Journal of Global Optimization (2005) 31:287–315

An Inexact Newton Method Derived from Efficiency Analysis*

NAIYANG DENG[1], YI XUE[2], JIANZHONG ZHANG[3] and PING ZHONG[1]
[1]*China Agricultural University, Beijing 100083, China (E-mail: dengnaiyang@vip.163.com)*
[2]*Beijing Polytechnic University, Beijing 100022, China*
[3]*City University of Hong Kong, Hong Kong*

(Received 30 March 2004 accepted 1 April 2004)

Abstract. We consider solving an unconstrained optimization problem by Newton-PCG like methods in which the preconditioned conjugate gradient method is applied to solve the Newton equations. The main question to be investigated is how efficient Newton-PCG like methods can be from theoretical point of view. An algorithmic model with several parameters is established. Furthermore, a lower bound of the efficiency measure of the algorithmic model is derived as a function of the parameters. By maximizing this lower bound function, the parameters are specified and therefore an implementable algorithm is obtained. The efficiency of the implementable algorithm is compared with Newton's method by theoretical analysis and numerical experiments. The results show that this algorithm is competitive.

Mathematics Subject Classification: 90C30, 65K05.

Key words: Cholesky factorization, Efficiency coefficient, Newton equation, Preconditioned conjugate gradient method, Unconstrained optimization.

1. Introduction

We consider local algorithms for the unconstrained optimization problem

$$\min f(x), \quad x \in R^n. \tag{1.1}$$

If the Hessian matrix is available, Newton-like method is popular for solving the problem (see e.g. [6] and [9]). In every iteration, its computation cost consists of the following two parts: (1) evaluating one Hessian $\nabla^2 f$ and one gradient ∇f which form a Newton equation; (2) solving a Newton equation. However, for a significant number of widely different applications, the cost of part (2) tends to be dominate (see e.g. [3]). We confine ourselves to such case in this paper.

*This work was supported by the National Science Foundation of China Grant No. 10371131, and Hong Kong Competitive Earmarked Research Grant CityU 1066/00P from Hong Kong University Grant Council

A number of modifications have been proposed to improve the original Newton's method. An interesting approach is Newton-CG like algorithms, which are based on the analysis on the inexact Newton method in [4] and the properties of conjugate gradient method (CG method) (see e.g. [14]). First, Steihaug and Toint proposed Newton-CG like algorithms (see [18] and [19]), in which the Newton equation is solved by CG method approximately. Later, these algorithms were well developed to form a kind of Newton-PCG algorithms, in which the CG method is replaced by PCG method (preconditioned CG method). In a PCG method, using good preconditioning strategy is important. Some general-purpose preconditioners have been proposed. The most important strategies of this type include symmetric successive overrelaxation (SSOR) (see e.g. [17]), incomplete Cholesky factorization (see e.g. [10]) and banded preconditioners (see e.g. [2]). It has been shown by a large amount of experiments that, generally speaking, Newton-PCG like algorithms are very successful (see e.g. [2, 7, 8, 13]. However, their success varies greatly from problem to problem (see e.g. [14]), and the superiority of this type of algorithms is usually shown by numerical results alone.

Following [5], this paper investigates further the question that how efficient the Newton-PCG like methods can be. Our research is undertaken mainly from a theoretical point of view via a rather rigorous analysis on the efficiency issue, but it is also supported by experimental results. In [5], a Newton-PCG like algorithm was proposed in which the termination criterion for the subiterations in the PCG step is

$$||r(s^k)|| = ||\nabla^2 f(x^k)s^k + \nabla f(x^k)|| \leqslant ||\nabla f(x^k)||^{2+\epsilon}, \tag{1.2}$$

where ϵ is a small positive scalar. The conclusions of [5] are obtained under rather strong assumptions, including that for any nonzero $h \in R^n$, $\nabla^3 f(x^*)$ satisfies

$$\nabla^3 f(x^*)hh \neq 0, \tag{1.3}$$

where x^* is the solution to (1.1).

In this paper, a general algorithmic model is established. Its behavior, including the convergence speed, is flexible. For example, there is no any restriction on the one-step convergence order for the sequence generated by the algorithmic model, while the precisely quadratic convergence is enforced in [5]. Instead of assigning the power in the termination criterion (1.2) to be $2 + \epsilon$ in [5], the corresponding power in the algorithmic model is a parameter, and the value of which can be selected. In fact the algorithm depends on a set of parameters, and the selection of the parameters should maximize the efficiency, which is a key point for the success of the algorithm. It is observed that some commonly used efficiency indexes are too restrictive to fit into our algorithmic model. Fortunately, efficiency of iterative methods was thoroughly investigated by Brent in [1] in which an

exact measure of efficiency was given that is applicable to our study. This measure forms one of the foundations of this paper, and according to it, the parameters are specified. Thus the implementable algorithm obtained in this way is more efficient than the algorithm in [5].

It should be pointed out that the assumption (1.3) has been removed in this paper. In fact, we only use the following standard assumptions:

ASSUMPTION (A1): $\nabla^2 f(x)$ is Lipschitz continuous with the constant L in a neighborhood of the solution x^* to (1.1);

ASSUMPTION (A2): $\nabla^2 f(x^*)$ is symmetric positive definite.

This paper is organized as follows. In Section 2, the algorithmic model containing several parameters is proposed. In Section 3, we derive a lower bound for the efficiency of the algorithmic model, which is a function of the parameters. By maximizing this lower bound, an implementable algorithm is established in Section 4, and the efficiency of which is also compared with Newton's method from a theoretical point of view in the same section. Section 5 contains main results of our numerical experiment.

For convenience of expression, we adopt the following convention:

$$\sum_{\alpha}^{\beta} \cdots = 0, \text{ when } \alpha > \beta. \tag{1.4}$$

2. A New Algorithmic Model

The algorithmic model that we propose here is a class of Newton-PCG like methods in which the Newton equation is solved either by a CF step or by a PCG step. The PCG step is obtained by using the standard preconditioned conjugate gradient method. More precisely, the following Algorithm PCG(C, A, b, l, e) is used to solve the linear system

$$As = b, \tag{2.1}$$

where C is the preconditioner, l is the maximum number of subiterations, and e is a scalar used in the termination criterion.

ALGORITHM PCG(C, A, b, l, e)

Step 0. Initial data: set the initial point $s_0 = 0, r_0 = -b$. Set $i = 1$.

Step 1. Termination test : if

$$||r_{i-1}|| \leqslant ||b||^{1+e} \quad \text{or} \quad i - 1 = l, \tag{2.2}$$

go to Step 4.

Step 2. Subiteration:

(a) set $z = Cr_{i-1}, t_{i-1} = z^T r_{i-1}$;

(b) if $i = 1$, then set $q = z$; otherwise set $\beta = t_{i-1}/t_{i-2}$ and $q = z + \beta q$;

(c) set $s_i = s_{i-1} + \lambda q$, where $\lambda = t_{i-1}/q^T w$ and $w = Aq$;

(d) set $r_i = r_{i-1} + \lambda w$.

Step 3. Set $i = i + 1$ and go to step 1.
Step 4. Set $\tilde{s} = s_{i-1}$.

Using Algorithm PCG(·), the following algorithmic model, called Algorithm CF-PCG($p; l_1 \ldots, l_p; \alpha_1, \ldots, \alpha_p$), is established, where $p, l_1, \ldots, l_p$, $\alpha_1, \ldots, \alpha_p$ are parameters: p is such a nonnegative integer that every CF step is followed by p PCG steps; the positive integers $l_1, \ldots, l_p$ are respectively the maximum numbers of subiterations in these PCG steps; the positive scalars $\alpha_1, \ldots, \alpha_p$ are used in the termination test of these PCG steps.

ALGORITHM CF-PCG($p; l_1, \ldots, l_p; \alpha_1, \ldots, \alpha_p$)
Step 0. Initial data: set the initial point $x^0 \in R^n$. Set $k = 0$.
Step 1. Termination test : if $\nabla f(x^k) = 0$, stop.
Step 2. Switch test: if k can be divided by $p + 1$ with no remainder, go to Step 3; Otherwise go to Step 4.
Step 3. CF Step: set

$$B^k = \nabla^2 f(x^k). \tag{2.3}$$

Find the solution s^k to the Newton equation

$$\nabla^2 f(x^k)s = -\nabla f(x^k) \tag{2.4}$$

by Cholesky factorization $\nabla^2 f(x^k) = L_k D_k L_k^T$. Set $m = 0$, then go to Step 5.
Step 4. PCG Step: set $B^k = B^{k-1}$ and $m = m + 1$. Find $\tilde{s}$ by Algorithm PCG$((B^k)^{-1}, \nabla^2 f(x^k), -\nabla f(x^k), l_m, l_m/\alpha_m)$. Set $s^k = \tilde{s}$.
Step 5. Update the iterate: set $x^{k+1} = x^k + s^k$. Set $k = k + 1$ and go to Step 1.

We write the sequence $\{x^k\}$ generated by Algorithm CF-PCG($p; l_1, \ldots, l_p, \alpha_1, \ldots, \alpha_p$) as

$$\{x^k\} = \{x_{CF}^{0(p+1)}, x_{PCG}^{0(p+1)1}, \ldots, x_{CF}^{j(p+1)}, x_{PCG}^{j(p+1)+1}, \ldots, x_{PCG}^{j(p+1)+p}, x_{CF}^{(j+1)(p+1)}, \ldots\} \tag{2.5}$$

where the subscripts CF or PCG are used to show which step (CF Step or PCG Step) is executed at the corresponding iterate (but these subscripts are often omitted in later sections in order to simplify notations).

3. The Efficiency Analysis of Algorithm CF-PCG($p; l_1, \ldots, l_p; \alpha_1, \ldots, \alpha_p$)

The efficiency of Algorithm CF-PCG($p; l_1, \ldots; l_p; \alpha_1, \ldots; \alpha_p$) is analyzed in this section under Assumptions (A1) and (A2) described in Section 1. The main purpose is to derive a lower bound of its efficiency which will be given in Theorem 3.9 below. First we cite a definition of the efficiency given by Brent [1] and define "Progress index" which is critical to the proof of Theorem 3.9.

