A linesearch algorithm with memory for unconstrained optimization

N. I. M. Gould[1], S. Lucidi[2],
M. Roma[2] and Ph. L. Toint[3]

Report 98/03 January 28, 1998

[1] Rutherford Appleton Laboratory,
Chilton, Oxfordshire OX11 0QX, England
Email: n.gould@rl.ac.uk
Current reports available by anonymous ftp from the directory
"pub/reports" on joyous-gard.cc.rl.ac.uk

[2] Dipartimento di Informatica e Sistemistica
Università di Roma "La Sapienza"
via Buonarroti 12, 00185 Roma, Italy
Email: lucidi@dis.uniroma1.it and roma@dis.uniroma1.it

[3] Department of Mathematics,
Facultés Universitaires ND de la Paix
61, rue de Bruxelles, B-5000 Namur, Belgium
Email: pht@math.fundp.ac.be
Current reports available by anonymous ftp from the directory
"pub/reports" on thales.math.fundp.ac.be

Keywords : unconstrained optimization, Newton's method, memory

A linesearch algorithm with memory for unconstrained optimization

N. I. M. Gould S. Lucidi M. Roma Ph. L. Toint

January 28, 1998

Abstract

This paper considers algorithms for unconstrained nonlinear optimization where the model used by the algorithm to represent the objective function explicitly includes memory of the past iterations. This is intended to make the algorithm less "myopic" in the sense that its behaviour is not completely dominated by the local nature of the objective function, but rather by a more global view. We present a non-monotone linesearch algorithm that has this feature and prove its global convergence.

1 Introduction

This paper considers the unconstrained minimization problem

$$\min_{x \in \mathbf{R}^n} f(x)$$

where the objective function f is a twice continuously differentiable function from $\mathbf{R}^n$ into $\mathbf{R}$. If one uses a variant of Newton's method to solve this problem, then each iteration of the algorithm uses the first three terms of the Taylor's expansion of f to (locally) represent the objective and decide on a direction in which a better approximate solution can be found, or, at least, descent can be obtained. Such algorithms are well-known and have a well-established convergence theory (see Dennis and Schnabel, 1983, Gill, Murray and Wright, 1981 and Fletcher, 1987) to support their typically good numerical performances. A typical iteration of such a method determines, at a given iterate x_k, a search direction

$$d_k = H_k^{-1} g_k, \tag{1.1}$$

where $g_k = \nabla_x f(x_k)$ and H_k is a symmetric matrix that one chooses as a positive definite modification of $\nabla_{xx} f(x_k)$. A linesearch is then performed to obtain the next iterate

$$x_{k+1} = x_k + \alpha_k d_k = x_k + s_k \tag{1.2}$$

by choosing $\alpha_k > 0$ to approximately minimize $f(x_k + \alpha d_k)$. If $H_k = \nabla_{xx} f(x_k)$, it is clear that the search direction depends purely on local information: the values of the gradient and Hessian of the objective at x_k. The same is typically true if H_k results from a modified factorization of the

Hessian that makes H_k positive definite, because the modification itself depends on $\nabla_{xx} f(x_k)$. In summary, the (modified) Newton iteration is memoryless, in that the information accumulated at iterations preceding iteration k is completely disregarded.

The points which we wish to make here are that this property is not always desirable, and also that some memory of the past can suitably be introduced in the algorithm. Typical situations were the purely local nature of the Newton iteration is detrimental is when the objective function is very nonlinear, in the sense that its second-order Taylor series varies quickly as a function of x. This is for instance the case when the function has local "ripples" which have little global effect on the shape of the objective, but introduce strong very local variations. A memoryless iteration may then be fooled by the local nature of the function, and may easily loose track of the more global picture, although the latter is crucial for determining search directions that will enable substantial progress of the algorithm. We anticipate that, in such situations, remembering the shape of the function observed in the past will be beneficial in the determination of the search direction d_k.

The paper is organized as follows. In Section 2, we introduce a linesearch algorithm that has an explicit memory of the past iterations. Section 3 is devoted to the analysis of its convergence properties. In Section 4 some preliminary numerical results are reported and, finally, some conclusions and perspectives are presented in Section 5.

2 A linesearch algorithm with memory

The crucial motivation of the iteration (1.2) is that the search direction d_k is chosen to ensure that the point $x_k + d_k$ minimizes the model

$$m_k(x) = f(x_k) + \langle g_k, x - x_k \rangle + \tfrac{1}{2}\langle x - x_k, H_k(x - x_k)\rangle \tag{2.1}$$

where $\langle \cdot, \cdot \rangle$ denotes the Euclidean inner product. The "myopic" nature of the iteration then results from the purely local nature of this model. If we wish to make the algorithm more far-sighted, it is natural to consider models that attempt to be of a more global nature. We will therefore consider models of the type

$$m_k^{\mathrm{M}}(x) = m_k^{\mathrm{M}} + \langle g_k^{\mathrm{M}}, x - x_k \rangle + \tfrac{1}{2}\langle x - x_k, H_k^{\mathrm{M}}(x - x_k)\rangle$$

where m_k^{M} approximates $f(x_k)$, g_k^{M} approximates g_k and H_k^{M} approximates H_k. The search direction d_k is then determined by the equation

$$d_k = -(H_k^{\mathrm{M}})^{-1} g_k^{\mathrm{M}}, \tag{2.2}$$

where, as for H_k, H_k^{M} will, if necessary, be modified to ensure that it is positive definite. In what follows, we will consider the case where $m_k^{\mathrm{M}}(x)$ is a weighted combination of the purely local model $m_k(x)$ and the model seen in the past, that is

