An interior-point method for solving box-constrained underdetermined nonlinear systems

J.B. Francisco\textsuperscript{a,1}, N. Krejić\textsuperscript{b,∗,2}, J.M. Martínez\textsuperscript{a,3}

\textsuperscript{a}Department of Applied Mathematics, IMECC-UNICAMP, University of Campinas, CP 6065, 13081-970 Campinas SP, Brazil
\textsuperscript{b}Department of Mathematics and Informatics, University of Novi Sad, Trg Dositeja Obradovića 4, 21000 Novi Sad, Serbia and Montenegro

Received 2 April 2004; received in revised form 26 August 2004

Abstract

A method introduced recently by Bellavia, Macconi and Morini for solving square nonlinear systems with bounds is modified and extended to cope with the underdetermined case. The algorithm produces a sequence of interior iterates and is based on globalization techniques due to Coleman and Li. Global and local convergence results are proved and numerical experiments are presented.

© 2004 Elsevier B.V. All rights reserved.

MSC: 65H10

Keywords: Underdetermined nonlinear systems; Newton method; Global convergence; Quadratic convergence

∗ Corresponding author. Tel.: +38121458136; fax: +38121350458.
E-mail addresses: juliano@ime.unicamp.br (J.B. Francisco), natasa@im.ns.ac.yu (N. Krejić), martinez@ime.unicamp.br (J.M. Martínez).

1 This author was supported by PRONEX-Optimization 76.79.1008-00, FAPESP (Grant 02/00702-0) and CNPq.
2 This author was supported by a visiting professor grant of FAPESP (Grant 99/03102-0).
3 This author was supported by PRONEX-Optimization 76.79.1008-00, FAPESP (Grant 01/04597-4), CNPq and FAEP-UNICAMP.

0377-0427/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
1. Introduction

The feasible set of practical nonlinear programming problems is usually represented by a set of equations subject to bounds on the variables. Many traditional [1,10,14–16,18–22] and modern practical optimization methods (see [12,13]) exploit this structure and require specific algorithms for restoring feasibility at every iteration. Usually, the feasible region is nonempty and points that are interior with respect to the bounds exist. Therefore it is important to develop procedures that solve bound-constrained underdetermined set of equations in a more efficient way that ordinary bound-constraint minimization algorithms do.

With the aim of solving square \((n \times n)\) nonlinear systems subject to bounds, Bellavia et al. [2,3] developed an algorithm that adapts most of the ideas of the bound-constrained optimization solvers of Coleman and Li [4,5] to the resolution of systems of equations. In the present paper we modify and extend the algorithm of Bellavia, Macconi and Morini for dealing with the underdetermined case.

So, the problem considered here is to solve

\[
F(x) = 0, \quad x \in \Omega, \tag{1}
\]

where

\[
\Omega = \{x \in \mathbb{R}^n | l \leq x \leq u\}, \tag{2}
\]

\(l_i \in \mathbb{R} \cup \{-\infty\}, u_i \in \mathbb{R} \cup \{\infty\}\) and \(l_i < u_i\) for \(i = 1, \ldots, n\). We assume that \(F : \Omega \to \mathbb{R}^m, (m \leq n)\) is continuously differentiable on an open set that contains \(\Omega\).

The resolution of unconstrained underdetermined nonlinear systems has been addressed using the quasi-Newton approach in [11,23] and using the Levenberg–Marquardt approach in [7]. Some ideas of these papers are incorporated in the present work. In particular, we will take the Newton-pseudoinverse (normal-flow) direction to generate a trial point whenever possible. Quasi-Newton minimum-norm directions have been used in [11,23]. Methods for solving underdetermined systems based on the pseudoinverse can be globalized using ordinary trust regions or line-search procedures (see, for example [6,17]). When bounds are present, Euclidian trust regions are not so suitable. In practice the affine-scaling trust regions introduced by Coleman and Li are able to deal with boxes in a more sensible way. When the current point is close to the boundary (but not to the solution), the Coleman–Li trust-region strategy generally forces a large step, a feature that is generally required for efficient and practical optimization methods. It is important to observe that the method that will be introduced in this paper can be applied to the case \(\Omega = \mathbb{R}^n\). In this case it reduces to a standard trust-region globalization of the normal-flow method for underdetermined nonlinear systems.

A brief description of a typical iteration of the new method follows. Given the current interior iterate \(x^k\), the Coleman–Li trust region with radius \(\Delta \geq \Delta_{\text{min}} > 0\) is defined. The Cauchy point within this trust region and the normal-flow Newtonian step are computed. If the Newtonian step satisfies the trust-region constraint and the decrease of the quadratic model defined by the Newtonian step is large enough when compared to the decrease of the quadratic model defined by the Cauchy step (preserving feasibility), the feasible multiple of the Newtonian step defines a trial point. Otherwise, a different trial point within the trust region and satisfying sufficient decrease of the model is defined. (A feasible multiple of the Cauchy step is an admissible trial point.) The predicted reduction of the quadratic model and the actual reduction of the objective function are computed at the trial point as in standard trust-region methods. If the actual reduction is large enough when compared to the predicted reduction, the trial point is accepted and a new
iteration begins. Otherwise, the trust-region radius is reduced. The philosophy of the method is to use the Newtonian direction as frequently as possible, taking advantage of its good local convergence properties. However, the globalization strategy is based on Coleman–Li trust regions, which are known to take into account efficiently the distance between the current point and the boundary.

Global convergence of the sequence generated by our method towards stationary points of the squared norm of the system will be proved. Under suitable assumptions we will also show that, if a limit point is interior, the whole sequence converges quadratically to this point. Although the convergence results are not restricted to interior points, the result related to interior limit points is stronger than the one related to limit points on the boundary. The reason is that we have in mind the application of the method as a subalgorithm for finding feasible points in nonlinear programming algorithms [12,13]. In these cases (usually \( m < n \)), interior points are preferred as basic points to start the optimality phase of the algorithms. If interior-point solutions do not exist, usual constraint qualifications are not satisfied and the nonlinear programming method generally fails. In other words, if all the stationary points of the squared norm of the system are solutions (interior or not) of the system, the limit points of the sequence will be solutions (perhaps noninterior). But we certainly prefer that the whole sequence does not converge to the boundary, due to the main applications of the method. A slow convergence to the boundary is better since it allows us to take a more interior inexact solution which, in general, is well suited for the application.

A method due to Kanzow et al. [9] is used here for numerical comparisons. This method applies to the solution of nonlinear systems with convex constraints. A first method introduced in [9] needs the resolution of rather hard subproblems (even in the bound-constrained case) but the second method, which is the one that we use for comparisons, has a fairly simple implementation. In this method a Levenberg–Marquardt step (that does not take into account the constraints at all) is computed at each iteration and then projected on the feasible region. If the norm of the system at the projected point is a fraction of the norm of the system at the current point, the projected Levenberg–Marquardt point is taken as new iterate. If the projected Levenberg–Marquardt point generates a descent direction, this direction is used for a line search. Otherwise, the method uses a projected gradient procedure to reduce the objective function. Global convergence in the sense that every limit point is stationary can be obtained using standard arguments and, under fairly weak assumptions, local and quadratic convergence is proved in [9].

This paper is organized as follows. Algorithm 2.1 is presented in Section 2 and convergence results are given in Section 3. The algorithm is tested against the projected Levenberg–Marquardt algorithm [9] and the numerical results are presented in Section 4. Finally some conclusions are drawn in Section 5.

