A TWO-DIMENSIONAL TRUST-REGION METHOD FOR LARGE SCALE BOUND-CONSTRAINED NONLINEAR SYSTEMS *

S. BELLAVIA, M. MACCONI, B. MORINI †

Abstract. We design a well-developed algorithm for solving large bound-constrained systems. The method combines a subspace trust-region strategy with a Newton-Krylov method and with a strategy for handling the bounds. To solve the trust-region problem efficiently, we look for an approximate solution and do not require the exact solution of the Newton equation. In addition, a strategy for obtaining an improvement on a generalized Cauchy point is discussed. Under mild assumptions, the convergence properties of the method are the same of a full space trust-region method for constrained systems. The reported numerical results show that our method is a robust and efficient way to solve large bound-constrained problems.

Key words. Inexact Newton methods, forcing terms, Krylov subspace methods, subspace trust-region problems, bound constraints.

1. Introduction. We consider bound-constrained nonlinear systems

\[ F(x) = 0, \quad l \leq x \leq u, \]  

where \( x \in \mathbb{R}^n \), \( l \in (\mathbb{R} \cup \{-\infty\})^n \), \( u \in (\mathbb{R} \cup \{\infty\})^n \), \( F : X \to \mathbb{R}^n \) with \( X \subseteq \mathbb{R}^n \). The function \( F \) is assumed to be continuously differentiable and \( X \) is assumed to be an open set containing the feasible region \( \Omega = \{ x \in \mathbb{R}^n : l \leq x \leq u \} \).

Bound-constrained nonlinear systems represent a wide range of problems in engineering and equilibrium modelling [4, 13, 14, 15]. Recent advances in designing numerical methods for (1.1) are given in [1, 3, 12, 16, 22, 28, 29]. These papers present methods for medium-size problems and deal with a general function \( F \).

Here, we are concerned with numerical methods that are computationally effective for large bound-constrained systems. The authors are aware of the works [2, 18] for problems with general nonlinear functions and of the paper [23] devoted to large Karush-Kuhn-Tucker (KKT) systems. Focusing on the approach proposed in [2], we design a procedure that combines a subspace trust-region approach with a Newton-Krylov method and with a strategy for handling the bounds.

The core of the adopted approach is a Newton-Krylov method for the problem \( F(x) = 0 \). Newton-Krylov methods are widely used to solve large unconstrained nonlinear systems [10, 11, 20]. They are Inexact methods [8] where the Newton equation

\[ F'(x_k)p = -F(x_k), \]  

is solved approximately by a Krylov method [25]. Fixed a forcing term \( \eta_k \in [0, 1) \), the Inexact Newton step \( p_k^I \) meets the following termination criterion

\[ \|F'(x_k)p_k^I + F(x_k)\| \leq \eta_k\|F(x_k)\|. \]  

The sequence \( \{\eta_k\} \) affects the local rate of convergence and suitable choices of the forcing terms ensure very good local behaviour, up to quadratic rate of convergence of the exact Newton method, [11].

*Work supported by MIUR, Rome, Italy, through “Cofinanziamenti Programmi di Ricerca Scientifica di Interesse Nazionale” and GNCS-INDAM.
†Dipartimento di Energetica, Univ. of Florence, via C. Lombroso 6/17, 50134 Florence, Italy, e-mail: stefania.bellavia@unifi.it, macconi@de.unifi.it, benedetta.morini@unifi.it
The globalization strategy proposed in [2] is a trust-region scheme where both elliptical and spherical trust-regions are allowed. In order to solve large problems effectively, the trust-region problem is defined over a small dimensional space. Thus, the work to solve the subspace trust-region problem is trivial while the dominant work has been shifted to the determination of the subspace. An important contribution of this proposal is the employment of the by-products of the Krylov method in the definition of the subspace trust-region problem and its solution. The theoretical study carried out in [2] shows how to control the inexactness arising from the linear iterative solver and how to restrict the trust-region problem to a low dimensional space while retaining fast convergence of the overall method. In particular, it has been shown that if spherical trust-regions are used then fast local convergence is achieved under milder assumptions.