$$m_k^{\mathrm{M}}(x) = (1 - \mu_k) m_k(x) + \mu_k m_{k-1}^{\mathrm{M}}(x), \tag{2.3}$$

for some parameter μ_k which we may choose at each iteration between 0 and some fixed upper bound $\bar{\mu} \in [0, 1)$. The amount of memory of the past models is controlled by this parameter, as $m_k^{\mathrm{M}}(x) = m_k(x)$ for $\mu_k = 0$. How should we choose μ_k? Since one expects a local model to be more useful when the length of the step is short, it seems reasonable to require that μ_k should be of the order of $\alpha_{k-1}\|d_{k-1}\|$, where $\|\cdot\|$ denotes the Euclidean norm. We formalize this intuition by requiring that

$$\mu_k \le \min[\bar{\mu}, \eta\|s_{k-1}\|^\tau] \tag{2.4}$$

for some constant $\eta > 0$ and some exponent $\tau > 0$. Furthermore, we set $\mu_0 = 0$, since there is nothing to remember at the first iteration.

We also observe that, although m_k^{M} is fully specified by (2.1) and (2.3), its actual value is irrelevant for the determination of d_k in (2.2). As a consequence, we are free to redefine the value of the model at x_k if needed. This is very useful, since using models of the type (2.3) creates the additional difficulty that a descent direction for $m_k^{\mathrm{M}}(x)$ may not be a descent direction for $m_k(x)$ or the objective function itself. Hence, there is in general no guarantee that the linesearch will find a step α_k such that

$$f(x_k + \alpha_k d_k) < f(x_k)$$

if d_k is a descent direction for $m_k^{\mathrm{M}}(x)$. Thus non-monotone linesearch techniques appear as a natural alternative in our context. That is, instead of requiring descent on the objective or $m_k(x)$, we will be satisfied if the condition

$$f(x_k + \alpha_k d_k) \le f_k^{\mathrm{M}} + \gamma\alpha_k\langle g_k^{\mathrm{M}}, d_k\rangle, \tag{2.5}$$

holds for some $\gamma \in (0, \frac{1}{2})$ and for some value f_k^{M} such that

$$f(x_k) \le f_k^{\mathrm{M}} \le \max_{0\le i\le p(k)} f(x_{k-i}) \tag{2.6}$$

where $p(k)$ is an integer satisfying the condition $p(k) \le \min[p(k-1)+1, M]$ for some $M > 0$. This is the first Wolfe condition applied to the function

$$m_k^{\mathrm{M}}(x) + (f_k^{\mathrm{M}} - m_k^{\mathrm{M}}),$$

and allows $f(x_k + \alpha_k d_k)$ to exceed $f(x_k)$ since $f_k^{\mathrm{M}} \ge f(x_k)$. Observe that the choice $f_k^{\mathrm{M}} = f(x_k)$ is possible. However, in this case, (2.5) requires that d_k must be a descent direction from x_k, which is to say that $\langle g_k, d_k\rangle$ must be sufficiently negative. We therefore also require in this case that

$$\langle g_k, d_k\rangle < \nu\langle g_k^{\mathrm{M}}, d_k\rangle \ \text{ when } \ f_k^{\mathrm{M}} = f(x_k) \tag{2.7}$$

for some constant $\nu \in (0, 1]$. Note that this condition is easy to enforce algorithmically. For instance, one may choose, when $f_k^{\mathrm{M}} = f(x_k)$, to redefine $g_k^{\mathrm{M}} = g_k$ or even to set $\mu_k = 0$, which then yields that $m_k^{\mathrm{M}}(x) = m_k(x)$ and thus that $g_k^{\mathrm{M}} = g_k$.

We are now in position to define our algorithm more precisely.

Linesearch algorithm with memory

Step 0: An initial point x_0 is given, together with the constants $\bar{\mu}$, τ, $\beta \in (0,1)$, $\epsilon \in (0,1)$, η, γ and M. Compute $f(x_0)$, $g_0 = \nabla_x f(x_0)$ and $H_0 = \nabla_{xx} f(x_0)$. Also set $k = 0$, $p(-1) = 0$ and $\mu_0 = 0$.

Step 1: Choose f_k^{M} according to (2.6). Compute g_k^{M} and H_k^{M} from (2.3) and (2.7).

Step 2: Compute the search direction d_k from (2.2), possibly modifying H_k^{M} to ensure that it is positive definite (with smallest eigenvalue at least ϵ).

Step 3: Calculate a steplength $\alpha_k = \beta^j$ such that j is the smallest nonnegative integer ensuring (2.5).

Step 4: Set $x_{k+1} = x_k + \alpha_k d_k$, and compute μ_{k+1} to satisfy the bound (2.4). Increment k by one and go back to Step 1.

End of algorithm

Note that our linesearch is of the Armijo- or backtracking type. Note also that we have not included any stopping criterion, because our aim is to study the convergence of the algorithm on an infinite number of iterations. We do not describe here how H_k^{M} can be modified to achieve the uniform positive definiteness required at Step 2, but refer the reader to Gill et al. (1981), Schnabel and Eskow (1991) and Cheng and Higham (1996) for further description of adequate procedures. A simple, but crude, way to achieve this condition is to add $(1+\epsilon)\|H_k^{\mathrm{M}}\|$ times the identity matrix to H_k^{M} when it is indefinite.

