Comput Optim Appl (2012) 53:771–794
DOI 10.1007/s10589-012-9469-8

# Constrained Dogleg methods for nonlinear systems with simple bounds

**Stefania Bellavia · Maria Macconi · Sandra Pieraccini**

Published online: 7 March 2012


**Abstract** We focus on the numerical solution of medium scale bound-constrained systems of nonlinear equations. In this context, we consider an affine-scaling trust region approach that allows a great flexibility in choosing the scaling matrix used to handle the bounds. The method is based on a dogleg procedure tailored for constrained problems and so, it is named Constrained Dogleg method. It generates only strictly feasible iterates. Global and locally fast convergence is ensured under standard assumptions. The method has been implemented in the `Matlab` solver `CoDoSol` that supports several diagonal scalings in both spherical and elliptical trust region frameworks. We give a brief account of `CoDoSol` and report on the computational experience performed on a number of representative test problems.

**Keywords** Bound-constrained equations · Diagonal scalings · Trust region methods · Dogleg methods · Newton methods · Global convergence

Work supported by the MIUR, Rome, Italy, through PRIN grants 2007 and 200834WK7H_004.

S. Bellavia (✉) · M. Macconi
Dipartimento di Energetica 'S. Stecco', Università di Firenze, Viale Morgagni 40/44, 50134 Firenze, Italy
e-mail: stefania.bellavia@unifi.it

M. Macconi
e-mail: maria.macconi@unifi.it

S. Pieraccini
Dipartimento di Scienze Matematiche, Politecnico di Torino, Corso Duca degli Abruzzi, 24, 10129 Torino, Italy
e-mail: pieraccini@calvino.polito.it

## 1 Introduction

The problem of interest is to find a vector $x \in \mathbb{R}^n$ satisfying

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

where $F : X \mapsto \mathbb{R}^n$ is a continuously differentiable mapping with Jacobian denoted by $F'$, $X \subseteq \mathbb{R}^n$ is an open set containing the $n$-dimensional box $\Omega = \{x \in \mathbb{R}^n \mid l \le x \le u\}$. Here, the inequalities are meant component-wise and the vectors $l \in (\mathbb{R} \cup -\infty)^n$, $u \in (\mathbb{R} \cup +\infty)^n$ are specified lower and upper bounds on the variables such that $\Omega$ has nonempty interior.

Newton method augmented with affine scaling trust region procedures forms a class of efficient methods for the solution of this problem. Such methods are well known to show good local convergence behavior. Further, they find a solution of (1) starting from arbitrary initial guesses or fail in one of a small number of easily detectable ways, i.e. they are globally convergent methods.

Originally proposed in the context of constrained optimization [6], the affine scaling trust region approach was then developed to form a robust theoretical and practical framework containing a number of globally convergent methods for smooth and nonsmooth bound-constrained systems of nonlinear equations [1, 2, 4, 16, 31]. In particular, the methods given in [1, 2, 4] are based on ellipsoidal trust regions defined by the diagonal scaling matrix proposed in [6]. The same scaling is used for the solution of large scale problems [3, 5] as well as for developing extensions to rectangular nonlinear systems [11, 21, 22].

We remark that the spirit of these methods is to use diagonal scalings to handle the bounds. At each iteration a quadratic model of the merit function $\frac{1}{2}\|F\|^2$ is minimized within a trust region around the current iterate and suitable stepsizes rules yield a new feasible trial point. Then, iterates within the feasible region $\Omega$ are generated in such a way that global and locally fast convergence is ensured.

A crucial point is that the classical dogleg procedure can not be used to approximately minimize the model within the trust region and handle the bounds at the same time. In fact, many theoretical properties of the dogleg curve are lost in the constrained context. More flexibility is needed in the choice of the dogleg path and suitable modifications of the classical dogleg method are required to ensure the strict feasibility of the iterates. The rules adopted to generate only feasible approximations of the solution play an important role in motivating different affine scaling algorithms for (1). On this subject, we remark that a sort of double dogleg or the switching to the scaled gradient has been employed in the previous approaches.

The present paper aims at analyzing an interior point trust region method for medium scale problems alternative to those adopted in the above works. Our target is to obtain more efficient algorithms and to allow a great flexibility in choosing the diagonal scaling matrices used to handle the bounds. Nevertheless, desirable features are maintained. In particular, all the iterates are required to be strictly feasible points and global, locally fast convergence must be ensured.

In the technical report [3], the authors introduced an inexact dogleg procedure for solving large scale problems. This procedure extends to constrained systems the method proposed in [27] for large scale unconstrained systems of nonlinear equations. In [3] the authors show how the presence of constraints is reflected in the scheme given in [27] and study global and fast convergence of an inexact dogleg method tailored for large scale bound constrained problem. The obtained procedure employs an Inexact Newton step and it is based on the minimization of the linear model of $F$ along a path founded on a scaled Cauchy step and an interior point inexact-Newton step. The convergence analysis of the method has been carried out without specifying the scaling matrix used to handle the bounds, allowing a rather general class of diagonal scalings. However, the choice of an appropriate scaling matrix has not been investigated in [3] and only preliminary results, using the pioneer Coleman-Li scaling matrix, on large scale problems have been given.

Here we focus on medium scale problems and adopt a procedure that is a special case of that given in [3] as the inexact Newton step is replaced by the exact solution of the Newton equation. In fact, when medium scale bound-constrained systems has to be solved it is realistic to assume that a direct method is used to solve the Newton equation. Then, the linear model of $F$ is minimized along a path founded on a scaled Cauchy step and an interior point Newton step, i.e. a projection of the Newton step within the feasible region $\Omega$. Therefore, even in this case the resulting dogleg curve is not the classical dogleg curve as it is defined in a constrained setting and its basement is a non-exact (projected) Newton step.

The convergence properties of the method can be stated by making easier the theoretical results given in [3]. In other words, we can appeal to the convergence analysis performed in [3] to claim that our method shows global and locally fast convergence under standard assumptions. We consider several diagonal scaling matrices proposed by different authors in the numerical optimization literature. We show that a number of choices satisfy the assumptions for ensuring global and fast convergence of the procedure and so can be used in our context.

We named the iterative procedure given here Constrained Dogleg (`CoDo`) method. We implemented the `CoDo` method in a `Matlab` code called `CoDoSol` (Constrained Dogleg Solver). This solver is freely accessible through the web site http://codosol.de.unifi.it, and its numerical behavior is showed here.

Features and capabilities of `CoDoSol` have been tested by extensive numerical experiments on a number of representative test problems. First of all, we verified the basic effectiveness of our proposal. To this end, we used the Coleman and Li scaling matrix and compared `CoDoSol` with the affine scaling trust region approach employed in the code STRSCNE [2] and with the `Matlab` implementation of the method IATR given in [4]. Here, we show the resulting numerical results. Their analysis indicates that `CoDoSol` turns out to be an efficient tool to solve medium-scale bound constrained nonlinear systems.

Since different matrices may be used in `CoDoSol`, it is quite important to state if the behavior of the proposed solver is relatively insensitive to variations in the scaling matrix. This question is examined by analyzing the numerical performances of `CoDoSol` for a set of suitable scaling matrices. We give the numerical results obtained and make in significant evidence the results of our comparison by the well known performance profile approach.

The paper is organized as follows. In Sect. 2 we present the constrained dogleg method and describe its convergence properties. In Sect. 3 we consider several scaling matrices from the numerical optimization literature and show how they match our requirements. In Sect. 4 we give a brief account of the `Matlab` solver `CoDoSol` and report on our computational experience. Some conclusions are presented in Sect. 5.

### 1.1 Notation

Throughout the paper we use the following notation. For any mapping $F : X \to \mathbb{R}^n$, differentiable at a point $x \in X \subset \mathbb{R}^n$, the Jacobian matrix of $F$ at $x$ is denoted by $F'(x)$ and $F(x_k)$ is denoted by $F_k$. To represent the $i$-th component of $x$ the symbol $(x)_i$ is used but, when clear from the context, the brackets are omitted. For any vector $y \in \mathbb{R}^n$, the 2-norm is denoted by $\|y\|$ and the open ball with center $y$ and radius $\rho$ is indicated by $B_\rho(y)$, i.e. $B_\rho(y) = \{x : \|x - y\| < \rho\}$.

## 2 Constrained Dogleg methods

In this section we discuss our approach that falls in the well known affine scaling interior point Newton methods. It is well known that every solution $x^*$ of the given problem (1) is also a solution of the box constrained optimization problem:

$$\min_{x \in \Omega} f(x) \tag{2}$$

with

$$f(x) = \frac{1}{2}\|F(x)\|^2.$$

Conversely, if $x^*$ is a minimum of (2) and $f(x^*) = 0$, then $x^*$ solves (1).

As shown by Heinkenschloss et al. in [13], the first order optimality conditions for problem (2) may be rewritten as the nonlinear system of equations

$$D\big(x^*\big)\nabla f\big(x^*\big) = 0, \tag{3}$$

where $\nabla f(x) = F'(x)^T F(x)$ and $D(x)$ is a proper diagonal scaling matrix of order $n$ with diagonal elements satisfying

$$d_i(x) \begin{cases} = 0 & \text{if } x_i = l_i \text{ and } \nabla f(x)_i > 0, \\ = 0 & \text{if } x_i = u_i \text{ and } \nabla f(x)_i < 0, \\ \geq 0 & \text{if } x_i \in \{l_i, u_i\} \text{ and } \nabla f(x)_i = 0, \\ > 0 & \text{otherwise.} \end{cases} \tag{4}$$

A rather general class of scaling matrices satisfying above requirements is suitable to define globally convergent affine scaling methods for the solution of the bound constrained problem (1). In particular, for such matrices, the direction of the scaled gradient $\hat{g}_k$ defined by

$$\hat{g}_k = -D_k \nabla f_k, \tag{5}$$

can be used to implicitly handle the bounds by means of the diagonal matrix $D_k = D(x_k)$ and to provide global convergence (see [3]). In this context, let $\lambda_k$ be the stepsize along $\hat{g}_k$ to the boundary, i.e.

$$\lambda_k = \min_{1\le i\le n} \Lambda_i \quad \text{where } \Lambda_i = \begin{cases} \max\left\{\frac{l_i-(x_k)_i}{(\hat{g}_k)_i}, \frac{u_i-(x_k)_i}{(\hat{g}_k)_i}\right\} & \text{if } (\hat{g}_k)_i \ne 0, \\ \infty & \text{if } (\hat{g}_k)_i = 0 \end{cases} \tag{6}$$

and let us consider scaling matrices satisfying the following properties:

**Assumption 1**

(i) $D(x)$ satisfies (4);
(ii) $D(x)$ is bounded in $\Omega \cap B_\rho(x)$ for any $x \in \Omega$ and $\rho > 0$;
(iii) there exists a $\bar{\lambda} > 0$ such that the stepsize $\lambda_k$ to the boundary from $x_k$ along $\hat{g}_k$ given by (6) satisfies $\lambda_k > \bar{\lambda}$ whenever $\|\nabla f_k\|$ is uniformly bounded above;
(iv) for any $\bar{x}$ in $int(\Omega)$ there exist two positive constants $\bar{\rho}$ and $\chi_{\bar{x}}$ such that $B_\rho(\bar{x}) \subset int(\Omega)$ and $\|D(x)^{-1}\| \le \chi_{\bar{x}}$ for any $x$ in $B_{\bar{\rho}/2}(\bar{x})$.

