KYBERNETIKA — VOLUME *28* (1992), NUMBER 4, PAGES 249 – 262

COMPUTATIONAL EXPERIENCE WITH IMPROVED CONJUGATE GRADIENT METHODS FOR UNCONSTRAINED MINIMIZATION

Ladislav Lukšan

The paper contains a description of new restart procedures for the conjugate gradient methods and a numerical investigation of the influence of line search and scaling on their efficiency. Computational results obtained by means of 15 sufficiently difficult problems are given.

1. INTRODUCTION

We are concerned with the finding of a local minimum $x^* \in \mathbb{R}^n$ of the function $F: X \to \mathbb{R}$ on an open set $X \subset \mathbb{R}^n$, i. e. a point $x^* \in \mathbb{R}^n$ that satisfies the inequality $F(x^*) \leq F(x)\ \forall x \in B(x^*, \varepsilon)$ for some $\varepsilon > 0$, where $B(x^*, \varepsilon) = \{x \in \mathbb{R}^n : \| x - x^* \| < \varepsilon\} \subset X$ is an open ball contained in $X \subset \mathbb{R}^n$. The most frequently used methods for this purpose are the variable metric (VM) methods whose iteration step has the form

$$x^+ = x + \alpha\, s \tag{1}$$

where x and x^+ are old and new vectors of variables respectively, α is a positive stepsize chosen so that

$$F^+ - F \leq \bar{\varepsilon}_1 \alpha\ s^{\mathrm{T}} g \tag{1.2a}$$

and

$$s^{\mathrm{T}} g^+ \geq \bar{\varepsilon}_2 s^{\mathrm{T}} g \tag{1.2b}$$

with $0 < \bar{\varepsilon}_1 < 1/2$ and $\bar{\varepsilon}_1 < \bar{\varepsilon}_2 < 1$, where F and F^+ are old and new values of the objective function respectively, g and g^+ are old and new gradients of the objective function respectively, and s is a direction vector satisfying the equation $Bs + g = 0$, where B is a symmetric positive definite approximation of the Hessian matrix that is constructed iteratively (see [17]).

If the number of variables is large, then matrix B cannot be stored, nor factored in a reasonable time, so other methods have to be used. There exist several classes of such methods: conjugate gradient methods [10], difference versions of truncated Newton methods [7], variable metric methods with limited storage [19], sparse variants of variable metric methods [26], and partitioned variable metric methods for separable problems [12]. The last two classes require the special structure of optimization problems. From the other classes the simplest are the conjugate gradient methods which need only 3–5 n-dimensional vectors (it depends on their implementation). Recently new attention has been given to these methods because they are globally convergent with mild and reasonable assumptions.

In this paper, we are concerned with an efficient implementation of the conjugate gradient (CG) methods. These methods are iterative and their iteration step has

the form (1.1) where α is a positive stepsize chosen so that

$$F^+ - F \leq \bar{\varepsilon}_1 \alpha \, s^{\mathrm{T}} g \tag{1.3a}$$

and

$$|s^{\mathrm{T}} g^+| \leq \bar{\varepsilon}_2 |s^{\mathrm{T}} g| \tag{1.3b}$$

with $0 < \bar{\varepsilon}_1 < \bar{\varepsilon}_2 < 1/2$ and s is a direction vector which is constructed iteratively by means of the formula

$$s^+ = -g^+ + \beta \, s \tag{4}$$

(with $s = -g$ in the first iteration). The parameter β is computed so that the CG method with perfect line search finds the minimum of a quadratic function after a finite number of steps. There are three possibilities: the Hestenes and Stiefel [13] method

$$\beta_{HS} = \frac{y^{\mathrm{T}} g^+}{y^{\mathrm{T}} s} \tag{1.5a}$$

the Polak and Ribire [20] method

$$\beta_{PR} = \frac{y^{\mathrm{T}} g^+}{g^{\mathrm{T}} g} \tag{1.5b}$$

and the Fletcher and Reeves [10] method

$$\beta_{FR} = \frac{(g^+)^{\mathrm{T}} g^+}{g^{\mathrm{T}} g} \tag{1.5c}$$

(we use notation $d = x^+ - x$ and $y = g^+ - g$). Although the Hestenes and Stiefel (HS) method (1.5a) is most general and the Fletcher and Reeves (FR) method (1.5c) is the simplest with good global convergence properties (theoretical), the most numerically efficient was proved to be the Polak and Ribier (PR) method (1.5b).

