Algebraic rules for quadratic regularization of Newton's method

E. W. Karas* S. A. Santos† B. F. Svaiter‡

August 6, 2013

Abstract

In this work we propose a class of quasi-Newton methods to minimize a twice differentiable function with Lipschitz continuous Hessian. These methods are based on the quadratic regularization of Newton's method, with algebraic explicit rules for computing the regularizing parameter. The convergence properties of this class of methods are analysed. We show that if the sequence generated by the algorithm converges then its limit point is stationary. We also establish local quadratic convergence in a neighborhood of a stationary point with positive definite Hessian. Encouraging preliminary numerical experiments are presented.

Keywords: smooth unconstrained minimization; Newton's method; regularization; global convergence; local convergence; computational results.

AMS Classification: 90C30, 90C53, 49M15.

Introduction

Quasi-Newton methods, which are modifications of Newton's method, are the main tools for solving smooth unconstrained minimization problems. These modifications address drawbacks of Newton's method in different ways and may be combined. Secant methods eliminate the computation of the Hessian of the objective function. Conjugate gradient methods eliminate the need of factoring the Hessian, requiring only its action upon vectors. Trust regions globalize the method, prevent convergence to local maxima and may improve its performance. Line searches are strategies that might be applyed to any quasi-Newton direction. This short list is by no means comprehensive. Detailed discussions about quasi-Newton methods can be found in [6, 7, 8, 10, 21] and the references therein.

The use of quadratic regularization dates back to the works of A. N. Tikhonov [23] for regularizing ill-posed problems, K. Levenberg [17] for solving non-linear least-squares problems, Goldfeld et al. [11] in the grounds of trust-region methods and Martinet [18] for solving variational inequalities. In this work we address the use of quadratic regularizations in the context of the smooth minimization

*Department of Mathematics, Federal University of Paraná, CP 19081, 81531-980, Curitiba, PR, Brazil. E-mail: ewkaras@ufpr.br. This author was partially supported by CNPq grants 307714/2011-0, 472313/2011-8.

†Department of Applied Mathematics, University of Campinas, Campinas, SP, Brazil. E-mail: sandra@ime.unicamp.br. This author was partially supported by CNPq grant 304032/2010-7, FAPESP grants 2013/05475-7, 2013/07375-0 and PRONEX Optimization.

‡IMPA, Estrada Dona Castorina 110, 22460-320, Rio de Janeiro, Brazil. E-mail: benar@impa.br. This author was partially supported by CNPq grants 302962/2011-5, 474944/2010-7, FAPERJ grant E-26/102.940/2011 and PRONEX Optimization.

problem, assuming that the (exact) gradient and Hessian of the objective function are available and that the Hessian is Lipschitz continuous. We propose analytic explicit rules for computing the quadratic regularization parameter based on an estimate of the Lipschitz constant, which is dynamicaly updated, and an estimate of the smallest eigenvalue of the Hessian. It is worth mentioning that such a specific eigenvalue can be efficiently estimated using the software ARPACK [16].

Hooked-constrained trust-region methods [15, 19] can also be interpreted as quadratic regularizations of Newton's method, where the regularizing parameter is the Lagrange multiplier of the trust-region constraint. In our approach, instead of updating a trust-region radius, we update the estimated Lispchitz constant, which is a measure for the departure from "quadraticity" of the objective function. Another possibility for regularizing Newton's method (for smooth minimization) is the use of cubic terms, as proposed by Griewank [13] and further developed by Nesterov and Polyak [20], Weiser et al. [25], Cartis et al.[2, 3, 4, 5] and Gould et al.[12].

The main features of the proposed rules for choosing the regularizing parameter are: (i) quadratic convergence under standard regularity conditions; (ii) if the estimated Lipschitz constant is larger than the actual Lipschitz constant and it is kept constant, then in all iterations the stepsize is one; (iii) encouraging (preliminary) numerical results. In view of itens (i) and (ii), with the Lipschitz constant at hand, one may choose *a priori* the regularizing parameter so that the full quasi-Newton step is accepted and quadratic convergence under usual assumptions is maintained.