DEFINITION 3.1.([1]) Efficiency coefficient: suppose that the sequence $\{x_0, x^1, \ldots, x^k, \ldots\}$ is generated by an algorithm. If $\{x^k\}$ converges to the solution x^* to (1.1), then the efficiency coefficient Γ of the algorithm is defined by

$$\Gamma = \liminf_{k\to\infty} \frac{\ln(-\ln ||x^k - x^*||)}{\sum_{i=1}^{k} Q[x^{i-1}, x^i]}, \tag{3.1}$$

where $Q[x^{i-1}, x^i]$ is the computation cost required to compute x^i from x^{i-1}.

DEFINITION 3.2. Progress index : suppose both x_{CF} and x_c are near the solution x^* to (1.1). The progress index $v = v[x_{CF}, x_c]$ from x_{CF} to x_c with respect to x^* is deffined as

$$v = v[x_{CF}, x_c] = \frac{\ln ||x_c - x^*||}{\ln ||x_{CF} - x^*||}. \tag{3.2}$$

The remainder of this section is to estimate the efficiency coefficient of Algorithm CFPCG($p; l_1, \ldots, l_p, \alpha_1, \ldots, \alpha_p$) and to derive Theorem 3.9 via the following five lemmas.

LEMMA 3.3. *Assume that* $x_+ = x_{CF} + s_{CF}$, *where* s_{CF} *is the solution to Newton equation*

$$\nabla^2 f(x_{CF})^s = -\nabla f(x_{CF}).$$

Then there exists $\delta \in (0, 1)$ *such that for the solution* x^* *to (1.1), when* $||x_{C_F} - x^*|| \leqslant \delta$, *we have*

$$v[x_{CF}, x_+] \geqslant 2 + \theta_1, \tag{3.3}$$

where $\theta_1 = \frac{\ln \gamma'}{\ln ||x_{CF} - x^*||} < 0, \gamma' \geqslant 1$ *is a constant which depends only on* $\nabla^2 f(x^*)$ *and the Lipschitz constant L.*

Proof. It is well known (see e.g. [6]) that Newton's method is locally convergent with at least q-order 2. More precisely, when δ is small enough, we have

$$||x_+ - x^*|| = ||x_{CF} - \nabla^2 f(x_{CF})^{-1} \nabla f(x_{CF}) - x^*|| \leqslant \gamma' ||x_{CF} - x^*||^2, \tag{3.4}$$

where $\gamma' \geqslant 1$ is a constant which depends only on $\nabla^2 f(x^*)$ and L. Thus, from (3.4) and Definition 3.2, we obtain (3.3). □

LEMMA 3.4 *Let the progress index v from* x_{CF} *to* x_c *with respect to* x^* *satisfy*

$$v = v[x_{CF}, x_c] \geqslant 1. \tag{3.5}$$

Assume that $\tilde{s}$ *is obtained by Algorithm PCG*$(\nabla^2 f(x_{CF})^{-1}, \nabla^2 f(x_c), -\nabla f(x_c), l, l/\alpha)$ with $\alpha \geqslant 1$. *Then there exists* $\delta \in (0, 1)$ *such that when*

$||x_{CF} - x^*|| \leqslant \delta$ and $||x_c - x^*|| \leqslant \delta$, *the residual* $r(\tilde{s}) = \nabla^2 f(x_c)\tilde{s} + \nabla f(x_c)$ *satisfies*

$$||r(\tilde{s})|| \leqslant \gamma ||\nabla f(x_c)||^{1+l/\max(v,\alpha)}, \tag{3.6}$$

where γ *is a constant which depends only on* $\nabla^2 f(x^*)$ *and the Lipschitz constant* L.

Proof. Without loss of generality, we assume $||\nabla f(x_c)|| < 1$. According to the termination condition (2.2), there are two possibilities: either

$$||r(\tilde{s})|| \leqslant ||\nabla f(x_c)||^{1+l/\alpha}, \tag{3.7}$$

or the number of subiterations is l with $l \leqslant n$, and in this case,

$$\tilde{s} = s_l. \tag{3.8}$$

The former case implies that the inequality (3.6) is true with $\gamma = 1$. So we only need to show (3.6) for the latter case.

First, we estimate the difference between $(\nabla^2 f(x_c))_{ij}$ and $(\nabla^2 f(x_{CF}))_{ij}$, where $(\cdot)_{ij}$ is the element in the i-th row and the j-th column of the matrix. By Assumption (A1), Definition 3.2 and (3.5), we see that when δ is small enough,

$$\begin{aligned} |(\nabla^2 f(x_c))_{ij} - (\nabla^2 f(x_{CF}))_{ij}| &\leqslant L||x_c - x_{CF}|| \leqslant L(||x_c - x^*|| + ||x_{CF} - x^*||) \\ &= L(||x_c - x^*|| + ||x_c - x^*||^{1/v}) \leqslant 2L||x_c - x^*||^{1/v}. \end{aligned} \tag{3.9}$$

On the other hand, by Assumption (A2), we conclude that when δ is small enough,

$$||\nabla f(x_c)|| = ||\nabla f(x_c) - \nabla f(x^*)|| \geqslant \frac{1}{2}\gamma_1 ||x_c - x^*||, \tag{3.10}$$

where $\gamma_1 = \min(\lambda_{\min}, 1)$, and $\lambda_{\min}$ is the smallest eigenvalue of the matrix $\nabla^2 f(x^*)$. It follows from (3.9) and (3.10) that

$$|(\nabla^2 f(x_c))_{ij} - (\nabla^2 f(x_{CF}))_{ij}| \leqslant 2L(2/\gamma_1)^{1/v} ||\nabla f(x_c)||^{1/v} \leqslant \gamma_2 ||\nabla f(x_c)||^{1/v}, \tag{3.11}$$

where $\gamma_2 = 4L/\gamma_1$, which is independent of v.

Second, define

$$\begin{aligned} A &= (\nabla^2 f(x_{CF}))^{-1/2} \nabla^2 f(x_c) (\nabla^2 f(x_{CF}))^{-1/2} \\ &= I + (\nabla^2 f(x_{CF}))^{-1/2} [\nabla^2 f(x_c) - \nabla^2 f(x_{CF})] (\nabla^2 f(x_{CF}))^{-1/2}. \end{aligned} \tag{3.12}$$

By Assumptions (A1) and (A2), we know that when δ is small enough A is symmetric positive definite. Now we estimate its condition number $k_2(A)$. Let

$$\gamma_3 = 2\max\{|(\nabla^2 f(x^*)^{-1/2})_{ij}||1 \leqslant i,j \leqslant n\}.$$

It is easy to see that, when δ is small enough,

$$|(\nabla^2 f(x_{CF})^{-1/2})_{ij}| \leqslant \gamma_3, \quad \forall i,j. \tag{3.13}$$

So from (3.13) and (3.11),

$$|(\nabla^2 f(x_{CF})^{-1/2}[\nabla^2 f(x_c) - \nabla^2 f(x_{CF})]\nabla^2 f(x_{CF})^{-1/2})_{ij}| \leqslant (\gamma_4/n)||\nabla f(x_c)||^{1+\nu}, \tag{3.14}$$

where $\gamma_4 = 2n^3\gamma_2\gamma_3^2$. By Gerschgorin Theorem (see, e.g. Theorem 2.3.9 in [15]), (3.12), and (3.14), we claim that every eigenvalue λ of A satisfies

$$1 - \gamma_4||\nabla f(x_c)||^{1/\nu} \leqslant \lambda \leqslant 1 + \gamma_4||\nabla f(x_c)||^{1/\nu}. \tag{3.15}$$

Therefore, the condition number $\kappa_2(A)$ has the following estimate

$$\kappa_2(A) \leqslant \frac{1 + \gamma_4||\nabla f(x_c)||^{1/\nu}}{1 - \gamma_4||\nabla f(x_c)||^{1/\nu}}. \tag{3.16}$$

Thus, noticing $\lim_{\delta \to 0} ||\nabla f(x_c)|| = 0$, we conclude that when δ is small enough,

$$\kappa_2(A) \leqslant 2(1 + \gamma_4), \tag{3.17}$$

$$\kappa_2(A) - 1 \leqslant \frac{2\gamma_4||\nabla f(x_c)||^{1/\nu}}{1 - \gamma_4||\nabla f(x_c)||^{1/\nu}} \leqslant 4\gamma_4||\nabla f(x_c)||^{1/\nu}. \tag{3.18}$$

Third, consider the case of solving the linear system

$$A\bar{s} = \bar{b} \tag{3.19}$$

by the conjugate gradient method (without a preconditioner), where A is defined by (3.12) and

$$\bar{b} = -(\nabla^2 f(x_{CF}))^{-1/2}\nabla f(x_c). \tag{3.20}$$

Suppose that the initial point $\bar{s_0} = 0$ and $\bar{s_l}$ is the approximate solution obtained after l subiterations. Let us estimate the residual

$$\bar{r}(\bar{s_l}) = A\bar{s_l} - \bar{b}. \tag{3.21}$$

It is not difficult to see that for all $z \in R^n$,

$$\sqrt{\lambda_{\min}}||z||_A \leqslant ||A_z|| \leqslant \sqrt{\lambda_{\max}}||z||_A, \tag{3.22}$$

where $\lambda_{\max}$ and $\lambda_{\min}$ are respectively the largest and smallest eigenvalues of A. Therefore,

$$\frac{||\bar{r}(\bar{s_l})||}{||\bar{b}||} = \frac{||A\bar{s_l} - \bar{b}||}{||A\bar{s_0} - \bar{b}||} = \frac{||A(\bar{s_l} - \bar{s}^*)||}{||A(0 - \bar{s}^*)||} \leqslant \frac{\sqrt{\lambda_{\max}} \cdot ||\bar{s_l} - \bar{s}^*||_A}{\sqrt{\lambda_{\min}} \cdot ||0 - \bar{s}^*||_A}, \tag{3.23}$$

$$||\bar{r}(\bar{s_l})||/||\bar{b}|| \leqslant \sqrt{k_2(A)} \cdot ||\bar{s_l} - \bar{s}^*||A/||0 - \bar{s}^*||_A, \tag{3.24}$$

where $\bar{s}^*$ is the solution to (3.19). It is well known that, for the conjugate gradient method the following result holds (see, e.g. [11])

$$||\bar{s_l} - \bar{s}^*||_A \leqslant 2||0 - \bar{s}^*||_A \left(\frac{\sqrt{\kappa_2(A)} - 1}{\sqrt{\kappa_2(A)} + 1}\right)^l. \tag{3.25}$$

Combining (3.24), (3.25), (3.20), (3.17) and (3.18), we have

$$\begin{aligned} ||\bar{r}(\bar{s_l})|| &\leqslant 2||\bar{b}|| \cdot \sqrt{k_2(A)} \cdot \left(\frac{\sqrt{\kappa_2(A)} - 1}{\sqrt{\kappa_2(A)} + 1}\right)^l \\ &\leqslant 2||\nabla^2 f(x_{CF})^{-1/2}|| \cdot ||\nabla f(x_c)|| \cdot \sqrt{\kappa_2(A)}(\sqrt{\kappa_2(A)} - 1)^l \\ &\leqslant 2||\nabla^2 f(x_{CF})^{-1/2}||\sqrt{2(1+\gamma_4)}(4\gamma_4)^l||\nabla f(x_c)||^{1+l/v} \\ &\leqslant \gamma_5||\nabla f(x_c)||^{1+l/v}, \end{aligned} \tag{3.26}$$

where $\gamma_5 = 4\sqrt{2(1+\gamma 4)}(4\gamma_4)^n \cdot ||\nabla^2 f(x^*)^{-1/2}||$.

