Inexact Trust Region Method for Large Sparse Systems of Nonlinear Equations

Lukšan, Ladislav
1993

Dostupný z `http://www.nusl.cz/ntk/nusl-33459`

Dílo je chráněno podle autorského zákona č. 121/2000 Sb.

Tento dokument byl stažen z Národního úložiště šedé literatury (NUŠL).

Datum stažení: 20.04.2024

Další dokumenty můžete najít prostřednictvím vyhledávacího rozhraní nusl.cz .

INSTITUTE OF COMPUTER SCIENCE

ACADEMY OF SCIENCES OF THE CZECH REPUBLIC

Prague

Inexact Trust Region Method for Large Sparse Systems of Nonlinear Equations

L. Lukšan

Technical Report No. V-547

February 1993

Akademie věd České republiky

ÚSTAV INFORMATIKY A VÝPOČETNÍ TECHNIKY

Institute of Computer Science, Academy of Sciences of the Czech Republic

Pod vodárenskou věží 2, 182 07 Prague 8, Czech Republic

E-mail: ICS@uivt.cas.cz

Fax: (+422) 8585789 Phone: (+422) 846669, (+422) 66051111

Inexact Trust Region Method for Large Sparse Systems of Nonlinear Equations

Ladislav Lukšan

Institute of Computer Science,
Academy of Sciences of the Czech Republic, Pod vodárenskou věží 2,
18207 Prague 8, Czech Republic

Technical Report No. V-547, February 1993

Abstract. The main purpose of this paper is to prove global convergence of the new trust region method based on smoothed CGS algorithm. This method is suprisingly convenient for numerical solution of large sparse systems of nonlinear equations as it is demonstrated by numerical experiments. A modification of the proposed trust region method do not use matrices, so it can be used for large dense systems of nonlinear equations.

Key Words. nonlinear equations, sparse systems, trust region methods

1. Introduction

Let $f_j: \ R^n \to R$, $1 \le j \le n$, be real-valued functions with continuous second order derivatives. We are concerned with finding a solution $x^\star \in R^n$ of the system of nonlinear equations.

$$f_j(x) = 0, \quad 1 \le j \le n \tag{1}$$

If such a solution exists then it is a global minimum of the function

$$F(x) = (1/2) \parallel f(x) \parallel^2 = (1/2) f^T(x) f(x) \tag{2}$$

where $f(x) = [f_1(x), \dots, f_n(x)]^T$.

Numerical methods for solving systems of nonlinear equations are usually derived from the Newton method. The Newton method is iterative and its iteration step has the form

$$x^+ = x + d \tag{3}$$

where x and x^+ are old and new vectors of variables respectively and $d = x^+ - x$ is the direction vector obtained as a solution of the linear system

$$J(x)d = -f(x) \tag{4}$$

where $J(x)$ is the Jacobian matrix of the original system (1) (its rows are gradients of the functions $f_j(x)$, $1 \le j \le n$).

If the Jacobian matrix $J(x)$ is not invertible then the linear system (4) has no solution and the direction vector $d \in R^n$ is not defined. Furthermore, if the Jacobian matrix $J(x)$ is ill-conditioned then the direction vector $d \in R^n$ can have a too large euclidean norm and, therefore, the next point $x^+ = x + d$ can lie outside the region where the linearization (4) holds. For this reason the Newton method (3)-(4) has to be modified to have global convergence properties.

The most commonly used globalization approach for the Newton method is based on a trust region strategy applied to minimization of the function (2). Let us denote $g_j(x)$ and $G_j(x)$ as the gradients and the Hessian matrices of the functions $f_j : \ R^n \to \ R$, $1 \le j \le n$ respectively, and $g(x)$ and $G(x)$ the gradient and the Hessian matrix of the function $F : \ R^n \to R$ respectively. Then, using (2), we obtain

$$g(x) = \sum_{j=1}^{n} f_j(x)\, g_j(x) = J^T(x)\, f(x) \tag{5}$$

and

$$G(x) = \sum_{j=1}^{n} g_j(x)\, g_j^T(x) + \sum_{j=1}^{n} f_j(x)\, G_j(x) = J^T(x)\, J(x) + \sum_{j=1}^{n} f_j(x)\, G_j(x) \tag{6}$$

Let $B(x)$ be some approximation of the Hessian matrix $G(x)$ (usually $B(x) = J^T(x)\, J(x)$) and

$$Q(d) = (1/2) d^T B d + g^T d \tag{7}$$

where $g = g(x)$ and $B = B(x)$. Then a typical iteration step of a trust region strategy applied to minimization of the function (2) has the following form.

(T1) Direction determination :
Choose $d \in R^n$ so that

$$\| d \| \le \Delta \tag{8a}$$

$$\| d \| < \Delta \Rightarrow \| Bd + g \| \le \omega \| g \| \tag{8b}$$

and

$$-Q(d) \ge \bar{\epsilon}_0 \| g \| \min \left(\| d \|, \| g \| / \| B \| \right) \tag{9}$$

where $\Delta > 0$ is a trust region bound, $0 \le \omega \le \bar{\omega}_0 < 1$, $\bar{\epsilon}_0 > 0$ ($\bar{\omega}_0$ and $\bar{\epsilon}_0$ do not depend on the iteration step) $g = g(x)$ and $B = B(x)$.

(T2) Stepsize selection :
Set

$$x^+ = x + d \quad \text{if} \quad F(x+d) < F(x) \tag{10a}$$

$$x^+ = x \quad \text{if} \quad F(x+d) \ge F(x) \tag{10b}$$

(T3) Trust region update :
Compute

$$\rho = [F(x+d) - F(x)]/Q(d) \tag{11}$$

When $\rho < \bar{\rho}_1$ then determine the value

$$\beta = \frac{1}{2\,\{1 - [F(x+d) - F(x)]/g^T d\,\}}$$

(quadratic interpolation) and set

$$\Delta^+ = \bar{\beta}_1 \parallel d \parallel \quad \text{if} \quad \beta < \bar{\beta}_1 \tag{12a}$$

$$\Delta^+ = \beta \parallel d \parallel \quad \text{if} \quad \bar{\beta}_1 \le \beta \le \bar{\beta}_2 \tag{12b}$$

$$\Delta^+ = \bar{\beta}_2 \parallel d \parallel \quad \text{if} \quad \bar{\beta}_2 < \beta \tag{12c}$$

When $\bar{\rho}_1 \le \rho \le \bar{\rho}_2$ then set

$$\Delta^+ = \min\,(\Delta, \bar{\gamma}_2 \parallel d \parallel) \tag{13}$$

When $\bar{\rho}_2 < \rho$ then set

$$\Delta^+ = \min\;(\max\,(\Delta, \bar{\gamma}_1 \parallel d \parallel), \bar{\gamma}_2 \parallel d \parallel, \bar{\Delta}) \tag{14}$$

Here $0 < \bar{\beta}_1 \le \bar{\beta}_2 < 1 < \bar{\gamma}_1 < \bar{\gamma}_2$, $0 < \bar{\rho}_1 < \bar{\rho}_2 < 1$ and $\bar{\Delta} > 0$ (barred constants do not depend on the iteration step).