The CG methods are more sensitive to their implementation than the VM methods:

1. The initial estimate $\alpha_1 = 1$ of α^+ in the line search algorithm does not have theoretical justification for CG methods. Therefore the CG methods are more sensitive to the initial estimate α_1 than the VM methods.
2. CG methods need more perfect line search than VM methods. We usually use $\bar{\varepsilon}_2 = 0.1$ in (1.3) instead of $\bar{\varepsilon}_2 = 0.9$ in (1.2).
3. CG methods strongly depend on restarts while VM methods need not be restarted.

In this paper we propose several efficient implementations of the CG methods based on recent convergence results. Computational efficiency of these implementations is demonstrated on 15 sufficiently complicated test problems.

2. RECENT CONVERGENCE RESULTS

It is well known that any CG method with perfect line search (with (1.3b) where $\bar{\varepsilon}_2 = 0$) finds the minimum of a quadratic function after at most n steps. This property implies that any convergent CG method with asymptotically perfect line search and with periodic restart is n-step quadratically convergent (see [2]). This result is very useful because asymptotically perfect line search can be easily realized by both quadratic and cubic interpolations.

The global convergence of CG methods can be assured by suitable restart rules. The simplest such rule is the so-called angle test which consists in setting $\beta = 0$ in (1.4) whenever

$$\cos \vartheta = -\frac{s^{\mathrm{T}} g}{\| s \| \, \| g \|} < \bar{\varepsilon}_0 \tag{1}$$

where $\overline{\varepsilon}_0$ is a prescribed constant (usually $\overline{\varepsilon}_0 = 10^{-3}$). A more complicated angle test is proposed in [25]. If the line search is asymptotically perfect, the global convergence of CG methods can be assured by periodic restarts (see [16]).

The first global convergence result which does not depend on restarts has been obtained by Zoutendijk [28] and Powell [22], who proved that the FR method with perfect line search is globally convergent in the sense that

$$\liminf \| g \| = 0 \tag{2}$$

where lim inf is taken over the iterative process (1.1). Later Al-Bali [1] generalized this result to include the FR method without perfect line search. He has shown that (2.2) holds for the FR method whenever $\overline{\varepsilon}_2 < 1/2$ in (1.3). Recently great effort was devoted to generalizing this result to other CG methods. Touati-Ahmed and Storey [27] have shown that the iterative process (1.1) and (1.4) with a line search satisfying (1.3) is globally convergent if

$$0 \leq \beta \leq \overline{\eta}_2 \beta_{FR} \tag{2.3}$$

and

$$\overline{\lambda} \| g^+ \|^2 \leq (1/\overline{\eta}_2)^k \tag{2.4}$$

hold in every iteration, where $0 < \overline{\lambda}, 1 < \overline{\eta}_2 < 1/(2\overline{\varepsilon}_2)$ are suitable constants and k is the iteration number counted from the last restart. Moreover, (2.3) and $1 < \overline{\eta}_2 < 1/(2\overline{\varepsilon}_2)$ imply the inequality

$$g^{\mathrm{T}} s \leq -\frac{1 - 2\overline{\varepsilon}_2 \overline{\eta}_2}{1 - \overline{\varepsilon}_2 \overline{\eta}_2} \| g \|^2 < 0 \tag{2.5}$$

independently of (2.4) (see the proof of Theorem 2.2 given in [27]). Therefore the CG method is a descent one if (2.3) and $1 < \overline{\eta}_2 < 1/(2\overline{\varepsilon}_2)$ hold. The most general result has been obtained by Gilbert and Nocedal [11], who have shown that the both PR and HS methods are globally convergent if they generate positive values of β and if (2.5) holds. This result is very important because it allows us to develop a great number of useful restart procedures for CG methods.

3. NEW RESTART PROCEDURES

We limit our attention to the PR method, but the same considerations can be used for the HS method. Usually the PR method is implemented with periodic restarts and with angle test which serves as a safeguard. Let us denote this possibility as $REST = 1$:

$$\begin{aligned} \beta &= 0 \qquad \text{if (2.1) holds,} \\ \beta &= 0 \qquad \text{if } k = n + 1, \\ \beta &= \beta_{PR} \quad \text{otherwise,} \end{aligned}$$

where k is the iteration number counted from the last restart (i. e. from the last iteration with $\beta = 0$).

