Les Cahiers du GERAD ISSN: 0711–2440

A regularized factorization-free method for equality-constrained optimization

S. Arreckx
D. Orban

G–2016–65

August 2016

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

A regularized factorization-free method for equality-constrained optimization

Sylvain Arreckx
Dominique Orban

GERAD & Department of Mathematics and Industrial Engineering, Polytechnique Montréal, Montréal (Québec) Canada

sylvain.arreckx@gerad.ca
dominique.orban@gerad.ca

August 2016

Les Cahiers du GERAD
G–2016–65

Abstract: We propose a factorization-free method for equality-constrained optimization based on a problem in which all constraints are systematically regularized. The regularization is equivalent to applying an augmented Lagrangian method but the linear system used to compute a search direction is reminiscent of regularized sequential quadratic programming (SQP). A limited-memory BFGS approximation to second derivatives allows us to employ iterative methods for linear least squares to compute steps, resulting in a factorization-free implementation. We establish global and fast local convergence under weak assumptions. In particular, we do not require the LICQ and our method is suitable for degenerate problems. Numerical experiments show that our method significantly outperforms IPOPT with limited-memory BFGS approximations, which is a state-of-the-art implementation of SQP on equality-constrained problems. We include a discussion on generalizing our framework to other classes of methods and to problems with inequality constraints.

Keywords: Sequential quadratic programming, regularization, augmented Lagrangian, limited-memory BFGS, factorization-free method

Résumé : Dans cet article, nous proposons une méthode d'optimisation sans factorisation pour les problèmes avec contraintes d'égalité pour lesquels toutes les contraintes sont systématiquement régularisées. Cette régularisation est équivalente à l'application d'une méthode de lagrangien augmenté dans laquelle les systèmes linéaires utilisés pour calculer une direction de recherche sont similaires à ceux des méthodes de programmation quadratique séquentielle (SQP). Grâce à l'emploi d'approximations BFGS à mémoire limitée des dérivées secondes, des méthodes itératives pour les moindres carrés linéaires peuvent être utilisées afin de calculer par étapes, faisant de la méthode proposée une méthode sans factorisation. Nous établissons rapidement une convergence globale et locale sous de faibles hypothèses. En particulier, la LICQ n'est pas requise et notre méthode est adaptée pour la résolution de problèmes dégénérés. Les tests numériques montrent que notre méthode est bien plus performante que IPOPT lorsque des approximations BFGS à mémoire limitée sont utilisées. Une discussion est incluse sur la généralisation de notre approche à d'autres classes de méthodes ainsi qu'aux problèmes avec inégalités.

Mots clés : Programmation quadratique séquentielle, régularisation, méthode de lagrangien augmenté, BFGS à mémoire limitée, méthode sans factorisation

Acknowledgments: Research partially supported by an NSERC Discovery Grant.

1 Introduction

We consider the general equality-constrained optimization problem

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\ f(x) \quad \text{subject to}\ c(x)=0. \tag{1}$$

The main objective of this paper is to devise an implementable factorization-free algorithm for (1) in the large-scale case that is somewhat resilient to constraint degeneracy and does not require matrix-vector products with $J(x)^T J(x)$, where J is the Jacobian of c, as is the case with a standard augmented Lagrangian method. We propose a framework inspired by that of (Armand, Benoist, and Orban, 2012) in which all constraints are systematically regularized. We show that the regularization can be interpreted as a proximal-point Hestenes-Powell augmented Lagrangian method applied to (1) in the same vein as Rockafellar (1976). Our method uses the proximal augmented Lagrangian as merit function to promote global convergence and asymptotically blends into a stabilized SQP method possessing fast local convergence properties. Thanks to appropriate limited-memory BFGS approximations of the Hessian of the Lagrangian, the linear system encountered at each iteration is symmetric and quasi-definite (SQD) (Vanderbei, 1995), permitting inexact solves and an entirely factorization-free implementation suggested by methods described by Arioli and Orban (2013). We assume that $f:\mathbb{R}^n\to\mathbb{R}$ and $c:\mathbb{R}^n\to\mathbb{R}^m$ are twice continuously differentiable, although we only use exact second derivatives as an instrument in the analysis and in numerical illustration. In practice, only first derivatives are required when employing L-BFGS approximations.

The Karush-Kuhn-Tucker (KKT) conditions for (1) are only necessary for optimality when a constraint qualification holds. The most widely used constraint qualification condition, the Linear Independence Constraint Qualification (LICQ), requires that all constraint gradients be linearly independent at a stationary point. When such a constraint qualification fails to hold, the KKT conditions cease to be reliable for optimality. When it fails to hold at intermediate iterates, computational difficulties also arise. In particular, the linear systems used to compute search directions may become singular. Our regularization scheme is designed so as to overcome such complications.

We show that our method possesses global convergence properties similar to those of augmented Lagrangian methods (Bertsekas, 1996; Birgin and Martínez, 2014). In addition, we show that whenever the sequence of iterates converges to an isolated minimizer, the algorithm reduces asymptotically to pure stabilized SQP iterations and converges superlinearly. The convergence rate is quadratic if second-derivative approximations and steps are sufficiently accurate. Our numerical experiments show that the proposed scheme is efficient and robust, even when solving problems for which the LICQ fails to hold at the solution or at intermediate iterates, and compares very favorably to IPOPT (Wächter and Biegler, 2006).

Related work

Sequential quadratic programming (SQP) methods (Boggs and Tolle, 1995; Wilson, 1963) are among the most successful methods for the solution of (1). They compute steps via a sequence of subproblems in which a quadratic model of the Lagrangian is minimized subject to linearized constraints. Convergence is enforced by requiring an improvement in a merit function at each step. Each iteration of an SQP method requires the solution of a linear system that involves the constraint Jacobian and its transpose. Most convergence analyses for SQP and most SQP implementations require that those linear systems be solved exactly. In many large-scale applications, constraint Jacobians are only available as linear operators. In such cases, systems must be solved iteratively and inexactly, and it is crucial to account for this inexactness in the design and convergence analysis. An inexact trust-region SQP algorithm for equality constrained optimization is introduced in Heinkenschloss and Ridzal (2014). A composite-step approach is described in which the step in decomposed into a quasi-normal and a tangential step. A set of stopping criteria designed for controlling the inexactness of substep computations is given and ensures global convergence of their algorithm. Byrd, Curtis, and Nocedal (2009) propose an inexact line-search SQP method for (1) where steps are computed from an inexact solution of a KKT system. A perturbation of the Hessian of the Lagrangian is employed to deal with nonconvexity. The perturbation is determined iteratively and may require repeated KKT solves per

step computation. Two distinct termination tests control the level of inexactness in the step computation procedure.

Stabilized SQP methods were designed to remedy the numerical and theoretical difficulties associated with degenerate problems (Fernàndez and Solodov, 2010; Hager, 1999; Wright, 2005). The term *stabilized* refers to the calming effect on multiplier estimates for degenerate problems (Hager, 1999; Wright, 1998). Stabilized SQP promises superlinear local convergence under certain assumptions, but not global convergence to a stationary point. Few globalizations of the local stabilized SQP scheme have been proposed so far. Fernández, Pilotta, and Torres (2012) combine stabilized SQP with the inexact restoration method to ensure convergence from an arbitrary starting point. Gill and Robinson (2013) establish connections between stabilized SQP and augmented Lagrangian methods, including primal-dual variants of the augmented Lagrangian. Izmailov, Solodov, and Uskov (2015) combine stabilized SQP with the usual augmented Lagrangian algorithm when inequalities are present. However, their algorithm doesn't allow the use of quasi-Newton approximations to second-order derivatives and the linear systems involved during optimization must be solved exactly. Armand and Omheni (2015) propose a primal-dual augmented Lagrangian approach to solve equality constrained optimization problems that is quadratically convergent. However, convergence assumes the LICQ, exact linear system solves and exact second derivatives.

Notation

Throughout the paper, $\|\cdot\|$ denotes the Euclidean norm and I denotes the identity matrix of appropriate size. For any symmetric and positive definite matrix H, the H-norm is defined as $\|u\|_H^2 := u^T H u$. We use $\lambda_{\min}(M)$ and $\lambda_{\max}(M)$ to denote the smallest and largest eigenvalue of any symmetric matrix M. Similarly, $\sigma_{\min}(A)$ and $\sigma_{\max}(A)$ denote the smallest and largest singular values of any matrix A. For two non-negative scalar sequences a_k and b_k converging to zero, we use the Landau symbols $a_k = o(b_k)$ if $\lim_{k\to+\infty} a_k/b_k = 0$ and $a_k = \omega(b_k)$ if $b_k = o(a_k)$. We write $a_k = O(b_k)$ if there exists a constant $C > 0$, such that $a_k \leq C b_k$ for large k and $a_k = \Theta(b_k)$ if $a_k = O(b_k)$ and $b_k = O(a_k)$.

The rest of the paper is organized as follows. Section 2 summarizes the connexion between augmented Lagrangians and regularized SQP methods. In Section 3, we describe our algorithm in detail. Its global convergence properties are given in Section 4. Local convergence is analyzed in Section 5. We describe our implementations and report on numerical experience in Section 6. We conclude and discuss extensions to our framework in Section 7.

2 A primal-dual regularization and regularized SQP methods

The Lagrangian for (1) is defined as

$$L(x, y) := f(x) - c(x)^T y, \tag{2}$$

where $y \in \mathbb{R}^m$ is the vector of Lagrange multipliers associated to the equality constraints. If x^* is a local minimizer of (1), the KKT conditions require that there exist y^* such that

$$g(x^*) - J(x^*)^T y^* = 0, \quad c(x^*) = 0, \tag{3}$$

where $g(x) := \nabla f(x)$ and $J(x)$ is the Jacobian of $c(x)$. Existence of such a y^* is only guaranteed provided a constraint qualification condition holds at x^*. Should constraint qualifications fail to hold at x^*, there may exist no y^* satisfying (3) or there may exist an unbounded set of them (Gauvin, 1977). In either case, numerical methods, such as SQP methods, may be confronted with degenerate direction-finding subproblems.

For the purposes of this paper, we say that (1) is *degenerate* at a feasible x if the LICQ fails to hold at x, i.e., the vectors $\nabla c_i(x)$, $i = 1, \ldots, m$, are linearly dependent.

Consider applying an augmented Lagrangian method to (1). If we denote y_k the current approximation of the Lagrange multipliers, the k-th subproblem has the form

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \;\; L(x, y_k) + \tfrac{1}{2}\delta_k^{-1}\|c(x)\|^2, \tag{4}$$

where $\delta_k > 0$ is a penalty parameter. Following the procedure outlined by Friedlander and Orban (2012), it is not difficult to see that (4) may equivalently be written as

$$\underset{x\in\mathbb{R}^n, u\in\mathbb{R}^m}{\text{minimize}} \;\; f(x) + \tfrac{1}{2}\delta_k\|u + y_k\|^2 \quad \text{subject to} \;\; c(x) + \delta_k u = 0, \tag{5}$$

for some new variables u. The problem (5) provides an interpretation of the augmented Lagrangian method as an adaptive constraint regularization process. Since the regularization acts on the constraints and adds a term to the objective involving the multipliers, we term it *dual*. Of paramount importance is the fact that the LICQ is satisfied at every feasible point of (5).

In addition to the dual regularization term, we follow Friedlander and Orban (2012) and add primal regularization in the form of a proximal-point term, so that the k-th subproblem takes the form

$$\underset{x\in\mathbb{R}^n, u\in\mathbb{R}^m}{\text{minimize}} \;\; f(x) + \tfrac{1}{2}\rho_k\|x - x_k\|^2 + \tfrac{1}{2}\delta_k\|u + y_k\|^2 \quad \text{subject to} \;\; c(x) + \delta_k u = 0, \tag{6}$$

for a primal regularization parameter $\rho_k \geq 0$, where x_k is the current primal iterate.

The KKT conditions for (6),

$$\begin{bmatrix} g(x) + \rho_k(x - x_k) - J(x)^T y \\ \delta_k(u + y_k) - \delta_k y \\ c(x) + \delta_k u \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{7}$$

are therefore unconditionally necessary for optimality for any fixed value of $\rho_k \geq 0$ and $\delta_k > 0$. The following relationship between KKT points of (1) and those of (6) illustrates the fact that the primal-dual regularization is *exact*.

