Applied Mathematics and Computation 219 (2012) 3198–3208

Contents lists available at SciVerse ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

A derivative-free method for solving box-constrained underdetermined nonlinear systems of equations

N. Echebest [a,*], M.L. Schuverdt [a,b], R.P. Vignau [a]

[a] *Department of Mathematics, FCE, University of La Plata, 50 y 115, CP 172, 1900 La Plata, Argentina*
[b] *CONICET, Av. Rivadavia 1917, C1033AAJ CABA, Argentina*

ARTICLE INFO

Keywords:
Underdetermined nonlinear systems
Bound constraints
Quasi-Newton method
Derivative-free line search
Global convergence

ABSTRACT

A derivative-free iterative method for solving bound-constrained underdetermined nonlinear systems is presented. The procedure consists of a quasi-Newton method that uses the Broyden update formula and a globalized line search that combines the strategy of Grippo, Lampariello and Lucidi with the Li and Fukushima one. Global convergence results are proved and numerical experiments are presented.

1. Introduction

We consider the problem of finding a point $x^* \in \mathbb{R}^n$ such that satisfies the conditions

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

where $\Omega = \{x \in \mathbb{R}^n : l \leqslant x \leqslant u\}$, $l_i < u_i$ for $i = 1, \ldots, n$ and $F : \mathbb{R}^n \to \mathbb{R}^m$ is a continuously differentiable function. We are interested in bound constrained systems where $m \leqslant n$ with special emphasis when $m < n$. Moreover, we are interested in systems of equations for which the Jacobian matrix is not available or requires a prohibitive amount of storage. This situation is common when functional values come from some physical, chemical or econometrical experiments or measurements.

Box-constrained systems of nonlinear equations appear in mathematical programming as the feasible set of general optimization problems. There exist in the literature many traditional [1,21,24–26] and modern [2,9,10,19,20] methods that need to solve constrained optimization problems by attempting to improve feasibility at every iteration. If all the constraints are linear, maintaining feasibility is straightforward but, when nonlinear constraints are present, then more elaborate procedures are required for restoring feasibility at every iteration. 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.

The main motivation of this work is to develop a global convergent method for (1) which may be applied for solving the restoration phase of a derivative-free method for nonlinear optimization problems. We are particularly interested in the filter Inexact Restoration framework introduced in [10].

It is well known that problem (1) is equivalent, when a solution exists, to the nonlinear least squares problem of finding a global minimizer of $f(x) = \frac{1}{2}\|F(x)\|^2$ subject to $x \in \Omega$. However, this formulation does not take into account the specific nonlinear system structure of the original problem.

A large number of derivative-free methods have been developed to solve the general feasibility problem or related variants, such as nonlinear systems of equations and nonlinear least-squares problems. These algorithms are based on various

[*] Corresponding author.
E-mail addresses: opti@mate.unlp.edu.ar (N. Echebest), schuverd@mate.unlp.edu.ar (M.L. Schuverdt), vignau@mate.unlp.edu.ar (R.P. Vignau).

0096-3003/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.amc.2012.09.056

techniques, including multivariate interpolation models in combination with trust-region strategies, quasi-Newton methods, derivative-free line searches and pattern search methods, see e.g., [3,11,15–17,22,23,28] among others. Recently, a derivative-free algorithm for the unconstrained least squares minimization problem was developed in [31]. The goal of such algorithm is to take advantage of the problem structure by building individual polynomial interpolation models for each function with the least possible number of function evaluations.

We have presented in [7] two derivative-free methods for solving specially underdetermined nonlinear systems without bounds. The first method can be seen as an extension of the DF-SANE algorithm proposed for solving unconstrained square systems in [15], while the second one, called DF-QNB, is a quasi-Newton method that uses the rank-one update formula due to Broyden [5]. Global convergence for both algorithms was proved by using a derivative-free line search that combines the nonmonotone strategy of Grippo et al. [13] with the Li and Fukushima scheme [16].

In the present work we introduce a new algorithm for solving (1) based on the idea developed in the DF-QNB algorithm [7]. Given a current point x_k, in step 2 of DF-QNB, the search direction is computed considering two possibilities: in a first attempt, as the solution of the constrained linear system

$$B_k d + F(x_k) = 0 \quad \text{and} \quad \|d\| \leqslant \Delta, \tag{2}$$

if this is possible; otherwise, the direction is computed as an approximate solution to the problem

$$\min_{d\in\mathbb{R}^n:\|d\|\leqslant\Delta} \|B_k d + F(x_k)\|, \tag{3}$$

where B_k is an approximation of the Jacobian matrix at x_k.

Once the current direction d_k has been computed, the line search algorithm in DF-QNB looks for a steplength α_k such that

$$f(x_k + \alpha_k d_k) \leqslant \max_{0\leqslant j\leqslant M-1} f(x_{k-j}) + \eta_k - \gamma\alpha_k^2\|d_k\|^2 \tag{4}$$

where M is a nonnegative integer, $0 < \gamma < 1$ and $\sum_{k=0}^{\infty}\eta_k = \eta < \infty,\ \eta_k > 0$.

In this paper, when bound constraints are present, we will try to find a search direction in the same way but inside the box. More precisely, in each step we will look for d verifying (2) or (3) and $x_k + d \in \Omega$. By considering this extra condition, in order to guarantee that $x_k + \alpha_k d$ remains inside the box, the line search algorithm will look for a steplength α_k such that $0 < \alpha_k \leqslant 1$.

The new iterative algorithm generates a sequence $\{x_k\}$, for $k = 0, 1, 2, \ldots$, starting from a given initial point $x_0 \in \Omega$. In a similar manner to the general approach in [7], we will prove the global convergence results of our method under suitable conditions. The present method can be viewed as an extension of the DF-QNB method [7] for the box-constrained case.

This paper is organized as follows. In Section 2 we present the algorithm that performs the derivative-free line search and we establish there some of its properties. In Section 3 we define the quasi-Newton method using the Broyden update formula, and analyze the conditions under which it is possible to obtain global convergence. The proposed algorithm is tested and the numerical results are presented in Section 4. Finally, some conclusions are drawn in Section 5.