**Notation:** \( A^\dagger \) denotes the Moore–Penrose pseudoinverse of the matrix \( A \). \( G_k \) means \( G(x^k) \). \( \| \cdot \| \) denotes the Euclidian norm of vectors and its subordinate matrix norm. \( \| \cdot \|_p \) denotes the \( p \)-norm. The \( i \)th coordinate of a point will be denoted \([x]_i\). If there is no place to confusion, we will denote also \( x_i = [x]_i \). Given \( v \in \mathbb{R}^n \), diag\((v)\) denotes the diagonal \( n \times n \) matrix whose diagonal terms are the \( v_1, \ldots, v_n \). \( B(y, \rho) \) is the Euclidian ball with center \( y \) and radius \( \rho \). \( \text{Int}(\Omega) \) denotes the interior of \( \Omega \). \( J(x) \) is the Jacobian matrix of \( F \) at the point \( x \). For all \( z \in \mathbb{R}^n \), \( P_\Omega(z) \) denotes the Euclidian projection on \( \Omega \).

We denote \( f_k = f(x_k), \nabla f_k = \nabla f(x_k), F_k = F(x_k) \).

2. Description of the algorithm

Let us define the natural merit function associated with (1):

\[
f(x) = \frac{1}{2} \| F(x) \|^2.
\]  (3)
As it is well known
\[ \nabla f(x) = J(x)^T F(x). \]

The quadratic model at each iteration is given by
\[
m_k(p) = \frac{1}{2} \| J(x^k) p + F(x^k) \|^2 \\
= \frac{1}{2} \| F(x^k) \|^2 + F(x^k)^T J(x^k) p + \frac{1}{2} p^T J(x^k)^T J(x^k) p.
\]

At the \( k \)th iteration, the trust region is defined by
\[ \| D_k p \| \leq \Delta, \]
where \( \Delta > 0 \) is the trust-region radius and \( D_k \equiv D(x^k) \) is the affine-scaling matrix introduced below.

Given \( x \in \text{Int}(\Omega) \), the affine-scaling matrix \( D(x) \) is defined in the following way
\[
D(x) = \begin{bmatrix}
|v_1(x)|^{-1/2} & \cdots & \cdot \\
\cdot & \cdot & \cdot \\
|v_n(x)|^{-1/2}
\end{bmatrix} \\
= \text{diag}(|v_1(x)|^{-1/2}, \ldots, |v_n(x)|^{-1/2}), \tag{5}
\]
where
\[
v_i(x) = \begin{cases} 
  x_i - u_i & \text{if } \nabla f(x)_i < 0 \text{ and } u_i < \infty, \\
  x_i - l_i & \text{if } \nabla f(x)_i \geq 0 \text{ and } l_i > -\infty, \\
  -1 & \text{if } \nabla f(x)_i < 0 \text{ and } u_i = \infty, \\
  1 & \text{if } \nabla f(x)_i \geq 0 \text{ and } l_i = -\infty.
\end{cases} \tag{6}
\]

Observe that \( D(x) \) is not defined on the boundary of \( \Omega \) but \( D(x)^{-1} \) can be extended continuously to it. This extension will be also denoted \( D(x)^{-1} \). So,
\[
D(x)^{-1} = \begin{bmatrix}
|v_1(x)|^{1/2} & \cdots & \cdot \\
\cdot & \cdot & \cdot \\
|v_n(x)|^{1/2}
\end{bmatrix},
\]
for all \( x \in \Omega \), where \( v_i(x) \) is given by (6). Clearly, \( D(x)^{-1} \) is continuous for all \( x \in \Omega \).

From now on we will consider the optimization problem associated with (1)
\[
\min f(x) \\
\text{s.t. } x \in \Omega. \tag{7}
\]

The first-order optimality conditions of (7) are
\[
\begin{cases} 
  \nabla f(x)_i = 0 & \text{if } l_i < x_i < u_i, \\
  \nabla f(x)_i \leq 0 & \text{if } x_i = u_i, \\
  \nabla f(x)_i \geq 0 & \text{if } x_i = l_i.
\end{cases} \tag{8}
\]

Points that satisfy these conditions are called stationary. In [5, Lemma 2.3]) the following lemma has been proved.

**Lemma 2.1.** Let \( x \in \Omega \). Then \( D(x)^{-1} \nabla f(x) = 0 \) if and only if \( x \) is stationary.
Given an iterate $x^k \in \text{Int}(\Omega)$ and a direction $p \in \mathbb{R}^n$, the definitions below are necessary to find a new interior iterate.

Define

$$
\lambda(p) = \arg \max\{t \geq 0 \mid x^k + tp \in \Omega\}. \tag{9}
$$

Observe that, if $\lambda(p) \leq 1$, one has that $x^k + p \notin \text{Int}(\Omega)$ and the step must be reduced. Otherwise, $x^k + p \in \text{Int}(\Omega)$. For given $\theta \in (0, 1)$, we define

$$
\xi(p) = \begin{cases} 1 & \text{if } \lambda(p) > 1, \\ \max\{\theta, 1 - \|p\|\} \lambda(p) & \text{otherwise}. \end{cases} \tag{10}
$$

Defining

$$
z(p) = \xi(p)p, \tag{11}
$$

we see that $x^k + z(p) \in \text{Int}(\Omega)$. In our approach we need to generate interior points because we want to define trust regions with enough space for decreasing the objective function. So, since $x^k + p$ might be infeasible or on the boundary, $z(p)$ is a safely interior multiple of $p$.

The *scaled steepest descent direction* $d^k$ is defined by

$$
d^k = -D_k^{-2}\nabla f(x^k). \tag{12}
$$

Given a trust-region radius $\Delta > 0$, the Cauchy point will be defined as the vector $p_C^k$ that minimizes the model $m_k$ (4) along $d^k$ restricted to the trust region. So,

$$
p_C^k = \tau^k d^k = -\tau^k D_k^{-2}\nabla f_k, \tag{13}
$$

where

$$
\tau^k = \arg \min_{\tau > 0} \{m_k(\tau d^k) \mid \|\tau D_k d^k\| \leq \Delta\}. \tag{14}
$$

Note that $x^k + p_C^k$ might not belong to $\Omega$.

The Cauchy steplength $\tau^k$ can be computed explicitly, as stated in the proposition below. See the proof in [17, p. 70].

**Proposition 2.1.** Let $p_C^k$ be the Cauchy direction (13). Then,

$$
\tau^k = \min \left\{ \frac{\|D_k^{-1}\nabla f_k\|^2}{\|J_k D_k^{-2}\nabla f_k\|^2}, \frac{\Delta}{\|D_k^{-1}\nabla f_k\|^2} \right\}. \tag{15}
$$

For given $p \in \mathbb{R}^n$ we define, as usually in trust-region methods, the *actual* and the *predicted* reductions $\text{ared}(p)$ and $\text{pred}(p)$ by

$$
\begin{align*}
\text{ared}(p) &= f(x^k) - f(x^k + z(p)), \\
\text{pred}(p) &= m_k(0) - m_k(z(p)).
\end{align*}
$$
A step \( p \) will be admissible if \( z(p) \) produces a sufficient reduction with respect to the reduction attained by the truncated Cauchy step \( z(p^k_C) \). In other words, acceptance of \( p \) is subject to the condition

\[
\rho^k_C(p) = \frac{\text{pred}(p)}{\text{pred}(p^k_C)} = \frac{m_k(0) - m_k(z(p^k_C))}{m_k(0) - m_k(z(p^k_C))} \geq \beta_1,
\]

where \( \beta_1 \in (0, 1) \) is given. In Section 4 (implementation features), we explain how we compute \( p \) in practice. Observe that the trivial choice \( p = p^k_C \) is clearly admissible. By (14), we consider that a step is admissible if it produces a sufficient reduction in the value of the quadratic model with respect to the reduction attained by the truncated Cauchy step. This is a classical requirement that appears in different ways in numerical optimization. The actual reduction of the objective function, in turn, will be required to be of the same order as the reduction predicted by the search direction and, so, it will be proportional to the reduction predicted by the Cauchy direction. This implies that, in the worst case, the method will behave as a Cauchy-like method.