In this work we discuss several basic issues to provide a theoretically well founded algorithm and describe its implementation in a reliable and efficient solver. In particular we comment on some implementation aspects of the adopted subspace method and discuss the following main points: the choice of the subspace trust-region problem employed in the globalization strategy; the numerical solution of the trust-region problem and the choice of the sequence \( \{\eta_k\} \).

This discussion gives rise to a Newton-Krylov method embedded into a two-dimensional trust-region technique. Spherical trust-regions are employed and cheap approximations to the subspace trust-region problems are computed. The procedure generates a sequence of iterates belonging to the strict interior of \( \Omega \) and it shows global convergence and potential quadratic convergence to a feasible solution. We demonstrate the benefit of our trust-region strategy with computational results. In fact, the ease and inexpensiveness with which the two-dimensional subspace and the approximate trust-region solution can be formed let our method to outperform the classical dogleg method.

In §2 we present the method and discuss a practical way to develop its implementation. In §3 we discuss numerical experiments and demonstrate the computational benefit of our approach.

\subsection*{1.1. Some notations.}

The Euclidean vector norm or the subordinate matrix norm are indicated by \( \| \cdot \| \). We use the subscript index \( k \) to represent the evaluation of a function at point \( x_k \) of the sequence \( \{x_k\} \). The gradient of the merit function \( f(x) = \|F(x)\|^2/2 \) is denoted by \( \nabla f(x) \), i.e. \( \nabla f(x) = F'(x)^T F(x) \). The symbol \( (x)_i \) represents the \( i \)th component of the vector \( x \in \mathbb{R}^n \). When clear from the context, the brackets are omitted.

We let \( v(x) \in \mathbb{R}^n \) denote the vector function with components \( v_i(x) \) given by

\[
(1.4) \quad v_i(x) = \begin{cases} 
    x_i - u_i & \text{if } (\nabla f(x))_i < 0 \text{ and } u_i < \infty \\
    x_i - l_i & \text{if } (\nabla f(x))_i > 0 \text{ and } l_i > -\infty \\
    \min\{x_i - l_i, u_i - x_i\} & \text{if } (\nabla f(x))_i = 0 \text{ and } l_i > -\infty \text{ or } u_i < \infty \\
    1 & \text{otherwise}
\end{cases}
\]

and \( D(x) \) be the diagonal matrix introduced in [7]

\[
(1.5) \quad D(x) = \text{diag}(|v_1(x)|^{-1/2}, |v_2(x)|^{-1/2}, \ldots, |v_n(x)|^{-1/2}).
\]

Finally, for each vector \( x \), we set

\[
(1.6) \quad x^L = l - x \quad \text{and} \quad x^U = u - x.
\]
2. The two-dimensional subspace interior trust-region method. The method under study in this work belongs to the class of the subspace interior trust-region procedures investigated in [2]. It generates a sequence of iterates \( \{ x_k \} \) belonging to the interior of \( \Omega \) and performs the following main steps at each iteration.

Let \( x_k \in \text{int}(\Omega) \) and \( \eta_k \in [0, 1) \) be given. First, an Inexact Newton step \( p^I_k \) is computed by imposing condition (1.3) and a two dimensional trust region problem is built. Second, the solution of the two-dimensional trust-region problem and a generalized Cauchy point are used to determine a trial step. Third, an acceptance mechanism is used to decide if the resulting trial point is the next iterate. A trust-region updating strategy completes the iteration step.

2.1. The two-dimensional trust-region step. In a trust-region procedure for nonlinear systems, a region around the current iterate \( x_k \) is defined and within such region the following local model of

\[
m_k(p) = \frac{1}{2} \| F(x_k) + F'(x_k)p \|^2,
\]

is trusted to be an adequate representation of the merit function

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

Specifically, the trust-region step is ideally

\[
\min_{p \in \mathbb{R}^n} \{ m_k(p) : \| p \| \leq \Delta_k \},
\]

where \( \Delta_k \) is the trust-region radius, i.e. the length of the step for which the local model is trusted.

Several algorithms exist for solving (2.3), e.g. see [6]. Some of them look for a nearly exact solution and typically involve the computation of eigenvalues and a Newton process applied to the secular equation. Other approaches find approximate solutions by simple strategies and are less costly to be implemented.