Notation

- $\|.\|$ denotes the Euclidean norm.
- $J(x)$ denotes the Jacobian matrix of F at x.
- $g(x) = J(x)^T F(x) = \nabla\left(\frac{1}{2}\|F(x)\|^2\right)$.
- A subsequence of $\{x_k\}_{k\in N}$ will be indicated by $\{x_k\}_{k\in K}$ where K is some infinite index set, $K \subseteq \mathbb{N},\ \mathbb{N} = \{0, 1, 2, \ldots\}$.

2. The nonmonotone line search without derivatives

In this section we shall be concerned with the nonmonotone derivative-free line search that will be used in the method defined in the following section. As we mentioned before, the strategy is based on the line search proposed in [15] for a derivative-free method for solving square systems and also used in [7] for the underdetermined case. Given the current iterate x_k and a search direction d_k, the algorithm looks for a steplength α_k such that

$$f(x_k + \alpha_k d_k) \leqslant \max_{0\leqslant j\leqslant M-1} f(x_{k-j}) + \eta_k - \gamma\alpha_k^2\|d_k\|^2, \tag{5}$$

where M is a nonnegative integer, $0 < \gamma < 1$ and $\sum_{k=0}^{\infty}\eta_k = \eta < \infty,\ \eta_k > 0$.

This procedure is a combination of the well known nonmonotone line search strategy for unconstrained optimization introduced by Grippo et al. [13]:

$$f(x_k + \alpha_k d_k) \leqslant \max_{0\leqslant j\leqslant M-1} f(x_{k-j}) + \gamma\alpha_k\nabla f(x_k)^T d_k, \tag{6}$$

with the Li-Fukushima derivative-free line search scheme [16]:

$$\|F(x_k + \alpha_k d_k)\| \leqslant (1 + \eta_k)\|F(x_k)\| - \gamma\alpha_k^2\|d_k\|^2. \tag{7}$$

The combined strategy produces a robust nonmonotone derivative-free line search that takes into account the advantages of both schemes. For completeness, we state here the implemented process.

Algorithm 1. Nonmonotone line search

```
Given $x_k,\ d_k \in \mathbb{R}^n,\ 0 < \tau_{min} < \tau_{max} < 1, 0 < \gamma < 1, M \in \mathbb{N}, \{\eta_k\}$ such that $\eta_k > 0$ and $\sum_{k=0}^{\infty}\eta_k = \eta < \infty$.
Step 1: Compute $\bar{f}_k = \max\{f(x_k), \ldots, f(x_{\max\{0,k-M+1\}})\}$
    $\alpha = 1$
Step 2:
    If $f(x_k + \alpha d_k) \leqslant \bar{f}_k + \eta_k - \gamma\alpha^2\|d_k\|^2$,
    define $\alpha_k = \alpha,\quad x_{k+1} = x_k + \alpha_k d_k$
    else
    choose $\alpha_{new} \in [\tau_{min}\alpha, \tau_{max}\alpha]$,
    $\alpha = \alpha_{new}$ and go to step 2.
```

The main difference between the previous algorithm and Algorithm 1 in [7] is that we search here for a point in the direction d_k avoiding the opposite direction $-d_k$ as is allowed in [7,15].

Proposition 1. *Algorithm 1 is well defined.*

Proof. See Proposition 1 of [15].

The new algorithm for solving (1)

Algorithm 2. General Algorithm

```
Given $x_0 \in \Omega,\ F(x_0),\ M \in \mathbb{N},\ 0 < \tau_{min} < \tau_{max} < 1,\ 0 \leqslant \epsilon < 1,\ 0 < \gamma < 1,\ \{\eta_k\}$ such that $\eta_k > 0$ and $\sum_{k=0}^{\infty}\eta_k = \eta < \infty,\ \Delta > 0$.
Set $k \leftarrow 0$.

Step 1: If $\|F(x_k)\| \leqslant \epsilon$, stop.
Step 2: Compute a search direction $d_k$ such that $x_k + d_k \in \Omega$ and $\|d_k\| \leqslant \Delta$.
Step 3: Find $\alpha_k$ and $x_{k+1} = x_k + \alpha_k d_k$ using Algorithm 1.
        Update $k \leftarrow k + 1$ and go to Step 1.
```

Observe that, since $x_k + d_k \in \Omega$ and the steplength α_k from Algorithm 1 is reduced then $x_{k+1} \in \Omega$. By considering the procedure above it is possible to get results that will be used in the next section for obtaining the convergence theorems.

The proofs of Propositions 2–5 below are quite similar to the ones of Proposition 2.2, 2.3, 2.4 and 2.5 in [7], updated for Algorithm 1.

Proposition 2. *Consider $U_k = \max\{f(x_{(k-1)M+1}), \ldots, f(x_{kM})\}$ for all $k = 1, 2, \ldots$ and define $v(k) \in \{(k-1)M + 1, \ldots, kM\}$ the index for which $f(x_{v(k)}) = U_k$. Then for all $k = 1, 2, \ldots$*

$$f(x_{v(k+1)}) \leqslant f(x_{v(k)}) + \eta$$

where $\eta = \sum_{i=0}^{\infty}\eta_i$.

Proposition 3.

$$\lim_{k\to\infty}\alpha_{v(k)-1}^2\|d_{v(k)-1}\|^2 = 0.$$

Proposition 4. *The sequence $\{x_k\}$ generated by the General Algorithm is contained in*

$$\Gamma = \{x \in \Omega : f(x) \leqslant f(x_{v(1)}) + \eta\}.$$

Proposition 5. *If we take $M = 1$ in Algorithm 1 then*

- *the sequence $\{x_k\}$ generated by the General Algorithm is contained in $\Gamma = \{x \in \Omega : f(x) \leqslant f(x_0) + \eta\}$.*
- *the series $\sum_{k=1}^{\infty}\alpha_k^2\|d_k\|^2$ is convergent.*

The results obtained up to here strongly depend on the line search technique without taking into account the way in which the direction d_k in the step 2 of Algorithm 2 was computed.

From now on we will consider the set of indices

$$K = \{\nu(1) - 1, \nu(2) - 1, \nu(3) - 1, \dots\}. \tag{8}$$

Then, from Proposition 3, we have that

$$\lim_{k \in K} \alpha_k^2 \|d_k\|^2 = 0. \tag{9}$$

3. A box-constrained derivative-free quasi Newton method

In this section we will define a quasi-Newton method based on the Broyden update formula and the derivative-free line search described in Algorithm 1 and prove the convergence results.

