms for nonlinear least squares such as the Levenberg-Marquardt method [15]. Under standard assumptions for this class of methods, we establish complexity guarantees for our framework. In a departure from standard literature, our analysis is based on a recently proposed stationarity criterion for least-squares problems [5]. To the best of our knowledge, these results are the first of their kind for implicitly constrained problems. In addition, our complexity results improve over bounds recently obtained in the unconstrained setting [4], thereby advancing our understanding of complexity guarantees for least-squares problems. Numerical experiments on PDE-constrained problems illustrate the practical relevance of the proposed stationarity criterion, and show that our framework handles both small and large residual problems, as well as nonlinearity in the implicit constraints.

The rest of this paper is organized as follows. In Section 2, we present our formulation of interest, and discuss how its least-squares structure is used to design our algorithmic framework. We establish complexity guarantees for several instances of our proposed method in Section 3. In Section 4, we investigate the performance of our algorithm on classical benchmark problems from PDE-constrained optimization. We finally summarize our work in Section 5.

2 Least-squares optimization with implicit constraints

In this paper, we discuss algorithms for least-squares problems of the form

$$\min_{u \in \mathbb{R}^d} J(y,u) := \frac{1}{2}\|R(y,u)\|^2 \quad \text{subject to} \quad c(y,u) = 0, \tag{1}$$

that involves both the variable u as well as a vector of auxiliary variables y. We are interested in problems where it is possible to (numerically) solve the constraint equation $c(y,u) = 0$ to

obtain a unique solution y given u. Problem (1) can then be reformulated as

$$\min_{u \in \mathbb{R}^d} J(y(u), u) = \frac{1}{2}\|R(y(u), u)\|^2, \tag{2}$$

where the constraint arises implicitly in the formulation [12]. In PDE-constrained optimization, the constraint is a PDE, that can be solved given a value for the control vector u to yield a state vector $y(u)$. In that setting, problem (2) is often called the *reduced formulation* [18, Chapter 1].

In this paper, we depart from classical literature by focusing on the least-squares nature of the problem. To this end, we describe in Section 2.1 how derivatives can be computed by the adjoint approach for problem (2) while leveraging the problem structure. Our algorithm is then given in Section 2.2.

2.1 Computing adjoints for a least-squares problem

In this section, we derive an adjoint formula associated with the reduced formulation (2). To this end, we make the following assumption on our problem, which is a simplified version of a standard requirement in implicitly constrained problems [12].

Assumption 2.1 *For any $u \in \mathbb{R}^d$, the following properties hold.*

(i) There exists a unique vector $y(u)$ such that $c(y, u) = 0$.

(ii) The functions J and c are continuously differentiable.

(iii) The Jacobian of c with respect to its first argument, denoted by $c_y(\cdot, \cdot)$, is invertible at any (y, u) such that $c(y, u) = 0$.

We now describe our approach for computing the Jacobian $G_R(y(u), u)$ given u, based on the adjoint equation. In what follows, we let $\hat{J}(u) := J(y(u), u)$ and $\hat{R}(u) = R(y(u), u)$.

Algorithm 1 Computing the gradient via adjoint equations

1: Given u, solve $c(y, u) = 0$ for $y(u)$.
2: Solve the adjoint equation

$$c_y(y(u), u)^{\mathrm{T}} \lambda = -\nabla_y J(y(u), u)$$

for $\lambda(u)$.
3: Compute $\nabla \hat{J}(u) = \nabla_u J(y(u), u) + c_u(y(u), u)^{\mathrm{T}} \lambda(u)$.

Using the least-squares structure, we obtain the following expressions for the derivatives

$$\left\{ \begin{array}{rcl} \nabla_y J(y(u), u) & = & G_y(y(u), u)^{\mathrm{T}} R(y(u), u) \\ \nabla_u J(y(u), u) & = & G_u(y(u), u)^{\mathrm{T}} R(y(u), u), \end{array} \right. \tag{3}$$

where $G_y(y, u)$ and $G_u(y, u)$ are the Jacobian matrices with respect to y and u.

Using this formula within Algorithm 1 leads to

$$\begin{aligned}
\nabla \hat{J}(u) &= \nabla_u J(y(u),u) + c_u(y(u),u)^{\mathrm{T}} \lambda(u) \\
&= G_u(y(u),u)^{\mathrm{T}} R(y(u),u) + c_u(y(u),u)^{\mathrm{T}} \lambda(u) \\
&= G_u(y(u),u)^{\mathrm{T}} R(y(u),u) - c_u(y(u),u)^{\mathrm{T}} \left[c_y(y(u),u)^{\mathrm{T}}\right]^{\dagger} \nabla_y J(y(u),u) \\
&= G_u(y(u),u)^{\mathrm{T}} R(y(u),u) - c_u(y(u),u)^{\mathrm{T}} \left[c_y(y(u),u)^{\mathrm{T}}\right]^{\dagger} G_y(y(u),u)^{\mathrm{T}} R(y(u),u) \\
&= \left[G_u(y(u),u) - G_y(y(u),u) c_y(y(u),u)^{\dagger} c_u(y(u),u)\right]^{\mathrm{T}} R(y(u),u) \\
&= \left[G_u - G_y c_y(y(u),u)^{\dagger} c_u(y(u),u)\right]^{\mathrm{T}} R(y(u),u).
\end{aligned}$$

According to this expression, we can identify the Jacobian of $\hat{R}(u)$. Denoting this Jacobian by $\hat{G}(u)$, we have

$$\hat{G}(u) := G_u - G_y c_y(y(u),u)^{\dagger} c_u(y(u),u). \tag{4}$$

Using these formulas, we can adapt Algorithm 1 to account for our particular problem structure, leading to Algorithm 2.

Algorithm 2 Computing the Jacobian via adjoint equations

1: Given u, solve $c(y,u) = 0$ for $y(u)$.
2: Solve the equation

$$c_y(y(u),u)\zeta = -K_u$$

for $\zeta(u)$.
3: Compute $\hat{G}(u) = G_u + G_y \zeta(u)$.

2.2 Algorithmic framework

We propose a regularization framework that accounts for the implicitly constrained, least-squares nature of the problem. Algorithm 3 describes the framework, which builds on the Levenberg-Marquardt paradigm [15] and more generally on quadratic regularization techniques.

Note that the subproblem in (3) is in itself a (linear) least-squares problem when $H_k + \gamma_k \succeq 0$.

Algorithm 3 can be instantiated into several frameworks. When H_k is the zero matrix, then the method can be viewed as an instance of proximal gradient. When $H_k = G_k^{\mathrm{T}} G_k$, the method is a regularized Gauss-Newton iteration, similar to the Levenberg-Marquardt method. Other formulas, such as quasi-Newton updates, could also be used without the need for second-order information.

The kth iteration of Algorithm 3 will be called *successful* if $u_{k+1} \neq u_k$, and *unsuccessful* otherwise.

3 Complexity analysis

In this section, we investigate the theoretical properties of Algorithm 3. Our goal consists in reaching a vector u_k such that

$$\|R(y(u_k),u_k)\| \leq \epsilon_R \quad \text{or} \quad \frac{\|G_R(y(u_k),u_k)^{\mathrm{T}} R(y(u_k),u_k)\|}{\|R(y(u_k),u_k)\|} \leq \epsilon_g. \tag{6}$$

Algorithm 3 Regularization method for constrained least squares

Require: Initial iterate $u_0 \in \mathbb{R}^n$; initial parameter $\gamma_0 > 0$; minimum regularization parameter $0 < \gamma_{\min} \leq \gamma_0$; step acceptance threshold $\eta \in (0, 1)$.