Theorem 1 *Suppose (x_k, u_k, y_k) is a KKT point of (6) for some $\rho_k \geq 0$ and $\delta_k > 0$. Then (x_k, y_k) is a KKT point of (1).*
Alternatively, suppose $\rho_k = 0$ and $(\bar{x}, \bar{u}, \bar{y})$ is a KKT point of (6) for some $\delta_k > 0$ and suppose $\bar{x}$ is feasible for (1). Then $\bar{u} = 0$, $\bar{y} = y_k$ and $(\bar{x}, \bar{y})$ is a KKT point of (1).
Conversely, suppose (x^*, y^*) is a KKT point of (1). Then $(x_k, 0, y_k) := (x^*, 0, y^*)$ is a KKT point of (6) for any $\rho \geq 0$ and $\delta > 0$.

Proof. Immediate, by direct comparison of (3) and (7). □

Sequential quadratic programming methods for (6) may be interpreted as applying Newton's method to (7). A Newton-like step for (7) from (x_k, u_k, y_k) solves the linear system

$$\begin{bmatrix} H_k + \rho_k I & & -J_k^T \\ & \delta_k I & -\delta_k I \\ J_k & \delta_k I & \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta u \\ \Delta y \end{bmatrix} = - \begin{bmatrix} g_k - J_k^T y_k \\ \delta_k u_k \\ c_k + \delta_k u_k \end{bmatrix}, \tag{8}$$

where $g_k := g(x_k)$, $c_k := c(x_k)$, $J_k := J(x_k)$ and H_k is a symmetric approximation of $\nabla_{xx} L(x_k, y_k)$. The elimination of $\Delta u = -u_k + \Delta y$ yields the reduced system

$$\begin{bmatrix} H_k + \rho_k I & J_k^T \\ J_k & -\delta_k I \end{bmatrix} \begin{bmatrix} \Delta x \\ -\Delta y \end{bmatrix} = - \begin{bmatrix} g_k - J_k^T y_k \\ c_k \end{bmatrix}, \tag{9}$$

which is the familiar system encountered in stabilized SQP methods, e.g., (Wright, 1998), while the system used in classical SQP methods corresponds to $\delta_k = 0$. For simplicity in the rest of this paper, the coefficient matrix of (9) is referred as K_k. The system (9) may be interpreted as the KKT conditions of the quadratic subproblem

$$\begin{aligned} \underset{\Delta x, \Delta u}{\text{minimize}} \quad & \nabla_x L(x_k, y_k)^T \Delta x + \tfrac{1}{2}\Delta x^T (H_k + \rho_k I)\Delta x + \tfrac{1}{2}\delta_k\|u_k + \Delta u\|^2 \\ \text{subject to} \quad & c_k + J_k \Delta x + \delta_k(u_k + \Delta u) = 0. \end{aligned} \tag{10}$$

Note that (10) itself always satisfies the LICQ and therefore infeasible subproblems never occur. The primal regularization term $\rho_k I$ may be interpreted as a convexifying term that encourages descent in an appropriate merit function.

Given a fixed $\bar{x} \in \mathbb{R}^n$, we define

$$\phi(x, y; \bar{x}, \rho, \delta) := f(x) - c(x)^T y + \tfrac{1}{2}\rho\|x - \bar{x}\|^2 + \tfrac{1}{2}\delta^{-1}\|c(x)\|^2. \tag{11}$$

For future reference, we note that

$$\nabla_x \phi(\bar{x}, y; \bar{x}, \rho, \delta) = \nabla_x L(\bar{x}, y) + \delta^{-1} J(\bar{x})^T c(\bar{x}) = g(\bar{x}) - J(\bar{x})^T (y - \delta^{-1} c(\bar{x})). \tag{12}$$

For simplicity of exposition, we write $\phi(x, y; \rho, \delta)$ instead of $\phi(x, y; x, \rho, \delta)$. We also let $w := (x, y)$ and define $F : \mathbb{R}^{n+m} \to \mathbb{R}^{n+m}$ as $F(w) := (\nabla_x L(w), c(x))$.

3 Main algorithm

Algorithm 1 is a simplification of (Armand et al., 2012, Algorithm 1) that ignores inequality constraints and allows symmetric Hessian approximations. In the description, we make use of the norm $\|F(w)\|_* := \|\nabla_x L(w)\| + \|c(x)\|$.

Algorithm 1 Outer iteration

1: Choose $\alpha \in (0, 1)$, $\theta \in (0, 1)$, and $\epsilon > 0$. Set $k = 0$.
2: If $\|F(w_k)\| < \epsilon$, terminate with final iterate w_k.
3: Choose a symmetric matrix H_k, $\rho_k \geq 0$, and $\delta_k \in [\min(\|F(w_k)\|, \alpha\delta_{k-1}), \delta_{k-1}]$. Compute a trial iterate w_k^+ as an approximate solution of

$$K_k(w_k^+ - w_k) + F(w_k) = 0. \tag{13}$$

4: Choose $\epsilon_k > 0$. If

$$\|F(w_k^+)\|_* \leq \theta\|F(w_k)\|_* + \epsilon_k, \tag{14}$$

then set $w_{k+1} = w_k^+$. Otherwise perform a sequence of inner iterations in order to find a new iterate w_{k+1} such that

$$\|F(w_{k+1})\|_* \leq \theta\|F(w_k)\|_* + \epsilon_k. \tag{15}$$

Increment k by one and return to Step 2.

The main idea of Algorithm 1 is to start each outer iteration with an extrapolation step (13). The extrapolation step is accepted if it achieves sufficient improvement in the first-order optimality residual. In the negative, an inner iteration procedure is started in order to identify an improved iterate.

A certain amount of flexibility is allowed in choosing the parameters ρ_k and δ_k at the beginning of each outer iteration. An important feature for global convergence is that it is allowed to keep δ_k fixed, at least after a certain number of iterations. An important feature for fast local convergence is that it is allowed to select $\delta_k = \|F(w_k)\|$ when close to an isolated minimizer.

Algorithm 2 describes the linesearch procedure used as the inner iteration, which is essentially a standard augmented Lagrangian subproblem solve in which steps are computed using the augmented form (16) followed by a Wolfe linesearch on the proximal augmented Lagrangian. During the inner iterations, y_k is kept fixed. The inner primal iterates and regularization parameter corresponding to the k-th outer iteration are denoted $x_{k,j}$, $\rho_{k,j}$ and $\delta_{k,j}$ for $j \geq 0$. The inner iterations stop as soon as the first-order optimality residual of (1) has sufficiently decreased. As in the standard augmented Lagrangian, the penalty parameter is decreased when dual feasibility improved but primal feasibility lags behind.

In Step 3 of Algorithm 1 and Step 3 of Algorithm 2, the linear system could be solved exactly, but there is flexibility to compute an inexact solution. The inexactness in the outer iteration is a departure from the framework of Armand et al. (2012).

In Algorithm 2, steplengths are computed so as to satisfy the Wolfe conditions. The reason for this requirement is that the global convergence analysis is based on a simple application of Zoutendijk's theorem

Algorithm 2 Inner iteration

1: Set j to 0. Choose an initial guess $w_{k,0}$ and $\delta_{k,0} > 0$. Choose c_1 and c_2 such that $0 < c_1 < c_2 < 1$.
2: If $\|\nabla_x L(x_{k,j}, y_k - \delta_{k,j}^{-1} c(x_{k,j}))\| \leq \theta \|\nabla_x L(x_{k,0}, y_k)\| + \frac{1}{2}\epsilon_k$, then

if $\|c(x_{k,j})\| \leq \theta \|c(x_{k,0})\| + \frac{1}{2}\epsilon_k$, stop with $w_{k+1} = (x_{k,j}, y_k - \delta_{k,j}^{-1} c(x_{k,j}))$,
otherwise, set $\delta_{k,j} = \delta_{k,j-1}/10$.

Go to Step 3.
3: Choose a symmetric matrix $H_{k,j}$ and $\rho_{k,j} \geq 0$. Compute Δx_j as an approximate solution to

$$\begin{bmatrix} H_{k,j} + \rho_{k,j} I & J_{k,j}^T \\ J_{k,j} & -\delta_{k,j} I \end{bmatrix} \begin{bmatrix} \Delta x_j \\ -\Delta y_j \end{bmatrix} = - \begin{bmatrix} g_{k,j} - J_{k,j}^T y_k \\ c_{k,j} \end{bmatrix}. \tag{16}$$

4: Set $x_{k,j+1} = x_{k,j} + \alpha_j \Delta x_j$, where α_j is obtained using a line search and satisfies the Wolfe conditions:

$$\phi(x_{k,j+1}, y_k; x_{k,j}, \rho_{k,j}, \delta_{k,j}) \leq \phi(x_{k,j}, y_k; \rho_{k,j}, \delta_{k,j}) + c_1 \alpha_j \nabla\phi(x_{k,j}, y_k; \rho_{k,j}, \delta_{k,j})^T \Delta x_j$$
$$\nabla\phi(x_{k,j+1}, y_k; x_{k,j}, \rho_{k,j}, \delta_{k,j})^T \Delta x_j \geq c_2 \nabla\phi(x_{k,j}, y_k; \rho_{k,j}, \delta_{k,j})^T \Delta x_j.$$

Increment j by one and return to Step 2.

(Nocedal and Wright, 2006, Theorem 3.2), which requires the Wolfe conditions. We believe it is possible to develop a global convergence analysis based solely on the Armijo condition, in the vein of Armand and Omheni (2015).

The next section examines the global convergence properties of Algorithms 1 and 2 and, in particular, what conditions should be imposed on inexact steps.

4 Global convergence

In this section, we examine in turn the convergence of Algorithms 1 and 2.

4.1 Convergence of the inner iterations

The convergence of the inner iterations is divided into two parts depending on whether linear systems are solved exactly or not. In Section 4.1.1, we assume that the linear systems of Algorithms 1 and 2 are solved exactly. That situation covers the case where the systems are solved via a factorization, whether exact second derivatives are used or not. It also covers the case where the linear systems are solved iteratively but with an extremely tight tolerance, although that is not realistic in practice. In Section 4.1.2, we assume that systems are solved inexactly and we describe the iterative procedure that we use. The latter relies on H_k being positive definite and such that linear systems with coefficient H_k can be solved easily and cheaply. Such a situation occurs when H_k is a limited-memory BFGS approximation to the second derivatives, and that is our selection of choice. With that choice, H_k itself is never really needed and we simply maintain its inverse implicitly (Liu and Nocedal, 1989). Other choices are of course possible, including $H_k = I$, or a positive-definite diagonal approximation of $\nabla^2_{xx} L(x_k, y_k)$.

In this section, k denotes the outer iteration index and appears everywhere to avoid ambiguity. We refer to the coefficient of the linear system at outer iteration k and inner iteration j in Step 3 of Algorithm 2 as $K_{k,j}$.

Our main working assumption is as follows.

Assumption 1 *The gradients $g_{k,j}$, the matrices $H_{k,j}$ and the matrices $J_{k,j}$ are uniformly bounded for all $j \in \mathbb{N}$. Moreover, $\rho_{k,j}$ is bounded for all $j \in \mathbb{N}$.*

4.1.1 Exact system solves

When $H_{k,j}$ is not positive definite, which is typically the case when exact second-derivatives are used, $\rho_{k,j}$ must be sufficiently large to ensure that Δx_j is a descent direction for the proximal augmented Lagrangian. Linear

systems may be solved using a symmetric indefinite factorization such as the multifrontal implementation MA57 of Duff (2004). Because this factorization reveals the inertia of $K_{k,j}$, the regularization parameter $\rho_{k,j}$ can be increased until the correct inertia is detected (Gould, 1985). Such a procedure is similar to that used in IPOPT (Wächter and Biegler, 2006). Those observations motivate the following assumption.