Each trial point \( x^k + z(p) \) will be accepted if and only if the actual reduction provided by \( p \) is big enough in comparison to the predicted reduction. This condition is

\[
\rho^k_j(p) = \frac{\text{ared}(p)}{\text{pred}(p)} = \frac{f(x^k) - f(x^k + z(p))}{m_k(0) - m_k(z(p^k_C))} \geq \beta_2,
\]

where \( \beta_2 \in (0, 1) \) is given.

The choice of the search direction will require an additional definition. We define the *Newtonian minimum norm direction* \( p^k_N \) by

\[
p^k_N = -J(x^k)^\dagger F(x^k).
\]

Now we are able to define the main algorithm.

**Algorithm 2.1.** Let \( x^0 \in \text{Int}(\Omega), A_{\min} > 0, \Delta > A_{\min} \) and \( \theta, \beta_1, \beta_2, \delta_0, \delta_1 \in (0, 1) \) such that \( \delta_0 < \delta_1 \).

Set \( k \leftarrow 0 \).

Step 1: Compute \( J_k, F_k \) and \( D_k \).

Step 2: If \( \|\nabla f(x^k)\| = 0 \) terminate the execution of the algorithm. In this case \( x^k \) is a stationary point of \( \min_{x \in \Omega} f(x) \).

Step 3: Compute

\[
p^k_N = -J(x^k)^\dagger F(x^k).
\]

Step 4: If

\[
\|D_k p^k_N\| \leq \Delta \quad \text{and} \quad \rho^k_C(p^k_N) \geq \beta_1,
\]

define \( p^k = p^k_N \).

Else, find \( p^k \) such that \( \|D_k p^k\| \leq \Delta \) and \( \rho^k(p^k) \geq \beta_1 \), with \( p^k \) given by (13).

Step 5: If

\[
\rho^k_j(p^k) \geq \beta_2
\]

compute \( x^{k+1} = x^k + z(p^k) \) and go to Step 6.
Else, choose $\Delta_{\text{new}} \in [\delta_0 \Delta, \delta_1 \Delta]$, set $\Delta \leftarrow \Delta_{\text{new}}$ and go to Step 4.

Step 6: Set $k \leftarrow k + 1$, choose $\Delta > \Delta_{\text{min}}$ and go to Step 1.

3. Convergence results

In order to prove the convergence results for Algorithm 2.1 we will use the following assumptions.

H1. The sequence $\{x^k\}$ generated by the algorithm is bounded (A sufficient condition for this is that the level set

$$\{x \in \Omega \mid f(x) \leq f(x^0)\}$$

is bounded.).

H2. For all $x, y$ in an open, bounded and convex set $L$ that contains the whole sequence generated by the algorithm and all the points of the form $x^k + \alpha(p^k)$, we have that

$$\|J(x) - J(y)\| \leq 2\gamma_0 \|x - y\|. \tag{19}$$

(Clearly, a sufficient condition for this is that (19) holds for all $x, y \in \Omega$.)

H3. $J(x)$ has full rank $m$ for all $x \in L$.

Note that since $F$ is continuously differentiable we also have that there exists $\gamma_1 > 0$ such that for all $x, y \in L$

$$\|F(x) - F(y)\| \leq \gamma_1 \|x - y\|. \tag{20}$$

Clearly, by H1 and H3, there exists $\mu > 0$ such that

$$\|J(x)^{\frac{1}{2}}\| = \|J(x)^{\frac{1}{2}}(J(x)J(x)^{\frac{1}{2}})^{-1}\| \leq \mu \tag{21}$$

for all $x \in L$.

Moreover, by H2, we have that for all $x, y \in \Omega$,

$$\|F(x) - F(y) - J(y)(x - y)\| = \left\| \int_0^1 [J(y + t(x - y)) - J(y)](x - y) \, dt \right\|
\leq 2\gamma_0 \left( \int_0^1 t \, dt \right) \|x - y\|^2 = \gamma_0 \|x - y\|^2. \tag{22}$$

The proof of the following lemma is essentially the one of Lemma 3.1 of [2].

**Lemma 3.1.** Assume that H2 holds and suppose that $p \in \mathbb{R}^n$ is such that

$$\|J(x^k)\alpha(p) + F(x^k)\| \leq \|F(x^k)\|.$$

Then

$$|\text{ared}(p) - \text{pred}(p)| \leq \varepsilon^k(p)\|\alpha(p)\|^2,$$
where
$$\varepsilon^k(p) = \gamma_0 \| F(x^k) \| + \frac{1}{2} \gamma_0^2 \| z(p) \|^2.$$

**Proof.** By the Mean-Value Theorem we have:
$$F(x^k + z(p)) = F(x^k) + \int_0^1 J(x^k + \zeta z(p)) z(p) \, d\zeta.$$

Define
$$s_k = \int_0^1 (J(x^k + \zeta z(p)) - J(x^k)) z(p) \, d\zeta.$$

Note that $$\| s_k \| \leq \gamma_0 \| z(p) \|^2.$$ So,

$$2 | \text{ared}(p) - \text{pred}(p) | = 2|m_k(z(p)) - f(x^k + z(p))|$$
$$= | \| F(x^k) + J(x^k)z(p) \|^2 - \| F(x^k) + J(x^k)z(p) + s_k \|^2 |$$
$$\leq 2 \| F(x^k) + J(x^k)z(p) \| \| s_k \| + \| s_k \|^2$$
$$\leq 2 \gamma_0 \| F(x^k) \| \| z(p) \|^2 + \gamma_0^2 \| z(p) \|^4.$$

Therefore,
$$| \text{ared}(p) - \text{pred}(p) | \leq \varepsilon^k(p) \| z(p) \|^2$$
and the thesis follows straightforwardly. □

Now we state a technical lemma. Its proof is quite similar to the ones of [2, Lemma 3.3] and [5, Lemma 3.1], with small modifications.

**Lemma 3.2.** If $$p^k$$ is such that $$\rho^k_c(p^k) \geq \beta_1,$$ then
$$\text{pred}(p^k) \geq \frac{1}{2} \beta_1 D_k^{-1} \nabla f_k \min \left\{ A, \frac{\| D_k^{-1} \nabla f_k \|}{\| D_k^{-1} J_k^T J_k D_k^{-1} \|}, \frac{\theta \| D_k^{-1} \nabla f_k \|}{\| \nabla f_k \|_{\infty}} \right\}, \tag{23}$$

where $$\theta$$ is the constant used in (10).

The next lemma says that the algorithm is well defined. Its proof follows closely the lines of Lemma 3.4 of [2].