We remark that (iii) implies the constraint compatibility of $\hat{g}_k$: this property avoids the problem of running directly into a bound by ensuring that the stepsize to the boundary remains bounded away from zero. Furthermore, it is straightforward to note that, as $D(x)$ satisfies (4), it is nonsingular for $x \in int(\Omega)$.

Given an iterate $x_k \in int(\Omega)$ and the trust region size $\Delta_k > 0$, we consider the following trust region subproblem

$$\min_{p\in\mathbb{R}^n} \left\{ m_k(p) : \|G_k p\| \le \Delta_k,\ x_k + p \in int(\Omega) \right\} \tag{7}$$

where $m_k$ is the norm of the linear model for $F(x)$ at $x_k$, i.e.

$$m_k(p) = \left\| F_k + F'_k p \right\| \tag{8}$$

and $G_k = G(x_k) \in \mathbb{R}^{n\times n}$ with $G : \mathbb{R}^n \mapsto \mathbb{R}^{n\times n}$. Different choices for the matrix $G$ lead to different algorithms. In particular, the choice $G_k = I$ yields the standard spherical trust region and $G_k = D_k^{-1/2}$ leads to an elliptical trust region framework.

A suitable way of approximating the solution of (7) is the following dogleg method. Let $p_k^N$ be the Newton step satisfying

$$F'_k\, p_k^N = -F_k. \tag{9}$$

Since $p_k^N$ does not guarantee that $x_k + p_k^N$ is a feasible point, we consider the projection of $x_k + p_k^N$ onto $\Omega$ followed by a step toward the interior of the feasible set. In other words, we consider the step $\bar{p}_k^N$ given by:

$$\bar{p}_k^N = \alpha_k \left( P\left(x_k + p_k^N\right) - x_k \right), \quad \alpha_k \in (0, 1), \tag{10}$$

where $P(x)$ is the projection of $x$ onto $\Omega$, i.e. $P(x)_i = \max\{l_i, \min\{x_i, u_i\}\}$, $1 \le i \le n$. Clearly, the point $x_k + \bar{p}_k^N$ is strictly feasible (see Fig. 1) and we have

$$\|\bar{p}_k^N\| < \|p_k^N\|. \tag{11}$$

To define the dogleg curve and find the next iterate, we move along the scaled gradient direction $\hat{g}_k$ and locate the so-called generalized Cauchy point $p_c(\Delta_k)$, i.e.

the minimizer of (8) along $\hat{g}_k$ constrained to be in the trust region and to satisfy $x_k + p_c(\Delta_k) \in int(\Omega)$. The vector $p_c(\Delta_k)$ has the form

$$p_c(\Delta_k) = \tau_k \hat{g}_k, \tag{12}$$

with $\hat{g}_k$ given by (5) and the scalar $\tau_k$ given by

$$\tau_k = \begin{cases} \tau_k' & \text{if } x_k + \tau_k' \hat{g}_k \in int(\Omega), \\ \theta\lambda_k, \theta \in (0,1) & \text{otherwise,} \end{cases} \tag{13}$$

where $\lambda_k$ is the stepsize along $\hat{g}_k$ to the boundaries, i.e. (6), and $\tau_k'$ is computed in the following way

$$\tau_k' = \operatorname*{argmin}_{\|\tau G_k \hat{g}_k\| \le \Delta_k} m_k(\tau \hat{g}_k) = \min\left\{-\frac{F_k^T F_k' \hat{g}_k}{\|F_k' \hat{g}_k\|^2}, \frac{\Delta_k}{\|G_k \hat{g}_k\|}\right\}. \tag{14}$$

Now, we consider the linear path $p(\gamma)$ given by:

$$p(\gamma) = (1-\gamma) p_c(\Delta_k) + \gamma \bar{p}_k^N, \quad \gamma \in \mathbb{R}, \tag{15}$$

and we look for the value of $\gamma$ minimizing the model $m_k(p)$, i.e. (8), along $p(\gamma)$ within the strictly feasible set and the trust region. In [3], the authors show that the convex function $\phi$ given by

$$\phi(\gamma) = \|F_k + F_k' p(\gamma)\|$$

reaches a minimum at $\gamma = \hat{\gamma}$ with

$$\hat{\gamma} = -\frac{a^T b}{b^T b} = -\frac{(F_k + F_k' p_c(\Delta_k))^T F_k' (\bar{p}_k^N - p_c(\Delta_k))}{\|F_k'(\bar{p}_k^N - p_c(\Delta_k))\|^2}, \tag{16}$$

and that $p(\gamma)$ has two intersections with the trust region boundary, at $\gamma = \gamma_\pm$ with $\gamma_\pm$ given by:

$$\begin{aligned} \gamma_\pm = \Big(& p_c(\Delta_k)^T G_k^2 \big(p_c(\Delta_k) - \bar{p}_k^N\big) \pm \Big(\big(p_c(\Delta_k)^T G_k^2 \big(p_c(\Delta_k) - \bar{p}_k^N\big)\big)^2 \\ & - \big\|G_k\big(p_c(\Delta_k) - \bar{p}_k^N\big)\big\|^2 \big(\big\|G_k p_c(\Delta_k)\big\|^2 - \Delta_k^2\big)\Big)^{1/2}\Big) / \\ & \big\|G_k\big(p_c(\Delta_k) - \bar{p}_k^N\big)\big\|^2. \end{aligned} \tag{17}$$

We remark that both $p_c(\Delta_k)$ and $\bar{p}_k^N$ are feasible steps. Then, $x_k + p(\gamma)$ belongs to the interior of $\Omega$ if $\gamma \in [0, 1]$. But, if we move along $p(\gamma)$ with $\gamma < 0$ or $\gamma > 1$ we need to check if the new point $x_k + p(\gamma)$ is strictly feasible and shorten the step if necessary. Then, we take into account that $p(\gamma)$ is given by

$$p(\gamma) = p_c(\Delta_k) + \gamma\big(\bar{p}_k^N - p_c(\Delta_k)\big),$$

and we consider the stepsize to the boundary from $x_k + p_c(\Delta_k)$ along $\bar{p}_k^N - p_c(\Delta_k)$. So, if $\gamma > 1$, we set:

$$\Lambda_i = \begin{cases} \max\{\frac{l_i-((x_k)_i+(p_c(\Delta_k))_i)}{(\bar{p}_k^N-p_c(\Delta_k))_i}, \frac{u_i-((x_k)_i+(p_c(\Delta_k))_i)}{(\bar{p}_k^N-p_c(\Delta_k))_i}\} & \text{if } (\bar{p}_k^N - p_c(\Delta_k))_i \neq 0, \\ +\infty & \text{if } (\bar{p}_k^N - p_c(\Delta_k))_i = 0 \end{cases}$$

and take

$$\bar{\gamma}_+ = \min_i \Lambda_i(p), \tag{18}$$

whereas if $\gamma < 0$ we set:

$$\Lambda_i = \begin{cases} \max\{\frac{l_i-((x_k)_i+(p_c(\Delta_k))_i)}{-(\bar{p}_k^N-p_c(\Delta_k))_i}, \frac{u_i-((x_k)_i+(p_c(\Delta_k))_i)}{-(\bar{p}_k^N-p_c(\Delta_k))_i}\} & \text{if } (-\bar{p}_k^N + p_c(\Delta_k))_i \neq 0, \\ +\infty & \text{if } (-\bar{p}_k^N + p_c(\Delta_k))_i = 0 \end{cases}$$

and

$$\bar{\gamma}_- = -\min_i \Lambda_i(p). \tag{19}$$

To summarize, the choice of $\gamma$ is made as follows. Since we want to minimize $\|F_k + F_k' p(\gamma)\|$, we seek $\gamma = \hat{\gamma}$ given by (16). Moreover, since $p(\gamma)$ must belong to the trust region and $x_k + p(\gamma)$ is required to be strictly feasible, we choose $\gamma = \min(\hat{\gamma}, \gamma_+, \theta\bar{\gamma}_+)$ if $\hat{\gamma} > 0$, whereas if $\hat{\gamma} < 0$, we choose $\gamma = \max(\hat{\gamma}, \gamma_-, \theta\bar{\gamma}_-)$, with $\theta \in (0, 1)$, $\gamma_\pm$, $\bar{\gamma}_+$, $\bar{\gamma}_-$ given by (17), (18), (19), respectively. With $\gamma$ at hand, we compute $p(\gamma)$ by (15) and we set the trial step $p(\Delta_k) = p(\gamma)$.

In brief, the following is the procedure we use for determining the trial steps.