The trust region strategy with the iteration step (T1)-(T3) has strong global convergence properties (see Refs. 1-3). Even if it also works well for indefinite matrices $B(x)$, we confine our attention to the positive semidefinite case which appears if $B(x) = J^T(x)\,J(x)$.

The most complicated part of the trust region strategy is computing the vector $d \in R^n$ satisfying the conditions (8)-(9). There exist two basic possibilities. First, the vector $d \in R^n$ can be obtained as a solution of the problem

$$d = \arg \min_{\|d(\lambda)\| \le \Delta} Q(d(\lambda))$$

which leads to the repeated solution of the equation $(B + \lambda I)\ d(\lambda) + g = 0$ for selected values of λ. This way produces a well-convergent algorithm but for a large number of variables it is time consuming. The second possibility is very natural. The equation $Bd + g = 0$ is solved by some iterative method which generates the vectors $d_i \in R^n$, $i \in N$, having the following properties.

(D1) There exists an index $k \in N$, such that

$$\| Bd_k + g \| \le \omega \| g \| \tag{15}$$

for a given $0 \le \omega < 1$.

(D2) The sequence $Q(d_i)$, $1 \le i \le k$, is nonincreasing, i. e.

$$Q(d_{i+1}) \le Q(d_i) \tag{16}$$

for $1 \le i < k$.

(D3) It holds that

$$Q(d_1) \le -\bar{\epsilon}_0 \ \| g \|^2 \ / \ \| B \| \tag{17}$$

and

$$Q(\lambda d_1) \le -\bar{\epsilon}_0 \| g \| \| \lambda d_1 \| \tag{18}$$

for $0 < \lambda \le 1$.

The resulting vector $d \in R^n$ is then obtained as

$$d = \lambda d_1 \quad \text{if} \quad \| d_1 \| \ge \Delta \tag{19a}$$

$$d = d_i + \lambda(d_{i+1} - d_i) \quad \text{if} \quad \| d_i \| < \Delta \le \| d_{i+1} \| \tag{19b}$$

$$d = d_k \quad \text{if} \quad \| d_k \| < \Delta \tag{19c}$$

where the scaling factor $\lambda > 0$ is chosen so that $\| d \| = \Delta$. It is obvious that this choice together with (D1)-(D3) implies (8)-(9).

Steihaug Ref.3 has proved that the conditions (D1)-(D3) are satisfied for the conjugate gradient method applied to the equation $Bd + g = 0$. Our main propose is to prove that these conditions are also satisfied for the smoothed CGS method applied to the equation $Jd + f = 0$. This leads to the new inexact trust region method whose numerical properties are suprisingly good as it will be shown in Section 3.

2. Inexact Trust Region Method Based on Smoothed CGS algorithm

First we slightly reformulate the conditions (8)-(9) and (D1)-(D3) to obtain conditions more convenient for systems of nonlinear equations. Instead of (8)-(9) we use the conditions

$$\| d \| \leq \Delta \tag{20a}$$

$$\| d \| < \Delta \Rightarrow \| Jd + f \| \leq \omega \| f \| \tag{20b}$$

and

$$\| f \| - \| Jd + f \| \geq 2\, \bar{\epsilon}_0 \ \min(\| J \| \| d \|, \| f \|) \tag{21}$$

where $\Delta > 0$ is a trust region bound $0 \leq \omega \leq \bar{\omega}_0 < 1$, $\bar{\epsilon}_0 > 0$ ($\bar{\omega}_0$ and $\bar{\epsilon}_0$ do not depend on the iteration step) $f = f(x)$ and $J = J(x)$.

Lemma 2.1 Let $B(x) = J^T(x)\, J(x)$. Then (21) imply (9).

Proof : Using $B = J^T J$ we obtain

$$Q(d) = (1/2) d^T J^T J d + f^T J d = (1/2)(\| Jd + f \|^2 - \| f \|^2) \tag{22}$$

(see (5) and (7)). Therefore (21) implies

$$\begin{aligned}
-Q(d) &= (1/2)(\| f \| + \| Jd + f \|)(\| f \| - \| Jd + f \|) \\
&\geq (1/2) \| f \| (\| f \| - \| Jd + f \|) \\
&\geq \bar{\epsilon}_0 \| f \| \min(\| J \| \| d \|, \| f \|) \\
&= \bar{\epsilon}_0 \| J \| \| f \| \min(\| d \|, \| J \| \| f \| / \| J \|^2) \\
&\geq \bar{\epsilon}_0 \| J^T f \| \min(\| d \|, \| J^T f \| / \| J^T J \|) \\
&= \bar{\epsilon}_0 \| g \| \min(\| d \|, \| g \| / \| B \|)
\end{aligned}$$

since $\| g \| = \| J^T f \| \leq \| J \| \| f \|$ and $\| B \| = \| J^T J \| = \| J \|^2$. □

Lemma 2.2 Let $B(x) = J^T(x)J(x)$. Then (20b) implies

$$\| d \| \geq (1 - \bar{\omega}_0) \| g \| / \| B \| \tag{23}$$

Proof : From (20b) we obtain

$$| \ \| Jd \| - \| f \| \ | \leq \| Jd + f \| \leq \bar{\omega}_0 \| f \|$$

since $\omega \leq \bar{\omega}_0$. Therefore either

$$\| Jd \| \geq \| f \|$$

or

$$\| Jd \| < \| f \| \quad \text{and} \quad \| f \| - \| Jd \| \leq \bar{\omega}_0 \| f \|$$

holds. This together gives

$$\| Jd \| \geq (1 - \bar{\omega}_0) \| f \|$$

since $\bar{\omega}_0 < 1$. Therefore

$$\| J \| \| d \| \geq \| Jd \| \geq (1 - \bar{\omega}_0) \| f \|$$

which implies

$$\begin{aligned} \| d \| &\geq (1 - \bar{\omega}_0) \| f \| / \| J \| = (1 - \bar{\omega}_0) \| J \| \| f \| / \| J \|^2 \\ &\geq (1 - \bar{\omega}_0) \| J^T f \| / \| J^T J \| = (1 - \bar{\omega}_0) \| g \| / \| B \| \end{aligned}$$

and (23) is proved. □

The inequality (23), which is also a consequence of (8b), is important for proving global convergence of the trust region method (see Ref.1).

Using Lemma 2.1 and Lemma 2.2 we can reformulate (D1)-(D3) as follows

(D1') There exists an index $k \in N$, such that

$$\| Jd_k + f \| \leq \omega \| f \| \tag{24}$$

for a given $0 \leq \omega < 1$. Note that this assumption requires that breakdown not occur (as we will see later).

(D2') The sequence $\| Jd_i + f \|$, $1 \leq i \leq k$, is nonincreasing, i.e.

$$\| Jd_{i+1} + f \| \leq \| Jd_i + f \| \tag{25}$$

for $1 \leq i < k$.

(D3') It holds that

$$\| Jd_1 + f \| - \| f \| \leq -2\, \bar{\epsilon}_1 \| f \| \tag{26}$$

and

$$\| J\lambda d_1 + f \| - \| f \| \leq -2\, \bar{\epsilon}_2 \| J \| \| \lambda d_1 \| \tag{27}$$

for $0 < \lambda \leq 1$.