Finally, we prove (3.6). Noticing (3.8), it is sufficient to prove

$$||r(s_l)|| \leqslant \gamma||\nabla f(x_c)||^{1+l/v}. \tag{3.27}$$

In fact, it is not difficult to show that

$$\bar{s_l} = \nabla^2 f(x_{CF})^{1/2} s_l,$$

and

$$r(s_l) = (\nabla^2 f(x_{CF}))^{1/2} \bar{r}(\bar{s_l}). \tag{3.28}$$

Therefore, from (3.28) and (3.26), we get (3.27) with $\gamma = 2\gamma_5||(\nabla^2 f(x^*))^{1/2}||$ provided that δ is small enough. □

LEMMA 3.5. *Let the condition* (3.5) *in Lemma* 3.4 *be held. Assume that* $x_+ = x_c + \tilde{s}$, *where* $\tilde{s}$ *is defined in Lemma 3.4. Then there exists* $\delta \in (0,1)$ *such that for the solution* x^* *to* (1.1), *when* $||x_{CF} - x^*|| \leqslant \delta$ *and* $||x_c - x^*|| \leqslant \delta$, we have

$$v[x_{CF}, x_+] \geqslant \psi(l, \alpha, v[x_{CF}, x_c]) + \theta_2, \tag{3.29}$$

where $\theta_2 = \frac{\ln \gamma''}{\ln ||x_{CF} - x^*||} < 0, \gamma'' \geqslant 1$ *is a constant which depends only on* $\nabla^2 f(x^*)$ *and the Lipschitz constant* L, *and*

$$\psi(l, \alpha, v) = \begin{cases} v + \min(1, \frac{v}{\alpha})l, & \text{if } l \leqslant \max(v, \alpha); \\ 2v, & \text{if } l > max(v, \alpha). \end{cases} \tag{3.30}$$

Proof. Obviously, (3.4) is still true when the subscript CF is replaced by c. Therefore, from (3.5) and Lemma 3.4, we know that when δ is small enough,

$$\begin{aligned}||x_+ - x^*|| &= ||x_c - \nabla^2 f(x_c)^{-1}\nabla f(x_c) - x^* + \nabla^2 f(x_c)^{-1} r(\tilde{s})|| \\ &\leqslant \gamma' ||x_c - x^*||^2 + 2||\nabla^2 f(x^*)^{-1}|| \cdot ||r(\tilde{s})|| \\ &\leqslant \gamma' ||x_c - x^*||^2 + 2\gamma ||\nabla^2 f(x^*)^{-1}||\ ||\nabla f(x_c)||^{1 + \frac{l}{\max(v[x_{CF}, x_c], \alpha)}}.\end{aligned} \tag{3.31}$$

However, by Assumption (A2), we know that when δ is small enough,

$$||\nabla f(x_c)|| = ||\nabla f(x_c) - \nabla f(x^*)|| \leqslant 2\,\lambda_{\max} ||x_c - x^*||, \tag{3.32}$$

where $\lambda_{\max}$ is the largest eigenvalue of $\nabla^2 f(x^*)$. Therefore,

$$||x_+ - x^*|| \leqslant \begin{cases} \gamma'' ||x_c - x^*||^{1 + l/\max(v[x_{CF}, x_c], \alpha)}, & \text{if } l \leqslant \max(v[x_{CF}, x_c], \alpha); \\ \gamma'' ||x_c - x^*||^2, & \text{if } l > \max(v[x_{CF}, x_c], \alpha), \end{cases} \tag{3.33}$$

where $\gamma'' = \gamma' + 8\gamma ||\nabla^2 f(x^*)^{-1}|| \cdot \max(1, \lambda_{\max}^2)$.

Next we examine the cases $l \leqslant \max(v[x_{CF}, x_c], \alpha)$ and $l > \max(v[x_{CF}, x_c], \alpha)$ separately. For the case $l \leqslant \max(v[x_{CF}, x_c], \alpha)$, we conclude from (3.33) that, when δ is small enough,

$$v[x_c, x_+] = \frac{\ln ||x_+ - x^*||}{\ln ||x_c - x^*||} \geqslant 1 + \frac{l}{\max(v[x_{CF}, x_c], \alpha)} + \frac{\ln \gamma''}{\ln ||x_c - x^*||}. \tag{3.34}$$

Therefore,

$$\begin{aligned} v[x_{CF}, x_+] &= v[x_{CF}, x_c] \cdot v[x_c, x_+] \\ &\geqslant v[x_{CF}, x_c] + \min(1, v[x_{CF}, x_c]/\alpha) l + \frac{\ln \gamma''}{\ln ||x_{CF} - x^*||}. \end{aligned} \tag{3.35}$$

For the case $l > \max(v[x_{CF}, x_c], \alpha)$, we conclude from (3.33) that, when δ is small enough,

$$v[x_c, x_+] = \frac{\ln ||x_+ - x^*||}{\ln ||x_c - x^*||} \geqslant 2 + \frac{\ln \gamma''}{\ln ||x_c - x^*||}. \tag{3.36}$$

Therefore

$$v[x_{CF}, x_+] = v[x_{CF}, x_c] \cdot v[x_c, x_+] \geqslant 2v[x_{CF}, x_c] + \frac{\ln \gamma''}{\ln ||x_{CF} - x^*||}. \tag{3.37}$$

Combining (3.35) and (3.37) yields (3.29).

LEMMA 3.6. *Let the first $p+2$ iterates generated by Algorithm CF–PCG$(p; l_1, \ldots, l_p; \alpha_1, \ldots, \alpha_p)$ be denoted by $\{x_{CF}, x_1, x_2, \ldots, x_{q+1}, \ldots, x_p, x_{p+1}\}$, where $x_{CF} = x^0$ is the initial point. Then*

$$\liminf_{x_{CF} \to x^*} v[x_{CF}, x_{q+1}] \geqslant \underline{v}_{q+1}, \quad q = 0, 1, \ldots, p, \tag{3.38}$$

where x^* *is the solution to* (1.1) *and* $\underline{v}_{q+1} = \underline{v}_{q+1}(l_1, \ldots, l_q, \alpha_1, \ldots, \alpha_q)$, $q = 0, 1, \ldots, p$, *are defined by* $\psi(\cdot)$ *given in* (3.30) *recursively:*

$$\underline{v}_1 \stackrel{def}{=} 2, \tag{3.39}$$

$$\underline{v}_{q+1} = \underline{v}_{q+1}(l_1, \ldots, l_q, \alpha_1, \ldots, \alpha_q) \stackrel{def}{=} \psi(l_q, \alpha_q, \underline{v}_q), \quad q = 1, \ldots, p. \tag{3.40}$$

Proof. Only the case $p \geqslant 1$ is considered here since the conclusion is obviously true for the case $p = 0$ by Lemma 3.3. Note that when $q = 0$, (3.38) can be obtained directly again by Lemma 3.3. So it is sufficient to assume the validity of (3.38) with $q = i - 1 (0 \leqslant q \leqslant p - 1)$:

$$\liminf_{x_{CF} \to x^*} v[x_{CF}, x_i] \geqslant \underline{v}_i \tag{3.41}$$

and to show its validity with $q = i$:

$$\liminf_{x_{CF} \to x^*} v[x_{CF}, x_{i+1}] \geqslant \underline{v}_{i+1}. \tag{3.42}$$

In fact, by Lemma 3.5, with x_c, x_+, l and α there being substituted by x_i, x_{i+1}, l_i and α_i, respectively, we conclude that

$$v[x_{CF}, x_{i+1}] \geqslant \psi(l_i, \alpha_i, v[x_{CF}, x_i]) + \theta_2, \tag{3.43}$$

where θ_2 is defined in Lemma 3.5. Therefore,

$$\liminf_{x_{CF} \to x^*} v[x_{CF}, x_{i+1}] \geqslant \psi(l_i, \alpha_i, v[x_{CF}, x_i]). \tag{3.44}$$

However, it is not difficult to see that the function $\psi(\cdot, \cdot, v)$ is continuous and strictly increasing with respect to v. Thus, combining (3.44) and (3.41) yields that

$$\liminf_{x_{CF} \to x^*} v[x_{CF}, x_{i+1}] \geqslant \psi(l_i, \alpha_i, \underline{v}_i). \tag{3.45}$$

Note that, by the definition (3.40), the right-hand-side of the above inequality is just $\underline{v}_{i+1}$. The proof is completed.

□

The next lemma answers the question that how the lower bound $\underline{v}_{p+1}$ in (3.38) depends on $l_t (t = 1, \ldots, p)$. It is shown that, roughly speaking, there is a critical value such that when l_t is larger than this value, a further increment of l_t cannot increase $\underline{v}_{p+1}$.