STEP SELECTION PROCEDURE

Input parameters: $x_k \in int(\Omega)$, $\Delta_k > 0$, $\hat{g}_k$, $\bar{p}_k^N$, $\theta \in (0, 1)$
  Compute $p_c(\Delta_k)$ by (12) and (13).
  Compute $\hat{\gamma}$ by (16).
  If $\hat{\gamma} > 0$
    compute $\gamma_+$ by (17)
    compute $\bar{\gamma}_+$ by (18)
    set $\gamma = \min\{\hat{\gamma}, \gamma_+, \theta\bar{\gamma}_+\}$
  Else
    compute $\gamma_-$ by (17)
    compute $\bar{\gamma}_-$ by (19)
    set $\gamma = \min\{\hat{\gamma}, \gamma_-, \theta\bar{\gamma}_-\}$
  Set $p(\Delta_k) = (1-\gamma)p_c(\Delta_k) + \gamma\bar{p}_k^N$

In Fig. 1 we can see how our step selection works. In this figure, the dotted ellipses represent level curves of the local linear model norm, the solid ellipse represents the trust region boundary and the box is the domain $\Omega$. The linear model is minimized along the segment in bold belonging to the line connecting the generalized Cauchy step and the projected Newton step.

It is worth noting that above choice of the trial step produces a decrease in the value of the model which is at least the decrease provided by the generalized Cauchy

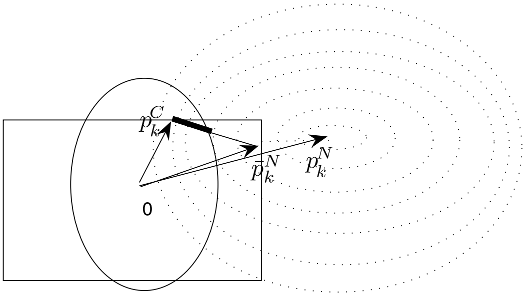


**Fig. 1** Illustrative constrained path $p(\gamma)$ in $\mathbb{R}^2$. The segment in *bold* is the path along with the model is minimized

step. In other words, the step satisfies the condition

$$\rho_c\big(p(\Delta_k)\big) = \frac{\|F_k\| - \|F_k + F_k' p(\Delta_k)\|}{\|F_k\| - \|F_k + F_k' p_c(\Delta_k)\|} \geq 1. \tag{20}$$

On the other hand, a trial step $p(\Delta_k)$ will be used to form the next iterate if the point $x_k + p(\Delta_k)$ produces a reduction of $\|F\|$ sufficiently large compared with the reduction predicted by the local linear model. Then, we test if the following sufficient improvement condition

$$\rho_f\big(p(\Delta_k)\big) = \frac{\|F_k\| - \|F(x_k + p(\Delta_k))\|}{\|F_k\| - \|F_k + F_k' p(\Delta_k)\|} \geq \beta \tag{21}$$

holds for a given constant $\beta \in (0, 1)$ independent of $k$. If (21) is satisfied, then $p(\Delta_k)$ is accepted, the new iterate $x_{k+1} = x_k + p(\Delta_k)$ is formed and the trust region radius may be increased. Otherwise, $p(\Delta_k)$ is rejected, $\Delta_k$ is shrunk and a new trial step is computed.

The following is our general constrained dogleg method where the positive constant $\Delta_{\min}$ is a lower bound on the initial trust region size allowed at each iteration.

Constrained Dogleg Method

Input parameters: the starting point $x_0 \in int(\Omega)$, the scaling matrix $D(x)$, the matrix $G(x)$, the scalar $\Delta_{\min} > 0$, the initial trust region size $\bar{\Delta}_0 > \Delta_{\min}$, the constants $\beta \in (0, 1)$, $\delta \in (0, 1)$, $\theta \in (0, 1)$.

For $k = 0, 1, \ldots$
1. Set $\Delta_k = \bar{\Delta}_k$.
2. Choose $\alpha_k \in (0, 1)$.
3. Compute the solution $p_k^N$ to (9).
4. Form $\bar{p}_k^N$ by (10).
5. Set $\hat{g}_k = -D_k \nabla f_k$.
6. Find $p(\Delta_k)$ by the Step Selection Procedure.
7. While $\rho_f(p(\Delta_k)) < \beta$
    7.1 Set $\Delta_k = \delta \Delta_k$.
    7.2 Find $p(\Delta_k)$ by the Step Selection Procedure.
8. Set $x_{k+1} = x_k + p(\Delta_k)$.
9. Choose $\bar{\Delta}_{k+1} > \Delta_{\min}$.

The Constrained Dogleg method outlined in the above algorithm is globally and fast locally convergent, as it can be easily derived from the convergence analysis carried out in [3]. In fact, from Theorem 3.2 of [3] the following global convergence result holds:

**Theorem 1** *Let* (i)–(iii) *in Assumption* 1 *be satisfied. If the sequence* $\{x_k\}$ *generated by the* CoDo *Method is bounded*, *then*:

- *All the limit points of* $\{x_k\}$ *are stationary points for problem* (2).
- *If there exists a limit point* $x^* \in int(\Omega)$ *of* $\{x_k\}$ *such that* $F'(x^*)$ *is nonsingular, then* $\|F_k\| \to 0$ *and all the accumulation points of* $\{x_k\}$ *solve problem* (1).
- *If there exists a limit point* $x^* \in \Omega$ *such that* $F(x^*) = 0$ *and* $F'(x^*)$ *is invertible, then* $x_k \to x^*$.

Moreover, Theorem 3.3 of [3] yields the following asymptotic convergence result:

**Theorem 2** *Let* (i)–(iv) *in Assumption* 1 *be satisfied*, $\|F'\|$ *be bounded above on*

$$L = \bigcup_{k=0}^{\infty} \bigl\{x \in X \ : \ \|x - x_k\| \le r\bigr\}, \quad r > 0,$$

*and* $F'$ *be Lipschitz continuous in an open*, *convex set containing* $L$. *Assume that there exists a solution* $x^*$ *of* (1) *such that* $F'(x^*)$ *is nonsingular and that the sequence* $\{x_k\}$ *converges to* $x^*$. *If* $\alpha_k$ *in* (10) *satisfies* $\alpha_k \to 1$, *as* $k \to \infty$, *and*

- *either* $G_k = I$, $k \ge 0$, *or*
- $G_k = D_k^{-1/2}$, $k \ge 0$, *and* $\|G_k p_k^N\| \to 0$ *as* $k \to \infty$,

*then*, *eventually*, $p(\bar{\Delta}_k)$ *satisfies* (21) *and the sequence* $\{x_k\}$ *converges to* $x^*$ *superlinearly. Moreover*, *if*

$$\alpha_k = 1 - O\bigl(\|F_k\|\bigr) \quad as\ k \to \infty,$$

*the convergence rate is quadratic*.

Summarizing, suitable choices of the scalar $\alpha_k$ in the computation of the projected Newton step ensure quadratic convergence of CoDo method whenever spherical trust region are used (i.e. $G_k = I$, $k \ge 0$), independently of the position of the solution. On the other hand, when elliptical trust regions are employed (i.e. $G_k = D_k^{-1/2}$, $k \ge 0$), fast convergence is ensured whenever the solution lies in the interior of the feasible set, while fast convergence is not guaranteed to solutions on the boundary of $\Omega$.

From above presentation, this procedure clearly is a generalization of the classical dogleg methods based on projected Newton Step. Presence of bounds is taken into account as well. This approach is expected to work better than those employed in [1, 2, 4] as the double-dogleg procedure is avoided here. More precisely, in [1, 2] a classical dogleg step is computed and then it is truncated to produce a feasible point. This projected dogleg step is rejected if it does not satisfy the Cauchy decrease condition. In this latter case, the scaled Cauchy step is employed. Then, all the computational work spent to compute the Newton step is completely lost. On the other hand, in [4], a first dogleg procedure is invoked to compute a classical dogleg step. Then, a projected dogleg step is computed and, whenever it does not satisfy the Cauchy condition, a further dogleg procedure is applied along the path connecting the scaled Cauchy step and the projected dogleg step. In these cases, the Newton step still contribute to the formation of the step, but requires two dogleg procedures. In the approach used in the present work, this double-dogleg is not required at all.

## 3 Scaling matrices

We remark that different methods in the above scheme distinguish themselves by the choice of the scaling matrix $D(x)$. Further, it is important for global and locally fast convergence to consider diagonal scalings from the rather general class of matrices satisfying Assumption 1. In this section we consider several scaling matrices proposed by different authors in the numerical optimization literature and we show how they match our requirements. More specifically, we consider the following diagonal matrices.

– $D^{CL}(x)$ given by Coleman and Li [6]. The diagonal entries are:

$$d_i^{CL}(x) = \begin{cases} u_i - x_i & \text{if } (\nabla f(x))_i < 0 \text{ and } u_i < \infty, \\ x_i - l_i & \text{if } (\nabla f(x))_i > 0 \text{ and } l_i > -\infty, \\ \min\{x_i - l_i, u_i - x_i\} & \text{if } (\nabla f(x))_i = 0 \\ & \text{and } l_i > -\infty \text{ or } u_i < \infty, \\ 1 & \text{otherwise.} \end{cases} \tag{22}$$

– $D^{HUU}(x)$ given by Heinkenschloss et al. in [13]. The diagonal entries have the following form:

$$d_i^{HUU}(x) = \begin{cases} d_i^{CL}(x) & \text{if } |\nabla f(x)_i| < \min\{x_i - l_i, u_i - x_i\}^p \text{ or} \\ & \min\{x_i - l_i, u_i - x_i\} < |\nabla f(x)_i|^p, \\ 1 & \text{otherwise} \end{cases} \tag{23}$$

where $p > 1$ is a fixed constant.

– $D^{KK}(x)$ given by Kanzow and Klug [15, 16]. The diagonal entries are:

$$d_i^{KK}(x) = \begin{cases} 1 \quad \text{if } l_i = -\infty \text{ and } u_i = +\infty \\ \min\{x_i - l_i + \gamma \max\{0, -\nabla f(x)_i\}, \\ u_i - x_i + \gamma \max\{0, \nabla f(x)_i\}\} \quad \text{otherwise} \end{cases} \tag{24}$$

for a given constant $\gamma > 0$.