The resulting vector $d \in R^n$ is again obtained by (19) so that (D1')-(D3') implies (20)-(21) with $\bar{\epsilon}_0 = \min\ (\bar{\epsilon}_1, \bar{\epsilon}_2)$. This together with Lemma 2.1 and Lemma 2.2 guarantees global convergence of the trust region method under the standard weak assumptions (see Ref.1).

The equation $Jd + f = 0$ can be solved by many iterative methods. Especially advantageous for our purpose are methods with short recurrences based on the unsymmetric Lanczos process. We focus our attention to the so-called transpose free methods since they allow us to easily compute

$$Jv \approx \frac{f(x + \delta v) - f(x)}{\delta \| v \|} \tag{28}$$

for an arbitrary vector v (δ is a small difference). The first discovered and most simple method of this type is the conjugate gradient squared (CGS) algorithm introduced in Ref.4 which is represented by the following iterative process.

$$\tilde{d}_0 = 0, \ d_0 = 0, \ \tilde{r}_0 = -f, \ r_0 = -f \tag{29a}$$

$$p_0 = 0, \ q_0 = 0, \ \beta_0 = 0, \ g_0 = rmarbitrary \tag{29b}$$

and

$$u_i = \tilde{r}_{i-1} + \beta_{i-1} q_{i-1} \tag{29c}$$

$$p_i = u_i + \beta_{i-1}(q_{i-1} + \beta_{i-1} p_{i-1}) \tag{29d}$$

$$v_i = J\ p_i \tag{29e}$$

$$\alpha_i = g_0^T \tilde{r}_{i-1} / g_0^T v_i \tag{29f}$$

$$q_i = u_i - \alpha_i v_i \tag{29g}$$

$$\tilde{d}_i = \tilde{d}_{i-1} + \alpha_i(u_i + q_i) \tag{29h}$$

$$\tilde{r}_i = \tilde{r}_{i-1} - \alpha_i J(u_i + q_i) \tag{29i}$$

$$\beta_i = g_0^T \tilde{r}_i / g_0^T \tilde{r}_{i-1} \tag{29j}$$

for $i \in N$. Note that $\tilde{r}_i = -(J\tilde{d}_i + f)$ for $i \in N$. In Ref.4 it was proved that the CGS algorithm terminates in at most n steps with $\tilde{r}_n = -(J\tilde{d}_n + f) = 0$ if the computations are exact and if division by zero (breakdown) does not occur. Therefore (D1') with $d_k = \tilde{d}_k$ is satisfied for the CGS algorithm if breakdown does not occur.

The main disadvantage of the CGS algorithm is the fact that the sequence $\| \tilde{r}_i \| = \| J\tilde{d}_i + f \|$, $1 \le i \le n$ is not nonincreasing. Therefore the CGS algorithm has to be smoothed. We use the quasi-minimized CGS algorithm (QCGS) described in Ref.5. The QCGS algorithm differs from the CGS algorithm in that it uses two additional recurrences

$$d_i = \tilde{d}_i + \mu_i(d_{i-1} - \tilde{d}_i) - \nu_i p_i \tag{29k}$$

$$r_i = \tilde{r}_i + \mu_i(r_{i-1} - \tilde{r}_i) + \nu_i v_i \tag{29l}$$

where μ_i and ν_i are chosen to minimize $\| r_i \|$ (again $r_i = -(Jd_i + f)$). From this minimization property we obtain $\| r_i \| \le \| \tilde{r}_i \|$ (if we set $\mu_i = 0$ and $\nu_i = 0$) and $\| r_i \| \le \| r_{i-1} \|$ (if we set $\mu_i = 1$ and $\nu_i = 0$). Therefore (D1') and (D2') are satisfied for the QCGS algorithm if the breakdown does not occur. It remains to formulate conditions that guarantee the assumption (D3').

Lemma 2.3 Let r_1 be generated by the QCGS algorithm. Then

$$\| r_1 \|^2 \le \left(1 - \frac{(r_0^T v_1)^2}{\| r_0 \|^2 \| v_1 \|^2}\right) \| f \|^2 \tag{30}$$

Proof: Let $\hat{r}_1 = r_0 + \nu_1 v_1$ where ν_1 is chosen to minimize $\| \hat{r}_1 \|$. Since

$$\| \hat{r}_1 \|^2 = \| r_0 \|^2 + 2\, \nu_1 r_0^T v_1 + \nu_1^2 \| v_1 \|^2 \tag{31}$$

we get

$$\partial \| \hat{r}_1 \|^2 / \partial \nu_1 = 2\, r_0^T v_1 + 2\, \nu_1 \| v_1 \|^2$$

so that $\| \hat{r}_1 \|^2$ (and consequently $\| \hat{r}_1 \|$) is minimal for

$$\nu_1 = -r_0^T v_1 / \| v_1 \|^2$$

If we substitute this value into (31), we obtain

$$\| \hat{r}_1 \|^2 = \| r_0 \|^2 - \frac{(r_0^T v_1)^2}{\| v_1 \|^2} = (1 - \frac{(r_0^T v_1)^2}{\| r_0 \|^2 \| v_1 \|^2}) \| f \|^2$$

since $\| r_0 \| = \| f \|$. But $\| r_1 \| \leq \| \hat{r}_1 \|$ from the minimization property of $\| r_1 \|$ so that (30) holds. □

Corollary 2.1 Let r_1 be generated by the QCGS algorithm and let

$$\cos \varphi_1 \triangleq r_0^T v_1 / (\| r_0 \| \| v_1 \|) \geq 2 \sqrt{\epsilon_1} \tag{32}$$

Then (26) holds.

Proof: From (30) we obtain

$$\| f \|^2 - \| r_1 \|^2 \geq \cos^2 \varphi_1 \| f \|^2$$

so that

$$\begin{aligned} 2 \| f \| (\| f \| - \| r_1 \|) &\geq (\| f \| + \| r_1 \|)(\| f \| - \| r_1 \|) \\ &= \| f \|^2 - \| r_1 \|^2 \geq \cos^2 \varphi_1 \| f \|^2 \end{aligned}$$

which using (32) gives

$$\| f \| - \| r_1 \| \geq (1/2) \cos^2 \varphi_1 \| f \| = 2\bar{\epsilon}_1 \| f \|$$

and (26) is proved since $\| r_1 \| = \| J d_1 + f \|$. □

Now we derive an explicit expression for the coefficients μ_i and ν_i in (29e). Denote $V_i = [r_{i-1} - \tilde{r}_i, v_i]$ and $c_i = [\mu_i, \nu_i]^T$. Then from (29l) we obtain

$$r_i = \tilde{r}_i + V_i c_i \tag{33}$$

since

$$\| r_i \|^2 = \tilde{r}_i^T \tilde{r}_i + 2 \tilde{r}_i^T V_i c_i + c_i^T V_i^T V_i c_i$$

we get

$$\partial \| r_i \|^2 / \partial c_i = 2 V_i^T \tilde{r}_i + 2 V_i^T V_i c_i$$

so that $\| r_i \|^2$ (and consequently $\| r_i \|$) is minimal for

$$c_i = -(V_i^T V_i)^{-1} V_i^T \tilde{r}_i \tag{34}$$

If we substitute this expression into (33), we obtain

$$r_i = \tilde{r}_i - V_i (V_i^T V_i)^{-1} V_i^T \tilde{r}_i = P_i \tilde{r}_i \tag{35}$$

where $P_i = I - V_i(V_i^T V_i)^{-1} V_i^T$ is an orthogonal projection matrix (symmetric and idempotent) which projects $\tilde{r}_i$ into the subspace orthogonal to the vectors $r_{i-1} - \tilde{r}_i$ and v_i so that $P_i(r_{i-1} - \tilde{r}_i) = 0$. This fact and (35) imply

$$r_i = P_i \tilde{r}_i = P_i r_{i-1} \tag{36}$$

Now we use (36) for proving (27) in the assumption (D3').