Solving the full space trust-region problem (2.3) is expensive if the problem is large. For example, approximating a solution by the dogleg method requires the solution of the Newton equation (1.2). An alternative consists in minimizing \( m_k \) on a small dimension subspace \( S_k \), replacing (2.3) by

\[
\min_{p \in S_k} \{ m_k(p) : \| p \| \leq \Delta_k \},
\]

see e.g. [2, 5, 26, 27]. Once the subspace \( S_k \) has been computed, the work to solve (2.4) is trivial due to the low dimension of \( S_k \).

Here, we choose a two-dimensional subspace \( S_k \) with the aid of a Krylov method. In fact, we take advantage of the use of sparse linear algebra techniques applied to (1.2) and we assign \( S_k = \text{span}\{ p^I_k, \nabla f_k \} \) where \( p^I_k \) is an inexact Newton step given in (1.3). In [2] it has been shown that this subspace and a suitable choice of the forcing terms provide fast convergence. Moreover, even if only an approximate solution of (2.4) is computed, global convergence and fast local convergence are guaranteed, as well. Then, we allow for an approximate solution \( p^S_k \) of the subspace trust-region problem and consider the following dogleg strategy to form such approximation at a very low computational cost.

**Dogleg Algorithm for (2.4)**

Given \( x_k \in \text{int}(\Omega) \), \( p^I_k \in \mathbb{R}^n \), \( \Delta_k > 0 \).
1. Compute \( \nabla f_k \) and \( w_1 = \nabla f_k / \| \nabla f_k \| \).
2. Compute \( w_2 = p_k^e T (w_1^T p_k^e) w_1 \). Set \( w_2 = w_2 / \| w_2 \| \).
3. Set \( W = [w_1, w_2] \in \mathbb{R}^{n \times 2} \).
4. Compute \( \tau_1 = \| W^T \nabla f_k \|^2 / \| F_k^e W W^T \nabla f_k \|^2 \),
   \( \tau_2 = \Delta_k / \| W^T \nabla f_k \| \),
   \( \tau = \min \{ \tau_1, \tau_2 \} \).
5. Compute the two-dimensional Cauchy step \( s^c = -\tau W^T \nabla f_k \).
6. Solve the linear system
   \[
   \begin{pmatrix}
   \| F_k^e w_1 \|^2 & w_2^T F_k^e F_k^e w_1 \\
   w_2^T F_k^e F_k^e w_1 & \| F_k^e w_2 \|^2
   \end{pmatrix}
   \begin{pmatrix}
   s_1^m \\
   s_2^m
   \end{pmatrix}
   = \begin{pmatrix}
   \| \nabla f_k \|^2 \\
   0
   \end{pmatrix},
   \]
   and set \( s^m = (s_1^m, s_2^m)^T \).
7. Determine the two-dimensional dogleg step \( s_k \):
   \[
   s_k = \begin{cases} 
   s^m & \text{if } \| s^m \| \leq \Delta_k; \\
   s^c & \text{if } \| s^c \| \leq \Delta_k; \\
   (1 - \sigma)s^c + \sigma s^m & \text{otherwise}
   \end{cases}
   \]
   where \( \sigma \in (0, 1) \) is uniquely determined so that \( \| s_k \| = \Delta_k \).
8. Compute \( p_k^S_s = W s_k \).

In Steps 1-3 we form an orthonormal basis \( W = [w_1, w_2] \in \mathbb{R}^{n \times 2} \) for \( S_k \). Thus, for each vector \( p \in S_k \) there exists \( s \in \mathbb{R}^2 \) such that \( p = W s \) and problem (2.4) can be written as
\[
(2.5) \quad \min_{s \in \mathbb{R}^2} \{ \| F_k + F_k^e W s \|^2 : \| s \| \leq \Delta_k \}.
\]

Steps 5-7 work in the reduced space and perform a standard dogleg method to solve (2.5). The Cauchy point \( s^c \in \mathbb{R}^2 \) is found in Steps 4-5 and the solution \( s^m \in \mathbb{R}^2 \) to the unconstrained minimization problem \( \min_{s \in \mathbb{R}^2} \| F_k + F_k^e W s \|^2 \) is evaluated in Step 6 by solving the \( 2 \times 2 \) normal linear system. Then, the dogleg solution \( s_k \) to (2.5) is formed in Step 7. We remark that the vector \( s^m \) could be computed also via the QR decomposition of the \( n \times 2 \) matrix \( F_k^e W \). Anyway, the computational effort of this dogleg procedure is very low.