Assumption 2 *The matrices $H_{k,j} + \rho_{k,j} I + \delta_{k,j}^{-1} J_{k,j}^T J_{k,j}$ are uniformly positive definite and uniformly bounded for all $j \in \mathbb{N}$, i.e., there exist constants $\overline{\sigma} \geq \underline{\sigma} > 0$ such that for all $j \in \mathbb{N}$ and all $d \in \mathbb{R}^n$,*

$$\underline{\sigma}\|d\|^2 \leq d^T (H_{k,j} + \rho_{k,j} I + \delta_{k,j}^{-1} J_{k,j}^T J_{k,j}) d \leq \overline{\sigma}\|d\|^2.$$

Assumption 2 implies that $\{H_{k,j} + \rho_{k,j} I\}$ is uniformly positive definite over the nullspace of $J_{k,j}$ for all $j \in \mathbb{N}$.

Theorem 2 (Inner iteration, exact solves) *Suppose that Assumption 2 holds and that $\phi(\cdot, y_k; \rho_{k,j}, \delta_{k,j})$ is bounded below for all j. Then Algorithm 2 generates a sequence of iterates $x_{k,j}$ such that*

$$\lim_{j \to +\infty} \|\nabla_x \phi(x_{k,j}, y_k; \rho_{k,j}, \delta_{k,j})\| = 0.$$

Proof. We eliminate Δy_j from (16) and use (12) to obtain

$$\left(H_{k,j} + \rho_{k,j} I + \delta_{k,j}^{-1} J_{k,j}^T J_{k,j}\right) \Delta x_j = -\nabla_x \phi(x_{k,j}, y_k; \rho_{k,j}, \delta_{k,j}). \tag{17}$$

We take the inner product of both sides of (17) with Δx_j and use Assumption 2, and obtain

$$-\nabla_x \phi(x_{k,j}, y_k; \rho_{k,j}, \delta_{k,j})^T \Delta x_j \geq \underline{\sigma}\|\Delta x_j\|^2. \tag{18}$$

Assumption 2 and (17) yield

$$\begin{aligned}
\|\Delta x_j\| &\geq \lambda_{\min}\left((H_{k,j} + \rho_{k,j} I + \delta_{k,j}^{-1} J_{k,j}^T J_{k,j})^{-1}\right) \|\nabla_x \phi(x_{k,j}, y_k; \rho_{k,j}, \delta_{k,j})\| \\
&= \left(\lambda_{\max}(H_{k,j} + \rho_{k,j} I + \delta_{k,j}^{-1} J_{k,j}^T J_{k,j})\right)^{-1} \|\nabla_x \phi(x_{k,j}, y_k; \rho_{k,j}, \delta_{k,j})\| \\
&\geq \overline{\sigma}^{-1} \|\nabla_x \phi(x_{k,j}, y_k; \rho_{k,j}, \delta_{k,j})\|.
\end{aligned}$$

Therefore

$$-\frac{\nabla_x \phi(x_{k,j}, y_k; \rho_{k,j}, \delta_{k,j})^T \Delta x_j}{\|\nabla_x \phi(x_{k,j}, y_k; \rho_{k,j}, \delta_{k,j})\| \, \|\Delta x\|} \geq \underline{\sigma}/\overline{\sigma} > 0.$$

Zoutendijk's theorem ensures that

$$\lim_{j \to \infty} \left(-\frac{\nabla_x \phi(x_{k,j}, y_k; \rho_{k,j}, \delta_{k,j})^T \Delta x}{\|\nabla \phi(x_{k,j}, y_k; \rho_{k,j}, \delta_{k,j})\| \|\Delta x\|} \right)^2 \|\nabla \phi(x_{k,j}, y_k; \rho_{k,j}, \delta_{k,j})\|^2 = 0,$$

from which the desired result follows immediately. □

Theorem 2 implies that the first stopping condition of Algorithm 1 is satisfied after a finite number of iterations because (12) can also be written

$$\nabla_x \phi(x_{k,j}, y_k; \rho_{k,j}, \delta_{k,j}) = \nabla_x L(x_{k,j}, y_k - \delta_{k,j}^{-1} c(x_{k,j})).$$

In order to determine when the second stopping condition holds, we consider the following assumption.

Assumption 3 *The sequence* $\{\delta_{k,j}\}_{j\in\mathbb{N}}$ *is bounded away from zero.*

If Assumption 3 holds, the mechanism of Algorithm 2 guarantees that the stopping condition $\|c(x_{k,j})\| \leq \theta\|c(x_{k,0})\| + \frac{1}{2}\epsilon_k$ is eventually satisfied as well.

If Assumption 3 fails, there is an index set $\mathcal{J}$ such that $\lim_{j\in\mathcal{J}} \delta_{k,j} = 0$. In the situation where $\nabla_x\phi(x_{k,j}, y_k; \rho_{k,j}, \delta_{k,j})$ remains bounded, we have

$$\begin{aligned} 0 &= \lim_{j\in\mathcal{J}} \delta_{k,j}\nabla_x\phi(x_{k,j}, y_k; \rho_{k,j}, \delta_{k,j}) \\ &= \lim_{j\in\mathcal{J}} \delta_{k,j}\left(g(x_{k,j}) - J(x_{k,j})^T y_k\right) + J(x_{k,j})^T c(x_{k,j}). \end{aligned}$$

Under Assumption 1, the first term of the previous left-hand side converges to zero, and therefore

$$\lim_{j\in\mathcal{J}} J(x_{k,j})^T c(x_{k,j}) = 0.$$

In other words, there is a limit point of the sequence generated by Algorithm 2 that is stationary for the (typically underdetermined) least-squares problem

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \ \tfrac{1}{2}\|c(x)\|^2.$$

This situation occurs when the augmented Lagrangian multiplier estimates $y_k - \delta_{k,j}^{-1}c(x_{k,j})$ remain bounded, which is a common assumption in the convergence analysis of augmented Lagrangian methods. Multiplier unboundedness may be an indication that the constraints are not linearly independent.

4.1.2 Inexact system solves: A quasi-Newton strategy

A difficulty associated with the iterative solution of systems of the form (16), e.g., using MINRES (Paige and Saunders, 1975), is the need to balance residuals associated to each block equation, as in (Byrd et al., 2009). The framework detailed in this section solves one of the block equations exactly while controlling the residual associated to the other. That is done by transforming (16) into the first-order optimality conditions of a preconditioned linear least-squares problem. The preconditioner used, $H_{k,j}^{-1}$ in the present case, must be applied at each iteration of an iterative method for least-squares problems. It is therefore crucial that $H_{k,j}^{-1}$ be positive definite and cheaply applicable.

In this section, we make the following assumption.

Assumption 4 *The matrices* $H_{k,j}$ *are uniformly positive definite for all* $j \in \mathbb{N}$.

For the above reasons, and in order to preserve hope for fast local convergence when close to an isolated minimizer, we set $H_{k,j}$ to a limited-memory BFGS approximation of the Hessian of the Lagrangian (Liu and Nocedal, 1989) that is possibly modified to take into account the fact that the Hessian of the Lagrangian cannot be expected to be positive definite. Details are not relevant here and will be given in Section 6. By construction, L-BFGS approximations are always positive definite. In addition, their inverse can be implicitly maintained and updated along the iterations, and can be applied to a vector cheaply using either the two-loop recursion or the compact storage format (Byrd, Nocedal, and Schnabel, 1994), or by maintaining the approximation in factored form (Dennis and Schnabel, 1996, Algorithm A9.4.2).

With such a choice, $K_{k,j}$ is SQD, and therefore nonsingular irrespective of the rank of $J_{k,j}$. We always set $\rho_{k,j} = 0$ when $H_{k,j}$ is a L-BFGS approximation.

We first cast (16) as a least-square problem. We introduce $\Delta\bar{y} := \Delta y + \delta_{k,j}^{-1}c_{k,j}$, and rewrite (16) equivalently as

$$\begin{bmatrix} H_{k,j} & J_{k,j}^T \\ J_{k,j} & -\delta_{k,j}I \end{bmatrix} \begin{bmatrix} \Delta x \\ -\Delta\bar{y} \end{bmatrix} = \begin{bmatrix} b_{k,j} \\ 0 \end{bmatrix} \tag{19}$$

where $b_{k,j} = -g_{k,j} + J_{k,j}^T(y_k - \delta_{k,j}^{-1} c_{k,j}) = -\nabla_x \phi(x_{k,j}, y_k; \rho_{k,j}, \delta_{k,j})$. Birgin and Martínez (2014) use a similar system obtained from linear algebra transformations of the Newton equations $\nabla^2_{xx}\phi(x, y; \rho)d = -\nabla_x \phi(x, y; \rho)$.

The shifted system (19) can be seen as the necessary and sufficient optimality conditions of the regularized and preconditioned least-squares problem

$$\underset{\Delta\bar{y}}{\text{minimize}} \ \tfrac{1}{2} \left\| J_{k,j}^T \Delta\bar{y} + b_{k,j} \right\|^2_{H_{k,j}^{-1}} + \tfrac{1}{2} \|\Delta\bar{y}\|^2_{\delta_{k,j} I}. \tag{20}$$

The latter can be solved approximately using a stopping criterion based exclusively on the residual of the second block equation of (19), i.e., the optimality residual of (20):

$$r_{k,j} := J_{k,j}\Delta x + \delta_{k,j}\Delta\bar{y}, \tag{21}$$

while the least-squares residual is $\Delta x := H_{k,j}^{-1}(J_{k,j}^T \Delta\bar{y} + b_{k,j})$.

Our choice is to solve (20) with the LSMR method of Fong and Saunders (2011) modified as recommended by Arioli and Orban (2013) to accommodate non-Euclidean norms. This choice is motivated by the fact that we wish to reduce the residual of (19), or equivalently, of (16), below a certain threshold. The least-squares interpretation guarantees that the first block equation is always satisfied exactly, by definition of the least-squares residual. The residual of the second block equation is precisely $r_{k,j}$. An important property of LSMR, that is not shared by other methods such as LSQR (Paige and Saunders, 1982), is that it decreases the norm of the optimality residual of (20), i.e., the norm of $r_{k,j}$, monotonically. Finally, Arioli and Orban (2013) show that using LSMR as described above can save half of the iterations as compared to using MINRES with the natural preconditioner blkdiag$(H_{k,j}, \delta_{k,j} I)$.

Our implementation uses two termination tests. The first guarantees sufficient descent.

Termination test 1 *Let $\{\gamma_j\}$ be any positive sequence that is bounded away from zero. A step $(\Delta x, \Delta\bar{y})$ is an acceptable inexact solution of (19) if*

$$\|r_{k,j}\|^2_{\delta_{k,j}^{-1} I} + \gamma_j \|b_{k,j}\|^2_{H_{k,j}^{-1}} \leq \|J_{k,j}^T \Delta\bar{y} + b_{k,j}\|^2_{H_{k,j}^{-1}} + \|\Delta\bar{y}\|^2_{\delta_{k,j} I}. \tag{22}$$

We defer comments on Termination test 1 until the end of this section.

Lemma 1 *Let Assumptions 1 and 4 be satisfied. Suppose that Termination test 1 is satisfied. Then there exists a constant $\gamma > 0$ such that*

$$-\nabla_x \phi(x_{k,j}, y_k; \rho_{k,j}, \delta_{k,j})^T \Delta x \geq \gamma \, \|\nabla_x \phi(x_{k,j}, y_k; \rho_{k,j}, \delta_{k,j})\|^2.$$

Proof. Let us denote $\phi_{k,j} := \phi(x_{k,j}, y_k; \rho_{k,j}, \delta_{k,j})$ and $M_{k,j} := H_{k,j} + \delta_{k,j}^{-1} J_{k,j}^T J_{k,j}$ for conciseness.