Lemma 2.4 Let r_1 be generated by the QCGS algorithm, let (26) hold and let

$$\kappa(J) \leq \bar{\epsilon}_1 / (2\, \bar{\epsilon}_2) \tag{37}$$

where $\kappa(J) = \| J \| \| J^{-1} \|$ is the condition number of the matrix J. Then (27) holds for $0 < \lambda \leq 1$.

Proof: The equalities in (29) imply $d_0 = 0$ and $r_0 = -f$. Using (36) we obtain

$$r_1 = P_1 r_0 = -P_1 f$$

so that

$$d_1 = d_1 - d_0 = J^{-1}(r_1 - r_0) = J^{-1}(f - P_1 f)$$

Therefore

$$\| d_1 \| \leq \| J^{-1} \| \| I - P_1 \| \| f \| = \| J^{-1} \| \| f \|$$

since the idempotent matrix $I - P_1$ has unit norm. This implies

$$\| J \| \| d_1 \| \leq \kappa(J) \| f \| \tag{38}$$

where $\kappa(J) = \| J \| \| J^{-1} \|$ is the condition number of the matrix J. Using (26) and (38) we get

$$\| f \| - \| J d_1 + f \| \geq 2\, \bar{\epsilon}_1 \| f \| \geq \frac{2\, \bar{\epsilon}_1}{\kappa(J)} \| J \| \| d_1 \| \tag{39}$$

If $0 < \lambda \leq 1$ then $0 < \lambda^2 \leq \lambda$ so that

$$\begin{aligned}
\| f \|^2 - \| J(\lambda d_1) + f \|^2 &= f^T f - \lambda^2 d_1^T J^T J d_1 - 2\lambda f^T J d_1 - f^T f \\
&= -\lambda^2 d_1^T J^T J d_1 - 2\lambda f^T J d_1 \\
&\geq -\lambda d_1^T J^T J d_1 - 2\lambda f^T J d_1 \\
&= \lambda(\| f \|^2 - \| J d_1 + f \|^2)
\end{aligned}$$

Therefore

$$\begin{aligned}2 \| f \| (\| f \| - \| J(\lambda d_1) + f \|) &\geq \| f \|^2 - \| J(\lambda d_1) + f \|^2 \\ &\geq \lambda(\| f \|^2 - \| Jd_1 + f \|^2) \\ &\geq \lambda \| f \| (\| f \| - \| Jd_1 + f \|)\end{aligned}$$

which together with (39) gives

$$\begin{aligned}\| f \| - \| J(\lambda d_1) + f \| &\geq (\lambda/2)(\| f \| - \| Jd_1 + f \|) \\ &\geq \frac{\lambda \bar{\epsilon}_1}{\kappa(J)} \| J \| \| d_1 \| \\ &= \frac{\bar{\epsilon}_1}{\kappa(J)} \| J \| \| \lambda d_1 \|\end{aligned}$$

Using (37) we then obtain (27). □

Now we are in a position to describe the complete inexact trust region method based on the smoothed CGS algorithm and prove its global convergence.

Algorithm 2.1

Data : $0 < \bar{\beta}_1 < \bar{\beta}_2 < 1 < \bar{\gamma}_1 < \bar{\gamma}_2$, $0 < \bar{\rho}_1 < \bar{\rho}_2 < 1$, $0 < \bar{\tau}_0 < 1$, $0 < \bar{\omega}_0 < 1$, $0 < \bar{\Delta}$, $0 < \bar{\epsilon}$, $\bar{k} \in N$, $\bar{l} \in N$.

Step 1 : Choose an initial point $x \in R^n$. Compute the values $f_j := f_j(x)$ of the functions $f_j : R^n \to R$, $1 \leq j \leq n$, at the point $x \in R^n$ and, consequently, the vector $f := f(x)$. Compute the value $F := F(x)$ of the objective function $F : R^n \to R$ by (2). Set $\Delta := 0$, $\tau := (\bar{\tau}_0)^{1/n}$. Set $k := 1$.

Step 2 : If $F \leq \bar{\epsilon}$ then stop. Otherwise compute the gradients $g_j := g_j(x)$ of the functions $f_j : R^n \to R$, $1 \leq j \leq n$, at the point $x \in R^n$ (by numerical differentiation) and, consequently, the Jacobian matrix $J := J(x)$. Compute the gradient $g := g(x)$ of the objective function $F : R^n \to R$ by (5). Set $l := 1$.

Step 3 : If $\Delta = 0$ then set $\Delta := \min(\| g \|^3 / \| Jg \|^2, 4F/ \| g \|, \bar{\Delta})$. Set $\omega := \min(\sqrt{\| f \|}, \tau^k, \bar{\omega}_0)$. Compute the vector $d \in R^n$ by the following subalgorithm :

Step 3a : Set $d := 0$, $\tilde{d} := 0$, $r := -f$, $\tilde{r} := -f$, $p := 0$, $q := 0$ and $\sigma := 1$. Set $i := 1$.

Step 3b : Set $\tilde{\sigma} := \sigma$ and compute $\sigma := g^T \tilde{r}$. Set $\beta := \sigma/\tilde{\sigma}$ and compute $u := \tilde{r} + \beta q$, $p := u + \beta(q + \beta p)$, $v := Jp$. Set $\alpha := \sigma/g^T v$ and compute $q := u - \alpha v$, $\tilde{d} := \tilde{d} + \alpha(u + q)$, $\tilde{r} := \tilde{r} - \alpha J(u + q)$. Set $V := [r - \tilde{r}, v]$ and compute $c := -(V^T V + D)^{-1} V^T \tilde{r}$ where D is a small diagonal matrix (usually zero) which serves to eliminate possible singularity of the matrix $V^T V$.

Step 3c : Compute $s := (c_1 - 1)(d - \tilde{d}) - c_2 p$ where c_1 and c_2 are elements of the vector $c \in R^2$. If $\| d + s \| > \Delta$ then determine $0 < \lambda < 1$ so that $\| d + \lambda s \| = \Delta$, set $d := d + \lambda s$ and go to Step 4. Otherwise set $d := d + s$ and compute $r := \tilde{r} + c_1(r - \tilde{r}) + c_2 v$.

Step 3d : If either $i = 2n$ or $\| r \| \leq \omega \| f \|$ then go to Step 4, otherwise set $i := i + 1$ and go to Step 3b.