LEMMA 3.7. *Using the convention* (1.4), *the function* $\underline{v}_{p+1} = \underline{v}_{p+1}(l_1, \ldots, l_p, \alpha_1, \ldots, \alpha_p)$ *defined by* (3.39) –(3.40) *has the following properties.*
(1) *If*

$$\alpha_m \leqslant 2 + \sum_{t=1}^{m-1} l_t, \quad m = 1, \ldots, p, \tag{3.46}$$

and

$$l_m \leqslant 2 + \sum_{t=1}^{m-1} l_t, \quad m = 1, \ldots, p, \tag{3.47}$$

then

$$\underline{v}_{p+1}(l_1, \ldots, l_p, \alpha_1, \ldots, \alpha_p) = 2 + \sum_{t=1}^{p} l_t; \tag{3.48}$$

(2) *Define*

$$l'_m = \min\{l_m, 2 + \sum_{t=1}^{m-1} l'_t\}, \quad m = 1, \ldots, p, \tag{3.49}$$

then

$$\underline{v}_{p+1}(l_1, \ldots, l_p, \alpha_1, \ldots, \alpha_p) \leqslant 2 + \sum_{t=1}^{p} l'_t. \tag{3.50}$$

Furthermore, if

$$\alpha_m \leqslant 2 + \sum_{t=1}^{m-1} l'_t, \quad m = 1, \ldots, p, \tag{3.51}$$

then (3.50) *is strengthened as*

$$\underline{v}_{p+1}(l_1, \ldots, l_p, \alpha_1, \ldots, \alpha_p) = 2 + \sum_{t=1}^{p} l'_t. \tag{3.52}$$

Proof. (1) To show (3.48), it is sufficient to prove, by induction, that

$$\underline{v}_{q+1}(l_1, \ldots, l_q, \alpha_1, \ldots, \alpha_q) = 2 + \sum_{t=1}^{q} l_t, \quad q = 0, 1, \ldots, p. \tag{3.53}$$

The definition (3.39) implies that the above equality with $q = 0$ is true. So the first step of the induction is completed.

As the second step, we assume the validity of (3.53) with $q = i - 1$ $(0 \leqslant q \leqslant p - 1)$

$$\underline{v}_i(l_1, \ldots, l_{i-1}, \alpha_1, \ldots, \alpha_{i-1}) = 2 + \sum_{t=1}^{i-1} l_t \tag{3.54}$$

and show the validity with $q = i$:

$$\underline{v}_{i+1}(l_1, \ldots, l_i, \alpha_1, \ldots, \alpha_i) = 2 + \sum_{t=1}^{i} l_t. \tag{3.55}$$

In fact, by the definition (3.40), (3.54) and (3.30),

$$\begin{aligned}
&\underline{v}_{i+1}(l_1,\ldots,l_i,\alpha_1,\ldots,\alpha_i)\\
&\quad=\psi(l_i,\alpha_i,\underline{v}_i)=\psi\left(l_i,\alpha_i,2+\sum_{t=1}^{i-1}l_t\right)\\
&\quad=\begin{cases}2+\sum_{t=1}^{i-1}l_t+\min\left(1,\frac{2+\sum_{t=1}^{i-1}l_t}{\alpha_i}\right)l_i, & \text{if } l_i\leqslant\max(2+\sum_{t=1}^{i-1}l_t,\alpha_i);\\ 2(2+\sum_{t=1}^{i-1}l_t), & \text{if } l_i>\max(2+\sum_{t=1}^{i-1}l_t,\alpha_i).\end{cases}
\end{aligned}\tag{3.56}$$

Note that the conditions (3.47) and (3.46) imply that

$$l_i\leqslant\max\left(2+\sum_{t=1}^{i-1}l_t,\alpha_i\right),$$

and

$$\frac{2+\sum_{t=1}^{i-1}l_t}{\alpha_i}\geqslant 1.$$

Therefore (3.55) is obtained from the above two inequalities and (3.56). Thus, the equality (3.48) is proved.

(2) We prove (3.52) first, and then (3.50). In order to prove (3.52), it is sufficient to show that

$$\underline{v}_{\tilde{i}+1}=2+\sum_{t=1}^{\tilde{i}-1}l_t+l'_{\tilde{i}}\tag{3.57}$$

if

$$l_m\leqslant 2+\sum_{t=1}^{m-1}l_t,\quad m=1,\ldots,\tilde{i}-1,\tag{3.58}$$

$$l_{\tilde{i}}>l'_{\tilde{i}}=2+\sum_{t=1}^{\tilde{i}-1}l_t,\tag{3.59}$$

and

$$\alpha_m\leqslant 2+\sum_{t=1}^{m-1}l_t,\quad m=1,\ldots,\tilde{i}.\tag{3.60}$$

Obviously, (3.39)–(3.40) and (3.30) lead to that, for $i=1,\ldots,p$,

$$\underline{v}_{i+1}-\underline{v}_i=\begin{cases}\min\left(1,\frac{\underline{v}_i}{\alpha_i}\right)l_i, & \text{if } l_i\leqslant\max(\underline{v}_i,\alpha_i);\\ \underline{v}_i, & \text{if } l_i>\max(\underline{v}_i,\alpha_i).\end{cases}\tag{3.61}$$

On the other hand, by the proof in part (1), the conditions (3.58), (3.60) and (3.49) ensure that

$$\underline{v}_{\tilde{i}} = 2 + \sum_{t=1}^{\tilde{i}-1} l_t = l'_{\tilde{i}}. \tag{3.62}$$

Therefore, (3.61) leads to

$$\underline{v}_{\tilde{i}+1} - \underline{v}_{\tilde{i}} = \begin{cases} \min\left(1, \frac{2+\sum_{t=1}^{\tilde{i}-1} l_t}{\alpha_{\tilde{i}}}\right) l_{\tilde{i}}, & \text{if } l_{\tilde{i}} \leqslant \max\left(2 + \sum_{t=1}^{\tilde{i}-1} l_t, \alpha_{\tilde{i}}\right); \\ 2 + \sum_{t=1}^{\tilde{i}-1} l_t, & \text{if } l_{\tilde{i}} > \max\left(2 + \sum_{t=1}^{\tilde{i}-1} l_t, \alpha_{\tilde{i}}\right). \end{cases} \tag{3.63}$$

Notice that (3.59) and (3.60) imply that

$$l_{\tilde{i}} > \max(2 + \sum_{t=1}^{\tilde{i}-1} l_t, \alpha_{\tilde{i}}). \tag{3.64}$$

Hence (3.63) yields

$$\underline{v}_{\tilde{i}+1} - \underline{v}_{\tilde{i}} = 2 + \sum_{t=1}^{\tilde{i}-1} l_t. \tag{3.65}$$

Now (3.57) is obtained immediately by substituting (3.62) into (3.65), and hence (3.52) is proved. □

In order to prove (3.50), it is sufficient to show that $\underline{v}_{p+1}(l_1, \ldots, l_p, \alpha_1, \ldots, \alpha_p)$ is non-increasing with respect to $\alpha_t (t = 1, \ldots, p)$, or by the definition (3.39)–(3.40), show that

$$\psi(l, \alpha^{(1)}, v) \geqslant \psi(l, \alpha^{(2)}, v), \quad \text{if } \alpha^{(1)} < \alpha^{(2)}, \tag{3.66}$$

and

$$\psi(l, \alpha, v^{(1)}) \leqslant \psi(l, \alpha, v^{(2)}), \quad \text{if } v^{(1)} < v^{(2)}, \tag{3.67}$$

where $\psi(\cdot)$ is defined by (3.30). In fact, for the cases

$$l \leqslant \max(v, \alpha^{(1)}) \quad \text{and} \quad l \leqslant \max(v, \alpha^{(2)}), \tag{3.68}$$

and

$$l > \max(v, \alpha^{(1)}) \quad \text{and} \quad l > \max(v, \alpha^{(2)}), \tag{3.69}$$

the validity of (3.66) is obvious. So we only need to examine the case

$$l > \max(v, \alpha^{(1)}) \quad \text{and} \quad l \leqslant \max(v, \alpha^{(2)}). \tag{3.70}$$

Obviously, (3.70) implies that

$$\psi(l, \alpha^{(1)}, v) = 2v \geqslant v + \min(1, v/\alpha^{(2)}) l = \psi(l, \alpha^{(2)}, v). \tag{3.71}$$

Hence, (3.66) is true. The validity of (3.67) can be proved similarly. Q.E.D.

Remark 3.8. According to the definition (3.1), the efficiency coefficient of Algorithm CF-PCG$(p; l_1, \ldots, l_p; \alpha_1, \ldots, \alpha_p)$ is related to not only the convergence speed, but also the computation costs Q_{HgD}, Q_{Hg}, Q_D and Q_I, which are defined as follows:

$$Q_{HgD} = Q_{Hg} + Q_D, \quad Q_{Hg} = Q_H + Q_g;$$

Q_H is the computation cost to evaluate a Hessian $\nabla^2 f$; Q_g is the computation cost to evaluate a gradient ∇f; Q_D is the computation cost in a CF Step; Q_I is an upper bound of the computation cost in one subiteration of a PCG Step.

It is reasonable to measure the above computation costs by the corresponding numbers of multiplicative operations involved. So, we regard Q_H and Q_g as respectively the number of multiplicative operations to evaluate a Hessian and a gradient, and Q_D is the number of such operations in a CF step, which is given by the formula:

$$Q_D = \frac{1}{6}n^3 + \frac{3}{2}n^2 - \frac{2}{3}n. \tag{3.72}$$

Note that the number of multiplicative operations in the first subiteration in a PCG Step is $2n^2 + 5n + 1$, but the number in each of the later subiterations is $2n^2 + 6n + 2$. Since the difference of these two numbers is rather small, the larger one

$$Q_I = \max\{2n^2 + 5n + 1, 2n^2 + 6n + 2\} = 2n^2 + 6n + 2 \tag{3.73}$$

is used as an upper bound of them.