As we have mentioned before, this algorithm is strongly based on the DF-QNB algorithm [7]. Given a current point x_k, we compute a search direction d_k, such that $x_k + d_k \in \Omega$, satisfying the condition (2) in a first attempt. Otherwise, d_k is computed as an approximate solution of (3), subject to $x_k + d_k \in \Omega$. The new iterate x_{k+1} is computed using Algorithm 1. The formula used for updating the matrix B_k, when $s_k = x_{k+1} - x_k \neq 0$, is

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)s_k^T}{\|s_k\|^2} \tag{10}$$

where $y_k = F(x_{k+1}) - F(x_k)$, as it was considered in [7,18].

Algorithm 3. BCDF-QNB

```
  Given $x_0 \in \Omega,\ F(x_0),\ 0 < \gamma < 1,\ 0 \leqslant \theta_0 < \bar{\theta} < 1,\ 0 < \tau_{min} < \tau_{max} < 1,\ 0 \leqslant \epsilon < 1,\ \Delta > 0,\ ind = 0,\ imax \in \mathbb{N},\ imax > 0$.
  Set $k \leftarrow 0$.

Step 1: If $\|F(x_k)\| \leqslant \epsilon$, stop.
Step 2: Computing the matrix $B_k$.
        If $k = 0$ or $ind = imax$ compute $B_k$ by finite differences as an approximation to the Jacobian matrix at $x_k$. Otherwise, update $B_k$ using (10).
Step 3: Computing the direction $d_k$.
Step 3.1:   Find $d$ such that

    $$B_k d + F(x_k) = 0, \quad x_k + d \in \Omega \quad \text{and} \quad \|d\| \leqslant \Delta. \tag{11}$$

If such direction $d$ is found, define $d_k = d$, $\theta_{k+1} = \theta_k, ind = 0$ and go to Step 4.
Step 3.2:   Find an approximate solution of $\min_{x_k + d \in \Omega} \|B_k d + F(x_k)\|$.
            If $d$ satisfies

    $$\|B_k d + F(x_k)\| \leqslant \theta_k \|F(x_k)\| \quad \text{and} \quad \|d\| \leqslant \Delta \tag{12}$$

define $d_k = d,\ \theta_{k+1} = \theta_k,\ ind = 0$ and go to Step 4.
Step 3.3:   Set $d_k = 0, x_{k+1} = x_k, \theta_{k+1} = \frac{\theta_k + \bar{\theta}}{2}$.
            If $ind = imax$, define $\bar{\theta} = \frac{\bar{\theta}+1}{2}$. Set $ind = 0$ and go to Step 5.
            If $ind < imax$, set $ind = ind + 1$ and go to Step 5.
Step 4: Find $\alpha_k$ and $x_{k+1} = x_k + \alpha_k d_k$ using Algorithm 1.
Step 5: Update $k \leftarrow k + 1$ and go to Step 1.
```

Observe that, when an approximate solution of the constrained problem $\min_{x_k + d \in \Omega} \|B_k d + F(x_k)\|$ verifies (12) we accept this solution as a search direction. Otherwise, we assume that it is still possible to find a solution verifying (12) by increasing θ_k. This is the main reason for which we include Step 3.3. When we increase θ_k *imax* times without finding a solution that satisfies (12) the parameter $\bar{\theta}$ is increased. In this case we consider that the matrix B_k is not a good approximation of the true Jacobian and we compute a new matrix using finite differences.

The following theorems state the necessary hypotheses for obtaining the convergence results.

Theorem 1. *Assume that Algorithm 3 generates an infinite sequence $\{x_k\}$. Suppose that*

$$\lim_{k \to \infty} \langle (B_k - J(x_k)) d_k, F(x_k) \rangle = 0 \tag{13}$$

and that $\bar{\theta}$ is increased a finite number of times. Then, every limit point of $\{x_k\}_{k \in K}$ is a solution of $F(x) = 0$ with $x \in \Omega$, where K is given by (8).

Proof. Let x^* be a limit point of $\{x_k\}_{k\in K}$, then there exists $K_1 \subset K$ such that $\lim_{k\in K_1} x_k = x^*$.

Since $\overline{\theta}$ is increased a finite number of times then there exists $k_0 \in \mathbb{N}$ such that, for all $k \geqslant k_0,\ \overline{\theta} = \widetilde{\theta} < 1$.

We will consider two cases:

(a) We assume that in the process of the Algorithm 3, the direction d_k was obtained a finite number of times by solving the linear system finitely times solving the linear system $B_k d + F(x_k) = 0$, with $x_k + d \in \Omega$ and $\|d\| \leqslant \Delta$.

Then, there exists $k_1 \in K_1$ such that for all $k \geqslant k_1,\ k \in K_1,\ d_k$ verifies the formula $\|B_k d + F(x_k)\| \leqslant \theta_k \|F(x_k)\|,\ x_k + d \in \Omega$ and $\|d\| \leqslant \Delta$. Thus, for all $k \in K_1,\ k \geqslant \max\{k_0, k_1\}$ we have that

$$\langle B_k d_k, F(x_k)\rangle \leqslant \frac{\overline{\theta}^2 - 1}{2} \|F(x_k)\|^2.$$

This implies that

$$\langle (B_k - J(x_k))d_k, F(x_k)\rangle + \langle J(x_k)d_k, F(x_k)\rangle \leqslant \frac{\overline{\theta}^2 - 1}{2} \|F(x_k)\|^2. \tag{14}$$

We know, from Proposition 3, that $\lim_{k\in K} \alpha_k^2 \|d_k\|^2 = 0$, so $\lim_{k\in K_1} \alpha_k^2 \|d_k\|^2 = 0$.

If $\lim_{k\in K_1} \alpha_k \neq 0$ then

$$\lim_{k\in K_1} \|d_k\| = 0. \tag{15}$$

Taking limits for $k \in K_1,\ k \geqslant \max\{k_0, k_1\}$ in (14), and using (13), (15) and the continuity of F and J, we obtain that

$$0 \leqslant \frac{\overline{\theta}^2 - 1}{2} \|F(x^*)\|^2.$$

Since $\overline{\theta} < 1$, we have that $\|F(x^*)\| = 0$ as we wanted to prove.