In [23], Powell points out that the PR method works better if it is restarted also whenever

$$\beta_{PR} < 0. \tag{3.1}$$

Let us denote this strategy as $REST = 2$:

$$\begin{aligned} \beta &= 0 \qquad \text{if (2.1) holds,} \\ \beta &= 0 \qquad \text{if } k = n + 1 \\ \beta &= 0 \qquad \text{if (3.1) holds,} \\ \beta &= \beta_{PR} \quad \text{otherwise.} \end{aligned}$$

Convergence results noted in the previous section together with our computational experiments show that the PR method is more efficient if it is restarted not only when (3.1) holds, but also whenever

$$\beta_{PR} > \overline{\eta}_2 \beta_{FR} \tag{3.2}$$

(see (2.3)) where $1 < \overline{\eta}_2 < 1/(2\overline{\varepsilon}_2)$ is a suitable constant (we recommend $\overline{\eta}_2 = 1.34$, all recommended values given in this paper were obtained experimentally by means of extensive computations). Let us denote this strategy as $REST = 3$:

$\beta = 0$ if (2.1) holds,

$\beta = 0$ if $k = n + 1$,

$\beta = 0$ if either (3.1) or (3.2) holds, and

$\beta = \beta_{PR}$ otherwise.

We have performed several tests with these three rules. The best results were obtained with the choice $REST = 3$, which is the first restart procedure we recommend. Note that the test (2.1) has not been active in any problem chosen for detailed study of restart causes.

The PR method with periodic restarts can be disadvantageous for some problems that require more restarts at the beginning of the iterative process. Therefore we tried to adapt the criterion (2.4) for substituting the periodic restarts. The original form (2.4) is also disadvantageous because it does not depend on the number of variables and for n large it can cause too frequent restarts. Therefore we are proposing a new rule which implies restart whenever

$$\overline{\lambda} \parallel g^+ \parallel^2 > \overline{\omega}^k \tag{3.3a}$$

where

$$\overline{\omega} = 10^{-\frac{\overline{\tau}}{n+1}} \tag{3.3b}$$

and where $\overline{\lambda}$ and $\overline{\tau}$ are suitable constants (we recommend $\overline{\lambda} = 10^{-8}$ and $\overline{\tau} = 4.1$). This leads to the new restart procedure we denote as $REST = 4$:

$\beta = 0$ if either $k = 12n$ or (2.1) holds,

$\beta = 0$ if either (3.1) or (3.2) holds,

$\beta = 0$ if (3.3) holds, and

$\beta = \beta_{PR}$ otherwise.

Test $k = 12n$ serves only as a safeguard and it has not been active in any problem chosen for detailed study of restart causes.

Another restart condition can be derived from gradient orthogonality. If any CG method with perfect line search is applied to a quadratic function, then necessarily $g^{\mathrm{T}} g^+ = 0$. In the general case, we can require $g^{\mathrm{T}} g^+ \cong 0$, which gives $\beta_{PR} \cong \beta_{FR}$. Therefore we are proposing the rule which implies restart whenever either (3.2) or

$$\beta_{PR} < \overline{\eta}_1 \beta_{FR} \tag{3.4}$$

holds, where $\overline{\eta}_1$ is a suitable constant (we recommend $\overline{\eta}_1 = 0.74$). Let us denote the resulting procedure as $REST = 5$:

$\beta = 0$ if either $k = 12n$ or (2.1) holds,

$\beta = 0$ if either (3.4) or (3.2) holds,

$\beta = \beta_{PR}$ otherwise.

Note that for $\overline{\eta}_1 = 0.8$ and $\overline{\eta}_2 = 1.2$ we obtain the Powell restart procedure proposed in [21]. The Powell restart procedure will be denoted as $REST = 6$.

The gradient orthogonality is not the only condition which can be used for monitoring the PR method. Another such condition is mutual conjugacy. If any CG method with perfect line search is applied to a quadratic function, then necessarily $y^{\mathrm{T}} s^{+} = 0$. Therefore we are proposing the new rule which implies restart whenever

$$|y^{\mathrm{T}} s^{+}| > \overline{\eta}_0 \parallel y \parallel \parallel s^{+} \parallel \tag{3.5}$$

where $\overline{\eta}_0$ is a suitable constant (we recommend $\overline{\eta}_0 = 0.015$). This leads to the new restart procedure we denote as $REST = 7$:

$\beta = 0$ if either $k = 12n$ or (2.1) holds,

$\beta = 0$ if either (3.1) or (3.2) holds,

$\beta = 0$ if (3.5) holds,

$\beta = \beta_{PR}$ otherwise

The proposed restart procedures were proved very efficient as will be shown in Section 5. Note that we have studied many other restart procedures, such as ones given in [11, 14, 27], but the results obtained have been worse then those given in Section 5.

4. LINE SEARCH AND SCALING

Since the CG methods require more perfect line search than other methods, they are very sensitive to its realization. We essentially use the standard line search implementation, which can be represented by the following algorithm:

Algorithm 4.1. Input data: $\overline{\Delta} > 0,\ 0 < \overline{\beta}_1 < \overline{\beta}_2 < 1,\ 0 < \overline{\varepsilon}_1 < \overline{\varepsilon}_2 < 1/2$.