**Lemma 3.3.** Assume that H2 is fulfilled. If $$\| \nabla f_k \| \neq 0,$$ then the loop that defines an iteration of Algorithm (2.1) finishes after a finite number of cycles.

**Proof.** It suffices to prove that, after a finite number of step reductions, the algorithm finds a direction $$p^k$$ such that $$\rho^k_f(p^k) \geq \beta_2$$ holds for some $$\Delta$$ small enough.

Assume that
$$\Delta \leq \min \left\{ \frac{\| D_k^{-1} \nabla f_k \|}{\| D_k^{-1} J_k^T J_k D_k^{-1} \|}, \frac{\theta \| D_k^{-1} \nabla f_k \|}{\| \nabla f_k \|_{\infty}} \right\},$$
So by Lemma 3.2,
\[ A \leq \tilde{C}_k \text{pred}(p^k), \]
where \( \tilde{C}_k = 2/(\beta_1 \|D_k^{-1} \nabla f_k\|) \). Since \( \|D_k p^k\| \leq A \), we get
\[ \|z(p^k)\| \leq \|p^k\| \leq \|D_k^{-1} A\| \leq \|D_k^{-1} \tilde{C}_k \text{pred}(p^k)\|. \] (24)

As \( m_k(z(p^k)) \leq m_k(0) \), by Lemma 3.1, the Eq. (24) and the fact that \( \|z(p^k)\| \leq \|D_k^{-1} A\| \) imply:
\[ |\text{ared}(p^k) - \text{pred}(p^k)| \leq \varepsilon^k(p^k) \|z(p^k)\|^2 \leq \varepsilon^k(p^k) \|D_k^{-1} \tilde{C}_k \text{pred}(p^k)\|. \]

Thus,
\[ |\rho^k_J(p^k) - 1| \leq \varepsilon^k(p^k) \|D_k^{-1} \tilde{C}_k A\|. \]

But
\[ \varepsilon^k(p^k) \leq c_0 \gamma_0 F_k + \frac{1}{2} \gamma_0^2 \|D_k^{-1} A\|, \]
so
\[ \lim_{A \to 0} |\rho^k_J(p^k) - 1| = 0. \]

Therefore, there exists \( A^* \) such that \( \rho^k_J(p^k) \geq \beta_2 \) for all \( A \leq A^* \). Taking
\[ A \leq \min \left\{ A^*, \frac{\|D_k^{-1} \nabla f_k\|}{\|D_k^{-1} J_k J_k^T D_k^{-1}\|}, \frac{\theta \|D_k^{-1} \nabla f_k\|}{\|\nabla f_k\|_{\infty}} \right\} \]
the condition \( \rho^k_J(p^k) \geq \beta_2 \) is satisfied and the proof is complete. \( \square \)

The following lemma, whose proof is straightforward, will be used in the convergence proof.

**Lemma 3.4.** Assume that \( H_1 \) holds. Then, there exists \( \chi_D > 0 \) such that
\[ \|D(x^k)^{-1}\| \leq \chi_D. \] (25)

Now we give the global convergence result.

**Theorem 3.1.** Assume that \( H_1 \) and \( H_2 \) are fulfilled and that the algorithm generates an infinite sequence \( \{x^k\} \). Then all the limit points are stationary points of
\[ \min_{x \in \Omega} f(x). \]

**Proof.** Let \( x^* \in \Omega \) be a limit point of \( \{x^k\} \). Then there exists \( \mathcal{N}_1 \), an infinite subset of \( \{0, 1, 2, \ldots\} \), such that \( x^k \to x^* \) for \( k \in \mathcal{N}_1 \).
First we are going to prove that
\[
\lim_{k \in \mathcal{N}_1} \| D_k^{-1} \nabla f_k \| = 0.
\] (26)

Assume that (26) is not true. Then there exists \( \varepsilon > 0 \) such that \( \| D_k^{-1} \nabla f_k \| \geq \varepsilon \) for an infinite set of indices \( \mathcal{N}_2 \subseteq \mathcal{N}_1 \).

By Lemma 3.2 and (15) we deduce that
\[
\text{ared}(p^k) = f(x^k) - f(x^{k+1}) \geq \beta_2 \text{pred}(p^k) \geq \frac{1}{2} \beta_1 \beta_2 \| D_k^{-1} \nabla f_k \| \min \left\{ A_k, \frac{\| D_k^{-1} \nabla f_k \|}{\| D_k^{-1} J_k J_k^T D_k^{-1} \|}, \frac{\theta \| D_k^{-1} \nabla f_k \|}{\| \nabla f_k \|_{\infty}} \right\},
\] (27)

for all \( k \in \mathcal{N}_1 \). By H1, there exists \( \lambda_g \) such that \( \| J(x)^T J(x) \| \leq \lambda_g \) and \( \| \nabla f(x) \|_{\infty} \leq \lambda_f \) for all \( x \in L \).

Since \( D(x)^{-1} \) is bounded in \( L \), there exists \( \lambda_f > 0 \) such that \( \| D_k^{-1} J_k J_k^T D_k^{-1} \| \leq \lambda_f \) for all \( k \in \mathcal{N}_1 \). Then by (27),
\[
f(x^k) - f(x^{k+1}) \geq \frac{1}{2} \beta_1 \beta_2 \varepsilon \min \left\{ A_k, \frac{\varepsilon}{\lambda_f}, \frac{\theta \varepsilon}{\lambda_g} \right\},
\] (28)

for all \( k \in \mathcal{N}_2 \). Since \( \{ f(x^k) \}_{k \in \{0,1,2,...\}} \) is monotone nonincreasing and bounded below,
\[
\lim_{k \to \infty} (f(x^k) - f(x^{k+1})) = 0.
\]

Let \( A_k \) be the final trust region radius defined at iteration \( k \). By (28), we have that
\[
\lim_{k \in \mathcal{N}_2} A_k = 0.
\]

Now, at each iteration \( k \) the first trial trust region radius is strictly greater than \( A_{\min} \). Thus, the fact that \( A_k \to 0 \) for \( k \in \mathcal{N}_2 \) implies that, for \( k \in \mathcal{N}_2 \) large enough, there exist \( \bar{A}_k \) and \( \bar{p}^k \equiv \bar{p}(\bar{A}_k) \) such that
\[
\lim_{k \to \infty} \bar{A}_k = 0, \rho_C(\bar{p}^k) \geq \beta_1, \| D_k \bar{p}^k \| \leq \bar{A}_k \quad \text{and} \quad \rho_f(\bar{p}^k) < \beta_2.
\]

By Lemma 3.4, we have that \( \| D_k^{-1} \| \leq \lambda_D \). So,
\[
\| \tilde{x}(\bar{p}^k) \| \leq \| \bar{p}^k \| \leq \lambda_D \| D_k \bar{p}^k \| \leq \lambda_D \bar{A}_k.
\]