Step 4 : Set $x^+ := x + d$. Compute the values $f_j^+ := f_j(x^+)$ of the functions $f_j : R^n \to R$, $1 \leq j \leq n$, at the point $x^+ \in R^n$ and, consequently, the vector $f^+ := f(x^+)$. Compute the value $F^+ := F(x^+)$ of the objective function $F : R^n \to R$ by (2). Compute the value $Q(d) = (\| Jd + f \|^2 - \| f \|^2)/2$ and set $\rho := (F^+ - F)/Q(d)$. When $\rho < \bar{\rho}_1$ then compute $\alpha := (F^+ - F)/f^T Jd$, $\beta := 1/(2(1 - \alpha))$ and set $\Delta := \bar{\beta}_1 \| d \|$ if $\beta < \bar{\beta}_1$, $\Delta := \beta \| d \|$ if $\bar{\beta}_1 \leq \beta \leq \bar{\beta}_2$, $\Delta := \bar{\beta}_2 \| d \|$ if $\bar{\beta}_2 < \beta$. When $\bar{\rho}_1 \leq \rho \leq \bar{\rho}_2$ then set $\Delta := \min(\Delta, \bar{\gamma}_2 \| d \|)$. When $\bar{\rho}_2 < \rho$ then compute $\Delta := \max(\Delta, \bar{\gamma}_1 \| d \|)$ and set $\Delta := \min(\Delta, \bar{\gamma}_2 \| d \|, \bar{\Delta})$.

Step 5 : If $\rho \leq 0$ and $l \geq \bar{l}$ then stop (too many reductions). If $\rho \leq 0$ and $l < \bar{l}$ then set $l := l + 1$ and go to Step 3. If $\rho > 0$ and $k \geq \bar{k}$ then stop (too many iterations). If $\rho > 0$ and $k < \bar{k}$ then set $x := x^+$, $f := f^+$, $F := F^+$, set $k := k + 1$ and go to Step 2.

The maximum number of iterations $\bar{k} \in N$ serves as an alternative termination criterion in the case when the convergence is too slow. The maximum number of reductions $\bar{l} \in N$ serves as a safeguard against a possible infinite cycle.

There are two possibilities when Algorithm 2.1 can fail. If $\tilde{\sigma} = 0$ or $g^T v = 0$ in Step 3b then division by zero (breakdown) prevents continued computations. We have not treated this situation since it did not appear in any of our test examples. The matrix $V^T V + D$ is used in Step 3b to remove the situation when $V^T V$ is singular. The technique for its construction is the same as in Ref.6.

We suppose in the subsequent considerations that all computations were performed accurately and that $\bar{k} = \bar{l} = \infty$. We use the following assumption on functions $f_j : R^n \to R$, $1 \leq j \leq n$.

(A) The functions $f_j : R^n \to R$, $1 \leq j \leq n$, have continuous second-order derivatives and there exist constants $C_1 > 0$, $C_2 > 0$, $C_3 > 0$ such that $| f_j(x) | \leq C_1$, $\| g_j(x) \| \leq C_2$, $\| G_j(x) \| \leq C_3$, $1 \leq j \leq n$, for all $x \in R^n$.

This assumption is relatively strong. Apparently, it could be weakened, but, for our purposes, it is quite convenient.

Theorem 2.1 Let Assumption (A) be satisfied. Let $x_k \in R^n$, $k \in N$, be the sequence generated by Algorithm 2.1, where breakdown does not occur. Let there exist constants $C_4 > 0$ and $C_5 > 0$ such that

$$\frac{\| f(x_k) \| \| J(x_k)f(x_k) \|}{f^T(x_k)J(x_k)f(x_k)} \le C_4 \tag{40}$$

$$\kappa(J(x_k)) \le C_5 \tag{41}$$

for $k \in N$. Then

$$\liminf_{k\to\infty} \| f(x_k) \| = 0 \tag{42}$$

Proof : From the definition of the Jacobian matrix we have

$$\begin{aligned} \| J^T(x)J(x) \| &\le \sum_{j=1}^{n} \| g_j(x)g_j^T(x) \| \\ &= \sum_{j=1}^{n} \| g_j(x) \|^2 \\ &\le nC_2^2 \end{aligned}$$

and (6) implies

$$\begin{aligned} \| G(x) \| &\le \| J^T(x)J(x) \| + \| \sum_{j=1}^{n} f_j(x)G_j(x) \| \\ &\le nC_2^2 + \sum_{j=1}^{n} | f_j(x) | \| G_j(x) \| \\ &\le n(C_2^2 + C_1C_3) \end{aligned}$$

Therefore both matrices $B(x) = J^T(x)J(x)$ and $G(x)$ are bounded from above. The conditions (D1') and (D2') are satisfied from the nature of the QCGS algorithm and from the fact that breakdown does not occur. Using (40) and Corollary 2.1 we get (26) with $\bar{\epsilon}_1 = 1/(4C_4^2)$. Using (41) and Lemma 2.4 we obtain (27) with $\bar{\epsilon}_2 = 1/(8C_4^2C_5)$. Therefore the condition (D3') is also satisfied. This together implies (20) and (21) that have the same significance as (8) and (9). Thus, Algorithm 2.1 is exactly the trust region method (T1)-(T3) described in Section 1. Therefore, since both matrices $B(x) = J^T(x)J(x)$ and $G(x)$ are bounded from above, we can apply the proof of global convergence proposed in Ref.1. $\Box$

3. Computational Experiments

In this section we present results of a comparative study of three trust region methods for large sparse systems of nonlinear equations. The first method, which we denote QCGS1, is represented by Algorithm 2.1. This algorithm contains several parameters. We have used the values $\bar{\beta}_1 = 0.05$, $\bar{\beta}_2 = 0.75$, $\bar{\gamma}_1 = 2$, $\bar{\gamma}_2 = 10^6$, $\bar{\rho}_1 = 0.1$, $\bar{\rho}_2 = 0.9$, $\bar{\tau}_0 = 10^{-3}$, $\bar{\omega}_0 = 0.4$, $\bar{\Delta} = 10^3$, $\bar{\epsilon} = 10^{-16}$, $\bar{k} = 1000$, $\bar{l} = 20$ in all numerical experiments. The derivatives (elements of the Jacobian matrix) are computed by the formula

$$J_{ji} = \frac{f_j(x + \delta e_i) - f_j(x)}{\delta} \tag{43}$$

where e_i is i-th column of the unit matrix and $\delta = 10^{-8}$. If the Jacobian matrix is sparse then only the nonzero elements are computed by formula (43).

The second method, which we denote QCGS2, is a modification of the previous one. Instead of computing the Jacobian matrix, we use formula (28) to compute of the vectors Jp and $J(u+q)$ in Step 3b of Algorithm 2.1. The quadratic function $Q(d)$ in Step 4 of Algorithm 2.1 is computed using the formula

$$Jd = \frac{f(x + \delta d) - f(x)}{\delta \parallel d \parallel}$$

where again $\delta = 10^{-8}$. Note that this method uses no matrices (it uses only several n-dimensional vectors).

