Constrained Dogleg Methods for nonlinear systems with simple bounds

by S. Bellavia[1], M. Macconi[1], S. Pieraccini[2]

Dipartimento di Energetica "Sergio Stecco",
Università di Firenze, Pubblicazione n. 6/2009.

[1] Dipartimento di Energetica "S. Stecco",
Università di Firenze,
via C. Lombroso 6/17, 50134 Firenze, Italia,
Email: stefania.bellavia@unifi.it, benedetta.morini@unifi.it.

[2] Dipartimento di Matematica, Politecnico di Torino, corso Duca degli Abruzzi, 24, 10129 Torino, Italia, e-mail: pieraccini@calvino.polito.it

Constrained Dogleg Methods for nonlinear systems with simple bounds *

Stefania Bellavia†, Maria Macconi†, Sandra Pieraccini‡

December 24, 2009

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.

Key words: bound-constrained equations, diagonal scalings, trust region methods, dogleg methods, Newton methods, global convergence

1 Introduction

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

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

where $F : X \mapsto \mathbb{R}^n$ is a continuosly differentiable mapping with Jacobian denoted by F', $X \subseteq \mathbb{R}^n$ is an open set containing the n-dimensional box $\Omega = \{x \in \mathbb{R}^n \mid l \leq x \leq u\}$. Here, the inequalities are meant componentwise 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 Ω has nonempty interior.

*Work supported by the MIUR, Rome, Italy, through grants PRIN07.

†Dipartimento di Energetica "S. Stecco", Università di Firenze, via C. Lombroso 6/17, 50134 Firenze, Italia, e-mail: `stefania.bellavia@unifi.it`, `maria.macconi@unifi.it`

‡Dipartimento di Matematica, Politecnico di Torino, corso Duca degli Abruzzi, 24, 10129 Torino, Italia, e-mail: `pieraccini@calvino.polito.it`

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 behaviour. 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, 28]. In particular, the methods given in [1],[2] and [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 [5, 3] as well as for developing extensions to rectangular nonlinear systems [11, 19, 18].

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 Ω 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 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 mantained. In particular, all the iterates are required to be strictly feasible points and global, locally fast convergence must be ensured.

To hit these targets we decided to incorporate the Newton's method into a dogleg scheme very similar to that discussed by the authors in [3] for large scale problems. The inexact dogleg procedure given in [3] extends to constrained systems the method proposed in [24] for large scale unconstrained systems of nonlinear equations. The aim of this latter work is to overcome the problematics related to the implementation of a dogleg strategy when a Krylov method is used to solve the Newton equations. In fact, in this situation, the exact Newton-step is not available and the classical dogleg procedure cannot be employed without suitable modifications. In [3] the authors show how the presence of constraints is reflected in this scheme and propose an inexact dogleg method tailored for large scale bound constrained problems 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 pioneer Coleman-Li scaling matrix has been used to test the numerical features of this method and perform a comparison with the Subspace Strategy proposed in [5] for large scale bound constrained systems, too.

In this paper, we focus on medium scale bound-constrained systems. Due to the dimension of the problems here considered, it is realistic to assume that a direct method is used to solve the Newton equation, getting the exact Newton step at each iteration. Nevertheless, in the same trust region framework of [3], 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 Ω. Then, 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.

We remark that our scheme allows a rather general class of diagonal scalings and that the convergence analysis can be performed without specifying the scaling matrix used to handle the bounds. We like to make clear that 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 named the iterative procedure given here Constrained Dogleg (`CoDo`) method. Since different `CoDo` algorithms distinguish themselves by the choice of the scaling matrix, we focus on several diagonal scalings proposed by different authors in the numerical optimization literature and we show that a number of choices are possible in our context, too.

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 behaviour 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 nolinear systems.

Since different matrices may be used in `CoDoSol`, it is quite important to state if the behaviour 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 Section 2 we present the constrained dogleg method and describe its convergence properties. In Section 3 we consider several scaling matrices from the numerical optimization literature and show

how they match our requirements. In Section 4 we give a brief account of the `Matlab` solver `CoDoSol` and report on our computational experience. Some conclusions are presented in Section 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 ρ 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:

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

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

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

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 are 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 λ_k be the stepsize along $\hat{g}_k$ to the boundary, i.e.