The first block equation of (19) and (21) yield

$$\begin{aligned} -\nabla_x \phi_{k,j}^T \Delta x &= \Delta x^T H_{k,j} \Delta x - \Delta\bar{y}^T J_{k,j} \Delta x \\ &= \Delta x^T H_{k,j} \Delta x + \delta_{k,j} \Delta\bar{y}^T \Delta\bar{y} - \Delta\bar{y}^T r_{k,j}. \end{aligned} \tag{23}$$

Isolating $\Delta\bar{y}$ in the second block equation of (19), substituting it into the first one and using (21) gives

$$-\nabla_x \phi_{k,j} = M_{k,j} \Delta x - \delta_{k,j}^{-1} J_{k,j}^T r_{k,j}. \tag{24}$$

Thus

$$-\nabla_x \phi_{k,j}^T \Delta x = \Delta x^T M_{k,j} \Delta x - \delta_{k,j}^{-1} \Delta x^T J_{k,j}^T r_{k,j}. \tag{25}$$

Adding (23) and (25) together, dividing by 2 and noting that the norm of the residual $r_{k,j}$ can be expressed as

$$\|J_{k,j}\Delta x + \delta_{k,j}\Delta\bar{y}\|^2 := r_{k,j}^T r_{k,j} = \Delta x^T J_{k,j}^T r_{k,j} + \delta_{k,j}\Delta\bar{y}^T r_{k,j}, \tag{26}$$

leads to

$$\begin{aligned}
-\nabla_x\phi_{k,j}^T\Delta x &= \tfrac{1}{2}\Delta x^T M_{k,j}\Delta x + \tfrac{1}{2}\Delta x^T H_{k,j}\Delta x + \tfrac{1}{2}\|\Delta\bar{y}\|^2_{\delta_{k,j}I} - \tfrac{1}{2}\|r_{k,j}\|^2_{\delta_{k,j}^{-1}I} \\
&= \tfrac{1}{2}\Delta x^T M_{k,j}\Delta x + \tfrac{1}{2}\|J_{k,j}^T\Delta\bar{y} + b_{k,j}\|^2_{H_{k,j}^{-1}} + \tfrac{1}{2}\|\Delta\bar{y}\|^2_{\delta_{k,j}I} - \tfrac{1}{2}\|r_{k,j}\|^2_{\delta_{k,j}^{-1}I} \\
&\geq \tfrac{1}{2}\|J_{k,j}^T\Delta\bar{y} + b_{k,j}\|^2_{H_{k,j}^{-1}} + \tfrac{1}{2}\|\Delta\bar{y}\|^2_{\delta_{k,j}I} - \tfrac{1}{2}\|r_{k,j}\|^2_{\delta_{k,j}^{-1}I} \\
&\geq \tfrac{1}{2}\|J_{k,j}^T\Delta\bar{y} + b_{k,j}\|^2_{H_{k,j}^{-1}} + \tfrac{1}{2}\|\Delta\bar{y}\|^2_{\delta_{k,j}I} - \tfrac{1}{2}\|r_{k,j}\|^2_{\delta_{k,j}^{-1}I}.
\end{aligned}$$

Using (22) yields $-\nabla_x\phi_{k,j}^T\Delta x \geq \frac{1}{2}\gamma_j\|\nabla_x\phi_{k,j}\|^2_{H_{k,j}^{-1}} \geq \gamma\|\nabla_x\phi_{k,j}\|^2$, where $\gamma := \frac{1}{2}\inf_j\gamma_j/\sup_j\lambda_{\max}(H_{k,j}) > 0$. □

The second termination test is standard in LSMR.

Termination test 2 *Let $\mu > 0$ and $0 \leq \beta_2 \leq 1$ be given constants. A step $(\Delta x, \Delta\bar{y})$ is an acceptable inexact solution of (19) if*

$$\|r_{k,j}\|_{\delta_{k,j}^{-1}I} \leq \mu\min(1, \delta_{k,j}^{\beta_2})\|b_{k,j}\|_{H_{k,j}^{-1}}. \tag{27}$$

Termination test 2 leads directly to convergence of the inner iterations.

Theorem 3 (Inner iteration, inexact solves) *Suppose that Assumptions 1 and 4 hold, and that Termination tests 1 and 2 are satisfied. Then Algorithm 2 generates a sequence of iterates $x_{k,j}$ such that*

$$\lim_{j\to\infty}\|\nabla_x\phi(x_{k,j}, y_k; \rho_{k,j}, \delta_{k,j})\| = 0.$$

Proof. We use the shorthands $\phi_{k,j} := \phi(x_{k,j}, y_k; \rho_{k,j}, \delta_{k,j})$ and $M_{k,j} := H_{k,j} + \delta_{k,j}^{-1}J_{k,j}^T J_{k,j}$ for conciseness. Assumption 4 implies that there exists $\kappa_H > 0$ such that $\|b_{k,j}\|_{H_{k,j}^{-1}} \leq \kappa_H\|b_{k,j}\|$ and $\kappa_M > 0$ such that $\|M_{k,j}^{-1}\| \leq \kappa_M$ for all j. From (24) and Termination test 2, we have

$$\begin{aligned}
\|\Delta x\| &\leq \|M_{k,j}^{-1}\|\,\|-\nabla_x\phi_{k,j} + \delta_{k,j}^{-1}J_{k,j}^T r_{k,j}\| \\
&\leq \kappa_M\left(\|\nabla_x\phi_{k,j}\| + \delta_{k,j}^{-1}\|J_{k,j}^T r_{k,j}\|\right) \\
&\leq \kappa_M\left(\|\nabla_x\phi_{k,j}\| + \|J_{k,j}^T\|\,\|r_{k,j}\|_{\delta_{k,j}^{-1}}\right) \\
&\leq \kappa_M\left(\|\nabla_x\phi_{k,j}\| + \mu\min(1, \delta_{k,j}^{\beta_2})\sigma_{\max}(J_{k,j})\kappa_H\|\nabla_x\phi_{k,j}\|\right) \\
&\leq \kappa\,\|\nabla_x\phi_{k,j}\|
\end{aligned} \tag{28}$$

with $\kappa := \sup_j\ \kappa_M(1 + \mu\sigma_{\max}(J_{k,j})\kappa_H) > 0$.

Lemma 1 and (28) together imply

$$-\frac{\nabla_x\phi_{k,j}^T\Delta x}{\|\nabla_x\phi_{k,j}\|\,\|\Delta x\|} \geq \frac{\gamma}{\kappa} > 0.$$

At this point, we are in position to apply Zoutendijk's theorem and conclude as in the proof of Theorem 2. □

In view of Theorem 3, it is easier to provide an interpretation of Termination test 1. Firstly, under Assumption 3, the sequence $\{\gamma_j\}$ may be chosen as $\gamma_j := \kappa_1 \delta_{k,j}^{\kappa_2}$ for certain positive constants κ_1 and κ_2. Secondly, the right-hand side of (22) is the objective value of (20). The first term in the left-hand side is the norm of (21), which represents the optimality residual of (20), and which decreases monotonically to zero along the LSMR iterations. The second term in the left-hand side is the norm of (12), which represents the first-order optimality conditions of the minimization of the augmented Lagrangian, and which approaches zero under the assumptions of Theorem 3. The role of the sequence $\{\gamma_j\}$ is to allow sufficient room for satisfaction of (22) even when the value of the right-hand side is small. Note however that the optimal value of the right-hand side can only be zero if $b_{k,j} = 0$, which means that $x_{k,j}$ is first-order stationary for the augmented Lagrangian.

As in Section 4.1.1, Assumption 3 ensures that the second stopping condition of Algorithm 2 is satisfied after a finite number of iterations.

4.2 Convergence of the outer iterations

We now analyze the global convergence of the outer iterations. In this section, we assume that Algorithm 2 succeeds in computing a new iterate w_{k+1} that satisfies (15) each time it is called at Step 4 of Algorithm 1. The following corollary results immediately from Theorems 2 and 3.

Corollary 1 *Assume Algorithm 2 succeeds each time it is called at Step 4 of Algorithm 1. Then Algorithm 1 generates iterates w_k such that*

$$\|F(w_{k+1})\|_* \leq \theta \|F(w_k)\|_* + \epsilon_k.$$

The following result, which is a direct consequence of (Armand et al., 2012, Theorem 3.3), describes the behavior of the sequence of outer iterates.

Theorem 4 (Outer iteration) *Assume Algorithm 2 succeeds each time it is called at Step 4 of Algorithm 1 and that the sequence $\{\epsilon_k\}$ converges to zero. Then Algorithm 1 generates a sequence of iterates w_k such that $\{F(w_k)\}$ converges to zero.*

Proof. Let $\ell := \limsup_{k\to\infty} \|F(w_k)\|_*$. Taking the limit superior in (15) yields

$$\ell \leq \theta\ell + \limsup_{k\to+\infty} \epsilon_k.$$

Because $\lim_{k\to\infty} \epsilon_k = \limsup_{k\to+\infty} \epsilon_k = 0$, we have $\ell \leq \theta\ell$, i.e., $\ell = 0$, which means that $\{F(w_k)\}$ converges to zero. □

5 Local convergence

In this section, we analyze the asymptotic behavior of $\{w_k\}$ under the assumption that it converges to a stationary point satisfying certain assumptions. We establish that the rate of convergence of $\{w_k\}$ is Q-superlinear provided δ_k asymptotically approaches zero sufficiently fast, which is ensured by selecting $\delta_k = \|F(w_k)\|$ in Algorithm 1. This last requirement is a departure from standard augmented Lagrangian methods and ensures the transition to a stabilized SQP method in the local regime.

The analysis in this section broadly follows that of Armand et al. (2012) and Wright (1998). The results differ in two important respects. Firstly, we allow quasi-Newton approximations to second-order derivatives. Secondly, we also allow inexact solutions to the extrapolation linear system (13).

Assume x^* is a stationary point of (1) that satisfies (3). We use $\mathcal{Y}$ to denote the set of Lagrange multipliers associated to x^*, i.e.,

$$\mathcal{Y} = \{y^* \in \mathbb{R}^m \mid (x^*, y^*) \text{ satisfies the KKT conditions (3)}\}.$$

We also use the notation $\mathcal{S} := \{x^*\} \times \mathcal{Y}$.

Because $\nabla_x L(x^*, \cdot)$ is linear, $\mathcal{Y}$ is a closed convex set. It is well known that $\mathcal{Y}$ is a singleton under the assumption that $J(x^*)$ has full row rank and may be unbounded or empty if that assumption fails to hold (Gauvin, 1977).

Our working assumptions for this section are as follows.

Assumption 5 *The sequence $\{w_k\}$ generated by Algorithm 1 converges to $w^* = (x^*, y^*)$ for a certain $y^* \in \mathcal{Y}$.*

Assumption 6 *The functions f and c are twice continuously differentiable with locally Lipschitz second derivatives on $\mathbb{R}^n$.*

Assumption 7 *δ_k is chosen as $\|F(w_k)\|$ at Step 3 of Algorithm 1.*

Assumption 8 *H_k is uniformly bounded, i.e., there exists $\kappa > 0$ such that $\|H_k\| \leq \kappa$ for all $k = 0, 1, \ldots$.*

When (13) is solved inexactly, Assumption 4 from the global regime is sufficient for our purposes. Whether (13) is solved exactly or not, we establish fast local convergence under the following, weaker, assumption.

Assumption 9 *The approximation H_k is sufficiently positive definite on the nullspace of $J(x^*)$ for all sufficiently large k, i.e., there exists $\eta > 0$, such that $z^T H_k z \geq \eta \|z\|^2$ for all $z \in \text{Null}(J(x^*))$.*

Assumption 9 implies that we may set $\rho_k = 0$ for all sufficiently large k.

The following assumption states that H_k is an increasingly accurate approximation of $\nabla^2_{xx} L(x^*, y^*)$ along Δx.

Assumption 10 *There exists $0 < \beta_1 \leq 1$ such that*

$$\|(H_k - \nabla^2_{xx} L(x^*, y^*))\Delta x\| = O(\|\Delta x\|^{1+\beta_1})$$

for all sufficiently large k, where Δx is computed from (13).

Assumption 10 is reminiscent of the Dennis and Moré (1977) condition in unconstrained optimization but is more demanding. Though it is unlikely that any quasi-Newton approximation satisfies Assumption 10 in the strong form given, the analysis below illustrates how the assumption allows us to specify the precise convergence rate of the sequence of iterates. It is more likely that a quasi-Newton approximation satisfies

$$(H_k - \nabla^2_{xx} L(x^*, y^*))\Delta x = o(\Delta x) \tag{29}$$

instead. As we comment at the end of the present section, the local convergence analysis holds under that weaker assumption, except that the exact convergence rate cannot be specified.