Solve the constraint $c(y, u_0) = 0$ for y to obtain $y(u_0)$.
Evalue $R_0 = R(y(u_0), u_0)$.
for $k = 0, 1, 2, \ldots$ **do**
  Compute a step s_k as an approximate solution to the following problem

$$\min_{s \in \mathbb{R}^n} m_k(u_k + s) := \frac{1}{2}\|R_k\|^2 + g_k^{\mathrm{T}} s + \frac{1}{2} s^{\mathrm{T}}(H_k + \gamma_k I)s, \tag{5}$$

  where g_k is the gradient of $\hat{J}$ at u_k and $H_k \in \mathbb{R}^{n \times n}$ is a symmetric matrix.
  Solve the constraint $c(y, u_k + s_k) = 0$ for y to obtain $y(u_k + s_k)$.
  Compute the ratio of actual to predicted decrease in f defined as

$$\rho_k \leftarrow \frac{\hat{J}(u_k) - \hat{J}(u_k + s_k)}{m_k(u_k) - m_k(u_k + s_k)}.$$

  if $\rho_k \geq \eta$ **then**
    Set $u_{k+1} \leftarrow u_k + s_k$ and $\gamma_{k+1} \leftarrow \max\{0.5\gamma_k, \gamma_{\min}\}$.
    Solve the constraint $c(y, u_{k+1}) = 0$ for y to obtain $y(u_{k+1})$.
    Evalue $R_{k+1} = R(y(u_{k+1}), u_{k+1})$ and the Jacobian $G_{k+1} = G_R(y(u_{k+1}), u_{k+1})$.
  else
    Set $u_{k+1} \leftarrow u_k$ and $\gamma_{k+1} \leftarrow 2\gamma_k$.
  end if
end for

This scaled gradient condition was previously used for establishing complexity guarantees for algorithms applied to nonlinear least-squares problems [4, 5, 11].

Section 3.1 provides an iteration complexity bound for all instances of the algorithm.

3.1 Iteration complexity

We begin by a series of assumptions regarding the reduced formulation (2).

Assumption 3.1 *The function $\hat{J} : u \mapsto J(y(u), u)$ is continuously differentiable in u. Moreover, the gradient of $\hat{J}$ with respect to u is L-Lipschitz continuous for $L > 0$.*

Note that the first part of Assumption 3.1 holds when R is a continuously differentiable function.

Assumption 3.2 *There exists a positive constant $M_H > 0$ such that $\|H_k\| \leq M_H$ for all k.*

Assumption 3.2 is trivially satisfied when H_k is the zero matrix, or whenever the iterates are contained in a compact set. When H_k is the full Hessian matrix, $M_H = L$ is a valid choice.

Assumption 3.3 *For any iteration k, the matrix H_k is chosen as a positive semidefinite matrix and g_k is chosen as the exact gradient $g_k = G_k^{\mathrm{T}} R_k$.*

Note that both the zero matrix and the Gauss-Newton matrix $G_k^{\mathrm{T}} G_k$ are positive semidefinite, and thus satisfy Assumption 3.3.

Lemma 3.1 *Let Assumptions 3.1,3.2 and 3.3 hold. Suppose that the subproblem* (5) *is solved exactly at iteration k. Then,*

$$s_k = -(H_k + \gamma_k I)^{-1} G_k^{\mathrm{T}} R_k \tag{7}$$

where I is the identity matrix in $\mathbb{R}^{d\times d}$. Moreover,

$$m_k(u_k) - m_k(u_k + s_k) \geq \frac{1}{2} \frac{\|G_k^{\mathrm{T}} R_k\|^2}{M_H + \gamma_k}. \tag{8}$$

Proof. Under Assumption 3.3, the subproblem (5) is a strongly convex quadratic subproblem. It thus possesses a unique global minimum given by $-(H_k+\gamma_k I)^{-1} G_k^{\mathrm{T}} R_k$, which is precisely (7). Using this formula for s_k, we obtain

$$\begin{aligned} m_k(u_k) - m_k(u_k + s_k) &\geq -R_k^{\mathrm{T}} G_k s_k - \frac{1}{2} s_k^{\mathrm{T}} (H_k + \gamma_k I) s_k \\ &= R_k^{\mathrm{T}} G_k (H_k + \gamma_k I)^{-1} G_k^{\mathrm{T}} R_k - \frac{1}{2} R_k^{\mathrm{T}} G_k (H_k + \gamma_k I)^{-1} G_k^{\mathrm{T}} R_k \\ &= \frac{1}{2} R_k^{\mathrm{T}} G_k (H_k + \gamma_k I)^{-1} G_k^{\mathrm{T}} R_k \\ &\geq \frac{1}{2} \frac{\|G_k^{\mathrm{T}} R_k\|^2}{\|H_k + \gamma_k I\|}. \end{aligned}$$

By Assumption 3.2, we have

$$\|H_k + \gamma_k\| \leq \|H_k\| + \gamma_k \leq M_H + \gamma_k.$$

Hence, we have

$$m_k(u_k) - m_k(u_k + s_k) \geq \frac{1}{2} \frac{\|G_k^{\mathrm{T}} R_k\|^2}{M_H + \gamma_k},$$

as required. □

Our second ingredient for a complexity proof consists in bounding the value of the regularization parameter.

Lemma 3.2 *Let Assumptions 3.1, 3.3 and 3.2 hold. Then,*

(i) If k is the index of an unsuccessful iteration, then $\gamma_k < \frac{L}{2(1-\eta)}$.

(ii) For any iteration k,

$$\gamma_k \leq \gamma_{\max} := \max\left\{1, \gamma_0, \frac{L}{1-\eta}\right\}. \tag{9}$$

Proof. Suppose that the kth iteration is unsuccessful, i.e. that $\rho_k < \eta$. Then, one has

$$\eta(m_k(u_k + s_k) - m_k(u_k)) < \hat{J}(u_k + s_k) - \hat{J}(u_k). \tag{10}$$

Using Assumption 3.1, a Taylor expansion of $\hat{J}$ around u_k yields

$$\begin{aligned}\hat{J}(u_k+s_k)-\hat{J}(u_k) &\leq \nabla\hat{J}(u_k)^{\mathrm{T}}s_k+\frac{L}{2}\|s_k\|^2\\ &= g_k^{\mathrm{T}}s_k+\frac{L}{2}\|s_k\|^2\\ &= m_k(u_k+s_k)-m_k(u_k)-\frac{1}{2}s_k^{\mathrm{T}}(H_k+\gamma_k I)s_k+\frac{L}{2}\|s_k\|^2\\ &\leq m_k(u_k+s_k)-m_k(u_k)+\frac{L}{2}\|s_k\|^2,\end{aligned}$$

where the last inequality holds because of Assumption 3.3. Combining this inequality with (10), we obtain that

$$\begin{aligned}\eta(m_k(u_k+s_k)-m_k(u_k)) &< \hat{J}(u_k+s_k)-\hat{J}(u_k)\\ \Rightarrow \eta(m_k(u_k+s_k)-m_k(u_k)) &< m_k(u_k+s_k)-m_k(u_k)+\frac{L}{2}\|s_k\|^2\\ \Rightarrow (1-\eta)(m_k(u_k)-m_k(u_k+s_k)) &< \frac{L}{2}\|s_k\|^2.\end{aligned}$$

From Lemma 3.1, we obtain both an expression for s_k and a bound on the left-hand side. Noting that

$$\|s_k\| \leq \frac{\|G_k^{\mathrm{T}}R_k\|}{\|H_k+\gamma_k I\|} \leq \frac{\|G_k^{\mathrm{T}}R_k\|}{M_H+\gamma_k},$$

we obtain

$$\begin{aligned}(1-\eta)(m_k(u_k)-m_k(u_k+s_k)) &< \frac{L}{2}\|s_k\|^2\\ \Leftarrow \frac{(1-\eta)}{2}\frac{\|G_k^{\mathrm{T}}R_k\|^2}{M_H+\gamma_k} &< \frac{L}{2}\frac{\|G_k^{\mathrm{T}}R_k\|^2}{(M_H+\gamma_k)^2}\\ \Leftrightarrow \frac{1-\eta}{2(M_H+\gamma_k)} &< \frac{L}{2(M_H+\gamma_k)^2}\\ \Leftrightarrow \gamma_k &< \frac{L}{(1-\eta)}-M_H < \frac{L}{(1-\eta)}.\end{aligned}$$

Overall, we have shown that if the kth iteration is unsuccessful, then necessarily $\gamma_k < \frac{L}{(1-\eta)}$. By a contraposition argument, we then obtain that $\gamma_k \geq \frac{L}{(1-\eta)}$ implies that the iteration is successful and that $\gamma_{k+1} \leq \gamma_k$. Combining this observation with the initial value of γ_0 and the update mechanism for γ_k, we find that γ_k can never exceed $\max\{\gamma_0, \frac{L}{1-\eta}\} \leq \gamma_{\max}$, proving the desired result. □

Note that we define $\gamma_{\max}$ to be greater than or equal to 1 to simplify our bounds later on.

