Study of a primal-dual algorithm for equality constrained minimization

Paul Armand · Joël Benoist · Riadh Omheni · Vincent Pateloup

February 4, 2014

Abstract The paper proposes a primal-dual algorithm for solving an equality constrained minimization problem. The algorithm is a Newton-like method applied to a sequence of perturbed optimality systems that follow naturally from the quadratic penalty approach. This work is first motivated by the fact that a primal-dual formulation of the quadratic penalty provides a better framework than the standard primal form. This is highlighted by strong convergence properties proved under standard assumptions. In particular, it is shown that the usual requirement of solving the penalty problem with a precision of the same size as the perturbation parameter, can be replaced by a much less stringent criterion, while guaranteeing the superlinear convergence property. A second motivation is that the method provides an appropriate regularization for degenerate problems with a rank deficient Jacobian of constraints. The numerical experiments clearly bear this out. Another important feature of our algorithm is that the penalty parameter is allowed to vary during the inner iterations, while it is usually kept constant. This alleviates the numerical problem due to ill-conditioning of the quadratic penalty, leading to an improvement of the numerical performances.

Keywords Nonlinear programming · Constrained optimization · Equality constraints · Primal-dual method · Quadratic penalty method

Mathematics Subject Classification (2000) 49M15 · 49M37 · 65K05 · 90C06 · 90C30 · 90C51

P. Armand
Université de Limoges - Laboratoire XLIM (France)
E-mail: paul.armand@unilim.fr

J. Benoist
Université de Limoges - Laboratoire XLIM (France)
E-mail: paul.armand@unilim.fr

R. Omheni
Université de Limoges - Laboratoire XLIM (France)
E-mail: riadh.omheni@unilim.fr

V. Pateloup
Université de Limoges - Laboratoire XLIM (France)
E-mail: vincent.pateloup@unilim.fr

1 Introduction

This paper proposes and analyzes a primal-dual algorithm for solving an equality constrained problem of the form

$$\text{minimize } f(x) \quad \text{subject to} \quad g(x) = 0, \tag{1}$$

where the functions $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}^m$ are smooth. The algorithm is basically a Newton-like method applied to a sequence of perturbed optimality systems. The globalization is done by introducing a real control of the iterates in both primal and the dual spaces.

This study is deliberately limited to the numerical solution of an optimization problem with only equality constraints, although we have in mind that our approach could be extended to problems with inequality constraints, for example by introducing and penalizing slack variables into a log-barrier term. Our main motivation is to inquire on the effective potential of a primal-dual approach as an alternative to the classical sequential quadratic programming method. This is why the quite simple framework of equality constrained programming seems to be well suited.

The paper is motivated by two main observations on a primal-dual approach. The first is that for the quadratic penalty formulation it leads to a better-behaved method than the traditional primal form. This is supported by strong global and asymptotic convergence properties that will be demonstrated under standard assumptions. The second is that it emphasizes an appropriate regularization for degenerate problems with a rank deficient Jacobian of constraints. This will be highlighted by the numerical experiments.

1.1 General algorithm scheme

To introduce the primal-dual framework, let us consider at first the quadratic penalty problem associated with (1):

$$\text{minimize } f(x) + \frac{1}{2\mu}\|g(x)\|^2, \tag{2}$$

where $\mu > 0$ is a penalty parameter and $\|\cdot\|$ is the usual Euclidean norm. The first order optimality conditions are $\nabla f(x) + \frac{1}{\mu}A(x)^\top g(x) = 0$, where $A(x) \in \mathbb{R}^{m\times n}$ is the Jacobian matrix of g at x. These conditions can be rewritten under the form

$$F(w, \mu) = 0, \tag{3}$$

where the mapping $F : \mathbb{R}^{n+m+1} \to \mathbb{R}^{n+m}$ is defined by

$$F(w, \mu) = \begin{pmatrix} \nabla f(x) + A(x)^\top y \\ g(x) - \mu y \end{pmatrix},$$

with $w := (x, y) \in \mathbb{R}^{n+m}$. The variable y then appears as a vector of Lagrange multipliers, because $F(w, 0) = 0$ corresponds to the first order optimality conditions of the initial problem (1).

Let $x^* \in \mathbb{R}^n$ denote an optimal solution of (1). Under standard regularity assumptions at x^*, there exists $y^* \in \mathbb{R}^m$ such that $w^* := (x^*, y^*)$ is a solution of

(3) with $\mu = 0$. The equation (3) implicitly defines a trajectory $\mu \mapsto \mathbf{w}(\mu)$ such that $\mathbf{w}(0) = w^*$ and $F(\mathbf{w}(\mu), \mu) = 0$ for sufficiently small values of μ.

The basic idea is to apply a Newton-like method to equation (3) for some sequence $\{\mu_k\}$ converging to zero. The linearization of (3) at the current iterate (w_k, μ_k), with respect to the pair (w, μ), provides the linear system

$$J_k\,(w_k^+ - w_k) + \frac{\partial F}{\partial \mu}(w_k, \mu_k)(\mu_k^+ - \mu_k) = -F(w_k, \mu_k),$$

where J_k is the Jacobian matrix of the function $F(\cdot, \mu_k)$ at w_k, or an approximation to it, where w_k^+ is the Newton iterate and μ_k^+ is a new value of the penalty parameter, chosen from the current value μ_k by some updating procedure. This linear system can be rewritten under the simplified form:

$$J_k\,(w_k^+ - w_k) = -F(w_k, \mu_k^+). \tag{4}$$

It is worth noting that the updated value of the penalty parameter appears here in the right-hand side of the system.

The global convergence of the Newton-like iteration (4) is done as follows. At first, if the norm of the residual $F(w_k^+, \mu_k^+)$ is deemed sufficiently small, then $\mu_{k+1} = \mu_k^+$ and $w_{k+1} = w_k^+$. Otherwise, a sequence of *inner iterations* is applied to decrease that residual. The goal of this procedure is to generate a sequence $\{\mu_k\}$ converging to zero and a sequence $\{w_k\}$ such that for all index k we have

$$\|F(w_{k+1}, \mu_{k+1})\| \le \varepsilon_k,$$

for some tolerance $\varepsilon_k > 0$. The choice of this tolerance value is very sensitive for the algorithm efficiency and it will be done dynamically, in the sense that its value will depend on w_k and μ_k. At this level, we emphasize that our assumption made on ε_k to get the superlinear convergence, is much weaker than the classical requirement that can be found in the literature and which is usually of the form $\varepsilon_k = \tau\mu_{k+1}$, for some constant $\tau > 0$. Indeed, we propose here a quite different choice for ε_k. It is based on a formula of the form

$$\varepsilon_k = \rho\|F(w_k, \mu_k)\| + \zeta_k,$$

where $\rho \in (0, 1)$ is a given constant and where $\{\zeta_k\}$ is a positive sequence converging to zero. This globally convergent scheme has been already used in the framework of interior methods in [3], where it is shown that this avoids unnecessary calls to the inner iterations algorithm and also shortens the length of the inner iteration sequences, which improves the algorithm efficiency.

The inner iterations are Newton-like iterations applied to Equation (3), but with the same value of μ in the matrix and on the right-hand side, corresponding to a linearization of (3) with respect to w only. The Newton direction is the solution of a system of the form

$$Jd = -F(w, \mu), \tag{5}$$

where J is the Jacobian of $F(\cdot, \mu)$ at w, or an approximation to it. The global convergence of the inner iterations is performed by applying a simple backtracking line search algorithm. The control of the iterates is done in both primal and dual spaces during the whole minimization process. The merit function thus depends on the variables x and y. It is of the form

$$\varphi_\mu(w) = f(x) + \frac{1}{2\mu}\|g(x)\|^2 + \frac{\nu}{2\mu}\|g(x) - \mu y\|^2, \tag{6}$$

where $\nu > 0$ is a scaling parameter to balance the quadratic penalty and the primal-dual term. To simplify the notation, since this parameter is fixed during the inner iterations, the dependence of the merit function to ν is not further specified.

In a primal-dual method, the penalty parameter is usually kept constant during the inner iterations. We adopt here a different strategy, with a penalty parameter which can increase during inner iterations. We will show that this strategy leads to shortening the length of the inner iterations sequence, while guaranteeing the global convergence.

The detailed algorithm is presented in Section 2 and its global convergence properties are analyzed in Section 3. The asymptotic behavior of the sequence of iterates $\{w_k\}$ is analyzed in Section 4. It is shown that whenever the sequence $\{w_k\}$ converges to a regular solution w^*, if $\{\mu_k\}$ converges to zero with a subquadratic rate of convergence, then the algorithm reduces asymptotically only to Newton iterations of the form (4), without the need for inner iterations, and $\{w_k\}$ follows the trajectory $\mathbf{w}$ in a tangential manner. The latter implies in particular that $\{w_k\}$ and $\{\mu_k\}$ have the same rate of convergence. By means of some numerical experiments, it will be shown in Section 5 that the primal-dual method is as efficient as the sequential quadratic programming (SQP) algorithm and is much more robust when solving degenerate problems for which the Jacobian of constraints is rank deficient during the minimization process.

1.2 Related work and motivation

The quadratic penalty method is one of the oldest ideas for handling equality constraints. A first proposal can be found in a paper of Courant in 1943 [17] and a theoretical analysis can be found in the book of Fiacco and McCormick [24]. The method has long been shunned by practitioners because of the increasing ill-conditioning of the Hessian matrix of the penalty function when the penalty parameter approaches its limit [38]. Solutions for dealing with this ill-conditioning were introduced by Broyden and Attia [9,10], and Gould [35,36]. Augmented Lagrangian methods were also proposed as another alternative. See the book of Conn, Gould and Toint [16, Ch. 14] for a thorough discussion on these issues.

At the present time, the sequential quadratic programming (SQP) approach is undoubtedly amongst the most effective method for nonlinear constrained minimization, see, e.g., [7, Ch. 6], [16, Ch. 15], [25, Ch. 12] or [40, Ch. 18]. It is worth noting that the SQP method is essentially a primal approach, in the sense that the globalization is applied in the space of primal variables, whatever the strategy, line search, trust region or filter technique.

With the interior point revolution [49], there has been a revival of penalty methods in nonlinear optimization and particularly through their primal-dual interpretation, see the survey papers by Forsgren, Gill and Wright [27], Gould, Orban and Toint [34], and Nemirovski and Todd [39]. The primal-dual interior method forms the background material of state-of-the-art softwares for nonlinear optimization, such as LOQO [43, 41, 5, 37], KNITRO [11, 13, 14, 48] and IPOPT [45–47]. These methods are basically the Newton method applied to a primal-dual formulation of the first order optimality conditions of the barrier problem, see for example [7, Ch. 6] or [40, Ch. 19].

Though they possess some good numerical performances, Newton-barrier methods sometimes exhibit some global convergence difficulties. Wächter and Biegler [44] have shown that, linearizing the constraints and maintaining the strict feasibility may lead some algorithms to converge to spurious solutions, see also [12]. In contrast, as recently shown by Chen and Goldfarb [15], penalty-barrier methods do not seem to suffer of these convergence issues. The idea is fairly old and goes back to [24]. It consists of penalizing the inequality constraints into a logarithmic barrier term and the equality constraints into a quadratic or ℓ_1 term. Algorithms based on a penalty-barrier approach have been studied by Armand [1], Chen and Goldfarb [15], Benchakroun, Dussault and Mansouri [4], Forsgren and Gill [26], Gertz and Gill [29], Goldfarb *et al.* [31], Gill and Robinson [30], Gould, Orban and Toint [33], Tits *et al.* [42], and Yamashita and Yabe [50]. The purpose of our paper is to investigate the convergence properties and to evaluate the numerical performances of such a primal-dual approach, in order to compare them against a standard SQP algorithm. The simple framework of equality constraints seems well suited for this comparison. As recently emphasized by Gill and Robinson [30], whose study is also focused on equality constraints, the benefits of a primal-dual approach are to offer a control of the dual variables during the whole minimization process and also to provide a natural regularization of the method.

1.3 Notation

For two vectors x and y in $\mathbb{R}^p$, their Euclidean scalar product is denoted by $x^\top y$ and the associated norm is $\|x\| = (x^\top x)^{1/2}$. For x in $\mathbb{R}^p$, the open ball of radius r and centered at x is denoted $B(x, r)$. The induced matrix norm is defined for all $M \in \mathbb{R}^{q\times p}$ by $\|M\| = \max\{\|Md\| : \|d\| \le 1\}$.

The inertia of a real symmetric matrix M, denoted by $\mathrm{In}(M)$, is the integer triple $(\lambda_+, \lambda_-, \lambda_0)$ giving the number of its positive, negative and null eigenvalues.

Let $\{a_k\}$ and $\{b_k\}$ be two nonnegative scalar sequences. We will use the standard Landau notation $a_k = \mathrm{O}(b_k)$ to mean that there exists a constant $C > 0$, such that $a_k \le C b_k$ for $k \in \mathbb{N}$ large enough. We will use $a_k = \mathrm{o}(b_k)$ to mean that there exists a sequence $\{\epsilon_k\}$ converging to zero, such that $a_k = \epsilon_k b_k$ for $k \in \mathbb{N}$ large enough. We use similar symbols with vector arguments, in which case they are understood normwise.