The global convergence analysis does not depend on whether, how, or how accurately we solve (13). The local analysis, however, depends on (13) crucially. In this section, we specify how accurate the step computation should be. Note that (13) can be shifted to least-squares form exactly as (19):

$$\begin{bmatrix} H_k & J_k^T \\ J_k & -\delta_k I \end{bmatrix} \begin{bmatrix} \Delta x \\ -\Delta \bar{y} \end{bmatrix} = \begin{bmatrix} b_k \\ 0 \end{bmatrix}, \tag{30}$$

where $\Delta\bar{y} := \Delta y + \delta_{k,j}^{-1} c_{k,j}$ and $b_k = -\nabla_x \phi(x_k, y_k; 0, \delta_k)$. We use Termination test 2 as our stopping condition. We repeat it here without mention of the index j.

Termination test 3 *Let $\mu > 0$ and $0 \leq \beta_2 \leq 1$ be given constants. A step $(\Delta x, \Delta\bar{y})$ computed in an inexact solve of (30) at Step 3 of Algorithm 1 is acceptable if*

$$\|r_k\|_{\delta_k^{-1} I} \leq \mu \min(1, \delta_k^{\beta_2}) \|b_k\|_{H_k^{-1}}. \tag{31}$$

According to Theorem 4, $\delta_k \to 0$ and for all sufficiently large k, it realizes the minimum in (31).

For any $\epsilon > 0$, we define

$$\mathcal{N}(\epsilon) := \{(x, y) \mid \exists \bar{y} \in \mathcal{Y} \ \|(x, y) - (x^*, \bar{y})\| \leq \epsilon\}.$$

We denote by P the projection onto $\mathcal{Y}$, i.e.,

$$P(y) := \arg\min\{\|y - \bar{y}\| \mid \bar{y} \in \mathcal{Y}\},$$

which is well defined because $\mathcal{Y}$ is closed and convex. Finally, the Euclidean distance from (x, y) to $\mathcal{S}$ is denoted

$$\operatorname{dist}((x, y), \mathcal{S}) := \inf\{\|(x, y) - (x^*, \bar{y})\| \mid \bar{y} \in \mathcal{Y}\} = \|(x, y) - (x^*, P(y))\|.$$

For any $w = (x, y) \in \mathcal{N}(\epsilon)$, we use the notation δ to denote $\|F(w)\|$, in accordance with Assumption 7.

Lemma 2 *Suppose that Assumptions 6 and 7 hold. Then there exists a constant $\epsilon > 0$ such that for all $(x, y) \in \mathcal{N}(\epsilon)$ we have $\operatorname{dist}((x, y), \mathcal{S}) = \Omega(\delta)$.*

Proof. Let $\epsilon > 0$ be arbitrary and $(x, y) \in \mathcal{N}(\epsilon)$. It follows from (3) and Assumption 6 that

$$\|\nabla_x L(x, y)\| = \|\nabla_x L(x, y) - \nabla_x L(x^*, P(y))\| = O(\operatorname{dist}((x, y), \mathcal{S})).$$

Similarly,

$$\|c(x)\| = \|c(x) - c(x^*)\| = O(\|x - x^*\|) = O(\operatorname{dist}((x, y), \mathcal{S})).$$

Thus $\delta = O(\operatorname{dist}((x, y), \mathcal{S}))$. □

Wright (1998) establishes the converse of Lemma 2 under the Mangasarian and Fromovitz constraint qualification condition, which, in the case of equality constraints, amounts to the linear independence constraint qualification condition. Izmailov and Solodov (2012) establish a similar result without assuming a constraint qualification but by restricting attention to a neighborhood of (x^*, y^*). We include the proof for completeness. We denote $\mathcal{B}_\epsilon(x^*, y^*)$ the ball centered at (x^*, y^*) of radius $\epsilon > 0$.

Lemma 3 *Suppose that Assumptions 6, 7 and 9 hold. Then there exists $\epsilon > 0$ such that for all $(x, y) \in \mathcal{B}_\epsilon(x^*, y^*)$, we have $\operatorname{dist}((x, y), \mathcal{S}) = O(\delta)$.*

Proof. By contradiction, suppose that for any $\epsilon > 0$, there exists $(x, y) \in \mathcal{B}_\epsilon(x^*, y^*)$ such that $\delta = o(\|x - x^*\|)$ and $\delta = o(\|y - y^*\|)$. By selecting a sequence $\{\epsilon_k\} \to 0$, we determine sequences $\{x_k\} \to x^*$ and $\{y_k\} \to y^*$ such that $\delta_k = o(\|x_k - x^*\|)$ and $\delta_k = o(\|y_k - y^*\|)$.

With the purpose of deriving a contradiction with Assumptions 6 and 9, we use (3) and the fact that $\nabla_x L(x_k, y_k) = O(\delta_k) = o(\|x_k - x^*\|)$ by assumption to deduce

$$\begin{aligned}
\nabla^2_{xx} L(x^*, y^*)(x_k - x^*) &= \nabla_x L(x_k, y^*) - \nabla_x L(x^*, y^*) + o(\|x_k - x^*\|) \\
&= \nabla_x L(x_k, y_k) - J(x_k)^T (y_k - y^*) + o(\|x_k - x^*\|) \\
&= -J(x_k)^T (y_k - y^*) + o(\|x_k - x^*\|) \\
&= -J(x^*)^T (y_k - y^*) - (J(x_k) - J(x^*))^T (y_k - y^*) + o(\|x_k - x^*\|) \\
&= -J(x^*)^T (y_k - y^*) + o(\|x_k - x^*\|). \qquad (32)
\end{aligned}$$

Similarly, our contradiction assumption gives

$$J(x^*)(x_k - x^*) = c(x_k) - c(x^*) + o(\|x_k - x^*\|) = o(\|x_k - x^*\|). \qquad (33)$$

Reducing to a subsequence if necessary, there exists a vector z with $\|z\| = 1$ such that $\{(x_k - x^*)/\|x_k - x^*\|\} \to z$. We take limits in (32) and (33), and obtain

$$\nabla^2_{xx} L(x^*, y^*) z \in \text{Range}(J(x^*)^T) \quad \text{and} \quad z \in \text{Null}(J(x^*)),$$

which contradicts Assumption 9. Thus there exists $\epsilon > 0$ such that for all $(x, y) \in \mathcal{B}_\epsilon(x^*, y^*)$, we have $\|x - x^*\| = O(\delta)$.

Let $(x, y) \in \mathcal{B}_\epsilon(x^*, y^*)$. The linear system $J(x^*)^T (y - \tilde{y}) = \nabla_x L(x^*, y)$ in the unknown $\tilde{y}$ possesses at least the solution y^*, and all solutions $\tilde{y}$ are in $\mathcal{Y}$. In particular, Hoffman's lemma (see, e.g., (Wright, 1997, Lemma A.3)), implies that there exists a solution $\tilde{y} \in \mathcal{Y}$ such that $y - \tilde{y} = O(\|\nabla_x L(x^*, y)\|)$. Thus,

$$\begin{aligned}
\|y - P(y)\| &\le \|y - \tilde{y}\| \\
&= O(\|\nabla_x L(x^*, y)\|) \\
&= O(\|\nabla_x L(x, y)\|) + O(\|\nabla_x L(x, y) - \nabla_x L(x^*, y)\|) \\
&= O(\delta) + O(\|x - x^*\|) \\
&= O(\delta),
\end{aligned}$$

where we used the first part of the proof. Finally, we have $\|x - x^*\| = O(\delta)$ and $\|y - P(y)\| = O(\delta)$, which concludes the proof. □

Lemmas 2 and 3 combine with Assumption 5 to yield the following corollary.

Corollary 2 *Suppose that Assumptions 5 to 7 and 9 hold. Then, for all sufficiently large k,* $\text{dist}((x_k, y_k), \mathcal{S}) = \Theta(\delta_k)$.

Let $\bar{m}$ be the rank of $J(x^*)^T$, with $0 \le \bar{m} \le m$. The singular value decomposition of $J(x^*)^T$ may be written as

$$J(x^*)^T = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}, \qquad (34)$$

where Σ is a diagonal matrix containing the $\bar{m}$ nonzero singular values, U_1 is $n \times \bar{m}$, U_2 is $n \times (n - \bar{m})$, V_1 is $m \times \bar{m}$ and V_2 is $m \times (m - \bar{m})$. Note that $\begin{bmatrix} U_1 & U_2 \end{bmatrix}$ and $\begin{bmatrix} V_1 & V_2 \end{bmatrix}$ are orthogonal and that the columns of U_2 constitute an orthonormal basis for the nullspace of $J(x^*)$. The next result follows (Wright, 1998, Theorem 3.2).

Theorem 5 *Suppose that Assumptions 5 to 9 hold. Suppose that the approximate solution $(\Delta x, -\Delta y)$ of (13) satisfies Termination test 3. Then, for all sufficiently large k,*

$$\Delta x = O(\delta_k) \quad \textit{and} \quad \Delta y = O(\delta_k^{\beta_2}).$$

Proof. If we decompose $\Delta x = U_1 \tilde{x}_{U_1} + U_2 \tilde{x}_{U_2}$ and $\Delta y = V_1 \tilde{y}_{V_1} + V_2 \tilde{y}_{V_2}$, we may rewrite (13) as

$$\begin{bmatrix} U_1^T H_k U_1 & U_1^T H_k U_2 & U_1^T J_k^T V_1 & U_1^T J_k^T V_2 \\ U_2^T H_k U_1 & U_2^T H_k U_2 & U_2^T J_k^T V_1 & U_2^T J_k^T V_2 \\ V_1^T J_k U_1 & V_1^T J_k U_2 & -\delta_k I & 0 \\ V_2^T J_k U_1 & V_2^T J_k U_2 & 0 & -\delta_k I \end{bmatrix} \begin{bmatrix} \tilde{x}_{U_1} \\ \tilde{x}_{U_2} \\ -\tilde{y}_{V_1} \\ -\tilde{y}_{V_2} \end{bmatrix} = - \begin{bmatrix} s_{U_1} \\ s_{U_2} \\ s_{V_1} \\ s_{V_2} \end{bmatrix}, \tag{35}$$

where

$$\begin{bmatrix} s_{U_1} \\ s_{U_2} \\ s_{V_1} \\ s_{V_2} \end{bmatrix} = \begin{bmatrix} U_1^T (g_k - J_k^T y_k) \\ U_2^T (g_K - J_k^T y_k) \\ V_1^T (c_k - r_k) \\ V_2^T (c_k - r_k), \end{bmatrix}$$

and r_k satisfies (31). According to Assumption 6 and Lemma 3, $J_k - J(x^*) = O(\|x_k - x^*\|) = O(\delta_k)$, so that (34) yields $U_1^T J_k^T V_1 = \Sigma + O(\delta_k)$, $U_1^T J_k^T V_2 = O(\delta_k)$, $U_2^T J_k^T V_1 = O(\delta_k)$, and $U_2^T J_k^T V_2 = O(\delta_k)$. We substitute those estimates into (35) and obtain

$$\begin{bmatrix} U_1^T H_k U_1 & U_1^T H_k U_2 & \Sigma + O(\delta_k) & O(\delta_k) \\ U_2^T H_k U_1 & U_2^T H_k U_2 & O(\delta_k) & O(\delta_k) \\ \Sigma + O(\delta_k) & O(\delta_k) & -\delta_k I & 0 \\ O(\delta_k) & O(\delta_k) & 0 & -\delta_k I \end{bmatrix} \begin{bmatrix} \tilde{x}_{U_1} \\ \tilde{x}_{U_2} \\ -\tilde{y}_{V_1} \\ -\tilde{y}_{V_2} \end{bmatrix} = - \begin{bmatrix} s_{U_1} \\ s_{U_2} \\ s_{V_1} \\ s_{V_2} \end{bmatrix}.$$

After eliminating $\tilde{y}_{V_2} = -\delta_k^{-1} s_{V_2} + O(\|\tilde{x}_{U_1}\|) + O(\|\tilde{x}_{U_2}\|)$, there remains

$$(M_k + O(\delta_k)) \begin{bmatrix} -\tilde{y}_{V_1} \\ \tilde{x}_{U_2} \\ \tilde{x}_{U_1} \end{bmatrix} = - \begin{bmatrix} s_{U_1} + O(\|s_{V_2}\|) \\ s_{U_2} + O(\|s_{V_2}\|) \\ s_{V_1} \end{bmatrix}, \tag{36}$$

where

$$M_k := \begin{bmatrix} \Sigma & U_1^T H_k U_2 & U_1^T H_k U_1 \\ 0 & U_2^T H_k U_2 & U_2^T H_k U_1 \\ 0 & 0 & \Sigma \end{bmatrix}.$$