THEOREM 3.9. *Algorithm CF – PCG$(p; l_1, \ldots, l_p; \alpha_1, \ldots, \alpha_p)$ is locally convergent. Furthermore, its efficiency coefficient Γ satisfies*

$$\Gamma \geqslant \underline{\Gamma} \stackrel{def}{=} \frac{\ln \underline{\nu}_{p+1}(l_1, \ldots, l_p, \alpha_1, \ldots, \alpha_p)}{Q_{HgD} + pQ_{Hg} + (\sum_{t=1}^{p} l_t)Q_I}, \tag{3.74}$$

where Q_{HgD}, Q_{Hg} and Q_I are the computation costs defined in Remark 3.8, and $\underline{\nu}_{p+1}(l_1, \ldots, l_p, \alpha_1, \ldots, \alpha_p)$ has the following property: if we define

$$l'_m = \min\left\{l_m, 2 + \sum_{t=1}^{m-1} l'_t\right\}, \quad m = 1, \ldots, p, \tag{3.75}$$

then

$$\underline{\nu}_{p+1}(l_1, \ldots, l_p, \alpha_1, \ldots, \alpha_p) \leqslant 2 + \sum_{t=1}^{p} l'_t. \tag{3.76}$$

Furthermore, if

$$\alpha_m \leqslant 2 + \sum_{t=1}^{m-1} l'_t, \quad m = 1, \ldots, p, \tag{3.77}$$

then

$$\underline{v}_{p+1}(l_1, \ldots, l_p, \alpha_1, \ldots, \alpha_p) = 2 + \sum_{t=1}^{p} l'_t = \underline{v}_{p+1}(l'_1, \ldots, l'_p, \alpha_1, \ldots, \alpha_p). \tag{3.78}$$

Proof. It is easy to see from the definition (3.39)–(3.40) that

$$\underline{v}_i \geqslant 2, \quad i = 1, \ldots, p+1.$$

Hence by Lemma 3.6, we conclude that, when x^0 is close enough to the solution x^* to (1.1), the sequence $\{x^k\}$ generated by Algorithm CF-PCG $(p; l_1, \ldots, l_p; \alpha_1, \ldots, \alpha_p)$ is convergent.

Now let us turn to the validity of (3.74). Since Lemma 3.7 shows that $\underline{v}_{p+1} = \underline{v}_{p+1}(l_1, \ldots, l_p, \alpha_1, \ldots, \alpha_p)$, which is defined by (3.39)–(3.40), has the property (3.75)–(3.78), it is sufficient to show that

$$\Gamma \geqslant \frac{\ln \underline{v}_{p+1}}{Q_{HgD} + pQ_{Hg} + (\sum_{t=1}^{p} l_t) Q_I}. \tag{3.79}$$

By definition (3.2),

$$\ln(-\ln ||x^{j(p+1)+q+1} - x^*||) = \mu^{j,q+1} + \mu^j + \cdots + \mu^1 + \ln(-\ln ||x^0 - x^*||), \tag{3.80}$$

where

$$\mu^i = \ln v[x^{(i-1)(p+1)}, x^{i(p+1)}], \qquad i = 1, \ldots, j, \tag{3.81}$$

$$\mu^{j,q+1} = \ln v[x^{j(p+1)}, x^{j(p+1)+q+1}]. \tag{3.82}$$

Hence, using the notation $Q[\cdot, \cdot]$ defined in (3.1), we have

$$\frac{\ln(-\ln ||x^k - x^*||)}{\sum_{i=1}^{k} Q[x^{i-1}, x^i]} = \eta(j, q) + \frac{\ln(-\ln ||x^0 - x^*||)}{Q^{j,q+1} + Q^j + \cdots + Q^1}, \tag{3.83}$$

where

$$\eta(j, q) = \frac{\mu^{j,q+1} + \mu^j + \cdots + \mu^1}{Q^{j,q+1} + Q^j + \cdots + Q^1}, \tag{3.84}$$

$$Q^i = Q[x^{(i-1)(p+1)}, x^{i(p+1)}], \quad i = 1, 2, \ldots, j, \tag{3.85}$$

$$Q^{j,q+1} = Q[x^{j(p+1)}, x^{j(p+1)+q+1}]. \tag{3.86}$$

It is easy to see that

$$Q_{HgD} + pQ_{Hg} + \left(\sum_{t=1}^{p} l_t\right) Q_I \geqslant Q^i \geqslant Q_{HgD} > 0, \quad i = 1, \ldots, j. \tag{3.87}$$

Therefore by definition (3.1), in order to show (3.79), it is sufficient to prove that

$$\liminf_{j\to\infty} \eta(j,q) \geqslant \frac{\ln \underline{v}_{p+1}}{Q_{HgD} + pQ_{Hg} + (\sum_{t=1}^{p} l_t) Q_I}, \tag{3.88}$$

where $\underline{v}_{p+1}$ is defined by (3.39)–(3.40). We prove (3.88) for two cases: $q = p$ and $0 \leqslant q \leqslant p-1$, respectively. For the first case $q = p, \eta(j,q)$ can be written as

$$\eta(j,q) = \frac{\mu^{j+1} + \mu^j + \cdots + \mu^1}{Q^{j+1} + Q^j + \cdots + Q^1}. \tag{3.89}$$

Note that the procedure of generating $x^{i(p+1)}$ from $x^{(i-1)(p+1)} (i = 1, 2, \ldots)$ is the same as the one that generates x^{p+1} from x^0. Therefore, by Lemma 3.6 we have

$$\liminf_{i\to\infty} \mu^i \geqslant \ln \underline{v}_{p+1}. \tag{3.90}$$

Combining (3.89), (3.90) and (3.87) yields

$$\begin{aligned} \liminf_{j\to\infty} \eta(j,q) &= \liminf_{j\to\infty} \frac{\mu^{j+1} + \cdots + \mu^1}{Q^{j+1} + \cdots + Q^1} \\ &\geqslant \frac{\ln \underline{v}_{p+1}}{Q_{HgD} + pQ_{Hg} + (\sum_{t=1}^{p} l_t) Q_I}. \end{aligned} \tag{3.91}$$

The proof is completed for the first case.

For the second case $0 \leqslant q \leqslant p-1$, it is apparent that

$$\eta(j,q) \leqslant \frac{\mu^j + \cdots + \mu^1}{Q^j + \cdots + Q^1} \cdot \frac{Q^j + \cdots + Q^1}{Q^{j,q+1} + Q^j + \cdots + Q^1}. \tag{3.92}$$

Note that the inequality (3.87) implies that

$$\lim_{j\to\infty} \frac{Q^j + \cdots + Q^1}{Q^{j,q+1} + Q^j + \cdots + Q^1} = 1. \tag{3.93}$$

Combining (3.92), (3.93) and (3.91) shows the validity of the inequality (3.88) for the second case. Thus the theorem is proved. □

4. The Implementable Algorithm and Its Efficiency

The purpose of this section is to derive an implementable algorithm from the algorithmic model CF-PCG$p(l_1, \ldots, l_p, \alpha_1, \ldots, \alpha_p)$ and investigate its efficiency.

In order to establish an implementable algorithm from CF-PCG(p; $l_1, \ldots, l_p, \alpha_1, \ldots, \alpha_p$), we should specify the parameters $p, l_1, \ldots, l_p, \alpha_1, \ldots, \alpha_p$. A natural idea is to select such parameters that maximize the corresponding efficiency coefficient. However, it is difficult to find the precise relationship between the efficiency coefficient and the parameters, or a definite relationship may not exist and the situation is problem dependent. Instead of the efficiency coefficient itself, its lower bound given by the right hand side of (3.74) in Theorem 3.9 is maximized as follows. We first investigate the selection of the parameters $\alpha_1, \ldots, \alpha_p$. Suppose that $l_1, \ldots, l_p$ are given, comparing (3.76) with (3.78) results in the selection

$$\alpha_m \leqslant 2 + \sum_{t=1}^{m-1} l'_t, \quad m = 1, \ldots, p,$$

where l'_t is defined by (3.75). However, α_m appears in Step 4 of Algorithm CF-PCG($\cdot$) and exercises influences on the termination of the subiterations in the PCG Step according to (2.2). It is easy to see that the larger the value α_m, the more possible to terminate earlier. Therefore, to save the possible computation cost, it makes sense to select

$$\alpha_m = 2 + \sum_{t=1}^{m-1} l'_t, \quad m = 1, \ldots, p. \tag{4.1}$$

Now we turn to the selection of $l_1, \ldots, l_p$ such that $\underline{\Gamma}$ in (3.74) is maximized. (4.1) yields the validity of (3.78), which implies that, for any $l_1, \ldots, l_p$ and the corresponding $l'_1, \ldots, l'_p$,

$$\frac{\ln \underline{\nu}_{p+1}(l_1, \ldots, l_p, \alpha_1, \ldots, \alpha_p)}{Q_{HgD} + pQ_{Hg} + (\sum_{t=1}^{p} l_t) Q_I} \leqslant \frac{\ln \underline{\nu}_{p+1}(l'_1, \ldots, l'_p, \alpha_1, \ldots, \alpha_p)}{Q_{HgD} + pQ_{Hg} + (\sum_{t=1}^{p} l'_t) Q_I}. \tag{4.2}$$

Obviously,

$$l'_m \leqslant 2 + \sum_{t=1}^{m-1} l'_t, \quad m = 1, \ldots, p.$$

The above two inequalities imply that $\underline{\Gamma}$ in (3.74) must have a maximizer which satisfies

$$l_m \leqslant 2 + \sum_{t=1}^{m-1} l_t, \quad m = 1, \ldots, p. \tag{4.3}$$

In other words, the maximum of Γ can still be achieved if we include the extra constraint (4.3). The constraint (4.3) and (3.75) mean that

$$l'_m = l_m, \quad m = 1, \ldots, p. \tag{4.4}$$

Thus the selection formula (4.1) becomes

$$\alpha_m = 2 + \sum_{t=1}^{m-1} l_t. \tag{4.5}$$

Under the conditions (4.5) and (4.3) and by (3.78), with (4.4) in our mind, maximizing $\underline{\Gamma}$ in (3.74) leads to the following optimization problem with respect to $p, l_1, \ldots, l_p$:

$$\begin{aligned} \max \quad & \frac{\ln(2 + \sum_{t=1}^{p} l_t)}{Q_{HgD} + pQ_{Hg} + (\sum_{t=1}^{p} l_t)Q_I}, \\ \text{s.t} \quad & l_m \leqslant 2 + \sum_{t=1}^{m-1} l_t, \quad m = 1, \ldots, p; \end{aligned} \tag{4.6}$$

p is a nonnegative integer and $l_1, l_2, \ldots, l_p$ are positive integers. (4.7)

Introducing a variable $\sigma = \sum_{t=1}^{p} l_t$, the above problem is transformed to the following optimization problem with respect to $p, l_1, \ldots, l_p$ and σ:

$$\begin{aligned} \max \quad & \tilde{v}(p, \sigma) = \frac{\ln(2 + \sigma)}{Q_{HgD} + pQ_{Hg} + \sigma Q_I}, \\ \text{s.t} \quad & l_m \leqslant 2 + \sum_{t=1}^{m-1} l_t, \quad m = 1, \ldots, p; \end{aligned} \tag{4.8}$$

$$\sigma = \sum_{t=1}^{p} l_t; \tag{4.9}$$

p is a nonnegative integer and $l_1, l_2, \ldots, l_p$ are positive integers. (4.10)