We remark that, if the linear system (1.2) is solved up to the full accuracy, i.e. the Krylov solver is applied with a tight stopping criterion, the two dimensional subspace dogleg strategy reduces to the standard dogleg strategy.

The step \( p_k^{TR} \) obtained in Step 8, may not be well suited for being used in our constrained context since the point \( x_k^{TR} = x_k + p_k^{TR} \) may be infeasible. Therefore, we need to introduce a further vector \( p_k^S_s \) which represents the actual trust-region step we employ. It is such that \( p_k^{TR} = p_k^s \) if \( x_k^s \in \text{in}(\Omega) \). Otherwise, \( p_k^{TR} \) gives a point \( x_k + p_k^{TR} \) within \( \Omega \) by shifting the projection \( \max \{ l, \min \{ x_k^S, u \} \} \) of \( x_k^S \) onto \( \Omega \) toward the interior of \( \Omega \). In other words, the trust region step \( p_k^{TR} \) is defined as follows
\[
(2.6) \quad (p_k^{TR})_i = \begin{cases} 
\min \{ (1 - \alpha)(x_k^L)_i, (2x_k^L - p_k^S)_i \}, & \text{if } (x_k^S)_i \leq l_i \\
\max \{ (1 - \alpha)(x_k^U)_i, (2x_k^U - p_k^S)_i \}, & \text{if } (x_k^S)_i \geq u_i \\
(p_k^S)_i, & \text{otherwise}
\end{cases}
\]
for \( i = 1, \ldots, n \). Here \( x_k^L \) and \( x_k^U \) are given by (1.6) and \( \alpha \in (0, 1) \) is the amplitude of the performed shift.
2.2. The generalized Cauchy point $p_k^C$. The vector $p_k^C$ was introduced in [7]. It represents a constrained scaled Cauchy point and combines the ideas of the direction of steepest descent in a scaled space with the treatment of the bound constraints.

First, we consider the following scaled gradient of $f$ at $x_k$

$$d_k = -D_k^{-2} \nabla f_k,$$

where $D_k = D(x_k)$ is defined by (1.5). Scaling is made to produce a direction which is well-angled with respect to the bounds.

Second, we compute the minimizer of $m_k$ along $d_k$ subject to satisfy the trust-region bounds and let $\tau_k = \arg\min_{\tau > 0} \{ m_k(\tau d_k) : \| \tau d_k \| \leq \Delta_k \}$.

Then, the generalized Cauchy point is such that

$$p_k^C = c_k d_k,$$

where

$$c_k = \begin{cases} \tau_k & \text{if } x_k + \tau_k d_k \in \text{int}(\Omega) \\ \lambda_k & \text{otherwise} \end{cases}$$

and

$$\tau_k = \min \{ \frac{\| D_k^{-1} \nabla f_k \|^2}{\| D_k^{-1} D_k^{-2} \nabla f_k \|^2}, \frac{\Delta_k}{\| D_k^{-1} \nabla f_k \|} \},$$

$$\lambda_k = \min_{1 \leq i \leq n} \theta \Lambda_i, \quad \theta \in (0, 1), \quad \Lambda_i = \left\{ \max \left\{ \frac{(x_i^T) \Delta_k}{(d_k)_{(i)}}, \frac{(z_i^T) \Delta_k}{(d_k)_{(i)}} \right\} \right\} \text{ if } (d_k)_i \neq 0 \quad \text{otherwise}$$

In practice, to enforce $x_k + p_k^C \in \text{int}(\Omega)$, a simple scaling is performed when $x_k + \tau_k d_k$ lies outside $\text{int}(\Omega)$.

2.3. The trial step $p_k$. Given the current iterate $x_k$, the trial step $p_k$ is designed to give a candidate $x_{k+1} \in \text{int}(\Omega)$ and to improve on the Cauchy point $p_k^C$. It means that $p_k$ is required to satisfy the decrease condition

$$\rho_c(p_k) = \frac{m_k(0) - m_k(p_k)}{m_k(0) - m_k(p_k^C)} \geq \beta_1,$$

with $\beta_1 \in (0, 1)$.