Assumption 8 and the orthogonality of U ensure that the blocks involving H_k are bounded above by κ. Thus, M_k is uniformly bounded. In addition, M_k is uniformly nonsingular because Σ is nonsingular and Assumption 9 ensures that $U_2^T H_k U_2$ is uniformly positive definite for all sufficiently large k. Thus for all sufficiently large k, $M_k + O(\delta_k)$ is also uniformly nonsingular. Then according to (36)

$$\|(\tilde{y}_{V_1}, \tilde{x}_{U_2}, \tilde{x}_{U_1})\| = O(\|(s_{U_1}, s_{U_2}, s_{V_1}, s_{V_2})\|),$$

and

$$\|\tilde{y}_{V_2}\| = O(\delta_k^{-1})\|s_{V_2}\| + O(\|(s_{U_1}, s_{U_2}, s_{V_1}, s_{V_2})\|).$$

From the right-hand side of (36), we have

$$\|(s_{U_1}, s_{U_2})\| = \|g_k - J_k^T y_k\| = \|\nabla_x L(x_k, y_k)\| = O(\delta_k),$$

and Termination test 3 implies that

$$\begin{aligned}\|s_{V_1}\| = \|V_1^T(c_k - r_k)\| \leq \|c_k - r_k\| \leq \|c(x_k) - c(x^*)\| + \|r_k\| \\ = O(\|x_k - x^*\|) + O(\delta_k^{1+\beta_2}) = O(\delta_k).\end{aligned}$$

In addition,

$$s_{V_2} = V_2^T(c_k - r_k) = V_2^T\left(c(x^*) + J(x^*)(x_k - x^*) + O(\|x_k - x^*\|^2) - r_k\right).$$

Because $c(x^*) = 0$ and $V_2^T J(x^*) = 0$, there remains

$$\|s_{V_2}\| = O(\|x_k - x^*\|^2) + \|r_k\| = O(\delta_k^2) + O(\delta_k^{1+\beta_2}) = O(\delta_k^{1+\beta_2}).$$

Therefore, $\tilde{y}_{V_2} = O(\delta_k^{\beta_2})$. We have established that $\Delta x = U_1\tilde{x}_{U_1} + U_2\tilde{x}_{U_2} = O(\delta_k)$ and $\Delta y = V_1\tilde{y}_{V_1} + V_2\tilde{y}_{V_2} = O(\delta_k^{\beta_2})$. □

Note that Theorem 5 holds even if $\beta_2 = 0$ in Termination test 3. However, in order to establish superlinear convergence, we need to be more demanding on the accuracy of the step computation.

Theorem 6 *Suppose that Assumptions 5 to 10 hold. Suppose that the approximate solution $(\Delta x, -\Delta y)$ of (9) satisfies Termination test 3 with $\beta_2 > 0$. Then, for all sufficiently large k,*

$$\delta_k^+ := \delta(x_k^+, y_k^+) = \delta(x_k + \Delta x, y_k + \Delta y) = O(\delta_k^{1+\beta}).$$

where $0 < \beta = \min(\beta_1, \beta_2) \leq 1$.

Proof. A straightforward Taylor expansion and the linearity of $\nabla_x L(x, \cdot)$ yield, for all sufficiently large k,

$$\begin{aligned}\nabla_x L(x_k + \Delta x, y_k + \Delta y) &= \nabla_x L(x_k, y_k) + \nabla_{xx}^2 L(x_k, y_k)\Delta x - J(x_k)^T\Delta y + O(\|\Delta x\|^2) \\ &= \left(\nabla_{xx}^2 L(x_k, y_k) - H_k\right)\Delta x H_k\Delta x + \nabla_x L(x_k, y_k) - J(x_k)^T\Delta y + O(\|\Delta x\|^2) \\ &= \left(\nabla_{xx}^2 L(x^*, y^*) - H_k\right)\Delta x + O(\delta_k^2) \\ &= O(\delta_k^{1+\beta_1}),\end{aligned}$$

where we used the first block equation of (13), Assumption 10 and Theorem 5.

Similarly, for all sufficiently large k,

$$\begin{aligned}c(x_k + \Delta x) &= c(x_k) + J(x_k)\Delta x + O(\|\Delta x\|^2) \\ &= r_k - \delta_k\Delta y + O(\|\Delta x\|^2) \\ &= O(\delta_k^{1+\beta_2}) + O(\delta_k^2) \\ &= O(\delta_k^{1+\beta_2}),\end{aligned}$$

where we used the second block equation of (13), Termination test 3 and Theorem 5. The result holds with $\beta := \min(\beta_1, \beta_2) > 0$. □

The following corollary states that, asymptotically, no inner iterations are performed and thus only the extrapolation step of Algorithm 1 is employed.

Corollary 3 *Suppose that Assumptions 5 to 10 hold. Suppose that the approximate solution $(\Delta x, -\Delta y)$ of (13) satisfies Termination test 3 with $\beta_2 > 0$. Assume that the sequence $\{\epsilon_k\}$ is chosen such that*

$$\epsilon_k = \omega(\delta_k^{1+\beta}),$$

where β is as in Theorem 6. For sufficiently large k, the iterates computed at Step 4 of Algorithm 1 satisfy $w_{k+1} = w_k^+$ and $\delta_k = \|F(w_k)\|$ converges to zero at the rate $1 + \beta$.

Proof. The result follows directly from Theorem 6 and the assumption on ϵ_k. □

Because β_1 may be unknown, it is safe to set $\epsilon_k = \Theta(\delta_k)$ in Corollary 3. A consequence of Lemma 2 and Corollary 3 is that $\{w_k\} \to w^*$ R-superlinearly. The next result establishes Q-superlinear convergence to the set $\mathcal{S}$, which, though weaker than convergence to w^*, results from the fact that $\mathcal{Y}$ may be an unbounded set.

Theorem 7 *Under the assumptions of Corollary 3, the sequence $\{\text{dist}(w_k, \mathcal{S})\}$, where $\{w_k\}$ is generated by Algorithm 1, converges Q-superlinearly to zero with rate $1 + \beta$.*

Proof. The result follows directly from Corollary 2 and Theorem 6. □

A more precise result follows when (9) is solved sufficiently accurately in the sense that $\beta_2 = 1$ in Termination test 3. The next corollary follows (Wright, 1998, Corollary 4.2).

Corollary 4 *Under the assumptions of Corollary 3 with $\beta_2 = 1$, the sequence $\{w_k\}$ generated by Algorithm 1 converges Q-superlinearly to w^* with rate $1 + \beta$, and $\|w_k - w^*\| = \Theta(\delta_k)$.*

Proof. By Theorem 5, there exists a constant $C > 0$ such that $\|(\Delta x_j, \Delta y_j)\| \leq C\delta_j$ for all sufficiently large j. By Theorem 6, $\{\delta_k\} \to 0$ superlinearly, and thus for all $\ell > k$ sufficiently large, we have

$$\|w_k - w_\ell\| \leq \sum_{j=k}^{\ell-1} \|(\Delta x_j, \Delta y_j)\| \leq C \sum_{j=k}^{\ell-1} \delta_j \leq 2C\delta_k.$$

In the limit, we obtain $\|w_k - w^*\| = O(\delta_k)$. Conversely, Lemma 2 yields $\delta_k = O(\text{dist}(w_k, \mathcal{S})) = O(\|w_k - w^*\|)$. □

We close this section by noting that if Hessian approximations are sufficiently accurate in the sense that $\beta_1 = 1$ in Assumption 10 and if (9) is solved sufficiently accurately in the sense that $\beta_2 = 1$ in Termination test 3, Theorem 6 reveals that $\beta = 1$ and quadratic convergence takes place. In particular, such situation occurs if exact second derivatives are used and (9) is solved exactly, e.g., by way of a stable factorization.

It is possible to weaken Assumption 10 and only require $(H_k - \nabla^2_{xx} L(x^*, y^*))\Delta x = o(\|\Delta x\|)$, which is closer to the original Dennis and Moré (1977) condition. In that case, the conclusion of Theorem 6 changes to $\delta_k^+ = o(\delta_k)$. Corollary 3, Theorem 7 and Corollary 4 all remain valid except that the rate of superlinear convergence cannot be specified.

Our requirements on the quality of the Hessian approximation and on the accuracy of the step computation are substantially weaker than those of, e.g., Armand and Omheni (2015), who require exact steps and the stringent bound $H_k - \nabla_{xx} L(x_k, y_k) = O(\delta_k)$. In view of Theorem 5, the latter is akin to requiring exact second derivatives.

6 Implementation and numerical results

In the following section we examine the practical behavior of Algorithms 1 and 2 and specify the details of our implementation. Our implementation is written in Python with the help of the open-source package NLP.py (Arreckx, Orban, and van Omme, 2016), a programming environment for designing numerical optimization methods. A Python implementation of LSMR is available in the PyKrylov package (Orban, 2009), a library of Krylov methods in pure Python.

Initial Lagrange multipliers Given a user-defined starting point x_s, the vector of Lagrange multipliers y_s is obtained as least-square solutions of $\nabla L(x_s, y) = 0$, i.e., by solving the linear system

$$\begin{bmatrix} I & J_s^T \\ J_s & -\zeta I \end{bmatrix} \begin{bmatrix} v \\ -y \end{bmatrix} = - \begin{bmatrix} g_s \\ 0 \end{bmatrix},$$

using MA57 (Duff, 2004) and discarding v or, alternatively, using LSMR to solve

$$\underset{y \in \mathbb{R}^m}{\text{minimize}} \ \tfrac{1}{2}\|J_s^T y - g_s\|^2 + \tfrac{1}{2}\zeta\|y\|^2.$$

In our implementation, ζ is set to 10^{-8}.

Initial point Using the starting point x_s and the least-squares vector of Lagrange multipliers y_s, we compute $(\Delta x_s, \Delta y_s)$ by solving (13) with $\delta = 0$. If $\|F(x_s + \Delta x_s, y_s + \Delta y_s)\| < \|F(x_s, y_s)\|$, we select the improved starting point $(x_0, y_0) := (x_s + \Delta x_s, y_s + \Delta y_s)$. Otherwise (x_0, y_0) is set to (x_s, y_s).

Penalty parameter The initial penalty parameter is set to $\delta_0 = \min\{0.1, \|F(w_0)\|\}$. In Step 3 of Algorithm 1, δ_k is chosen as

$$\delta_k = \max\left\{\min\{\|F(w_k)\|,\ 0.9\delta_{k-1},\ \delta_{k-1}^{1.1}\},\ \delta_{\min}\right\}$$

where $\delta_{\min}$ is a lower bound imposed on the penalty parameter. This rule ensures that δ_k does not deviate much from the value selected by the augmented Lagrangian mechanism in the global regime, and is set to $\|F(w_k)\|$ asymptotically so that Assumption 7 of the local convergence analysis is satisfied.

When w_k^+ does not satisfy (14), a sequence of inner iterations using Algorithm 2 is started from the initial guess $w_{k,0} = w_k$.

Even though global convergence of Algorithm 2 is promoted by way of a Wolfe linesearch, we have found that a simple Armijo linesearch is nearly as effective. In the first Wolfe condition, we use $c_1 = 10^{-4}$. At the end of the inner iterations, the current value of the penalty parameter is returned to the outer iteration and is used as δ_k.

Optimality conditions Optimality is declared when the norm of the optimality conditions at w_k satisfies

$$\|F(w_k)\| < \epsilon_{\text{tol}}\ \|F(w_0)\|$$

where $\epsilon_{\text{tol}} = 10^{-6}$. In (15), we set $\epsilon_k = 10\delta_k$ and $\theta = 0.99$.

Exact system solves We use the linear solver MA57 to solve (13) and (16). The primal regularization parameter ρ_k is updated until a correct inertia is detected according to Wächter and Biegler (2006, Algorithm IC), with the same values for the various constants.