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

and let us consider scaling matrices satisfying the following properties:

Assumption A1:

(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 λ_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 exists $\bar{\rho} > 0$ and $\chi_{\bar{x}}$ such that $B_\rho(\bar{x}) \subset int(\Omega)$ and $\|D(x)^{-1}\| \leq \chi_{\bar{x}}$ 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

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

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

$$m_k(p) = \|F_k + F_k' p\| \tag{8}$$

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 Ω followed by a step toward the interior of the feasible set. In other words, we consider the step $\bar{p}_k^N$ given by:

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

where $P(x)$ is the projection of x onto Ω, i.e. $P(x)_i = \max\{l_i, \min\{x_i, u_i\}\}, 1 \leq i \leq n$. Clearly, the point $x_k + \bar{p}_k^N$ is strictly feasible (see Figure 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 τ_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 λ_k is the stepsize along $\hat{g}_k$ to the boundaries, i.e. (6), and τ'_k is computed in the following way

$$\tau'_k = \underset{\|\tau G_k \hat{g}_k\| \leq \Delta_k}{\operatorname{argmin}} \; m_k(\tau \hat{g}_k) = \min \left\{ -\frac{F_k^T F'_k \hat{g}_k}{\|F'_k \hat{g}_k\|^2}, \frac{\Delta_k}{\|G_k \hat{g}_k\|} \right\}. \tag{14}$$

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

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

and we look for the value of γ 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 ϕ 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:

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

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 Ω 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

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

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:

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

and take

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

whereas if $\gamma < 0$ we set:

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

and

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

To summarize, the choice of γ 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 γ 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 γ_+ by (17)
    compute $\bar{\gamma}_+$ by (18)
    set $\gamma = \min\{\hat{\gamma}, \gamma_+, \theta\bar{\gamma}_+\}$
  Else
    compute γ_- 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 Figure 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 Ω. 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 step. In other words, the step satisfies the condition

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

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

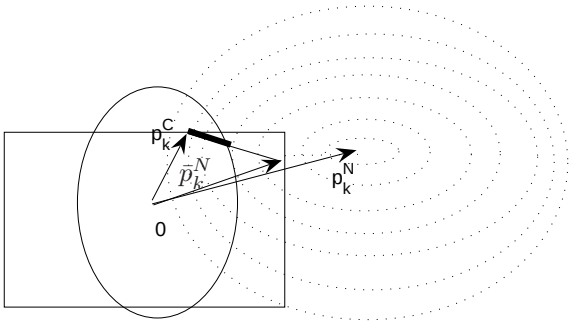

Figure 1: Illustrative constrained path $p(\gamma)$ in $\mathbb{R}^2$. The segment in bold is the path along with the model is minimized.

the reduction predicted by the local linear model. Then, we test if the following sufficient improvement condition

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

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, Δ_k is shrunk and a new trial step is computed.

The following is our general constrained dogleg method where the positive constant Δ_{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 2.1 *Let (i-iii) in Assumption A1 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 2.2 *Let (i-iv) in Assumption A1 be satisfied,* $\|F'\|$ *be bounded above on*

$$L = \cup_{k=0}^{\infty} \{x \in X \ : \ \|x - x_k\| \leq r\}, \qquad r > 0,$$

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* α_k *in (10) satisfies* $\alpha_k \to 1$, *as* $k \to \infty$, *and*

- *either* $G_k = I$, $k \geq 0$, *or*
- $G_k = D_k^{-1/2}$, $k \geq 0$, *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*

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

the convergence rate is quadratic.

Summarizing, suitable choices of the scalar α_k in the computation of the projected Newton step ensure quadratic convergence of `CoDo` method whenever spherical trust region are used (i.e. $G_k = I$, $k \geq 0$), independently of the

position of the solution. On the other hand, when elliptical trust regions are employed, (i.e. $G_k = D_k^{-1/2}$, $k \geq 0$), 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 Ω.

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 Assumptions A1. 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:

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

- $D^{HUU}(x)$ given by Heinkenschloss et al. in [13]. The diagonal entries $D^{HUU}(x)$ 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:

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

  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

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

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 λ_k is the classical Barzilai-Borwein parameter. The method proposed in [12] moves 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 A1.

(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 A1.