By Lemma 3.1, we have:
\[
| \text{ared}(\bar{p}^k) - \text{pred}(\bar{p}^k) | \leq \varepsilon k(\bar{p}^k) \lambda_D^2 \bar{A}_k^2 = \eta_k \bar{A}_k,
\]

where \( \eta_k = \varepsilon k(\bar{p}^k) \lambda_D^2 \bar{A}_k^2 \). Moreover, by Lemma 3.2 and the contradiction hypothesis, for \( k \in \mathcal{N}_2 \) large enough we have that
\[
\text{pred}(\bar{p}^k) \geq \frac{1}{2} \beta_1 \varepsilon \bar{A}_k,
\]

so
\[
| \text{ared}(\bar{p}^k) - \text{pred}(\bar{p}^k) | \leq \frac{2}{\beta_1 \varepsilon} \eta_k \text{pred}(\bar{p}^k).
\]
and, therefore,
\[ |\rho_k^f(\bar{p}^k) - 1| \leq \frac{2}{\beta_1} \eta_k, \]
for \( k \in \mathcal{N}_2 \) large enough.
Since \( \{\varepsilon_k(\bar{p}^k)\}_k \) is bounded, it follows that \( \lim_{k \in \mathcal{N}_2} \eta_k = 0 \) and
\[ \lim_{k \in \mathcal{N}_2} |\rho_k^f(\bar{p}^k) - 1| = 0. \]
This contradicts the fact that \( \rho_k^f(\bar{p}^k) < \beta_2 \) for \( k \in \mathcal{N}_2 \) large enough. Therefore,
\[ \lim_{k \in \mathcal{N}_1} \|D_k^{-1}\nabla f_k\| = 0. \]
By Lemma 2.1 the proof is complete. □

**Corollary 3.1.** Under the assumptions of Theorem 3.1 and H3, if a limit point \( x^* \) belongs to \( \text{Int}(\Omega) \) then \( F(x^*) = 0 \).

**Proof.** By Theorem 3.1, \( x^* \) is a stationary point of the problem (7). But, since \( x^* \in \text{Int}(\Omega) \),
\[ J(x^*)^T F(x^*) = 0. \]
Then, by H3, \( F(x^*) = 0 \). □

The following auxiliary lemma will be used in the quadratic convergence proof.

**Lemma 3.5.** Let \( z \in \text{Int}(\Omega) \). Then there exist \( r > 0 \) and \( \mathcal{D}_1 > 0 \) such that \( \|D(x)\| < \mathcal{D}_1 \) for all \( x \in \mathcal{B}(z, r) \subset \text{Int}(\Omega) \).

**Proof.** Since \( z \in \text{Int}(\Omega) \), there exists \( r \in (0, 1] \) such that \( \mathcal{B}(z, 2r) \subset \text{Int}(\Omega) \). Define \( \mathcal{D}_1 = \sqrt{1/r} \). Then, for all \( x \in \mathcal{B}(z, r) \),
\[ |l_i - x_i|, |u_i - x_i| > r \quad \text{for } i = 1, \ldots, n. \]
So, by (5), it follows that \( \|D(x)\| < \sqrt{1/r} = \mathcal{D}_1 \). □

In the following lemmas we prove that the minimum-norm Newtonian step \( p_N^k \) is necessarily accepted in a neighborhood of an interior solution \( x^* \).

**Lemma 3.6.** Assume that H1–H3 are fulfilled, \( K \) is an infinite sequence of indices such that
\[ \lim_{k \in K} x^k = x^* \in \text{Int}(\Omega) \]
and
\[ F(x^*) = 0. \]
Then there exists \( k_0 \in \{0, 1, 2, \ldots\} \) such that for \( k \geq k_0 \), \( k \in K \), the Newton step \( p_N^k \) given by (16) satisfies (17) and (18).
Proof. By the continuity of $F$,

$$\lim_{k \in K} \| F_k \| = \| F(x^*) \| = 0. \quad (29)$$

Since $\{x^k\}_{k \in K}$ is bounded and $x^* \in \text{Int}(\Omega)$, then by Lemma 3.5 there exists $\varnothing > 0$ such that $\| D_k \| \leq \varnothing$ for all $k \in K$. So,

$$\| D_k p_N^k \| \leq \| D_k \| \| - J(x^k)^T F_k \| \leq \mu \varnothing \| F_k \|$$

and

$$\lim_{k \in K} \| D_k p_N^k \| = 0.$$

Then, there exists $k_1 \in \{0, 1, 2, \ldots \}$ such that for $k \geq k_1$, $k \in K$,

$$\| D_k p_N^k \| \leq \lambda_{\min}.$$

By (29) and (21), we have that

$$\lim_{k \in K} \| p_N^k \| \leq \mu \lim_{k \in K} \| F_k \| = 0.$$

By (9) and (10), since $x^* \in \text{Int}(\Omega)$, there exists $k_2 \in \{0, 1, 2, \ldots \}$ such that $\lambda(p_N^k) > 1$ for $k \geq k_2$, $k \in K$.

Consequently, $\zeta(p_N^k) = 1$ and $\alpha(p_N^k) = p_N^k$ for $k \geq k_2$, $k \in K$.

Define $\tilde{k} = \max\{k_1, k_2\}$. Note that, for all $k \geq \tilde{k}$, $k \in K$,

$$\text{pred}(p_N^k) = m_k(0) - m_k(p_N^k) = \frac{1}{2} \| F_k \|^2 \quad (30)$$

and

$$\text{pred}(p_C^k) \leq m_k(0) = \frac{1}{2} \| F_k \|^2.$$

So $\rho_C^k(p_N^k) \geq \beta_1$.

Now, since $\lim_{k \in K} \| F_k \| = \lim_{k \in K} \| p_N^k \| = 0$, defining $\epsilon^k(p)$ as in Lemma 3.1 we have

$$\lim_{k \in K} \epsilon^k(p_N^k) = 0.$$

Since $\| J_k p_N^k + F_k \| = 0 \leq \| F_k \|$, by Lemma 3.1 we have that

$$| \text{ared}(p_N^k) - \text{pred}(p_N^k) | \leq \epsilon^k(p_N^k) \| p_N^k \|^2.$$

Dividing by $\text{pred}(p_N^k)$ and using (30), we obtain

$$\left| \frac{\text{ared}(p_N^k)}{\text{pred}(p_N^k)} - 1 \right| \leq 2 \epsilon^k(p_N^k) \| p_N^k \|^2 \| F_k \|^2,$$

for all $k \geq \tilde{k}$, $k \in K$. 

But $\|p_N^k\| \leq \mu \|F_k\|$, so for $k \geq \tilde{k}$,
\[
|\rho_f^k(p_N^k) - 1| \leq 2\mu^2 \varepsilon^k(p_N^k).
\]
Since $\varepsilon^k(p_N^k) \to 0$, this implies that
\[
\lim_{k \in K} \rho_f^k(p_N^k) = 1.
\]
Therefore, there exists $k_0 \geq \tilde{k}$ such that $\rho_f^k(p_N^k) \geq \beta_2$ for all $k \geq k_0, k \in K$. This completes the proof. □

**Lemma 3.7.** Assume that H1–H3 are fulfilled, the interior point $x^*$ is a limit point of the sequence $\{x^k\}$ and $F(x^*) = 0$.

Then, there exists $\varepsilon > 0$ such that, whenever $\|x^k - x^*\| \leq \varepsilon$, the Newton step $p_N^k$ satisfies (17) and (18).

**Proof.** Assume that the thesis is not true. Then, for all $\varepsilon > 0$ there exists an iterate $x_k$ ($k$ depending on $\varepsilon$) such that $\|x^k - x^*\| \leq \varepsilon$ and at least one of the conditions (17)–(18) is not fulfilled by $p_N^k$. We consider two possibilities:

(a) There exists $\varepsilon > 0$ such that the number of iterates for which $\|x^k - x^*\| \leq \varepsilon$ and at least one of the conditions (17)–(18) is not fulfilled by $p_N^k$ is finite.