We now provide our first iteration complexity bound, that focuses on successful iterations.

Lemma 3.3 *Let Assumptions 3.1, 3.3 and 3.2 hold. Let $\epsilon_g \in (0,1)$, and let $\mathcal{S}_{\epsilon_g,\epsilon_R}$ denote the set of successful iterations for which u_k does not satisfy* (6)*. Then,*

$$\left|S_{\epsilon_g,\epsilon_R}\right| \leq \left\lceil \mathcal{C}_{\mathcal{S}} \log(2\hat{J}(u_0)\epsilon_R^{-2})\epsilon_R^{-2} \right\rceil + 1, \tag{11}$$

where $\mathcal{C}_{\mathcal{S}} = \frac{2(M_H+\gamma_{\max})}{\eta}$.

Proof. Let $k \in \mathcal{S}_{\epsilon_g,\epsilon_R}$. By definition, the corresponding iterate u_k satisfies

$$\|R_k\| \geq \epsilon_R \quad \text{and} \quad \frac{\|G_k^{\mathrm{T}} R_k\|}{\|R_k\|} \geq \epsilon_g. \tag{12}$$

Moreover, since k corresponds to a successful iteration, we have $\rho_k \geq \eta$, i.e.

$$\hat{J}(u_k) - \hat{J}(u_k + s_k) \geq \eta \left(m_k(u_k) - m_k(u_k + s_k)\right) \geq \eta \frac{\|G_k^{\mathrm{T}} R_k\|^2}{2(M_H + \gamma_k)} \geq \eta \frac{\|G_k^{\mathrm{T}} R_k\|^2}{2(M_H + \gamma_{\max})},$$

where we used the results of Lemmas 3.1 and 3.2 to bound the model decrease and γ_k, respectively. Combining the last inequality with (12) leads to

$$\begin{aligned}
\hat{J}(u_k) - \hat{J}(u_k + s_k) &\geq \frac{\eta}{2(M_H + \gamma_{\max})} \|G_k^{\mathrm{T}} R_k\|^2 \\
&= \frac{\eta}{2(M_H + \gamma_{\max})} \frac{\|G_k^{\mathrm{T}} R_k\|^2}{\|R_k\|^2} \|R_k\|^2 \\
&\geq \frac{\eta}{2(M_H + \gamma_{\max})} \epsilon_g^2 \|R_k\|^2 \\
&= \frac{\eta}{M_H + \gamma_{\max}} \epsilon_g^2 \hat{J}(u_k),
\end{aligned}$$

where the last line follows by definition of $\hat{J}(u_k)$. Since $\frac{\eta}{M_H+\gamma_{\max}}\epsilon_g^2 \in (0,1)$ by definition of all quantities involved, we obtain that

$$\left(1 - \frac{\eta}{M_H + \gamma_{\max}} \epsilon_g^2\right) \hat{J}(u_k) \geq \hat{J}(u_{k+1}). \tag{13}$$

Let now $\mathcal{S}^k_{\epsilon_g,\epsilon_R} := \{\ell < k | \ell \in \mathcal{S}_{\epsilon_g,\epsilon_R}\}$. Recalling that the iterate only changes on successful iterations and that the function $\hat{J}$ is bounded below by 0, we obtain that

$$\begin{aligned}
\left(1 - \frac{\eta}{M_H + \gamma_{\max}} \epsilon_g^2\right)^{\left|\mathcal{S}^k_{\epsilon_g,\epsilon_R}\right|} \hat{J}(u_0) &\geq \hat{J}(u_k) \\
\left(1 - \frac{\eta}{M_H + \gamma_{\max}} \epsilon_g^2\right)^{\left|\mathcal{S}^k_{\epsilon_g,\epsilon_R}\right|} \hat{J}(u_0) &\geq \frac{1}{2}\epsilon_R^2,
\end{aligned}$$

where the last line uses $k \in \mathcal{S}_{\epsilon_g,\epsilon_R}$. Taking logarithms and re-arranging, we arrive at

$$\begin{aligned}
\left|\mathcal{S}^k_{\epsilon_g,\epsilon_R}\right| \ln\left(1 - \frac{\eta}{M_H + \gamma_{\max}} \epsilon_g^2\right) &\geq 2 \ln\left(\epsilon_R^2 / (2\hat{J}(u_0))\right) \\
\left|\mathcal{S}^k_{\epsilon_g,\epsilon_R}\right| &\leq \frac{2 \ln\left(\epsilon_R^2/(2\hat{J}(u_0)\right)}{\ln\left(1 - \frac{\eta}{M_H+\gamma_{\max}}\epsilon_g^2\right)} \\
&\leq 2 \ln\left(2\hat{J}(u_0)\epsilon_R^{-2}\right) \frac{M_H + \gamma_{\max}}{\eta} \epsilon_g^{-2},
\end{aligned}$$

where the last inequality comes from $-\ln(1-t) \geq t$ for any $t \in (0,1)$. As a result, we obtain that

$$\left|\mathcal{S}_{\epsilon_g,\epsilon_R}\right| \leq 1 + 2 \ln\left(2\hat{J}(u_0)\epsilon_R^{-2}\right) \frac{M_H + \gamma_{\max}}{\eta} \epsilon_g^{-2},$$

where the additional 1 accounts for the largest iteration in $\mathcal{S}_{\epsilon_g,\epsilon_R}$. □

Lemma 3.4 *Under the assumptions of Lemma 3.3, let* $\mathcal{U}_{\epsilon_g,\epsilon_R}$ *be the set of unsuccessful iterations for which* (6) *does not hold. Then,*

$$\left|\mathcal{U}_{\epsilon_g,\epsilon_R}\right| \leq \lceil 1 + \log_2(\gamma_{\max}) \rceil \left|\mathcal{S}_{\epsilon_g,\epsilon_R}\right|. \tag{14}$$

Proof. The proof tracks that of [7, Lemma 2.5] for the trust-region case. Between two successful iterations, the value of γ_k only increases by factors of 2. Combining this observation with the fact that $\gamma_k \leq \gamma_{\max}$ per Lemma 3.2 and accounting for the first successful iteration leads to the final result. □

Combining Lemmas 3.3 and 3.4 finally yields our main complexity result.

Theorem 3.1 *Under Assumptions 3.1, 3.3 and 3.2, the number of successful iterations (and Jacobian evaluations) before reaching an iterate satisfying* (6) *satisfies*

$$|\mathcal{S}_{\epsilon_g,\epsilon_R}| = \mathcal{O}\left(\log(\epsilon_R^{-1})\epsilon_g^{-2}\right) \tag{15}$$

and the total number of iterations (and residual evaluations) before reaching such an iterate satisfies

$$|\mathcal{S}_{\epsilon_g,\epsilon_R}| + |\mathcal{U}_{\epsilon_g,\epsilon_R}| = \mathcal{O}\left(\log(\epsilon_R^{-1})\epsilon_g^{-2}\right). \tag{16}$$

The result of Theorem 3.1 improves over that obtained by Bergou et al [4] in a more general setting, and is consistent with that in Gould et al [11], where a series of results with vanishing dependencies in ϵ_R were established. Our result retains a logarithmic dependency but does not depend on increasingly larger constants.

To end this section, we provide a result tailored to our implicit constrained setup, and the operations that are performed throughout the course of the algorithm.