If $\lim_{k\in K_1} \alpha_k = 0$, since the sequence $\{d_k\}_{k\in K_1}$ is bounded, there exists $K_2 \subset K_1$ and $\bar{d} \in \mathbb{R}^n$ such that $\lim_{k\in K_2} d_k = \bar{d}$.

In the line search, to compute the step α_k, Algorithm 3 tests the following inequality

$$f(x_k + \alpha d_k) \leqslant \overline{f}_k + \eta_k - \gamma \alpha^2 \|d_k\|^2. \tag{16}$$

The initial value of α is 1. Since $\lim_{k\in K_2} \alpha_k = 0$, there exists $\bar{k} \in K_2$ such that $\alpha_k < 1$ for all $k \in K_2,\ k \geqslant \bar{k}$. Thus, for those iterations k there exist steps $\overline{\alpha}_k$ that do not satisfy (16) and $\lim_{k\in K_2} \overline{\alpha}_k = 0$. So we have that

$$f(x_k + \overline{\alpha}_k d_k) > \overline{f}_k + \eta_k - \gamma(\overline{\alpha}_k)^2 \|d_k\|^2 > \overline{f}_k - \gamma(\overline{\alpha}_k)^2 \|d_k\|^2.$$

Since $\|d_k\| \leqslant \Delta$ we obtain that

$$\frac{f(x_k + \overline{\alpha}_k d_k) - f(x_k)}{\overline{\alpha}_k} > -\gamma \overline{\alpha}_k \Delta^2.$$

By the Mean Value Theorem there exists $\xi_k \in [0, 1]$ such that

$$\langle g(x_k + \xi_k \overline{\alpha}_k d_k), d_k\rangle > -\gamma \bar{\alpha}_k \Delta^2. \tag{17}$$

Taking limits in (17) when $k \to \infty,\ k \in K_2$, we obtain that

$$\langle g(x^*), \bar{d}\ \rangle = \langle J(x^*)\bar{d}, F(x^*)\ \rangle \geqslant 0.$$

And taking limits in (14) when $k \to \infty,\ k \in K_2$, we obtain that

$$0 \leqslant \langle J(x^*)\bar{d}, F(x^*)\rangle \leqslant \frac{\overline{\theta}^2 - 1}{2} \|F(x^*)\|^2 \leqslant 0. \tag{18}$$

Thus, by (18), $\|F(x^*)\| = 0$ as we wanted to prove.

(b) We assume that in the process of the Algorithm 3, the direction d_k was obtained infinitely many times by solving the linear system $B_k d + F(x_k) = 0$, with $x_k + d \in \Omega$ and $\|d\| \leqslant \Delta$.

Then, there exists $K_2 \subset K_1$ such that for all $k \in K_2$ we have that $B_k d_k + F(x_k) = 0$. Thus, for all $k \in K_2$,

$$\langle (B_k - J(x_k))d_k, F(x_k)\rangle + \langle J(x_k)d_k, F(x_k)\rangle = -\|F(x_k)\|^2. \tag{19}$$

We know that $\lim_{k\in K} \alpha_k^2 \|d_k\|^2 = 0$, so $\lim_{k\in K_2} \alpha_k^2 \|d_k\|^2 = 0$.

If $\lim_{k\in K_2} \alpha_k \neq 0$ then

$$\lim_{k\in K_2} \|d_k\| = 0. \tag{20}$$

Taking limits in (19) when $k \to \infty,\ k \in K_2$, and using (13), (20) and the continuity of F and J, we obtain that $\|F(x^*)\| = 0$ as we wanted to prove.

If $\lim_{k\in K_2} \alpha_k = 0$, since the sequence $\{d_k\}_{k\in K_2}$ is bounded, there exists $K_3 \subset K_2$ and $\bar{d} \in \mathbb{R}^n$ such that $\lim_{k\in K_3} d_k = \bar{d}$.

As we did in case (a) we obtain that $\langle J(x^*)\bar{d}, F(x^*)\rangle \geqslant 0$, and taking limits in (19) and using (13) we obtain that $\langle J(x^*)\bar{d}, F(x^*)\rangle = -\|F(x^*)\|^2 \leqslant 0$. So $\|F(x^*)\| = 0$ and the proof is complete. □

Observe that, according to Proposition 4, we have that $\|F(x_k)\|$ is bounded. Thus, if

$$\lim_{k\to\infty}\frac{\|(B_k - J(x_k))d_k\|}{\|d_k\|} = 0$$

we obtain the hypothesis (13). The last condition is known as a necessary and sufficient condition for obtaining q-superlinear convergence of classical quasi-Newton methods [5].

If we cannot guarantee that $\bar{\theta}$ is increased a finite number of times then we can prove the following result which is similar to one obtained in [7,18].

Theorem 2. *Suppose that, in Algorithm 3, $\bar{\theta}$ is increased infinitely many times and define K_* the set of indices for which that happens.*

Assume that

$$\lim_{k\in K_*}\|B_k - J(x_k)\| = 0. \tag{21}$$

Then, every limit point x^* of the sequence $\{x_k\}_{k\in K_*}$ is a solution of (1) or it is a global minimizer of $\|F(x^*) + J(x^*)(x - x^*)\|$ subject to $x \in \Omega$.

Proof. See Theorem 3.2 of [18].

Remark 1. If $B_k,\ k \in K_*$, is computed by finite differences using a step h that satisfies $h \leqslant \min\{\epsilon, 1 - \bar{\theta}\}$ then the hypothesis (21) can be omitted.

A point $x^* \in \Omega$ that is a global minimizer of the function $\|F(x^*) + J(x^*)(x - x^*)\|$ with $x \in \Omega$ can be viewed as the solution of the linear least squares problem

$$\min_{x\in\Omega}\|A(x - x^*) - b\| \tag{22}$$

where $A = J(x^*)$ and $b = -F(x^*)$. The linear function is the affine model of the function F around x^*. We cannot expect, in general, to find $x^* \in \Omega$ such that $F(x^*) = 0$ since the problem could not have a solution. Likewise, we cannot expect to find $x^* \in \Omega$ such that $F(x^*) + J(x^*)(x - x^*) = 0$ since this is an underdetermined linear system of equations and $J(x^*)$ could not have full rank. Because of that, it seems reasonable to find a global minimizer of (22) when the problem has no solutions.