Step 1. Determine the initial estimate α_1 of α^{+}. Set $\sigma_1 := 0$. Set $i := 1$.

Step 2. Set $\alpha_i := \min(\alpha_i, \overline{\Delta}/ \parallel s \parallel)$. Set $\rho_i := \sigma_i$ and $\sigma_i := \alpha_i$. If the conditions (1.3) are satisfied with F^{+} and g^{+} replaced by $F(x + \alpha_i s)$ and $g(x + \alpha_i s)$ respectively, then set $\alpha^{+} := \alpha_i$ and terminate the computation. If both (1.3a) and $s^{\mathrm{T}} g(x + \alpha_i s) < 0$ hold then go to Step 3, else go to Step 4.

Step 3. If $\alpha_i = \overline{\Delta}/ \parallel s \parallel$ then set $\alpha^{+} := \alpha_i$ and terminate the computation, else determine the new estimate α_i by cubic extrapolation. Set $\alpha_i := \max(\alpha_i, \sigma_i/\overline{\beta}_2)$, set $\alpha_i := \min(\alpha_i, \sigma_i/\overline{\beta}_1)$, and go to Step 2.

Step 4. Determine the new estimate α_i by cubic interpolation. Set $\alpha_i := \max(\alpha_i, \rho_i +$

$+ \overline{\beta}_1(\sigma_i - \rho_i))$, set $\alpha_i := \min(\alpha_i, \rho_i + \overline{\beta}_2(\sigma_i - \rho_i))$.

Step 5. If the conditions (1.3) are satisfied, with F^{+} and g^{+} replaced by $F(x+\alpha_i s)$ and $g(x+\alpha_i s)$ respectively, then set $\alpha^{+} := \alpha_i$ and terminate the computation. If both (1.3a) and $s^{\mathrm{T}} g(x + \alpha_i s) < 0$ hold then set $\rho_i := \alpha_i$, else set $\sigma_i := \alpha_i$. Go to Step 4.

Comments.

1) CG methods are sensitive to the order of interpolation. Therefore we recommend the cubic interpolation given in [5] over the quadratic one. The results obtained with the quadratic interpolation was much worse that those given in Section 5. Good results were obtained also with the conic interpolation proposed in [3].

2) All results shown in Section 5 were obtained with the input data $\overline{\varepsilon}_1 = 0.0001, \overline{\varepsilon}_2 = 0.1, \overline{\beta}_1 = 0.01, \overline{\beta}_2 = 0.9$. We have performed also tests with other values of $\overline{\varepsilon}_2$, but $\overline{\varepsilon}_2 = 0.1$ was proved very reasonable. Bound $\overline{\Delta}$, which serves to safeguard against overflows, depends on the problem to be solved, and corresponding values are given in Section 5.

Great attention was devoted to the choice of the initial estimate α_1 in Step 1 of the line search algorithm. There exist two standard choices:

$$\alpha_1 = 1 \tag{4.1}$$

for the Newton method and

$$\alpha_1 = 2\,\frac{F_{\min} - F}{s^{\mathrm{T}} g} \tag{4.2}$$

for the CG methods (see [10]), where $F_{\min}$ is the lower bound for the minimal value $F(x^*)$. These simple choices are inefficient, but they are often combined. In [9], the initial estimate

$$\alpha_1 = \min\left(1, 2\,\frac{F_{\min} - F}{s^{\mathrm{T}} g}\right) \tag{4.3}$$

is recommended, while in [6] and [15], the authors propose the initial estimates

$$\alpha_1 = \min\left(1, 4\,\frac{F_{\min} - F}{s^{\mathrm{T}} g}\right) \tag{4.4}$$

and

$$\alpha_1 = \min\left(2, 2\,\frac{F_{\min} - F}{s^{\mathrm{T}} g}\right) \tag{4.5}$$

respectively. The choice (4.4) is frequently used for the VM methods. Our experience show that the choice (4.3) is slightly better then (4.4) and (4.5) and we recommend it over all choices (4.1) – (4.5).

Other initial estimates can be derived from the assumption that $F^+ - F \cong F - F^-$, where F^- is the value of the objective function in the previous iteration. Therefore, we can substitute $F - F^-$ for $F_{\min} - F$ in (4.2) and we obtain

$$\alpha_1 = 2\,\frac{F - F^-}{s^{\mathrm{T}} g} \tag{4.6}$$

as in [8]. Again (4.3) can be combined with (4.1). The resulting initial estimate has the form

$$\alpha_1 = \min\left(1, 2\,\frac{F - F^-}{s^{\mathrm{T}} g}\right), \tag{4.7}$$

which generalizes (4.3). The choice (4.7) is shown to be very effective in Section 5. This choice was proved more efficient than other more complicated choices we have tested, and we recommend it in connection with CG methods.