Corollary 3.1 *Under the assumptions of Theorem 3.1, the number of solves of the implicit constraint for* y *is*

$$1 + |\mathcal{S}_{\epsilon_g,\epsilon_R}| + |\mathcal{U}_{\epsilon_g,\epsilon_R}| = \mathcal{O}\left(\log(\epsilon_R^{-1})\epsilon_g^{-2}\right), \tag{17}$$

while the number of adjoint solves (using Algorithm 2) is

$$1 + |\mathcal{S}_{\epsilon_g,\epsilon_R}| = \mathcal{O}\left(\log(\epsilon_R^{-1})\epsilon_g^{-2}\right). \tag{18}$$

3.2 Inexact variants

We now consider solving the subproblem (7) in an inexact fashion. Such a procedure is classical in large-scale optimization, and would apply in the case of a nonzero H_k.

Assumption 3.4 *For any iteration* k*, the step* s_k *is chosen so as to satisfy*

$$(H_k + \gamma_k I)s_k = -g_k + t_k, \quad \|t_k\| \leq \theta\sqrt{\frac{\gamma_k}{\|H_k\| + \gamma_k}}\|g_k\| \tag{19}$$

for $\theta \in [0, 1)$.

Assuming that the linear system is solved to the accuracy expressed in condition (19), one can establish the following result.

Lemma 3.5 *Let Assumptions 3.3, 3.2 and 3.4 hold. For any iteration k, the step s_k satisfies*

$$\|s_k\| \leq \frac{(1+\theta)\|G_k^{\mathrm{T}} R_k\|}{\gamma_k} \tag{20}$$

and

$$m_k(u_k) - m_k(u_k + s_k) \geq \frac{1-\theta^2}{2} \frac{\|G_k^{\mathrm{T}} R_k\|^2}{M_H + \gamma_k}. \tag{21}$$

Proof. Using Assumption 3.4 gives

$$\|t_k\| \leq \theta \sqrt{\frac{\gamma_k}{\|H_k\| + \gamma_k}} \|G_k^{\mathrm{T}} R_k\| \leq \theta \|G_k^{\mathrm{T}} R_k\|.$$

Since $s_k = (H_k + \gamma_k I)^{-1}(-g_k + t_k)$ by construction, we obtain

$$\begin{aligned} \|s_k\| = \left\|(H_k + \gamma_k I)^{-1}(-g_k + t_k)\right\| &\leq \frac{\|g_k\| + \|t_k\|}{\|H_k + \gamma_k I\|} \\ &\leq (1+\theta) \frac{\|g_k\|}{\|H_k\| + \gamma_k} \\ &\leq \frac{(1+\theta)\|G_k^{\mathrm{T}} R_k\|}{\gamma_k}, \end{aligned}$$

proving (20).

We now use this ineqality together with the definition of s_k to bound the model decrease:

$$\begin{aligned} m_k(u_k) - m_k(u_k + s_k) &= -g_k^{\mathrm{T}} s_k - \frac{1}{2} s_k^{\mathrm{T}} (H_k + \gamma_k I) s_k \\ &= -g_k^{\mathrm{T}} (H_k + \gamma_k I)^{-1}(-g_k + t_k) - \frac{1}{2}(-g_k + t_k)^{\mathrm{T}} (H_k + \gamma_k I)^{-1}(-g_k + t_k) \\ &= \frac{1}{2} g_k^{\mathrm{T}} (H_k + \gamma_k I)^{-1} g_k - \frac{1}{2} t_k^{\mathrm{T}} (H_k + \gamma_k I)^{-1} t_k. \end{aligned}$$

Using Cauchy-Schwarz inequality, we obtain on one hand

$$g_k^{\mathrm{T}} (H_k + \gamma_k I)^{-1} g_k \geq \frac{\|g_k\|^2}{\|H_k + \gamma_k I\|} \geq \frac{\|g_k\|^2}{\|H_k\| + \gamma_k},$$

while on the other hand

$$t_k^{\mathrm{T}} (H_k + \gamma_k I)^{-1} t_k \leq \frac{\|t_k\|^2}{\|H_k + \gamma_k I\|} \leq \frac{\|t_k\|^2}{\gamma_k} \leq \frac{\theta^2 \|g_k\|^2}{\|H_k\| + \gamma_k},$$

where the last inequality comes from Assumption 3.4. As a result, we arrive at

$$\begin{aligned} m_k(u_k) - m_k(u_k + s_k) &\geq \frac{1}{2} \frac{\|g_k\|^2}{\|H_k\| + \gamma_k} - \frac{\theta^2}{2} \frac{\|g_k\|^2}{\|H_k\| + \gamma_k} \\ &= \frac{1-\theta^2}{2} \frac{\|g_k\|^2}{\|H_k\| + \gamma_k} \\ &\geq \frac{1-\theta^2}{2} \frac{\|g_k\|^2}{M_H + \gamma_k}, \end{aligned}$$

using Assumption 3.2 to bound $\|H_k\|$. This proves (21). □

Similarly to the exact case, we now prove that the regularization parameter is bounded from above.

Lemma 3.6 *Let Assumptions 3.1, 3.2, 3.3 and 3.4 hold. Then,*

(i) If k is the index of an unsuccessful iteration, then $\gamma_k < \frac{L(1+\theta)^2}{(1-\eta)(1-\theta^2)}$.

(ii) For any iteration k,

$$\gamma_k \leq \gamma_{\max}^{in} := \max\left\{1, \gamma_0, \frac{L(1+\theta)^2}{(1-\eta)(1-\theta^2)}\right\}. \tag{22}$$

Proof. By the same reasoning as in the proof of Lemma 3.2, we know that for any unsuccessful iteration, we have

$$(1-\eta)(m_k(u_k) - m_k(u_k+s_k)) < \frac{L}{2}\|s_k\|^2. \tag{23}$$

Using now the properties (20) and (21) in (23), we obtain:

$$\begin{aligned}
(1-\eta)(m_k(u_k) - m_k(u_k+s_k)) &< \frac{L}{2}\|s_k\|^2 \\
\Leftarrow \frac{(1-\eta)(1-\theta^2)}{2}\frac{\|G_k^{\mathrm{T}}R_k\|^2}{M_H+\gamma_k} &< \frac{L(1+\theta)^2}{2}\frac{\|G_k^{\mathrm{T}}R_k\|^2}{(M_H+\gamma_k)^2} \\
\Leftrightarrow M_H+\gamma_k &< \frac{L(1+\theta)^2}{(1-\eta)(1-\theta^2)} \\
\Leftrightarrow \gamma_k &< \frac{L(1+\theta)^2}{(1-\eta)(1-\theta^2)} - M_H < \frac{L(1+\theta)^2}{(1-\eta)(1-\theta^2)}.
\end{aligned}$$

Overall, we have shown that if the kth iteration is unsuccessful, then necessarily $\gamma_k < \frac{L(1+\theta)^2}{(1-\eta)(1-\theta^2)}$. Because of the updating rules on γ_k and accounting for γ_0 we obtain that

$$\gamma_k \leq \max\left\{\gamma_0, \frac{L(1+\theta)^2}{(1-\eta)(1-\theta^2)}\right\} \leq \gamma_{\max}^{in}$$

for all k, proving the desired result. □

We can now state an iteration complexity result for the inexact variant.

Lemma 3.7 *Let Assumptions 3.1, 3.2, 3.3 and 3.4 hold. Let $\epsilon_g \in (0,1)$, and let $\mathcal{S}_{\epsilon_g,\epsilon_R}^{in}$ denote the set of successful iterations for which u_k does not satisfy* (6)*. Then,*

$$\left|\mathcal{S}_{\epsilon_g,\epsilon_R}^{in}\right| \quad \leq \quad \left\lceil \mathcal{C}_{\mathcal{S}}^{in} \log(2\hat{J}(u_0)\,\epsilon_R^{-2})\epsilon_g^{-2}\right\rceil + 1, \tag{24}$$

where $\mathcal{C}_{\mathcal{S}}^{in} = \frac{2(M_H+\gamma_{\max}^{in})}{\eta(1-\theta^2)}$.