In order to choose $p_k$, we follow the lines of [1]. We let $p_k^{TR}$ play the role of the trial step, i.e. $p_k = p_k^{TR}$, if it gives the reduction (2.10). Otherwise, we let

$$p_k = t p_k^C + (1 - t) p_k^{TR},$$

where $t \in [0, 1)$ is fixed so that $\rho_c(p_k) = \beta_1$. This way, the value of $t$ is given by

$$t = \frac{z^T u - w}{\| u \|^2},$$

where $u = F_k(p_k^C - p_k^{TR}), \quad z = -(F_k + F_k p_k^{TR}), \quad w = \left( (z^T u)^2 - 2 \| u \|^2 (F_k^T F_k (p_k^{TR} - \beta_1 p_k^C) + \frac{1}{2} \| F_k p_k^{TR} \|^2 - \frac{1}{2} \| F_k p_k^C \|^2) \right)^{\frac{1}{2}}$. 

5
2.4. Accepting $p_k$. The trial step $p_k$ must provide a good agreement between $m_k$ and the merit function $f$, too. Thus, the condition

\begin{equation}
\rho_f(p_k) = \frac{f(x_k) - f(x_k + p_k)}{m_k(0) - m_k(p_k)} \geq \beta_2,
\end{equation}

is tested for a given constant $\beta_2 \in (0, 1)$. If this condition doesn’t hold, then the step $p_k$ is rejected, the trust-region radius $\Delta_k$ is reduced and a new trial step is computed.

If (2.13) holds, then the new iterate $x_{k+1} = x_k + p_k$ is formed and the trust-region radius is increased or kept the same for the next iteration. In practice, we adopt the procedure commonly used for unconstrained nonlinear systems, i.e. at the end of each iteration, we test the condition

\begin{equation}
\rho_f(p_k) \geq \beta_3,
\end{equation}

where $\beta_3$ is a given positive constant such that $\beta_2 \leq \beta_3 \leq 1$. If (2.14) holds, we allow a possible increasing in $\Delta_k$, otherwise the trust region radius is kept the same.

2.5. The method and its implementation. We have discussed the main steps of each iteration of the inexact subspace trust-region method. Now we are ready to define it formally.

**Two-dimensional subspace interior trust-region method**

Given: $x_0 \in \text{int}(\Omega)$, $\Delta_0 > 0$, $\alpha \in (0, 1)$, $\beta_1 \in (0, 1)$, $0 < \beta_2 < \beta_3 < 1$, $\theta \in (0, 1)$.

For $k = 0, 1, \ldots$ do

1. Choose $\eta_k \in [0, 1)$.
2. Apply a Krylov method to (1.2) to find a vector $p_k^I$ satisfying (1.3).
3. Compute $p_k^S$ by the Dogleg Algorithm.
4. Compute $p_k^{TR}$ by (2.6).
5. Compute $p_k^C$ by (2.8).
6. If $\rho_c(p_k^{TR}) \geq \beta_1$ set $p_k = p_k^{TR}$ otherwise compute $p_k$ by (2.11) with $t$ given by (2.12).
7. If $\rho_f(p_k) < \beta_2$ reduce $\Delta_k$ and go to Step 3.
8. Set $x_{k+1} = x_k + p_k$.
9. Choose $\Delta_{k+1} \geq \Delta_k$ according to (2.14).

This method fits a specific procedure in the Subspace Interior Affine Trust-Region framework recently proposed in [2]. Then, the convergence properties of this method can be stated using the convergence results provided in that paper. First, Lemma 3.2 of [2] ensures that if $F'_k$ is not singular, then an acceptable step $p_k$ is produced after a finite number of reductions of the trust-region radius. Moreover, under standard conditions, the method is globally convergent to a solution $x^*$ of (1.1) and a suitable choice of the sequence $\{\eta_k\}$ yields fast local convergence up to the rate of Newton method. The global and local convergence properties of our method are summarized in the next Theorem.

**Theorem 2.1.** Let $\{x_k\}$ be the sequence generated by the method. Assume that there exists $r > 0$ such that $F'$ is Lipschitz continuous in $L = \bigcup_{k=0}^{\infty} \{x \in X : \|x - x_k\| \leq r\}$ and $\|F'\|$ is bounded above on $L$. 