3 Convergence theory

We now wish to verify that the algorithm is well defined and that it converges globally in the sense that all limit points of the sequence of iterates are first-order critical, irrespective of the choice of the initial approximation x_0. The analysis now proceeds in two stages. In the first we analyze the mechanism of the proposed method to show that certain general conditions on the model m_k^{M} are satisfied. In the second, we show that these general conditions are enough to guarantee global convergence of the algorithm.

3.1 The memory model

We start by analyzing the structure of the memory model.

Lemma 3.1 *If the model $m_k^{\mathrm{M}}(x)$ is defined by (2.3), then, for each k and all x,*

$$m_k^{\mathrm{M}}(x) = \sum_{i=0}^{k}(1-\mu_i)\left(\prod_{j=i+1}^{k}\mu_j\right) m_i(x). \tag{3.1}$$

Moreover,

$$\sum_{i=0}^{k}(1-\mu_i)\left(\prod_{j=i+1}^{k}\mu_j\right) \le \frac{1}{1-\bar{\mu}} \tag{3.2}$$

for all k.

Proof. We easily verify that

$$\begin{aligned} m_k^{\mathrm{M}}(x) &= (1-\mu_k)m_k(x) + \mu_k m_{k-1}^{\mathrm{M}}(x) \\ &= (1-\mu_k)m_k(x) + \mu_k(1-\mu_{k-1})m_{k-1}(x) + \mu_k\mu_{k-1}m_{k-2}^{\mathrm{M}}(x) \\ &= \dots \\ &= \sum_{i=0}^{k}(1-\mu_i)\left(\prod_{j=i+1}^{k}\mu_j\right) m_i(x), \end{aligned}$$

which proves (3.1). We also have, because of (2.4) and the bound $\bar{\mu} < 1$, that

$$\sum_{i=0}^{k}(1-\mu_i)\left(\prod_{j=i+1}^{k}\mu_j\right) \le \sum_{i=0}^{k}\prod_{j=i+1}^{k}\mu_j \le \sum_{i=0}^{k}\bar{\mu}^{k-i} = \sum_{i=0}^{k}\bar{\mu}^i \le \sum_{i=0}^{\infty}\bar{\mu}^i = \frac{1}{1-\bar{\mu}},$$

and the proof of the lemma is complete. □

The first part of the lemma simply expresses the value of the model with memory as a function of the memoryless (local) models at all past iterates.

In what follows, we require the following assumptions.

AS0: The objective function is bounded below on $\mathbf{R}^n$.

AS1: The iterates $\{x_k\}$ generated by the algorithm remain in a certain compact set $\Omega \subset \mathbf{R}^n$.

Note that we could have made the stronger assumption that the level set $\{x \in \mathbf{R}^n \mid f(x) \le f(x_0)\}$ is compact, which then guarantees AS0 and AS1, the latter because $f(x_k) \le f(x_0)$ for all k.

AS2: The modification to H_k^{M} to make it uniformly positive definite is such that the norm of the modified matrix is at most $2+\epsilon$ the norm of the original one.

The technique of adding $(1+\epsilon)\|H_k^{\mathrm{M}}\|$ times the identity matrix to H_k^{M} when it is indefinite satisfies AS2, but again more elaborate methods may be preferable.

Our first result shows that (2.2) ensures that d_k is a good descent direction on the model m_k^{M}.

Lemma 3.2 *There exist constants $\kappa_1 > 0$ and $\kappa_2 > 0$ such that, for all k,*

$$\langle g_k^{\mathrm{M}}, d_k\rangle \le -\kappa_1\|g_k^{\mathrm{M}}\|^2 \tag{3.3}$$

and

$$\|d_k\| \le \kappa_2\|g_k^{\mathrm{M}}\|. \tag{3.4}$$

Proof. We first note that (3.1) implies that

$$H_k^{\mathrm{M}} = \sum_{i=0}^{k}(1-\mu_i)\left(\prod_{j=i+1}^{k}\mu_j\right)\nabla_{xx}m_i(x_k) = \sum_{i=0}^{k}(1-\mu_i)\left(\prod_{j=i+1}^{k}\mu_j\right)H_i.$$

Now, if we denote

$$\kappa_H = \max_{x\in\Omega}\|\nabla_{xx}f(x)\|,$$

which is well defined because of AS1, we obtain from (3.2) that, for all k,

$$\|H_k^{\mathrm{M}}\| \le \max_{i=0,\ldots,k}\|H_i\|\sum_{i=0}^{k}(1-\mu_i)\left(\prod_{j=i+1}^{k}\mu_j\right) \le \frac{\kappa_H}{1-\bar{\mu}}.$$

The property (3.3) then follows from the inequality

$$\langle g_k^{\mathrm{M}}, d_k\rangle = -\langle g_k^{\mathrm{M}}, (H_k^{\mathrm{M}})^{-1}g_k^{\mathrm{M}}\rangle \le -\frac{1-\bar{\mu}}{\kappa_H}\|g_k^{\mathrm{M}}\|^2.$$

The inequality (3.4) results from

$$\|d_k\| = \|(H_k^{\mathrm{M}})^{-1}g_k^{\mathrm{M}}\| \le \|(H_k^{\mathrm{M}})^{-1}\|\,\|g_k^{\mathrm{M}}\| \le \frac{1}{\epsilon}\|g_k^{\mathrm{M}}\|,$$

where we have used the uniform positive definiteness of H_k^{M}. □

We now observe that (3.3) and (3.4) also cover the case where the system $H_k^{\mathrm{M}}d_k = -g_k^{\mathrm{M}}$ (giving (2.2)) is not solved exactly, provided its approximate solution is obtained by minimizing the quadratic $m_k^{\mathrm{M}}(x)$ along a set of descent directions that includes the negative gradient.