The third method, which we denote CGLS, is derived from the conjugate gradient method applied to the system $J^T Jd + J^T f = 0$ (see Refs. 7-8). This method uses the following iterative process:

$$d_0 = 0, \ \ r_0 = -f \tag{44a}$$

$$v_1 = J^T r_0, \ \ \gamma_1 = \parallel v_1 \parallel^2 \tag{44b}$$

$$p_1 = v_1 \tag{44c}$$

and

$$u_i = Jp_i, \ \ \delta_i = \parallel u_i \parallel^2 \tag{44d}$$

$$d_i = d_{i-1} + (\gamma_i/\delta_i)p_i, \ \ r_i = r_{i-1} - (\gamma_i/\delta_i)u_i \tag{44e}$$

$$v_{i+1} = J^T r_i, \ \ \gamma_{i+1} = \parallel v_{i+1} \parallel^2 \tag{44f}$$

$$p_{i+1} = v_{i+1} + (\gamma_{i+1}/\gamma_i)p_i \tag{44g}$$

for $i \in N$, instead of (29). The other parts of Algorithm 2.1 remain unchanged.

All test results were obtained by means of the 17 problems given in the Appendix. All these problems were considered with 100 variables. Therefore sparse Jacobian matrices were used. A summary of the results for all problems is given in tables 1-3.

Table 1. Results of experiments for QCGS1 algorithm ($n = 100$)

Problem	IT	IF	P
4.1	11	55	−21
4.2	142	443	−24
4.3	3	19	−19
4.4	8	33	−19
4.5	97	509	−17
4.6	16	64	−16
4.7	51	216	−17
4.8	17	103	−17
4.9	17	135	−23
4.10	7	62	−22
4.11	16	42	−20
4.12	17	52	−16
4.13	20	57	−16
4.14	7	28	−19
4.15	8	63	−18
4.16	14	57	−23
4.17	6	24	−17
$\sum$	457	1962	
Time		0:50.04	

Table 2. Results of experiments for QCGS2 algorithm ($n = 100$)

Problem	IT	IF	P
4.1	11	355	−19
4.2	173	823	−21
4.3	3	13	−19
4.4	8	47	−19
4.5	105	1373	−19
4.6	16	117	−16
4.7	65	817	−20
4.8	17	155	−16
4.9	17	121	−22
4.10	7	55	−22
4.11	17	73	−26
4.12	21	739	−14
4.13	20	203	−16
4.14	7	51	−19
4.15	8	59	−18
4.16	13	1063	−16
4.17	6	35	−17
$\sum$	514	6099	
Time		1:09.05	

Table 3. Results of experiments for CGLS algorithm ($n = 100$)

Problem	IT	IF	P
4.1	21	106	−17
4.2	317	955	−11
4.3	3	19	−16
4.4	24	133	+01
4.5	400	2000	+03
4.6	18	72	−16
4.7	95	383	−23
4.8	334	2003	+00
4.9	37	324	+00
4.10	16	142	−22
4.11	42	109	−21
4.12	24	74	−16
4.13	25	71	−16
4.14	11	44	−16
4.15	10	79	−21
4.16	503	2002	−08
4.17	10	40	−20
$\sum$	1890	8556	
Time		7:57.47	

Rows of these tables correspond to individual problems and columns contain numbers of iterations denoted IT, numbers of objective function evaluations denoted IF and the logarithms of the final values of the objective function denoted P.

Tables 1-3 show that the QCGS1 algorithm is much better, measured in both the number of function evaluations and the number of successfully solved problems, than the CGLS algorithm which is frequently used for nonlinear least squares. The CGLS algorithm found a wrong local minimum of the function (2) in the case of problems 4 and 9. Also in the case of problems 5, 8 and 16 the CGLS algorithm probably converged to a wrong local minimum of the function (2) but 2000 function evaluations did not suffice.

The QCGS2 algorithm is slightly worse, measured in the number of function evaluations, than the QCGS1 algorithm. In the other hand the QCGS2 algorithm does not use matrices so it is very convenient for large dense problems.

4. Appendix

Our test problems consist of searching for a solution to the system of nonlinear equations

$$f_k(x) = 0, \quad 1 \le k \le n$$

We begin at the starting point $\overline{x}$. We suppose n is even and use functions the *div* (integer division) and *mod* (remainder after integer division).

Problem 4.1 Countercurrent reactors problem 1 (Ref.9).

$$\alpha = 1/2$$

$$\begin{aligned}
f_k(x) &= \alpha - (1-\alpha)x_{k+2} - x_k(1+4x_{k+1}) && , \ k=1 \\
f_k(x) &= -(2-\alpha)x_{k+2} - x_k(1+4x_{k-1}) && , \ k=2 \\
f_k(x) &= \alpha x_{k-2} - (1-\alpha)x_{k+2} - x_k(1+4x_{k+1}) && , \ mod\,(k,2)=1\ , \ 2<k<n-1 \\
f_k(x) &= \alpha x_{k-2} - (2-\alpha)x_{k+2} - x_k(1+4x_{k-1}) && , \ mod\,(k,2)=0\ , \ 2<k<n-1 \\
f_k(x) &= \alpha x_{k-2} - x_k(1+4x_{k+1}) && , \ k=n-1 \\
f_k(x) &= \alpha x_{k-2} - (2-\alpha) - x_k(1+4x_{k-1}) && , \ k=n
\end{aligned}$$

$$\begin{aligned}
\overline{x}_l &= 0.1 && , \ mod\,(l,8)=1 \\
\overline{x}_l &= 0.2 && , \ mod\,(l,8)=2 \text{ or } mod\,(l,8)=0 \\
\overline{x}_l &= 0.3 && , \ mod\,(l,8)=3 \text{ or } mod\,(l,8)=7 \\
\overline{x}_l &= 0.4 && , \ mod\,(l,8)=4 \text{ or } mod\,(l,8)=6 \\
\overline{x}_l &= 0.5 && , \ mod\,(l,8)=5
\end{aligned}$$

Problem 4.2 Extended Powell badly scaled function (Ref.13).

$$\begin{aligned}
f_k(x) &= 10000\ x_k\ x_{k+1} - 1 && , \ mod\,(k,2)=1 \\
f_k(x) &= exp(-x_{k-1}) + exp(-x_k) - 1.0001 && , \ mod\,(k,2)=2
\end{aligned}$$

$$\begin{aligned}\overline{x}_l &= 0 \quad , \ mod\ (l,2)=1\\ \overline{x}_l &= 1 \quad , \ mod\ (l,2)=0\end{aligned}$$

Problem 4.3 A trigonometric system (Ref.10).

$$i = div\ (k-1,5)$$

$$\begin{aligned}f_k(x) &= 5-(i+1)(1-\cos x_k)-\sin x_k - \\ &- \sum_{j=5i+1}^{5i+5} \cos x_j\end{aligned}$$

$$\overline{x_l} = 1/n \qquad l \geq 1$$

Problem 4.4 A trigonometric - exponential system (trigexp 1) (Ref.10).

$$\begin{aligned}f_k(x) &= 3x_k^3 + 2x_{k+1} - 5 + \\ &\quad + \sin(x_k - x_{k+1})\sin(x_k + x_{k+1})\ , && k = 1\\ f_k(x) &= 3x_k^3 + 2x_{k+1} - 5 + \\ &\quad + \sin(x_k - x_{k+1})\sin(x_k + x_{k+1}) + \\ &\quad +4x_k - x_{k-1}\exp(x_{k-1} - x_k) - 3\ , && 1 < k < n\\ f_k(x) &= 4x_k - x_{k-1}\exp(x_{k-1} - x_k) - 3\ , && k = n\end{aligned}$$

$$\overline{x}_l = 0 \qquad l \geq 1$$

Problem 4.5 A trigonometric - exponential system (trigexp 2) (Ref.10).