Proof. Let $k \in \mathcal{S}_{\epsilon_g,\epsilon_R}$. By definition, the kth iteration is successful, and we have per Lemma 3.6

$$\hat{J}(u_k) - \hat{J}(u_k+s_k) \geq \eta\,(m_k(u_k) - m_k(u_k+s_k)) \geq \frac{\eta(1-\theta^2)}{2}\frac{\|G_k^{\mathrm{T}}R_k\|^2}{M_H+\gamma_k} \geq \frac{\eta(1-\theta^2)}{2}\frac{\|G_k^{\mathrm{T}}R_k\|^2}{M_H+\gamma_{\max}^{in}},$$

where the last inequality is a consequence of Lemma 3.6. In addition, the corresponding iterate u_k satisfies (12), leading to

$$\begin{aligned}\hat{J}(u_k) - \hat{J}(u_k + s_k) &\geq \frac{\eta(1-\theta^2)}{2} \frac{\|G_k^{\mathrm{T}} R_k\|^2}{M_H + \gamma_{\max}^{in}} \\ &= \frac{\eta(1-\theta^2)}{2(M_H + \gamma_{\max}^{in})} \frac{\|G_k^{\mathrm{T}} R_k\|^2}{\|R_k\|^2} \|R_k\|^2 \\ &= \frac{\eta(1-\theta^2)}{M_H + \gamma_{\max}^{in}} \frac{\|G_k^{\mathrm{T}} R_k\|^2}{\|R_k\|^2} J(u_k) \\ &\geq \frac{\eta(1-\theta^2)}{M_H + \gamma_{\max}^{in}} \epsilon_g^2 \hat{J}(u_k).\end{aligned}$$

Using that $\frac{\eta(1-\theta^2)}{M_H+\gamma_{\max}^{in}} < 1$ then leads to

$$\left(1 - \frac{\eta(1-\theta^2)}{M_H + \gamma_{\max}} \epsilon_g^2\right) \hat{J}(u_k) \geq \hat{J}(u_{k+1}). \tag{25}$$

By proceeding as in the proof of Lemma 3.3 and using (25) in lieu of (25), one establishes that

$$\left|\mathcal{S}_{\epsilon_g,\epsilon_R}^{in}\right| \leq 1 + 2\ln\left(2\hat{J}(u_0)\epsilon_R^{-2}\right) \frac{M_H + \gamma_{\max}}{\eta(1-\theta^2)} \epsilon_g^{-2},$$

proving the desired result. □

To connect the number of unsuccessful iterations with that of successful iterations, we use the same argument as in the exact case by replacing the bound (9) with (22).

Lemma 3.8 *Under the assumptions of Lemma 3.7, let $\mathcal{U}_{\epsilon_g,\epsilon_R}^{in}$ be the set of unsuccessful iterations for which* (6) *does not hold. Then,*

$$\left|\mathcal{U}_{\epsilon_g,\epsilon_R}^{in}\right| \leq \left\lceil 1 + \log_2\left(\gamma_{\max}^{in}\right)\right\rceil \left|\mathcal{S}_{\epsilon_g,\epsilon_R}^{in}\right|. \tag{26}$$

Our next theorem gives the total iteration complexity result by combining Lemmas 3.7 and 3.8.

Theorem 3.2 *Under Assumptions 3.1, 3.2, 3.3 and 3.4, the number of successful iterations (and inexact step calculations) before reaching an iterate satisfying* (6) *satisfies*

$$|\mathcal{S}_{\epsilon_g,\epsilon_R}^{in}| = \mathcal{O}\left(\frac{1}{(1-\theta^2)^2}\log(\epsilon_R^{-1})\epsilon_g^{-2}\right) \tag{27}$$

and the total number of iterations (and residual evaluations) before reaching such an iterate satisfies

$$|\mathcal{S}_{\epsilon_g,\epsilon_R}^{in}| + |\mathcal{U}_{\epsilon_g,\epsilon_R}^{in}| = \mathcal{O}\left(\frac{1}{(1-\theta^2)^2}\log(\epsilon_R^{-1})\epsilon_g^{-2}\right). \tag{28}$$

The results of Theorem 3.2 match that of Theorem 3.1 in terms of dependencies on ϵ_g and ϵ_R. To emphasize the use of inexact steps, we highlighted the dependency with respect to the inexact tolerance θ. As expected, one notes that this dependency vanishes when $\theta = 0$ (i.e. when we consider exact steps as in Section 3.1), and that the complexity bounds worsen as θ gets closer to 1. A similar observation holds for the results in the next corollary, that is a counterpart to Corollary 3.1.

Corollary 3.2 *Under the assumptions of Theorem 3.2, the number of solves for y is*

$$1 + |\mathcal{S}^{in}_{\epsilon_g,\epsilon_R}| + |\mathcal{U}^{in}_{\epsilon_g,\epsilon_R}| = \mathcal{O}\left(\log\left(\frac{1}{1-\theta^2}\right)\frac{1}{(1-\theta^2)^2}\,\log(\epsilon_R^{-1})\epsilon_g^{-2}\right), \tag{29}$$

while the number of adjoint solves (using Algorithm 2) is

$$1 + |\mathcal{S}^{in}_{\epsilon_g,\epsilon_R}| = \mathcal{O}\left(\frac{1}{(1-\theta^2)^2}\,\log(\epsilon_R^{-1})\epsilon_g^{-2}\right). \tag{30}$$

In the case of inexact steps, we can however provide more precise guarantees on the number of operations necessary to compute steps at every iteration. More precisely, suppose that we apply an iterative solver to the system $(H_k + \gamma_k I)s = -g_k$ in order to find an approximate solution satisfying Assumption 3.4. In particular, one can resort to iterative linear algebra techniques such as Conjugate Gradient (CG), and obtain guarantees on the number of matrix-vector products necessary to reach the desired accuracy [15]. A result tailored to our setting is presented below.

Proposition 3.1 *Let Assumption 3.3 hold. Suppose that we apply conjugate gradient (CG) to the linear system $(H_k + \gamma_k I)s = -g_k$, where g_k, H_k, γ_k are obtained from the kth iteration of Algorithm 3. Then, the conjugate gradient method computes an iterate satisfying* (19) *after at most*

$$\min\left\{n, \frac{1}{2}\sqrt{\kappa_k}\log\left(\frac{2\kappa_k}{\theta}\right)\right\} \tag{31}$$

iterations or, equivalently, matrix-vector products, where $\kappa_k = \frac{\|H_k\|+\gamma_k}{\gamma_k}$.

Proof. Let $s^{(q)}$ be the iterate obtained after applying q iterations of conjugate gradient to $(H_k + \gamma_k I)s = -g_k$. If $q = n$, then necessarily the linear system has been solved exactly and (19) is trivially satisfied. Thus we assume in what follows that $q < n$.

Standard CG theory gives [16, Proof of Lemma 11]:

$$\|(H_k + \gamma_k I)s^{(q)} + g_k\| \leq 2\sqrt{c_k}\left(\frac{\sqrt{c_k}-1}{\sqrt{c_k}+1}\right)^q \|g_k\|, \tag{32}$$

where c_k is the condition number of $H_k + \gamma_k I$. Noticing that $c_k \leq \kappa_k$, we see that (32) implies

$$\|(H_k + \gamma_k I)s^{(q)} + g_k\| \leq 2\sqrt{\kappa_k}\left(\frac{\sqrt{\kappa_k}-1}{\sqrt{\kappa_k}+1}\right)^q \|g_k\|. \tag{33}$$

Suppose now that $s^{(q)}$ does not satisfy (19). Then,

$$\|(H_k + \gamma_k I)s^{(q)} + g_k\| \geq \theta\sqrt{\frac{\gamma_k}{\|H_k\|+\gamma_k}}\|g_k\| = \frac{\theta}{\sqrt{\kappa_k}}\|g_k\|. \tag{34}$$

Combining (33) and (34) yields

$$\begin{aligned}\frac{\theta}{\sqrt{\kappa_k}}\|g_k\| &\leq 2\sqrt{\kappa_k}\left(\frac{\sqrt{\kappa_k}-1}{\sqrt{\kappa_k}+1}\right)^q \|g_k\| \\ \frac{\theta}{2\kappa} &\leq \left(\frac{\sqrt{\kappa_k}-1}{\sqrt{\kappa_k}+1}\right)^q.\end{aligned}$$

Taking logarithms and rearranging, we arrive at

$$q \leq \frac{\ln(\theta/(2\kappa_k))}{\ln\left(\frac{\sqrt{\kappa_k}-1}{\sqrt{\kappa_k}+1}\right)} \leq \frac{\ln(2\kappa_k/\theta)}{\ln\left(1+\frac{2}{\sqrt{\kappa_k}-1}\right)} \leq \frac{1}{2}\sqrt{\kappa_k}\ln\left(\frac{2\kappa_k}{\theta}\right), \tag{35}$$

where the last inequality used $\ln(1+\frac{1}{t}) \geq \frac{1}{t+1/2}$. Combining (35) with the fact that $q \leq n$ yields our desired bound. □

Using the bounds on γ_k and $\|H_k\|$ from our complexity analysis, we see that the value (31) can be bounded from above by

$$\min\left\{n, \frac{1}{2}\sqrt{\kappa}\log\left(\frac{2\kappa}{\theta}\right)\right\}, \tag{36}$$

with $\kappa = \frac{M_H+\gamma_{\max}^{in}}{\gamma_{\min}}$. Combining this result with the complexity bound of Theorem 3.2, we are able to bound the number of matrix-vector products as follows.

