JOURNAL OF OPTIMIZATION THEORY AND APPLICATIONS: Vol. 119, No. 1, pp. 65–82, October 2003 (©2003)

Use of the Minimum-Norm Search Direction in a Nonmonotone Version of the Gauss-Newton Method[1]

F. Lampariello[2] and M. Sciandrone[2]

Communicated by P. Tseng

Abstract. In this work, a new stabilization scheme for the Gauss-Newton method is defined, where the minimum norm solution of the linear least-squares problem is normally taken as search direction and the standard Gauss-Newton equation is suitably modified only at a subsequence of the iterates. Moreover, the stepsize is computed by means of a nonmonotone line search technique. The global convergence of the proposed algorithm model is proved under standard assumptions and the superlinear rate of convergence is ensured for the zero-residual case. A specific implementation algorithm is described, where the use of the pure Gauss-Newton iteration is conditioned to the progress made in the minimization process by controlling the stepsize. The results of a computational experimentation performed on a set of standard test problems are reported.

Key Words. Gauss-Newton method, nonlinear least-squares problems, minimum norm solution, nonmonotone line search techniques.

1. Introduction

We consider the following nonlinear least-squares problem:

$$\min_{x\in R^n} f(x)=(1/2)\|r(x)\|^2=(1/2)\sum_{i=1}^{m} r_i^2(x), \qquad m\geq n,$$

where each component r_i: $R^n\rightarrow R$ of the residual vector $r(x)$ is a continuously differentiable function. Problems of this kind arise in many practical

[1]The authors are indebted to the anonymous referees for useful suggestions.

[2]Researcher, Istituto di Analisi dei Sistemi ed Informatica, National Research Council, Rome, Italy.

applications involving, for instance, the solution of nonlinear equations or data fitting.

Let $J(x)$ be the Jacobian matrix of $r(x)$. Then, the gradient $\nabla f(x)$ and the Hessian matrix $\nabla^2 f(x)$ are given by

$$\nabla f(x) = J(x)^T r(x), \qquad \nabla^2 f(x) = J(x)^T J(x) + \sum_{i=1}^{m} r_i(x) \nabla^2 r_i(x).$$

Most algorithms for nonlinear least-squares exploit the particular structure of $\nabla^2 f(x)$. Since the first term $J(x)^T J(x)$ is often more important than the second-order term, the latter can be discarded. The Gauss-Newton method is a modification of the Newton method and consists of the iteration

$$x_{k+1} = x_k - [J(x_k)^T J(x_k)]^{-1} J(x_k)^T r(x_k),$$

provided that the matrix $J(x_k)^T J(x_k)$ is nonsingular.

As it is well-known, the method is only locally convergent to a stationary point x^*, assuming that $\nabla^2 f(x^*)$ is nonsingular and that the second-order term is small relative to the first one, whereas it can diverge from arbitrarily close points to x^* if the second-order term is too large.

The Ben-Israel method (Ref. 1) is a modification of the Gauss-Newton method designed for the case when $J(x)$ does not have full rank and consists of the iteration

$$x_{k+1} = x_k - J(x_k)^\dagger r(x_k),$$

where the symbol $\dagger$ denotes the pseudoinverse matrix. Its convergence has been proved under the assumption that $J(x)^\dagger$ is Lipschitz continuous. Moreover, in Ref. 2, using the Lyapunov stability theory, it has been shown that any limit point x^* of the sequence $\{x_k\}$ generated by the iteration

$$x_{k+1} = x_k - \alpha_k J(x_k)^\dagger r(x_k)$$

is a stationary point of f, provided that the Jacobian matrix is of constant rank in some bounded convex set Ω containing x^*, that the starting point x_o belongs to Ω, and that the stepsize α_k is bounded above by a number related to the Lipschitz constant of $J(x)^T J(x)$. Note that these assumptions are stringent as observed in Ref. 3, where for removing them, the construction of an auxiliary least-squares problem of higher dimension is proposed. It is shown that the new problem is a well-posed one, provided that the rank deficiency of the Jacobian is small. However, as remarked by the author, this technique seems to be useful only for small-dimensional problems and moreover it appears not easily implementable.

Alternatively to the use of the pseudoinverse, the stabilization strategies of the Gauss-Newton method are based on the iteration

$$x_{k+1} = x_k - \alpha_k [J(x_k)^T J(x_k) + D_k]^{-1} J(x_k)^T r(x_k),$$

where D_k is a diagonal matrix suitably chosen to ensure that the search directions are gradient related. The stepsize α_k is computed for instance by the Armijo rule, in such a way that the sequence $\{f(x_k)\}$ is monotonically decreasing.

We observe first that, to obtain a behavior as close as possible to that of the pure Gauss-Newton method, the matrix $J(x_k)^T J(x_k)$ could be modified by means of D_k only at intervals, instead of at every iteration. In this case, to get global convergence, in addition to ensuring the convergence of the sequence $\{f(x_k)\}$, we would have to guarantee also that $\|x_{k+1} - x_k\| \to 0$. Obviously, the line search must be performed along descent directions. However, as shown in many papers (see e.g. Refs. 4–6), the enforcement of the monotonic descent may cause inefficiency, especially when the objective function (i.e., the residual in our case) is highly nonlinear. Nonmonotone algorithms for nonlinear least-squares problems have been described in Refs. 7–8 that extend the approach of Refs. 4–5 proposed in the context of Newton-type methods.