Lemma 3.3 *Assume that d_k is computed as*

$$d_k = -\sum_{i=1}^{m}\frac{\langle g_k^{\mathrm{M}}, v_i\rangle}{\langle v_i, H_k^{\mathrm{M}}v_i\rangle}v_i, \tag{3.5}$$

for some m between 1 and n, and where the directions $\{v_i\}_{i=1}^m$ satisfy the conditions

$$v_1 = -g_k^{\mathrm{M}} \;\; \textit{and} \;\; \langle g_k^{\mathrm{M}}, v_i\rangle \le 0 \quad (i=1,\ldots,m).$$

Then conditions (3.3) and (3.4) hold.

Proof. We have that

$$\langle g_k^{\mathrm{M}}, d_k\rangle = -\sum_{i=1}^{m}\frac{\langle g_k^{\mathrm{M}}, v_i\rangle^2}{\langle v_i, H_k^{\mathrm{M}}v_i\rangle} \le -\sum_{i=1}^{m}\frac{\langle g_k^{\mathrm{M}}, v_i\rangle^2}{\kappa_H\|v_i\|^2} \le -\frac{1}{\kappa_H}\|g_k^{\mathrm{M}}\|^2,$$

because of the definition of κ_H and where we have restricted the sum to its first term to obtain the last inequality. This gives (3.3) with $\kappa_1 = \kappa_H^{-1}$. Our assumption on d_k also yields that

$$\|d_k\| \le \|g_k^{\mathrm{M}}\|\sum_{i=1}^{m}\frac{\|v_i\|^2}{\langle v_i, H_k^{\mathrm{M}}v_i\rangle} \le \frac{m}{\epsilon}\|g_k^{\mathrm{M}}\| \le \frac{n}{\epsilon}\|g_k^{\mathrm{M}}\|,$$

where we have used the Cauchy-Schwarz inequality, the uniform positive definiteness of H_k^{M} and the bound $m \leq n$. Hence (3.4) holds with $\kappa_2 = n/\epsilon$. □

This result is of practical importance because it covers the case where the search direction d_k is computed by a truncated conjugate-gradient algorithm (see Dembo and Steihaug, 1983), a very common situation in large-scale problems.

We next verify that $\nabla_x m_k^{\mathrm{M}}(x_k)$ and $\nabla_x m_k(x_k) = \nabla_x f(x_k)$ asymptotically coincide when the iterates get closer.

Lemma 3.4 *We have that*

$$\lim_{k\to\infty} \|g_k^{\mathrm{M}} - g_k\| = 0 \tag{3.6}$$

whenever

$$\lim_{k\to\infty} \|x_{k+1} - x_k\| = 0. \tag{3.7}$$

Proof. Again, we deduce from (3.1) that

$$\nabla_x m_k^{\mathrm{M}}(x_k) = g_k^{\mathrm{M}} = \sum_{i=0}^{k}(1-\mu_i)\left(\prod_{j=i+1}^{k}\mu_j\right)\nabla_x m_i(x_k).$$

Thus, for all k,

$$\begin{aligned} g_k^{\mathrm{M}} - g_k &= \sum_{i=0}^{k-1}(1-\mu_i)\left(\prod_{j=i+1}^{k}\mu_j\right)\nabla_x m_i(x_k) + (1-\mu_k)\nabla_x m_k(x_k) - g_k \\ &= \sum_{i=0}^{k-1}(1-\mu_i)\left(\prod_{j=i+1}^{k}\mu_j\right)\nabla_x m_i(x_k) - -\mu_k \nabla_x m_k(x_k) \end{aligned} \tag{3.8}$$

as $\nabla_x m_k(x_k) = g_k$. We now observe that AS1 and the twice continuously differentiable nature of the objective function imply that $\|g_k\|$, $\|H_k\|$ and $\|x_k - x_i\|$ are uniformly bounded, and therefore, since $\nabla_x m_i(x) = g_i + H_i(x - x_i)$ that there exists a constant $\kappa_g > 0$ such that

$$\kappa_g = \max_{x\in\Omega} \|\nabla_x m_i(x)\|$$

for all i. If we now define

$$\theta_k \stackrel{\mathrm{def}}{=} \sum_{i=0}^{k-1}(1-\mu_i)\left(\prod_{j=i+1}^{k-1}\mu_j\right)$$

then (3.8) and (2.4) give that

$$\begin{aligned} \|g_k^{\mathrm{M}} - g_k\| &\leq \kappa_g \theta_k \mu_k + \mu_k \|\nabla_x m_k(x_k)\| \\ &\leq \mu_k \kappa_g (\theta_k + 1) \\ &\leq \eta \kappa_g (\theta_k + 1)\|s_{k-1}\|^{\tau}. \end{aligned} \tag{3.9}$$

We now observe that (3.2) ensures that $\theta_k \leq \frac{1}{1-\bar{\mu}}$, and hence that (3.9) gives that

$$\|g_k^{\mathrm{M}} - g_k\| \leq \frac{\kappa_g \eta (2-\bar{\mu})}{1-\bar{\mu}} \|s_{k-1}\|^{\tau}.$$

Since (3.7) means that $\|s_{k-1}\| = \|x_k - x_{k-1}\|$ converges to zero, this last bound implies that (3.6) holds. □

3.2 Global convergence

Our general conditions are (2.6)–(2.7) and (3.3)–(3.7). We now proceed to prove that they are sufficient for obtaining global convergence of our algorithm. We first verify that the linesearch procedure of Step 3 is well defined.