$$\begin{aligned}f_k(x) &= 3(x_k - x_{k+2})^3 - 5 + 2x_{k+1} + \\ &\quad + \sin(x_k - x_{k+1} - x_{k+2})\sin(x_k + x_{k+1} - x_{k+2})\ \ , \ mod\ (k,2) = 1,\ k = 1\\ f_k(x) &= -6(x_{k-2} - x_k)^3 + 10 - 4x_{k-1} - \\ &\quad -2\sin(x_{k-2} - x_{k-1} - x_k)\sin(x_{k-2} + x_{k-1} - x_k) + \\ &\quad +3(x_k - x_{k+2})^3 - 5 + 2x_{k+1} + \\ &\quad + \sin(x_k - x_{k+1} - x_{k+2})\sin(x_k + x_{k+1} - x_{k+2})\ \ , \ mod\ (k,2) = 1,\ 1 < k < n\\ f_k(x) &= -6(x_{k-2} - x_k)^3 + 10 - 4x_{k-1} - \\ &\quad -2\sin(x_{k-2} - x_{k-1} - x_k)\sin(x_{k-2} + x_{k-1} - x_k)\ , \ mod\ (k,2) = 1,\ k = n\\ f_k(x) &= 4x_k - (x_{k-1} - x_{k+1})\exp(x_{k-1} - x_k - x_{k+1}) - 3\ \ , \ mod\ (k,2) = 0\end{aligned}$$

$$\overline{x}_l = 1, \qquad l \geq 1$$

Problem 4.6 Singular Broyden problem (Ref.11).

$$\begin{array}{rcll} f_k(x) &=& ((3-2x_k)x_k - 2x_{k+1} + 1)^2 & , \ k=1 \\ f_k(x) &=& ((3-2x_k)x_k - x_{k-1} - 2x_{k+1} + 1)^2 & , \ 1<k<n \\ f_k(x) &=& ((3-2x_k)x_k - x_{k-1} + 1)^2 & , \ k=n \end{array}$$

$$\overline{x}_l = -1, \quad l \geq 1$$

Problem 4.7 Tridiagonal system (Ref.12).

$$\begin{array}{rcll} f_k(x) &=& 4(x_k - x_{k+1}^2) & , \ k=1 \\ f_k(x) &=& 8x_k(x_k^2 - x_{k-1}) - 2(1-x_k) + 4(x_k - x_{k+1}^2) & , \ 1<k<n \\ f_k(x) &=& 8x_k(x_k^2 - x_{k-1}) - 2(1-x_k) & , \ k=n \end{array}$$

$$\overline{x}_l = 12, \quad l \geq 1$$

Problem 4.8 Five-diagonal system (Ref.12).

$$\begin{array}{rcll} f_k(x) &=& 4(x_k - x_{k+1}^2) + x_{k+1} - x_{k+2}^2 & , \ k=1 \\ f_k(x) &=& 8x_k(x_k^2 - x_{k-1}) - 2(1-x_k) + & \\ && +4(x_k - x_{k+1}^2) + x_{k+1} - x_{k+2}^2 & , \ k=2 \\ f_k(x) &=& 8x_k(x_k^2 - x_{k-1}) - 2(1-x_k) + & \\ && +4(x_k - x_{k+1}^2) + x_{k-1}^2 - x_{k-2} + x_{k+1} - x_{k+2}^2 & , \ 2<k<n-1 \\ f_k(x) &=& 8x_k(x_k^2 - x_{k-1}) - 2(1-x_k) + & \\ && +4(x_k - x_{k+1}^2) + x_{k-1}^2 - x_{k-2} & , \ k=n-1 \\ f_k(x) &=& 8x_k(x_k^2 - x_{k-1}) - 2(1-x_k) + x_{k-1}^2 - x_{k-2} & , \ k=n \end{array}$$

$$\overline{x}_l = -2, \quad l \geq 1$$

Problem 4.9 Seven-diagonal system (Ref.12).

$$\begin{array}{rcll} f_k(x) &=& 4(x_k - x_{k+1}^2) + x_{k+1} - x_{k+2}^2 + x_{k+2} - x_{k+3}^2 & , \ k=1 \\ f_k(x) &=& 8x_k(x_k^2 - x_{k-1}) - 2(1-x_k) + & \\ && +4(x_k - x_{k+1}^2) + x_{k-1}^2 + x_{k+1} - x_{k+2}^2 + x_{k+2} - x_{k+3}^2 & , \ k=2 \\ f_k(x) &=& 8x_k(x_k^2 - x_{k-1}) - 2(1-x_k) + & \\ && +4(x_k - x_{k+1}^2) + x_{k-1}^2 - x_{k-2} + x_{k+1} - x_{k+2}^2 + & \\ && +x_{k-2}^2 + x_{k+2} - x_{k+3}^2 & , \ k=3 \\ f_k(x) &=& 8x_k(x_k^2 - x_{k-1}) - 2(1-x_k) + & \end{array}$$

$$\begin{aligned}
& +4(x_k - x_{k+1}^2) + x_{k-1}^2 - x_{k-2} + x_{k+1} - x_{k+2}^2 + \\
& +x_{k-2}^2 + x_{k+2} - x_{k-3} - x_{k+3} && , \ 3 < k < n-2 \\
f_k(x) &= 8x_k(x_k^2 - x_{k-1}) - 2(1 - x_k) + \\
& +4(x_k - x_{k+1}^2) + x_{k-1}^2 - x_{k-2} + x_{k+1} - x_{k+2}^2 + \\
& +x_{k-2}^2 + x_{k+2} - x_{k-3} && , \ k = n-2 \\
f_k(x) &= 8x_k(x_k^2 - x_{k-1}) - 2(1 - x_k) + \\
& +4(x_k - x_{k+1}^2) + x_{k-1}^2 - x_{k-2} + x_{k+1} + \\
& +x_{k-2}^2 - x_{k-3} && , \ k = n-1 \\
f_k(x) &= 8x_k(x_k^2 - x_{k-1}) - 2(1 - x_k) + x_{k-1}^2 - x_{k-2} + \\
& +x_{k-2}^2 - x_{k-3} && , \ k = n
\end{aligned}$$

$$\overline{x}_l = -3, \quad l \geq 1$$

Problem 4.10 Structured Jacobian problem (Ref.11).

$$\begin{aligned}
f_k(x) &= -2x_k^2 + 3x_k - 2x_{k+1} + 3x_{n-4} - x_{n-3} - \\
& -x_{n-2} + 0.5x_{n-1} - x_n + 1 && , \ k = 1 \\
f_k(x) &= -2x_k^2 + 3x_k - x_{k-1} - 2x_{k+1} + 3x_{n-4} - x_{n-3} - \\
& -x_{n-2} + 0.5x_{n-1} - x_n + 1 && , \ 1 < k < n \\
f_k(x) &= -2x_k^2 + 3x_k - x_{k-1} + 3x_{n-4} - x_{n-3} - \\
& -x_{n-2} + 0.5x_{n-1} - x_n + 1 && , \ k = n
\end{aligned}$$

$$\overline{x}_l = -1, \quad l \geq 1$$

Problem 4.11 Extended Rosenbrock function.