We consider the unconstrained minimization problem

$$\min_{x\in\mathbb{R}^n} \; f(x) \tag{1}$$

assuming $f:\mathbb{R}^n\to\mathbb{R}$ to be a twice differentiable function with L-Lipschitz continuous Hessian, that is,

$$\|\nabla f^2(x) - \nabla^2 f(y)\| \le L\|x-y\| \qquad \forall x,y\in\mathbb{R}^n, \tag{2}$$

where at the right-hand side of this inequality we have the Euclidean norm and at the left-hand side we have the operator norm induced by the Euclidean norm. We are concerned with a quasi-Newton method in which the iterations are

$$s_k = -(\nabla^2 f(x_k) + \mu_k I)^{-1}\nabla f(x_k), \;\; x_{k+1} = x_k + t_k s_k$$

where:

1. $\mu_k \ge 0$ is chosen to guarantee $\nabla^2 f(x_k) + \mu_k I$ to be positive definite;
2. $t_k \in (0,1]$ is chosen to guarantee a "sufficient decrease" of f.

Item 2 is generally accomplished by an Armijo line search, in pure form, or modified with the incorporation of higher order information gathered at the mistrials.

Our concern is item 1. The parameter μ_k shall be neither "too small" nor "too large". If μ_k is chosen too small, the associated quasi-Newton step s_k may become "too large" and the linearization error at $x_k + s_k$ may be also too large, causing t_k to be too small and resulting in a small decrease of f. If μ_k is too large, the associated quasi-Newton step becomes too small, also resulting in a small decrease of f.

In view of assumption (2), when L is available one can bound the linearization error at quasi-Newton steps, and this *a priori* bound can be used for choosing μ_k. In general L is not available, but if one uses an approximation of it, data gathered at previous iterations can be used to improve

this estimation. Even when L is available, it is (in general) a crude *global* estimation, and the local or effective Lispchitz constant of $\nabla^2 f$ can be much smaller. In this case, the use of a dynamically updated approximation of L may improve the performance of the algorithm.

This work is organized as follows. In Sec. 1 we establish basic results useful for the formulation and analysis of the proposed algorithm. Sec. 2 describes the algorithm and some of its properties. The global and local convergence of the algorithm are proved in Sec. 3. Numerical experiments are discussed in Sec. 4, and our contributions are summarized in the final remarks of Sec. 5. An Appendix with the complete numerical results finishes the text.

1 Preliminary results

In this section we define the notation and establish theoretical basis for the formulation and analysis of the proposed algorithm. First we review some classical bounds for the linearization errors of a C^2 function with Lipschitz continuous Hessian.

Lemma 1.1 (See [7, Lemmas 4.1.12 and 4.1.14])**.** *For any $x, s \in \mathbb{R}^n$,*

$$\left\|\nabla f(x+s) - \big(\nabla f(x) + \nabla^2 f(x)s\big)\right\| \leq \frac{L}{2}\|s\|^2,$$
$$\left|f(x+s) - \left(f(x) + \langle s, \nabla f(x)\rangle + \frac{1}{2}\langle s, \nabla^2 f(x)s\rangle\right)\right| \leq \frac{L}{6}\|s\|^3.$$

Next we analyze the interrelationships among the linearization error at a quasi-Newton step, the (quadratic) regularization parameter, the length of the quasi-Newton step and intrinsic parameters of the problem, as the Hessian's Lipschitz constant and the smallest eingenvalue of the Hessian at the current point. From now on, $\lambda_{\min}(A)$ stands for the smallest eigenvalue of a symmetric matrix A.