(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

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

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 $d_i^{KK}(x) \leq x_i - l_i$. Then, if $(\nabla f_k)_i > 0$, from (27) and (22) we have

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

Similarly, if $(\nabla f_k)_i < 0$ we get

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

This implies

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

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 $d_i^{CL}(x) \geq \min\{x_i - l_i, u_i - x_i\} \geq \rho/2$, it is easy to see that $\|(D^{CL}(x))^{-1}\| \leq 2/\rho$ whenever $x \in B_{\rho/2}(\bar{x})$ with $\bar{x} \in int(\Omega)$ and ρ 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 $X_i(x) \geq \min\{x_i - l_i, u_i - x_i\} \geq \rho/2$. 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 Assumptions A1 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 analize 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 Ω 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, $\Omega = \{x \in \mathbb{R}^n : x \geq 0\}$ 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 ith axis. In fact, we have

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

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

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

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 ith component of the step is not reduced by a factor x_i, but has length x_i. In other words, since the ith 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 ith 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 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 $\nabla f(x)_i(x)$ 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 behaviour 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 problem 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 Jacobians 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.

4.2 The numerical experiments

All numerical experiments have been performed on a 3.4 Ghz Intel Xeon (TM) with 1GB of RAM using the `Matlab 7.6` version of the code `CoDoSol` and machine precision $\epsilon_m \approx 2.10^{-16}$.

The experiments were carried out on a set of 26 problems with dimension between $n = 2$ and $n = 451$. All problems arise from mathematical models related to real-life phenomena. This set of problems provides us with various type of representative constrained systems and it is specified in Table 1. To form this set we used the nonlinear constrained systems given in Chapter 14 of [10] (from Pb1 to Pb8), two equilibrium problems modeled by parameter dependent nonlinear equations (Pb9 and Pb10), the Chandrasekhar H-equation with the challenging value $c = 0.99$ (Pb11). This latter problem was solved on a 100-points mesh with the bounds $l = 0$, $u = 5$. Further, we used four chemical equilibrium systems given in [21, 20, 26] (from Pb12 to Pb15) and four nonlinear complementarity problems given in [14, 8] (from Pb16 to Pb20). The nonlinear complementarity problems were reformulated as systems of smooth box-constrained nonlinear equations (see [27]). Finally, the last six problems (from Pb21 to Pb26) can be found in [25] and [22].

The selected tests include systems with solutions both within the feasible region Ω and on the boundary of Ω, systems with only lower (upper) bounds and systems with variable components bounded from above and below.

We performed our experiments starting from good and poor initial guesses. In particular, 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, $x_0 = 10^\nu(1, \ldots, 1)^T$, $\nu = 0, 1, 2$, for the other problems. We remark that the vector x_0 obtained with $\nu = 3$ is solution of problem Pb5. Further, the Jacobian matrices of Pb6 and Pb10 are singular at the starting point obtained with $\nu = 2$. These critical values of ν have been replaced by $\nu = 2.5$. Moreover, the Jacobians $J(x)$ of problems Pb21 and Pb22 are singular at points such that $(x)_1 = (x)_2$. So, in these cases, we modified the first component of x_0 by putting $(x_0)_1 = 0.5$. Finally, the starting points $x_0 = l + 0.25\nu(u - l)$, $\nu = 1.5, 2.5, 3.5$ have been used to solve the problem Pb26 because the standard values $\nu = 1, 2, 3$ lead to solutions of the problem.

Summarizing, 26 problems occurring in applications have been choosen and solved starting from three different initial points for a total of 78 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.

Further, 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

$$\|F_k\| \leq 10^{-6}$$

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 conjuction 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]:

$$\begin{cases} \alpha_0 = \max(1.d-2, \|\nabla f_0\|) \\ \alpha_k = \max(1.d-2, (p_k^T(\nabla f_k - \nabla f_{k-1}))/(p_k^T p_k)), \quad p_k = x_k - x_{k-1}. \end{cases}$$

Pb#	Name and Source	n
1	Himmelblau function [10, 14.1.1]	2
2	Equilibrium Combustion [10, 14.1.2]	5
3	Bullard-Biegler system [10, 14.1.3]	2
4	Ferraris-Tronconi system [10, 14.1.4]	2
5	Brown's almost linear system [10, 14.1.5]	5
6	Robot kinematics problem [10, 14.1.6]	8
7	Series of CSTRs, $R = .935$ [10, 14.1.8]	2
8	Series of CSTRs, $R = .995$ [10, 14.1.8]	2
9	Chemical reaction problem [17, Problem 5]	67
10	A Mildly-Nonlinear BVP [17, Problem 7]	451
11	H-equation, $c = 0.99$ [23, Problem 4]	100
12	Chemical equilibrium system [21, system 1]	11
13	Chemical equilibrium system [21, system 2]	5
14	Combustion system (Lean case) [20]	10
15	Combustion system (Rich case) [20]	10
16	Kojima-Shindo problem [8]	8
17	Josephy problem [8]	8
18	Mathiesen problem [8]	8
19	Problem HS34 [14]	16
20	Problem Wachter and Biegler [26]	9
21	Effati-Grosan 1, $a = 2$ [25]	2
22	Effati-Grosan 1, $a = 100$ [25]	2
23	Effati-Grosan 2, $a = 2$ [25]	2
24	Effati-Grosan 2, $a = 100$ [25]	2
25	Steering problem [22]	3
26	Merlet problem [25]	2