Related to Problem (1), we denote by $\nabla f(x) \in \mathbb{R}^n$ the gradient of f at x. The Jacobian matrix of g at x is denoted by $A(x)$. It is an $m \times n$ matrix, whose ith row is the vector $\nabla g_i(x)^\top$. For $w := (x, y) \in \mathbb{R}^{n+m}$, the Lagrangian function is denoted by $\mathcal{L}(w) = f(x) + y^\top g(x)$. The gradient of the Lagrangian at $w \in \mathbb{R}^{n+m}$ is the vector $\nabla_x \mathcal{L}(w) = \nabla f(x) + A(x)^\top y$ and the Hessian of the Lagrangian at w is the matrix $\nabla^2_{xx}\mathcal{L}(w) = \nabla^2 f(x) + \sum_{i=1}^m y_i \nabla^2 g_i(x)$.

2 Algorithm description

The algorithm uses two kinds of iterations. At an *outer* iteration the value of the penalty parameter is reduced and the solution of the linear system (4), called also *extrapolation* step, is computed. The aim is to get a sufficiently small value of the residual norm $\|F(w,\mu)\|$. In the case where this value is deemed insufficient, a sequence of *inner* iterations, which are typically minimization steps, is applied to get a sufficient reduction of the residual norm.

2.1 Coefficient matrix of the linear systems

Let $w = (x, y)$ be the current iterate. Whether it is an outer or an inner iteration, each step of the algorithm requires the solution of a linear system whose coefficient matrix is of the form

$$J = \begin{pmatrix} H & A(x)^\top \\ A(x) & -\mu I \end{pmatrix}, \tag{7}$$

where H is the Hessian of the Lagrangian at w, or an approximation to it, I is the identity matrix and $\mu > 0$.

It is well-known that $\mathrm{In}(J) = (n, m, 0)$ if and only if the matrix

$$K = H + \frac{1}{\mu} A(x)^\top A(x) \tag{8}$$

is positive definite (see, e.g., [26, Lemma 4.1]). In this case, not only the matrix J is nonsingular, what makes the solution of the corresponding linear system well defined, but the solution of (5) is a descent direction for some merit function.

From Debreu theorem [19], we know that if H is positive definite on the null space of $A(x)$, with a matrix $A(x)$ of full row rank and a sufficiently small positive scalar μ, then K is positive definite. This is precisely the case when H is the Hessian of the Lagrangian evaluated at a point near a regular solution of (1) and μ is small enough. But far from an optimal solution, this matrix can be indefinite. To overcome this difficulty, a common regularization technique consists of adding multiples of the identity matrix to the top-left block until the factorization reveals a correct inertia. This is what we did in our numerical experiments. This issue is further detailed in Section 5.

It is worth noting that the inertia correction is applied to each iteration, including the outer iterations, although they do not require the descent property because a merit function is not used at this level. This strategy of course helps to get a regular system, but also to prevent convergence to a stationary point that would not be a minimum.

2.2 Outer iteration

We describe now the outer iterations to solve (3). Choose the constants $\sigma \in (0, 1)$, $a \in (0, 1)$, $b \geq a$ and $c > 0$. Then choose a positive sequence $\{s_k\}$ that tends to zero and such that

$$\limsup \frac{s_k^{1+\sigma}}{s_{k+1}} < 1. \tag{9}$$

The purpose of the sequence $\{s_k\}$ is solely to force the convergence of $\{\mu_k\}$ to zero. At the beginning, the algorithm starts with a point $w_0 \in \mathbb{R}^{n+m}$, an initial value $0 < \mu_0 \le s_0$ and the iteration index is initialized to $k = 0$.

Algorithm 1 (Outer iteration)

Step 1. *Choose* μ_k^+ *such that*

$$\min\{\mu_k^{1+\sigma}, a\mu_k, s_k\} \le \mu_k^+ \le \min\{b\mu_k, s_k\}. \tag{10}$$

Step 2. *Choose a symmetric matrix* H_k *such that* $\mathrm{In}(J_k) = (n, m, 0)$*, where*

$$J_k = \begin{pmatrix} H_k & A(x_k)^\top \\ A(x_k) & -\mu_k I \end{pmatrix}.$$

Step 3. *Compute the Newton iterate* w_k^+ *by solving the linear system*

$$J_k(w_k^+ - w_k) = -F(w_k, \mu_k^+). \tag{11}$$

Step 4. *Choose* ε_k *such that*

$$\varepsilon_k \ge c\mu_k^+. \tag{12}$$

Step 5. *If* $\|F(w_k^+, \mu_k^+)\| \le \varepsilon_k$*, then set* $w_{k+1} = w_k^+$ *and* $\mu_{k+1} = \mu_k^+$*. Otherwise, apply the inner iteration algorithm to find* w_{k+1} *and* $\mu_{k+1} \in [\mu_k^+, s_k]$ *such that*

$$\|F(w_{k+1}, \mu_{k+1})\| \le \varepsilon_k. \tag{13}$$

The choice of μ_k^+ at Step 1 and the fact that $\mu_{k+1} \in [\mu_k^+, s_k]$ at Step 5, imply that $\{\mu_k\}$ goes to zero with a rate of convergence at most superlinear with order $1+\sigma$ (see, e.g., [6]). Indeed, from (9), we have for k large enough, $\mu_{k+1}^{1+\sigma} \le s_k^{1+\sigma} \le s_{k+1}$. It follows that, for sufficiently large k, the minimum on the left of (10) is achieved by $\mu_k^{1+\sigma}$. A critical consequence of this choice is that the sequence of iterates $\{w_k\}$ becomes asymptotically tangent to the trajectory $\mathbf{w}$, meaning that $w_k = \mathbf{w}(\mu_k) + \mathrm{o}(\mu_k)$. This property and the choice of the tolerance ε_k at Step 4, imply that the algorithm asymptotically reduces to full Newton steps. These properties are analyzed in Section 4 and stated in Theorem 3, one of the main results of the paper.

The choice of the stopping tolerance ε_k in (13) is also quite important for the algorithm efficiency. Our selected value has been guided by the following considerations. The extrapolation step $w_k^+ - w_k$ follows from the linearization of the function F at (w_k, μ_k) with respect to the pair (w, μ). Whenever $J_k = F'_w(w_k, \mu_k)$, the directional derivative of $\|F\|$ at (w_k, μ_k) in the direction $(w_k^+ - w_k, \mu_k^+ - \mu_k)$ is equal to $-\|F(w_k, \mu_k)\|$. This follows from a property of the Newton method and is valid for any norm. A natural choice for the tolerance at Step 1 could be to set $\varepsilon_k = \rho\|F(w_k, \mu_k)\|$, for some constant $\rho \in (0, 1)$. But this choice would be much too restrictive in the case where the current iterate w_k is near to the trajectory $\mathbf{w}$, which would imply that $\|F(w_k, \mu_k)\|$ is nearly zero and would require that the next iterate also to stay near to $\mathbf{w}$, an unnecessary and costly requirement, especially when w_k is far from the solution. We then propose to introduce a relaxation parameter $\zeta_k > 0$, leading to a tolerance of the form $\varepsilon_k = \rho\|F(w_k, \mu_k)\| + \zeta_k$. Such a choice has already been successfully proposed in the framework of interior methods [3]. Moreover, by choosing a relaxation parameter ζ_k greater than a constant times μ_k^+, the inequality (12) will be satisfied. Finally, a condition for a non-monotonic decrease of the residual norm is also introduced to further relax the stopping criterion. All these considerations lead us to the choice

$$\varepsilon_k = \rho \max\{\|F(w_i, \mu_i)\| : \ell_k \leq i \leq k\} + \zeta_k, \tag{14}$$

for all $k \in \mathbb{N}$, where $\rho \in (0,1)$, $\ell_k := \max\{0, k - \ell + 1\}$ and for a predefined integer $\ell \geq 1$. The following result shows that our choice is relevant to guarantee the convergence of $\{F(w_k, \mu_k)\}$ to zero.

Proposition 1 *Assume that Algorithm 1 generates an infinite sequence $\{w_k\}$, where ε_k is defined by (14), with $\rho \in (0,1)$, $\ell \geq 1$ and a bounded positive sequence $\{\zeta_k\}$ such that for all $k \in \mathbb{N}$, $\zeta_k \geq c\mu_k^+$. Then, we have*

$$\limsup \|F(w_k, \mu_k)\| \leq \frac{1}{1-\rho} \limsup \zeta_k.$$

In particular, if $\{\zeta_k\}$ goes to zero, then $\{F(w_k, \mu_k)\}$ converges to zero.

Proof For $k \in \mathbb{N}$, formula (13) reads as follows:

$$r_{k+1} \leq \rho \max\{r_i : \ell_k \leq i \leq k\} + \zeta_k, \tag{15}$$

where $r_k := \|F(w_k, \mu_k)\|$ denotes the kth residual.

Define $\bar{\zeta} := \sup\{\zeta_k : k \in \mathbb{N}\}$ and $r := \max\{r_0, \bar{\zeta}/(1-\rho)\}$. Let us show by induction on $k \in \mathbb{N}$ that $r_k \leq r$. The base case is clearly true. For a given $k \in \mathbb{N}$, assume that $r_i \leq r$ for all $0 \leq i \leq k$. Then, from (15) we have $r_{k+1} \leq \rho r + \bar{\zeta}$, and thus $r_{k+1} \leq r$. Consequently, we have proved that the limit superior of $\{r_k\}$ is finite. Taking now the limit superior in inequality (15), we get

$$\limsup r_k \leq \rho \limsup r_k + \limsup \zeta_k,$$

which ends the proof. □

In our implementation, we take the infinity norm in (14) and a relaxation parameter of the form

$$\zeta_k = \theta \mu_k, \tag{16}$$

for some constant $\theta > 0$. By the choice of μ_k^+ in (10), the inequality (12) is satisfied with $c = \theta/b$.

2.3 Inner iteration

During the inner iterations, the control of the iterates is performed in both the primal and the dual space thanks to the primal-dual quadratic penalty function (6). This merit function is frequently used in a primal-dual framework, see for example [26,30,50]. It is easy to see that w is a stationary point of φ_μ if and only if it is a solution of (3). Therefore, the realization of the condition (13) can be done by applying a minimization procedure to the function $\varphi_{\mu_{k+1}}$.

Whenever the penalty parameter μ is small, the minimization problem of φ_μ becomes ill-conditioned, it may then be necessary to apply a long sequence of inner iterations to get a sufficient reduction of this function and thus to satisfy the stopping criterion (13). To remedy this situation, we propose to increase the value of the penalty parameter along the inner iterations, while guaranteeing their global convergence. The idea is to possibly update the value of μ by taking the minimum of the convex function $\mu \mapsto \frac{1}{\mu}\|g(x) - \mu y\|^2$. If this minimum, namely $\hat{\mu} = \|g(x)\|/\|y\|$, is greater than the current value of μ, then we set μ to $\hat{\mu}$. In this manner, we have $\varphi_{\hat{\mu}}(w) \leq \varphi_\mu(w)$ and thus the global convergence is safeguarded.

Let k be the current value of the index of the outer iteration calling the inner iteration algorithm. To simplify the notation used in the description and the analysis of the inner iteration algorithm, we omit the outer iteration index k to denote the different sequences generated by the algorithm. The inner iteration index is set as a superscript, so that the ith inner iterate is denoted by $w^i := (x^i, y^i)$.

The inner iteration index is set to $i = 0$ and an initial inner iterate w^0 is chosen. A fixed value $\nu > 0$ is chosen for the scaling parameter of the merit function (6). To prevent the sequence of penalty parameters from becoming unbounded, a maximum admissible value $\bar{\mu} \in [\mu_k^+, s_k]$ is initially defined. Then an initial value for the barrier parameter $\mu^0 \in [\mu_k^+, \bar{\mu}]$ is chosen.

During the numerical experiments, we found that a good choice was to set $w^0 = w_k^+$, $\mu^0 = \mu_k^+$ and $\bar{\mu} = s_k := 1/(k+1)$.

Two constants $\omega \in (0,1)$ and $\kappa \in (0, \frac{1}{2}]$ are chosen. Typical values are $\omega = 0.01$ and $\kappa = 0.1$.

Algorithm 2 (Inner iteration)

Step 1. *For $i \geq 1$. If $y^i \neq 0$, then compute $\hat{\mu}^i = \|g(x^i)\|/\|y^i\|$, otherwise set $\hat{\mu}^i = +\infty$. Set*

$$\mu^i = \begin{cases} \max\{\hat{\mu}^i, \mu^{i-1}\} & \text{if } \hat{\mu}^i \leq \bar{\mu}, \\ \mu^{i-1} & \text{otherwise.} \end{cases}$$

Step 2. *Choose a symmetric matrix H^i such that $\mathrm{In}(J^i) = (n, m, 0)$, where*

$$J^i = \begin{pmatrix} H^i & A(x^i)^\top \\ A(x^i) & -\mu^i I \end{pmatrix}.$$

Step 3. *Compute the direction d^i by solving the linear system*

$$J^i d = -F(w^i, \mu^i). \tag{17}$$

Step 4. *Set $\alpha^i = 1$. While the condition*

$$\varphi_{\mu^i}(w^i + \alpha^i d^i) \leq \varphi_{\mu^i}(w^i) + \omega\alpha^i \nabla\varphi_{\mu^i}(w^i)^\top d^i \tag{18}$$

is not satisfied, choose a new trial step length $\alpha^i \in [\kappa\alpha^i, (1-\kappa)\alpha^i]$.