– $D^{HMZ}(x)$ implicitly given by Hager et al. in [12]. The diagonal entries are:

$$d_i^{HMZ}(x) = \frac{X_i(x)}{\alpha(x) X_i(x) + |\nabla f(x)_i|} \tag{25}$$

being

$$X_i(x) = \begin{cases} u_i - x_i & \text{if } \nabla f(x)_i < 0 \text{ and } u_i < \infty, \\ x_i - l_i & \text{if } \nabla f(x)_i > 0 \text{ and } l_i > -\infty, \\ 1 & \text{if } \nabla f(x)_i = 0 \end{cases} \tag{26}$$

and $\alpha(x)$ is a continuous function, strictly positive for any $x$ and uniformly bounded away from zero.

Concerning this last matrix, we underline that in [12], the authors do not give explicitly the scaling matrix $D^{HMZ}(x)$. In fact, they study a cyclic Barzilai-Borwein gradient method for bound constrained minimization problems and replace the Hessian of the objective function with $\lambda_k I$, where $\lambda_k$ is the classical Barzilai-Borwein parameter. Then, in order to compute the new iterate, they move along the scaled gradient $d_k = -D^{HMZ}(x_k)\nabla f(x_k)$, with $\alpha(x_k) = \lambda_k$ in (25).

In what follows, we verify if the above scaling matrices satisfy the four requirements specified in Assumption 1.

(i) The scaling matrices $D^{CL}$ , $D^{HUU}$ and $D^{KK}$ clearly satisfy this condition. In fact, as noted in [13] and [16], they satisfy (4). It is easy to prove that also $D^{HMZ}(x)$ given by (25) satisfies (4). Indeed, we clearly have $d_i(x_k) \geq 0$. In particular, if $(x_k)_i = l_i$ and $\nabla f(x_k)_i > 0$ or if $(x_k)_i = u_i$ and $\nabla f(x_k)_i < 0$ we have $X_i(x_k) = 0$, hence $d_i^{HMZ}(x_k) = 0$. If $\nabla f(x_k)_i = 0$ and $(x_k)_i \in \{l_i, u_i\}$ we have $d_i^{HMZ}(x_k) \geq 0$. Finally, if $l_i < (x_k)_i < u_i$, we clearly have $d_i^{HMZ}(x_k) > 0$. Then, all the matrices verify condition (i) in Assumption 1.

(ii) This condition is satisfied by all the matrices as for any $\bar{x} \in B_\rho(x)$, we have $u_i - \bar{x}_i \leq u_i - x_i + \rho$ and $\bar{x}_i - l_i \leq x_i + \rho - l_i$. Note that in all the four matrices the term $u_i - x_i$ appears only if $u_i$ is finite and the same is true for $x_i - l_i$.

(iii) To discuss this condition, it is useful to note that, given $\hat{g}_k = -D_k \nabla f_k$, we have

$$\max\left\{\frac{l_i - (x_k)_i}{(\hat{g}_k)_i}, \frac{u_i - (x_k)_i}{(\hat{g}_k)_i}\right\} = \begin{cases} \frac{l_i - (x_k)_i}{(\hat{g}_k)_i} & \text{if } (\nabla f(x))_i > 0, \\ \frac{u_i - (x_k)_i}{(\hat{g}_k)_i} & \text{if } (\nabla f(x))_i < 0. \end{cases} \tag{27}$$

Let us consider the $D^{CL}(x)$ scaling matrix. From (27), (6) and (22) we have

$$\lambda_k = \min_i \left(\frac{1}{|(\nabla f_k)_i|}\right) \geq \frac{1}{\|\nabla f_k\|_\infty}, \tag{28}$$

and therefore condition (iii) is satisfied.

$D^{HUU}(x)$ given by (23) does not join the nice property (iii). In fact, let us focus on the case $\min\{x_i - l_i, u_i - x_i\}^p < |\nabla f(x)_i| < (\min\{x_i - l_i, u_i - x_i\})^{1/p}$ and $(\nabla f_k)_i > 0$. In this case $d_i^{HUU}(x) = 1$ and $\frac{l_i - (x_k)_i}{(\hat{g}_k)_i} = \frac{(x_k)_i - l_i}{(\nabla f_k)_i}$. Note that, in this case, as $(x_k)_i$ approaches $l_i$, $(\nabla f_k)_i$ tends to zero and $\frac{(x_k)_i - l_i}{(\nabla f_k)_i}$ is not guaranteed to be bounded away from zero.

$D^{KK}(x)$ is defined by (24). From its definition we have $d_i^{KK}(x) \le x_i - l_i$, whenever $(\nabla f_k)_i > 0$. Then, in this latter case from (27) and (22) we have

$$\lambda_k = \frac{l_i - (x_k)_i}{(\hat{g}_k)_i} = \frac{(x_k)_i - l_i}{d_i^{KK}(x_k)(\nabla f_k)_i} \ge \frac{1}{(\nabla f_k)_i}.$$

Similarly, if $(\nabla f_k)_i < 0$, we have $d_i^{KK}(x) \le u_i - x_i$ and we get

$$\lambda_k = \frac{u_i - (x_k)_i}{-d_i^{KK}(x_k)(\nabla f_k)_i} \ge \frac{1}{|(\nabla f_k)_i|}.$$

This implies

$$\lambda_k \ge \min_i \left( \frac{1}{|(\nabla f_k)_i|} \right) \ge \frac{1}{\|\nabla f_k\|_\infty}$$

and then, $D^{KK}$ verifies condition (iii).

$D^{HMZ}(x)$ verifies condition (iii), too. In fact, from Lemma 3.4 in [12] it follows that $\lambda_k > 1$ and this ensures that condition (iii) is satisfied.

(iv) This condition is satisfied by all the matrices considered. In fact, since $d_i^{CL}(x) \ge \min\{x_i - l_i, u_i - x_i\} \ge \rho/2$, it is easy to see that $\|(D^{CL}(x))^{-1}\| \le 2/\rho$ whenever $x \in B_{\rho/2}(\bar{x})$ with $\bar{x} \in int(\Omega)$ and $\rho$ sufficiently small (see also [1, Corollary 3.1]). The same is true, clearly, for the $D^{HUU}(x)$ and $D^{KK}(x)$ matrices. Regarding the $D^{HMZ}(x)$ matrix, we have $X_i(x) \ge \min\{x_i - l_i, u_i - x_i\} \ge \rho/2$. Furthermore $X_i(x)$ and $\nabla f(x)_i$ are bounded above in $B_{\rho/2}(\bar{x})$ and this yields boundedness for $(D^{HMZ}(x))^{-1}$ in $B_{\rho/2}(\bar{x})$.

To conclude, the scaling matrices $D^{CL}(x)$ , $D^{KK}(x)$ and $D^{HMZ}(x)$ match the four conditions of Assumption 1 while $D^{HUU}(x)$ is not suitable for our constrained dogleg approach, as the constrained compatibility condition (iii) is not ensured to hold.

We end this section with the following general observations that stress some theoretical implications of choosing the scaling matrix. First of all, we note that $D^{CL}(x)$ is, in general, discontinuous at points where there exists an index $i$ such that $\nabla f(x)_i = 0$ and this may happen even at the solution, while the scaling matrix $D^{KK}(x)$ has the advantage of being locally Lipschitz continuous and, finally, scaling matrix $D^{HMZ}(x)$ (25) is continuous.

We remark that in [15, 16], Kanzow and Klug motivate the choice of the scaling matrix $D^{KK}(x)$ and carefully analyze the differences with $D^{CL}(x)$. Their analysis justifies our interest in investigating the effect of $D^{KK}(x)$ in our context, too.

Further remarks arise from a comparison between $D^{CL}(x)$ and $D^{HMZ}(x)$. The Coleman-Li matrix $D^{CL}(x)$ takes into account the distance from $x$ to the boundary of $\Omega$ while the scaling $D^{HMZ}$ takes into account the value of $\nabla f(x)_i$: the larger it is, the smaller is the scaling. To make evident the effects of this scaling let us assume, for the sake of simplicity, $\Omega = \{x \in \mathbb{R}^n : x \ge 0\}$ and consider a point $x \in \Omega$ close to $\partial\Omega$ such that the component $x_i$ is small while $\nabla f(x)_i$ is large and positive. The Coleman-Li matrix $D^{CL}(x)$ prevents from taking a step in the $-\nabla f(x)$ direction which is too large along the $i$th axis. In fact, we have

$$\left(D^{CL}(x)\nabla f(x)\right)_i = x_i \nabla f(x)_i$$

which, if $x_i$ is small, is much smaller than $\nabla f(x)_i$. Further, since $D^{CL}$ does not depend on the value of $\nabla f(x)_i$ it follows that $d_i^{CL}(x) = x_i$ independently of the value of $\nabla f(x)_i$. For this reason, if $\|\nabla f_k\|_\infty$ is not small, it is not guaranteed that a full-step may be taken in the scaled direction without violating the bounds (see (28)). In other words, the scaling $D^{CL}$ ensures that the distance to the boundary along the scaled gradient is bounded away from zero. The effectiveness of the scaling $D^{HMZ}$ is more evident. In fact, as previously remarked, Lemma 3.4 in [12] ensures that the distance to the boundary along the scaled gradient is bounded away from one when the scaling $D^{HMZ}$ is used. Further, we remark that

$$\left(D^{HMZ}(x)\nabla f(x)\right)_i = \frac{x_i}{\alpha(x)x_i + \nabla f(x)_i}\nabla f(x)_i,$$

and then, if $x_i$ is small, we have $(D^{HMZ}(x)\nabla f(x))_i \simeq \frac{x_i}{\nabla f(x)_i}\nabla f(x)_i = x_i$, i.e. the $i$th component of the step is not reduced by a factor $x_i$, but has length $x_i$. In other words, since the $i$th component of the scaled gradient is essentially the distance from the boundary, a full-step along the scaled direction may be taken, without violating the bounds. We remark that, since the $i$th component of the gradient scaled by $D^{HMZ}(x)$ results essentially equal to $x_i$, the scaled gradient has actually lost any gradient-related information. This is a possible drawback of using $D^{HMZ}(x)$ when getting onto situations as above described. On the other hand, when $x_i \gg 0$ and $\nabla f(x)_i$ is close to zero, the scaling $D^{HMZ}(x)$ reduces to the value $1/\alpha(x)$ and does not take into account the distance from the boundary, while the Coleman and Li $D^{CL}(x)$ scales the gradient by $x_i$.