Inexact system solves In Step 3 of Algorithm 1 and Step 3 of Algorithm 2, LSMR is used to solve the preconditioned linear least-squares problem (20). The preconditioner, $H_{k,j}^{-1}$, is obtained by maintaining a limited-memory BFGS approximation of the Hessian of the Lagrangian in inverse form. It is well known that a standard BFGS approximation is ineffective in the presence of constraints because the Hessian of the Lagrangian is typically indefinite at a solution—see, e.g., (Byrd, Tapia, and Zhang, 1992). If $s_k = x_{k+1} - x_k$ and $t_k = \nabla L(x_{k+1}, y_{k+1}) - \nabla L(x_k, y_{k+1})$, the curvature condition $s_k^T t_k > 0$ is unlikely to hold asymptotically, and the pair (s_k, t_k) will be rejected.

Powell (1978) suggests to use a damped BFGS update that compensates for the lack of positive definiteness in the Hessian at the solution. His approach is based on the Hessian of the Lagrangian, instead of its inverse. In the current work, we follow the same idea, but formulated in terms of $B_k = H_k^{-1}$. Let $q_k := \theta_k s_k + (1-\theta_k) B_k t_k$ where $\theta_k \in (0, 1]$ is defined as

$$\theta_k = \begin{cases} 1 & \text{if } s_k^T t_k \geq \eta\, t_k^T B_k t_k \\ (1-\eta)\, t_k^T B_k t_k / (t_k^T B_k t_k - s_k^T t_k) & \text{otherwise,} \end{cases}$$

for some $\eta \in (0, 1)$. In our implementation, we set $\eta := 0.2$. A straightforward derivation similar to that of Powell (1978) shows that if B_k is positive definite, B_{k+1} is also positive definite. Thus starting this damped BFGS approximation by a scaled identity matrix $B_0 = \gamma_0 I$ with $\gamma_0 > 0$ ensures positive definiteness of all subsequent approximations. In addition, only a small number of pairs (s_k, q_k) is stored so as to provide a limited-memory version of the damped BFGS method. Application of this approximation to a vector is performed using the two loop recursion.

Another way of ensuring the positive-definiteness of H_k is to maintain a BFGS approximation of the Hessian of the augmented Lagrangian ϕ (Byrd et al., 1992). This proposal is motivated by the fact that if δ is sufficiently large, the Hessian of $\phi(x, y; \rho, \delta)$ is positive definite for all (x, y) close to an isolated solution. However, in our numerical experiments, results were not as good as with Powell's damped BFGS. A reason for this is that when δ increases, convergence of the BFGS approximation is disrupted. Similar observations were also reported by Gill and Wong (2011).

In Termination test 3, we set $\beta_2 = 0.5$ and $\mu = 0.2$. During the inner iterations, $\gamma_j = 10^{-4}$ for all j in Termination test 1.

6.1 Numerical experiments

We perform preliminary comparative tests between our implementation of Algorithm 1, named RegSQP, AUGLAG (Arreckx, Lambe, Martins, and Orban, 2015) and IPOPT 3.12.1 (Wächter and Biegler, 2006) on a few problems from the CUTEst (Gould, Orban, and Toint, 2015) and COPS (Dolan, Moré, and Munson, 2004) collections. All models are formulated using the AMPL modeling language (Fourer, Gay, and Kernighan, 2003). Problems tested range from 2 to 20,192 variables and from 1 to 10,000 equality constraints, most of them having more than 1,400 variables and 900 constraints. Table 1 summarizes the characteristics of selected test problems: number of variables, constraints, and nonzero elements in the Jacobian and Hessian. The problems of Table 1 were chosen because the default version of IPOPT performs well on them and the parameter values and updates described above show encouraging performance, both in the direct and iterative variants. We are currently trying to identify parameter values and updates that perform well on a larger test set.

IPOPT is a good comparison when using exact second derivatives because in the absence of inequality constraints, it reduces to a filter/linesearch factorization-based SQP method that is similar to RegSQP. An important difference is that IPOPT treats the parameter δ_k as a static ad-hoc constraint regularization parameter in case rank deficiency of the Jacobian is detected. Because IPOPT also offers the possibility to use L-BFGS approximations of the second derivatives, we use it as a basis for comparison although, strictly speaking, it is not factorization free. It does however allow us to compare RegSQP against a typical SQP implementation that uses quasi-Newton Hessian approximations. An additional difference is that IPOPT sets H_k to a (undamped) L-BFGS approximation of the Hessian of the Lagrangian, even though the latter cannot be expected to be positive definite in the limit. When using L-BFGS approximations, a better candidate for comparison is AUGLAG, an ℓ_∞-norm trust-region augmented Lagrangian method similar in spirit to LANCELOT (Conn, Gould, and Toint, 1992), with the difference that trust-region subproblems are solved using our matrix-free implementation of TRON (Lin and Moré, 1998). The main differences with the original implementation of TRON are that the Hessian is only required as an operator, and that the conjugate gradient method used to minimize the model on a face of the trust region does not use a preconditioner. AUGLAG is thus completely factorization free.

Table 1: Characteristics of selected test problems from CUTEst and COPS: number of variables (nvar), constraints (ncon), nonzero elements in the Jacobian (nnz(J)) and in the Hessian (nnz(H))

name	nvar	ncon	nnz(J)	nnz(H)
elec-1	150	50	150	11325
elec-2	300	100	300	45150
elec-3	600	200	600	180300
aug2d	20192	9996	39984	19800
aug3d	3873	1000	6546	2673
aug3dc	3873	1000	6546	3873
bt1	2	1	2	2
dtoc1l	14985	9990	82889	14985
dtoc1na	1485	990	15735	6385
dtoc1nb	1485	990	15735	6385
dtoc1nc	1485	990	15735	6385
eigencco	30	15	125	465
gridnetb	13284	6724	26568	13284
hager1	10000	5000	14999	5001
hager2	10000	5000	14999	14999
hager3	10000	5000	14999	24998
integreq	100	100	10000	100

We do not apply any scaling procedure to the problems. The IPOPT option `nlp_scaling_method` is set to `none`. We also set the IPOPT options `tol`, `dual_inf_tol` and `constr_viol_tol` to 10^{-6}. IPOPT stops as soon as

$$\max\left\{\frac{\|\nabla_x L(x,y)\|_\infty}{s_d},\ \|c(x)\|_\infty\right\} \leq 10^{-6},$$

where $s_d > 0$ is a scaling factor. AUGLAG is set to stop when

$$\|\nabla_x L(x,y))\|_\infty \leq 10^{-6}\|\nabla_x L(x_0,y_0))\|_\infty \quad \text{and} \quad \|c(x)\|_\infty \leq 10^{-6}\|c(x_0)\|_\infty.$$

The maximum number of iterations is set to 3000 for all solvers and a limit of 1 hour of run time is imposed. Tests are conducted on a MacBook Pro 2.4 GHz equipped with a Intel Core i5 processor and 8 Gb of memory running OSX 10.10.5.

As a measure of efficiency of all the algorithms, we use the number of function and gradient calls and either the number of Hessian calls when exact second derivatives are used, or the number of Jacobian vector products when L-BFGS approximations are used. We deliberately choose not to include cpu time comparisons because our algorithm and AUGLAG are implemented using a high-level language (Python) whereas IPOPT is implemented in a compiled language (C++).

We first compare both methods using exact second derivatives. Both IPOPT and RegSQP use MA57 (Duff, 2004). Table 2 shows a comparison in terms of number of objective, gradient and Hessian evaluations. The results show that IPOPT and RegSQP perform similarly on those problems. Note however that RegSQP seems less efficient than IPOPT in terms of function calls and number of Hessian evaluations. These observations are encouraging because IPOPT is a mature implementation that has benefited from years of development.

Next, we compare both methods using L-BFGS approximations. In this case, IPOPT does not use an iterative method to solve augmented systems, but relies on a factorization and on the Sherman-Morrison-Woodbury formula to take the low-rank update into account. All solvers store 6 pairs in the history. Table 3 presents the results of the comparison. As IPOPT does not provide the number of Jacobian-vector products, we estimate it as $\#J$ calls $\times \min\{m,n\}$. RegSQP significantly outperforms IPOPT in terms of number of Jacobian-vector products. We also note that AUGLAG requires fewer Jacobian-vector products than RegSQP but generally requires substantially more function and gradient evaluations.

The analysis of Sections 4 and 5 does not rely on the LICQ. The following two small examples, `HS026` and `HS039`, demonstrate the behavior of our algorithm on problems that do not satisfy the LICQ. One way to create a degenerate problem is to add an additional constraint of the form $c_1(x) = c_1(x)^2$ to the model, where c_1 is the first constraint of the model. The Jacobian is thus rank deficient everywhere.

Table 2: Comparison of IPOPT and RegSQP using exact second derivatives in terms of number of function calls (#f), gradient calls (#g) and Hessian calls (#H))

	IPOPT			RegSQP		
	#f	#g	#H	#f	#g	#H
elec-1	47	42	41	123	76	45
elec-2	262	187	186	304	162	125
elec-3	420	293	292	290	173	115
aug2d	2	2	1	2	2	1
aug3d	3	3	2	2	2	1
aug3dc	2	2	1	2	2	1
bt1	11	8	7	12	10	8
dtoc1l	7	7	6	9	7	6
dtoc1na	7	7	6	10	8	7
dtoc1nb	7	7	6	7	7	6
dtoc1nc	21	16	15	22	15	14
eigencco	13	13	12	21	17	14
gridnetb	2	2	1	2	2	1
hager1	2	2	1	2	2	1
hager2	2	2	1	2	2	1
hager3	2	2	1	2	2	1
integreq	4	4	3	4	4	3

Table 3: Comparison of AUGLAG, IPOPT and RegSQP using L-BFGS approximations in terms of number of function calls (#f), gradient calls (#g) and products with the Jacobian or its transpose (#Jprod))

	AUGLAG			IPOPT			RegSQP		
	#f	#g	#Jprod	#f	#g	#Jprod	#f	#g	#Jprod
elec-1	6862	5154	5154	306	177	8850	375	197	3423
elec-2	1069	766	766	746	434	43400	548	255	4799
elec-3	1783	1241	1241	1832	1055	211000	2672	728	9541
aug2d	11224	8357	8357	2	2	19992	51	36	16243
aug3d	2190	1857	1857	3	3	3000	37	26	1982
aug3dc	488	250	250	2	2	2000	17	13	727
bt1	224	67	67	9	7	14	54	11	69
dtoc1l	938	657	657	129	26	259740	33	24	3123
dtoc1na	801	540	540	24	22	21780	33	23	2883
dtoc1nb	1000	683	683	23	23	22770	43	28	3319
dtoc1nc	1322	902	902	54	44	43560	104	81	7485
eigencco	821	604	604	98	94	1410	139	72	1393
gridnetb	52094	31477	31477	425	103	692572	129	81	50239
hager1	6045	5312	5312	4	4	20000	31	26	7577
hager2	7103	4117	4117	6	6	30000	47	28	7330
hager3	10834	9895	9895	6	6	30000	82	34	14491
integreq	89	64	64	4	4	400	12	10	153

IPOPT regularizes the augmented matrix by adding a nonzero diagonal term $-\delta_c I$ to the $(2,2)$ block when the Jacobian appears to be rank deficient. The parameter δ_c is controlled by the `jacobian_regularization_value` option whose default value is 10^{-8}.

For these two problems, exact second derivatives and the direct solver MA57 were used. Tables 4 and 5 show a comparison between IPOPT and RegSQP in terms of number of calls to the objective, gradient and Hessian when both use exact second derivatives. Only two examples are shown here but they are representative of the behavior of the two solvers on degenerate problems. Although IPOPT and RegSQP behave similarly on nondegenerate problems, the situation is different on degenerate problems. The performance of RegSQP does not degrade too much in the presence of degeneracy, while degeneracy noticeably impairs IPOPT's performance.