In this work, we present a nonmonotone modified version of the Gauss-Newton method. In particular, we consider the direction

$$d_k^{(m)} = -J(x_k)^\dagger r(x_k)$$

of the Ben-Israel iteration, which is the minimum norm solution of the standard Gauss-Newton equation

$$J(x_k)^T J(x_k) d = -J(x_k)^T r(x_k),$$

and we exploit the property that it is a descent direction for f at x_k, even when the matrix $J(x_k)^T J(x_k)$ is singular; thus, it is normally taken as search direction. Moreover, the matrix $J(x_k)^T J(x_k)$ is possibly modified to ensure, together with the descent property, even an angle condition, only at a subsequence of points. Finally, in order to obtain the global convergence of the whole sequence $\{x_k\}$, a suitable line search technique for computing the stepsize is defined, which guarantees that $\|x_{k+1} - x_k\| \to 0$, and also the superlinear rate of convergence for the zero-residual case.

As regards the computation of the minimum norm solution $d_k^{(m)}$ of the Gauss-Newton equation, it is possible to use the Moore-Penrose generalized inverse (Ref. 9). However, this choice is not suitable when the problem dimension becomes large, so that it appears more convenient the use of an iterative method, instead of a direct computation. In particular, we observe that, among the infinitely many solutions of the Gauss-Newton equation, only $d_k^{(m)}$ belongs to $\mathcal{R}(J(x_k)^T)$, i.e., to the range or column space of $J(x_k)^T$. Therefore, the conjugate gradient method (CG, Ref. 10) starting from a point in $\mathcal{R}(J(x_k)^T)$, e.g. from the null vector, can be used for computing $d_k^{(m)}$. Indeed, such a method converges towards solutions of the Gauss-Newton equation, and since all the iterates remain in $\mathcal{R}(J(x_k)^T)$, the convergence to $d_k^{(m)}$ is ensured. Note that also the Barzilai and Borwein gradient method (BB, Ref. 11) can be used, although it converges only asymptotically. However, in some cases, it may be competitive with the CG method, as observed in Ref. 12.

In Section 2, we present a nonmonotone stabilization algorithm model for the Gauss-Newton method and in Section 3 we prove the convergence results under standard assumptions without any condition on the rank of the Jacobian matrix. In Section 4, we discuss some implementative aspects and we describe a specific algorithm, where the reliability of the pure Gauss-Newton iteration is evaluated according to the acceptance of the unit step-size. The numerical results obtained by solving a set of test problems are compared with those derived by applying an efficient standard routine (NAG library) devoted to nonlinear least-squares problems.

2. Modified Gauss-Newton Method

In the Gauss-Newton method, the search direction is determined by solving the linear least-squares problem

$$\min_d \|J(x_k)d + r(x_k)\|^2. \tag{1}$$

If $J(x_k)$ is of full column rank, the unique solution d_k of (1) is a descent direction for f and therefore it represents a suitable direction for a line search. If $J(x_k)$ is rank deficient, Eq. (1) admits infinitely many solutions.

Let

$$S = \{d \in R^n : J(x_k)^T J(x_k) d = -J(x_k)^T r(x_k)\}$$

be the set of the solutions of (1). We consider the problem

$$\min_{d \in S} \|d\|,$$

which admits a unique solution, referred to as the minimum norm solution of (1), i.e., the one given by

$$\begin{aligned} d_k^{(m)} &= -J(x_k)^\dagger r(x_k) \\ &= -[J(x_k)^T J(x_k)]^\dagger J(x_k)^T r(x_k). \end{aligned}$$

It is known that $d_k^{(m)}$ has the following property (Ref. 13, p. 343).

Proposition 2.1. Let x_k be a nonstationary point of f. Then $d_k^{(m)}$, the minimum norm solution of (1), is such that, for sufficiently small $\alpha > 0$, $\|r(x_k + \alpha d_k^{(m)})\| < \|r(x_k)\|$.

Therefore, $d_k^{(m)}$ is a descent direction for f and hence it represents a suitable choice for the search direction.

Our goal is to design an algorithm based on the Ben-Israel iteration

$$x_{k+1} = x_k + d_k^{(m)},$$

which can be interpreted as the pure form of the Gauss-Newton method. Obviously, to ensure the global convergence, it is necessary to introduce a suitable acceptance rule for the unit stepsize, i.e., to define a line search technique for the stepsize α_k along $d_k^{(m)}$, such that the unit stepsize is taken as frequently as possible. Moreover, we have to take into account that the descent property of the search direction is not sufficient to ensure the global convergence of an iterative scheme of the form

$$x_{k+1} = x_k + \alpha_k d_k, \tag{2}$$

so that it is necessary to impose, at least at a subsequence of points, a suitable condition on d_k. More specifically, if the columns of the Jacobian matrix were uniformly linearly independent, the unique solution d_k of (1) would be gradient related, so that the global convergence of the sequence $\{x_k\}$ would be ensured by using, for instance, an Armijo-type line search for computing α_k. Therefore, when the uniform full-rank condition is not fulfilled, it is necessary to modify the matrix $J(x_k)^T J(x_k)$ by adding to it a diagonal matrix D_k such that $J(x_k)^T J(x_k) + D_k$ is uniformly positive definite. For example, D_k may be chosen in accordance with the Cholesky factorization scheme or as a positive multiple of the identity matrix (Levenberg-Marquardt method). With these choices of D_k, also the direction computed by solving

$$[J(x_k)^T J(x_k) + D_k]\, d = -J(x_k)^T r(x_k) \tag{3}$$

turns out to be gradient related, thus allowing us to obtain the global convergence of $\{x_k\}$ by means of a suitable line search technique or, in the case of the Levenberg-Marquardt method, by adopting a trust-region strategy.

However, we observe that the solution of (3) could be significantly different from the minimum norm solution of (1), which we want to take as search direction. In the sequel, we show that the global convergence can be ensured by modifying the matrix $J(x_k)^T J(x_k)$ by means of D_k only at a subsequence of the iterates, provided that the difference between the iteration numbers of two consecutive iterates where the modification is performed, be bounded by a prefixed integer p.