Corollary 3.3 *Under the assumptions of Theorem 3.2, suppose that we apply conjugate gradient to compute inexact steps in Algorithm 3. Then, the algorithm reaches a point satisfying* (6) *in at most*

$$\begin{aligned} & \min\left\{n, \tfrac{1}{2}\sqrt{\kappa}\log\left(\tfrac{2\kappa}{\theta}\right)\right\} \times \left(1+|\mathcal{S}^{in}_{\epsilon_g,\epsilon_R}|\right) \\ = \;& \mathcal{O}\left(\min\left\{n, \sqrt{\kappa}\log\left(\tfrac{\kappa}{\theta}\right)\right\}\log\left(\tfrac{1}{1-\theta^2}\right)\tfrac{1}{(1-\theta^2)^2}\log(\epsilon_R^{-1})\epsilon_g^{-2}\right) \end{aligned} \tag{37}$$

matrix-vector products.

As a final note, we point out that there exist variants of the conjugate gradient method that take advantage of a Gauss-Newton approximation $H_k = G_k^{\mathrm{T}}G_k$. For such variants, each iteration would require two Jacobian-vector products, resulting in an additional factor of 2 in the above complexity bound.

4 Numerical illustration

In this section, we illustrate the performance of several instances of our framework on classical PDE-constrained optimization problems. Our goal is primarily to investigate the significant effect of using condition (6) as a stopping criterion. In addition, we wish to study the performance of the Gauss-Newton and inexact Gauss-Newton formulas. For this reason, we are primarily interested in the evaluation and iteration cost of our method, and therefore we report these statistics in the rest of the section.

All algorithms were implemented in MATLAB R2023a. We run three variants of the methods corresponding to using gradient steps, Gauss-Newton steps and inexact Gauss Newton based on conjugate gradient (thereafter denoted by Gauss-Newton+CG). All variants used $\eta = 0.1$, $\gamma_{\min} = 10^{-10}$ and $\gamma_0 = \max\{1, \|g_0\|, \|u_0\|_\infty + 1\}$. The inexact condition (31) was replaced by $\|t_k\| \leq \theta\|g_k\|$ with $\theta = 0.1$. Runs were completed on HP EliteBook x360 1040 G8 Notebook PC with 32Go RAM and 8 cores 11th Gen Intel Core i7-1165G7 @ 2.80GHz.

4.1 Elliptic PDE-constrained problem

We first consider a standard elliptic optimal control problem, where the control is chosen so that the temperature distribution (the state) matches a desired distribution as closely as possible [9, 18]. The resulting problem can be written as

$$\min_{y,u} J(y,u) := \frac{1}{2}\int_{\mathcal{D}} \left[(y(u(x)) - z(x))^2 + \lambda u(x)^2 \right] dx, \tag{38}$$

$$\begin{aligned} \text{subject to} \qquad -\nabla \cdot (a(x)\nabla y(x)) &= u(x), \quad \text{in } \mathcal{D}, \\ y(x) &= 0, \qquad \text{on } \partial\mathcal{D}, \end{aligned} \tag{39}$$

where z is the desired state and $\lambda > 0$ is a regularization parameter. We set $\mathcal{D} := [0,1]^2$ and $a(x) \equiv 1,\ \forall x \in \mathcal{D}$. We discretize (38) and (39) using piecewise linear finite elements on a triangular grid yields

$$\frac{1}{2}(\mathbf{y}-\mathbf{z})^T M(\mathbf{y}-\mathbf{z}) + \frac{\lambda}{2}\mathbf{u}^T M\mathbf{u},$$

and

$$K\mathbf{y} = M\mathbf{u},$$

or, equivalently,

$$\mathbf{y} = K^{-1}M\mathbf{u},$$

where the vectors $\mathbf{y}, \mathbf{z}, \mathbf{u}$ denote the discrete forms of the state, the desired state, respectively, and the control variables. Moreover, K and M represent the stiffness and mass matrices, respectively [9]. Note that the boundary constraints are incorporated into the stiffness matrix.

Note that the cost function (4.1) can be written as $\frac{1}{2}\|R(\mathbf{y},\mathbf{u})\|^2$ with

$$R(\mathbf{y},\mathbf{u}) = \begin{bmatrix} M^{1/2}(\mathbf{y}-\mathbf{z}) \\ \sqrt{\beta}M^{1/2}\mathbf{u} \end{bmatrix}, \tag{40}$$

fitting our formulation of interest (1).

Tables 1 and 2 represent our results with control dimension $n = 1829$, $\lambda = 0.001$, and using the vector of all ones as a starting control value $\mathbf{u}_0$, (that is, $\mathbf{u}_0 = \mathbf{1}$). We consider two different examples of the desired state in our experiments, namely $\mathbf{z} = \mathbf{0}$ and $\mathbf{z} = \mathbf{1}$.

Table 1 corresponds to a zero desired state. Note that in this case, case $\mathbf{u} = \mathbf{0}$ gives a zero residual and the problem has a zero residual solution. Using $\epsilon_g = 10^{-5}$ and $\epsilon_R = 10^{-9}$, we observe that the residual criterion of (6) is triggered before the scaled gradient condition, and that only Gauss-Newton reaches the desired accuracy (note however that all final residual values correspond to an objective function value smaller than 10^{-11}). The Gauss-Newton+CG variant reverted to a gradient step after 30 iterations due to small curvature encountered while applying conjugate gradient (such behavior only occurred on this specific example). Still, it produced an iterate with smaller residual than gradient descent at a lower cost than exact Gauss-Newton in terms of Jacobian-vector products. Indeed, considering that one Jacobian evaluation requires n Jacobian-vector products, one obtains that $888 < 32n = 58528$.

In Table 2, we use the same tolerances but the desired state is now the vector of all ones, leading to a problem with large residuals. As a result, the scaled gradient condition is a better stopping criterion, as evidenced by the results. Note that all methods converge, with the Gauss-Newton variants taking less iterations and producing the lowest residual solution.

Method	Gradient	Gauss-Newton	Gauss-Newton+CG
Iterations	300	32	300
Jacobian/Jacobian-vector products	162	33	888
Final residual	1.57e-07	8.51e-11	1.16e-09
Final scaled gradient norm	2.25e-04	2.17e-04	5.94e-04

Table 1. Results for three variants of Algorithm 3 on the elliptic PDE problem (4.1) *using* $\mathbf{z} = \mathbf{0}$ *as desired state.*

Method	Gradient	Gauss-Newton	Gauss-Newton+CG
Iterations	37	25	25
Jacobian/Jacobian-vector products	30	26	290
Final residual	7.17e-01	7.17e-01	7.17e-01
Final scaled gradient norm	7.02e-06	5.07e-06	5.07e-06

Table 2. Results for three variants of Algorithm 3 on the elliptic PDE problem (4.1) *using* $\mathbf{z} = \mathbf{1}$ *as desired state.*

4.2 Burgers' equation

We now describe our second test problem, based on Burgers' equation, a simplified model for turbulence [2, 8, 19, 12]. Control problems based on this equation are often considered as the most fundamental nonlinear problem to handle. In our case, they illustrate the performance of our algorithms in a nonlinear, implicitly constrained setting.