Step 5. *Set $w^{i+1} = w^i + \alpha^i d^i$.*

When the algorithm is called from the outer iteration algorithm, it should stop with a pair (w^{i+1}, μ^i) satisfying $\|F(w^{i+1}, \mu^i)\| \leq \varepsilon_k$, where ε_k is the tolerance chosen when the sequence of inner iterations was initiated. Therefore, (w^{i+1}, μ^i) is assigned to the next pair (w_{k+1}, μ_{k+1}).

The inertia control of J^i at Step 2 ensures the descent property, that is $\nabla\varphi_{\mu^i}(w^i)^\top d^i < 0$. In particular, the number of backtracking trial steps is finite at Step 4.

The following proposition shows that the sequence of the merit function values at w^i is decreasing during the inner iterations, even if the penalty parameter value is modified.

Proposition 2 *For all inner iterations $i \geq 1$, we have*

$$\mu^{i-1} \leq \mu^i \leq \bar{\mu} \quad \textit{and} \quad \varphi_{\mu^i}(w^i) \leq \varphi_{\mu^{i-1}}(w^i) < \varphi_{\mu^{i-1}}(w^{i-1}).$$

Proof Let $i \geq 1$. We have $\mu^i = \hat{\mu}^i$ if and only if $\mu^{i-1} \leq \hat{\mu}^i \leq \bar{\mu}$, otherwise we have $\mu^i = \mu^{i-1}$, which proves the first part.

The last inequality follows from the line search condition (18). It remains to prove the penultimate inequality. If $\mu^i = \mu^{i-1}$, then the inequality is trivially satisfied. Suppose now that $\mu^i = \hat{\mu}^i$. Since $\mu^i \geq \mu^{i-1}$ and $\hat{\mu}^i$ is the minimum of the function $\mu \mapsto \frac{1}{\mu}\|g(x^i) - \mu\, y^i\|^2$ on $(0, +\infty)$, we have

$$\begin{aligned}
\varphi_{\mu^i}(w^i) &= f(x^i) + \frac{1}{2\mu^i}\|g(x^i)\|^2 + \frac{\nu}{2\hat{\mu}^i}\|g(x^i) - \hat{\mu}^i y^i\|^2 \\
&\leq f(x^i) + \frac{1}{2\mu^{i-1}}\|g(x^i)\|^2 + \frac{\nu}{2\mu^{i-1}}\|g(x^i) - \mu^{i-1} y^i\|^2 \\
&= \varphi_{\mu^{i-1}}(w^i).
\end{aligned}$$

□

Figures 1 and 2 compare the evolution of the penalty parameter and of the residual norm $\|F(w, 0)\|$ for two problems, for the cases where the penalty parameter is constant or increasing during the inner iterations. Such behavior has been also observed for interior methods, see [2]. Two problems are considered, orthrega and eigenbco, see Table 2[1].

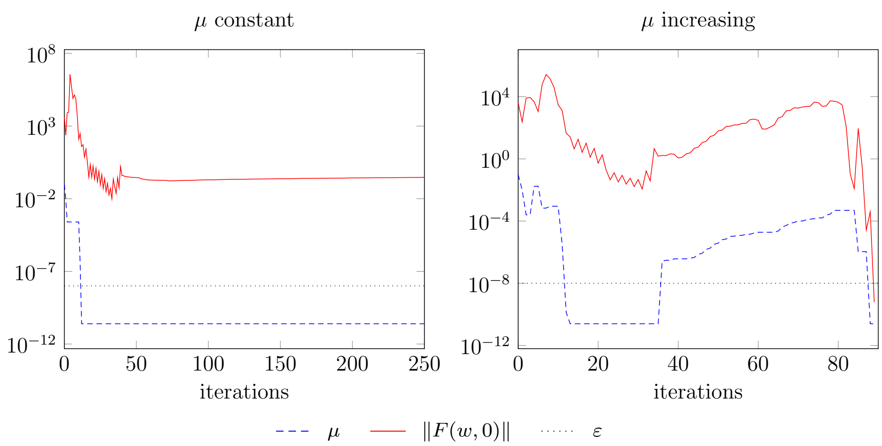

Fig. 1: orthrega

For a strategy that left μ constant during the inner iterations, we can see on the left figures that, for both problems, μ decreases very quickly right from the beginning, then remains constant during a very long sequence of inner iterations. Only 250 iterations are shown, but in fact for orthrega the convergence occurs

[1] The tables appear in the online supplement of the paper.

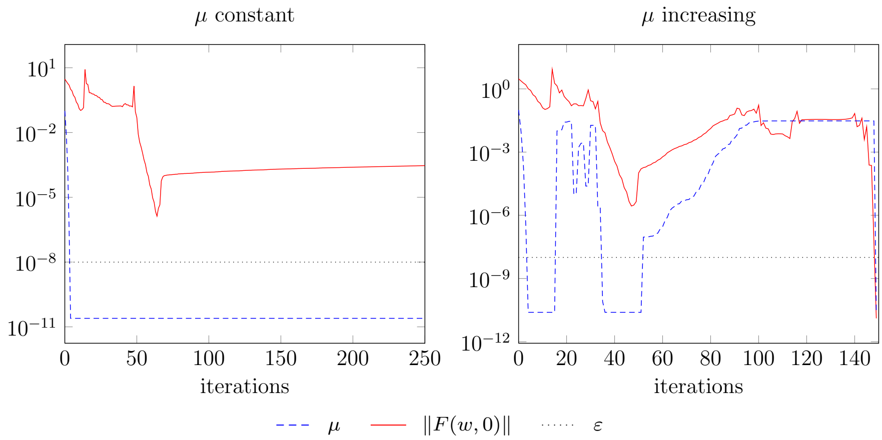

Fig. 2: eigenbco

after 57265 iterations, of which 31 are outer ones and 57234 are inner ones, while for eigenbco it finally comes after 706608 iterations, of which 37 are outer ones and 706571 are inner ones. By using the increasing procedure in Step 1 of Algorithm 2, for both problems the number of iterations is up to more reasonable values.

2.4 Descent direction

The following proposition gives a formula for the directional derivative of the merit function along the Newton direction and a sufficient condition to get a descent direction.

Proposition 3 *Let $w := (x, y) \in \mathbb{R}^{n+m}$, $\mu > 0$, $\delta \geq 0$ and $d := (d_x, d_y)$ be a solution of the linear system $Jd = -F(w, \mu)$, where J is defined by (7). We have the equality*

$$\nabla \varphi_\mu(w)^\top d = -d_x^\top K d_x - \frac{\nu}{\mu} \|A(x)d_x - \mu d_y\|^2. \tag{19}$$

In addition, if K is positive definite and if $F(w, \mu)$ is nonzero, then d is a descent direction of the merit function at w, i.e., $\nabla \varphi_\mu(w)^\top d < 0$.

Proof Using (5) we have

$$Hd_x + A(x)^\top (y + d_y) + \nabla f(x) = 0 \quad \text{and} \quad A(x)d_x + g(x) - \mu(y + d_y) = 0.$$

By using these linear equations, we get

$$\begin{aligned}
&\nabla \varphi_\mu(w)^\top d \\
&= \nabla f(x)^\top d_x + \frac{1}{\mu} g(x)^\top A(x)d_x + \frac{\nu}{\mu}(g(x) - \mu y)^\top (A(x)d_x - \mu d_y) \\
&= -d_x^\top H d_x + \frac{1}{\mu}(g(x) - \mu(y + d_y))^\top A(x)d_x - \frac{\nu}{\mu}\|A(x)d_x - \mu d_y\|^2,
\end{aligned}$$

from which (19) follows.

Given the positive definiteness assumption, it is clear that $\nabla\varphi_\mu(w)^\top d$ is non-positive. It is equal to zero if and only if $d_x = 0$ and $A(x)d_x - \mu d_y = 0$, and thus $d = 0$, which implies $F(w, \mu) = 0$. □

3 Global convergence analysis

This section is devoted to the global convergence of the algorithm. An analysis of the sequence of inner iterates generated by Algorithm 2 is first performed. The convergence analysis of Algorithm 1 is then established.

3.1 Inner iterations

Let $\{w^i\}$ and $\{\mu^i\}$ be the sequences generated by Algorithm 2. The convergence analysis of the inner iterations is carried out under the following three assumptions.

Assumption 1 *The functions f and g are continuously differentiable over $\mathbb{R}^n$.*

It is not assumed that the functions are twice continuously differentiable. This way, the analysis encompasses the case where $H(w^i)$ is a quasi-Newton approximation of the Hessian of the Lagrangian. Note also that the Lipschitz continuity of the gradients is not required, which allows to address a wide class of problems.

Assumption 2 *The sequences $\{A(x^i)\}$ and $\{H^i\}$ are bounded.*

This assumption is the most restrictive, but is rather standard in a global convergence analysis of a primal-dual method, see for example [15, 29, 50]. The boundedness of the Jacobian matrices can be fulfilled by assuming that the sequence $\{x^i\}$ remains in a compact set, but here we prefer to include in our analysis the case, not uncommon in practice, of an unbounded sequence of iterates.

Assumption 3 *The matrices K^i are uniformly positive definite for $i \in \mathbb{N}$, i.e., there exists some constant $\lambda > 0$ such that for all $i \in \mathbb{N}$ and all $u \in \mathbb{R}^n$,*

$$u^\top K^i u \geq \lambda \|u\|^2.$$

This last assumption can be satisfied by means of the inertia correction strategy used in the numerical experiments.

We can now state the global convergence result of the inner iteration sequence.

Theorem 1 *Under Assumption 1-3, at least one of the following situations occurs.*

- *(i) Algorithm 2 converges in the sense that* $\liminf \|F(w^i, \mu^i)\| = 0$.
- *(ii) The sequence of objective function values $\{f(x^i)\}$ goes to $-\infty$.*

Proof The reasoning is by contradiction. Suppose that none of the situations (i) and (ii) occur.

In view of (7), the boundedness of the sequence $\{J^i\}$ follows straightforwardly from Assumption 2 and from the fact that the sequence $\{\mu^i\}$ is bounded above by

$\bar{\mu}$. In view of (17) and since $\liminf \|F(w^i, \mu^i)\| > 0$, we also have $\liminf \|d^i\| > 0$. Then, there exists $\varepsilon > 0$ such that for all $i \in \mathbb{N}$

$$\|d^i\| \geq \varepsilon. \tag{20}$$

The remainder of the proof is divided into three parts.

Part 1. It is first proved that there exists $\bar{\lambda} > 0$ such that for all $i \in \mathbb{N}$,

$$-\nabla \varphi_{\mu^i}(w^i)^\top d^i \geq \bar{\lambda} \|d^i\|^2. \tag{21}$$

Suppose that the conclusion fails. Then for all $k \geq 1$, there exists $i_k \in \mathbb{N}$ such that

$$-\nabla \varphi_{\mu^{i_k}}(w^{i_k})^\top d^{i_k} < \frac{1}{k} \|d^{i_k}\|^2.$$

In particular, we have $d^{i_k} \neq 0$ for all $k \geq 1$. According to Proposition 3 and Assumption 3, we deduce that for all $k \geq 1$

$$\lambda \|u_x^k\|^2 + \frac{\nu}{\mu^{i_k}} \|A(x^{i_k}) u_x^k - \mu^{i_k} u_y^k\|^2 < \frac{1}{k},$$

where $u^k := \frac{d^{i_k}}{\|d^{i_k}\|}$ is a unit vector. By taking if necessary a subsequence, we may assume that the sequence $\{u^k\}$ converges to a unit vector $u := (u_x, u_y)$. Taking the limit $k \to \infty$ in the last inequality, we obtain

$$u_x = 0 \quad \text{and} \quad \lim_{k \to \infty} \frac{1}{\mu^{i_k}} \|A(x^{i_k}) u_x^k - \mu^{i_k} u_y^k\|^2 = 0.$$

Using the fact that the sequence $\{\mu^{i_k}\}$ is lower-bounded by μ^0 and that $\{A(x^{i_k})\}$ is bounded, we also have $u_y = 0$, and thus $u = 0$, which contradicts $\|u\| = 1$. The desired assertion is then proved.

Part 2. It is now proved that the sequence $\{w^i\}$ converges to a point denoted $\bar{w} := (\bar{x}, \bar{y})$ and that the sequence $\{\alpha^i\}$ converges to zero.