## 4 Experimental studies

In this section, we report on the numerical experiments we performed to prove the computational feasibility of the proposed approach and give general information about its numerical performance. Our primary goal is to show the basic effectiveness of the proposed dogleg approach compared with the affine scaling trust region methods STRSCNE given in [1, 2] and IATR given in [4]. Then, the numerical behavior of the new algorithm for a set of alternative scaling matrices is analyzed.

We implemented the Constrained Dogleg method in the `Matlab` code `CoDoSol` and, in the following, we give a brief account of this solver. Then, we discuss the details of the major issues addressed in performing the numerical experiments and describe the set of test problems used. Finally, we show the results obtained.

### 4.1 The `Matlab` solver `CoDoSol`

We implemented the Constrained Dogleg method, with both spherical and elliptical trust region, in the `Matlab` code `CoDoSol`. This solver is freely accessible through the web site:

http://codosol.de.unifi.it

Its usage is carefully described in helpful comments making easy to understand the use of multiple input and output arguments. The simplest usage of `CoDoSol` is to write a function that evaluates the given $F$ and then call the solver. The minimum information that the solver must be given is the initial point $x_0$, the name of the function defining the system, and the bounds defining the feasible region.

A finite difference approximation to the Jacobian is provided, freeing the user from computing the derivatives of $F$. However, if the Jacobian of $F$ is available in analytic form, the user can provide the code to compute it.

The user can choose among the three scaling matrices: $D^{CL}(x)$, $D^{KK}(x)$, and $D^{HMZ}(x)$ or can apply his own scaling matrix. Moreover, spherical or elliptical trust region may be selected.

The default choice is elliptical trust region in conjunction with Coleman-Li scaling matrix.

If the problem to be solved has sparse Jacobian and a relatively big size, the user can choose to work with sparse memory storage. Then, the Newton step is computed via the built-in `Matlab` function `LU` with the syntax for calling the `UMFPACK` package [7], when `Matlab 6.5` or later versions are used.

Several different output levels may be requested by the user. The convergence history of the algorithm and a variety of diagnostic information allow the user to be safeguarded against unsatisfactory approximations of the required solution.

In particular, in the code different values of an output flag `ierr` are provided corresponding to the following situations:

0 upon successful termination, i.e. fulfillment of the condition `norm(F_k)<=tol`;
1 the limiting number of iterations has been reached;
2 the limiting number of $F$-evaluations has been reached;
3 the trust region radius Delta has become too small (`Delta<sqrt(eps)`);
4 no improvement for the nonlinear residual could be obtained:
`abs(norm(F(x_k))-norm(F(x_{k-1})))`
`<=100*eps*norm(F(x_k));`
5 the sequence has approached a minimum of $f$ in the box:
`norm(D(x_k)*grad(f(x_k)))<100*eps;`
6 an overflow would be generated when computing the scaling matrix $D$ since the sequence is approaching a bound.

Finally, we point out here how the code deals with possible singular Jacobian matrices. If $F'_k$ is singular, the new step is just set equal to the Cauchy step. We remark that the code only checks an "exact" singularity of $F'_k$. A nearly singular Jacobian is not discarded. In fact, if a nearly singular Jacobian is encountered, we expect a "bad" Newton step to be computed. If this is the case, the step selection rule automatically neglects the contribution of the Newton step by choosing a $\gamma$ very close to 0. On the other hand, if the step computed—despite obtained with an inaccurate solution of the Newton step—satisfies sufficient improvement conditions (20)–(21), it means that the new step is a "good" step for our purposes and it is kept.

## 4.2 The numerical experiments

All numerical experiments have been performed on a 3.4 GHz Intel Xeon (TM) with 1 GB of RAM using the `Matlab 7.6` version of the code `CoDoSol` and machine precision $\epsilon_m \approx 2 \cdot 10^{-16}$.

The experiments were carried out on a set of 36 problems with dimension between $n = 2$ and $n = 12500$, specified in Tables 1 and 2. This set of problems provides us with various types of representative constrained systems, and includes systems with solutions both within the feasible region $\Omega$ and on the boundary of $\Omega$, systems with only lower (upper) bounds and systems with variable components bounded from above and below.

To form this set we used the nonlinear constrained systems given in Chap. 14 of [10] (problems Pb1 to Pb6) and in [28] (problems Pb20 to Pb23), two equilibrium problems modeled by parameter dependent nonlinear equations (Pb7 and Pb8), four chemical equilibrium systems given in [23, 24, 29] (Pb9 to Pb12), seven nonlinear complementarity problems (NCPs) given in [8, 14, 29] (Pb13 to Pb19). The NCPs were reformulated as systems of smooth box-constrained nonlinear equations (see [30]). Seven problems arise from finite discretization of continuous problems, specifically Pb34 and Pb35 given in [20] are finite differences analogues of two PDE

**Table 1** Test problems equipped with bounds

| Pb # | Name and source | $n$ |
|---|---|---|
| 1 | Bullard-Biegler system [10, 14.1.3] | 2 |
| 2 | Ferraris-Tronconi system [10, 14.1.4] | 2 |
| 3 | Brown's almost linear system [10, 14.1.5] | 5 |
| 4 | Robot kinematics problem [10, 14.1.6] | 8 |
| 5 | Series of CSTRs, $R = .935$ [10, 14.1.8] | 2 |
| 6 | Series of CSTRs, $R = .995$ [10, 14.1.8] | 2 |
| 7 | Chemical reaction problem [19, Problem 5] | 67 |
| 8 | A Mildly-Nonlinear BVP [19, Problem 7] | 451 |
| 9 | Chemical equilibrium system [24, system 1] | 11 |
| 10 | Chemical equilibrium system [24, system 2] | 5 |
| 11 | Combustion system (Lean case) [23] | 10 |
| 12 | Combustion system (Rich case) [23] | 10 |
| 13 | Josephy problem [8] | 8 |
| 14 | Problem HS34 [14] | 16 |
| 15 | Problem Wachter and Biegler [29] | 9 |
| 16 | Bratu NCP [8] | 12500 |
| 17 | Trafelas [8] | 2904 |
| 18 | Opt_cont_31 [8] | 2048 |
| 19 | Obstacle [8] | 7500 |
| 20 | Effati-Grosan 1, $a = 2$ [28] | 2 |
| 21 | Effati-Grosan 1, $a = 100$ [28] | 2 |
| 22 | Effati-Grosan 2, $a = 2$ [28] | 2 |
| 23 | Effati-Grosan 2, $a = 100$ [28] | 2 |

**Table 2** Test problems with bounds added to select specific solutions

| Pb # | Name and source | $n$ | Box |
|---|---|---|---|
| 24 | Discrete boundary value function [26, Problem 28] | 500 | $[-100, 100]$ |
| 25 | Discrete integral [26, Problem 29] | 1000 | $[-10, 10]$ |
| 26 | Trigexp1 [20, Problem 4.4] | 1000 | $[-100, 100]$ |
| 27 | Troesch [20, Problem 4.21] | 500 | $[-1, 1]$ |
| 28 | Trigonometric system [20, Problem 4.3] | 5000 | $[\pi, 2\pi]$ |
| 29 | Tridiagonal exponential [20, Problem 4.18] | 2000 | $[e^{-1}, e]$ |
| 30 | H-equation, $c = 0.99$[17] | 400 | $[0, 5]$ |
| 31 | Countercurrent reactors [20, Problem 4.1] | 10000 | $[-1, 10]$ |
| 32 | Five diagonal [20, Problem 4.8] | 5000 | $[1, \infty]$ |
| 33 | Seven diagonal [20, Problem 4.9] | 5000 | $[0, \infty]$ |
| 34 | Bratu problem [20, Problem 4.24] | 10000 | $[-\infty, 1.5]$ |
| 35 | Poisson problem [20, Problem 4.25] | 10000 | $[-10, 10]$ |
| 36 | Integral equation [18] | 1000 | $[-\infty, 0]$ |

problems, while Pb24 [25] and Pb27 [20] comes from nonlinear BVPs; problems Pb25 and Pb36 are discretization of integral equation and Pb30 is the discretization of the well-known Chandrasekhar H-equation, that we solved with the challenging value $c = 0.99$. Finally, Problems Pb26, Pb28, Pb29, Pb31, Pb32 and Pb33 are widely used nonlinear systems given in the Test Problems collection [20].

Problems Pb1–Pb23 appear in literature already equipped with bounds on the variables, which are specified in the references given in Table 1. Problems Pb24–Pb36 have more than one solution and we added bounds in order to select specific solutions. The bounds used in the numerical results are reported in Table 2.

We performed our experiments starting from good and poor initial guesses. As a general rule, the starting points $x_0 = l + 0.25\nu(u - l)$, $\nu = 1, 2, 3$, have been used for problems having finite lower and upper bounds, whereas $x_0 = 10^\nu(1, \ldots, 1)^T$ and $x_0 = -10^\nu(1, \ldots, 1)^T$, $\nu = 0, 1, 2$, have been used for problems with infinite upper and lower bounds, respectively. We remark that the vector $x_0$ obtained with $\nu = 3$ is solution of problem Pb3. Further, the Jacobian matrices of Pb4 and Pb8 are singular at the starting point obtained with $\nu = 2$. These critical values of $\nu$ have been replaced by $\nu = 2.5$. Moreover, the Jacobians of problems Pb20 and Pb21 are singular at points such that $(x)_1 = (x)_2$. Despite these cases can be addressed by `CoDoSol` (see previous subsection), other codes we used here for comparisons would fail. So, in these cases, we modified the first component of $x_0$ by putting $(x_0)_1 = 0.5$.

Summarizing, 36 problems occurring in applications have been chosen and solved starting from three different initial points for a total of 108 tests.

In `CoDoSol`, the trust region size is updated as in [5], i.e. at the step 7.1 of the `CoDo` algorithm we reduced the trust region radius by setting $\Delta_k = \min\{0.25\,\Delta_k,\ 0.5\,\|p_k\|\}$ and, at the step 9, we allowed the next iteration with an increased trust region radius if condition (21) holds with $\beta = 0.25$ (in this case, we set $\bar{\Delta}_{k+1} = \max\{\Delta_k, 2\|p_k\|\}$) otherwise, we left unchanged the radius. We remark that the parameter $\Delta_{\min}$, which gives a lower bound on the initial trust region radius at each iteration, helps in simplifying the convergence theory and it is an internal parameter in the code. It is set equal to $\sqrt{\epsilon_m}$.