Theorem 3.5 *If (2.6) and (2.7) hold, then the algorithm is well defined in the sense that (2.5) holds for a finite j.*

Proof. Assume by contradiction that, at the iteration k, the test (2.5) is never satisfied. Then there exits a sequence $\{\alpha_j\}$, with $\alpha_j \to 0$ as $j \to \infty$, such that

$$f(x_k + \alpha_j d_k) > f_k^{\mathrm{M}} + \gamma \alpha_j \langle t g_k, d_k \rangle,$$

from which we have that

$$f(x_k + \alpha_j d_k) - f(x_k) > f_k^{\mathrm{M}} - f(x_k) + \gamma \alpha_j \langle g_k^{\mathrm{M}}, d_k \rangle. \tag{3.10}$$

Remember now that, because of (2.6), we have that $f_k^{\mathrm{M}} - f(x_k) \geq 0$. Now if $f_k^{\mathrm{M}} - f(x_k) > 0$, since $\alpha_j \to 0$, for sufficiently large j, (3.10) yields that $f_k^{\mathrm{M}} - f(x_k) \leq 0$, which is a contradiction. If, instead, $f_k^{\mathrm{M}} - f(x_k) = 0$, then dividing both terms of (3.10) by α_j and taking the limit for $j \to \infty$, we obtain that

$$\langle g_k, d_k \rangle \geq \gamma \langle g_k^{\mathrm{M}}, d_k \rangle,$$

which then contradicts (2.7). Hence (2.5) must be satisfied eventually. □

The rest of our convergence proof is strongly inspired by that of Grippo, Lampariello and Lucidi (1986) for the case where $g_k^{\mathrm{M}} = g_k$ for all k.

Theorem 3.6 *Assume that AS1 and AS2 hold. Then either the algorithm terminates at some x_p such that $g(x_p) = 0$, or it produces an infinite sequence $\{x_k\}$ whose every limit point $x^* \in \Omega$ satisfies $g(x^*) = 0$.*

Proof. Let $\ell(k)$ be an index such that

$$k - p(k) \leq \ell(k) \leq k \quad \text{and} \quad f(x_{\ell(k)}) = \max_{i=0,\ldots,p(k)} f(x_{k-i}). \tag{3.11}$$

From the linesearch condition (2.5) and (2.6), we obtain that

$$f(x_{k+1}) = f(x_k + \alpha_k d_k) \leq f_k^{\mathrm{M}} + \gamma \alpha_k \langle g_k^{\mathrm{M}}, d_k \rangle \leq f(x_{\ell(k)}) + \gamma \alpha_k \langle g_k^{\mathrm{M}}, d_k \rangle. \tag{3.12}$$

This in turn implies that

$$\begin{aligned} f(x_{\ell(k+1)}) &= \max_{i=0,\ldots,p(k+1)} f(x_{\ell(k+1-i)}) \\ &\leq \max_{i=0,\ldots,p(k)+1} f(x_{\ell(k+1-i)}) \\ &= \max[f(x_{\ell(k)}), f(x_{k+1})] \\ &= f(x_{\ell(k)}), \end{aligned}$$

where we have used the fact that $p(k+1) \le p(k)+1$. Thus the sequence $\{f(x_{\ell(k)})\}$ must therefore be non-increasing. Moreover, (2.5) and (2.6) also imply that

$$f(x_{\ell(k)}) = f(x_{\ell(k)-1} + \alpha_{\ell(k)-1} d_{\ell(k)-1}) \le f(x_{\ell(k-1)}) + \gamma \alpha_{\ell(k)-1} \langle g^{\rm M}_{\ell(k)-1}, d_{\ell(k)-1} \rangle.$$

But AS0 guarantees that $f(x_{\ell(k)})$ is bounded below, and we must therefore obtain that

$$\lim_{k\to\infty} \alpha_{\ell(k)-1} \langle g^{\rm M}_{\ell(k)-1}, d_{\ell(k)-1} \rangle = 0, \tag{3.13}$$

Observe now that (3.3), (3.4) and the bound $\alpha_k \le 1$ give that

$$\alpha_k^2 \|d_k\|^2 \le \alpha_k \|d_k\|^2 \le \kappa_2^2 \alpha_k \|g^{\rm M}_k\|^2 \le \frac{\kappa_2^2}{\kappa_1} \alpha_k |\langle g^{\rm M}_k, d_k \rangle|, \tag{3.14}$$

which, together with (3.13), yields that

$$\lim_{k\to\infty} \alpha_{\ell(k)-1} \|d_{\ell(k)-1}\| = 0. \tag{3.15}$$

We now intend to show that (3.15) is valid not only for the sequence $\{\ell(k)\}$ but for the complete sequence $\{k\}$. Let

$$\hat\ell(k) \stackrel{\rm def}{=} \ell(k+M+2). \tag{3.16}$$

First, we prove by induction that, for any $j \ge 1$,

$$\lim_{k\to\infty} \alpha_{\hat\ell(k)-j} \|d_{\hat\ell(k)-j}\| = 0 \tag{3.17}$$

and

$$\lim_{k\to\infty} f(x_{\hat\ell(k)-j}) = \lim_{k\to\infty} f(x_{\hat\ell(k)}). \tag{3.18}$$