Let $i \in \mathbb{N}$. The sufficient decrease condition (18) can be rewritten as follows

$$-\omega \alpha^i \nabla \varphi_{\mu^i}(w^i)^\top d^i \leq \varphi_{\mu^i}(w^i) - \varphi_{\mu^i}(w^{i+1}).$$

From Proposition 2, we also have $\varphi_{\mu^{i+1}}(w^{i+1}) \leq \varphi_{\mu^i}(w^{i+1})$, implying that

$$-\omega \alpha^i \nabla \varphi_{\mu^i}(w^i)^\top d^i \leq \varphi_{\mu^i}(w^i) - \varphi_{\mu^{i+1}}(w^{i+1}).$$

Using inequalities (20) and (21) and recalling that $w^{i+1} - w^i = \alpha^i d^i$, we have

$$\varepsilon \bar{\lambda} \omega \|w^{i+1} - w^i\| \leq \varphi_{\mu^i}(w^i) - \varphi_{\mu^{i+1}}(w^{i+1}).$$

Adding these inequalities for i from 0 to an arbitrary integer p, we obtain

$$\varepsilon \bar{\lambda} \omega \sum_{i=0}^{p} \|w^{i+1} - w^i\| \leq \varphi_{\mu^0}(w^0) - f(x^{p+1}),$$

since by definition of φ we have $\varphi_{\mu^{p+1}}(w^{p+1}) \geq f(x^{p+1})$. Recalling that the situation (ii) fails, we conclude that the series $\sum (w^{i+1} - w^i)$ is absolutely convergent. It follows that the sequence $\{w^i\}$ converges to a point $\bar{w}$. Moreover, the inequality $\|w^{i+1} - w^i\| = \alpha^i \|d^i\| \geq \alpha^i \varepsilon$ allows to prove that the sequence $\{\alpha^i\}$ converges to zero.

Part 3. Finally, the proof is ended by highlighting the contradiction.

Since the sequence $\{\alpha^i\}$ goes to zero, for i large enough $\alpha^i < 1$. For such an index i, there is at least one backtracking step, meaning that there is a trial step length $\bar{\alpha}^i > 0$ such that $\alpha^i \in [\kappa\bar{\alpha}^i, (1-\kappa)\bar{\alpha}^i]$, for which the sufficient decrease condition (18) is not satisfied, that is

$$\varphi_{\mu^i}(\bar{w}^i) > \varphi_{\mu^i}(w^i) + \omega\bar{\alpha}^i \nabla\varphi_{\mu^i}(w^i)^\top d^i, \tag{22}$$

where $\bar{w}^i := w^i + \bar{\alpha}^i d^i$. For i large, we have $\|w^{i+1} - w^i\| = \alpha^i \|d^i\| \geq \kappa\|\bar{\alpha}^i d^i\|$, it follows that the sequence $\{\bar{\alpha}^i d^i\}$ converges to zero and consequently the sequence $\{\bar{w}^i\}$ converges to $\bar{w}$.

Let i large enough such that (22) occurs. By virtue of the mean value theorem there exists a point $\widetilde{w}^i$ belonging to the segment $[w^i, \bar{w}^i]$ such that

$$\varphi_{\mu^i}(\bar{w}^i) - \varphi_{\mu^i}(w^i) = \bar{\alpha}^i \nabla\varphi_{\mu^i}(\widetilde{w}^i)^\top d^i.$$

By substitution in (22), we then get

$$\nabla\varphi_{\mu^i}(\widetilde{w}^i)^\top d^i - \nabla\varphi_{\mu^i}(w^i)^\top d^i > -(1-\omega)\nabla\varphi_{\mu^i}(w^i)^\top d^i.$$

Applying the Cauchy-Schwarz inequality and using (21), we obtain

$$\|\nabla\varphi_{\mu^i}(\widetilde{w}^i) - \nabla\varphi_{\mu^i}(w^i)\| > (1-\omega)\bar{\lambda}\|d^i\|.$$

Define $h_1(w) := f(x) - \nu y^\top g(x)$, $h_2(w) := \frac{\nu}{2}\|y\|^2$ and $h_3(w) := \frac{1+\nu}{2}\|g(x)\|^2$ for $w = (x, y) \in \mathbb{R}^{n+m}$. Then according to the definition of the merit function (6), we have $\varphi_{\mu^i} = h_1 + \mu^i h_2 + \frac{1}{\mu^i} h_3$. Using the above inequality, the triangle inequality and recalling that $\mu^i \in [\mu^0, \bar{\mu}]$, we also have

$$(1 + \bar{\mu} + \frac{1}{\mu^0}) \max\Big\{\|\nabla h_k(\widetilde{w}^i) - \nabla h_k(w^i)\| : k \in \{1,2,3\}\Big\} > (1-\omega)\bar{\lambda}\|d^i\|.$$

By continuity of the three functions ∇h_1, ∇h_2 and ∇h_3 at $\bar{w}$, the left hand side of the inequality goes to zero when i goes to infinity. Thus the sequence $\{d^i\}$ converges to zero, which is in contradiction with (20) and concludes the proof. □

3.2 Overall algorithm

We now return to the overall algorithm and analyze the global convergence of the outer iterations.

In this section, it is assumed that Algorithm 2 succeeds each time it is called at Step 5 to compute a pair (w_{k+1}, μ_{k+1}) that satisfies (13). This is a reasonable hypothesis, because Theorem 1 shows that if f is bounded below and if Assumptions 1–3 hold at each iteration k, then the inner iterations algorithm succeeds in finding a point that satisfies (13). Under this assumption, Algorithm 1 generates an infinite sequence $\{w_k\}$.

The following result provides more insights on the behavior of the sequence of outer iterates with respect to the primal and dual feasibilities.

Theorem 2 *Let $\{w_k\}$ be a sequence generated by Algorithm 1. If the sequence $\{\varepsilon_k\}$ tends to zero, then $\{F(w_k, \mu_k)\}$ goes to zero, in particular*

$$\lim \nabla_x \mathcal{L}(w_k) = 0.$$

Moreover, if it is assumed that the sequence $\{(\nabla f(x_k), A(x_k))\}$ is bounded, then the primal iterates approach stationarity of the measure of infeasibility, which means that

$$\lim A(x_k)^\top g(x_k) = 0.$$

In addition, one of following outcomes occurs.

(*i*) *The sequence $\{y_k\}$ is unbounded. In this situation, the iterates approach failure of the linear independence constraint qualification, which means that there exists a sequence of unit vectors $\{u_k\}$ such that*

$$\liminf \|A(x_k)^\top u_k\| = 0.$$

(*ii*) *The sequence $\{y_k\}$ is bounded. In this case, the sequence of iterates is asymptotically feasible for the problem* (1)*, meaning that*

$$\lim g(x_k) = 0.$$

Proof The first assertion follows easily from Step 5 of Algorithm 1.

Assume now that $\{(\nabla f(x_k), A(x_k))\}$ is bounded. For all $k \in \mathbb{N}$, we have

$$A(x_k)^\top g(x_k) = \mu_k(\nabla f(x_k) + A(x_k)^\top y_k) - \mu_k \nabla f(x_k) + A(x_k)^\top (g(x_k) - \mu_k y_k).$$

By taking the norm on both sides, we have

$$\|A(x_k)^\top g(x_k)\| \leq \mu_k \|\nabla_x \mathcal{L}(w_k)\| + \mu_k \|\nabla f(x_k)\| + \|A(x_k)\| \|g(x_k) - \mu_k y_k\|.$$

We then get $\lim \|A(x_k)^\top g(x_k)\| = 0$.

Suppose now that $\{y_k\}$ is unbounded. By taking if necessary a subsequence, we can assume that $\lim \|y_k\| = \infty$ and $y_k \neq 0$ for all $k \in \mathbb{N}$. Let us define $u_k := \frac{y_k}{\|y_k\|}$ for $k \in \mathbb{N}$. We have for all $k \in \mathbb{N}$

$$\|A(x_k)^\top u_k\| \leq \frac{1}{\|y_k\|}(\|\nabla_x \mathcal{L}(w_k)\| + \|\nabla f(x_k)\|),$$

from which the outcome (i) follows.

At last, if $\{y_k\}$ is bounded, then $\|g(x_k)\| \leq \|g(x_k) - \mu_k y_k\| + \mu_k \|y_k\|$, from which we deduce that $\lim \|g(x_k)\| = 0$ and conclude the proof. □

4 Asymptotic analysis

In this section, it is assumed that Algorithm 1 generates a convergent sequence $\{w_k\}$ to a primal-dual solution $w^* := (x^*, y^*) \in \mathbb{R}^{n+m}$ of the initial problem (1).

We will analyze the asymptotic behavior of $\{w_k\}$. We first state the assumptions and establish some basic results. We then show that the sequence $\{w_k\}$ is asymptotically tangent to the trajectory $\mathbf{w}$ and the algorithm eventually reduces to only a sequence of full Newton steps. As a consequence, the rate of convergence of $\{w_k\}$ to w^* is the same as the one of $\{\mu_k\}$ to zero.

4.1 Assumptions and basic results

It is assumed that the following standard assumptions are satisfied, that is, smoothness, linear independence constraint qualification and second order optimality conditions.

Assumption 4 *The functions f and g are twice continuously differentiable and their second derivatives are Lipschitz continuous over an open neighborhood of x^*.*

Assumption 5 *The Jacobian matrix $A(x^*)$ is of full row rank.*

Assumptions 4 and 5 imply that y^* is the unique vector of Lagrange multipliers such that $F(w^*, 0) = 0$.

Assumption 6 *The strong second order conditions hold at w^*, i.e., for all $u \in \mathbb{R}^n$, if $u \neq 0$ and $A(x^*)u = 0$, then $u^\top \nabla^2_{xx}\mathcal{L}(w^*)u > 0$.*

Under Assumptions 4–6, the Jacobian F'_w is Lipschitzian and nonsingular in a neighborhood of $(w^*, 0)$. As a consequence, by applying the implicit function theorem, the trajectory $\mathbf{w}$ exists and is Lipschitzian in a neighborhood of zero. All these properties are well known (see, e.g., [24]) and are summarized in the following lemma.

Lemma 1 *Under Assumptions 4–6, there exist positive constants r^*, μ^*, L, M, and C, as well as a continuously differentiable function $\mathbf{w} : (-\mu^*, \mu^*) \to \mathbb{R}^{n+m}$ such that for all $w, w' \in B(w^*, r^*)$ and $\mu, \mu' \in (-\mu^*, \mu^*)$, we have*

(i) $\|F'_w(w, \mu) - F'_w(w', \mu')\| \leq L(\|w - w'\| + |\mu - \mu'|)$,
(ii) $\|F'_w(w, \mu)^{-1}\| \leq M$,
(iii) $F(w, \mu) = 0$ *if and only if* $\mathbf{w}(\mu) = w$,
(iv) $\|\mathbf{w}(\mu) - \mathbf{w}(\mu')\| \leq C|\mu - \mu'|$.

Our asymptotic analysis requires that J_k is nearly equal to $F'_w(w_k, \mu_k)$ in a neighborhood of the solution.

Assumption 7 *There exists $\beta > 0$ such that for all $k \in \mathbb{N}$*

$$\|\nabla^2_{xx}\mathcal{L}(w_k) - H_k\| \leq \beta\mu_k.$$

When using $H_k = \nabla^2_{xx}\mathcal{L}(w_k) + \delta_k I$, with some regularization parameter $\delta_k \geq 0$, as we did in our experiments, this assumption is satisfied near a strong minimizer. Indeed, as we discussed in Subsection 2.1, under Assumptions 5 and 6, if w_k is near w^* and μ_k small enough, then δ_k can be set to zero so that $\mathrm{In}(J_k) = (n, m, 0)$.

The following lemma gives an estimate of the distance of the Newton iterate to the trajectory $\mathbf{w}$.

Lemma 2 *Assume that Assumptions 4–7 hold. The sequence of iterates generated by Algorithm 1 satisfies*

$$\|w_k^+ - \mathbf{w}(\mu_k^+)\| = \mathrm{O}(\|w_k - \mathbf{w}(\mu_k)\|^2 + \mu_k^2).$$

Proof There exists $k_0 \in \mathbb{N}$ such that for all $k \geq k_0$, Lemma 1 applies, that is $w_k \in B(w^*, r^*)$ and $\mu_k \leq \min\{\mu^*/b, \mu^*\}$.