Table 1: Test Problems

4.3 The experimental study

We first investigated the robusteness 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 decided to measure 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 the Tables 2 and 3 where, for each problem, we give the value of the parameter ν used to compute the initial guess x_0 and the value $\|F_0\|$ of $\|F(x_0)\|$. 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 $m.10^c$.

We now comment in more details our results.

The `CoDoSol` successfully solved 18 test problems starting from all the three initial guesses, 5 test problems starting from two of the three x_0 used and 3 test problems starting from only one initial guess. So, the new method succeeded 67 times on a total of 78 runs.

The STRSCNE solver performed equally well. It succeeded 67 times: it solved 16 problems starting from all the three x_0 used, 9 problems starting from two initial guesses and only one problem with only one x_0.

The IATR method was a bit less robust than the other two. On the chosen set of tests it solved 16 problems starting from the three x_0, 7 problems starting from two x_0 and 2 problems starting from one x_0. Further, it was not able to solve Pb17 since failure occurred for all the x_0 used. However, only IATR solved Pb15, Pb18 and Pb20 when $\nu = 2$ was used to compute x_0. Summarizing, it succeeded in solving 64 tests.

Tables 2 and 3 provide us with other informations on the numerical performance of the constrained dogleg method proposed here. It does not seem to be very expensive since all the successful runs were performed with a small number of iterations and with $R \leq 5$ (except for one run where we count $R = 10$ and two runs where it results $R = 8$). The other two solvers required more computational work. We count $R \geq 5$ six times (with a maximum of $R = 18$) for IATR and five times (with a maximum of $R = 17$) for STRSCNE.

To make in more significant evidence above observations we compare the three algorithms by adopting the performance profile approach (see [9]). In Figure 2 the computational effort is measured in terms of the number It of the iterations performed while in Figure 3 the computational effort is measured in terms of the number Fe of the F-evaluations. Again, the profiles indicate that the `CoDoSol` outperforms IATR and STRSCNE as it is more efficient in solving about the 67% 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 drops to 7.5% if we exclude Problems Pb16-Pb20, i.e. problems obtained reformulating nonlinear complementarity problems. This could be expected as our method does not exploit at all the special structure of nonlinear complementarity problems.

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

			CoDoSoll		IATR		STRSCNE	
Pb#	ν	$\|F_0\|$	It	Fe	It	Fe	It	Fe
1	1	7(1)	5	6	5	6	5	6
	2	3(1)	4	5	4	5	4	5
	3	2(1)	4	5	4	5	4	5
2	1	4(4)	12	13	12	13	12	13
	2	3(5)	14	15	14	15	14	15
	3	1(6)	16	17	16	17	16	17
4	1	3(-1)	6	10	5	6	5	6
	2	7(-1)	5	6	5	6	5	6
	3	2(-1)	4	5	6	8	6	8
5	1	2(1)	6	7	9	10	9	10
	2	1(1)	6	7	7	8	7	8
	2.5	6(0)	4	5	4	5	4	5
6	1	1(0)	6	7	6	7	9	10
	2.5	2(0)	7	8	6	7	6	7
	3	2(0)	5	6	5	6	7	8
8	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
9	1	2(6)	22	25	24	31	24	31
	2	6(0)	9	11	17	23	23	32
	3	8(4)	15	17	17	20	18	20
10	1	6(5)	19	20	18	19	19	20
	2.5	1(5)	17	18	17	18	17	18
	3	6(5)	20	22	19	20	19	20
13	0	4(0)	65	84	76	95	74	92
	1	3(3)	12	13	12	13	12	13
	2	3(6)	16	17	16	17	16	17
14	0	8(13)	26	27	26	27	26	27
	1	8(16)	32	33	37	39	32	33
	2	8(19)	36	37	38	40	39	41
25	1	1(-3)	4	6	4	6	4	6
	2	1(-2)	7	8	7	8	7	8
	3	5(-2)	10	14	9	12	9	12
26	1.5	2(0)	4	6	4	6	4	6
	2.5	2(0)	4	5	4	5	4	5
	3.5	2(0)	3	5	3	4	3	4