It is evident that, when p is taken small, the possible advantage of using the pure Gauss-Newton iteration would be insufficiently exploited, particularly in a region around a small-residual solution, with respect to the case where $p = 1$, that corresponds to a classical implementation of the Gauss-Newton method. Hence, according to our intention, a relatively large value for p should be taken. However, we observe that the Gauss-Newton local quadratic model may not approximate adequately the objective function, mainly in regions far from a solution. This occurrence can be monitored on the basis of the progress obtained in the minimization process, measured for instance in terms of the objective function reduction, or the stepsize, or both (see e.g. the strategy of the adaptive algorithm described in Ref. 14). Therefore, it appears advisable to introduce a control criterion (CR) for deciding whether the matrix $J(x_k)^T J(x_k)$ will be modified or not. Thus, if CR is satisfied, the minimum norm solution of (1) is taken as search direction; otherwise, the latter is determined by solving (3). Therefore, the solution of (3) is used whenever CR is not fulfilled, and in any case after $p-1$ consecutive iterations performed using $d_k^{(m)}$.

We remark, however, that the introduction of a criterion CR is not necessary for getting global convergence, which can be ensured simply by using the solution of (3) every p iterations.

Moreover, we define a nonmonotone line search technique which allows us to accept the unit stepsize more frequently than a standard monotone one. In particular, the strategy consists in the enforcement of a descent for f with respect to the maximum value attained in a prefixed number M of preceding iterations (see e.g. Ref. 4). The acceptance rule of the stepsize is defined by a quadratic function in order to obtain, together with the global convergence, even the superlinear rate, at least for the case of zero-residual problems, as shown later.

Nonmonotone Line Search Algorithm (Algorithm NLS)

```
Data.   $\gamma \in (0, 1)$, $0 < \sigma_1 < \sigma_2 < 1$, integer $M > 1$.
Step 0. Set $\alpha = 1$.
Step 1. If
            $f(x_k + \alpha d_k) \le \max_{0 \le j \le \min(k, M)} [f(x_{k-j})] - \gamma \alpha^2 \|d_k\|^3,$    (4)
        set $\alpha_k = \alpha$ and stop.
Step 2. Choose $\sigma \in [\sigma_1, \sigma_2]$, set $\alpha = \sigma\alpha$ and go to Step 1.
```

Assuming that d_k is a descent direction for f at x_k, it can be shown easily that Algorithm NLS is well defined.

The following algorithmic scheme resumes the globalization strategy discussed above.

Nonmonotone Gauss-Newton Stabilization Algorithm (Algorithm NMGN)

```
Data.    $x_o \in R^n$, integer $p > 1$.
Step 0.  Set $k = 0$, $i = 1$.
Step 1.  Compute $J(x_k)$ and $\nabla f(x_k) = J(x_k)^T r(x_k)$; verify the stopping
         criterion.
Step 2.  If $i = 1$ or if $i < p$ and CR is satisfied, compute the minimum
         norm solution $d_k^{(m)}$ of (1); set $d_k = d_k^{(m)}$, $i = i + 1$, and go to
         Step 4.
Step 3.  Compute the solution of (3) (where possibly $D_k = 0$), take it
         as $d_k$, and set $i = 1$.
Step 4.  Compute the stepsize $\alpha_k$ by means of Algorithm NLS. Set
         $x_{k+1} = x_k + \alpha_k d_k$, $k = k + 1$, and go to Step 1.
```

Different algorithms can be defined depending on the criterion CR adopted, on the way of computing at Step 2 the minimum norm solution of (1), and on the way of modifying at Step 3 the matrix $J(x_k)^T J(x_k)$ by means of D_k.

3. Convergence Analysis

Let $K_p \subset \{0, 1, \dots\}$ be the subset of iterates where the solution of (3) is taken as search direction. In order to ensure the global convergence of an algorithm in the class defined above, we have to establish suitable conditions on the search directions d_k, computed for $k \in K_p$. Let us assume that the matrix D_k is such that $\nabla f(x_k)^T d_k < 0$ and that the following conditions on its minimum and maximum eigenvalues [$\lambda_m(D_k)$ and $\lambda_M(D_k)$] are fulfilled:

(a) there exists a constant $c > 0$ such that, for all $k \in K_p$,

$$0 \leq \lambda_m(D_k) \leq \lambda_M(D_k) \leq c \|\nabla f(x_k)\|;$$

(b) for every infinite subset $K \subseteq K_p$,

$$\lim_{k \to \infty,\, k \in K} \lambda_m(D_k) = 0 \text{ implies } \lim_{k \to \infty,\, k \in K} \|\nabla f(x_k)\| = 0.$$

These conditions ensure essentially that the subsequence $\{d_k\}_{k \in K_p}$ is gradient related to $\{x_k\}_{k \in K_p}$. To ensure this property, we could even consider the following simpler condition: there exist positive numbers c_1, c_2 such that, for all $k \in K_p$,

$$c_1 \|\nabla f(x_k)\| \leq \lambda_m(D_k) \leq \lambda_M(D_k) \leq c_2 \|\nabla f(x_k)\|,$$

which implies (a) and (b). However, we note that, in this case, it would be $D_k \neq 0$ for all $k \in K_p$, while it is not necessary to modify the matrix $J(x_k)^T J(x_k)$ when nonsingular. Therefore, conditions (a) and (b) are less restrictive.

Then, we can prove the following convergence result.