Let us first show that for k large enough,

$$\|J_k^{-1}\| \leq 2M \tag{23}$$

where M is defined in Lemma 1. Let $k \geq k_0$. Define $A_k = F'_w(w_k, \mu_k)$ and $B_k = J_k$. From outcome (ii) of Lemma 1 and Assumption 7, we have

$$\begin{aligned}\|A_k^{-1}(B_k - A_k)\| &\leq \|A_k^{-1}\|\|B_k - A_k\| \\ &\leq M\|H_k - \nabla^2_{xx}\mathcal{L}(w_k)\| \\ &\leq M\beta\mu_k.\end{aligned}$$

Consequently, there exists $k_1 \geq k_0$ such that for all $k \geq k_1$, $\|A_k^{-1}(B_k - A_k)\| \leq \frac{1}{2}$ and according to [20, Theorem 3.1.4], we obtain

$$\|J_k^{-1}\| = \|B_k^{-1}\| \leq \frac{\|A_k^{-1}\|}{1 - \|A_k^{-1}(B_k - A_k)\|} \leq 2M.$$

Let us now complete the proof. Let $k \geq k_1$. Define $e_k := \mathbf{w}(\mu_k^+) - w_k$. From Step 3 of Algorithm 1, we have

$$\begin{aligned}w_k^+ - \mathbf{w}(\mu_k^+) &= -J_k^{-1}F(w_k, \mu_k^+) - e_k \\ &= J_k^{-1}\big(F(\mathbf{w}(\mu_k^+), \mu_k^+) - F(w_k, \mu_k^+) - J_k e_k\big) \\ &= J_k^{-1}\int_0^1 \big(F'_w(w_k + te_k, \mu_k^+) - F'_w(w_k, \mu_k)\big)e_k \,\mathrm{d}t \qquad (24) \\ &\quad + J_k^{-1}(\nabla^2_{xx}\mathcal{L}(w_k) - H_k)(\mathbf{x}(\mu_k^+) - x_k) \qquad (25)\end{aligned}$$

Let us define $\bar{b} := \max\{1, b-1\}$, so that $|\mu_k^+ - \mu_k| \leq \bar{b}\mu_k$. Taking the norm on both sides of (24), then applying outcome (i) of Lemma 1, inequality (23) and Assumption 7, we obtain

$$\|w_k^+ - \mathbf{w}(\mu_k^+)\| \leq 2ML(\frac{1}{2}\|e_k\|^2 + \bar{b}\mu_k\|e_k\|) + 2M\beta\mu_k\|e_k\|.$$

Using conclusion (iv) of Lemma 1, we have

$$\begin{aligned}\|e_k\| &\leq \|w_k - \mathbf{w}(\mu_k)\| + \|\mathbf{w}(\mu_k) - \mathbf{w}(\mu_k^+)\| \\ &\leq \|w_k - \mathbf{w}(\mu_k)\| + C\bar{b}\mu_k,\end{aligned}$$

from which the conclusion follows. □

4.2 Asymptotic result

Before stating the main result of this section, we begin by establishing an estimate of the distance of the next iterate to the trajectory $\mathbf{w}$.

Lemma 3 *Assume that Assumptions 4–7 hold. The sequence of iterates generated by Algorithm 1 satisfies*

$$\|w_{k+1} - \mathbf{w}(\mu_{k+1})\| = \mathrm{O}(\|w_k^+ - \mathbf{w}(\mu_k^+)\|).$$

Proof Since $F'_w(w^*,0)$ is nonsingular, there exist $\tilde{r} \in (0,\bar{r})$, $\tilde{\mu} \in (0,\mu^*)$ and two constants $0 < L_1 < L_2$ such that, for all $w, w' \in B(w,\tilde{r})$ and $\mu \in (0,\tilde{\mu})$,

$$L_1\|w - w'\| \leq \|F(w,\mu) - F(w',\mu)\| \leq L_2\|w - w'\|, \tag{26}$$

see, e.g., [20, Lemma 4.1.16]. According to Lemma 2 and since the sequence $\{w_k\}$ converges to w^*, the sequence $\{w_k^+\}$ also converges to w^*. There exists an integer N such that for all $k \geq N$,

$$w_{k+1}, w_k^+, \mathbf{w}(\mu_{k+1}), \mathbf{w}(\mu_k^+) \in B(w,\tilde{r}) \quad \text{and} \quad \mu_{k+1}, \mu_k^+ \in (0,\tilde{\mu}).$$

Let $k \geq N$. If there is no inner iteration at the kth outer iteration of Algorithm 1, then we trivially have $\|w_{k+1} - \mathbf{w}(\mu_{k+1})\| = \|w_k^+ - \mathbf{w}(\mu_k^+)\|$, which implies

$$\|w_{k+1} - \mathbf{w}(\mu_{k+1})\| \leq \frac{L_2}{L_1}\|w_k^+ - \mathbf{w}(\mu_k^+)\|. \tag{27}$$

Suppose now that (w_{k+1},μ_{k+1}) is determined by a sequence of inner iterations. We then have $\|F(w_k^+,\mu_k^+)\| > \varepsilon_k$. As a result, we deduce that

$$\begin{aligned} L_1\|w_{k+1} - \mathbf{w}(\mu_{k+1})\| &\leq \|F(w_{k+1},\mu_{k+1}) - F(\mathbf{w}(\mu_{k+1}),\mu_{k+1})\| \\ &= \|F(w_{k+1},\mu_{k+1})\| \\ &\leq \varepsilon_k \\ &< \|F(w_k^+,\mu_k^+)\| \\ &= \|F(w_k^+,\mu_k^+) - F(\mathbf{w}(\mu_k^+),\mu_k^+)\| \\ &\leq L_2\|w_k^+ - \mathbf{w}(\mu_k^+)\|. \end{aligned}$$

The inequality (27) is once again satisfied, which completes the proof. □

We are now in position to state the main result.

Theorem 3 *Assume that Assumptions 4–7 hold. Then, the sequence of iterates generated by Algorithm 1 satisfies*

$$w_k = \mathbf{w}(\mu_k) + \mathrm{o}(\mu_k).$$

Moreover, for sufficiently large $k \in \mathbb{N}$*, we have* $\|F(w_k^+,\mu_k^+)\| \leq \varepsilon_k$*, which implies that* $(w_{k+1},\mu_{k+1}) = (w_k^+,\mu_k^+)$*, meaning that Algorithm 1 asymptotically reduces to full Newton steps.*

Proof Let us first note that the choice of μ_k^+ at Step 1 implies that

$$\mu_k^2 = \mathrm{o}(\mu_k^+). \tag{28}$$

From Lemmas 2 and 3, there exists a positive constant N such that the inequality $\|w_{k+1} - \mathbf{w}(\mu_{k+1})\| \leq N(\|w_k - \mathbf{w}(\mu_k)\|^2 + \mu_k^2)$ holds for all $k \in \mathbb{N}$. Let us define $e_k := N\|w_k - \mathbf{w}(\mu_k)\|$ and $\tilde{\mu}_k := N\mu_k$. We have for all $k \in \mathbb{N}$

$$e_{k+1} \leq e_k^2 + \tilde{\mu}_k^2. \tag{29}$$

By relation (28) there exists $\bar{k} \in \mathbb{N}$ such that for all $k \geq \bar{k}$, $\mu_k^2 \leq \frac{1}{2N}\mu_k^+$. By the choice of μ_{k+1} at Step 5 of Algorithm 1, for all $k \geq \bar{k}$ we then have $\tilde{\mu}_k^2 \leq \frac{1}{2}\tilde{\mu}_{k+1}$.

Suppose now that

$$\tilde{\mu}_k < e_k, \quad \text{for all } k \geq \bar{k}. \tag{30}$$

On one hand, the first inequality in (10) implies that there exist $a_1 > 0$ and $q_1 \in (0,1)$ such that, for k large enough,

$$a_1 q_1^{(1+\sigma)^k} \leq \tilde{\mu}_k.$$

On the other hand, from inequalities (29) and (30), we have $e_{k+1} \leq 2e_k^2$, for all $k \geq \bar{k}$. It follows that there exist $a_2 > 0$ and $q_2 \in (0,1)$ such that, for k large enough,

$$e_k \leq a_2 q_2^{2^k}.$$

Hence, according to inequality (30), we would have for k large enough

$$a_1 q_1^{(1+\sigma)^k} \leq a_2 q_2^{2^k}.$$

This is in contradiction with the fact that $1 + \sigma < 2$ and consequently relation (30) fails to hold.

What we have just proved means that there exists an integer $k_0 \geq \bar{k}$ such that $e_{k_0} \leq \tilde{\mu}_{k_0}$. Let us show now by induction that for all $k \geq k_0$,

$$e_k \leq \tilde{\mu}_k.$$

The base case is clear. As inductive hypothesis suppose that $e_k \leq \tilde{\mu}_k$ for some integer $k \geq k_0$. By inequality (29), we deduce that $e_{k+1} \leq 2\tilde{\mu}_k^2 \leq \tilde{\mu}_{k+1}$ and thus our claim is true for $k+1$.

Using inequality (29) and the above inequality proved by induction, we obtained $e_{k+1} = \mathrm{O}(\mu_k^2)$. Finally according to equality (28) and $\mu_k^+ \leq \mu_{k+1}$ that follows from Step 5 of Algorithm 1, we have $e_{k+1} = \mathrm{o}(\mu_{k+1})$, which shows that

$$w_k = \mathbf{w}(\mu_k) + \mathrm{o}(\mu_k).$$

Let us now prove the second part of Theorem 3. Using the second inequality of (26), we have for k large enough

$$\|F(w_k^+, \mu_k^+)\| = \|F(w_k^+, \mu_k^+) - F(\mathbf{w}(\mu_k^+), \mu_k^+)\| \leq L_2 \|w_k^+ - \mathbf{w}(\mu_k^+)\|.$$

According to Lemma 2, we deduce that $\|F(w_k^+, \mu_k^+)\| = \mathrm{O}(\|w_k - \mathbf{w}(\mu_k)\|^2 + \mu_k^2)$. Applying the result obtained in the first part, we finally obtain

$$\|F(w_k^+, \mu_k^+)\| = \mathrm{O}(\mu_k^2).$$

According to the relation (28) and to the choice of ε_k by (12), we conclude that the inequality $\|F(w_k^+, \mu_k^+)\| \leq \varepsilon_k$ holds for k large enough. □

The conclusion of Theorem 3 implies that whenever y^* is nonzero, the rate of convergence of $\{w_k\}$ is the same as those of $\{\mu_k\}$. Indeed, we have

$$w_{k+1} - w^* = \mathbf{w}'(0)\mu_k + \mathrm{o}(\mu_k)$$

and, by differentiating $F(\mathbf{w}(\mu), \mu)$ at $(w^*, 0)$, we can remark that

$$\mathbf{w}'(0) = F'_w(w^*, 0)^{-1} \begin{pmatrix} 0 \\ y^* \end{pmatrix} \neq 0.$$

It follows that the two sequences $\{\|w_{k+1} - w^*\| / \|w_k - w^*\|\}$ and $\{\mu_{k+1}/\mu_k\}$ are equivalent.

5 Numerical experiments

The algorithm presented in this article is referred as SPDOPT, which stands for Strongly Primal-Dual Optimization. The implementation has been done in C. In order to show the feasibility of this primal-dual approach, this algorithm is compared to IPOPT [47]. The latter is a line-search filter method which behaves like an SQP method when there is no inequality constraint.

5.1 SPDOPT implementation

Starting values

The starting point $x_{\rm s}$ is supposed to be given by the user. The initial vector of multipliers is set to

$$y_{\rm s} = (1, \dots, 1)^\top.$$

To be able to solve a convex quadratic problem in only one iteration, a first Newton iterate $\widehat{w} = (\widehat{x}, \widehat{y})$ is computed by solving the following linear system:

$$\begin{pmatrix} \nabla^2_{xx}\mathcal{L}(w_{\rm s}) & A(x_{\rm s})^\top \\ A(x_{\rm s}) & 0 \end{pmatrix} \begin{pmatrix} \widehat{x} - x_{\rm s} \\ \widehat{y} - y_{\rm s} \end{pmatrix} = - \begin{pmatrix} \nabla_x \mathcal{L}(w_{\rm s}) \\ c(x_{\rm s}) \end{pmatrix}.$$

Then, the choice of the starting point of the outer iteration algorithm is done as follows:

$$w_0 = \begin{cases} \widehat{w} & \text{if } \|F(\widehat{w}, 0)\|_\infty \le \|F(w_{\rm s}, 0)\|_\infty, \\ w_{\rm s} & \text{otherwise.} \end{cases}$$

In our experiments, the iterate $\widehat{w}$ is accepted in about sixty percent of cases.

Once the starting point for the first outer iteration is chosen, the initial value of the penalty parameter is defined by

$$\mu_0 = \min\{0.1, \|F(w_0, 0)\|_\infty\}.$$

Stopping conditions

The overall stopping test is done right from the beginning of each outer *and* inner iteration. If the norm of the optimality conditions at a current iterate w satisfies

$$\|F(w, 0)\|_\infty \le \varepsilon \qquad (31)$$

with $\varepsilon = 10^{-8}$, then it is considered that an optimal solution has been found by the algorithm.

The choice of ε_k in (13) is given by (14) and (16) with the values $\rho = 0.9$, $\ell = 5$ and $\theta = 10$.

Update of the penalty parameter

The new value of the penalty parameter is defined by

$$\mu_k^+ = \max\{\min\{\mu_k/10, \mu_k^{1.8}\}, \mu_k^{\min}\}, \tag{32}$$

where $\mu_k^{\min}$ is a value that is discussed hereafter. On one hand, the first argument of the maximum gives a linear, then a superlinear decrease of the penalty parameter. Moreover, formula (14), (32) and (16) imply that the inequalities (10) and (12) are satisfied with $\sigma = 0.8$, $a = b = 0.1$ and $c = 100$.