The case when $M = 1$ in the derivative-free line search used in Algorithm 3 deserves a separate comment. As it was noticed in [7] for the DF-QNB algorithm, the presence of η_k imposes an almost monotone behavior of the merit function when x_k is close to a solution.

As it was done in [7], under this condition it is possible to demonstrate that our algorithm verifies the assumption (13). Thus this particular line search improves the results of Theorem 1. For the sake of completeness we enunciate here those results.

Lemma 1 (*Lemma 2.6, [16]*). *Let us suppose that the set $\Gamma = \{x \in \mathbb{R}^n : f(x) \leqslant f(x_0) + \eta\}$ is bounded and that $J(x)$ is Lipschitz continuous in Γ.*

If

$$\sum_{k=1}^{\infty}\alpha_k^2\|d_k\|^2 < \infty, \quad \text{then} \quad \lim_{k\to\infty}\frac{1}{k}\sum_{i=0}^{k-1}\rho_i^2 = 0$$

where

$$\rho_k = \frac{\|(A_{k+1} - B_k)d_k\|}{\|d_k\|} \quad \text{and} \quad A_{k+1} = \int_0^1 J(x_k + t\alpha_k d_k)dt.$$

In particular, there is a subsequence of $\{\rho_k\}$ tending to zero.

Theorem 3. *Assume that Algorithm 3 generates an infinite sequence $\{x_k\}$, that $M = 1$ in the line search and that the hypotheses of Lemma 1 hold. If $\bar{\theta}$ is increased a finite number of times, there is a limit point x^* of $\{x_k\}_{k\in\mathbb{N}}$ that is a solution of (1).*

4. Numerical experiments

In this section we present some computational results obtained with a Fortran 77 implementation of the BCDF-QNB algorithm. All experiments were run on a personal computer with INTEL (R) Core (TM) 2 Duo CPU E8400 at 3.00 GHz and 3.23 GB of RAM.

We compare the practical performance of our algorithm with the following two codes: BOBYQA [23] and Algorithm 1 in [8], called FKM from now on. BOBYQA is a derivative-free iterative algorithm for finding a minimum of a function subject to bounds on the variables. In these experiments BOBYQA solves the least squares problem: $\min \|F(x)\|^2$, subject to $x \in \Omega$. FKM is an interior point method for solving box-constrained underdetermined nonlinear systems that uses the natural merit function associated $f(x) = \frac{1}{2}\|F(x)\|^2$. The implemented version in [8] allows the use of finite differences to approximate the Jacobian in all the iterations and this particular version, that was kindly supplied by the authors, is used here.

We realize that both approaches belong to different classes of derivative-free algorithms. BOBYQA uses a general function without taking into account the advantage of the structure of the function appearing in a least squares problem and uses polynomial interpolations to build approximate models of the objective function. The comparison with BOBYQA is not quite fair since it does not make use of the structure of the least squares function. Because of that we have considered FKM, which takes into account the least squares function but it approximates the derivatives by finite differences in all iterations. We strongly take into account that the same idea was previously considered in [31] in order to compare the performance of a derivative-free algorithm for least squares minimization without constraints.

As it is usual in derivative-free optimization articles we are interested in the number of function evaluations needed for satisfying the stopping criteria.

4.1. Test problems

We have used a set of problems defined by feasible sets of nonlinear programming problems given in Hock and Schittkowski [14] and others in Ryoo and Sahinidis [27]. We have also used test problems described in [4].

Some of the test problems analyzed here have been previously examined in [8] for solving the same problem but using derivatives.

In Table 1 we show the data of the problems. In column 1 we show the number of the problem, in column 2 the source of the problem and in the last columns the number of variables (n) and equations (m).

In the Problems 2, 5, 6, 9 and 10 we used the original initial points and bounds. Problems 1, 4, 7 and 8 do not have bound constraints. In those problems we have used the artificial bounds introduced in [8] keeping the original initial points.

In Problem 3 we have changed the initial point because FKM requires an interior point. In this case we took $x_0 = (0.5, 2, 0.5, 0.5, 0.5, 2)^T$. In Problems 11–16 we have used the original bounds and the initial points: $x_0 = (1.5, 1.5, 1, 1.5, 1, 90, 90, 3.5, 3, 150)^T, x_0 = \frac{1}{2}(u + l),\ x_0 = (10, 40, -10, 200)^T,\ x_0 = (1000, 1000, 1000, 120, 120)^T,\ x_0 = \frac{1}{2}(u + l)$ and $x_0 = (5, 5, 200)^T$, respectively.

We have considered two choices (case (a) and case (b)) of the initial points and bounds in Problems 17 and 18. The case (a) corresponds to the use of the initial point $x_0(1:n) = 2$ with $u_i = 100, l_i = -100,\ i = 1, \ldots, n$. Case (b) corresponds to the use of $x_0(1:n) = 150$ as initial point with $u_i = 300,\ l_i = -300,\ i = 1, \ldots, n$.

Table 1
Data of the problems.

Problem	Source	n	m
1	Problem 46 of [14]	5	2
2	Problem 53 of [14]	5	3
3	Problem 55 of [14]	6	6
4	Problem 56 of [14]	7	4
5	Problem 60 of [14]	3	1
6	Problem 63 of [14]	3	2
7	Problem 77 of [14]	5	2
8	Problem 79 of [14]	5	3
9	Problem 81 of [14]	5	3
10	Problem 111 of [14]	10	3
11	Problem 1 of [27]	10	7
12	Problem 3 of [27]	12	9
13	Problem 4 of [27]	4	2
14	Problem 5 of [27]	5	3
15	Problem 10 of [27]	2	1
16	Problem 11 of [27]	3	2
17	Problem 2 of [4]	300	150
18	Problem 4 of [4]	300	150

4.2. Implementation details

Details on the implementation of BCDF-QNB:

1. The parameters for Algorithm 1 were:

$$M = 2, \quad \tau_{min} = 0.1, \quad \tau_{max} = 0.5, \quad \gamma = 10^{-4}, \quad \eta_0 = 1,$$

- $\forall k \in \mathbb{N},\ k \geqslant 1,\ \eta_k = \frac{\|F(x_0)\|}{(1+k)^2}$, when $n = 300$.
- $\forall k \in \mathbb{N},\ k \geqslant 1\ \eta_k = \frac{\|F(x_0)\|}{2^k}$ otherwise.