Table 4: HS026: $n = 3$, $m = 1$

	IPOPT		RegSQP	
	original	degenerate	original	degenerate
# objective evals	26	267	17	54
# gradient evals	26	55	18	40
# Hessian evals	25	54	17	39

Table 5: HS039: $n = 4$, $m = 2$

	IPOPT		RegSQP	
	original	degenerate	original	degenerate
# objective evals	14	114	12	17
# gradient evals	14	51	13	18
# Hessian evals	13	50	12	17

7 Discussion

The main contribution of this paper is the formulation and analysis of an algorithm for equality-constrained optimization that combines the favorable global properties of augmented Lagrangian methods and local properties of stabilized SQP methods. The use of positive-definite limited-memory approximations to the Hessian of the Lagrangian presents the significant advantage that the linear system encountered at each iteration is always SQD. An appropriate interpretation of that system in terms of a linear least-squares problem permits efficient inexact system solves and an entirely factorization-free implementation. The numerical results of Section 6 indicate that the proposed method is robust and efficient, even when the problem is degenerate.

The use of our BFGS-LSMR strategy for solving (13) or (16) inexactly is not restricted to the specific scope of the present research. It could also be employed when applying a quadratic penalty method to (1) in the same vein as described by Armand, Benoist, Omheni, and Pateloup (2014). Following the procedure of Section 2 leads to the fully regularized subproblem

$$\underset{x\in\mathbb{R}^n, r\in\mathbb{R}^m}{\text{minimize}} \;\; f(x) + \tfrac{1}{2}\rho_k\|x - x_k\|^2 + \tfrac{1}{2}\delta_k\|r\|^2 \quad \text{subject to} \;\; c(x) + \delta_k r = 0, \tag{37}$$

for $\delta_k > 0$, $\rho_k \geq 0$ and for some new variables r. Applying Newton's method to the KKT conditions of (37) yields

$$\begin{bmatrix} H_k + \rho_k I & J_k^T \\ J_k & -\delta_k I \end{bmatrix} \begin{bmatrix} \Delta x \\ -\Delta y \end{bmatrix} = - \begin{bmatrix} g_k - J_k^T y_k \\ c_k + \delta_k y_k \end{bmatrix}. \tag{38}$$

Note that the coefficient of (38) is the same as that of (16), only the right-hand side differs.

The local convergence properties of Section 5 assume that quasi-Newton approximations converge superlinearly to the exact Hessian at the solution along the primal steps. Byrd et al. (1992) establish that fast local convergence of a SQP method can take place provided H_k is defined as the BFGS approximation of the Hessian of the *augmented* Lagrangian. It may be possible to transpose their results to the present context because (9) is precisely a step on an augmented Lagrangian. Unfortunately, the performance of the augmented Lagrangian approximation is poor compared to that of the damped BFGS update of Powell (1978). Local convergence properties of the damped update remains an open question. Powell proves that if convergence occurs, it does so at a R-superlinear rate. Further exploration of those considerations is beyond the scope of the present paper and is left for future research.

Our method can be extended to problems with inequality constraints. A first approach could be to use an augmented Lagrangian function that takes inequalities into account, such as that of Birgin and Martínez (2014). Another approach is to reformulate the optimization problem by adding slack variables and treat bounds via a logarithmic barrier. The precise form of the Newton equations used is likely to be of crucial importance in practice when designing the factorization-free variant (Greif, Moulding, and Orban, 2014).

Finally, allowing the penalty parameter to increase during the inner iterations as proposed in Armand and Omheni (2015) might further improve performance.

References

M. Arioli and D. Orban. Iterative methods for symmetric quasi-definite linear systems—part I: Theory. Cahier du GERAD G-2013-32, GERAD, Montréal, QC, Canada, 2013.

P. Armand and R. Omheni. A globally and quadratically convergent primal-dual augmented Lagrangian algorithm for equality constrained optimization. Optimization Methods and Software, 2015. doi: 10.1080/10556788.2015.1025401. Online First.

P. Armand, J. Benoist, and D. Orban. From global to local convergence of interior methods for nonlinear optimization. Optimization Methods and Software, 28(5):1051–1080, 2012. doi: 10.1080/10556788.2012.668905.

P. Armand, J. Benoist, R. Omheni, and V. Pateloup. Study of a primal-dual algorithm for equality constrained minimization. Computational Optimization and Applications, 59(3):405–433, July 2014. doi: 10.1007/s10589-014-9679-3.

S. Arreckx, A. Lambe, J. R. R. A. Martins, and D. Orban. A matrix-free augmented Lagrangian algorithm with application to large-scale structural design optimization. Optimization and Engineering, pages 1–26, October 2015. doi: 10.1007/s11081-015-9287-9.

S. Arreckx, D. Orban, and N. van Omme. NLP.py — a large-scale optimization toolkit in Python. Cahier du GERAD G-2016-42, GERAD, Montréal, QC, Canada, 2016.

D. P. Bertsekas. Constrained optimization and Lagrange multiplier methods. Academic press, 1996. doi: 10.1016/b978-0-12-093480-5.50005-2.

E. G. Birgin and J. M. Martínez. Practical augmented Lagrangian methods for constrained optimization. SIAM, May 2014. doi: 10.1137/1.9781611973365.

P. T. Boggs and J. W. Tolle. Sequential quadratic programming. Acta Numerica, 4:1–51, January 1995. doi: 10.1017/S0962492900002518.

R. H. Byrd, R. A. Tapia, and Y. Zhang. An SQP augmented lagrangian BFGS algorithm for constrained optimization. SIAM Journal on Optimization, 2(2):210–241, 1992.

R. H. Byrd, J. Nocedal, and R. B. Schnabel. Representations of quasi-Newton matrices and their use in limited memory methods. Mathematical Programming, 63(1):129–156, 1994. doi: 10.1007/BF01582063.

R. H. Byrd, F. E. Curtis, and J. Nocedal. An inexact Newton method for nonconvex equality constrained optimization. Mathematical Programming, 122(2):273–299, September 2009. doi: 10.1007/s10107-008-0248-3.

A. R. Conn, N. I. M. Gould, and Ph. L. Toint. LANCELOT: A Fortran Package for Large-Scale Nonlinear Optimization (Release A). Springer Publishing Company, Incorporated, first edition, 1992.

J. E. Dennis and Jorge J. Moré. Quasi-Newton methods, motivation and theory. SIAM Review, 19(1):46–89, January 1977. doi: 10.1137/1019005.

J. E. Jr Dennis and R. B. Schnabel. Numerical methods for unconstrained optimization and nonlinear equations, volume 16. SIAM, January 1996. doi: 10.1137/1.9781611971200.

E. D. Dolan, J. J. Moré, and T. S. Munson. Benchmarking Optimization Software with COPS 3.0. Technical Report ANL/MCS-TM-273, Argonne National Laboratory, Mathematics and Computer Science division, 2004.

I. S. Duff. MA57—a code for the solution of sparse symmetric definite and indefinite systems. ACM Transactions on Mathematical Software, 30(2):118–144, June 2004. doi: 10.1145/992200.992202.

D. Fernàndez and M. Solodov. Stabilized sequential quadratic programming for optimization and a stabilized Newton-type method for variational problems. Mathematical Programming, 125(1):47–73, September 2010. doi: 10.1007/s10107-008-0255-4.

D. Fernández, E. A. Pilotta, and G. A. Torres. An inexact restoration strategy for the globalization of the sSQP method. Computational Optimization and Applications, 54(3):595–617, 2012. doi: 10.1007/s10589-012-9502-y.

D. C.-L. Fong and M. A. Saunders. LSMR: An iterative algorithm for sparse least-squares problems. SIAM Journal on Scientific Computing, 33(5):2950–2971, 2011. doi: 10.1137/10079687X.

R. Fourer, D. M. Gay, and B. W. Kernighan. AMPL: A Modeling Language for Mathematical Programming. Duxbury Press / Brooks/Cole Publishing Company, second edition, 2003.

M. P. Friedlander and D. Orban. A primal-dual regularized interior-point method for convex quadratic programs. Mathematical Programming Computation, 4(1):71–107, 2012. doi: s12532-012-0035-2.

J. Gauvin. A necessary and sufficient regularity condition to have bounded multipliers in nonconvex programming. Mathematical Programming, Series B, 12:136–138, 1977.

P. Gill and D. Robinson. A globally convergent stabilized SQP method. SIAM Journal on Optimization, 23(4): 1983–2010, January 2013. doi: 10.1137/120882913.

P. E. Gill and E. Wong. Sequential quadratic programming methods. In Jon Lee and Sven Leyffer, editors, Mixed Integer Nonlinear Programming, volume 154 of The IMA Volumes in Mathematics and its Applications, pages 147–224. Springer New York, nov 2011. doi: 10.1007/978-1-4614-1927-3_6.

N. I. M. Gould. On practical conditions for the existence and uniqueness of solutions to the general equality quadratic programming problem. Mathematical Programming, 32(1):90–99, May 1985. doi: 10.1007/BF01585660.

N. I. M. Gould, D. Orban, and Ph. L. Toint. CUTEst: a Constrained and Unconstrained Testing Environment with safe threads. Computational Optimization and Applications, 60(3):545–557, 2015. doi: 10.1007/s10589-014-9687-3.

C. Greif, E. Moulding, and D. Orban. Bounds on the eigenvalues of matrices arising from interior-point methods. SIAM Journal on Optimization, 24(1):49–83, 2014. doi: 10.1137/120890600.

W. W. Hager. Stabilized sequential quadratic programming. Computational Optimization and Applications, 12(1): 253–273, 1999. doi: 10.1023/A:1008640419184.

M. Heinkenschloss and D. Ridzal. A matrix-free trust-region sqp method for equality constrained optimization. SIAM Journal on Optimization, 24(3):1507–1541, 2014.

A. F. Izmailov and M. V. Solodov. Stabilized SQP Revisited. Mathematical Programming, 133(1–2):93–120, June 2012. doi: 10.1007/s10107-010-0413-3.

A. F. Izmailov, M. V. Solodov, and E. I. Uskov. Combining stabilized SQP with the augmented Lagrangian algorithm. Computational Optimization and Applications, 62(2):405–429, 2015. doi: 10.1007/s10589-015-9744-6.

C.-J. Lin and J. J. Moré. Newton's method for large bound-constrained optimization problems. SIAM Journal on Optimization, 9:1100–1127, 1998. doi: 10.1137/S1052623498345075.

D. C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. Mathematical Programming, 45(1):503–528, August 1989. doi: 10.1007/BF01589116.

J. Nocedal and S. J. Wright. Numerical Optimization. Springer, New York, second edition, 2006. doi: 10.1007/b98874.

D. Orban. PyKrylov: Krylov subspace methods in pure Python. github.com/PythonOptimizers/pykrylov, July 2009.

C. Paige and M. A. Saunders. Solution of sparse indefinite systems of linear equations. SIAM Journal on Numerical Analysis, 12(4):617–629, September 1975. doi: 10.1137/0712047.

C. C. Paige and M. A. Saunders. LSQR: An algorithm for sparse linear equations and sparse least squares. ACM Transactions on Mathematical Software, 8(1):43–71, 1982. doi: 10.1145/355984.355989.

M. J. D. Powell. A fast algorithm for nonlinearly constrained optimization calculations. In Lecture Notes in Mathematics, pages 144–157. Springer Science + Business Media, 1978. doi: 10.1007/bfb0067703.

R. T. Rockafellar. Augmented Lagrangians and applications of the proximal point algorithm in convex programming. Mathematics of Operations Research, 1:97–116, 1976. doi: 10.1287/moor.1.2.97.

R. J. Vanderbei. Symmetric quasi-definite matrices. SIAM Journal on Optimization, 5(1):100–113, 1995.

A. Wächter and L. T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Mathematical Programming, 106:25–57, 2006. doi: 10.1007/s10107-004-0559-y.

R. B. Wilson. A Simplicial Method for Convex Programming. PhD thesis, Harvard University, Boston, USA, 1963.

S. J. Wright. Primal-Dual Interior-Point Methods. SIAM, 1997. doi: 10.1137/1.9781611971453.

S. J. Wright. Superlinear Convergence of a Stabilized SQP Method to a Degenerate Solution. Computational Optimization and Applications, 11(3):253–275, 1998. doi: 10.1023/A:1018665102534.

S. J. Wright. An algorithm for degenerate nonlinear programming with rapid local convergence. SIAM Journal on Optimization, 15(3):673–696, 2005. doi: 10.1137/030601235.