$$\begin{aligned}
f_k(x) &= 10(x_{k+1} - x_k^2) && , \ mod\ (k,2) = 1 \\
f_k(x) &= 1 - x_{k-1} && , \ mod\ (k,2) = 0 \\
\overline{x}_l &= -1.2 && , \ mod\ (l,2) = 1 \\
\overline{x}_l &= 1.0 && , \ mod\ (l,2) = 0
\end{aligned}$$

Problem 4.12 Extended Powell singular function.

$$\begin{aligned}
f_k(x) &= x_k + 10x_{k+1} && , \ mod\ (k,4) = 1 \\
f_k(x) &= \sqrt{5}(x_{k+1} - x_{k+2}) && , \ mod\ (k,4) = 2 \\
f_k(x) &= (x_{k-1} - 2x_k)^2 && , \ mod\ (k,4) = 3 \\
f_k(x) &= \sqrt{10}(x_{k-3} - x_k)^2 && , \ mod\ (k,4) = 0
\end{aligned}$$

$$\begin{array}{rcll} \overline{x}_l & = & 3 & , \; mod\,(l,4) = 1 \\ \overline{x}_l & = & -1 & , \; mod\,(l,4) = 2 \\ \overline{x}_l & = & 0 & , \; mod\,(l,4) = 3 \\ \overline{x}_l & = & 1 & , \; mod\,(l,4) = 0 \end{array}$$

Problem 4.13 Extended Cragg and Levy function.

$$\begin{array}{rcll} f_k(x) & = & (\exp(x_k) - x_{k+1})^2 & , \; mod\,(k,4) = 1 \\ f_k(x) & = & 10(x_k - x_{k+1})^3 & , \; mod\,(k,4) = 2 \\ f_k(x) & = & \tan^2(x_k - x_{k+1}) & , \; mod\,(k,4) = 3 \\ f_k(x) & = & x_k - 1 & , \; mod\,(k,4) = 0 \end{array}$$

$$\begin{array}{rcll} \overline{x}_l & = & 1 & , \; mod\,(l,4) = 1 \\ \overline{x}_l & = & 2 & , \; mod\,(l,4) \neq 1 \end{array}$$

Problem 4.14 Broyden tridiagonal function.

$$\begin{array}{rcll} f_k(x) & = & x_k(0.5x_k - 3) + 2x_{k+1} - 1 & , \; k = 1 \\ f_k(x) & = & x_k(0.5x_k - 3) + x_{k-1} + 2x_{k+1} - 1 & , \; 1 < k < n \\ f_k(x) & = & x_k(0.5x_k - 3) - 1 + x_{k-1} & , \; k = n \end{array}$$

$$\overline{x}_l = -1, \quad l \geq 1$$

Problem 4.15 Broyden banded problem (Ref.13).

$$k_1 = \max\,(1, k-5), \quad k_2 = \min\,(n, k+1)$$

$$f_k(x) = (2 + 5x_k^2)x_k + 1 + \sum_{i=k_1}^{k_2} x_i(1 + x_i)$$

$$\overline{x}_l = -1, \quad l \geq 1$$

Problem 4.16 Discrete boundary value problem (Ref.13).

$$h = 1/(n+1)$$

$$\begin{aligned}
f_k(x) &= 2x_k + h^2(x_k+1+hk)^3/2 - x_{k+1} &&, \ k=1\\
f_k(x) &= 2x_k + h^2(x_k+1+hk)^3/2 - x_{k-1} - x_{k+1} &&, \ 1<k<n\\
f_k(x) &= 2x_k + h^2(x_k+1+hk)^3/2 - x_{k-1} &&, \ k=n
\end{aligned}$$

$$\overline{x}_l = lh\,(lh-1), \quad l \geq 1$$

Problem 4.17 Broyden tridiagonal problem (Ref.13).

$$\begin{aligned}
f_k(x) &= (3-2x_k)x_k - 2x_{k+1} + 1 &&, \ k=1\\
f_k(x) &= (3-2x_k)x_k - x_{k-1} - 2x_{k+1} + 1 &&, \ 1<k<n\\
f_k(x) &= (3-2x_k)x_k - x_{k-1} + 1 &&, \ k=n
\end{aligned}$$

$$\overline{x}_l = -1, \quad l \geq 1$$

References

1. Powell, M.J.D., *On the Global Convergence of Trust Region Algorithms for Unconstrained Minimization,* Mathematical Programming, Vol. 29, pp. 297-303, 1984.

2. Shultz, G.A., and Schnabel, R.B., and Byrd, R.H., *A Family of Trust-Region-Based Algorithms for Unconstrained Minimization with Strong Global Convergence Properties,* SIAM Journal on Numerical Analysis, Vol 22, pp. 47-67, 1985.

3. Steihaug, T., *The Conjugate Gradient Method and Trust Regions in Large-Scale Optimization,* SIAM Journal on Numerical Analysis, Vol. 20, pp. 626-637, 1983.

4. Sonneveld, P., *CGS, a Fast Lanczos-Type Solver for Nonsymmetric Linear Systems,* SIAM Journal on Scientific and Statistical Computations, Vol 10, pp. 36-52, 1989.

5. Tong, C.H., *A Comparative Study of Preconditioned Lanczos Methods for Nonsymmetric Linear Systems,* Sandia National Laboratories, Sandia Report No. SAND91-8240B, Livermore, 1992.

6. Gill, P.E., and Murray, W., *Newton Type Methods for Unconstrained and Linearly Constrained Optimization,* Mathematical Programming, Vol.7, pp. 311-350, 1974.

7. Lukšan, L., *Inexact Trust Region Method for Large Sparse Nonlinear Least Squares,* Academy of Sciences of the Czech Republic, Institute of Computer and Information Sciences Report No. 501, Prague, 1991.

8. Paige, C.C., and Saunders, M.A., *LSQR: An Algorithm for Sparse Linear Equations and Sparse Least Squares,* ACM Transactions on Mathematical Software, Vol. 8, pp. 43-71, 1982.

9. Bogle, I.D.L., and Perkins, J.D., *A New Sparsity Preserving Quasi-Newton Update for Solving Nonlinear Equations,* SIAM Journal on Scientific and Statistical Computations, Vol 11, pp. 621-630, 1990.

10. Toint, P.L., *Numerical Solution of Large Sets of Algebraic Equations,* Mathematics of Computation, Vol. 46, pp. 175-189, 1986.

11. Gomez-Ruggiero, M.A., and Martinez, J.M., and Moretti, A.C., *Comparing Algorithms for Solving Sparse Nonlinear Systems of Equations,* SIAM Journal on Scientific and Statistical Computations, Vol. 13, pp. 459-483, 1992.

12. Li, G., *Successive Column Correction Algorithms for Solving Sparse Nonlinear Systems of Equations,* Mathematical Programming, Vol. 43, pp. 187-207, 1989.

13. Moré, J.J., and Garbow, B.S., and Hillström, K.E., *Testing Unconstrained Optimization Software,* ACM Transactions on Mathematical Software, Vol. 7, pp. 17-41, 1981.