Proposition 1.2. *Suppose that $\mu \in \mathbb{R}$, $(\nabla^2 f(x) + \mu I)s = -\nabla f(x)$ and let $\lambda = \lambda_{\min}(\nabla^2 f(x))$. Then*

1. $f(x+s) \leq f(x) + \frac{1}{2}\langle s, \nabla f(x)\rangle + \frac{\|s\|^2}{6}(L\|s\| - 3\mu)$ *and* $\langle s, \nabla f(x)\rangle \leq -(\lambda+\mu)\|s\|^2$;
2. *if $\lambda + \mu > 0$ then*

$$\langle s, \nabla f(x)\rangle \leq 0, \qquad \|s\| \leq \frac{\|\nabla f(x)\|}{\lambda+\mu}, \qquad L\|s\| - 3\mu \leq \frac{L\|\nabla f(x)\| - 3\mu(\lambda+\mu)}{\lambda+\mu};$$

3. *if $\nabla f(x) \neq 0$ and $\mu \geq \dfrac{\sqrt{\lambda^2 + 4L\|\nabla f(x)\|/a} - \lambda}{2}$ with $a \in (0, 3]$, then $\mu > 0$, $\lambda + \mu > 0$, and*

$$f(x+s) \leq f(x) + \frac{1}{2}\langle s, \nabla f(x)\rangle + \frac{\|s\|^2}{6}(a-3)\mu \leq f(x) + \frac{1}{2}\langle s, \nabla f(x)\rangle.$$

Proof. To prove the first inequality in item 1, use the second inequality in Lemma 1.1 to conclude that

$$f(x+s) \leq f(x) + \langle s, \nabla f(x)\rangle + \frac{1}{2}\langle \nabla^2 f(x)s, s\rangle + \frac{L}{6}\|s\|^3,$$

observe that (in view of the assumption on s) $\nabla^2 f(x)s = -(\nabla f(x) + \mu s)$, and combine these two results.

Since $\lambda + \mu = \lambda_{\min}(\nabla^2 f(x) + \mu I)$,

$$(\lambda + \mu)\|s\|^2 \leq \langle s, (\nabla^2 f(x) + \mu I)s\rangle = -\langle s, \nabla f(x)\rangle \leq \|s\|\|\nabla f(x)\|$$

where the equality follows from the assumption on s and the second inequality follows from Cauchy-Schwartz inequality. The second inequality in item 1, as well as the two first inequalities of item 2 follow trivially from the above result. The third inequality in item 2 follows from the second one in item 2.

To prove item 3, assume that its premises hold. First observe that the largest root of the quadratic function $q(t) = L\|\nabla f(x)\| - at(t + \lambda)$ is

$$\mu^* = \frac{\sqrt{\lambda^2 + 4L\|\nabla f(x)\|/a} - \lambda}{2} > 0.$$

Observe also that $\lambda + \mu^* > 0$. Since $\mu \geq \mu^*$, $\mu > 0$ and $\lambda + \mu > 0$. Moreover, using also the fact that the quadratic q is concave we have

$$\begin{aligned} L\|\nabla f(x)\| - 3\mu(\lambda + \mu) &= L\|\nabla f(x)\| - a\mu(\lambda + \mu) + (a - 3)\mu(\lambda + \mu) \\ &\leq (a - 3)\mu(\lambda + \mu) \leq 0 \end{aligned}$$

and the conclusion follows from the above inequalities, the first inequality in item 1 and the last inequality in item 2. □

In the minimization scenario, a vector s is a *descent diretion* at x if $\langle s, \nabla f(x)\rangle < 0$. Recall that the *Armijo rule* for choosing the stepsize t along a descent direction s at x, is given by

$$f(x + ts) \leq f(x) + \beta t\langle s, \nabla f(x)\rangle,$$

with $0 < \beta < 1/2$. Choosing β within this range guarantees that the full Newton step is accepted for minimizing a strictly convex quadratic function. The next result is instrumental for our aim, which is to define analytic rules for computing the quadratic regularization parameter.