Proposition 3.1. Let $\{x_k\}$ be the sequence generated by the algorithm. Assume that the level set $\Omega_o = \{x \in R^n : f(x) \leq f(x_o)\}$ is compact and that conditions (a) and (b) are satisfied. Then:

(i) the sequence $\{f(x_k)\}$ converges;
(ii) $\lim_{k\to\infty} \|x_{k+1} - x_k\| = 0$;
(iii) every limit point of $\{x_k\}$ is a stationary point of f.

Proof. For simplicity of notation, we disregard the first M iterates, i.e., we assume $k \geq M$. Let $l(k)$ be an integer, $k - M \leq l(k) \leq k$, such that

$$f(x_{l(k)}) = \max_{0 \leq j \leq M} [f(x_{k-j})].$$

We note first that the sequence $\{f(x_{l(k)})\}$ is nonincreasing. Indeed, we have

$$\begin{aligned} f(x_{l(k+1)}) &= \max_{0 \leq j \leq M} [f(x_{k+1-j})] \\ &\leq \max_{0 \leq j \leq M+1} [f(x_{k+1-j})] \\ &= \max\{f(x_{l(k)}), f(x_{k+1})\} \\ &= f(x_{l(k)}), \end{aligned}$$

where the last equality follows from the fact that, by (4),

$$f(x_{k+1}) < f(x_{l(k)}).$$

Again by (4), we can write

$$\begin{aligned} f(x_{l(k)}) &\leq \max_{0 \leq j \leq M} [f(x_{l(k)-1-j})] - \gamma \alpha_{l(k)-1}^2 \|d_{l(k)-1}\|^3 \\ &= f(x_{l(l(k)-1)}) - \gamma \alpha_{l(k)-1}^2 \|d_{l(k)-1}\|^3. \end{aligned} \tag{5}$$

Since

$$f(x_k) \leq f(x_o), \qquad \text{for all } k,$$

we have $\{x_k\} \subset \Omega_o$, so that, by the compactness of Ω_o, the sequence $\{f(x_{l(k)})\}$ admits a limit for $k \to \infty$. Hence, from (5), it follows that

$$\lim_{k\to\infty} \alpha^2_{l(k)-1} \|d_{l(k)-1}\|^3 = 0,$$

which implies that, for every infinite subset $K \subseteq \{0, 1, \dots\}$,

$$\text{either} \quad \lim_{k\to\infty,\, k\in K} \alpha_{l(k)-1} \|d_{l(k)-1}\| = 0,$$
$$\text{or} \quad \lim_{k\to\infty,\, k\in K} \|d_{l(k)-1}\| = 0.$$

The second case implies the first one, since $\alpha_k \leq 1$ for all k (see Algorithm NLS), so that

$$\lim_{k\to\infty} \alpha_{l(k)-1} \|d_{l(k)-1}\| = 0,$$

which is Eq. (8) in Ref. 4. Then, we use the same arguments employed for proving the theorem of Section 3 in Ref. 4. In particular, let

$$\hat{l}(k) = l(k + M + 2);$$

taking into account that f is uniformly continuous on Ω_o, it is shown (Ref. 4) by induction that, for any given $j \geq 1$,

$$\lim_{k\to\infty} \alpha_{\hat{l}(k)-j} \|d_{\hat{l}(k)-j}\| = 0$$

and that, as $\hat{l}(k) - k - 1 \leq M + 1$, it follows that

$$\lim_{k\to\infty} \|x_{k+1} - x_{\hat{l}(k)}\| = 0.$$

Then, since $\{f(x_{l(k)})\}$ admits a limit, from the uniform continuity of f on Ω_o, (i) follows. Moreover, from (4), taking limits for $k \to \infty$, we have

$$\lim_{k\to\infty} \alpha_k^2 \|d_k\|^3 = 0,$$

which implies that ($\alpha_k \leq 1$, for all k)

$$\lim_{k\to\infty} \alpha_k \|d_k\| = 0,$$

i.e., (ii).

To prove (iii), let us consider any infinite subset $K' \subseteq \{0, 1, \dots\}$ such that $\{x_k\}_{k\in K'} \to \hat{x}$. For each $k \in K'$, let $m(k) \in [0, p-1]$ be the integer such that $k + m(k) \in K_p$. Then, since

$$\|x_{k+m(k)} - x_k\| \leq \|x_{k+m(k)} - x_{k+m(k)-1}\| + \cdots + \|x_{k+1} - x_k\|,$$

by (ii) we have

$$\lim_{k\to\infty,\, k\in K'} \|x_{k+m(k)} - x_k\| = 0,$$

which implies

$$\lim_{k\to\infty,\, k\in K'} x_{k+m(k)} = \hat{x}.$$

Therefore, as $k + m(k) \in K_p$, there exists an infinite subset $K \in K_p$ such that

$$\lim_{k\to\infty,\, k\in K} x_k = \hat{x}. \tag{6}$$

From (ii), it follows that

$$\text{either} \quad \lim_{k\to\infty,\, k\in K} \|d_k\| = 0,$$
$$\text{or} \quad \lim_{k\to\infty,\, k\in K} \alpha_k = 0.$$

In the first case, since from (3) we have

$$\|\nabla f(x_k)\| \leq [\|J(x_k)^T J(x_k)\| + \|D_k\|]\|d_k\|,$$

taking into account that $\{x_k\}$ belongs to the compact set Ω_o, by the continuity assumption on ∇f and condition (a), there exists a constant $\mu > 0$ such that, for all $k \in K$,

$$\|\nabla f(x_k)\| \leq \mu \|d_k\|, \tag{7}$$

and then,

$$\nabla f(\hat{x}) = 0.$$

In the second case, if $\lim_{k\to\infty,\, k\in K} \lambda_m(D_k) = 0$, condition (b) implies again $\nabla f(\hat{x}) = 0$. Otherwise, there exists a constant $\hat{\lambda}$ such that $0 < \hat{\lambda} \leq \lambda_m(D_k)$, for sufficiently large $k \in K$. Since d_k is solution of (3) and taking into account inequality (7), we have

$$\begin{aligned} |\nabla f(x_k)^T d_k| / \|d_k\| &= |d_k^T (J(x_k)^T J(x_k) + D_k) d_k| / \|d_k\| \\ &\geq \lambda_m(D_k)\|d_k\| \geq \hat{\lambda} \|\nabla f(x_k)\| / \mu. \end{aligned} \tag{8}$$