For the simplification of subsequent considerations, we denote by $INIT = 1$ the choice (4.1), by $INIT = 2$ the choice (4.2) if $F_{\min}$ is given or (4.1) otherwise, by $INIT = 3$ the choice (4.3) if $F_{\min}$ is given or (4.1) otherwise, by $INIT = 4$ the choice (4.6), and by $INIT = 5$ the choice (4.7).

Another useful tool for improving CG methods is scaling, which was originally developed for VM methods (see [24]). The scaling consists in replacing (1.4) by

$$s^+ = \gamma^+(-g^+ + \beta s), \tag{4.8}$$

where γ^+ is the scaling factor. We can use the same scaling factor as for the BFGS method (see [17]). Then

$$\gamma^+ = \frac{y^{\mathrm{T}} d}{y^{\mathrm{T}} y} \tag{4.9}$$

(again $y - g^+ - g$ and $d = x^+ - x$). Note that when we use (4.8) then (1.5b) and (1.5c) have to be replaced by

$$\beta_{PR} = \frac{1}{\gamma} \frac{y^{\mathrm{T}} g^+}{g^{\mathrm{T}} g} \tag{4.10b}$$

and

$$\beta_{FR} = \frac{1}{\gamma} \frac{(g^+)^{\mathrm{T}} g^+}{g^{\mathrm{T}} g} \tag{4.10c}$$

where γ is the scaling factor used in the previous iteration.

For the simplification of subsequent considerations, we denote by $SCAL = 1$ the choice

$$\gamma^+ = 1$$

and by $SCAL = 2$ the choice

$$\begin{aligned} \gamma^+ &= \overline{\gamma}_1, \quad \text{if } \tfrac{y^{\mathrm{T}} d}{y^{\mathrm{T}} y} < \overline{\gamma}_1 \\ \gamma^+ &= \overline{\gamma}_2, \quad \text{if } \tfrac{y^{\mathrm{T}} d}{y^{\mathrm{T}} y} > \overline{\gamma}_2 \\ \gamma^+ &= \tfrac{y^{\mathrm{T}} d}{y^{\mathrm{T}} y}, \quad \text{otherwise,} \end{aligned}$$

where $0 < \overline{\gamma}_1 < 1 < \overline{\gamma}_2$ (we recommend $\overline{\gamma}_1 = 0.005$ and $\overline{\gamma}_2 = 200$). The bounds $\overline{\gamma}_1$ and $\overline{\gamma}_2$ serve for improvement of stability.

5. COMPUTATIONAL EXPERIMENTS

In this section, we present results of a comparative study of CG methods obtained by means of 15 sufficiently difficult test problems given in [17], which are modifications of test problems given in [4]:

1. The chained Rosenbrock function.
2. The chained Wood function.
3. The chained Powell singular function.
4. The chained Cragg and Levy function.
5. A generalization of the Broyden tridiagonal function.
6. A generalization of the Broyden banded function.
7. Toint's 7-diagonal generalization of the Broyden tridiagonal function.
8. A generalization of the Nazareth trigonometric function.
9. A generalization of the Toint trigonometric function.
10. A penalty function.
11. An augmented Lagrangian function.
12. A generalization of the first Brown function.
13. A generalization of the second Brown function.
14. A discrete boundary value problem.
15. A discrete variational problem.

All test problems were solved for 20 ($n = 20$) and 100 ($n = 100$) variables and selected problems were solved for 200 ($n = 200$) and 500 ($n = 500$) variables. For most problems we used the bound $\overline{\Delta} = 1000$ (see Algorithm 4.1), for problems 4, 10, 12, 13 we chose $\overline{\Delta} = 10$, and for problems 9, 11 we chose $\overline{\Delta} = 1$. The computation was always stopped whenever the gradient norm became less then 10^{-6}. The results of our experiments are summarized in several tables. Table 1, Table 2, Table 4 and Table 5 contain detailed tests for 20 and 100 variables. Rows of these tables correspond to 15 test problems and columns correspond to selected methods. The results are presented in the form IT–IF, where IT is the number of iterations and IF is both the number of function evaluations and the number of gradient evaluations (in our line search algorithm the value and the gradient of the objective function are evaluated at the same time). The asterisk in the second row of several tables means that a nonoptimal point with the gradient norm less then 10^{-6} was obtained. The row denoted by Σ contains the total number of iterations and the total number

of function evaluations. Table 3 contains only total numbers of iterations and total numbers of function evaluations. Table 6 contains detailed tests for 200 variables. Test problems 2, 8, 9, 14, and 15 were not used because problem 2 has many almost stationary points, problems 8 and 9 are too dense, problem 14 is too ill-conditioned, and problem 15 is unbounded.