Table 2: `CoDoSol` with Coleman-Li scaling, IATR and STRSCNE: comparative numerical results for problems successfully solved by all codes

			CoDoSol		IATR		STRSCNE	
Pb#	ν	$\|F_0\|$	It	Fe	It	Fe	It	Fe
3	1	5(4)	21	30	20	26	21	30
	2	2(5)	6	7	27	34	17	21
	3	5(5)	*		*		*	
7	1	3(-1)	*		*		*	
	2	4(0)	*		*		*	
	3	2(2)	10	11	10	11	10	11
11	1	3(0)	6	7	6	7	6	7
	2	2(1)	5	6	5	6	5	6
	3	1(3)	*		*		*	
12	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
15	0	2(14)	20	21	20	21	20	21
	1	2(17)	25	26	25	26	25	26
	2	2(20)	*		34	36	*	
16	0	2(1)	9	11	7	9	10	12
	1	1(3)	*		9	10	11	12
	2	1(5)	*		13	14	22	23
17	0	2(1)	*		*		10	12
	1	1(3)	15	18	*		14	15
	2	1(5)	18	19	*		19	20
18	0	5(0)	5	6	5	6	8	9
	1	2(2)	*		*		17	18
	2	2(4)	*		11	12	*	
19	0	1(2)	27	38	29	39	31	41
	1	3(5)	23	26	14	15	15	16
	2	4(45)	113	116	*		*	
20	0	6(0)	7	9	8	10	9	10
	1	3(2)	9	11	9	10	10	11
	2	3(4)	*		18	21	*	
21	1	2(0)	7	9	*		*	
	2	2(0)	4	5	4	5	4	5
	3	2(0)	5	7	4	5	4	5
22	1	1(2)	10	11	10	11	10	11
	2	3(0)	4	5	4	5	4	5
	3	1(2)	8	9	*		*	
23	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	1	3(3)	15	18	*		*	
	2	1(0)	1	2	2	3	1	2
	3	5(21)	53	54	*		55	56

Table 3: CoDoSol with Coleman-Li scaling, IATR and STRSCNE: comparative numerical results for problems where failures occurrred

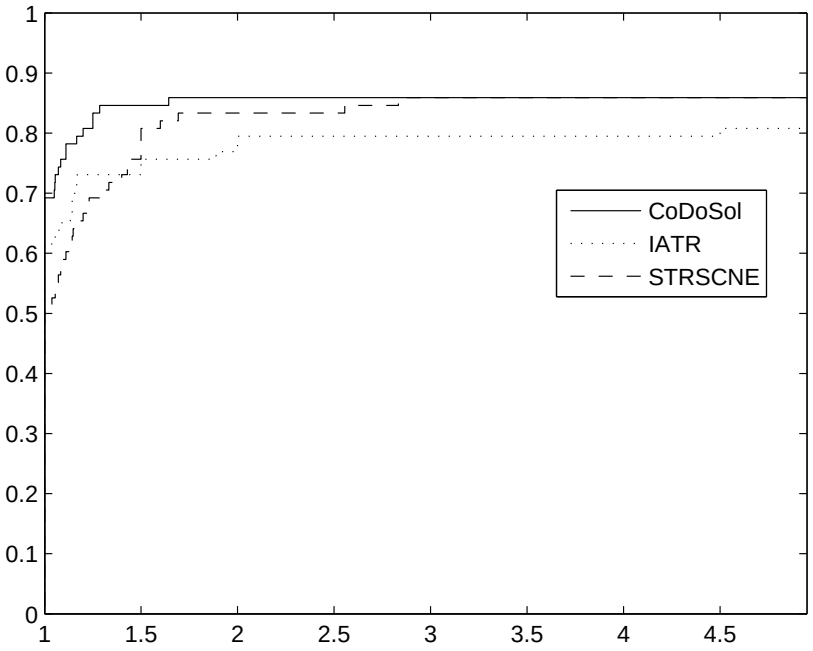

Figure 2: `CoDoSol` with Coleman-Li scaling, IATR and STRSCNE: Performance profiles in terms of Nonlinear Iterations