By the instructions of Algorithm NLS, since $\alpha_k \to 0$ for $k \to \infty$, $k \in K$, we have that, for sufficiently large $k \in K$,

$$f(x_k + (\alpha_k/\sigma_k) d_k) > f(x_k) - \gamma(\alpha_k^2/\sigma_k^2)\|d_k\|^3,$$

where $\sigma_k \in [\sigma_1, \sigma_2] \subset (0, 1)$, and by the mean-value theorem it follows that

$$\nabla f(x_k + \theta_k(\alpha_k/\sigma_k) d_k)^T d_k / \|d_k\| > -\gamma(\alpha_k/\sigma_k)\|d_k\|^2,$$

where $\theta_k \in (0, 1)$. Taking limits for $k \to \infty$, $k \in K$, since $\alpha_k \|d_k\| \to 0$, $\sigma_k \geq \sigma_1$, and $\{\|d_k\|\}_{k\in K}$ is bounded [since $\lambda_m(D_k)$ is bounded away from zero], we have that $\nabla f(\hat{x})^T \hat{d} \geq 0$. On the other hand, by the continuity assumption on

∇f and the descent property of d_k, we can write

$$\lim_{k\to\infty, k\in K} \nabla f(x_k)^T d_k/\|d_k\| = \nabla f(\hat{x})^T \hat{d} \leq 0,$$

so that $\nabla f(\hat{x})^T \hat{d} \geq 0$ and from (8) we get

$$\nabla f(\hat{x}) = 0. \qquad \square$$

Note that property (ii) could be ensured also by means of an Armijo-type acceptance rule (even nonmonotone), provided that the search direction d_k satisfies, for instance, conditions of the following kind [see (2)–(3) in Ref. 4]: there exist positive numbers c_1, c_2 such that

$$\nabla f(x_k)^T d_k \leq -c_1 \|\nabla f(x_k)\|^2,$$
$$\|d_k\| \leq c_2 \|\nabla f(x_k)\|^2.$$

However, it may be that the columns of the Jacobian matrix are not uniformly linearly independent, so that these conditions could not be satisfied when the minimum norm search direction is used.

We remark that the same convergence result could be obtained even by using, instead of (4), the parabolic stepsize selection rule defined by

$$f(x_k + \alpha d_k) \leq \max_{0\leq j\leq \min(k, M)} [f(x_{k-j})] - \gamma\alpha^2 \|d_k\|^2,$$

i.e., a nonmonotone version of that proposed in Ref. 15. However, the quadratic power of $\|d_k\|$ is not sufficient to ensure the superlinear convergence rate, which is obtained if the stepsize $\alpha_k = 1$ is accepted for k sufficiently large. Indeed, it is possible to show that this property holds by using inequality (4), for the case of zero-residual problems, as follows.

Let us consider the iteration of the algorithm model rewritten in the form

$$x_{k+1} = x_k - \alpha_k B_k \nabla f(x_k),$$

where

$$B_k = \begin{cases} (J(x_k)^T J(x_k))^\dagger, & \text{for } k \notin K_p, \\ \hat{H}_k^{-1}, & \text{for } k \in K_p, \end{cases}$$

and where $\hat{H}_k$ is the matrix $J(x_k)^T J(x_k)$ or a suitable modification of it, obtained by means of D_k. Under the assumption that f is twice continuously differentiable, we can prove the following result.

Proposition 3.2. Let $\{x_k\}$ be the sequence generated by the algorithm. Assume that $\{x_k\}$ converges to x^*, where $f(x^*) = 0$, $\nabla f(x^*) = 0$, and $\nabla^2 f(x^*)$ is positive definite. Then, there exists an integer $\bar{k} \geq 0$ such that, for $k > \bar{k}$, $\alpha_k = 1$ and the sequence $\{x_k\}$ converges superlinearly.

Proof. The proof is based on Proposition 1.15 in Ref. 16.

As $f(x^*) = 0$ (zero-residual case), we have

$$\nabla^2 f(x^*) = J(x^*)^T J(x^*).$$

Since

$$\lim_{k\to\infty} \nabla f(x_k) = 0,$$

condition (a) implies that $D_k \to 0$ as $k \to \infty$ $(k \in K_p)$, so that

$$\lim_{k\to\infty} (B_k - \nabla^2 f(x^*)^{-1}) = 0,$$

and hence the Dennis-Morè condition

$$\lim_{k\to\infty} [\|(B_k - \nabla^2 f(x^*)^{-1})\nabla f(x_k)\| / \|\nabla f(x_k)\|] = 0$$

is satisfied.

From Proposition 1.15 in Ref. 16, it follows that, for k sufficiently large, the stepsize $\alpha = 1$ satisfies the Armijo rule and hence even more so the following nonmonotone Armijo-type rule

$$f(x_k + \alpha d_k) \leq \max_{0\leq j\leq \min(k,M)} [f(x_{k-j})] - \eta\alpha \left|\nabla f(x_k)^T d_k\right|, \tag{9}$$

where $\eta < 1/2$.