Table 1 shows the efficiency of individual restart procedures for "standard" choices $INIT = 5$ and $SCAL = 2$. We can see that the periodic restart procedure ($REST = 3$) can be less efficient for problem 14, which requires more frequent restarts.

Table 1

(PR)	$INIT = 5,\ SCAL = 2$				
$n = 20$	$REST = 3$	$REST = 4$	$REST = 5$	$REST = 6$	$REST = 7$
1	221–433	244–476	235–489	244–504	247–479
2	147–291	137–271	226–447	124–247	138–273
3	55–117	55–117	58–121	52–116	51–122
4	139–282	136–276	141–289	116–234	132–267
5	19–39	19–39	20–44	19–40	19–39
6	25–59	25–59	25–59	26–63	28–68
7	20–42	20–42	20–43	19–43	19–43
8	24–68	24–68	24–68	30–82	24–68
9	36–83	37–88	32–76	32–76	34–83
10	73–111	73–111	73–111	71–108	74–116
11	170–346	118–255	178–382	175–387	135–285
12	10–28	10–28	10–28	14–48	10–31
13	3–10	3–10	3–10	3–10	3–10
14	236–462	129–253	101–197	180–348	102–198
15	35–69	51–100	41–84	40–81	49–97
Σ	1213–2440	1081–2193	1187–2448	1145–2387	1065–2179

Table 2 contains results obtained for different initial estimates in the line search algorithm (for the choices $REST = 3$ and $SCAL = 2$). This table demonstrates the great effectiveness of the initial estimate (4.7) and it shows that (4.3) is better than both (4.1) and (4.2). Similar results were obtained also for other restart procedures.

Table 2

(PR)	$REST = 3,\ SCAL = 2$				
$n = 20$	$INIT = 1$	$INIT = 2$	$INIT = 3$	$INIT = 4$	$INIT = 5$
1	209–438	204–635	211–422	193–383	221–433
2	445–951*	305–1078	329–678	168-339	147–291
3	59–140	64–139	49–104	60–132	55–117
4	149–319	136–917	134–267	166–322	139–282
5	20–48	18–26	18–36	17–43	19–39
6	25–66	26–52	25–51	25–67	25–59
7	22–48	20–162	20–38	22–49	20–42
8	34–127	28–73	36–81	24–70	24–68
9	37–90	37–90	37–90	36–83	36–83
10	75–112	73–114	72–108	74–102	73–111
11	162–354	168–532	167–364	119–264	170–346
12	9–31	11–38	12–31	10–33	10–28
13	3–7	3–6	3–6	3–16	3–10
14	234–458	326–741	221–435	251–479	236–462
15	45–92	45–92	45–92	39–75	35–69
Σ	1528–3281	1464–4695	1379–2803	1207–2457	1213–2440

More extensive tests are presented in Table 3. Here the influence of initial estimates and scaling options is shown for all 7 restart procedures described in Section 4. We can see that the new restart procedures are very efficient in connection with the "standard" choices $INIT = 5$ and $SCAL = 2$, while the simplest restart procedures based on periodic restarts (first two rows in the table) perform worse in this case.

Table 3

(PR)	$INIT = 3$	$INIT = 3$	$INIT = 5$	$INIT = 5$
$n = 20$	$SCAL = 1$	$SCAL = 2$	$SCAL = 1$	$SCAL = 2$
$REST = 1$	1477–3818	1445–2928	1615–3234	1753–3480
$REST = 2$	1456–3796	1505–3040	1320–2650	1445–2890
$REST = 3$	1531–3889	1379–2803	1299–2623	1213–2440
$REST = 4$	1276–3320	1309–2652	1362–2728	1081–2193
$REST = 5$	1641–4164	1668–3409	1380–2764	1187–2448
$REST = 6$	1475–3858	1343–2818	1155–2370	1145–2387
$REST = 7$	1817–4541	1763–3553	1082–2231	1065–2179

Although almost all tests were performed for the PR method, Table 4 contains some experiments with other CG methods. Here the surprisingly good performance of the FR method with the choice $REST = 7$ is shown and the worse efficiency of the HS method in comparison with the PR method is demonstrated.