If $j=1$, (3.17) follows from (3.15). This latter limit also implies that (3.18) holds because of the uniform continuity of the objective on Ω. Assume now that (3.17)–(3.18) hold for a given $j \ge 1$. Then the linesearch condition (2.5) ensures that

$$f(x_{\hat\ell(k)-j}) \le f(x_{\ell(\hat\ell(k)-j-1)}) + \gamma \alpha_{\hat\ell(k)-j-1} \langle g^{\rm M}_{\hat\ell(k)-j-1}, d_{\hat\ell(k)-j-1} \rangle.$$

Using (3.18), we then deduce that

$$\lim_{k\to\infty} \alpha_{\hat\ell(k)-(j+1)} \langle g^{\rm M}_{\hat\ell(k)-(j+1)}, d_{\hat\ell(k)-(j+1)} \rangle = 0,$$

and thus, from (3.3), that

$$\lim_{k\to\infty} \alpha_{\hat\ell(k)-(j+1)} \|d_{\hat\ell(k)-(j+1)}\| = 0.$$

The uniform continuity of the objective on Ω then implies that

$$\lim_{k\to\infty} f(x_{\hat\ell(k)-(j+1)}) = \lim_{k\to\infty} f(x_{\hat\ell(k)-j}) = \lim_{k\to\infty} f(x_{\hat\ell(k)}),$$

and we may therefore conclude that (3.17) and (3.18) hold for any $j \ge 1$.

Now, (3.16) implies that $\hat{\ell}(k) \geq k$, and we therefore have that, for each k,

$$x_{k+1} = x_{\hat{\ell}(k)} - \sum_{j=1}^{\hat{\ell}(k)-k-1} \alpha_{\hat{\ell}(k)-j} d_{\hat{\ell}(k)-j}. \tag{3.19}$$

But the first part of (3.11) ensures that

$$\hat{\ell}(k) - k - 1 = \ell(k + M + 2) - k - 1 \leq M + 1,$$

and thus (3.19) and (3.17) imply that

$$\lim_{k \to \infty} \|x_{k+1} - x_{\hat{\ell}(k)}\| = 0. \tag{3.20}$$

Since the sequence $\{f(x_{\ell(k)})\}$ converges, the uniform continuity of the objective on Ω and (3.20) yield that

$$\lim_{k \to \infty} f(x_{k+1}) = \lim_{k \to \infty} f(x_{\hat{\ell}(k)}). \tag{3.21}$$

But the linesearch condition (2.5) can be used again to obtain that

$$f(x_{k+1}) \leq f(x_{\ell(k)}) + \gamma \alpha_k \langle g_k^{\mathrm{M}}, d_k \rangle.$$

We may now take the limit for k tending to infinity in this last inequality, and deduce, using (3.21), that

$$\lim_{k \to \infty} \alpha_k \langle g_k^{\mathrm{M}}, d_k \rangle = 0,$$

and therefore, using (3.14) as before, that

$$\lim_{k \to \infty} \alpha_k \|d_k\| = 0. \tag{3.22}$$

Moreover, the last inequality in (3.14) also implies that

$$\lim_{k \to \infty} \alpha_k \|g_k^{\mathrm{M}}\|^2 = 0. \tag{3.23}$$

Consider now $\bar{x}$, any limit point of the sequence of iterates. Note that such a limit point must exist because of AS1, and a subsequence indexed by K_1 converging to $\bar{x}$. Then (3.23) ensures that either

$$\lim_{k \to \infty, k \in K_1} \|g_k^{\mathrm{M}}\| = 0 \tag{3.24}$$

or there exists a subsequence indexed by $K_2 \subseteq K_1$ such that

$$\lim_{k \to \infty, k \in K_2} \alpha_k = 0. \tag{3.25}$$

If (3.24) holds, then (3.22) and Lemma 3.4 together imply that

$$\|g(\bar{x})\| \leq \lim_{k \to \infty} \|g_k^{\mathrm{M}}\| + \lim_{k \to \infty} \|g(x_k) - g_k^{\mathrm{M}}\| = 0. \tag{3.26}$$

and $\bar{x}$ is a first-order stationary point. Now suppose that (3.25) holds. In this case, the mechanism of the linesearch implies that there exists an index k_0 such that, for all $k \geq k_0$, $k \in K_2$,

$$f(x_k + \frac{\alpha_k}{\beta} d_k) > f_k^{\mathrm{M}} + \gamma \frac{\alpha_k}{\beta} \langle g_k^{\mathrm{M}}, d_k \rangle \geq f(x_k) + \gamma \frac{\alpha_k}{\beta} \langle g_k^{\mathrm{M}}, d_k \rangle,$$

where we have used (2.6) to deduce the last inequality. Applying now the mean-value theorem, we find, for all $k \geq k_0$, $k \in K_2$, a point $u_k \in [x_k, x_k + (\alpha_k/\beta)d_k]$ such that

$$\langle g(u_k), d_k \rangle \geq \gamma \langle g_k^{\rm M}, d_k \rangle,$$

from which we have that

$$|\langle g_k^{\rm M}, d_k \rangle| \leq \frac{1}{1-\gamma} |\langle g_k^{\rm M} - g(u_k), d_k \rangle|. \tag{3.27}$$