Let us consider the positive value of α for which the right-hand sides of (4) and (9) are equal, i.e., the value

$$\hat{\alpha}_k = \eta |\nabla f(x_k)^T d_k| / (\gamma \|d_k\|^3),$$

so that, for any $\alpha \in [0, \hat{\alpha}_k]$, we have

$$\begin{aligned}&\max_{0\leq j\leq \min(k,M)} [f(x_{k-j})] - \eta\alpha \left|\nabla f(x_k)^T d_k\right| \\ &\leq \max_{0\leq j\leq \min(k,M)} [f(x_{k-j})] - \gamma\alpha^2 \|d_k\|^3;\end{aligned}$$

hence, a stepsize $\alpha \leq \hat{\alpha}_k$ satisfying (9) will satisfy also inequality (4). Therefore, we have only to show that, for k sufficiently large, $[0,1] \subset [0, \hat{\alpha}_k]$.

We have

$$\hat{\alpha}_k = \eta |d_k^T B_k^{-1} d_k| / (\gamma \|d_k\|^3) \geq \eta \lambda_m(B_k^{-1}) / (\gamma \|d_k\|),$$

where $\lambda_m(B_k^{-1})$ is the minimum eigenvalue of B_k^{-1}, and then, as $\|d_k\| \to 0$, since $B_k^{-1} \to \nabla^2 f(x^*)$, which is assumed to be positive definite, there exists an index $\bar{k}$ such that, for $k \geq \bar{k}$, $\hat{\alpha}_k > 1$. □

4. Implementative Issues and Numerical Results

In this section, we discuss some implementative aspects of the algorithm model described in Section 2 and we report some numerical results.

As already observed, different algorithms can be obtained depending on the criterion CR adopted and on the method used for computing the minimum norm solution $d_k^{(m)}$ of (1) and for modifying the matrix $J(x_k)^T J(x_k)$ by means of D_k in such a way that conditions (a) and (b) are satisfied.

In particular, for computing $d_k^{(m)}$, we use the conjugate gradient (CG) method, as discussed in the introduction. As regards the methods for computing the diagonal matrix D_k and the solution of (3), it is possible, for instance, to use the modified Cholesky factorization of $J(x_k)^T J(x_k)$ described in Ref. 16. However, it appears more suitable even for large-scale problems, to use again the CG method for solving (3), where D_k can be taken as the following positive multiple of the identity matrix (Ref. 7):

$$D_k = \min\{\beta, \|\nabla f(x_k)\|\} I, \qquad \beta > 0. \tag{10}$$

Note that this choice implies directly the fulfillment of conditions (a) and (b).

Finally, we have chosen a rule based on the stepsize for establishing whether the matrix $J(x_k)^T J(x_k)$ will be modified or not (criterion CR). In particular, if the unit stepsize is rejected by a relaxed (nonmonotone) acceptance rule, it is likely that the progress obtained in the minimization process is poor, and in this case it could be convenient to use a gradient related search direction. Therefore, the matrix $J(x_k)^T J(x_k)$ will be modified by means of D_k, whenever the pure Gauss-Newton iteration [i.e., the unit stepsize along $d_k^{(m)}$] was rejected by the nonmonotone line search at the previous iteration, and in any case after $p-1$ consecutive iterations performed using $d_k^{(m)}$. Thus, p represents the maximum number of iterates within which the search direction is computed by solving (3).

The numerical results reported below have been obtained by the algorithm described in Section 2 with the following implementative choices (it is named Algorithm NMGN):

(i) $p = 20$;
(ii) stopping criterion: $\|\nabla f(x_k)\| \leq 10^{-6}$;
(iii) the standard CG method is used for computing both the minimum norm solution $d_k^{(m)}$ of (1) and the solution of (3), where D_k is given by (10) where $\beta = 1$;
(iv) line search parameters: $\gamma = 10^{-4}$, $M = 10$, $\sigma_1 = 0.1$, $\sigma_2 = 0.5$; at Step 2, the scalar σ is computed by means of a quadratic interpolation formula.

Note that the acceptance rule (4), unlike the classical Armijo rule based on the directional-derivative, is sensitive to scaling. Therefore, it could be advisable to take γ as the product of a standard value, say 10^{-6}, and the user's estimate of a typical magnitude of f, for instance $f(x_o)$. The latter may be possibly updated during the minimization process.

Table 1. Comparison of the results obtained with Algorithm NMGN ($p = 1$ and $p = 20$).

	Algorithm		NMGN ($p = 1$)		NMGN ($p = 20$)	
Function	n	m	n_i	n_f	n_i	n_f
Scaled Rosenbrock, $C = 10^4$	2	2	12	14	7	11
$C = 10^5$			13	16	7	12
$C = 10^6$			14	17	7	12
Extended Rosenbrok, $C = 10^4$	10	10	14	16	7	11
$C = 10^5$			15	18	7	12
$C = 10^6$			16	19	7	12
Scaled cube, $C = 10^4$	2	2	10	11	7	8
$C = 10^5$			10	11	7	8
$C = 10^6$			11	12	7	8
Scaled sine valley, $C = 10^4$	2	2	22	29	3	6
$C = 10^5$			28	40	5	11
$C = 10^6$			33	50	33	69
Scaled power valley, $C = 10^2$	2	2	15	18	7	13
$C = 10^3$			76	79	78	84
$C = 10^4$			1153	1762	822	1264
Powell badly scaled	2	2	8280	8281	11	12
Brown badly scaled	2	3	34	58	14	39
Freudenstein and Roth	2	2	13	14	9	10
Beale	2	3	9	10	10	13
Gulf Research and Development	3	3	38	40	23	34
Box three-dimensional	3	4	78	79	4	5
Gaussian	3	15	10	11	6	7
Powell singular	4	4	12	13	10	11
Miele and Cantrell	4	4	78	83	26	31
Wood	4	6	157	163	67	80
Helical valley	5	10	10	11	7	9
Penalty II	5	10	6	9	90	158
Biggs EXP6	6	7	72	73	7	8
Chebyquad	9	9	7	9	10	14
Brown almost-linear	10	10	5	6	4	5
Broyden tridiagonal	10	10	4	6	5	7
Trigonometric	10	10	7	8	6	7
Penalty I	10	11	323	324	158	213
Dixon	10	11	429	430	278	397
Variably dimensioned	10	12	8	9	8	9
Watson	12	31	23	24	4	5