6
If \( \{x_k\} \) is bounded and \( F'_k \) is nonsingular for all \( k \geq 0 \), then the limit points of \( \{x_k\} \) are stationary points for the problem \( \min_{x \in \Omega} f \). Further, if there exists a limit point \( x^* \in \text{int}(\Omega) \) of \( \{x_k\} \) such that \( F'(x^*) \) is nonsingular, then \( \|F_k\| \to 0 \) and all the accumulation points of \( \{x_k\} \) solve problem (1.1) (see [2, Theorem 4.1]).

If \( \{x_k\} \) is bounded and \( x^* \) is an isolated limit point of \( \{x_k\} \) such that \( F(x^*) = 0 \) and \( F'(x^*) \) is nonsingular, then \( \{x_k\} \) converges to \( x^* \) (see [2, Theorem 4.2]).

If \( \{x_k\} \) converges to a solution \( x^* \) of (1.1) such that \( F'(x^*) \) is nonsingular and \( \eta_k \to 0 \) as \( k \to \infty \), then \( \{x_k\} \) converges to \( x^* \) superlinearly. If \( \eta_k = O(||F_k||) \) as \( k \to \infty \), the convergence rate is quadratic (see [2, Theorem 4.3]).

The given method can be implemented so as to keep the computational cost at a reasonable level. In fact, recalling the discussion of \( \S 2.1-2.4 \), Steps 3-6 require a very low computational effort. In particular, \( \nabla f_k \) must be formed and this calls for a matrix-vector product. Computing \( p_k^2 \) requires one step of the Gram-Schmidt procedure and the solution of a \( 2 \times 2 \) linear system. To form the linear system two matrix-vector products are required. The evaluation of the generalized Cauchy step \( p_k^C \) involves one matrix-vector product to compute the scalar \( \tau_k \) in (2.9) and the computation of the actual step \( p_k \) requires two matrix-vector products for obtaining the scalar \( t \) given in (2.12).

The main cost of the overall procedure is given by the computation of the Inexact Newton step \( p_k^1 \) by a Krylov method. In our implementation, to compute \( p_k^1 \) we used GMRES(20) [25] with a maximum of 49 restart, for a total of 1000 GMRES iterations. If after 1000 GMRES iterations condition (1.3) is not met, the implemented algorithm is stopped. According to the results of [21] we implemented the further choice \( \eta_k \) given by the last computed GMRES iterate. The safeguards suggested in [21, p. 305] are also used.

Regarding the choice of the forcing terms, we recall that appropriate choices yield fast local convergence and minimize oversolving, i.e. avoid wasting effort in the initial stages of the method, [11]. According to the results of [21] we implemented the following choices of the forcing terms

\[
(2.15) \quad \text{Choice 1: } \eta_0 = 0.9, \quad \eta_k = \frac{||F_k|| - ||F_{k-1} + F'_k p_k||}{||F_{k-1}||}, \quad k > 0,
\]

\[
(2.16) \quad \text{Choice 2: } \eta_0 = 0.9, \quad \eta_k = 0.9 \left( \frac{||F_k||}{||F_{k-1}||} \right)^2, \quad k > 0.
\]

Moreover, we used the safeguards suggested in [21, p. 305].

To compare the performance of our trust-region dogleg strategy with the standard one, we implemented the further choice

\[
(2.17) \quad \text{Choice 3: } \eta_0 = 0.9, \quad \eta_k = \max \{100 \epsilon_m / ||F_k||, 100 \epsilon_m\}, \quad k > 0,
\]

where \( \epsilon_m \) is the machine precision. This tight stopping criterion allows for the solution of (1.2) with a very high accuracy. Thus, the dogleg strategy involving the Newton step is performed.

In Step 7, the trust-region radius is reduced by \( \Delta_k = \min \{\Delta_k/4, ||p_k||/2\} \) while in Step 9 we allow a possible increase in the trust region radius and we set \( \Delta_{k+1} = \max \{\Delta_k, 2 ||p_k||\} \) if (2.14) holds and \( \Delta_{k+1} = \Delta_k \) otherwise. The value \( \Delta_0 = 1 \) is used as initial trust-region radius.