If we now consider a further subsequence $K_3 \subseteq K_2$ such that

$$\lim_{k \to \infty, k \in K_3} x_k = \bar{x} \;\text{ and }\; \lim_{k \to \infty, k \in K_3} \frac{d_k}{\|d_k\|} = \bar{d},$$

we may deduce from and (3.27), (3.4), (3.3) and the Cauchy-Schwarz inequality that

$$\begin{aligned}
\|g(\bar{x})\| &= \lim_{k \to \infty, k \in K_3} \|g_k\| \\
&\leq \lim_{k \to \infty, k \in K_3} \|g_k^{\rm M}\| + \lim_{k \to \infty, k \in K_3} \|g_k^{\rm M} - g_k\| \\
&\leq \kappa_2 \lim_{k \to \infty, k \in K_3} \frac{\|g_k^{\rm M}\|^2}{\|d_k\|} + \lim_{k \to \infty, k \in K_3} \|g_k^{\rm M} - g_k\| \\
&\leq \frac{\kappa_2}{\kappa_1} \lim_{k \to \infty, k \in K_3} \frac{|\langle g_k^{\rm M}, d_k \rangle|}{\|d_k\|} + \lim_{k \to \infty, k \in K_3} \|g_k^{\rm M} - g_k\| \\
&\leq \frac{\kappa_2}{\kappa_1(1-\gamma)} \lim_{k \to \infty, k \in K_3} |\langle g_k^{\rm M} - g(u_k), \frac{d_k}{\|d_k\|} \rangle| + \lim_{k \to \infty, k \in K_3} \|g_k^{\rm M} - g_k\| \\
&\leq \frac{\kappa_2 + 1}{\kappa_1(1-\gamma)} \lim_{k \to \infty, k \in K_3} \|g_k^{\rm M} - g_k\| + \frac{\kappa_2}{\kappa_1(1-\gamma)} \lim_{k \to \infty, k \in K_3} \|g_k - g(u_k)\|
\end{aligned}$$

But both $\|g_k^{\rm M} - g_k\|$ and $\|u_k - x_k\|$ must converge to zero because of (3.22) and Lemma 3.4. Hence again $\bar{x}$ is first-order critical, which concludes the proof. □

We conclude this convergence analysis by a few comments. The first is that conditions (2.6), (3.3), (3.4), (3.6)–(3.7) and (2.7) are not specific to our specific context, but also apply for the more general case where a non-monotone linesearch is applied on a problem where the gradients (and Hessians) are only approximated. In particular, conditions (3.6)–(3.7) give a criterion on how accurate the gradient approximation should be at each iteration. Interestingly, this criterion is similar in spirit to that proposed by Moré (1983) for the case of monotone trust-region algorithm (he requested (3.6) to hold whenever the sequence $\{x_k\}$ is convergent), and to that proposed by Toint (1988) in the same context (where the error in the gradient has to be bounded by the trust-region radius). Our second comment is more practical. If we used the condition (2.4) for obtaining our global convergence results, nothing prevents us from imposing additional restrictions on the memory parameter. In particular, one could impose a condition of the form

$$\mu_k \leq \min[\bar{\mu}, \eta_g \|g_k^{\rm M}\|^{\tau_g}, \eta_s \|s_{k-1}\|^{\tau_s}]. \tag{3.28}$$

The intuitive justification for this alternative definition is that one expects a local model to produce asymptotically fast converging iterates when a critical point is approached.

4 Numerical Experience

In this section, we present the results of our preliminary numerical experience with the new global approach we have proposed in this paper. While our experiments are far from exhaustive, our aim has been to examine the reliability and effectiveness of this new global framework. With this in mind, we implemented an algorithm belonging to the class of the linesearch algorithms with memory defined in Section 2. Specifically, we used the following settings:

$$\gamma = 10^{-3}, \quad M = 3, \quad \beta = \tfrac{1}{2}, \quad \bar{\mu} = 0.5, \quad \nu = 0.9.$$

The crucial parameter μ_k was computed as

$$\mu_k = 10^{-i} \min[\bar{\mu}, \|g_k^{\mathrm{M}}\|, \|s_{k-1}\|], \tag{4.1}$$

where i is the smallest nonnegative integer for which the condition

$$\langle g_k, d_k \rangle < \nu \langle g_k^{\mathrm{M}}, d_k \rangle \tag{4.2}$$

is satisfied. This requirement on the choice of μ_k is stronger than the restriction (2.7) required at Step 1 of the linesearch algorithm with memory, but empirically leads to a significant improvement in the numerical performance of the new algorithm. This is likely because any search direction satisfying (4.2) is guaranteed to be "sufficiently good" descent direction for the model $m_k^{\mathrm{M}}(x_k)$.

We tested this algorithm on a set of large scale unconstrained problems from the CUTE collection Bongartz, Conn, Gould and Toint (1995); the chosen problems are a significant subset of the most difficult examples in the collection. We used double precision Fortran 90 codes compiled under xlf90 with the optimization compiling option. All of our tests were performed on an IBM RISC System/6000 375.

We summarize the results of this preliminary numerical study in Table 4.1. For each problem, we report the number of gradients evaluations ($\#g$), number of function evaluations ($\#f$), CPU time (in seconds) and the final objective function value. We compare our new linesearch algorithm with memory against a basic algorithm which computes search directions in the same way, but which makes no use of memory (i.e. obtained by setting $\mu = 0$ and $M = 1$).

Firstly, we note that the performance of the two algorithms are not directly comparable on problems `BROYDN7D`, `CHAINWOO`, `CURLY10` and `CURLY30` since the algorithms converge to different local minima. Of the remaining problems, the algorithm with memory provides significant savings in terms of number of function and gradient evaluations and CPU time relatives to the basic one without memory on problems `CURLY20`, `DIXMAANB`, `GENROSE`, `NCB20B` and `PENALTY1` For `FLETCHCR` and `WOODS` the algorithm with memory is superior in terms of number of gradient and function evaluations while it is comparable in term of CPU time. This is because improvements in terms of numbers of function and gradient evaluations are counterbalanced by the cost of updating the parameter μ_k. The only two cases where the algorithm without memory performs significantly better than the new algorithm are `COSINE` and `GENHUMPS`.