Proposition 1.3. *For any $x \in \mathbb{R}^n$, the following conditions are sufficient for the quasi-Newton direction*

$$s = -(\nabla^2 f(x) + \mu I)^{-1}\nabla f(x)$$

to be a well-defined descent direction, accepted with stepsize 1 on the Armijo search:

$$\lambda_{\min}(\nabla^2 f(x)) + \mu > 0, \tag{3}$$

$$L\|s\| - 3\mu < 0. \tag{4}$$

Moreover, using the notation $\lambda = \lambda_{\min}(\nabla^2 f(x))$, the above two conditions hold if $\nabla f(x) \neq 0$ and

$$\mu \geq \frac{\sqrt{\lambda^2 + 4L\|\nabla f(x)\|} - \lambda}{2}.$$

Proof. Condition (3) guarantees that $\lambda_{\min}(\nabla^2 f(x) + \mu I) > 0$, which implies that s is well defined. The remaining results follow from Proposition 1.2. □

2 The algorithm

In this section we propose the algorithm and we analyze some of its basic properties.

Algorithm 1.

Input : $x_0 \in \mathbb{R}^n$, $\beta \in (0, 1/2)$, $\eta \in [0, 1)$, $L_0 > 0$, and $\delta \in [0, L_0]$

1. $k \leftarrow 0$
2. **while** $\nabla f(x_k) \neq 0$ **do**
3. Compute $\lambda_k = \lambda_{\min}(\nabla^2 f(x_k))$
4. Choose $\mu_k \in [\mu_k^-, \mu_k^+]$ where

$$\mu_k^- = \frac{\sqrt{(\lambda_k)^2 + 4L_k\|\nabla f(x_k)\|} - \lambda_k}{2},\ \mu_k^+ = \max\{-\lambda_k, 0\} + \sqrt{L_k\|\nabla f(x_k)\|}$$

5. Compute $s_k = -(\nabla^2 f(x_k) + \mu_k I)^{-1}\nabla f(x_k)$
6. $t \leftarrow 1$
7. **while** $f(x_k + ts_k) > f(x_k) + \beta t\langle s_k, \nabla f(x_k)\rangle$ **do**
8. Choose $\theta \in [0.1, 0.9]$ and set $t \leftarrow \theta t$
9. **end while**
10. $t_k = t$
11. $x_{k+1} = x_k + t_k s_k$
12. **if** $t_k < 1$ **then**
13. $L_{k+1} = 2L_k$
14. **else**
15. Ared $= f(x_k) - f(x_{k+1})$, Pred $= -\left[\langle s_k, \nabla f(x_k)\rangle + \langle(\mu_k I + \nabla^2 f(x_k))s_k, s_k\rangle/2\right]$
16. **if** Ared $> \eta$ Pred **then**
17. $L_{k+1} = \max\{L_k/2, \delta\}$
18. **else**
19. $L_{k+1} = L_k$
20. **end if**
21. **end if**
22. $k \leftarrow k + 1$
23. **end while**

Some remarks about this algorithm are in order.

i) Iteration m begins with $k = m - 1$, and ends with $k = m$ if $\nabla f(x_{m-1}) \neq 0$.

ii) If the algorithm does not end at iteration $k + 1$, then

$$\lambda_{\min}(\nabla^2 f(x_k) + \mu_k I) = \lambda_k + \mu_k > 0. \tag{5}$$

Hence, s_k is well defined, it is a descent direction, and the Armijo line search on Steps **7**-9 of Algorithm 1 has finite termination.