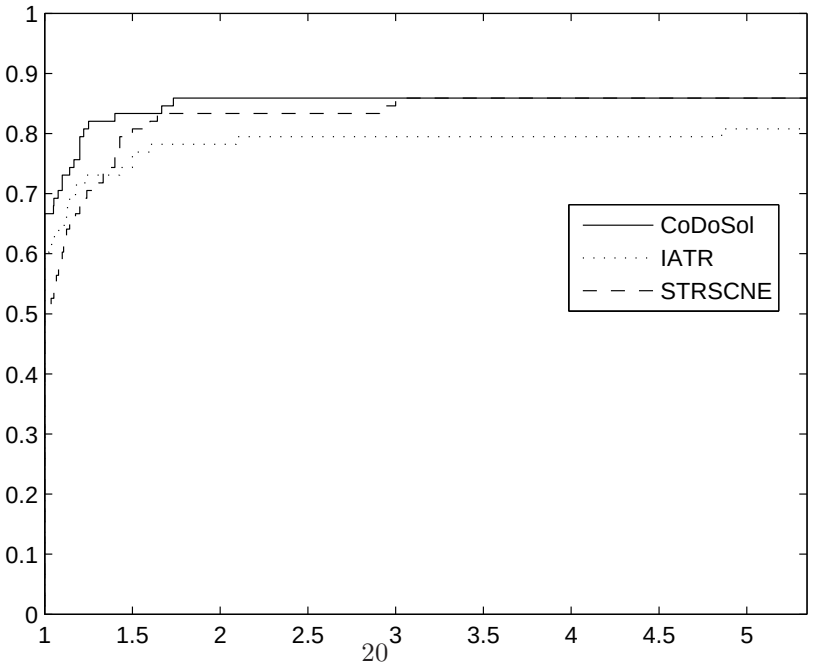

Figure 3: `CoDoSol` with Coleman-Li scaling, IATR and STRSCNE: Performance profiles in terms of F-evaluations

approach.

For each scaling, we tested the algorithm on our set of 78 tests using as initial trust region size both $\Delta_0 = 1$ and $\Delta_0 = D_0^{-1}\nabla f_0$. 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.

Remarkably, six cases could not be solved by any choice of the scaling with both the fixed Δ_0. These cases are problems Pb3, Pb11 started from x_0 with $\nu = 3$ and problems Pb7, Pb18 started from x_0 with $\nu = 1, 2$.

Further, only the choice $D^{HMZ}(x)$ allowed to solve Pb16 when $\nu = 1$ was used to compute x_0, Pb17 when $\nu = 0$ was used to compute x_0 and Pb20 when $\nu = 2$ was used to compute x_0. Only the choice $D = D^{CL}(x)$ allowed to solve Pb24 when $\nu = 3$ was used to compute x_0.

Moreover, 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 fact, if $\Delta_0 = 1$ is used, we count 10 failures with $D = D^{CL}(x)$, 22 failures with $D = D^{HMZ}(x)$ and 13 failures with $D = D^{KK}(x)$ while, if $\Delta_0 = D_0^{-1}\nabla_0 f$, we count 17 failures with the choice $D = D^{CL}(x)$, 13 failures with $D = D^{HMZ}(x)$ and 16 failures with $D = D^{KK}(x)$. So, it seems that $D = D^{CL}(x)$ and $D = D^{KK}(x)$ should be used in conjunction with the choice $\Delta_0 = 1$, while when $D = D^{HMZ}(x)$ is employed is fundamental to adopt the choice $\Delta_0 = D_0^{-1}\nabla_0 f$. This is not surpring 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.

We give the iteration and F-evaluation performance profiles in Figure 4 and Figure 5 respectively. To obtain these pictures, we consider the more efficient tested variants, i.e. we consider the results obtained by using $\Delta_0 = 1$ 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. On the other hand, the Hager-Maier-Zhang scaling seems to be clearly the less efficient even if this latter scaling matrix provides the desirable property to have a well centered step along the scaled gradient, as it is ensured that a step of length one can always be taken along the scaled gradient without crashing in to the bounds.