While it would be unwise to draw firm conclusions from these results, they indicate some promise for the new approach we have proposed in this paper. We appreciate that further

		Algorithm with no memory				Algorithm with memory			
Problem	n	$\#g$	$\#f$	Time	value	$\#g$	$\#f$	Time	value
BROYDN7D	1000	42	295	6.82	4.5533E+02	37	172	5.75	4.9660E+02
BRYBND	1000	15	43	3.14	1.0122E−14	19	29	4.20	4.6134E−14
CHAINWOO	1000	107	433	10.74	2.3580E+02	96	309	9.72	2.8984E+02
COSINE	1000	13	14	0.85	−9.9900E+02	22	23	1.58	−9.990E+02
CRAGGLVY	1000	15	15	1.40	3.3642E+02	14	14	1.40	3.3642E+02
CURLY10	1000	82	704	27.68	−1.0029E+05	107	973	36.37	−1.0025E+05
CURLY20	1000	71	492	89.61	−1.0030E+05	54	278	68.33	−1.0030E+05
CURLY30	1000	54	311	178.53	−1.0030E+05	112	919	375.31	−1.0023E+05
DIXMAANA	1500	12	12	1.59	1.0000E+00	12	12	1.75	1.0000E+00
DIXMAANB	1500	81	475	12.39	1.0000E+00	70	337	11.06	1.0000E+00
DIXMAANE	1500	33	124	4.78	1.0000E+00	33	88	5.22	1.0000E+00
DQRTIC	1000	31	31	1.07	1.4683E−07	29	29	1.18	1.4451E−07
FLETCHCR	1000	1475	1676	108.31	1.9184E−16	1442	1443	112.73	4.0489E−16
FREUROTH	1000	42	296	4.80	1.2147E+05	44	285	4.89	1.2147E+05
GENHUMPS	1000	2293	5065	203.44	7.8695E−15	3374	6977	308.79	2.2418E−14
GENROSE	1000	625	1352	48.07	1.0000E+00	571	882	44.73	1.0000E+00
NCB20B	1000	30	82	39.68	1.6760E+03	24	58	32.59	1.6760E+03
NONDQUAR	1000	18	18	5.56	1.0591E−09	17	17	5.31	1.6815E−09
PENALTY1	1000	41	43	2375.18	9.6862E−03	37	39	2159.24	9.6862E−03
POWELLSG	1000	18	18	0.73	4.2713E−07	18	18	0.78	1.2948E−08
QUARTC	1000	31	31	1.03	1.4683E−07	29	29	1.09	1.4451E−07
SINQUAD	1000	12	12	4.27	1.1721E−08	12	12	4.21	8.7381E−09
WOODS	1000	40	59	2.16	3.5323E−19	38	40	2.20	1.1238E−20

Table 4.1: A comparison of the new algorithm with memory with a memoryless algorithm

testing is needed to tune our algorithmic parameters, and to investigate other effects. Further investigations within our global framework will be the subject of future work.

5 Conclusion and perspectives

We have presented a linesearch algorithm that adds memory to the Newton model of the objective in the hope of making the method less myopic when applied on strongly nonlinear unconstrained optimization problems. We have also provided a global convergence theory for the new algorithm.

Our choice of the linesearch paradigm for unconstrained optimization is not the only possible one: one could equally consider the class of trust-region methods. We only mention here that a similar extension of these methods to include memory in Newton's model is also possible and will be described elsewhere.

Acknowledgments

Nick Gould and Philippe Toint are grateful to the Department of Informatics and Systems (Rome) for its hospitality. The first three authors appreciate the support provided by the British Council/MURST travel grant ROM/889/95/53.

References

I. Bongartz, A. R. Conn, N. I. M. Gould and Ph. L. Toint. CUTE: Constrained and unconstrained testing environment. *ACM Transaction on Mathematical Software*, **21**, 123–160, 1995.

S. H. Cheng and N. J. Higham. A modified Cholesky algorithm based on a symmetric indefinite factorization. Numerical Analysis Report No. 289, Manchester Centre for Computational Mathematics, Manchester, England, 1996.

R. S. Dembo and T. Steihaug. Truncated-Newton algorithms for large-scale unconstrained optimization. *Mathematical Programming*, **26**, 190–212, 1983.

J. E. Dennis and R. B. Schnabel. *Numerical methods for unconstrained optimization and nonlinear equations*. Prentice-Hall, Englewood Cliffs, New Jersey, 1983.

R. Fletcher. *Practical Methods of Optimization*. J. Wiley and Sons, Chicester and New York, second edn, 1987.

P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization*. Academic Press, London and New York, 1981.

L. Grippo, F. Lampariello, and S. Lucidi. A nonmonotone line search technique for Newton's method. *SIAM Journal on Numerical Analysis*, **23**(4), 707–716, 1986.

J. J. Moré. Recent developments in algorithms and software for trust region methods. *In* A. Bachem, M. Grötschel and B. Korte, eds, 'Mathematical Programming: The State of the Art', pp. 258–287, Springer Verlag, Heidelberg, Berlin, New York, 1983.

R. B. Schnabel and E. Eskow. A new modified Cholesky factorization. *SIAM Journal on Scientific Computing*, **11**(6), 1136–1158, 1991.

Ph. L. Toint. Global convergence of a class of trust region methods for nonconvex minimization in Hilbert space. *IMA Journal of Numerical Analysis*, **8**, 231–252, 1988.