2. The initial parameters were:

$$\epsilon = 10^{-6}, \quad \theta_0 = 0.5, \quad \bar{\theta} = 0.999, \quad \Delta = 10^{12}, \quad imax = 10.$$

3. The first matrix B_0 was computed by finite differences as an approximation to the Jacobian matrix at x_0.
4. We have used the DACCIM algorithm described in [6] to find a solution of the linear system $B_k d = -F(x_k)$, such that $x_k + d \in \Omega$. For solving the least squares problem $\min_{x_k+d\in\Omega}\|B_k d + F(x_k)\|^2$ we have used the BVLS algorithm described in [29].
5. The stopping condition for our algorithm was $\|F(x_k)\| \leqslant \epsilon$.
6. The maximum number of function evaluations allowed was 10000 (considering the evaluations used to calculate the initial matrix B_0 and the matrices B_k computed by finite differences when the procedure of BCDF-QNB requires it).

The implementation of BOBYQA, in Fortran 77, is the original version of M.J.D. Powell [23] with its stopping criterion, that is, the algorithm stops when the trust-region radius is lower than a tolerance $\rho_{end} = 10^{-6}$. Also BOBYQA stops when it reaches the allowed number of function evaluations or when the value $\|F(x_k)\|$ is less than the one achieved by BCDF-QNB.

The implementation of FKM in Matlab is the original version of Francisco, Krejić and Martínez [8]. As in BCDF-QNB the maximum number of function evaluations was 10000 and the algorithm stops when $\|F(x_k)\| \leqslant 10^{-6}$. In this implementation we considered the evaluations used to approximate the Jacobian matrix in all iterations.

4.3. Numerical results

In Table 2 we show the results obtained taking into account the number of function evaluations and the final value $\|F(x_{end})\|$ for the codes: BCDF-QNB (BCDF), BOBYQA (BQA) and FKM. The results correspond to the stopping criterion satisfaction or to internal conditions that do not allow further improvement. It also shows the number of the problem (column 1), the number of function evaluations (Feval, column 2), and the final functional values obtained for each code ($\|F(x_{end})\|$, column 3).

Table 2
Results of test problems.

Prob.	F eval.			$\|F(x_{end})\|$		
	BCDF	BQA	FKM	BCDF	BQA	FKM
1	28	111	49	1.841433D-07	3.418782D-08	1.297465D-07
2	7	59	13	1.026235D-09	1.250089D-10	8.441465D-09
3	8	94	15	2.101248D-09	1.399019D-09	1.989612D-08
4	13	76	25	3.366884D-07	5.284583D-09	5.070781D-07
5	12	66	61	3.043343D-10	2.722729D-15	9.073879D-07
6	14	55	21	6.312838D-09	2.607497D-10	1.140705D-10
7	16	119	37	2.765712D-07	1.147991D-08	1.254718D-11
8	14	89	25	6.017154D-08	4.784848D-08	4.388764D-09
9	13	63	25	9.827276D-07	2.075567D-07	5.815963D-13
10	26	250	57	4.219129D-09	3.109068D-09	1.724752D-10
11	20	57	282	2.468423D-07	1.515770D-07	3.638097D-12
12	30	115	274	4.372665D-08	1.434150D-09	2.239968D-08
13	11	141	21	5.922943D-07	2.128820D-07	1.931713D-07
14	30	360	55	2.363432D-08	1.157911D-03	6.177695D-09
15	9	30	13	1.610945D-08	2.088413D-09	7.235173D-07
16	7	48	13	6.664413D-08	1.918797D-08	5.982311D-07
17(a)	303	696	2409	4.544648D-11	6.324139D-11	3.140185D-16
18(a)	337	10000	2108	4.107578D-07	5.715843D-03	1.707833D-08
17(b)	303	794	2409	2.166159D-14	9.206523D-11	3.080427D-13
18(b)	853	10000	5118	9.871958D-07	*	5.625058D-12

* It does not reach an approximate solution of the problem.

These results illustrate BCDF-QNB effectiveness. We have to mention that, for this test problems, BCDF-QNB has used finite differences only for computing the initial matrix B_0. That is the main reason for which BCDF-QNB always has required less function evaluations than FKM. We can notice that our algorithm always has done less function evaluations than BOBYQA in all the problems meanwhile FKM has done it in 16 problems. We believe that this behavior is due to the fact that BOBYQA does not take advantage of the structure of the problem.

On the other hand, when we consider $\|F(x_{end})\|$ as a measure of the performance of the algorithms we can see that FKM outperforms BCDF-QNB in 13 of the 20 problems. We believe that these results are associated to an accurate approximation of the Jacobian matrix.

4.4. Illustrative example of a restoration phase of a filter inexact restoration method

The objective of this subsection is to illustrate the main motivation of the present work. We will show how BCDF-QNB can be used as a subalgorithm for the restoration phase of a derivative-free method based on filters.

We consider a general optimization problem with equality and inequality constraints:

$$\text{Minimize} f_o(x) \text{ subject to } c(x) = 0, \quad \bar{c}(x) \leqslant 0, \; x \in \Omega, \tag{23}$$

where $f_o : \mathbb{R}^n \to \mathbb{R}, \; c : \mathbb{R}^n \to \mathbb{R}^m, \; \bar{c} : \mathbb{R}^n \to \mathbb{R}^p$ are continuously differentiable functions on $\mathbb{R}^n$.

Each iteration of an inexact restoration method as those defined in [10,19,20] is composed of two phases: the first one is the restoration phase, which reduces a measure of infeasibility; and the second one, the optimality phase, which reduces the objective function in a tangential approximation of the feasible set. These two phases are totally independent, and they can be combined using merit functions or filters to measure the progress of the algorithm. As we are interested on the method defined in [10] we will describe it briefly in the following lines.

We define the auxiliary function

$$f_i^+(x) = \begin{cases} c_i(x) & i = 1, \ldots, m \\ max\{0, \bar{c}_i(x)\} & i = m+1, \ldots, m+p \end{cases}$$

and we consider the following measure of the infeasibility in a point $x : h(x) = \|f^+(x)\|$. Filter algorithms define a forbidden region by memorizing the pairs $(f_o(x_k), h(x_k))$ from former iterations, and then avoiding points dominated by those by the usual Pareto domination rule: x dominates y if and only if $f_o(x) \leqslant f_o(y)$ and $h(x) \leqslant h(y)$.