This problem can be solved by two stages:

(1) For a fixed σ, find the optimal values $p = \overline{p}(\sigma)$ and $l_m = \overline{l}_m(\sigma)$, $m = 1, \ldots, p$. Obviously, these optimal values should make the objective function $\tilde{v}(p, \sigma)$ as large as possible, i.e. $p = \overline{p}(\sigma)$ should be as small as possible. This fact, by (4.9), leads to the conclusion that $l_m = \overline{l}_m(\sigma)$ should be as large as possible. Thus if $\sigma = 0$, then $p = 0$; if $\sigma = 1$, then $l_1 = 1$ and $p = 1$; if $\sigma \geqslant 2$, then

$$\overline{l}_m(\sigma) = 2 + \sum_{t=1}^{m-1} \overline{l}_t(\sigma) = 2^m, \quad m = 1, 2, \ldots, p-1,$$

$$\overline{l}_p(\sigma) = \sigma - \sum_{t=1}^{p-1} \overline{l}_t(\sigma) = \sigma - (2^p - 2),$$

and $p = \overline{p}(\sigma)$ is the integer satisfying $\sum_{m=1}^{\overline{p}(\sigma)-1} 2^m < \sigma \leqslant \sum_{m=1}^{\overline{p}(\sigma)} 2^m$, or explicitly, $p = \overline{p}(\sigma)$ is the smallest integer not smaller than

$$\frac{\ln(\sigma+2)}{\ln 2} - 1. \tag{4.11}$$

(2) Find the optimal solution σ^* and the corresponding $p^* = \overline{p}(\sigma^*)$ and $l_m^* = \overline{l}_m(\sigma^*), \quad m = 1, \ldots, p$. According to the results in stage (1), σ^* should be a global solution to the one-dimensional optimization problem

$$\max \quad v(\sigma) = \frac{\ln(2+\sigma)}{Q_{HgD} + \overline{p}(\sigma) Q_{Hg} + \sigma Q_I}, \tag{4.12}$$

$$\text{s.t.} \quad \sigma \text{ is a nonegative integer}, \tag{4.13}$$

where $\overline{p}(\sigma)$ is the smallest integer not smaller than $\frac{\ln(2+\sigma)}{\ln 2} - 1$. Thus after getting σ^*, the corresponding optimal parameters p^* and l_m^* can be obtained easily. This yields the following algorithm:

ALGORITHM CF-PCG

In Algorithm CF-PCG$(p; l_1, \ldots, l_p; \alpha_1, \ldots, \alpha_p)$, the parameters are specified as follows:

(1) Find the global solution σ^* to the one-dimensional optimization problem (4.12)–(4.13).

(2) $p = p^* = \overline{p}(\sigma^*)$, where $\overline{p}(\sigma^*)$ is the smallest integer not smaller than $\frac{\ln(2+\sigma^*)}{\ln 2} - 1$.

(3) if $\sigma^* = 1, l_1 = l_1^* = 1$; if $\sigma^* \geqslant 2$,

$$l_m = l_m^* = \begin{cases} 2^m, & m = 1, \ldots, p^* - 1; \\ \sigma^* - 2^{p^*} + 2, & m = p^*. \end{cases}$$

(4) $\alpha_m = \alpha_m^* = 2^m, \quad m = 1, \ldots, p^*$.

THEOREM 4.1. *If the initial point x^0 is close enough to the solution x^*, the above Algorithm CF-PCG is well defined.*

Proof. To prove that Algorithm CF-PCG is well-defined, we only need to show the existence of the global solution σ^* to (4.12)–(4.13). In fact, it is easy to see that for any nonnegative integer σ,

$$v(\sigma) < \frac{1}{Q_I} \cdot \frac{\ln(2+\sigma)}{\sigma}.$$

Therefore, noticing

$$\lim_{\sigma \to \infty} \frac{\ln(2+\sigma)}{\sigma} = 0$$

and

$$v(0) = \frac{\ln 2}{Q_{HgD}} > 0,$$

we conclude that the series $\{v(\sigma), \sigma = 0, 1, 2, \ldots\}$ has a finite maximizer σ^*. □

The efficiency coefficient of the above Algorithm CF-PCG is estimated by the following theorem.

THEOREM 4.2. *The efficiency coefficient* Γ *of Algorithm CF-PCG satisfies*

$$\Gamma \geqslant v^* \stackrel{def}{=} v(\sigma^*), \tag{4.14}$$

where $v(\cdot)$ *is defined by* (4.12), *and* σ^* *is the global solution to* (4.12)–(4.12).

Proof. The conclusion comes from Theorem 3.9 directly. Since Algorithm CF-PCG contains PCG Steps which are used to solve the Newton equation, its efficiency depends on the spectrum of the Hessian matrix $\nabla^2 f$. The worst case is considered in Theorem 4.2, while the best case is shown in the following theorem. □

THEOREM 4.3. *Suppose that when* x *is close enough to the solution* x^*,

$$\kappa_2(\nabla^2 f(x)) = 1, \tag{4.15}$$

where $\kappa_2(\cdot)$ *is the condition number. Then, the estimate (4.14) in Theorem 4.2 is improved as*

$$\Gamma \geqslant \bar{v}^*, \tag{4.16}$$

where

$$\bar{v}^* = \frac{\ln(2 + \sigma^*)}{Q_{HgD} + \bar{p}(\sigma^*) Q_{Hg} + \bar{p}(\sigma^*) Q_I}. \tag{4.17}$$

Proof. Consider the sequence

$$\{x^0, x^1, \ldots, x^{\sigma^*+1}, \ldots, x^{j(\sigma^*+1)}, x^{j(\sigma^*+1)+1}, \ldots, x^{j(\sigma^*+1)+t}, \ldots, x^{(j+1)(\sigma^*+1)}, \ldots\} \tag{4.18}$$

generated by Algorithm CF-PCG. According to the structure of the algorithm, the maximum number of subiterations in the PCG Step at $x^{j(\sigma^*+1)+t}$ is either 2^t (when $1 \leqslant t < \bar{p}(\sigma^*)$), or $\sigma^* - \sum_{t=1}^{p-1} 2^t$ (when $t = p = \bar{p}(\sigma^*)$). Now we show that, under the condition (4.15), this maximum number is reduced to 1. In fact, the PCG step at $x^{j(\sigma^*+1)+t}$ is used to solve the Newton equation

$$\nabla^2 f(x^{j(\sigma^*+1)+t}) s = -\nabla f(x^{j(\sigma^*+1)+t}). \tag{4.19}$$

It is well known that this PCG Step is equivalent to the conjugate gradient subiterations to the preconditional linear system

$$A^{j,t}\bar{s} = -\nabla^2 f(x^{j(\sigma^*+1)})^{-1/2}\nabla f(x^{j(\sigma^*+1)+t}) \tag{4.20}$$

with the relationship

$$A^{j,t} = \nabla^2 f(x^{j(\sigma^*+1)})^{-1/2}\nabla^2 f(x^{j(\sigma^*+1)+t})\nabla^2 f(x^{j(\sigma^*+1)})^{-1/2}, \tag{4.21}$$

and

$$\bar{s} = \nabla^2 f\left(x^{j(\sigma^*+1)}\right)^{1/2} s.$$

Obviously, the condition (4.15) implies that, for sufficientyly large j,

$$\kappa_2(A^{j,t}) = 1. \tag{4.22}$$

Therefore, one conjugate gradient subiteration is enough to find the exact solution to (4.20) to get the zero residual for both the preconditioned system (4.20) and the original system (4.19). Therefore, according to the termination condition (2.2), the maximum number of subiterations in the PCG Step at $x^{j(\sigma^*+1)+t}$ is 1. Thus, the upper bound for the total number of subiterations in all PCG Steps from $x^{j(\sigma^*+1)}$ to $x^{(j+1)(\sigma^*+1)}$ is reduced to $\sum_{t=1}^{\bar{p}(\sigma^*)} 1 = \bar{p}(\sigma^*)$ from the previous estimate $\sum_{t=1}^{\bar{p}(\sigma^*)} l_t = (\sigma^*)$. The theorem is completed. □

For the purpose of comparison, the efficiency of Newton's method is given in the following theorem, which can be considered as a particular case of Theorem 3.9 with $p = 0$, or obtained directly from the result in [1].

THEOREM 4.4. *The efficiency coefficient Γ_N of Newton's method satisfies*

$$\Gamma_N \geqslant v_N = v_N(n) \stackrel{def}{=} \frac{\ln 2}{Q_{HgD}}. \tag{4.23}$$

Now let us compare the lower bound of the efficiency coefficient of Algorithm CF-PCG, v^*, with the one of Newton's method, v_N. Note that, as shown in Section 1, we are mainly interested in middle and large scale problems where the arithmetic operations dominate the computing time, i.e., the problems in which both Q_H and Q_g are relatively small compared with other costs. So, we consider the extreme case

$$Q_H = Q_g = 0 \tag{4.24}$$

in the following theorem.