Table 4

	(FR) $INIT = 5,\ SCAL = 2$		(HS) $INIT = 5,\ SCAL = 2$	
$n = 20$	$REST = 3$	$REST = 7$	$REST = 3$	$REST = 7$
1	259–479	260–528	226–439	262–490
2	164–327	114–232	160–318	205–401
3	79–159	56–123	49–110	49–110
4	136–273	123–254	110–220	176–346
5	19–39	21–46	18–37	18–37
6	26–55	30–70	26–61	26–61
7	20–40	18–41	20–42	20–42
8	44–101	21–57	24–68	24–68
9	56–114	33–76	37–92	38–88
10	98–147	76–119	74–115	74–115
11	137–270	173–377	162–333	318–619
12	27–61	12–37	11–31	11–31
13	5–11	3–10	3–10	3–10
14	346–670	105–209	261–509	208–405
15	40–81	29–60	41–79	54–106
Σ	1456–2827	1074–2239	1222–2464	1486–2929

Table 5 shows the efficiency of individual restart procedures for 100 variables. Here a good robustness of the choice $REST = 7$ is demonstrated.

Table 5

(PR)	$INIT = 5,\ SCAL = 2$				
$n = 100$	$REST = 3$	$REST = 4$	$REST = 5$	$REST = 6$	$REST = 7$
1	649–1152	737–1298	854–1761	855–1756	829–1591
2	886–1718*	730–1409	986–1951*	1525–3063*	705–1377*
3	48–103	48–103	47–103	47–103	70–148
4	158–319	249–496	159–325	225–468	157–312
5	20–41	20–41	20–41	21–45	22–44
6	28–68	28–68	28–68	30–72	28–68
7	24–50	24–50	25–53	25–53	24–53
8	37–95	620–752	19–54	19–54	39–92
9	125–252	134–256	123–250	115–235	118–237
10	91–149	91–149	92–152	93–173	91–154
11	164–327	226–440	162–336	147–305	112–234
12	14–40	14–34	14–40	15–40	15–50
13	3–10	3–10	3–10	3–10	3–10
14	8819–17489	6707–13340	3197–6371	3051–6075	3197–6371
15	9–18	9–18	9–18	9–18	9–18
Σ	11075–21831	9640–18464	5738–11533	6180–12470	5419–10759

The influence of initial estimates and scaling options for 200 variables is shown in Table 6. Here the expressive efficiency of the initial estimate (4.7) and the scaling (4.9) is demonstrated.

Table 6

(PR)	$REST = 7$			
$n = 200$	$INIT = 3$ $SCAL = 1$	$INIT = 3$ $SCAL = 2$	$INIT = 5$ $SCAL = 1$	$INIT = 5$ $SCAL = 2$
1	1492–7073	1185–2374	1566–2993	1532–2902
3	53–107	51–107	52–114	52–118
4	157–375	132–277	161–314	137–276
5	24–45	27–53	21–50	20–46
6	29–54	30–59	29–66	30–67
7	24–55	24–49	24–55	25–54
10	47–123	48–99	47–134	46–89
11	96–222	879–1749	134–269	108–217
12	17–44	17–45	18–53	18–54
13	3–7	3–7	3–10	3–10
Σ	1942–8105	2396–4819	2055–4058	1971–3833

Finally our restarted CG method (the PR method with choices $REST = 3, INIT = 5$, and $SCAL = 2$) was compared with other optimization methods. Table 7 contains the comparison of this CG method with the 5-step limited memory (LM) BFGS method proposed in [19], with the difference version of the truncated Newton (TN) method given in [7], and with the partitioned variable metric (VM) method described in [12] implemented with an imperfect iterative solution of the linear system $Bs+g = 0$. Problem 10 and Problem 12 were modified for the computations yielding Table 7 since a partitioned structure was required.

Table 7

$n = 500$	CG	LM	TN	VM
1	2537–4052	2564–2819	823–10423	1497–1825
3	44–97	240–257	23–124	44–45
4	253–536	108–115	21–221	38–40
5	23–51	23–25	14–64	18–20
6	28–65	34–36	15–145	147–148
7	25–61	33–38	12–56	21–25
10	138–291	172–217	26–281	18–21
11	246–451	124–132	109–387	85–89
12	191–441	127–135	104–302	117–120
13	83–88	83–84	81–164	84–85
Σ	3568–6133	3508–3858	1228–12167	2069–2418
Time	7:32.86	7:11.17	12:49.67	11:49.69

Results in the last table show that the simple conjugate gradient method can be competitive with other more complicated methods.

(Received February 28, 1991.)