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

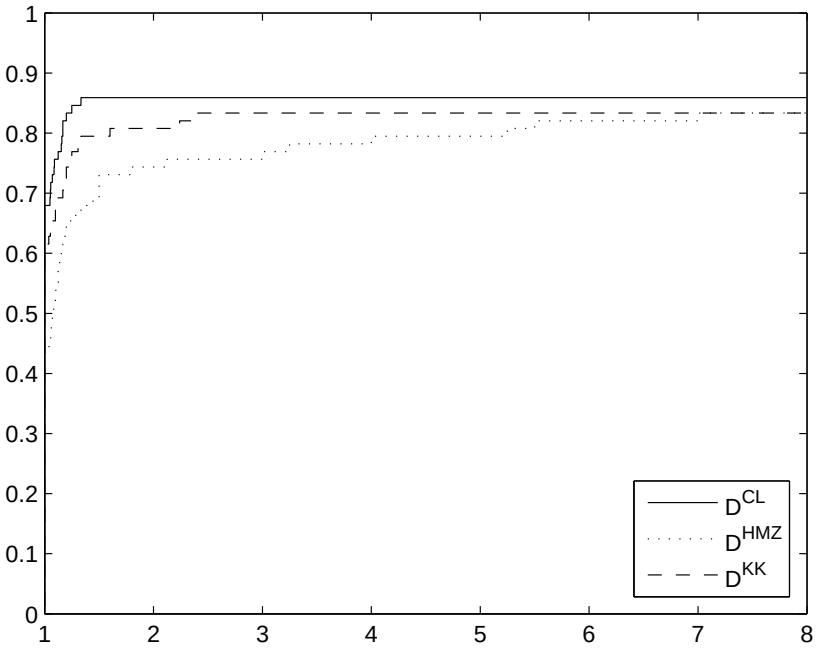

Figure 4: `CoDoSol` with different scalings: Perfomance profiles in terms of Nonlinear Iterations

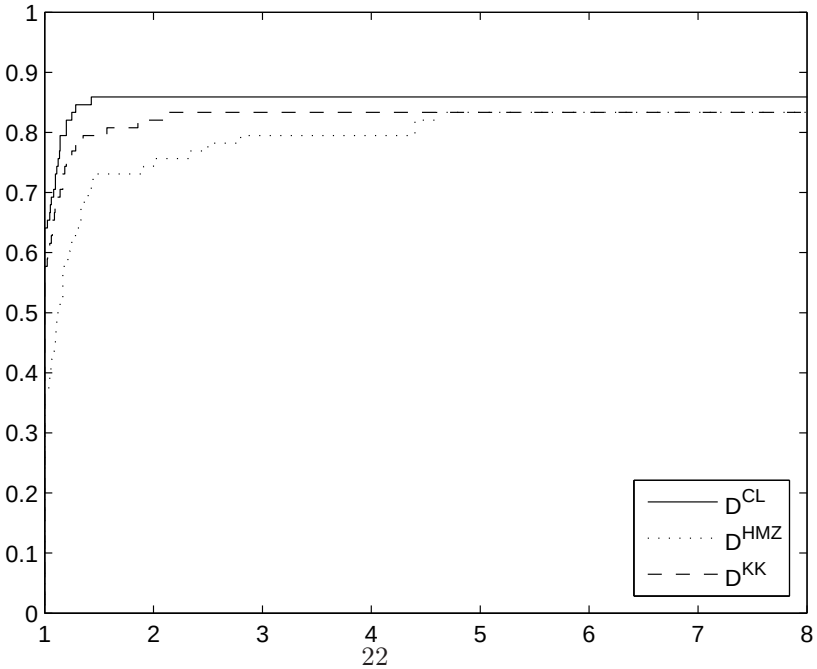

Figure 5: `CoDoSol` with different scalings: Performance profiles in terms of F-evaluations

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 Ω. The convergence analysis is performed without specifying 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] S. Bellavia, M. Macconi, and B. Morini, *An affine scaling trust-region approach to bound-constrained nonlinear systems*, Applied Numerical Mathematics, 44 (2003), pp. 257–280.

[2] ——, *STRSCNE: A scaled trust-region solver for constrained nonlinear equations*, Computational Optimization and Applications, 28 (2004), pp. 31–50.

[3] S. Bellavia, M. Macconi, and S. Pieraccini, *On affine scaling inexact dogleg methods for bound-constrained nonlinear systems*, Tech. Report 5/2009, 2009.

[4] S. Bellavia and B. Morini, *An interior global method for nonlinear systems with simple bounds*, Optimization Methods and Software, 20 (2005), pp. 1–22.

[5] ——, *Subspace trust-region methods for large bound constrained nonlinear equations*, SIAM J. Numer. Anal., 44 (2006), pp. 1535–1555.