(b) For all $\varepsilon > 0$ there exist infinitely many iterates such that $\|x^k - x^*\| \leq \varepsilon$ and at least one of the conditions (17)–(18) is not fulfilled by $p_N^k$.

Consider the Case (a). Let $\bar{x}^k$ be the closest iterate to $x^*$ such that $\|x^k - x^*\| \leq \varepsilon$ and at least one of the conditions (17)–(18) is not fulfilled by $p_N^k$. Take $\bar{\epsilon} = \|x^k - x^*\|/2$. Then all the iterates that satisfy $\|x^k - x^*\| \leq \bar{\epsilon}$ are such that $p_N^k$ fulfills (17)–(18). This is precisely the thesis of the Lemma.

So, we only need to consider Case (b). Let us construct a subsequence of $\{x^k\}$ as follows: $x_j^{(1)}$ is such that $\|x_j^{(1)} - x^*\| \leq 1$ and at least one of the conditions (17)–(18) is not fulfilled by $p_N^{j(1)}$;

For $k > 1$, $x_j^{(k)}$ is such that $\|x_j^{(k)} - x^*\| \leq 1/k$, at least one of the conditions (17)–(18) is not fulfilled by $p_N^{j(k)}$ and $j(k) > j(k - 1)$. The fact that there exist infinitely many iterates whose distance to $x^*$ is smaller than $1/k$ and not satisfying at least one of the conditions (17)–(18) guarantees that it is possible to choose $j(k) > j(k - 1)$.

By construction, the sequence $\{x_j^{(k)}\}$ is a subsequence of $\{x^k\}$, converges to $x^*$ and each one of its elements does not satisfy at least one of the conditions (17)–(18). This contradicts Lemma 3.6. □

**Lemma 3.8.** Assume that H1–H3 hold and let $x^* \in \text{Int}(\Omega)$ be such that $F(x^*) = 0$. Let $\varepsilon > 0$ as given by the thesis of Lemma 3.7. Then there exists $\delta > 0$ such that if $x^{k_0} \in \mathcal{B}(x^*, \delta)$ the sequence $\{x^k\}$ generated by the main algorithm satisfies

\[
x^k \in \mathcal{B}(x^*, \varepsilon)
\]

and
\[
x^{k+1} = x^k + p_N^k
\]

for all $k \geq k_0$. 
Proof. Define
\[ c_1 = \gamma_0 \mu, \]
\[ \delta = \min \left\{ \frac{\varepsilon}{2(1 + 2\mu \gamma_1)}, \frac{1}{2c_1 \mu \gamma_1} \right\}, \]
where \( \gamma_1 \) is defined in (20).
Let \( x_{k_0} \in \mathcal{B}(x^*, \delta) \). We are going to prove by induction that for all \( k \geq k_0 \),
\[ x^k \in \mathcal{B}(x^*, \varepsilon). \]
Since \( \delta \leq \varepsilon \), (32) is obviously true for \( k = k_0 \).
Assume, as inductive hypothesis, that
\[ x^j \in \mathcal{B}(x^*, \varepsilon), \quad j = k_0, k_0 + 1, \ldots, k. \]
We wish to prove that \( x^{k+1} \in \mathcal{B}(x^*, \varepsilon) \). By Lemma 3.7 and the inductive hypothesis,
\[ x^{j+1} = x^j + p_N^j = x^j - J(x^j)^\dagger F(x^j), \quad j \in \{k_0, k_0 + 1, \ldots, k\}. \]
By (21),
\[ \|p_N^j\| \leq \mu \|F(x^j)\| \]
for all \( j \in \{k_0, k_0 + 1, \ldots, k\} \).
Since \( J(x^j)J(x^j)^\dagger = I_{m \times m} \), by (22) we have that
\[ \|F(x^{j+1})\| = \|F(x^{j+1}) - F(x^j) - J(x^j)p_N^j\| \leq \gamma_0 \|p_N^j\|^2 \]
for all \( j \in \{k_0, k_0 + 1, \ldots, k\} \).
Then, by (35),
\[ \|p_N^j\| \leq \gamma_0 \mu \|p_N^{j-1}\|^2 = c_1 \|p_N^{j-1}\|^2 \]
for all \( j \in \{k_0 + 1, \ldots, k\} \).
Now, by the inductive hypothesis and (34),
\[ \|x^{k+1} - x^*\| = \|x^k + p_N^k - x^*\| \leq \|x^k - x^*\| + \|p_N^k\| \]
\[ = \|x^{k-1} + p_N^{k-1} - x^*\| + \|p_N^k\| \]
\[ \leq \|x^{k-1} - x^*\| + \|p_N^{k-1}\| + \|p_N^k\| \]
\[ \vdots \]
\[ \leq \|x^{k_0} - x^*\| + \sum_{j=k_0}^k \|p_N^j\|. \]
By (36),
\[
\|p_N^j\| \leq c_1 \|p_N^{j-1}\|^2 \leq c_1 c_1^2 \|p_N^{j-2}\|^2 \\
\vdots \leq c_1 c_1^2 \ldots c_1^{j-k_0-1} \|p_N^{j-k_0}\|^2 \leq c_1^{j-k_0} \|p_N^{j-k_0}\|^2 \\
= c_1^{j-k_0} \|p_N^{j-k_0}\|^2,
\]
for all \(j \in \{k_0 + 1, \ldots, k\}\).

But, by (20) and (21),
\[
\|p_N^{k_0}\| = \|J(x^{k_0})^\top F(x^{k_0})\| \leq \mu \|F(x^{k_0})\| \\
= \mu \|F(x^{k_0}) - F(x^*)\| \leq \mu \gamma_1 \|x^{k_0} - x^*\|.
\]

Since \(\sum_{i=0}^{j-k_0-1} 2^i = 2^{j-k_0} - 1\), by (37) and (38) we obtain
\[
\|x^{k+1} - x^*\| \leq \|x^{k_0} - x^*\| + \sum_{j=k_0}^{k} c_1^{j-k_0} \|p_N^{j-k_0}\|^2 \\
\leq \|x^{k_0} - x^*\| + \sum_{j=k_0}^{k} c_1^{j-k_0-1} (\mu \gamma_1)^{2^{j-k_0}} \|x^{k_0} - x^*\|^{2^{j-k_0}} \\
\leq \delta + \mu \gamma_1 \delta \sum_{j=k_0}^{k} (c_1 \mu \gamma_1 \delta)^{2^{j-k_0}}.
\]

(39)

So, by (31) and (39),
\[
\|x^{k+1} - x^*\| \leq \delta + \mu \gamma_1 \delta \sum_{j=k_0}^{k} \left(\frac{1}{2}\right)^{2^{j-k_0}} \\
\leq \delta \left(1 + \mu \gamma_1 \sum_{j=k_0}^{k} \left(\frac{1}{2}\right)^{j-k_0}\right) \leq \delta (1 + 2 \mu \gamma_1) \\
\leq \frac{\varepsilon}{2}.
\]

Therefore \(x^{k+1} \in B(x^*, \varepsilon)\). This completes the proof. \(\square\)

Now we can state our final convergence result.