REFERENCES

[1] M. Al-Bali: Descent property and global convergence of the Fletcher-Reeves method with inexact line search. IMA J. Numer. Anal. *5* (1985), 121–124.
[2] P. Baptist and J. Stoer: On the relation between quadratic termination and convergence properties of minimization algorithms. Part 2. Applications. Numer. Math. *28* (1977), 367–391.
[3] P. Bjorstadt and J. Nocedal: Analysis of a new algorithm for one-dimensional minimization. Computing *22* (1979), 93–100.
[4] A. R. Conn, N. I. M. Gould and P. L. Toint: Testing a class of methods for solving minimization problems with simple bounds on the variables. Math. Comp. *50* (1988), 399–430.
[5] W. C. Davidon: Variable metric method for minimization. A.E.C. Research and Development Report ANL-5990, 1959.
[6] W. C. Davidon: Optimally conditioned optimization algorithms without line searches. Math. Programming *9* (1975), 1–30.
[7] R. S. Dembo and T. Steihaug: Truncated-Newton algorithms for large-scale unconstrained minimization. Math. Programming *26* (1983), 190–212.
[8] R. Fletcher: A FORTRAN subroutine for minimization by the method of conjugate gradients. Report No. AERE-R7073, Atomic Energy Research Establishment, Harwell 1972.
[9] R. Fletcher and M. J. D. Powell: A rapidly convergent descent method for minimization. Computer J. *6* (1963), 163–168.
[10] R. Fletcher and C. M. Reeves: Function minimization by conjugate gradients. Computer J. *7* (1964), 149–154.
[11] J. C. Gilbert and J. Nocedal: Global convergence properties of conjugate gradient methods for optimization. Report No. 1268, Institut National de Recherche en Informatique et en Automatique, 1990.
[12] A. Griewank and P. L. Toint: Partitioned variable metric updates for large structured optimization problems. Numer. Math. *39* (1982), 119–137.
[13] M. R. Hestenes and C. M. Stiefel: Methods of conjugate gradient for solving linear systems. J. Res. Nat. Bur. Standards *49* (1964), 409–436.
[14] Y. F. Hu and C. Storey: A Global Convergence Result for Conjugate Gradient Methods. Report No. A134, Loughborough University of Technology, 1990.
[15] K. M. Khoda, Y. Liu and C. Storey: A Generalized Polak-Ribire Algorithm. Report No. A128, Loughborough University of Technology, 1990.
[16] L. Lukšan: Variable Metric Methods. Unconstrained Minimization. Academia, Prague 1990. In Czech.
[17] L. Lukšan: Computational experience with improved variable metric methods for unconstrained minimization. Kybernetika *26* (1990), 415–431.
[18] J. J. Moré, B. S. Garbow and K. E. Hillström: Testing unconstrained optimization software. ACM Trans. Math. Software *7* (1981), 17–41.
[19] J. Nocedal: Updating quasi-Newton matrices with limited storage. Math. Comp. *35* (1980), 773–782.
[20] E. Polak and G. Ribire: Note sur la convergence de methodes de directions conjugees. Revue Francaise Inform. Mech. Oper. *16-R1* (1969), 35–43.
[21] M. J. D. Powell: Restart procedures of the conjugate gradient method. Math. Programming *12* (1977), 241–254.
[22] M. J. D. Powell: Nonconvex Minimization Calculations and the Conjugate Gradient Method. Report No. DAMTP 1983/NA14, University of Cambridge, 1983.

[23] M .J. D. Powell: Convergence Properties of Algorithms for Nonlinear Optimization. Report No. DAMPT 1985/NA1, University of Cambridge, 1985.
[24] D. F. Shanno: Conditioning of quasi-Newton methods for function minimization. Math. Comp. *24* (1970), 647–656.
[25] D. F. Shanno: Globally convergent conjugate gradient algorithms. Math. Programming *33* (1985), 61–67.
[26] P. L. Toint: On sparse and symmetric matrix updating subject to a linear equation. Math. Comp. *31* (1987), 954–961.
[27] D. Touati-Ahmed and C. Storey: Efficient hybrid conjugate gradient techniques. J. Optim. Theory Appl. *64* (1990), 379–397.
[28] G. Zoutendijk: Nonlinear programming, computational methods. In: Integer and Nonlinear Programming (J. Abadie, ed.), North-Holland, Amsterdam 1970, pp. 93–121.

Ing. Ladislav Lukšan, DrSc., Ústav informatiky a výpočetní techniky ČSAV (Institute of Computer and Information Science – Czechoslovak Academy of Sciences), Pod vodárenskou věží 2, 182 07 Praha 8. Czechoslovakia.