On the other hand, to avoid numerical difficulties due to very small values of μ_k, a lower bound $\mu_k^{\min}$ is imposed on the value of the penalty parameter. Let us show now how to choose the lower bound, without hindering the overall convergence. Suppose that μ_k achieves a constant and minimum value $\mu^{\min}$ in Algorithm 1. According to Proposition 1 and the choice (16), for sufficiently large k we have

$$\|F(w_k, \mu^{\min})\|_\infty \le \mu^{\min}\,\theta/(1-\rho) + \varepsilon/2,$$

where ε is the overall stopping tolerance. Using the triangle inequality, we have

$$\begin{aligned}\|F(w_k, 0)\|_\infty &\le \|F(w_k, \mu^{\min})\|_\infty + \mu^{\min}\|y_k\|_\infty \\ &\le (\theta/(1-\rho) + \|y_k\|_\infty)\mu^{\min} + \varepsilon/2.\end{aligned}$$

It follows that the overall stopping test (31) will be satisfied at w_k if

$$\mu^{\min} \le \widetilde{\mu}_k := \frac{\varepsilon}{2(\|y_k\|_\infty + \theta/(1-\rho))}.$$

To define a stationary sequence $\{\mu_k^{\min}\}$, set $\mu_0^{\min} := \widetilde{\mu}_0$, and for $k \ge 0$

$$\mu_{k+1}^{\min} := \begin{cases} \mu_k^{\min} & \text{if } \mu_k^{\min} \le \widetilde{\mu}_k, \\ \min\{\mu_k^{\min}/2, \widetilde{\mu}_k\} & \text{otherwise.}\end{cases}$$

It follows that $\{\mu_k^{\min}\}$ is a nonincreasing sequence and is stationary if and only if $\{y_k\}$ is bounded. To avoid numerical difficulties with very small values of μ_k we also do not let $\mu_k^{\min}$ becomes smaller than $100\,\epsilon_{\mathrm{m}}$ where $\epsilon_{\mathrm{m}} \simeq 10^{-16}$ is the machine epsilon.

At last, at the beginning of each outer iteration, once the value μ_k^+ is determined, we check if $\|F(w_k, \mu_k^+)\|_\infty \le \theta\mu_k$ is satisfied. If this is the case, we have $\|F(w_k, \mu_k^+)\|_\infty \le \varepsilon_k$, therefore we set $w_{k+1} := w_k$, $\mu_{k+1} := \mu_k^+$ and restart a new outer iteration. This skipping strategy ends after a finite number of steps. Indeed, if it not the case, we would have $\|F(w_k, \mu_k^{\min})\|_\infty \le \theta\mu_k^{\min}$ for k large enough, which would imply that the overall stopping condition (31) holds at w_k.

Solution of the linear system

The linear systems (11) and (17) are solved with a regularized Jacobian matrix of the form (7) in which $H(w) = \nabla^2_{xx}\mathcal{L}(w) + \delta I$. The factorization is done with the symmetric indefinite factorization code MA57 [23]. Since this factorization reveals the inertia, the regularization parameter δ is updated until a correct inertia is detected. This update is done according to [47, Algorithm IC], with the same values for the different constants, except that the step IC-2 of this algorithm is not applied in our case, because the bottom-right block of the matrix (7) is already nonzero. If δ becomes greater than 10^{20} then the run is halted and an error is returned.

In addition, if the solution of (17) does not satisfy

$$d_x^\top K^i d_x \geq 10^{-8} \|d_x\|^2,$$

the linear system is solved again with a correction term equal to $\delta + 10^{-4}$, where δ is the current value.

Line search

The merit function used in our experiments is defined by (6). At the beginning of Algorithm 2, the scaling parameter is set to $\nu = \mu^0$.

At Step 4 of Algorithm 2, the computation of the step length α uses quadratic and cubic interpolations (see, e.g., [40, pp. 57–59]) of the merit function φ_μ. The two parameters are set to $\omega = 0.01$ and $\kappa = 0.1$. Whenever the step length α becomes lower than 10^{-20}, the run is stopped and an error is returned.

5.2 Numerical results

We made a comparative test of SPDOPT against IPOPT 3.10.1, on 109 equality constrained problems from the CUTEr [32] and COPS [21] collections. The test problems are listed in Table 2, where n and m are the numbers of variables and constraints. All models are formulated using the AMPL modeling language [28]. Tests were run on a MacBook Pro 2.3GHz with an Intel Core i5 and 4 Go of memory.

As we said in the introduction, the primal-dual approach offers a natural regularization of the linear system when the Jacobian of constraints is rank deficient. To evaluate the behavior of our algorithm in this degenerate case, we used the same strategy as the one proposed in [18]. More precisely, for each problem, we add an additional constraint of the form $g_1(x) = g_1(x)^2$, where g_1 is the first constraint of the model. The Jacobian matrix of the constraints for these problems is rank deficient everywhere.

In this manner, we obtain a total of 218 problems, half of them are called *standard*, while the other half are called *degenerate*.

Knowing that a scaling procedure of an optimization problem may have a great influence on the performance of an optimization solver, simply by a stopping criteria which behaves differently for different scaling factors, we performed the numerical tests without any scaling strategy. In particular, IPOPT was applied with the option `nlp_scaling_method` set to `none`. To get a similar stopping test as (31) for both algorithms, we also set the IPOPT options `dual_inf_tol` and `constr_viol_tol` to `1e-8`. The maximum number of iterations is set to 3000 for both solvers.

IPOPT was also applied with the linear solver MA57. The coefficient matrix of the linear system in IPOPT is of the form

$$\begin{pmatrix} \nabla^2_{xx}\mathcal{L}(w) + \delta_w I & A(x)^\top \\ A(x) & -\delta_c I \end{pmatrix},$$

where δ_w and $\delta_c \geq 0$ are regularization parameters chosen so that the inertia of the matrix equals $(n, m, 0)$. In particular, the size of δ_c depends on the value of the parameter called `jacobian_regularization_value` whose default value is `1e-8`. To improve the robustness of IPOPT to solve the degenerate problems, we performed experiments with the value `1e-4`.

Tables 3 and 4 summarize the results obtained by the solvers on standard and degenerate problems. The results of IPOPT in Table 4 are those obtained with a regularization parameter of the Jacobian equal to 10^{-4}. The columns are labeled as follows: `f` is the value of the objective function at the final point, `#f` and `#g` are the numbers of function and gradient evaluations, `cpu` is the total runtime in seconds, including the function evaluations. A positive number in the column `e` indicates a failure. Table 1 shows the different kinds of errors returned by the solvers.

e	error message
1	maximum number of iterations reached
2	objective function value exceeded bound value
3	optimality conditions norm exceeded bound value
4	restoration phase failed
5	search direction is becoming too small
6	cannot recompute multipliers for feasibility problems
7	solved to acceptable level

Table 1: Error messages returned by SPDOPT (1–3) and IPOPT (1, 4–7)

SPDOPT solves all standard problems, whereas IPOPT solves all but one of them. It is worth noting that different optimal values are sometimes obtained. For problems bt04, dixchlng, dtoc1nd, lukvle04, mwright and robot, SPDOPT found a better solution, whereas for bt07, lukvle05, lukvle15 and s338, the reverse is true.

Figure 3 summarizes the numerical tests by means of logarithmic performance profiles [22] on the number of function evaluations, gradient evaluations and factorizations. For $\tau \geq 0$, the value $\rho_s(\tau)$ represents the fraction of problems for which the performance of the solver is within a factor 2^τ of the best one.

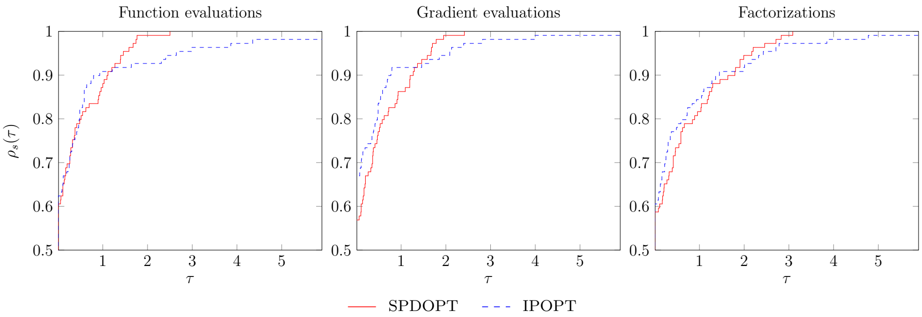

Fig. 3: Performance profiles of SPDOPT and IPOPT on the collection of 109 standard problems.

From these first tests, we can conclude that our primal-dual method is slightly less efficient than an SQP-like method in terms of gradient evaluations and factorizations on a set of standard problems. This conclusion should be tempered by the fact that IPOPT is a mature software that takes advantage of years of experience. The observed discrepancy is smaller with respect to the number of function evaluations. This is due to the fact that SPDOPT does not necessarily perform a backtracking line search at each iteration, while IPOPT can potentially do it at each iteration. Figure 4 shows the overall performances of the solvers on the

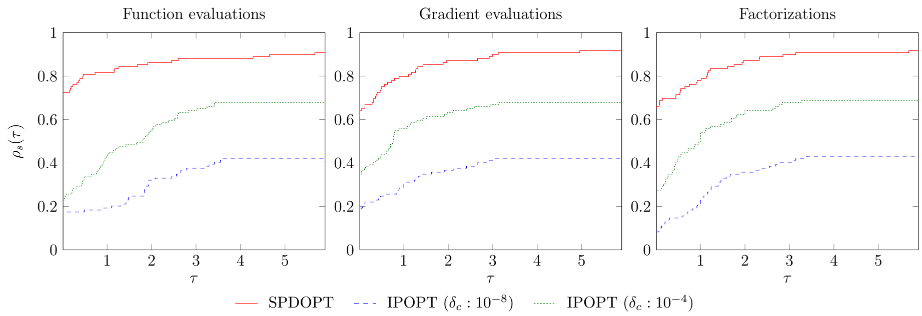

Fig. 4: Performance profiles of SPDOPT and IPOPT on the collection of 109 degenerate problems. The two dashed curves are when the regularization parameter of the Jacobian in IPOPT is set to 10^{-8} (bottom curve) and 10^{-4} (middle curve).

collection of degenerate problems. It is clear that SDOPT outperforms the solver IPOPT for these problems. The robustness of SPDOPT is remarkable since it is able to solve more than ninety percent of the degenerate problems. We performed two experiments with IPOPT with different values of the Jacobian regularization parameter. With the default value equal to 10^{-8}, IPOPT is able to satisfy the final optimality test in only about forty percent of the degenerate problems, while by increasing this value to 10^{-4} we observed that IPOPT solves nearly seventy percent of these problems. But we did not find a value for the regularization parameter such that the performances of IPOPT are as good as the ones of SPDOPT for this class of problems. The regularization alone is not sufficient to achieve strong robustness. One possible explanation is that the introduction of the regularization parameter into the linear system of an SQP-like method, transforms the equation of the contraints linearization $A(x)d_x = -g(x)$, into an equation of the form $A(x)d_x - \delta_c d_y = -g(x)$. It follows that for $\delta_c > 0$, the solution d_x can no longer be a descent direction for the function $\alpha \mapsto -\|g(x + \alpha d_x)\|$, and so it can be difficult to reduce the infeasibility measure $\|g\|$.

We also compared the computational times. To ensure a fair comparison, we have only included the problems that are solved by both solvers and whose CPU times are greater than 0.05 seconds. Indeed, for problems quickly solved, IPOPT spends much of the time in the initialization process, which introduces a bias on the runtimes comparison of these problems. The comparison is then done on 39 standard problems and 21 degenerate problems. The performance profiles on Figure 5 confirm that for the standard problems the performances of both solvers are comparable, while the primal-dual algorithm implemented in SPDOPT is well suited to solving degenerate problems.

The numerical results raise the question of whether the asymptotic convergence properties of our algorithm still hold without the linear independence constraint qualification assumption. Tables 3 and 4 show that there is an obvious loss of performances for degenerate problems, but we did not observe empirical evidence of a loss of the superlinear convergence property. Without this assumption, our analysis breaks down right from the beginning, because of the singularity of the Jacobian at the solution and also from the unboundedness and possibly the emptiness of the set of multipliers. Such interesting considerations are, unfortunately, beyond the scope of the present paper and could be examined in a future research.

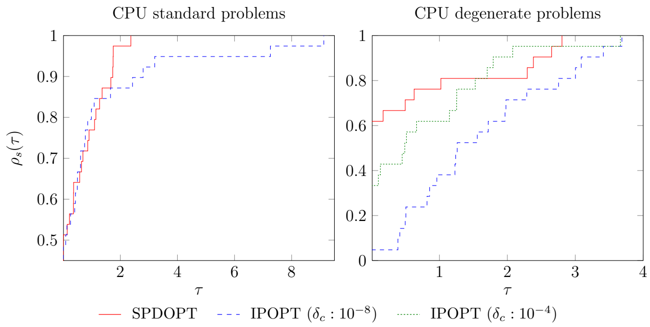

Fig. 5: Performance profiles of SPDOPT and IPOPT on 39 standard problems and 21 degenerate problems. The selected problems are those for which the solvers succeeded in finding an optimal solution with more than 0.05 seconds of CPU times.

Acknowledgements We would like to thank Elsa Bousquet [8] for discussions on a primitive version of the algorithm and Michel Bouard for his serious efforts to implement the optimization software SPDOPT. We also thanks the referees for their valuable efforts in reading the paper and their helpful critical comments.