The application of our method requires further parameters. We set \( \alpha = 5 \times 10^{-2} \), \( \beta_1 = 0.1 \), \( \beta_2 = 0.25 \), \( \beta_3 = 0.75 \), \( \theta = 0.99995 \).

Convergence is declared when

\[
(2.18) \quad ||F_k|| \leq \text{tol}.
\]
where $tol > 0$ is a specified tolerance.

Failure is declared if one of the following situations occurs:

- a maximum number of 200 iterations are performed;
- a maximum number of 1000 F-evaluations are performed;
- the trust-region size is reduced below $\sqrt{\epsilon_m}$;
- $\|F_{k+1} - F_k\| \leq 100\epsilon_m \|F_k\|$, i.e. the algorithm did not manage to escape from a local minimizer of the merit function.
- $\|D_k^{-1}\nabla f_k\| < 100\epsilon_m$, i.e. the sequence $\{x_k\}$ is approaching a minimum of $f$ in $\Omega$.

3. Numerical results. We implemented the inexact two-dimensional trust-region method in a Double Precision FORTRAN program. Here we report numerical experiments performed with machine precision $\epsilon_m = 2 \times 10^{-16}$. For the stopping criterion (2.18) we set $tol = 10^{-6}$. The problems solved are the following.

P1. The Chandrasekhar H-equation. As a first problem we chose to solve the discrete analog [17, p. 87] of the Chandrasekhar H-equation arising in radiative transfer problems. It is known that the Chandrasekhar H-equation and the discrete analog depend on a parameter $c \in (0, 1]$ and become more difficult to solve as $c$ approaches 1. Further, only one solution has physical meaning. We approximated this solution by solving the problem within the box defined by $l_i = 0$ and $u_i = 10^{300}$ for $i = 1, \ldots, n$. In our experiments we fixed $n = 400$ and considered three challenging cases: $c = 0.99, 0.9999, 1$. For each value of $c$ we started from four different initial points, i.e. $x_0 = 10^{k-2}$ with $k = 0, 1, 2, 4$. This way, we obtained 12 tests.

P2. A PDE problem. As a second problem, we chose to solve the elliptic PDE given in [11]. This problem has more than one solution but only one is positive everywhere. Centered differences on a $100 \times 100$ uniform grid give a nonlinear system of dimension $n = 10^4$. To approximate the positive solution we solved the nonlinear system within the feasible region defined by $l_i = 10^{-4}$ and $u_i = 10^{300}$ for $i = 1, \ldots, n$. We started our method from the initial guess given in [11] which depends on a parameter $k$. We solved the problem with several values of $k$ and here we report the results obtained by setting $k = 100, 500, 1000$. This way, we generated three tests.

P3. A chemical equilibrium system. A nonlinear system of $n = 11000$ equations was obtained by augmenting System 1 of [19]. Taking into account the physical meaning of the solutions, we solved this problem within the feasible region $\Omega$ defined by $l_i = 0$ and $u_i = 10^{300}$ for $i = 1, \ldots, n$. The initial guesses used have the form $x_0 = 10^{k-2}$ with $k = 0, 1, 2, 4$ and give rise to four tests.

P4. The Bratu problem. This nonlinear complementarity problem [9] was reformulated as a system of $n = 5 \times 10^4$ smooth bound-constrained nonlinear equations with $l_i = 0$ and $u_i = 10^{300}$ for $i = 1, \ldots, n$. It depends on a parameter $\lambda$ and we considered the values $\lambda = 3, 4, 5, 6$. For each value of $\lambda$ we started from $x_0 = 10^{k-2}$ with $k = 0, 1, 2, 4$. Then, we obtained sixteen tests.

In all our runs the Jacobian of $F$ was computed analytically. All tests from problems P1 and P3 were solved without preconditioning while the incomplete LU factorization ILU(0) preconditioner was used in the solution of the tests from problems P2 and P4.

We first report the results obtained using Choice 1 and Choice 2. For each problem, Table 3.1. reports the dimension $n$ of the problem, the number of test associated, the number $S$ of successes and the number $F$ of failures. Both implementations are
fairly robust and show that the accuracy requirements proposed are sufficient to guarantee the global convergence of the inexact method. On a total of 35 tests, 28 cases were successfully solved using both Choice 1 and Choice 2. Furthermore, the results show six and five failures for Choice 1 and Choice 2, respectively. All the tests arising from problems P1 and P3 were solved while failures occurred in problem P4 starting from $x_0 = 10^2$ for all the values of the parameter $\lambda$ and for both choices of the forcing terms.