Our formulation is as follows:

$$\begin{cases} \min_{y,u} J(y,u) := & \frac{1}{2}\int_0^T \int_0^L \left[(y(t,x)-z(t,x))^2 + \omega u(t,x)^2\right] dt\, dx & \\ \text{subject to} & y_t + \frac{1}{2}\left(y^2+\nu y_x\right)_x = f+u & (x,t)\in(0,L)\times(0,T) \\ & y(t,0) = y(t,L) = 0 & t\in(0,T) \\ & y(0,x) = y_0(x) & x\in(0,L). \end{cases} \tag{41}$$

Here L and T are space and time horizons, respectively; $u : [0,T]\times[0,L]\to\mathbb{R}$ is the control of our optimization problem; $y : [0,T]\times[0,L]\to\mathbb{R}$ is the state; $z : [0,T]\times[0,L]\to\mathbb{R}$ is the desired state; $\omega > 0$ is a regularization parameter; f is a source term, and ν is the viscosity parameter.

Given u, y can be computed by solving the PDE

$$\begin{array}{lll} y_t + \frac{1}{2}\left(y^2+\nu y_x\right)_x & = f+u & (x,t)\in(0,L)\times(0,T) \\ y(t,0) = y(t,L) & = 0 & t\in(0,T) \\ y(0,x) & = y_0(x) & x\in(0,L). \end{array} \tag{42}$$

We discretize (42) in time by applying the backward Euler scheme to Burgers' equation and a rectangle rule for the discretization of the objective function, while the spatial variable is approximated by piecewise linear finite elements. As a result, we obtain the following discretized version of problem (41):

$$\begin{cases} \text{minimize}_{u_0,\ldots,u_{N_t}\in\mathbb{R}^{N_x}} & J(y_0,\ldots,y_{N_t},u_0,\ldots,u_{N_t}) \\ \text{subject to} & c_{i+1}(y_i,y_{i+1},u_{i+1};\nu) = 0, \quad i = 0,\ldots,N_t-1, \end{cases} \tag{43}$$

where

$$J(y_0, \ldots, y_{N_t}, u_0, \ldots, u_{N_t}) := \delta_t \sum_{i=0}^{N_t} \left(\tfrac{1}{2}(y_i - z)^{\mathrm{T}} M (y_i - z) + \tfrac{\omega}{2} u_i^{\mathrm{T}} M u_i \right) \tag{44}$$

and

$$c_{i+1}(y_i, y_{i+1}, u_{i+1}; \nu) = \frac{1}{\delta_t} M y_{i+1} - \frac{1}{\delta_t} M y_i + \frac{1}{2} B y_{i+1} \odot y_{i+1} + \nu C y_{i+1} - f - M u_{i+1}, \tag{45}$$

and $\odot$ denotes the entrywise product. In those equations, $\delta_t = \frac{T}{N_t}$ represents the time step of the discretization, while $M, B, C, \{f_i\}$ are discretized versions of the operators and the source term arising from the continuous formulation. More precisely, we have

$$M = \frac{h}{6} \begin{bmatrix} 4 & 1 & & & \\ 1 & 4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 4 & 1 \\ & & & 1 & 4 \end{bmatrix} \in \mathbb{R}^{N_x \times N_x}, \quad B = \begin{bmatrix} 0 & 1/2 & & & \\ -1/2 & 0 & 1/2 & & \\ & \ddots & \ddots & \ddots & \\ & & -1/2 & 0 & 1/2 \\ & & & -1/2 & 0 \end{bmatrix} \in \mathbb{R}^{N_x \times N_x}$$

and

$$C = \frac{1}{h} \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix} \in \mathbb{R}^{N_x \times N_x}, \quad f = \begin{bmatrix} f_1 \\ \vdots \\ f_{N_x} \end{bmatrix} \in \mathbb{R}^{N_x},$$

with $h = \frac{L}{N_x}$ being the space discretization step. Following previous work [2, 12], we assume that the desired state z does not depend on time.

To reduce the effects of boundary layers, we discretize Burgers' equation using continuous piecewise linear finite elements built on a piecewise uniform mesh. We then solve the resulting discretized nonlinear PDE at each time step using Newton's method [2].

Letting $\mathbf{u}$ (resp. $\mathbf{y}$) as the concatenation of $u_0, \ldots, u_{N_t}$ (resp. $y_0, \ldots, y_{N_t}$), one observes that the objective function can be written as $\frac{1}{2}\|R(\mathbf{y}, \mathbf{u})\|^2$ with

$$R(\mathbf{y}, \mathbf{u}) = \begin{bmatrix} \sqrt{\delta_t} M^{1/2}(y_0 - z) \\ \vdots \\ \sqrt{\delta_t} M^{1/2}(y_{N_t} - z) \\ \sqrt{\omega \delta_t} M^{1/2} u_0 \\ \vdots \\ \sqrt{\omega \delta_t} M^{1/2} u_{N_t} \end{bmatrix} \in \mathbb{R}^{2(N_t+1)N_x}. \tag{46}$$

In our experimental setup, we use $L = T = 1$, $N_x = N_t = 50$, $\omega = 0.05$, and $f = 0$. We set $z = y_0$ with the first $N_x/2$ coefficients equal to 1 and the others equal to 0, while the initial control u_0 is set to the zero vector.

We ran our three algorithms using $\epsilon_g = 10^{-5}$ and $\epsilon_R = 10^{-9}$. Tables 3 and 4 report results for two values of the viscosity parameter. Both show that the stopping criterion (6) is satisfied thanks to the scaled gradient condition. As illustrated by Figures 1 and 2, the solutions

Method	Gradient	Gauss-Newton	Gauss-Newton+CG
Iterations	29	19	19
Jacobian/Jacobian-vector products	23	20	218
Final residual	4.35e-01	4.35e-01	3.43e-01
Final scaled gradient norm	8.82e-06	5.01e-06	5.01e-06

Table 3. Results for the variants of Algorithm 3 for the optimal control problem (41) *using* $\nu = 0.1$.

Method	Gradient	Gauss-Newton	Gauss-Newton+CG
Iterations	63	23	23
Jacobian/Jacobian-vector products	39	24	406
Final residual	3.43e-01	3.43e-01	3.43e-01
Final scaled gradient norm	6.56e-06	6.77e-06	6.77e-06

Table 4. Results for the variants of Algorithm 3 for the optimal control problem (41) *using* $\nu = 0.01$.

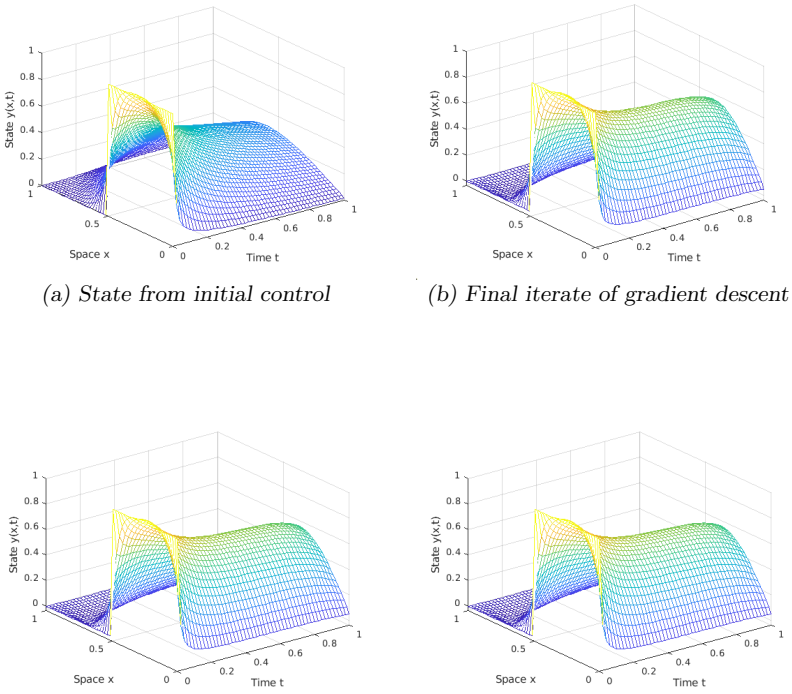
returned by the method yield similar state functions. However, we point out that using Gauss-Newton+CG steps results in the lowest number of iterations together with the lowest cost (since one Jacobian evaluation amounts to $n = N_t N_x$ Jacobian-vector products).