We have considered a set of standard zero-residual test problems from the literature (Refs. 17–19); the Penalty I and II functions represent small-residual problems. The scale factor C in the first five functions is the same as described in Ref. 17. Moreover, for the Freudenstein and Roth function, we have taken the initial point $x_o = (-10, 20)$, since starting from that suggested in Ref. 17, it is likely that any nonglobal algorithm will converge to the nonzero residual solution.

In order to show the usefulness of the minimum norm solution of (1) as search direction, we compare in Table 1 the results obtained by Algorithm NMGN with those obtained by the same algorithm where $p = 1$, that corresponds to the classical Gauss-Newton method with a nonmonotone line search. For each problem, we report the number n_i of iterations and the number n_f of function evaluations required for satisfying the stopping criterion. Note that the gradient is evaluated only once at each iteration, so that $n_g = n_i$.

Over 36 problems, Algorithm NMGN with $p = 1$ performs clearly better than with $p = 20$ only for the Penalty II function and is slightly better in other 5 cases. In the remaining 30 problems, the behavior of Algorithm NMGN with $p = 20$ is significantly superior in 9 cases. These results give evidence of

Table 2. Comparison of the results obtained with Algorithms NMGN-P and NMGN, both with $p = 20$.

	Algorithm		NMGN-P		NMGN	
Function	n	m	n_i	n_f	n_i	n_f
Scaled Rosenbrock, $C = 10^4$	2	2	12	23	7	11
$C = 10^5$			13	26	7	12
$C = 10^6$			13	26	7	12
Extended Rosenbrock, $C = 10^4$	10	10	12	23	7	11
$C = 10^5$			13	26	7	12
$C = 10^6$			13	26	7	12
Scaled sine valley, $C = 10^5$	2	2	5	12	5	11
$C = 10^6$			44	111	33	69
Scaled power valley, $C = 10^2$	2	2	10	24	7	13
$C = 10^3$			76	86	78	84
$C = 10^4$			808	1243	822	1264
Brown badly scaled	2	3	10	30	14	39
Beale	2	3	11	19	10	13
Gulf Research and Development	3	3	14	28	23	34
Miele and Cantrell	4	4	28	33	26	31
Wood	4	6	85	122	67	80
Helical valley	5	10	8	10	7	9
Penalty II	5	10	325	757	90	158
Chebyquad	9	9	13	27	10	14
Penalty I	10	11	65	86	158	213
Dixon	10	11	517	1186	278	397

Table 3. Comparison of the results obtained with the E04GBF routine and with Algorithm NMGN ($p = 20$).

	Algorithm		E04GBF			NMGN		
Function	n	m	n_i	n_f	n_g	n_i	n_f	n_g
Scaled Rosenbrock, $C = 10^4$	2	2	12	31	31	7	11	7
$C = 10^5$			13	34	34	7	12	7
$C = 10^6$			13	34	34	7	12	7
Extended Rosenbrock, $C = 10^4$	10	10	12	31	31	7	11	7
$C = 10^5$			13	34	34	7	12	7
$C = 10^6$			13	34	34	7	12	7
Scaled cube, $C = 10^4$	2	2	6	8	8	7	8	7
$C = 10^5$			6	8	8	7	8	7
$C = 10^6$			6	8	8	7	8	7
Scaled sine valley, $C = 10^4$	2	2	4	12	12	3	6	3
$C = 10^5$			5	13	13	5	11	5
$C = 10^6$			58	151	151	33	69	33
Scaled power valley, $C = 10^2$	2	2	12	31	31	7	13	7
$C = 10^3$			72	92	92	78	84	78
$C = 10^4$			$(*)$			822	1264	822
Powell badly scaled	2	2	25	47	47	11	12	11
Brown badly scaled	2	3	23	57	57	14	39	14
Freudenstein and Roth	2	2	9	10	10	9	10	9
Beale	2	3	6	8	8	10	13	10
Gulf Research and Development	3	3	57	204	204	23	34	23
Box three-dimensional	3	4	4	5	5	4	5	4
Gaussian	3	15	6	7	7	6	7	6
Powell singular	4	4	10	11	11	10	11	10
Miele and Cantrell	4	4	33	62	62	26	31	26
Wood	4	6	47	88	88	67	80	67
Helical valley	5	10	8	11	11	7	9	7
Penalty II	5	10	200	250	250	90	158	90
Biggs EXP6	6	7	7	12	12	7	8	7
Chebyquad	9	9	11	28	28	10	14	10
Brown almost-linear	10	10	4	5	5	4	5	4
Broyden tridiagonal	10	10	4	9	9	5	7	5
Trigonometric	10	10	6	7	7	6	7	6
Penalty I	10	11	66	128	128	158	213	158
Dixon	10	11	35	60	60	278	397	278
Variably dimensioned	10	12	8	9	9	8	9	8
Watson	12	31	4	5	5	4	5	4

$(*)$ Line search failure.

the computational advantage deriving from the use of the minimum norm search direction.