The overall performance from Choice 1 and Choice 2 are depicted in Figure 3.1. In these graphs results on the tests solved by both choices are given. The tests solved are numbered progressively. More specifically, tests from 1 through 12 arise from P1, test number 13 is P2 with $x_0$ defined with $k = 100$, tests from 14 to 17 arise from P3 and tests from 18 through 28 arise from P4. For each test we display the number

<table>
<thead>
<tr>
<th>Problem</th>
<th>Tests</th>
<th>n</th>
<th>S</th>
<th>F</th>
<th>S</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>12</td>
<td>400</td>
<td>12</td>
<td>0</td>
<td>12</td>
<td>0</td>
</tr>
<tr>
<td>P2</td>
<td>3</td>
<td>1000</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>P3</td>
<td>4</td>
<td>11000</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>P4</td>
<td>16</td>
<td>50000</td>
<td>12</td>
<td>4</td>
<td>11</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 3.1

Number of successes and failures with Choice 1 and Choice 2

Fig. 3.1. Performance of the method with Choice 1 and Choice 2
of function evaluations and GMRES iterations required. The number of function evaluations is small for all runs but Choice 2 performs slightly better than Choice 1. The number of nonlinear iterations is small as well, since it is almost the same as the number of functions evaluations. Therefore, in practice the trust-region performs well and the method is locally fast.

Concerning the number of GMRES iterations, Choice 1 and Choice 2 are equivalent except for problem P4 with $\lambda = 6$. In this latter case, the best performance is from Choice 1 since it produces larger forcing terms and avoid the “oversolving” phenomenon. However, it is worth noting that conversely the number of nonlinear iterations increases.

Finally, we investigated if there is a significant difference between using the dogleg subspace trust-region step or the classical trust-region step in the globalization strategy. To this end we adopted Choice 3 for the forcing terms. With this choice, all tests from problems P1 and P3 were successfully solved, while two convergence failures occurred for two tests from P4. Further, the classical dogleg strategy was not applicable for two tests from P2 and five tests from P4 since the stopping criterion (1.3) was not met by GMRES within 1000 iterations. We remark that all the tests solved by using Choice 3 were solved by our inexact procedure, too. Then, the obtained numerical results confirm that the robustness of the classical dogleg method is not deteriorated by the discussed inexact approach.

Now we focus on the set of 25 tests where the method succeeded for all the choices

Fig. 3.2. *Performance of the two-dimensional dogleg and the classical dogleg method*
of the forcing terms and compare the number of function evaluations and GMRES iterations required. The results for these tests cases are shown in Figure 3.2. Also in Table 3.2 for each family of tests and each choice of the forcing terms, we report the mean of the number of nonlinear iterations ($It$), the mean of the number of function evaluations ($Fev$) and the mean of the number of GMRES iterations ($It_{GM}$) performed. We remark that Choice 3 is more expensive in terms of GMRES iterations. This is a predictable conclusion due to the use of overly stringent accuracy requirements. At the same time, except for tests from Problem P3, the number of F-evaluations required by Choice 3 is almost the same as by Choice 2. The results of Choice 3 on tests from Problem P3 are unexpected; in fact, with this tight choice of the forcing terms the method was substantially slower than with Choice 1 and Choice 2.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Choice 1</th>
<th>Choice 2</th>
<th>Choice 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>It</td>
<td>Fev</td>
<td>It</td>
</tr>
<tr>
<td>P1</td>
<td>17</td>
<td>18</td>
<td>29</td>
</tr>
<tr>
<td>P2</td>
<td>18</td>
<td>19</td>
<td>288</td>
</tr>
<tr>
<td>P3</td>
<td>29</td>
<td>31</td>
<td>176</td>
</tr>
<tr>
<td>P4</td>
<td>19</td>
<td>20</td>
<td>199</td>
</tr>
</tbody>
</table>

Table 3.2

Summary test results over the problems solved with the three choices of the forcing terms

REFERENCES