THEOREM 4.5. *Compare Algorithm CF-PCG with Newton's method and consider the case that (4.24) holds. Suppose v^* and v_N are respectively given in Theorems 4.2 and 4.4. Then the ratio v^*/v_N has the following properties:*

(1) when $n = 16$, we have

$$v^*/v_N > 1. \tag{4.25}$$

(2) when $n \geqslant 16$, *the ratio* v^*/v_N *is strictly increasing with respect to* n.
(3) when $n \to \infty$,

$$v^*/v_N \sim \ln n / \ln 2. \tag{4.26}$$

Proof. (1) Since $Q_H = Q_g = 0$, the problem (4.12)–(4.13) is reduced to

$$\max \quad v(\sigma) = \frac{\ln(2+\sigma)}{Q_D + \sigma Q_I}, \tag{4.27}$$

$$\text{s.t.} \quad \sigma \text{ is a nonnegative integer.} \tag{4.28}$$

In order to describe the dependency on n, we denote

$$Q_D = Q_D(n), \quad Q_I = Q_I(n),$$

and express $v(\sigma)$ in (4.27) as $v(\sigma, n)$. It is easy to see that $v(1,16) > v(0,16)$, and the ratio $v(1,n)/v(0,n)$ is increasing with respect to n. Therefore, when $n \geqslant 16, v(1,n) > v(0,n)$. This means that when $n \geqslant 16$, the solution σ^* to (4.27)–(4.28) is positive, and

$$v(\sigma^*, n) \geqslant v(1,n) > v(0,n). \tag{4.29}$$

Therefore, (4.25) is proved by $v^* = v(\sigma^*, n)$ and $v_N = v(0,n)$ when $n = 16$.
(2) It suffices to show that if $n_2 > n_1 \geqslant 16$, then

$$\frac{v(\sigma_2^*, n_2)}{v_N(n_2)} > \frac{v(\sigma_1^*, n_1)}{v_N(n_1)}, \tag{4.30}$$

where σ_1^* and σ_2^* are the solutions to (4.27)–(4.28) with $n = n_1$ and $n = n_2$, respectively. To prove (4.30), define

$$w(\sigma, n) = \frac{v(\sigma, n)}{v_N(n)} = \frac{\ln(2+\sigma)}{1 + \sigma(Q_I(n)/Q_D(n))} \cdot \frac{1}{\ln 2}. \tag{4.31}$$

Obviously, for any fixed $\sigma > 0$, $w(\sigma, n)$ is strictly increasing with respect to n since $Q_I(n)/Q_D(n)$ is strictly decreasing. Recall that it is shown in part(1) that when $n \geqslant 16, \sigma^* > 0$. Therefore, if $n_2 > n_1 \geqslant 16$,

$$w(\sigma_1^*, n_2) > w(\sigma_1^*, n_1). \tag{4.32}$$

However, σ_2^* is the solution to (4.27)–(4.28) with $n = n_2$, which implies

$$w(\sigma_2^*, n_2) = \frac{v(\sigma_2^*, n_2)}{v_N} \geqslant \frac{v(\sigma_1^*, n_2)}{v_N} = w(\sigma_1^*, n_2). \tag{4.33}$$

Combining (4.33), (4.32) and (4.31) yields (4.30).

(3) To show (4.26), we maximize the objective function $v(\sigma)$ in (4.27) with continuous variable σ and estimate its maximizer $\tilde{\sigma}^*$ when $n \to \infty$.

The necessary condition $v'(\tilde{\sigma}^*) = 0$ leads to the conclusion that $\tilde{\sigma}^*$ satisfies the equation

$$\frac{1}{2+\sigma}[Q_D/Q_I + \sigma] - \ln(2+\sigma) = 0,$$

or equivalently,

$$\frac{2+\sigma}{e}\ln\frac{2+\sigma}{e} = \frac{Q_D/Q_I - 2}{e}. \tag{4.34}$$

Introducing

$$z = \frac{2+\sigma}{e}, \qquad c = \frac{Q_D/Q_I - 2}{e}, \tag{4.35}$$

the Equation (4.34) is transfered to

$$z\ln z = c. \tag{4.36}$$

Note that the function $z\ln z$ is increasing when $z > e$. So it is easy to see that when $c > e$, the solution z^* to (4.36) satisfies

$$\frac{c}{\ln c} < z^* < \frac{c}{\ln c - \ln\ln c}.$$

Therefore, when $c \to \infty$,

$$z^* \sim \frac{c}{\ln c}. \tag{4.37}$$

Since $Q_D/Q_I \sim n/12$ when $n \to \infty$, we conclude by (4.35) and (4.37) that when $n \to \infty$, the maximizer $\tilde{\sigma}^*$ satisfies

$$\tilde{\sigma}^* \sim ez^* \sim \frac{n/12}{\ln n}. \tag{4.38}$$

At last, let us prove (4.26). Obviously, the solution σ^* to (4.27)–(4.28) satisfies

$$\tilde{\sigma}^* - 1 \leqslant \sigma^* \leqslant \tilde{\sigma}^* + 1.$$

So (4.38) implies that when $n \to \infty$,

$$\sigma^* \sim \frac{n/12}{\ln n}. \tag{4.39}$$

Therefore, when $n \to \infty$,

$$v(\sigma^*) = \frac{\ln(2+\sigma^*)}{Q_I(Q_D/Q_I + \sigma^*)} \sim \frac{\ln(2+(n/12)/\ln n)}{Q_I(n/12 + (n/12)/\ln n)} \sim \frac{\ln n}{Q_I \cdot n/12}, \tag{4.40}$$

or

$$\frac{v^*}{v_N} = \frac{v(\sigma^*)}{v(0)} \sim \frac{\ln n}{Q_I \cdot n/12}\cdot\frac{Q_D}{\ln 2} \sim \frac{\ln n}{\ln 2}.$$

This is just (4.26). □

Remark 4.6. It is not difficult to show the order

$$\vec{v}^* \geqslant v^* \geqslant v^*_{[5]} \geqslant v_N, \tag{4.41}$$

where $\vec{v}^*$, v^* and v_N are respectively the lower bounds of the efficiency given in Theorems 4.3, 4.2 and 4.4, and $v^*_{[5]}$ is the corresponding lower bounds which we obtained by using the concept of efficiency of [1] and the method given in this paper for the algorithm proposed in [5]. Some values of the ratios $\vec{v}^*/v_N$, v^*/v_n, and $v^*_{[5]}/v_N$ with $Q_H = Q_g = 0$ are given in the following table. If these lower bounds are taken as an efficiency measure of the corresponding algorithms, Algorithm CF-PCG is the best one.

	$n = 100$	$n = 200$	$n = 300$	$n = 400$	$n = 500$
$\bar{\mathrm{v}}^*/\mathrm{v_N}$	2.29	2.94	3.41	3.66	3.95
$\mathrm{v}^*/\mathrm{v_N}$	1.79	2.27	2.59	2.83	3.03
$v^*_{[5]}/v_N$	1.57	2.05	2.40	2.66	2.85

5. Numerical Experiments

Algorithm CF-PCG and Newton's method are tested in our numerical experiments. The test problems are quoted from the unconstrained optimization problems in [12]. Because we are interested in middle and large scale problems, only the problems with the variable dimension n, i.e. Problems 21–26 and Problem 35, are considered. Since the code of Problem 35 is complicated, it is omitted. Note that this paper is mainly concerned with the case where the arithmetic operations dominate the computing time. So Problem 26, where the ratio $(Q_H + Q_g)/Q_D$ approximates 11 when $n = 500$, is also removed from our test. Thus, our experiments include five problems: Problems 21–25. Table 1 lists their names and standard start points. The ratios $(Q_H + Q_g)/Q_D$ for these five problems are all less than 0.3 when $n = 500$. In order to compute the parameters $p, l_1, \ldots, l_t, \alpha_1, \ldots, \alpha_t$ in running the CF-PCG algorithm and to calculate $\vec{v}^*, v^*$ and v_N, we need to use the numbers of multiplicative operations Q_D, Q_I, Q_g and Q_H. Here Q_D and Q_I are computed by (3.72) and (3.73), and Q_H and Q_g are counted when the Hessian and the gradient are computed.

Algorithm CF-PCG is executed in C++ routines with double precision. The initial points of the test problems are standard start points. Notice that the algorithm is a local algorithm and our theoretical results are valid in a neighbourhood of x^*, where

$$||\nabla f(x)|| < 1.$$

However, the standard start points given in Table 1 may be rather far away from x^*. So, to prevent the premature case, in our code the termination criterion (2.2) in the PCG step is modified as

$$||r_{i-1}|| \leqslant \min\{||b||^{1+e}, 0.9\}.$$

In addition, the condition

$$||\nabla f(x^k)|| \leqslant 10^{-6} \tag{5.1}$$

is used for the termination test, that is, when the condition is satisfied, computation stops.

In Figures 1–5, the horizontal axis stands for the dimension n, while the vertical axis stands for the ratio of the CPU times spent by Newton's method and by Algorithm CF-PCG. There are two kinds of ratios:

(1) the ratio of CPU times measured by the practical computation, which is defined as

$$R_{\text{prac}} = \frac{\text{The CPU time by Newton's method}}{\text{The CPU time by Algorithm CF-PCG}}. \tag{5.2}$$

The ratios R_{prac} are dotted by $*$ in Figures 1–5 for Problems 21–25, respectively. Notice that in Figure 4 there is no $*$ for $n = 240, 260, \ldots, 500$ since both Algorithm CF-PCG and Newton's method are not convergent. The same phenomena appears in Figure 5.

(2) The ratio of CPU times estimated by the theoretical analysis. Notice that the efficiency of a method is approximately reverse proportional to the CPU time spent, and the lower bounds of the efficiency coefficient, e.g. $\bar{v}^*, v^*$, or. v_N are approximations to the efficiency of the corresponding method. Therefore, the ratios $\bar{R}_{\text{theo}} = \bar{v}^*/v_N$ and $R_{\text{theo}} = v^*/v_N$ should reflect, to certain extent, the ratio of the CPU time spent by Newton's method over the one by Algorithm CF-PCG. We plot $\bar{R}_{\text{theo}}$ and R_{theo} in Figures 1–5 which form the upper theoretical curve and the lower theoretical curve, respectively.