References

1. Armand, P.: A quasi-Newton penalty barrier method for convex minimization problems. Comput. Optim. Appl. **26**(1), 5–34 (2003)
2. Armand, P., Benoist, J., Orban, D.: Dynamic updates of the barrier parameter in primal-dual methods for nonlinear programming. Comput. Optim. Appl. **41**(1), 1–25 (2008)
3. Armand, P., Benoist, J., Orban, D.: From global to local convergence of interior methods for nonlinear optimization. Optim. Methods Softw. **28**(5), 1051–1080 (2013).
4. Benchakroun, A., Dussault, J.P., Mansouri, A.: A two parameter mixed interior-exterior penalty algorithm. ZOR—Math. Methods Oper. Res. **41**(1), 25–55 (1995)
5. Benson, H.Y., Vanderbei, R.J., Shanno, D.F.: Interior-point methods for nonconvex nonlinear programming: filter methods and merit functions. Comput. Optim. Appl. **23**(2), 257–272 (2002)
6. Bertsekas, D.P.: Constrained optimization and Lagrange multiplier methods. Computer Science and Applied Mathematics. Academic Press Inc. [Harcourt Brace Jovanovich Publishers], New York (1982)

7. Biegler, L.T.: Nonlinear programming:Concepts, algorithms, and applications to chemical processes, *MOS-SIAM Series on Optimization*, vol. 10. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA (2010)
8. Bousquet, E.: Optimisation non linéaire et application au réglage d'un réseau de télescopes. Ph.D. thesis, Université de Limoges, École Doctorale S2i (2009)
9. Broyden, C.G., Attia, N.F.: A smooth sequential penalty function method for solving nonlinear programming problems. In: System modelling and optimization (Copenhagen, 1983), *Lecture Notes in Control and Information Sciences*, vol. 59, pp. 237–245. Springer, Berlin (1984)
10. Broyden, C.G., Attia, N.F.: Penalty functions, Newton's method and quadratic programming. J. Optim. Theory Appl. **58**(3), 377–385 (1988)
11. Byrd, R.H., Gilbert, J.C., Nocedal, J.: A trust region method based on interior point techniques for nonlinear programming. Math. Program. **89**(1, Ser. A), 149–185 (2000)
12. Byrd, R.H., Marazzi, M., Nocedal, J.: On the convergence of Newton iterations to non-stationary points. Math. Program. **99**(1, Ser. A), 127–148 (2004)
13. Byrd, R.H., Nocedal, J., Waltz, R.A.: Feasible interior methods using slacks for nonlinear optimization. Comput. Optim. Appl. **26**(1), 35–61 (2003)
14. Byrd, R.H., Nocedal, J., Waltz, R.A.: KNITRO: An integrated package for nonlinear optimization. In: Large-scale nonlinear optimization, *Nonconvex Optimization and Its Applications*, vol. 83, pp. 35–59. Springer, New York (2006)
15. Chen, L., Goldfarb, D.: Interior-point l_2-penalty methods for nonlinear programming with strong global convergence properties. Math. Program. **108**(1, Ser. A), 1–36 (2006)
16. Conn, A.R., Gould, N.I.M., Toint, P.L.: Trust-region methods. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA (2000)
17. Courant, R.: Variational methods for the solution of problems of equilibrium and vibrations. Bull. Amer. Math. Soc. **49**, 1–23 (1943)
18. Curtis, F.E., Nocedal, J., Wächter, A.: A matrix-free algorithm for equality constrained optimization problems with rank-deficient Jacobians. SIAM J. Optim. **20**(3), 1224–1249 (2009)
19. Debreu, G.: Definite and semidefinite quadratic forms. Econometrica **20**, 295–300 (1952)
20. Dennis Jr., J.E., Schnabel, R.B.: Numerical methods for unconstrained optimization and nonlinear equations. Prentice Hall Series in Computational Mathematics. Prentice Hall Inc., Englewood Cliffs, NJ (1983)
21. Dolan, E., Moré, J., Munson, T.: Benchmarking optimization software with COPS 3.0. Tech. rep., Argonne National Laboratory (2004)
22. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. **91**(2, Ser. A), 201–213 (2002)
23. Duff, I.S.: Ma57—a code for the solution of sparse symmetric definite and indefinite systems. ACM Trans. Math. Software **30**, 118–144 (2004)
24. Fiacco, A.V., McCormick, G.P.: Nonlinear programming: Sequential unconstrained minimization techniques. John Wiley and Sons, Inc., New York-London-Sydney (1968)
25. Fletcher, R.: Practical methods of optimization, second edn. John Wiley & Sons Ltd., Chichester (1987)
26. Forsgren, A., Gill, P.E.: Primal-dual interior methods for nonconvex nonlinear programming. SIAM J. Optim. **8**(4), 1132–1152 (1998)
27. Forsgren, A., Gill, P.E., Wright, M.H.: Interior methods for nonlinear optimization. SIAM Rev. **44**(4), 525–597 (2002)
28. Fourer, R., Gay, D.M., Kernighan, B.W.: AMPL: A Modeling Language for Mathematical Programming, 2 edn. Brooks/Cole (2002)
29. Gertz, E.M., Gill, P.E.: A primal-dual trust region algorithm for nonlinear optimization. Math. Program. **100**(1, Ser. B), 49–94 (2004)
30. Gill, P.E., Robinson, D.P.: A primal-dual augmented Lagrangian. Comput. Optim. Appl. **51**(1), 1–25 (2012)
31. Goldfarb, D., Polyak, R., Scheinberg, K., Yuzefovich, I.: A modified barrier-augmented Lagrangian method for constrained minimization. Comput. Optim. Appl. **14**(1), 55–74 (1999)
32. Gould, N., Orban, D., Toint, P.: CUTEr and SifDec: A constrained and unconstrained testing environment, revisited. ACM Trans. Math. Soft. **29**(4), 373–394 (2003)
33. Gould, N., Orban, D., Toint, P.: An interior-point ℓ_1-penalty method for nonlinear optimization. Tech. Rep. RAL-TR-2003-022, Rutherford Appleton Laboratory, Chilton, Oxfordshire, England (2003)

34. Gould, N., Orban, D., Toint, P.: Numerical methods for large-scale nonlinear optimization. Acta Numer. **14**, 299–361 (2005)
35. Gould, N.I.M.: On the accurate determination of search directions for simple differentiable penalty functions. IMA J. Numer. Anal. **6**(3), 357–372 (1986)
36. Gould, N.I.M.: On the convergence of a sequential penalty function method for constrained minimization. SIAM J. Numer. Anal. **26**(1), 107–128 (1989)
37. Griva, I., Shanno, D.F., Vanderbei, R.J., Benson, H.Y.: Global convergence of a primal-dual interior-point method for nonlinear programming. Algorithmic Oper. Res. **3**(1), 12–29 (2008)
38. Murray, W.: Analytical expressions for the eigenvalues and eigenvectors of the Hessian matrices of barrier and penalty functions. J. Optimization Theory Appl. **7**, 189–196 (1971)
39. Nemirovski, A.S., Todd, M.J.: Interior-point methods for optimization. Acta Numer. **17**, 191–234 (2008)
40. Nocedal, J., Wright, S.J.: Numerical optimization, second edn. Springer Series in Operations Research and Financial Engineering. Springer, New York (2006)
41. Shanno, D.F., Vanderbei, R.J.: Interior-point methods for nonconvex nonlinear programming: orderings and higher-order methods. Math. Program. **87**(2, Ser. B), 303–316 (2000). Studies in algorithmic optimization
42. Tits, A.L., Wächter, A., Bakhtiari, S., Urban, T.J., Lawrence, C.T.: A primal-dual interior-point method for nonlinear programming with strong global and local convergence properties. SIAM J. Optim. **14**(1), 173–199 (2003)
43. Vanderbei, R.J., Shanno, D.F.: An interior-point algorithm for nonconvex nonlinear programming. Comput. Optim. Appl. **13**(1-3), 231–252 (1999). Computational optimization—a tribute to Olvi Mangasarian, Part II
44. Wächter, A., Biegler, L.T.: Failure of global convergence for a class of interior point methods for nonlinear programming. Math. Program. **88**(3, Ser. A), 565–574 (2000)
45. Wächter, A., Biegler, L.T.: Line search filter methods for nonlinear programming: local convergence. SIAM J. Optim. **16**(1), 32–48 (2005)
46. Wächter, A., Biegler, L.T.: Line search filter methods for nonlinear programming: motivation and global convergence. SIAM J. Optim. **16**(1), 1–31 (2005)
47. Wächter, A., Biegler, L.T.: On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math. Program. **106**(1, Ser. A), 25–57 (2006)
48. Waltz, R.A., Morales, J.L., Nocedal, J., Orban, D.: An interior algorithm for nonlinear optimization that combines line search and trust region steps. Math. Program. **107**(3, Ser. A), 391–408 (2006)
49. Wright, M.H.: The interior-point revolution in optimization: history, recent developments, and lasting consequences. Bull. Amer. Math. Soc. (N.S.) **42**(1), 39–56 (2005)
50. Yamashita, H., Yabe, H.: An interior point method with a primal-dual quadratic barrier penalty function for nonlinear optimization. SIAM J. Optim. **14**(2), 479–499 (2003)

Appendix

Table 2: Test problems

Problem	n	m	Problem	n	m
aug2d	20192	9996	hs050-052	5	3
aug3d	3873	1000	hs061	3	2
aug3dc	3873	1000	hs077	5	2
bt01	2	1	hs078-079	5	3
bt02	3	1	hs100lnp	7	2
bt03	5	3	lukvle01	1000	998
bt04-05	3	2	lukvle02	1000	993
bt06	5	2	lukvle03	1000	2
bt07	5	3	lukvle04	1000	998
bt08	5	2	lukvle05	1000	996
bt09	4	2	lukvle06	999	499
bt11-12	5	3	lukvle07	1000	4
byrdsphr	3	2	lukvle08	1000	998
catena	3002	1001	lukvle09	1000	6
chain1	799	600	lukvle10	1000	998
chain2	1599	1200	lukvle11	998	664
chain3	3199	2400	lukvle12	997	747
dixchlng	10	5	lukvle13-14	998	664
dtoc1na-c	1485	990	lukvle15-18	997	747
dtoc1nd	735	490	maratos	2	1
dtoc2	5994	3996	mwright	5	3
dtoc5	9998	4999	orthrdm2	4003	2000
dtoc6	10000	5000	orthrega	517	256
eigena∗	110	55	orthregb	27	6
eigenb∗	110	55	orthregc	10005	5000
eigenc2	462	231	orthregd	10003	5000
eigencco	30	15	orthrgdm	10003	5000
elec1	150	50	orthrgds	10003	5000
elec2	300	100	robot	7	2
elec3	600	200	s216	2	1
gilbert	1000	1	s219	4	2
gridnetb	13284	6724	s235	3	1
hager1-3	10000	5000	s269	5	3
hs006-007	2	1	s316-322	2	1
hs009	2	1	s335-336	3	2
hs026-028	3	1	s338	3	2
hs039	4	2	s344-345	3	1
hs040	4	3	s375	10	9
hs046	5	2	s378	10	3
hs047	5	3	s394	20	1
hs048-049	5	2	s395	50	1

Table 3: Numerical results of SPDOPT and IPOPT for standard problems

Problem	SPDOPT				IPOPT			
	#f/ #g	fact	cpu	e	#f/ #g	fact	cpu	e
aug2d	2/ 2	1	.1839		2/ 2	1	.3095	
aug3d	2/ 2	1	.2843		3/ 3	5	.8965	
aug3dc	2/ 2	1	.0336		2/ 2	1	.0634	
bt01	11/ 11	18	.0008		15/ 8	21	.0184	
bt02	12/ 12	11	.0008		13/ 13	12	.0228	
bt03	2/ 2	1	.0006		2/ 2	1	.0059	
bt04	11/ 11	15	.0011		10/ 10	16	.0188	
bt05	9/ 9	8	.0009		8/ 8	7	.0136	
bt06	15/ 15	20	.0011		18/ 14	13	.0262	
bt07	39/ 32	59	.0029		32/ 14	19	.0268	
bt08	70/ 70	172	.0031		29/ 29	84	.0573	
bt09	10/ 10	13	.0013		14/ 14	16	.0243	
bt11	10/ 10	9	.0010		9/ 9	8	.0189	
bt12	7/ 7	11	.0008		5/ 5	4	.0097	
byrdsphr	13/ 12	22	.0010		19/ 19	28	.0355	
catena	31/ 28	31	.2727		633/ 198	447	152.4	
chain1	15/ 15	21	.1043		8/ 8	9	.0646	
chain2	32/ 31	45	.4151		10/ 9	10	.1232	
chain3	18/ 18	25	.4921		1257/ 286	685	75.27	
dixchlng	29/ 27	71	.0050		11/ 11	10	.0214	
dtoc1na	7/ 7	6	.2257		7/ 7	6	.1243	
dtoc1nb	9/ 9	8	.2993		7/ 7	6	.1237	
dtoc1nc	11/ 11	17	.4494		21/ 16	22	.3488	
dtoc1nd	105/ 89	194	2.312		40/ 28	56	.4472	
dtoc2	14/ 14	26	3.100		13/ 11	14	.9749	
dtoc5	5/ 5	4	.1384		5/ 5	4	.1653	
dtoc6	20/ 20	19	.5989		13/ 13	12	.4674	
eigena2	9/ 9	14	.0565		3/ 3	4	.0241	
eigenaco	11/ 11	15	.0798		4/ 4	4	.0289	
eigenb2	10/ 10	22	.0884		14/ 14	24	.0968	
eigenbco	224/ 151	323	2.119		105/ 83	180	1.419	
eigenc2	87/ 85	203	10.58		26/ 22	31	3.175	
eigencco	14/ 14	24	.0104		13/ 13	24	.0359	
elec1	38/ 38	85	.6493		47/ 42	95	1.377	
elec2	57/ 57	127	4.560		302/ 206	503	42.05	
elec3	111/ 96	210	36.48		257/ 164	401	254.0	
gilbert	19/ 19	18	.0453		20/ 20	22	.1176	
gridnetb	2/ 2	1	.1053		2/ 2	1	.2086	
hager1	2/ 2	1	.0335		2/ 2	1	.0661	
hager2	2/ 2	1	1.038		2/ 2	1	1.104	
hager3	2/ 2	1	.0658		2/ 2	1	.0930	
hs006	6/ 6	6	.0007		7/ 6	10	.0113	
hs007	9/ 9	8	.0008		28/ 28	55	.0480	
hs009	13/ 13	23	.0011		6/ 4	5	.0089	
hs026	28/ 28	27	.0012		26/ 26	25	.0393	
hs027	11/ 11	13	.0008		160/ 58	70	.1158	
hs028	2/ 2	1	.0006		2/ 2	1	.0043	
hs039	10/ 10	13	.0015		14/ 14	16	.0262	
hs040	5/ 5	4	.0008		4/ 4	3	.0074	
hs046	23/ 23	22	.0014		20/ 20	19	.0329	
hs047	34/ 33	46	.0017		21/ 20	19	.0338	
hs048	2/ 2	1	.0008		2/ 2	1	.0043	
hs049	21/ 21	20	.0015		21/ 21	20	.0337	
hs050	10/ 10	9	.0009		10/ 10	9	.0153	
hs051	2/ 2	1	.0010		2/ 2	1	.0044	