**Theorem 3.2.** Assume that the assumptions H1–H3 hold and that \(x^* \in \text{Int}(\Omega)\) is a limit point of the sequence \(\{x^k\}\) generated by Algorithm 2.1. Then \(F(x^*) = 0\) and \(x^k\) converges quadratically to \(x^*\).

**Proof.** The fact that \(F(x^*) = 0\) follows from \(J(x)^T F(x^*) = 0\) and Corollary 3.1. By Lemma 3.8 there exists \(\delta > 0\) such that if \(\|x^{k_0} - x^*\| \leq \delta\) then \(x^{k+1} = x^k + p_N^k\) for all \(k \geq k_0\). Since \(x^*\) is a limit point of
{\{x^k\}} it follows that \(x^{k+1} = x^k + p_N^k\) for all \(k\) large enough. So by Walker and Watson [23, Theorem 2.1], the sequence converges quadratically to some solution. Since \(x^*\) is a limit point, the sequence converges to \(x^*\). This completes the proof. □

4. Numerical experiments

We tested Algorithm 2.1 against a projected Levenberg–Marquardt method (called KYF from now on) for solving nonlinear systems on convex sets proposed recently by Kanzow, Yamashita and Fukushima. See Algorithm 3.12 of [9].

The KYF method is described below.

**Algorithm 4.1 (KYF).** Choose \(x^0 \in \Omega, \mu > 0, \beta, \sigma, \gamma \in (0, 1)\).

Set \(k \leftarrow 0\).

---

**Step 1:** If \(F(x^k) = 0\), terminate the execution of the algorithm.

**Step 2:** Define \(\mu_k = \mu \parallel F_k \parallel^2\) and compute \(d_U^k\) by

\[
(J(x^k)^T J(x^k) + \mu_k I) d_U = -J(x^k)^T F_k.
\]

**Step 3:** If \(\parallel F(P_{\Omega}(x^k + d_U^k)) \parallel \leq \gamma \parallel F_k \parallel\) define \(x^{k+1} = P_{\Omega}(x^k + d_U^k)\), set \(k \leftarrow k + 1\) and go to Step 1. Else, go to Step 4.

**Step 4:** Define \(s_{LM}^k = P_{\Omega}(x^k + d_U^k) - x^k\). If \(\nabla f(x^k)^T s_{LM}^k \leq -\rho \parallel s_{LM}^k \parallel^p\), compute \(t_k = \max\{\beta^i \mid i = 0, 1, \ldots\}\) such that

\[
f(x^k + t_k s_{LM}^k) \leq f(x^k) + t_k \sigma \nabla f(x^k)^T s_{LM}^k.
\]

Define \(x^{k+1} = x^k + t_k s_{LM}^k\), set \(k \leftarrow k + 1\) and go to Step 1. Otherwise, go to Step 5.

**Step 5:** Compute \(t_k = \max\{\beta^i \mid i = 0, 1, \ldots\}\) such that

\[
f(x^k(t_k)) \leq f(x^k) + \sigma \nabla f(x^k)^T (x^k(t_k) - x^k),
\]

where \(x^k(t) = P_{\Omega}(x^k - t \nabla f(x^k))\). Define \(x^{k+1} = x^k(t_k)\), set \(k \leftarrow k + 1\) and go to Step 1.

We implemented Algorithm KYF with different values for the regularization parameter \(\mu\): \(10^{-1}, 10^{-5}\) and \(10^{-7}\). Proceeding as in [9], another update formula of \(\mu_k\), called here \(\mu_\ast\), was tried too: Starting with \(\mu_0 = \frac{1}{2} 10^{-8} \parallel F(x^0) \parallel\), the formula is given by

\[
\mu_k = \min\{\mu_{k-1}, \parallel F(x^k) \parallel^2\}.
\]

This choice does not change the theoretical properties of the algorithm. The remaining parameters were the ones used in [9] (\(\beta = 0.9, \gamma = 0.99995, \rho = 10^{-8}, p = 2.1\) and \(\sigma = 10^{-4}\)). The algorithms were implemented in Matlab 5.3.

We used test problems described in [7] and a set of problems defined by feasible sets of nonlinear programming problems in the book of Hock and Schittkowski [8]. These problems are of the form

\[
F(x) = 0, \quad x \in \Omega,
\]

where \(\Omega\) is a box and \(F : \mathbb{R}^n \rightarrow \mathbb{R}^m\). Table 1 shows the problem data in the following way:

Column 1: Number of the problem;
Table 1
Problems data

<table>
<thead>
<tr>
<th>Problem</th>
<th>Source</th>
<th>m</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Problem 46 in [8]</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>Problem 53 in [8]</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>Problem 56 in [8]</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>Problem 63 in [8]</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>Problem 75 in [8]</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>Problem 77 in [8]</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>7</td>
<td>Problem 79 in [8]</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>8</td>
<td>Problem 81 in [8]</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>9</td>
<td>Problem 87 in [8]</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>10</td>
<td>Problem 107 in [8]</td>
<td>6</td>
<td>9</td>
</tr>
<tr>
<td>11</td>
<td>Problem 109 in [8]</td>
<td>6</td>
<td>9</td>
</tr>
<tr>
<td>12</td>
<td>Problem 111 in [8]</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>13</td>
<td>Problem 2 in [9]</td>
<td>150</td>
<td>300</td>
</tr>
<tr>
<td>14</td>
<td>Problem 4 in [9]</td>
<td>150</td>
<td>300</td>
</tr>
<tr>
<td>15</td>
<td>—</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

Column 2: Source of the problem;
Columns 3 and 4: dimensions $m$ and $n$.

Problems 1, 3, 6 and 7 were, originally, unconstrained. So, we introduced artificial bounds $0 \leq x_i \leq 2.5$ for all $i$. Therefore, only the problems 5, 13 and 14 do not have bound constraints. In the problems 10 and 11 the variables $x_1$ and $x_2$ are unbounded above, and in the problem 4 all the variables are unbounded above. Problem 13 is a linear system and Problem 14 is a quadratic system. We also tested both methods for the problem defined by

$$F(x_1, x_2) = x_2 - \frac{1}{100} x_1,$$

which was called Problem 15.

Let us enumerate some implementation features:

1. For both methods the Jacobian matrix was approximated by finite differences.
2. For both methods $x^k$ was accepted as a solution if $\| F(x^k) \| \leq 10^{-6}$.
3. We allowed each method to perform 5000 iterations and 10 000 function evaluations (not considering the evaluations used to approximate the Jacobian).
4. Algorithm 2.1 was also prepared to stop when $A \leq 10^{-8}$ or when $\| D_k^{-1} \nabla f(x^k) \| < 10^{-10}$ but this never happened in our experiments. If $\| D_k^{-1} \|_\infty \leq 7.45 \times 10^{-155}$ we say that $D_k^{-1}$ is numerically singular and we stop the process.
5. The parameters for Algorithm 2.1 were: $A = \| D(x^0)^{-1} \nabla f(x^0) \|$, $\lambda_{\text{min}} = 5 \times 10^{-4}$, $\theta = 0.99995$, $\beta_1 = 0.1$, $\beta_2 = 0.25$ and $\delta_1 = 0.25$.
6. The choice of $\lambda_{\text{new}}$ at Step 5 of Algorithm 2.1 was

$$\lambda_{\text{new}} = \min\{ \delta_1 A, \frac{1}{2} \| D_k x(p_k) \| \}.$$
7. At Step 6, if $\rho_k^f(p^k) \geq 0.75$ then,