Table 1. List of test problems

Problem no.	Problem name	Standard start point
21.	Extended Rosenbrock function	(-1.2,1,-1.2,1,... ,−1.2,1)
22.	Extended Powell singular function	(3,-1,0,1, ... ,3,-1,0,1)
23.	Penalty function I	(1,2, ... ,n)
24.	Penalty function II	(1/2, ... ,1/2)
25.	Variably dimensioned function	($\frac{n-1}{n}$,... ,$\frac{1}{n}$,0)

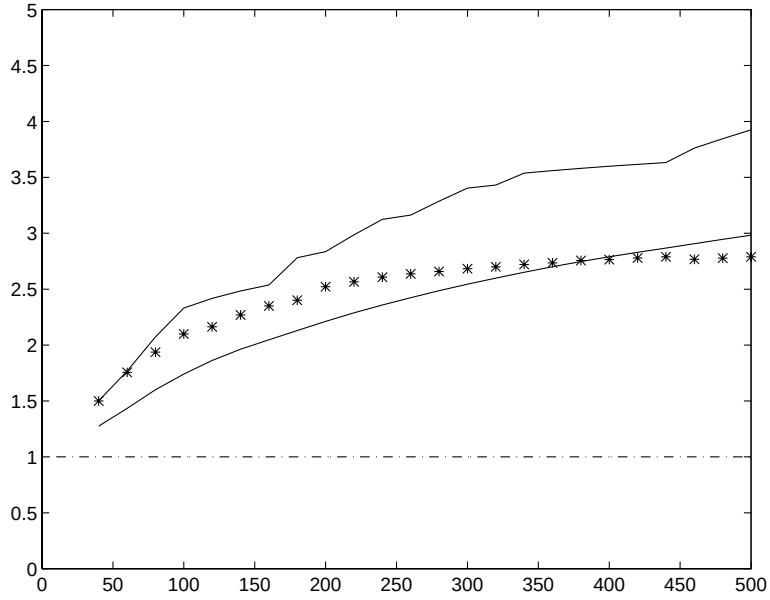

Figure 1. Algorithm CF-PCG versus Newton's method (Problem 21)

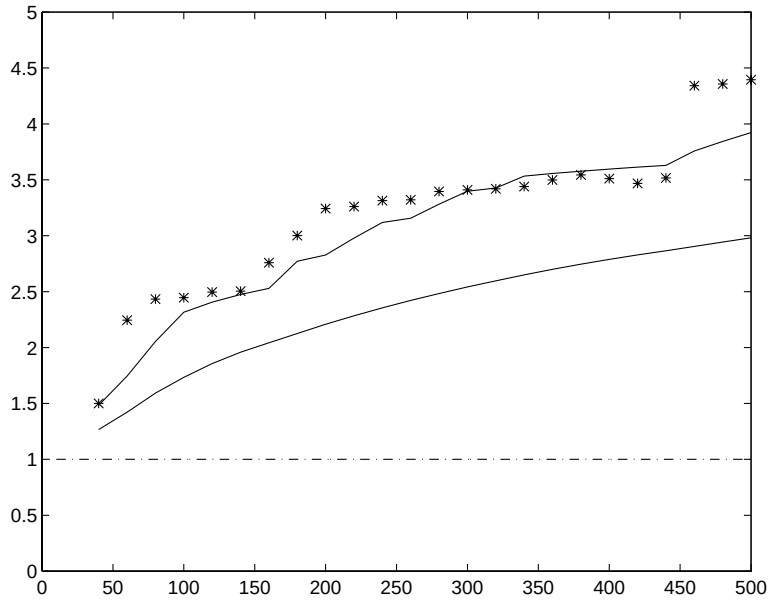

Figure 2. Algorithm CF-PCG versus Newton's method (Problem 22)

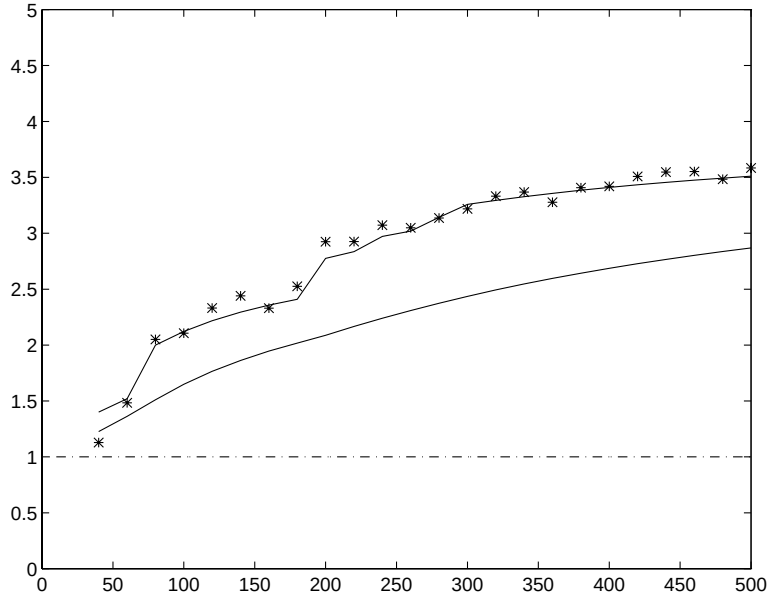

Figure 3. Algorithm CF-PCG versus Newton's method (Problem 23)

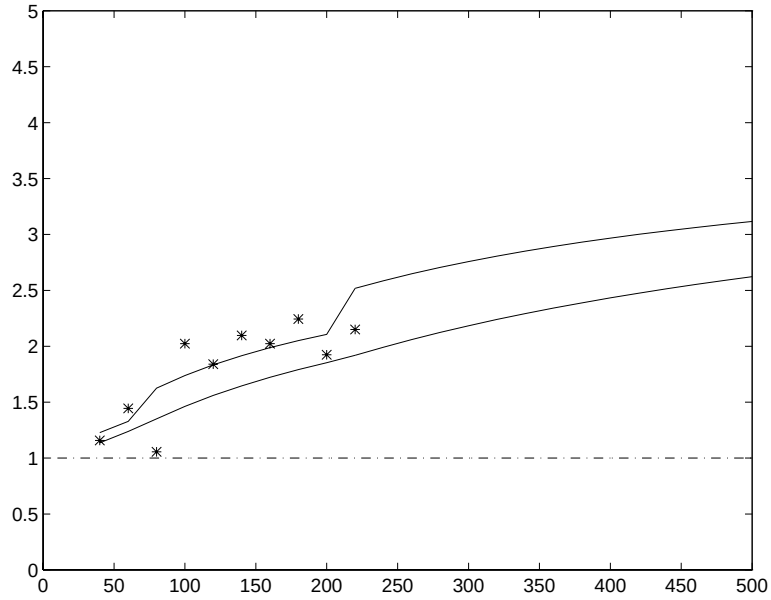

Figure 4. Algorithm CF-PCG versus Newton's method (Problem 24)

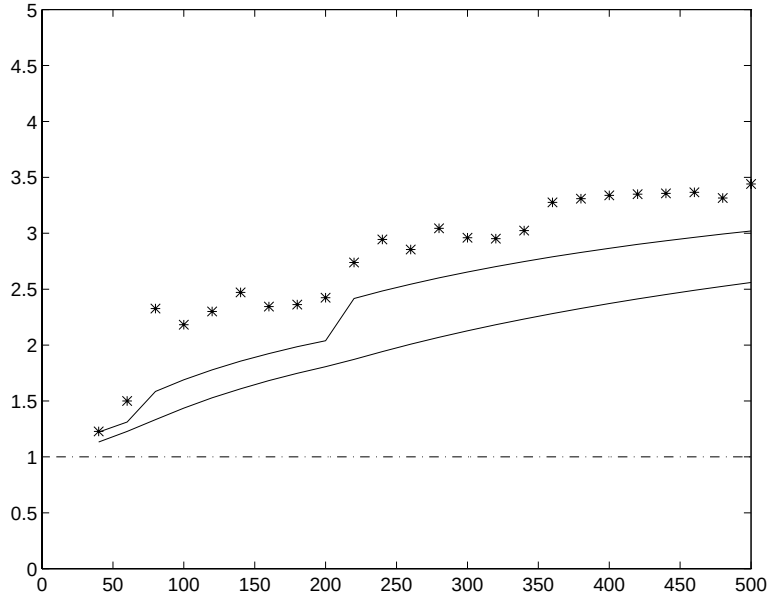

Figure 5. Algorithm CF-PCG versus Newton's method (Problem 25)

The " * " in Figures 1–5 show that Algorithm CF-PCG is more efficient than Newton's method. Furthermore, the tendency of the practical ratios R_{prac} coincide with the theoretical curves. This confirms the validity of our theoretical analysis in some sense. As we said in the first section, there are several popular CG preconditioners. In this paper we studied only a relatively simpler case. It would be useful to conduct efficiency analysis for other preconditioners, but we expect that such a future work shall be more challenging.

References

1. Brent, R. (1973), Some efficient algorithms for solving systems of nonlinear equation, *SIAM Journal on Numerical Analysis* 10, 327–344.
2. Conn, A.R., Gould, N.I.M. and Toint, P.L. (1992), *LANCELOT: A Fortran Package for Large-Scale Nonlinear Optimization (Release A), Springer Ser. Comput. Math.* Vol. 17, Springer-Verlag, Heidelberg, Berlin, New York.
3. Conn, A.R., Gould, N. and Toint, P.L. (1992), Numerical Experiments with the LANCELOT package (Release A) for large-scale nonlinear optimization, Technical Report, 92–075, Rutherford Appleton Laboratory, Chilton, England.
4. Dembo, R., Eisenstat, S. and Steihaug, T. (1982), Inexact Newton method, *SIAM Journal on Numerical Analysis* 19, 400–408.
5. Deng, N.Y. and Wang, Z.Z. (2000), Theoretical efficiency of an inexact Newton method, *Journal of Optimization Theory and Applications* 105, 97–112.

6. Dennis, J.E. and Schnabel, R.B. (1983), *Numerical methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.
7. Dixon, L.C.W. and Price, R.C. (1988), Numerical Experience with the Truncated Newton method for unconstrained optimization, *Journal of Optimization Theory and Applications* 56, 245–255.
8. Eisenstat, S.C. and Walker, H.F. (1996), Choosing the forcing terms in an inexact Newton method, *SIAM Journal on Scientific Computing* 17, 33–46.
9. Gill, P.E., Murray, W. and Wright, M.H (1981), *Practical Optimization*, Academic Press, London and New York.
10. Gould, G.H. and Vanloan, C.F. (1996), *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore.
11. Luenberger, D.G. (1984), *Linear and Nonlinear Programming*, 2nd ed., Addison-Wesley Publishing Company.
12. More, J.J., Garbow, B.S. and Hillstrom, K.E. (1981), Testing unconstrained optimization software, *ACM Transactions on Mathematical Software* 7, 17–41.
13. Nocedal, J. (1996), Large scale unconstrained optimization, Report-DEECS, Northwestern University.
14. Nocedal, J. and Wright, S.J. (1999), *Numerical Optimization*, Springer.
15. Ortega, J.M. and Rheinboldt, W.C. (1970), *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, London.
16. Ostrowski, A. (1960), *Solution of Equations and Systems of Equations*; Academic Press, New York.
17. Saad, Y. (1996), *Iterative Methods for Sparse Linear Systems*, PWS Publishing Company.
18. Steihaug, T. (1993), The conjugate gradient method and trust region in large scale optimization, *SIAM Journal on Numerical Analysis* 20, 626–637.
19. Toint, P.L. (1981), Duff, I.S. (ed.), *Towards an Efficient Sparsity Exploiting Newton Method for Minimization, Sparse Matrices and Their Uses*, Academic Press, London, England, 57–88.