Problem	SPDOPT				IPOPT			
	#f/ #g	fact	cpu	e	#f/ #g	fact	cpu	e
hs052	2/ 2	1	.0007		2/ 2	1	.0053	
hs061	23/ 23	23	.0016		10/ 10	10	.0183	
hs077	12/ 12	16	.0010		13/ 12	11	.0207	
hs078	6/ 6	5	.0011		5/ 5	4	.0099	
hs079	17/ 16	15	.0011		5/ 5	4	.0088	
hs100lnp	14/ 13	21	.0018		21/ 21	25	.0376	
lukvle01	10/ 10	9	.0690		7/ 7	6	.0537	
lukvle02	10/ 10	9	.1035		10/ 10	9	.0939	
lukvle03	11/ 11	10	.0263		11/ 11	10	.0497	
lukvle04	13/ 13	12	.0713		18/ 18	17	.1087	
lukvle05	49/ 38	59	.3902		24/ 20	24	.1824	
lukvle06	16/ 16	15	.1476		16/ 16	21	.1953	
lukvle07	10/ 10	16	.0194		13/ 13	19	.0474	
lukvle08	14/ 13	21	.0954		17/ 17	21	.1343	
lukvle09	24/ 24	42	.0346		24/ 23	32	.0781	
lukvle10	14/ 14	17	.0866		21/ 14	18	.1166	
lukvle11	12/ 12	11	.0389		9/ 9	8	.0479	
lukvle12	9/ 9	8	.0595		9/ 9	13	.0906	
lukvle13	76/ 67	155	.7195		39/ 26	39	.2797	
lukvle14	25/ 25	29	.1092		30/ 27	35	.1868	
lukvle15	1703/ 561	1319	4.723		8439/ 2453	5315	25.41	
lukvle16	9/ 9	8	.0354		9/ 9	12	.0601	
lukvle17	13/ 13	18	.0484		10/ 10	17	.0730	
lukvle18	13/ 13	19	.0512		14/ 14	20	.0929	
maratos	5/ 5	4	.0007		5/ 5	4	.0135	
mwright	21/ 21	33	.0018		11/ 11	16	.0270	
orthrdm2	16/ 16	33	.5666		8/ 7	6	.1687	
orthrega	145/ 91	207	1.099		78/ 61	138	.9450	
orthregb	17/ 16	34	.0070		3/ 3	4	.0078	
orthregc	34/ 32	61	13.18		28/ 15	27	8.451	
orthregd	18/ 18	33	1.575		15/ 13	17	.7118	
orthrgdm	18/ 17	31	1.508		373/ 59	384	21.31	4
orthrgds	23/ 22	40	1.875		22/ 17	29	1.000	
robot	13/ 13	29	.0013		10/ 9	24	.0195	
s216	9/ 9	8	.0011		9/ 7	6	.0128	
s219	12/ 12	15	.0013		75/ 51	67	.0848	
s235	27/ 16	15	.0011		26/ 14	13	.0282	
s269	2/ 2	1	.0010		2/ 2	1	.0043	
s316	8/ 8	7	.0010		8/ 8	15	.0157	
s317	8/ 8	7	.0009		10/ 10	17	.0173	
s318	8/ 8	7	.0008		11/ 11	18	.0186	
s319	9/ 9	8	.0008		12/ 12	19	.0185	
s320	11/ 11	10	.0010		14/ 14	21	.0227	
s321	12/ 12	11	.0008		20/ 17	30	.0341	
s322	12/ 12	11	.0014		95/ 33	72	.0777	
s335	15/ 15	14	.0010		26/ 26	25	.0404	
s336	12/ 12	15	.0011		18/ 18	31	.0336	
s338	25/ 24	49	.0021		39/ 39	81	.0691	
s344	7/ 7	6	.0009		8/ 8	7	.0136	
s345	19/ 18	18	.0014		13/ 11	12	.0203	
s375	25/ 23	46	.0055		20/ 20	37	.0384	
s378	18/ 18	26	.0034		16/ 16	23	.0283	
s394	16/ 16	24	.0019		17/ 17	27	.0353	
s395	16/ 16	26	.0040		19/ 19	29	.0339	

Table 4: Numerical results of SPDOPT and IPOPT for degenerate problems

Problem	SPDOPT				IPOPT			
	#f/#g	fact	cpu	e	#f/#g	fact	cpu	e
aug2d	7/7	6	.9228		11/11	13	2.202	
aug3d	7/7	11	1.558		12/12	20	2.130	
aug3dc	5/5	4	.1064		12/11	13	.4495	
bt01	13/13	24	.0010		74/49	70	.0884	
bt02	35/31	92	.0021	3	171/82	178	.1645	
bt03	16/16	22	.0012		58/16	31	.0385	
bt04	67/51	91	.0031		63/16	42	.0514	6
bt05	17/17	28	.0012		63/11	34	.0442	6
bt06	41/37	91	.0026		74/43	86	.0756	
bt07	81/65	138	.0046		137/35	61	.0946	
bt08	78/78	191	.0052		57/20	50	.0512	
bt09	32/26	42	.0018		129/63	115	.1296	
bt11	26/24	42	.0018		44/39	56	.0672	
bt12	10/9	15	.0013		106/39	97	.0874	
byrdsphr	35/30	68	.0028		267/16	304	.3952	6
catena	59/53	76	.6732		5624/1305	6006	2773	4
chain1	15/15	21	.1030		14/13	22	.3353	
chain2	17/17	21	.1953		13/12	19	2.477	
chain3	18/18	25	.5029		8651/62	3087	110.0	1
dixchlng	3381/758	2282	.1618	5	21/15	22	.0269	
dtoc1na	7/7	6	.2093		7/7	9	.1484	
dtoc1nb	9/9	8	.2983		7/7	9	.1470	
dtoc1nc	11/11	17	.4404		25/19	37	.4821	
dtoc1nd	90/62	131	1.597		28/23	43	.3260	
dtoc2	525/145	289	19.06		27/23	40	2.729	
dtoc5	10/10	9	.2995		19/13	15	.4743	
dtoc6	23/23	31	.8665		251/154	176	5.239	7
eigena2	9/9	14	.0522		5/5	7	.0256	
eigenaco	12/12	20	.0980		7/7	11	.0375	
eigenb2	10/10	22	.0793		95/88	211	.8205	
eigenbco	114/101	234	1.502		100/72	159	.9725	
eigenc2	212/191	459	31.62		35/24	52	5.415	
eigencco	71/48	81	.0354		13/13	24	.0313	
elec1	1182/356	425	4.265		47/47	109	.8174	
elec2	5804/872	1234	62.22		153/110	266	10.62	
elec3	256/165	403	68.28		128/41	150	63.38	4
gilbert	56/56	161	49.15		1620/1513	1590	1912	7
gridnetb	10/10	9	.7359		38/10	18	1.754	
hager1	22/22	21	.3957		103/49	68	1.577	
hager2	17/17	16	.5931		102/29	64	2.021	
hager3	17/17	16	.7123		102/29	64	2.465	
hs006	8030/459	1107	.0336		112/112	181	.1722	6
hs007	66/42	76	.0018		139/21	190	.2285	6
hs009	6/6	5	.0007		1/1	0	.0025	6
hs026	77/76	184	.0032		80/49	84	.0896	
hs027	20/20	33	.0012		162/106	236	.2011	
hs028	2/2	1	.0006		2/2	2	.0054	
hs039	27/26	43	.0020		288/202	266	.3707	
hs040	6/6	5	.0009		9/9	15	.0151	6
hs046	28/28	43	.0023		38/29	43	.0553	
hs047	73/62	120	.0035		53/47	88	.0824	
hs048	2/2	1	.0008		2/2	2	.0059	
hs049	21/21	20	.0013		21/21	23	.0394	
hs050	10/10	9	.0012		10/10	12	.0176	
hs051	2/2	1	.0008		2/2	2	.0059	

Problem	SPDOPT				IPOPT			
	#f/#g	fact	cpu	e	#f/#g	fact	cpu	e
hs052	12/12	12	.0013		42/11	22	.0284	
hs061	22/20	38	.0013		52/10	45	.0594	6
hs077	26/26	58	.0029		82/20	37	.0606	
hs078	13/13	17	.0014		24/18	28	.0352	
hs079	18/17	28	.0014		67/36	54	.0728	
hs100lnp	17425/2823	6979	.2655		105/91	136	.1918	
lukvle01	15/14	27	.1262		17/2	21	.2086	4
lukvle02	33/32	79	.4903		130/86	110	1.086	
lukvle03	53/48	105	.1748		63/29	52	.1716	
lukvle04	881/338	418	2.288	2	214/44	218	1.340	
lukvle05	71/64	150	.7700		81/37	79	.6842	
lukvle06	98/56	94	.7252		26/26	42	.2894	
lukvle07	24/23	51	.0534		64/35	61	.1435	
lukvle08	14/13	22	.0979		3006/3001	3010	20.42	1
lukvle09	3/2	4	.0058	3	3221/2970	6085	16.00	1
lukvle10	3/3	2	.0122	3	168/64	116	.6071	
lukvle11	123/73	164	.4123		105/62	120	.4139	7
lukvle12	19/18	22	.1537		42/18	30	.2158	
lukvle13	73/45	76	.3923		147/69	98	.7140	
lukvle14	24102/3002	7601	24.41	1	329/293	342	1.757	
lukvle15	5698/2071	4972	15.18		3438/1997	4695	21.68	
lukvle16	15/13	19	.0564		64/22	66	.2339	
lukvle17	35/33	74	.1492		36/34	51	.2245	
lukvle18	23/21	41	.0933		41/39	58	.2189	
maratos	8/8	7	.0008		73/19	37	.0485	6
mwright	28/26	43	.0023		18/2	18	.0477	4
orthrdm2	29/29	43	.7810		13/12	23	.3014	
orthrega	15318/3002	7580	20.81	1	178/108	250	.8366	
orthregb	56/52	129	.0280		316/183	602	.4635	5
orthregc	14/14	25	2.698		58/40	88	7.773	
orthregd	48/46	102	4.388		1206/841	1032	42.37	7
orthrgdm	34/34	73	5.088		36/23	50	1.761	
orthrgds	22031/3002	3058	276.4	1	118/79	191	7.487	5
robot	140/100	222	.0102		191/82	163	.1495	
s216	17/16	28	.0010		53/8	59	.0747	6
s219	62/55	142	.0040		440/74	209	.2449	
s235	26/23	56	.0023		135/40	73	.0847	
s269	12/12	11	.0010		42/11	22	.0271	
s316	13/12	21	.0012		36/8	13	.0292	6
s317	19/17	18	.0011		35/9	18	.0294	6
s318	17/16	21	.0010		91/31	53	.0849	6
s319	21/21	34	.0016		72/13	70	.1018	6
s320	25/25	39	.0019		60/16	59	.0807	6
s321	35/26	31	.0016		61/11	60	.0852	6
s322	12/12	11	.0012		63/22	60	.0784	6
s335	72/65	168	.0044		67/22	71	.0944	6
s336	13/13	16	.0013		79/78	109	.1187	
s338	42/31	63	.0028		79/40	73	.0703	
s344	40/37	99	.0029		77/62	134	.1041	
s345	68/65	185	.0035		61/47	110	.0814	
s375	20/17	31	.0029		119/28	87	.1144	6
s378	116/43	105	.0100	2	533/83	348	.3851	5
s394	73/68	173	.0198		45/2	80	.1269	4
s395	150/102	178	.1060		133/41	177	.2827	