The projected step $\bar{p}_k^N$ is computed by using $\alpha_k = \max\{0.99995, 1 - \|F_k\|\}$ for all $k$.

We stopped the runs when the condition

$$\|F_k\| \le 10^{-6} \tag{29}$$

was met. Such occurrence was indicated as a successful termination.

Failure was declared either if the number of iterations was greater than 300 or if the number of $F$-evaluations was greater than 1000.

Since different algorithms in our constrained dogleg framework distinguish themselves by the choice of the scaling matrix, we tested the algorithm with the scaling matrices analyzed in the previous section and studied the effect of this choice on numerical performance of the resulting Constrained Dogleg method. More specifically, for $D^{KK}(x)$ and $D^{HMZ}(x)$ we decided to adopt the constant choices used in the practical implementations of [16] and [12], respectively.

So, we tested `CoDoSol` in conjunction with the following choices of the scaling matrix $D_k$, at each iteration $k$:

- $D^{CL}(x_k)$ given by (22),
- $D^{KK}(x_k)$ given by (24) with $\gamma = 1$ as suggested in [16],
- $D^{HMZ}(x_k)$ given by (25) where $\alpha_k = \alpha(x_k)$ is computed by the following rule given in [12]:

$$\begin{cases} \alpha_0 = \max(10^{-10}, \ \|\nabla f_0\|), \\ \alpha_k = \max\left(10^{-10}, \ \frac{p_k^T(\nabla f_k - \nabla f_{k-1})}{p_k^T p_k}\right), \quad p_k = x_k - x_{k-1}. \end{cases}$$

## 4.3 The experimental study

We first investigated the robustness and efficiency of the new algorithm compared with the affine scaling trust region methods STRSCNE given in [1, 2] and IATR given in [4]. These methods are based on elliptical trust region approaches and use the Coleman and Li diagonal scaling. So, we applied `CoDoSol` with the options $D(x) = D^{CL}(x)$, $G(x) = D(x)^{-1/2}$, $\Delta_0 = 1$. Further, for sake of comparison, we run the three `Matlab` solvers with the same values of the common parameters and the same choices of algorithmic options. More specifically, we used analytical Jacobian matrices and $\Delta_{\min} = \sqrt{\epsilon_m}$, $\beta = 0.75$, $\theta = 0.99995$, $\delta = 0.25$.

We measured the efficiency of the three algorithms by the number $It$ of iterations and the number $Fe$ of $F$-evaluations performed to obtain convergence. We remark that the computational cost of all the algorithms considered increases with the number of rejected trial steps. In fact, if the initial trial step is accepted, we have $Fe = It + 1$ and so, the measure of efficiency in terms of iteration count is equivalent to considering the number of function evaluations. On the other hand, if $Fe > It + 1$, the number of trial steps rejected is given by $R = Fe - (It + 1)$.

All results are summarized in Tables 3, 4 and 5 where, for each problem, we give the value of the parameter $\nu$ used to compute the initial guess $x_0$ and the value $\|F_0\|$

of $\|F(x_0)\|$.[1] Further, for each method, we show the number of iterations (*It*) and the number of function evaluations (*Fe*) performed to reach convergence. Finally the symbol $*$ indicates a failure and the rather standard notation $m(c)$ denotes $m\cdot 10^c$.

We now comment in more details our results.

The CoDoSol successfully solved 26 test problems starting from all the initial guesses, 5 test problems starting from two of the three $x_0$ used and 4 test problems starting from only one initial guess. It failed with all initial guesses here considered only with one problem. As a whole, the new method succeeded 89 times on a total of 105 runs. Moreover, 50% of failures were detected with output flag $\texttt{ierr} = 1$, that means the stopping criterion (29) was not met within the maximum number of nonlinear iterations; 25% of failures were due to trust region radius becoming too small ($\texttt{ierr} = 3$) and in the remaining failures the solver stopped as a stagnation was detected ($\texttt{ierr} = 4$).

The STRSCNE solver succeeded 85 times: it solved 20 problems starting from all the $x_0$ used, 12 problems starting from two initial guesses and 4 problems with only one $x_0$. It solved all problems with at least one starting guess.

The IATR method was a bit less robust than the other two. On the chosen set of tests it solved 20 problems starting from all the $x_0$, 9 problems starting from two $x_0$ and 5 problems starting from one $x_0$. Further, it failed on two problems with all initial guesses. Summarizing, it succeeded in solving 80 tests.

Tables 3, 4 and 5 provide us with other information on the numerical performance of the constrained dogleg method proposed here. It does not seem to be very expensive since most of the successful runs were performed with a small number of iterations and with a moderate value of $R$.

To make in more significant evidence above observations we compare the three algorithms by adopting the performance profile approach (see [9]). In Fig. 2, the computational effort is measured both in terms of the number *It* of iterations performed (left picture) and in terms of the number *Fe* of $F$-evaluations (right picture). Again, the profiles indicate that the CoDoSol outperforms IATR and STRSCNE as it is more efficient in solving about the 70%/65% (in terms of *It*/*Fe*) of the tests and it solves the 85% of tests within a factor 2 from the best solver. Finally, it fails in solving the 15% of tests. However, it should be taken into account that this percentage of failures decreases to 12% if we exclude Problems Pb13–Pb19, i.e. problems obtained reformulating NCPs. This could be expected as our method does not exploit at all the special structure of NCPs. The performance profiles corresponding to tests with no NCPs are depicted in Fig. 3.

The second step of our experimental study is to verify how the choice of the scaling matrix affects the numerical performance of the constrained dogleg approach.

We run CoDoSol using both spherical and elliptical trust regions. Our runs clearly indicated that the elliptical trust region yields a more robust and efficient approach than the spherical one. Then, here we report the results of our numerical experience adopting the elliptical trust region.

[1] Problems P2 and P3 appear in Table 3 with just 2 and 1 initial guesses, respectively, as for the missing initial guesses the three solvers under comparison did not converge to the same solution. This reduces the number of compared runs to 105.

**Table 3** `CoDoSol` with Coleman-Li scaling, IATR and STRSCNE: comparative numerical results for problems successfully solved by all codes

| Pb# | $n$ | $\nu$ | $\|F_0\|$ | `CoDoSol` | | IATR | | STRSCNE | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | *It* | *Fe* | *It* | *Fe* | *It* | *Fe* |
| 2 | 2 | 2 | 7(−1) | 5 | 6 | 5 | 6 | 5 | 6 |
| | | 3 | 2(−1) | 4 | 5 | 6 | 8 | 6 | 8 |
| 3 | 5 | 1 | 2(1) | 6 | 7 | 9 | 10 | 9 | 10 |
| 4 | 8 | 1 | 1(0) | 6 | 7 | 6 | 7 | 9 | 10 |
| | | 2.5 | 2(0) | 6 | 7 | 5 | 6 | 6 | 7 |
| | | 3 | 2(0) | 5 | 6 | 5 | 6 | 7 | 8 |
| 6 | 2 | 1 | 5(−1) | 3 | 4 | 3 | 4 | 3 | 4 |
| | | 2 | 1(1) | 5 | 6 | 4 | 5 | 6 | 7 |
| | | 3 | 1(1) | 7 | 8 | 8 | 9 | 8 | 9 |
| 7 | 67 | 1 | 2(6) | 20 | 23 | 24 | 31 | 24 | 31 |
| | | 2 | 6(0) | 9 | 11 | 24 | 34 | 26 | 37 |
| | | 3 | 8(4) | 15 | 17 | 17 | 20 | 18 | 20 |
| 8 | 451 | 1 | 6(5) | 18 | 19 | 18 | 19 | 18 | 19 |
| | | 2.5 | 1(5) | 17 | 18 | 17 | 18 | 17 | 18 |
| | | 3 | 6(5) | 20 | 22 | 19 | 20 | 19 | 20 |
| 10 | 5 | 0 | 4(0) | 66 | 86 | 76 | 95 | 74 | 92 |
| | | 1 | 3(3) | 12 | 13 | 12 | 13 | 12 | 13 |
| | | 2 | 3(6) | 16 | 17 | 16 | 17 | 16 | 17 |
| 16 | 12500 | 0 | 9(1) | 18 | 24 | 15 | 16 | 18 | 19 |
| | | 1 | 9(1) | 14 | 20 | 14 | 15 | 23 | 24 |
| | | 2 | 1(2) | 15 | 22 | 15 | 16 | 18 | 19 |
| 22 | 2 | 1 | 1(0) | 5 | 6 | 5 | 6 | 5 | 6 |
| | | 2 | 1(0) | 1 | 2 | 2 | 3 | 1 | 2 |
| | | 3 | 1(0) | 5 | 6 | 5 | 6 | 5 | 6 |
| 24 | 500 | 1 | 8(1) | 14 | 15 | 14 | 15 | 14 | 15 |
| | | 2 | 5(−6) | 2 | 3 | 2 | 3 | 2 | 3 |
| | | 3 | 5(1) | 14 | 15 | 14 | 15 | 14 | 15 |
| 25 | 1000 | 1 | 1(6) | 11 | 12 | 10 | 11 | 10 | 11 |
| | | 2 | 1(1) | 5 | 6 | 5 | 6 | 5 | 6 |
| | | 3 | 1(6) | 11 | 12 | 11 | 12 | 10 | 11 |
| 27 | 600 | 1 | 6(−1) | 9 | 11 | 9 | 11 | 9 | 11 |
| | | 2 | 6(−1) | 6 | 7 | 6 | 7 | 6 | 7 |
| | | 3 | 2(−1) | 7 | 8 | 7 | 8 | 7 | 8 |
| 29 | 2000 | 1 | 5(1) | 8 | 9 | 8 | 9 | 8 | 9 |
| | | 2 | 5(1) | 7 | 8 | 7 | 8 | 7 | 8 |
| | | 3 | 2(1) | 7 | 8 | 7 | 8 | 7 | 8 |
| 31 | 10000 | 1 | 3(1) | 17 | 19 | 17 | 19 | 17 | 19 |
| | | 2 | 1(0) | 19 | 21 | 19 | 21 | 19 | 21 |
| | | 3 | 2(4) | 20 | 22 | 20 | 22 | 20 | 22 |
| 32 | 5000 | 0 | 4(0) | 5 | 6 | 5 | 6 | 5 | 6 |
| | | 1 | 5(0) | 7 | 8 | 7 | 8 | 7 | 8 |
| | | 2 | 2(3) | 12 | 15 | 12 | 13 | 12 | 13 |
| 34 | 10000 | 0 | 7(−6) | 2 | 3 | 2 | 3 | 2 | 3 |
| | | 1 | 6(−1) | 3 | 4 | 3 | 4 | 3 | 4 |
| | | 2 | 2(−1) | 6 | 7 | 6 | 7 | 6 | 7 |