[6] T. F. Coleman and Y. Li, *An interior trust-region approach for nonlinear minimization subject to bounds*, SIAM J. Optim., 6 (1996), pp. 418–445.

[7] T. A. Davis, *Algorithm 832: UMFPACK, an unsymmetric-pattern multifrontal method*, ACM Transactions on Mathematical Software, 30 (2004), pp. 196–199.

[8] S. P. Dirkse and M. C. Ferris, *MCPLIB: A collection of nonlinear mixed complementary problems*, Optimization Methods and Software, 5 (1995), pp. 319–345.

[9] E. D. Dolan and J. J. Moré, *Benchmarking optimization software with performance profiles*, Mathematical Programming, 91 (2002), pp. 201–213.

[10] C. A. Floudas and al., *Handbook of test problems in local and global optimization*, Kluwer Academic Publishers, Nonconvex Optimization and its Applications, 33, 1999.

[11] J. B. Francisco, N. Krejic, and J. M. Martinez, *An interior-point method for solving box-constrained underdetermined nonlinear systems*, Journal of Computational and Applied Mathematics, 177 (2005), pp. 67–88.

[12] W. W. Hager, B. A. Mair, and H. Zhang, *An affine-scaling interior-point CBB method for box-constrained optimization*, Mathematical Programming, (2007). Published online.

[13] M. Heinkenschloss, M. Ulbrich, and S. Ulbrich, *Superlinear and quadratic convergence of affine-scaling interior-point Newton methods for problems with simple bounds without strict complementarity assumptions*, Mathematical Programming, 86 (1999), pp. 615–635.

[14] W. Hock and K. Schittkowski, *Text examples for nonlinear programming codes*, in Lecture notes in Economics and Mathematicsl Systems, vol. 187, 1981.

[15] C. Kanzow and A. Klug, *On affine-scaling interior-point Newton methods for nonlinear minimization with bound constraints*, Computational Optimization and Applications, 35 (2006), pp. 177–197.

[16] ——, *An interior-point affine-scaling trust-region method for semismooth equations with box constraints*, Computational Optimization and Applications, 37 (2007), pp. 329–353.

[17] D. N. Kozakevich, J. M. Martinez, and S. A. Santos, *Solving nonlinear systems of equations with simple bounds*, Journal of Computational and Applied Mathematics, 16 (1997), pp. 215–235.

[18] M. Macconi, B. Morini, and M. Porcelli, *A Gauss-Newton method for solving bound-constrained underdetermined nonlinear systems*, Optimization Methods and Software, 24 (2009), pp. 219–235.

[19] ——, *Trust-region quadratic methods for nonlinear systems of mixed equalities and inequalities*, Applied Numerical Mathematics, 59 (2009), pp. 859–876.

[20] K. Meintjes and A. P. Morgan, *A methodology for solving chemical equilibrium systems*, Applied Math. Comput., 22 (1987), pp. 333–361.

[21] ———, *Chemical equilibrium systems as numerical tests problems*, ACM Trans. Math. Soft., 16 (1990), pp. 143–151.

[22] J. P. Merlet, *Kinematic analysis of a steering mechanism: a follow up with interval analysis.* http://www-sop.inria.fr/coprin/logiciels/ALIAS/Examples/kinematic_jmd2003.pdf, 2003.

[23] J. J. Moré, *A collection of nonlinear model problems*, Lectures in Applied Mathematics, 26 (1990), pp. 723–762.

[24] R. P. Pawlowski, J. P. Simonis, H. F. Walker, and J. N. Shadid, *Inexact Newton dogleg methods*, SIAM J. Numer. Anal., 46 (2008), pp. 2112–2132.

[25] I. G. Tsoulos and A. Stavrakoudis, *On locating all roots of systems of nonlinear equations inside bounded domain using global optimization methods*, Nonlinear Analysis: Real World Applications, (2009). to appear.

[26] A. Wachter and L. T. Biegler, *Failure of global convergence for a class of interior point methods for nonlinear programming*, Mathematical Programming (Series A), 88 (2000), pp. 565–574.

[27] T. Wang, R. D. C. Monteiro, and J.-S. Pang, *An interior point potential reduction method for constrained equations*, Mathematical Programming, (1996), pp. 159–195.

[28] D. Zhu, *An affine scaling trust-region algorithm with interior backtracking technique for solving bound-constrained nonlinear systems*, Journal of Computational and Applied Mathematics, 184 (2005), pp. 343–361.