As illustrated by Table 4, the computation becomes more challenging as ν is smaller, since the instability grows exponentially with the evolution time [14]. Nevertheless, Figure 2 shows that all three methods improve significantly over the state corresponding to the initial control value. Note that plot on the top left hand panel in Figure 1 and Figure 2 represents the state corresponding to the initial control while the other three plots in each figure represent the final iterate of the states computed with the respective variants of Algorithm 3.

5 Conclusion

In this paper, we proposed a regularization method for least-squares problems subject to implicit constraints, for which we derived complexity guarantees that improve over recent bounds derived in the absence of constraints. To this end, we leveraged a recently proposed convergence criterion that is particularly useful when the optimal solution corresponds to nonzero objective value. Numerical testing conducted on PDE-constrained optimization problems showed that the criterion used to derive our complexity bounds bears a practical significance.

Our results can be extended in a number of ways. Deriving complexity results for second-order methods, that are common in scientific computing, is a natural continuation of our analysis. Besides, we aim at considering stochastic optimization problems under implicit constraints, in order to tackle not only machine learning problems, but also optimization problem under stochastic PDE constraints.



(a) State from initial control *(b) Final iterate of gradient descent*

(c) Final iterate of Gauss-Newton *(d) Final iterate of Gauss-Newton+CG*

Figure 1. State values for the optimal control problem (41) *using* $\nu = 0.1$. *The first plot shows the state for the initial value of the control, while the others show the state for the control returned by the corresponding variant of Algorithm 3.*

References

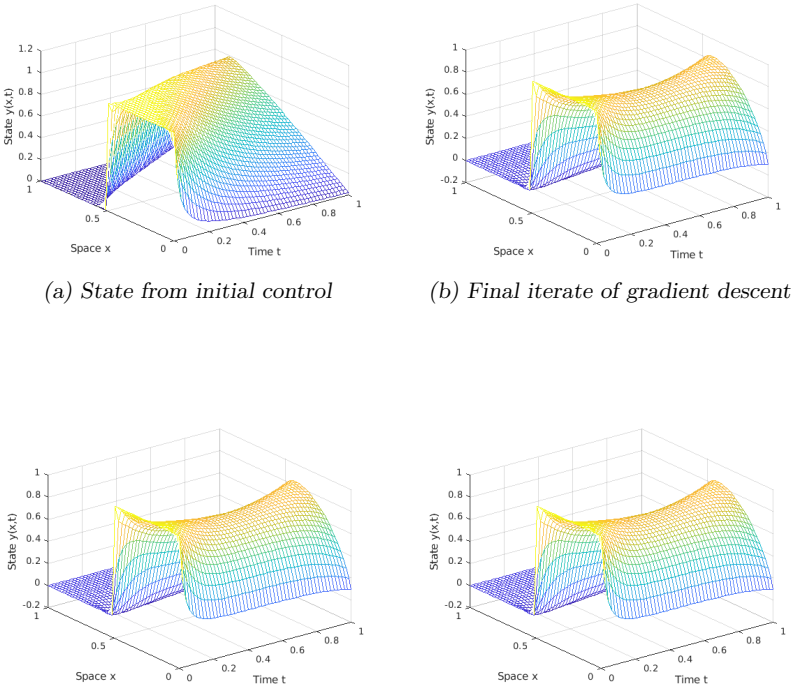
[1] H. Antil, D. P. Khouri, M.-D. Lacasse, and D. Ridzal, editors. *Frontiers in PDE-Constrained Optimization*, volume 163 of *The IMA Volumes in Mathematics and its Applications*. Springer, New York, NY, USA, 2016.

[2] M. M. Baumann. Nonlinear Model Order Reduction using POD/DEIM for Optimal Control of Burgers' Equation. Master's thesis, Faculty of Electrical Engineering, Mathematics and Computer Science Delft Institute of Applied Mathematics, Delft University of Technology, 2013.

[3] E. Bergou, Y. Diouane, and V. Kungurtsev. Convergence and complexity analysis of a Levenberg-Marquardt algorithm for inverse problems. *J. Optim. Theory Appl.*, 185:927–944, 2020.

[4] E. Bergou, Y. Diouane, V. Kungurtsev, and C. W. Royer. A stochastic Levenberg-Marquardt method for using random models with complexity results and application to data assimilation. *SIAM/ASA J. Uncertain. Quantif.*, 10:507–536, 2022.

[5] C. Cartis, N. I. M. Gould, and Ph. L. Toint. On the evaluation complexity of cubic regularization methods for potentially rank-deficient nonlinear least-squares problems and its relevance to constrained nonlinear optimization. *SIAM J. Optim.*, 23(3):1553–1574, 2013.



(a) State from initial control *(b) Final iterate of gradient descent*

(c) Final iterate of Gauss-Newton *(d) Final iterate of Gauss-Newton+CG*

Figure 2. State values for the optimal control problem (41) *using* $\nu = 0.01$. *The first plot shows the state for the initial value of the control, while the others show the state for the control returned by the corresponding variant of Algorithm 3.*

[6] C. Cartis, N. I. M. Gould, and Ph. L. Toint. *Evaluation Complexity of Algorithms for Nonconvex Optimization: Theory, Computation and Perspectives*, volume MO30 of *MOS-SIAM Series on Optimization*. SIAM, 2022.

[7] F. E. Curtis, D. P. Robinson, C. W. Royer, and S. J. Wright. Trust-region Newton-CG with strong second-order complexity guarantees for nonconvex optimization. *SIAM J. Optim.*, 31:518–544, 2021.

[8] J. C. de los Reyes and K. Kunisch. A comparison of algorithms for control constrained optimal control of the burgers equation. *CALCOLO*, 41:203 – 225, 2001.

[9] H. Elman, D. Silvester, and A. Wathen. *Finite Elements and Fast Iterative Solvers*, volume Second Edition. Oxford University Press, 2014.

[10] N. I. M. Gould, T. Rees, and J. A. Scott. A higher order method for solving nonlinear least-squares problems. Technical Report RAL-TR-2017-010, STFC Rutherford Appleton Laboratory, 2017.

[11] N. I. M. Gould, T. Rees, and J. A. Scott. Convergence and evaluation-complexity analysis of a regularized tensor-Newton method for solving nonlinear least-squares problems. *Comput. Optim. Appl.*, 73:1–35, 2019.

[12] M. Heinkenschloss. Lecture notes CAAM 454 / 554 – Numerical Analysis II. Rice University, Spring 2018.

[13] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich. *Optimization with PDE Constraints*. Springer Dordrecht, 2009.

[14] N. C. Nguyen, G. Rozza, and A. T. Patera. Reduced basis approximation and a posteriori error estimation for the time-dependent viscous Burgers' equation. *Calcolo*, 46:157–185, 2009.

[15] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer-Verlag, New York, second edition, 2006.

[16] C. W. Royer and S. J. Wright. Complexity analysis of second-order line-search algorithms for smooth nonconvex optimization. *SIAM J. Optim.*, 28:1448–1477, 2018.

[17] L. Ruthotto and E. Haber. Deep neural networks motivated by partial differential equations. *J. Math. Imaging Vision*, 62:352–364, 2020.

[18] F. Troeltzsch. *Optimal Control of Partial Differential Equations: Theory, Methods and Applications*. American Mathematical Society, 2010.

[19] F. Troeltzsch and S. Volkwein. The SQP method for the control constrained optimal control of the Burgers equation. *ESAIM: Control, Optimisation and Calculus of Variations*, 6:649 – 674, 2001.

[20] M. Ulbrich. *Semismooth Newton Methods for Variational Inequalities and Constrained Optimization Problems in Function Spaces*, volume MO11 of *MOS-SIAM Series on Optimization*. SIAM, 2011.