$$A = \max\{A_{\min}, A, 2\|D_k z(p^k)\|\},$$
otherwise

$$A = \max\{A_{\min}, A\}.$$ 

8. At Step 4 of Algorithm 2.1 we need to find a direction $p^k$ satisfying $\rho_C^k(p^k) \geq \beta_1$ and $\|D_k p^k\| \leq A$. With this objective, we used the dogleg method (see [17]). The dogleg direction $p_d^k$ is computed as follows:

$$p_d^k = \begin{cases} -\Delta D^{-2} \nabla f_k, & \text{if } \|D_k p_C^k\| \geq A \\ \frac{\Delta}{\|D_k \nabla f_k\|} \frac{D_k^{-1} \nabla f_k}{\|D_k^{-1} \nabla f_k\|} & \text{otherwise} \\ p_C^k + (\mu - 1)(p_N^k - p_C^k) & \text{otherwise}, \end{cases}$$

where $\mu$ is the positive solution of

$$\|D_k(p_C^k + (\mu - 1)(p_N^k - p_C^k))\|^2 = A^2.$$ 

If $p_d^k$ does not satisfy the condition $\rho_C^k(p_d^k) \geq \beta_1$, we choose $p^k = p_C^k$. In this way we guarantee that the step that satisfies (14) is necessarily obtained.

9. For finding the minimum-norm Newton direction we used the QR factorization.

The strictly feasible initial approximation $x^0$ was chosen to be $x^0 = (l + u)/2$ except in the following situations.

1. In Problems 2 and 8 we took $x^0 = l + \frac{1}{4}(u - l)$, because the initial point above is a stationary point of $\min_{x \in \Omega} f(x)$.
2. In Problems 13 and 14 we took $x^0 = 150(1, \ldots, 1)^T$, which is one of the initial guesses suggested in [9].
3. In Problem 15 we consider $x_0 = (-\frac{1}{2}, \frac{1}{2})$

In Table 2 we present the results of Algorithm 2.1 for this set of problems. The fourth column shows the norm of $F$ evaluated on the respective initial approximation. The last column shows the number of iterations in which the projected gradient direction ($p_C^k$), the dogleg direction ($p_d^k$) and the Newton direction ($p_N^k$) were accepted, respectively. The notation MAXIT indicates that the allowed number of iterations was exhausted. MAXFUN indicates that the allowed number of evaluations was exhausted and SINGUL indicates that $D_k^{-1}$ is nearly singular.

In Table 3 we compare Algorithm 2.1 with Algorithm 4.1. In the sixth column we display the number of Levenberg–Marquardt (LM), Line-Search (LS) and Cauchy directions accepted by Algorithm 4.1, respectively.
5. Conclusions

We introduced a new algorithm for solving nonlinear systems of equations with bounded variables. Our main motivation is the “feasibility phase” of nonlinear programming algorithms based on periodic restoration. In this case, most solutions are interior points. Moreover, one is often interested only in interior solutions. With this in mind, our algorithm may be interpreted as a globalization of the normal-flow method for solving underdetermined systems. The strategy for globalization is essentially the one introduced in [5] and adapted in [2] for the resolution of bounded square nonlinear systems. A limited number of numerical experiments show that the algorithm behaves as expected. In most iterations, it reduces to the normal-flow method and, when it does not, the global strategy is able to lead the iterates to a solution. For most cases, quadratic convergence was “observed” in practice, in the vicinity of interior solutions. The KYF algorithm [9] can also be considered a globalization of the normal-flow method, at least when the regularization parameter is very small, which corresponds to the situation in which the best results were observed. In general, the rather simple globalization strategy of KYF, based on the projected gradient is not as efficient as the interior-point strategy used in Algorithm 2.1. Nevertheless, it must be mentioned that KYF was introduced with the aim of solving more general problems than the ones considered in this paper. Since many nonlinear programming problems are large, both in the number of constraints as in the number of variables, future research will consider the extension of this type of algorithms in order to deal with sparsity of the Jacobian and in order to consider the possibility of using iterative linear solvers for computing the Newtonian normal-flow direction. Moreover, the difficulties in evaluating the Jacobian leads one to analyze the extension of underdetermined quasi-Newton methods [11,23] to the bound-constrained case.
Table 3
Comparison of Algorithm KYF with Algorithm 2.1

<table>
<thead>
<tr>
<th>Problem</th>
<th>Algorithm KYF</th>
<th>Algorithm 2.1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\mu$</td>
<td>Iterations</td>
</tr>
<tr>
<td>1</td>
<td>$10^{-1}$</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>$10^{-5}$</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>$\mu_s$</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>$10^{-1}$</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>$10^{-5}$</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>$\mu_s$</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>$10^{-1}$</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>$10^{-5}$</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>$\mu_s$</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>$10^{-1}$</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>$10^{-5}$</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>$\mu_s$</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>$10^{-1}$</td>
<td>MAXIT</td>
</tr>
<tr>
<td></td>
<td>$10^{-5}$</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>$10^{-7}$</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>$\mu_s$</td>
<td>20</td>
</tr>
<tr>
<td>6</td>
<td>$10^{-1}$</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>$10^{-5}$</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>$\mu_s$</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>$10^{-1}$</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>$10^{-5}$</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>$\mu_s$</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>$10^{-1}$</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>$10^{-5}$</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>$10^{-7}$</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>$\mu_s$</td>
<td>21</td>
</tr>
<tr>
<td>9</td>
<td>$10^{-1}$</td>
<td>MAXIT</td>
</tr>
<tr>
<td></td>
<td>$10^{-5}$</td>
<td>237</td>
</tr>
</tbody>
</table>
Table 3 (continued)

<table>
<thead>
<tr>
<th>Problem</th>
<th>Algorithm KYF</th>
<th>Algorithm 2.1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>μ</td>
<td>Iterations</td>
</tr>
<tr>
<td>10</td>
<td>10^{-7}</td>
<td>93</td>
</tr>
<tr>
<td></td>
<td>10^{-1}</td>
<td>310</td>
</tr>
<tr>
<td></td>
<td>10^{-5}</td>
<td>255</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>245</td>
</tr>
<tr>
<td></td>
<td>μa</td>
<td>92</td>
</tr>
<tr>
<td></td>
<td>10^{-1}</td>
<td>MAXIT</td>
</tr>
<tr>
<td>11</td>
<td>10^{-5}</td>
<td>565</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>μa</td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>10^{-1}</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>10^{-5}</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>μa</td>
<td>50</td>
</tr>
<tr>
<td>12</td>
<td>10^{-1}</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>μa</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>MAXIT</td>
<td>5.9 × 10^{3}</td>
</tr>
<tr>
<td>13</td>
<td>10^{-5}</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>μa</td>
<td>20</td>
</tr>
<tr>
<td>14</td>
<td>10^{-1}</td>
<td>MAXIT</td>
</tr>
<tr>
<td></td>
<td>10^{-5}</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>μa</td>
<td>29</td>
</tr>
<tr>
<td>15</td>
<td>10^{-1}</td>
<td>MAXIT</td>
</tr>
<tr>
<td></td>
<td>10^{-5}</td>
<td>MAXIT</td>
</tr>
<tr>
<td></td>
<td>10^{-7}</td>
<td>MAXIT</td>
</tr>
</tbody>
</table>

Acknowledgements

We are indebted to an anonymous referee whose comments helped us a lot for improving the first version of this paper.

References