The general scheme of the filter algorithm in [10] is the following.

```
Filter Algorithm.
Data: $x_0 \in \mathbb{R}^n, \; F_0 = \emptyset, \; \mathcal{F}_0 = \emptyset, \; \alpha \in (0,1)$.
$k = 0$.
Repeat
  $(\tilde{f}_o, \tilde{h}) = (f_o(x_k) - \alpha h(x_k), (1-\alpha) h(x_k))$.
  Construct the set $\overline{F}_k = F_k \cup \{(\tilde{f}_o, \tilde{h})\}$
  Define the set $\bar{\mathcal{F}}_k = \mathcal{F}_k \cup \{x \in \mathbb{R}^n : f_o(x) \geqslant \tilde{f}_o, h(x) \geqslant \tilde{h}\}$
  Restoration phase
    If $h(x_k) = 0$ then $z_k = x_k$,
    else, compute $z_k \notin \bar{\mathcal{F}}_k$ such that $h(z_k) < (1-\alpha) h(x_k)$,
      if impossible then stop with unsuccess.
  Optimization phase
    If $z_k$ is a stationary point then stop with success.
    Else, compute $x_{k+1} \notin \bar{\mathcal{F}}_k$ such that $x_{k+1}$ is in a linearization of the feasible set in $z_k$ and $f_o(x_{k+1}) \leqslant f_o(z_k)$.
  Filter update
    If $f_o(x_{k+1}) \leqslant f_o(x_k)$ then
      $F_{k+1} = F_k, \mathcal{F}_{k+1} = \mathcal{F}_k$ ($f_o$- iteration)
    else
      $F_{k+1} = \overline{F}_k, \mathcal{F}_{k+1} = \bar{\mathcal{F}}_k$ ($h$- iteration)
```

Let us focus now on the Restoration phase. Since the merit function used in BCDF-QNB is the function h of the filter method described previously, we can use it iteratively to solve this phase. Thus, given an iterate x_k we can apply BCDF-QNB until a new point $z_k \notin \bar{\mathcal{F}}_k$ satisfying the descent condition $h(z_k) < (1-\alpha) h(x_k)$, for a fixed α, is found.

It is important to notice that, once z_k was computed in the restoration phase, we could consider a new Broyden matrix $\overline{B}$ by updating the last one computed in the process and use it as an approximation of the Jacobian matrix at z_k. It is known this new matrix behaves as the true Jacobian on the step just taken, but its behavior on other directions is uncertain [5].

We realize that the inexactness of the approximate Jacobian affects the computation of the linearized tangent to the constraints. Nevertheless, in this first attempt we will use it in order to compute a point in the optimization phase. A similar idea

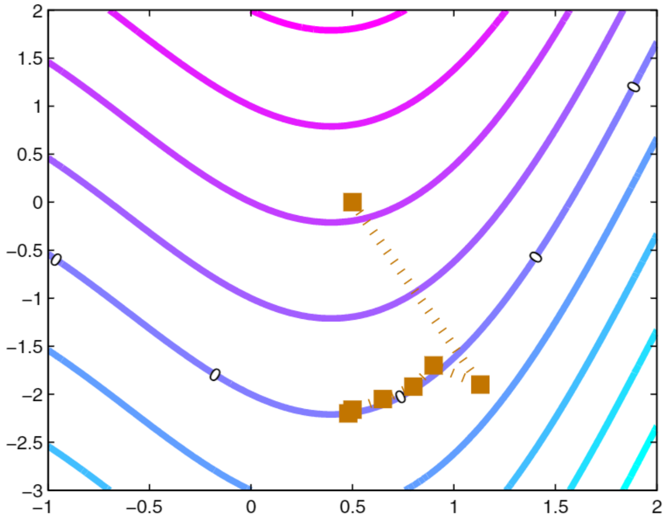

Fig. 1. Iterates x_k.

Table 3
$(f_o(x_k), h(x_k))$.

k-iter	$f_o(x_k)$	$h(x_k)$
0	0	2.1900
1	−2.0350	0.4100
2	−1.8000	0.2000
3	−2.0400	0.1000
4	−2.1700	0.0300
5	−2.1900	0.0200
6	−2.2000	0.0003

was used in [30] by considering the TR1 update [12] to generate an approximation of the Jacobian in the context of a trust region algorithm.

The algorithm is applied to the resolution of a small problem appearing in [10] with the intention to use BCDF-QNB in the restoration phase, defining the linearization of the feasible set by using the Broyden update formula as mentioned before. The example has only one nonlinear equality constraint and the objective function is affine which does not require the calculation of the derivative. We emphasize that some points still open will be studied in the future: the optimization phase of a derivative-free restoration method, the optimality criteria and the efficient use of the filter. In the present example, in the implementation of the optimization phase we use the projected gradient of the objective function over the approximate tangent defined by $\overline{B}(x - z_k) = 0$.

The bidimensional problem considered is:

$$\text{Minimize } x(2) \text{ subject to } x(2) + (2 + x(1))\cos(x(1)) = 0.$$

Table 3 shows the values $(f_o(x_k), h(x_k))$, starting from $x_0 = (0.5, 0.0)$, obtained doing 6 iterations and Fig. 1 shows the obtained points x_k. (See Table 3)

Although the problem is small we can observe a satisfactory performance since the sequence tends to the optimum $x^* = (0.39, -2.21)$.

5. Conclusions

Many practical optimization methods require specific algorithms for improving feasibility during the procedure. Thus, our aim in this paper was to define an algorithm capable of dealing with the feasible set defined by underdetermined nonlinear system of equations with bound constraints. We present a derivative-free algorithm that exploits this structure based on the Broyden method. This algorithm can be viewed as generalization of the one defined in [7].

From a theoretical point of view we were able to obtain some convergence results. Under usual assumptions on the Jacobian matrix and a Dennis Moré type condition we established global convergence. We have shown that this condition can be dropped out for a particular line search.

Numerical experiments suggest that our algorithm behaves as expected, it outperforms others from the point of view of the required number of function evaluations. These results are reasonable, and they show the importance to develop algorithms that take full advantage of the problem structure. However, it will be desirable to test the performance of our algorithm with a more challenging set of problems.