**Table 4** `CoDoSol` with Coleman-Li scaling, IATR and STRSCNE: comparative numerical results for problems where failures occurred

| Pb# | $n$ | $\nu$ | $\|F_0\|$ | `CoDoSol` | | IATR | | STRSCNE | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | *It* | *Fe* | *It* | *Fe* | *It* | *Fe* |
| 1 | 2 | 1 | 5(4) | 21 | 30 | 20 | 26 | 21 | 30 |
| | | 2 | 2(5) | 6 | 7 | 27 | 34 | 17 | 21 |
| | | 3 | 5(5) | * | | * | | * | |
| 5 | 2 | 1 | 3(−1) | * | | * | | * | |
| | | 2 | 4(0) | * | | * | | * | |
| | | 3 | 2(2) | 10 | 11 | 10 | 11 | 10 | 11 |
| 9 | 11 | 0 | 1(2) | 13 | 14 | 12 | 13 | 16 | 17 |
| | | 1 | 4(4) | 20 | 21 | 18 | 20 | 26 | 28 |
| | | 2 | 4(7) | 26 | 27 | * | | 32 | 33 |
| 12 | 10 | 0 | 2(14) | 20 | 21 | 20 | 21 | 20 | 21 |
| | | 1 | 2(17) | 27 | 28 | 25 | 26 | 25 | 26 |
| | | 2 | 2(20) | 31 | 32 | 34 | 36 | * | |
| 11 | 10 | 0 | 8(13) | 26 | 27 | 26 | 27 | 26 | 27 |
| | | 1 | 8(16) | 32 | 33 | 37 | 39 | 32 | 33 |
| | | 2 | 8(19) | 38 | 40 | 36 | 37 | * | |
| 13 | 8 | 0 | 2(1) | * | | * | | 10 | 12 |
| | | 1 | 1(3) | 15 | 18 | * | | 14 | 15 |
| | | 2 | 1(5) | 18 | 19 | * | | 19 | 20 |
| 14 | 16 | 0 | 1(2) | 27 | 38 | 29 | 39 | 30 | 40 |
| | | 1 | 3(5) | 23 | 26 | 14 | 15 | 15 | 16 |
| | | 2 | 4(45) | 113 | 116 | * | | * | |
| 15 | 9 | 0 | 6(0) | 7 | 9 | 8 | 10 | 9 | 10 |
| | | 1 | 3(2) | 9 | 11 | 9 | 10 | 10 | 11 |
| | | 2 | 3(4) | * | | 18 | 22 | * | |
| 17 | 2904 | 0 | 1(2) | 61 | 65 | 44 | 45 | 55 | 56 |
| | | 1 | 1(2) | 38 | 42 | 121 | 124 | 42 | 43 |
| | | 2 | 2(2) | * | | * | | * | |
| 18 | 2048 | 0 | 5(1) | 12 | 16 | * | | 25 | 26 |
| | | 1 | 5(1) | 15 | 20 | * | | 24 | 25 |
| | | 2 | 5(1) | 13 | 16 | * | | 29 | 30 |

For each scaling, we tested the algorithm using as initial trust region size both $\Delta_0 = 1$ and $\Delta_0 = D_0^{-1}\nabla f_0$. We observed that the choice of the initial trust region radius is crucial for the performance of `CoDoSol`, and the optimal choice depends on the adopted scaling matrix. In particular, numerical evidence shows that $D = D^{CL}(x)$ and $D = D^{KK}(x)$ should be used in conjunction with the choice $\Delta_0 = 1$, whereas when $D = D^{HMZ}(x)$ is employed is fundamental to adopt the choice $\Delta_0 = D_0^{-1}\nabla_0 f$. This is not surprising as this scaling matrix depends on the size of the gradient, and $\Delta_0 = 1$ yields in practice a too small initial trust region size. Hence, the performance profiles depicted in Fig. 4 are obtained with these choices

**Table 5** `CoDoSol` with Coleman-Li scaling, IATR and STRSCNE: comparative numerical results for problems where failures occurred

| Pb# | $n$ | $\nu$ | $\|F_0\|$ | `CoDoSol` | | IATR | | STRSCNE | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | *It* | *Fe* | *It* | *Fe* | *It* | *Fe* |
| 19 | 7500 | 0 | 4(−2) | * | | 49 | 58 | 11 | 12 |
| | | 1 | 4(−1) | * | | 9 | 10 | 14 | 15 |
| | | 2 | 3(0) | * | | * | | 16 | 17 |
| 20 | 2 | 1 | 2(0) | 7 | 9 | * | | * | |
| | | 2 | 2(0) | 4 | 5 | 4 | 5 | 4 | 5 |
| | | 3 | 2(0) | 5 | 7 | 4 | 5 | 4 | 5 |
| 21 | 2 | 1 | 1(2) | 10 | 11 | 10 | 11 | 10 | 11 |
| | | 2 | 3(0) | 4 | 5 | 4 | 5 | 4 | 5 |
| | | 3 | 1(2) | 8 | 9 | * | | * | |
| 23 | 2 | 1 | 3(3) | 13 | 14 | * | | * | |
| | | 2 | 1(0) | 1 | 2 | 2 | 3 | 1 | 2 |
| | | 3 | 5(21) | 55 | 56 | * | | 55 | 56 |
| 26 | 1000 | 1 | 1(7) | * | | * | | * | |
| | | 2 | 2(2) | * | | * | | * | |
| | | 3 | 1(7) | 23 | 26 | 23 | 26 | 24 | 27 |
| 28 | 5000 | 1 | 7(4) | 18 | 28 | 18 | 19 | * | |
| | | 2 | 4(4) | 17 | 26 | 18 | 19 | 23 | 24 |
| | | 3 | 1(4) | 16 | 25 | 16 | 17 | 17 | 18 |
| 30 | 400 | 1 | 6(0) | 7 | 8 | 7 | 8 | 7 | 8 |
| | | 2 | 4(1) | 7 | 8 | 7 | 8 | 7 | 8 |
| | | 3 | 8(3) | * | | * | | * | |
| 33 | 5000 | 0 | 8(1) | * | | * | | * | |
| | | 1 | 8(1) | * | | * | | * | |
| | | 2 | 4(−1) | 4 | 5 | 4 | 5 | 4 | 5 |
| 35 | 10000 | 1 | 1(2) | 232 | 233 | * | | * | |
| | | 2 | 1(1) | 6 | 7 | 6 | 7 | 6 | 7 |
| | | 3 | 8(1) | 10 | 11 | 12 | 13 | 12 | 13 |
| 36 | 1000 | 0 | 1(2) | * | | * | | * | |
| | | 1 | 1(2) | * | | * | | * | |
| | | 2 | 3(0) | 4 | 5 | 4 | 5 | 4 | 5 |

($\Delta_0 = 1$ in the cases $D = D^{CL}(x)$, $D = D^{KK}(x)$ and $\Delta_0 = D_0^{-1}\nabla f_0$ in the case $D = D^{HMZ}(x)$).

As it can be clearly seen by the performance profiles, the pioneer Coleman and Li scaling seems to be preferable to the other two scaling matrices, despite its weaker theoretical properties in terms of continuity. The Hager-Maier-Zhang scaling matrix shares with the Coleman-Li scaling the property of giving the best solver in 55% of the tests, in terms of number of iterations. In terms of $F$-evaluations, its behavior is in between the other two scalings, despite it provides the desirable property of

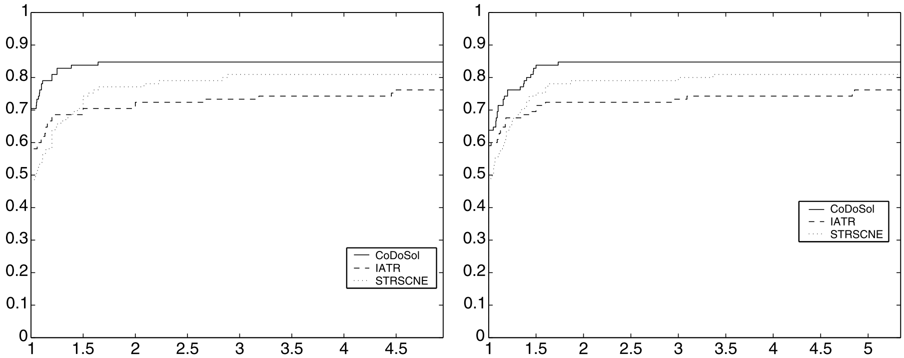


**Fig. 2** `CoDoSol` with Coleman-Li scaling, IATR and STRSCNE: Performance profiles in terms of iterations (*left*) and $F$-evaluations (*right*)

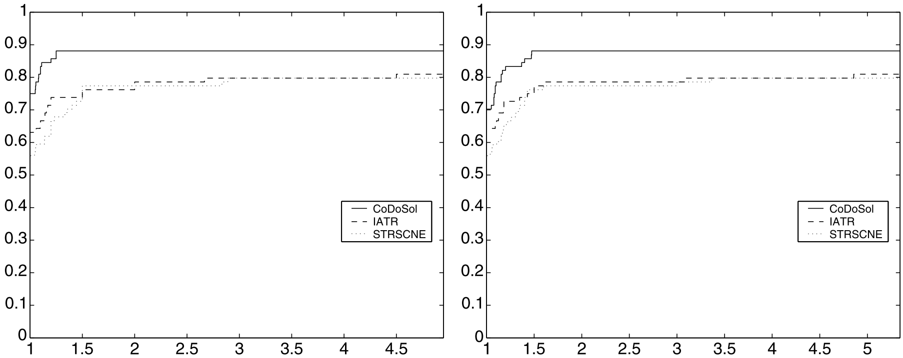


**Fig. 3** `CoDoSol` with Coleman-Li scaling, IATR and STRSCNE: Performance profiles in terms of iterations (*left*) and $F$-evaluations (*right*). Without NCPs

having a well centered step along the scaled gradient; in fact a step of length one can always be taken along the scaled gradient without crashing in to the bounds. In terms of robustness, the Hager-Maier-Zhang scaling is just a bit less robust than the Coleman-Li scaling. As a whole, the Kanzow-Klug scaling appears to give both the less efficient and the less robust solver.