It may be of interest to compare the behavior of Algorithm NMGN with that of the same algorithm without using any criterion CR, which is a version where the modification of the matrix $J(x_k)^T J(x_k)$ is performed only periodically (Algorithm NMGN-P). In Table 2, the results obtained for the problems, among those considered, where there is a difference in the performance of the two algorithms are reported. Over the remaining 21 problems, Algorithm NMGN-P performs clearly better only for the Penalty I function and is slightly better in other 4 cases. This indicates the usefulness of introducing the criterion CR. Note also that, in the 15 problems where the results are identical, the unit stepsize is accepted in all iterations; i.e., the pure Gauss-Newton iteration is always performed, since $n_i < p$.

Although a finite value of the integer p is theoretically needed for getting global convergence, one could have the impression that, in practice, better results could be obtained by taking $p = \infty$. However, Algorithm NMGN with $p = \infty$ gives the same results obtained with $p = 20$ for all the problems considered, but for the Penalty I function only ($n_i = 45$, $n_f = 47$). This substantial equivalence can be explained noting that, according to criterion CR, the matrix $J(x_k)^T J(x_k)$ is in any case modified when the unit stepsize is not accepted, so that the behavior of the algorithm is not greatly influenced by the value of p, provided that it is relatively large.

Finally, we compare in Table 3 the results obtained by Algorithm NMGN with those obtained by the E04GBF routine (Nag Library), which is a robust implementation of the Gauss-Newton method (Ref. 20). Since this routine computes the gradient at each function evaluation, in order to facilitate the comparison we report also the number n_g of gradient evaluations.

We observe that the two algorithms are substantially comparable in terms of number of iterations, while in many cases the numbers n_f and n_g for Algorithm NMGN are considerably smaller than those required by the E04GBF routine. These results indicate that the nonmonotone version of the Gauss-Newton method proposed here can deal efficiently with nonlinear least-squares problems, mainly with the zero residual ones.

We conclude by observing that, in Algorithm NMGN, it is assumed that, at every iteration, problem (1) or system (3) are solved exactly. For large-scale problems it could be more effective to use inexact methods that find an approximate solution satisfying some appropriate conditions, and this could be the topic of further work.

References

1. Ben-Israel, A., *A Newton-Raphson Method for the Solution of Systems of Equations*, Journal of Mathematical Analysis and Applications, Vol. 15, pp. 243–252, 1966.
2. Boggs, P. T., *The Convergence of the Ben-Israel Iteration for Nonlinear Least Squares Problems*, Mathematics of Computation, Vol. 30, pp. 512–522, 1976.

3. Menzel, R., *On Solving Nonlinear Least-Squares Problems in Case of Rank-Deficient Jacobians*, Computing, Vol. 34, pp. 63–72, 1985.
4. Grippo, L., Lampariello, F., and Lucidi, S., *A Nonmonotone Line Search Technique for Newton's Method*, SIAM Journal on Numerical Analysis, Vol. 23, pp. 707–716, 1986.
5. Grippo, L., Lampariello, F., and Lucidi, S., *A Class of Nonmonotone Stabilization Methods in Unconstrained Optimization*, Numerische Mathematik, Vol. 59, pp. 779–805, 1991.
6. Ferris, M. C., Lucidi, S., and Roma, M., *Nonmonotone Curvilinear Line Search Methods for Unconstrained Optimization*, Computational Optimization and Applications, Vol. 6, pp. 117–136, 1996.
7. Ferris, M. C., and Lucidi, S., *Nonmonotone Stabilization Methods for Nonlinear Equations*, Journal of Optimization Theory and Applications, Vol. 81, pp. 815–832, 1994.
8. Zhang, J. Z., and Chen, L. H., *Nonmonotone Levenberg-Marquardt Algorithms and Their Convergence Analysis*, Journal of Optimization Theory and Applications, Vol. 92, pp. 393–418, 1997.
9. Golub, G. H., and Van Loan, C. F., *Matrix Computations*, 2nd Edition, The John Hopkins University Press, Baltimore, Maryland, 1989.
10. Hestenes, M. R., *Conjugate Direction Methods in Optimization*, Springer Verlag, Berlin, Germany, 1980.
11. Barzilai, J., and Borwein, J. M., *Two-Point Stepsize Gradient Methods*, IMA Journal of Numerical Analysis, Vol. 8, pp. 141–148, 1988.
12. Fletcher, R., *On the Barzilai-Borwein Method*, Dundee Numerical Analysis Report NA/207, University of Dundee, 2001.
13. Björck, A., *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, Pennsylvania, 1996.
14. Lampariello, F., and Sciandrone, M., *Global Convergence Technique for the Newton Method with Periodic Hessian Evaluation*, Journal of Optimization Theory and Applications, Vol. 111, pp. 341–358, 2001.
15. De Leone, R., Gaudioso, M., and Grippo, L., *Stopping Criteria for Line-Search Methods without Derivatives*, Mathematical Programming, Vol. 30, pp. 285–300, 1984.
16. Bertsekas, D. P., *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, New York, NY, 1980.
17. Moré, J. J., Garbow, B. S., and Hillstrom, K. E., *Testing Unconstrained Optimization Software*, ACM Transactions on Mathematical Software, Vol. 7, pp. 17–41, 1981.
18. Schittkowski, K., *More Test Examples for Nonlinear Programming Codes*, Springer Verlag, Berlin, Germany, 1987.
19. Wolfe, M. A., *Numerical Methods for Unconstrained Optimization*, Van Nostrand Reinhold, New York, NY, 1978.
20. Gill, P. E., and Murray, W., *Algorithms for the Solution of the Nonlinear Least-Squares Problem*, SIAM Journal on Numerical Analysis, Vol. 15, pp. 977–992, 1978.