We also illustrate how our algorithm can be used as a subalgorithm for the restoration phase of a derivative-free method based on filters. In this context in a future work we will study the open problem presented in Section 4.4.

Acknowledgements

We wish to thank Professor Dr. Hugo Scolnik, University of Buenos Aires, for his valuable suggestions.

References

[1] J. Abadie, J. Carpentier, Generalization of the wolfe reduced-gradient method to the case of nonlinear constraints, in: R. Fletcher (Ed.), Optimization, Academic Press, New York, 1968, pp. 37–47.
[2] R.H. Bielschowsky, F.A.M. Gomes, Dynamic control of infeasibility in equality constrained optimization, SIAM Journal on Optimization 19 (3) (2008) 1299–1325.
[3] A.R. Conn, K. Scheinberg, L.N. Vicente, Global convergence of general derivative-free trust-region algorithms to first and second-order critical points, SIAM Journal on Optimization 20 (2009) 387–415.
[4] N. Dan, N. Yamashita, M. Fukushima, Convergence properties of inexact Levenberg–Marquardt method under local error bound conditions, Optimization Methods and Software 17 (2002) 605–626.
[5] J.E. Dennis, R.B. Schnabel, Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Prentice-Hall, Englewood Cliffs, 1983.
[6] N. Echebest, M.T. Guardarucci, H.D. Scolnik, M.C. Vacchino, An acceleration scheme for solving convex feasibility problems using incomplete projection algorithms, Numerical Algorithms 35 (2–4) (2004) 335–350.
[7] N. Echebest, M.L. Schuverdt, R.P. Vignau, Two derivative-free methods for solving underdetermined nonlinear systems of equations, Computational and Applied Mathematics 30 (2011) 217–245.
[8] J.B. Francisco, N. Krejić, J.M. Martínez, An interior point method for solving box-constrained underdetermined nonlinear systems, Computational and Applied Mathematics 177 (2005) 67–88.
[9] N.I.M. Gould, Ph.L. Toint, Nonlinear programming without a penalty function or a filter, Mathematical Programming 122 (1) (2010) 155–196.
[10] C.C. Gonzaga, E.W. Karas, M. Vanti, A globally convergent filter method for nonlinear programming, SIAM Journal on Optimization 14 (3) (2003) 646–669.
[11] A. Griewank, The global convergence of Broyden-like methods with a suitable line search, Journal of the Australian Mathematical Society, Series B 28 (1) (1986) 75–92.
[12] A. Griewank, A. Walther, On constrained optimization by adjoint-based quasi-Newton methods, Optimization Methods and Software 17 (2002) 869–889.
[13] L. Grippo, F. Lampariello, S. Lucidi, A nonmonotone line search technique for Newton's method, SIAM Journal on Numerical Analysis 23 (1986) 707–716.
[14] W. Hock, K. Schittkowski, Test Examples for Nonlinear Programming Codes, Lectures Notes in Economics Mathematical Systems, Springer, 1981.
[15] W. La Cruz, J.M. Martínez, M. Raydan, Spectral residual method without gradient information for solving large-scale nonlinear systems of equations, Mathematics of Computation 75 (2006) 1429–1448.
[16] D.H. Li, M. Fukushima, A derivative-free line search and global convergence of Broyden-like method for nonlinear equations, Optimization Methods and Software 13 (2000) 181–201.
[17] C. Ma, L. Chen, D. Wang, A globally and superlinearly convergent smoothing Broyden-like method for solving nonlinear complementarity problem, Applied Mathematics and Computation 198 (2) (2008) 592–604.
[18] J.M. Martínez, Quasi-inexact Newton methods with global convergence for solving constrained nonlinear systems, Nonlinear Analysis 30 (1997) 1–7.
[19] J.M. Martínez, Inexact restoration method with Lagrangian tangent decrease and new merit function for nonlinear programming, Journal of Optimization Theory and Applications 111 (2001) 39–58.
[20] J.M. Martínez, E.A. Pilotta, Inexact restoration algorithms for constrained optimization, Journal of Optimization Theory and Applications 104 (2000) 135–163.
[21] A. Miele, E.M. Sims, V.K. Basapur, Sequential gradient-restoration algorithm for mathematical programming problems with inequality constraints, Part 1, Theory, Rice University, Aero-Astronautics, Report No. 168, 1983.
[22] P.Y. Nie, A derivative-free method for the system of nonlinear equations, Nonlinear Analysis: Real World Applications 7 (2006) 378–384.
[23] M.J.D. Powell, The BOBYQA algorithm for bound constrained optimization without derivatives, Technical Report, Department of Applied Mathematics and Theoretical Physics, Cambridge University, Cambridge, England, 2009.
[24] M. Rom, M. Avriel, Properties of the sequential gradient-restoration algorithm (SGRA), part 1: introduction and comparison with related methods, Journal of Optimization Theory and Applications 62 (1989) 77–98.
[25] M. Rom, M. Avriel, Properties of the sequential gradient-restoration algorithm (SGRA), part 2: convergence analysis, Journal of Optimization Theory and Applications 62 (1989) 99–126.
[26] J.B. Rosen, Two-Phase Algorithm for Nonlinear Constraint Problems, in: O.L. Mangasarian, R.R. Meyer, S.M. Robinson (Eds.), Nonlinear Programming 3, Academic Press, London and New York, 1978, pp. 97–124.
[27] N.V. Ryoo, N.D. Sahinidis, Global optimization of nonconvex NLPs and MINLPs with applications in process design, Computers & Chemical Engineering 19 (1995) 551–566.
[28] J.R. Sharma, R.K. Guha, P. Gupta, Some efficient derivative free methods with memory for solving nonlinear equations, Applied Mathematics and Computation 219 (2) (2012) 699–707.
[29] P.B. Stark, R.L. Parker, Bounded variable least squares: an algorithm and applications, Journal Computational Statistics 10 (1995) 129–141.
[30] A. Walther, L.T. Biegler, Numerical experiments with an inexact Jacobian trust-region algorithm, Computational Optimization and Applications 48 (2) (2011) 255–271.
[31] H. Zhang, A.R. Conn, K. Scheinberg, A derivative-free algorithm for least-squares minimization, SIAM Journal on Optimization 20 (6) (2010) 3555–3576.