## 5 Conclusions

We presented an affine scaling trust region algorithm for medium scale bound constrained systems of nonlinear equations. The algorithm combines Newton method and trust region procedures where the merit function used is the norm of the nonlinear residual. The trust region and the corresponding scaled gradient are defined by suitable diagonal scaling avoiding the problem of running directly into a bound. The trust region problem is approximately solved by a constrained dogleg method where the scaled Cauchy step is combined with an interior point Newton step, i.e. the projection of the Newton step onto the interior of $\Omega$. The convergence analysis

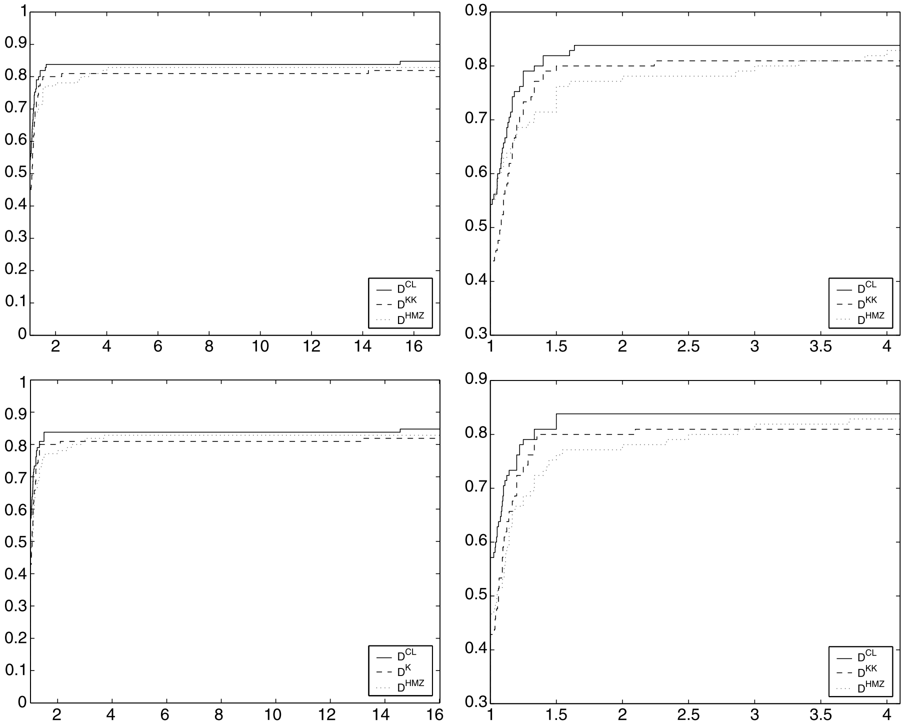


**Fig. 4** CoDoSol with different scalings: Performance profiles in terms of iterations (*top*) and $F$-evaluations (*bottom*). *Right*: details of the left-sided plots

does not require to specify the scaling matrix used to handle the bounds. Focusing on diagonal scalings from the numerical optimization literature, we show that a number of choices are possible in our context, too. It follows that different constrained dogleg algorithms can be defined. To test the numerical features of the proposed approach, we implemented the constrained dogleg method in the freely accessible `Matlab` solver `CoDoSol`. We give a brief account of this solver and discuss the major issues addressed in performing our numerical experiments.

From the obtained results the new proposal seems to be a useful tool to solve bound-constrained systems of nonlinear equations. We are well aware that the actual performance of an algorithm may strongly depend on the set of test problems used to perform numerical experiments. Then, we consider the conclusions of this study indicative to suggest that the flexibility of the adopted constrained dogleg method in choosing the scaling matrices is appropriate for application dependent purposes.

## References


1. Bellavia, S., Macconi, M., Morini, B.: An affine scaling trust-region approach to bound-constrained nonlinear systems. Appl. Numer. Math. **44**, 257–280 (2003)
2. Bellavia, S., Macconi, M., Morini, B.: STRSCNE: A scaled trust-region solver for constrained nonlinear equations. Comput. Optim. Appl. **28**, 31–50 (2004)

3. Bellavia, S., Macconi, M., Pieraccini, S.: On affine scaling inexact dogleg methods for bound-constrained nonlinear systems. Tech. Rep. 5/2009 (2009)
4. Bellavia, S., Morini, B.: An interior global method for nonlinear systems with simple bounds. Optim. Methods Softw. **20**, 1–22 (2005)
5. Bellavia, S., Morini, B.: Subspace trust-region methods for large bound constrained nonlinear equations. SIAM J. Numer. Anal. **44**, 1535–1555 (2006)
6. Coleman, T.F., Li, Y.: An interior trust-region approach for nonlinear minimization subject to bounds. SIAM J. Optim. **6**, 418–445 (1996)
7. Davis, T.A.: Algorithm 832: UMFPACK, an unsymmetric-pattern multifrontal method. ACM Trans. Math. Softw. **30**, 196–199 (2004)
8. Dirkse, S.P., Ferris, M.C.: MCPLIB: A collection of nonlinear mixed complementary problems. Optim. Methods Softw. **5**, 319–345 (1995)
9. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. **91**, 201–213 (2002)
10. Floudas, C.A., et al.: Handbook of Test Problems in Local and Global Optimization. Nonconvex Optimization and Its Applications, vol. 33. Kluwer Academic, Dordrecht (1999)
11. Francisco, J.B., Krejic, N., Martinez, J.M.: An interior-point method for solving box-constrained underdetermined nonlinear systems. J. Comput. Appl. Math. **177**, 67–88 (2005)
12. Hager, W.W., Mair, B.A., Zhang, H.: An affine-scaling interior-point CBB method for box-constrained optimization. Math. Program., Ser. A **119**, 1–32 (2009)
13. Heinkenschloss, M., Ulbrich, M., Ulbrich, S.: Superlinear and quadratic convergence of affine-scaling interior-point Newton methods for problems with simple bounds without strict complementarity assumptions. Math. Program. **86**, 615–635 (1999)
14. Hock, W., Schittkowski, K.: Test Examples for Nonlinear Programming Codes. Lecture Notes in Economics and Mathematical Systems, vol. 187. Springer, Berlin (1981)
15. Kanzow, C., Klug, A.: On affine-scaling interior-point Newton methods for nonlinear minimization with bound constraints. Comput. Optim. Appl. **35**, 177–197 (2006)
16. Kanzow, C., Klug, A.: An interior-point affine-scaling trust-region method for semismooth equations with box constraints. Comput. Optim. Appl. **37**, 329–353 (2007)
17. Kelley, C.T.: Iterative Methods for Linear and Nonlinear Equations. Frontiers in Applied Mathematics. SIAM, Philadelphia (1995)
18. Kelley, C.T., Northrup, J.I.: A pointwise Quasi-Newton method for integral equations. SIAM J. Numer. Anal. **25**, 1138–1155 (1988)
19. Kozakevich, D.N., Martinez, J.M., Santos, S.A.: Solving nonlinear systems of equations with simple bounds. J. Comput. Appl. Math. **16**, 215–235 (1997)
20. Luksan, L., Vlcek, J.: Sparse and partially separable test problems for unconstrained and equality constrained optimization. Technical Report N. 767, Institute of Computer Science, Academy of Sciences of the Czech Republic (1999)
21. Macconi, M., Morini, B., Porcelli, M.: A Gauss-Newton method for solving bound-constrained underdetermined nonlinear systems. Optim. Methods Softw. **24**, 219–235 (2009)
22. Macconi, M., Morini, B., Porcelli, M.: Trust-region quadratic methods for nonlinear systems of mixed equalities and inequalities. Appl. Numer. Math. **59**, 859–876 (2009)
23. Meintjes, K., Morgan, A.P.: A methodology for solving chemical equilibrium systems, Applied. Math. Comput. **22**, 333–361 (1987)
24. Meintjes, K., Morgan, A.P.: Chemical equilibrium systems as numerical tests problems. ACM Trans. Math. Softw. **16**, 143–151 (1990)
25. Moré, J.J.: A collection of nonlinear model problems. Lect. Appl. Math. **26**, 723–762 (1990)
26. Moré, J.J., Garbow, B., Hillstrom, K.: Testing unconstrained optimization software. ACM Trans. Math. Softw. **7**, 136–140 (1981)
27. Pawlowski, R.P., Simonis, J.P., Walker, H.F., Shadid, J.N.: Inexact Newton dogleg methods. SIAM J. Numer. Anal. **46**, 2112–2132 (2008)
28. Tsoulos, I.G., Stavrakoudis, A.: On locating all roots of systems of nonlinear equations inside bounded domain using global optimization methods. Nonlinear Anal., Real World Appl. **11**, 2465–2471 (2010)
29. Wachter, A., Biegler, L.T.: Failure of global convergence for a class of interior point methods for nonlinear programming. Math. Program., Ser. A **88**, 565–574 (2000)
30. Wang, T., Monteiro, R.D.C., Pang, J.-S.: An interior point potential reduction method for constrained equations. Math. Program. **74**, 159–195 (1996)
31. Zhu, D.: An affine scaling trust-region algorithm with interior backtracking technique for solving bound-constrained nonlinear systems. J. Comput. Appl. Math. **184**, 343–361 (2005)