Altogether, Algorithm 1 is well defined and either it terminates after m steps with $\nabla f(x_{m-1}) = 0$ or it generates infinite sequences (x_k), (s_k), (t_k), (λ_k), (μ_k) and (L_k).

iii) The backtracking procedure of the Armijo search may be implemented by bisection or polynomial interpolation. This point is detailed in Section 4.

iv) L_k plays the role of the Lipschitz constant of $\nabla^2 f$. If $\delta > 0$, this parameter plays the role of a safeguard which prevents L_k from becoming too small.

v) The value of λ_k could be replaced by a lower bound of the smallest eigenvalue of $\nabla^2 f(x_k)$.

vi) If λ_k is an upper approximation of the smallest eigenvalue of $\nabla^2 f(x_k)$ and $\nabla^2 f(x_k) + \mu_k I$ is *not* positive definite, this condition is detected during a Cholesky factorization of this operator, and a smaller value of λ_k may be tested/used.

From now on in this paper we assume that Algorithm 1 with input $x_0 \in \mathbb{R}^n$, $\beta \in (0, 1/2)$, $\eta \in [0, 1)$, $L_0 > 0$ and $\delta \in [0, L_0]$ does not stop at Step **2** and that (x_k), (s_k), (t_k), (λ_k), (μ_k),(L_k) are the (infinite) sequences generated by it. First we analyze the basic properties of Algorithm 1.

Proposition 2.1. *If $L_k \geq L$, then $t_k = 1$ and $L_{k+1} = \max\{L_k/2, \delta\}$.*

Proof. Suppose that $L_k \geq L$. It follows from this assumption and the definition of μ_k (in Step **4**) that

$$\mu_k \geq \frac{\sqrt{(\lambda_k)^2 + 4L\|\nabla f(x_k)\|} - \lambda_k}{2}.$$

Since λ_k is the smallest eigenvalue of $\nabla^2 f(x_k)$, it follows from the above inequality, the definition of s_k (in Step **5**), and item 3 of Proposition 1.2 with $\mu = \mu_k$, $x = x_k$, $s = s_k$, $a = 1$ that

$$f(x_k + s_k) \leq f(x_k) + \frac{1}{2}\langle s_k, \nabla f(x_k)\rangle.$$

Moreover, using the definition of Pred and (again) the definition of s_k we have

$$\text{Pred} = -\frac{1}{2}\langle \nabla f(x_k), s_k\rangle.$$

The conclusion follows from the two above equations and the algorithm's definition. □

Proposition 2.2. *For all k,*

$$L_{k+1} \leq \max\{2L, L_k/2, \delta\} \quad \textit{and} \quad \delta \leq L_k \leq \max\{L_0, 2L\}; \tag{6}$$

and for infinitely many ks,

$$t_k = 1.$$

Proof. To prove the first inequality, take $k \in \mathbb{N}$. If $L_k \geq L$, using Proposition 2.1, we conclude that $L_{k+1} = \max\{L_k/2, \delta\}$. If $L_k < L$ then

$$L_{k+1} \in \{2L_k, \max\{L_k/2, \delta\}, L_k\}, \quad \max\{2L_k, L_k\} < 2L,$$

which implies again the first inequality.

The second inequality follows trivially from the algorithm's definition. To prove the third inequality, use induction in k. This inequality holds trivially for $k = 0$. Suppose it holds for some k. Using the first inequality and the fact that $\delta \leq L_0$ we conclude that

$$L_{k+1} \leq \max\{2L, \max\{L_0, 2L\}/2, \delta\} \leq \max\{L_0, 2L\}$$

which completes the induction proof.

To prove the second part of the corollary, suppose that $t_k < 1$ for any $k \geq k_0$. Then

$$L_k = 2^{k-k_0} L_{k_0}, \qquad k = k_0, k_0 + 1, \ldots$$

in contradiction with the last inequality in (6). □

It follows from Proposition 2.2 that $L_k \leq \max\{2L, \delta\}$, for all $k \geq \lfloor \log_2(L_0/L) \rfloor$.

Proposition 2.3. *For each k*

$$f(x_{k+1}) \leq f(x_k) + \beta t_k \langle s_k, \nabla f(x_k) \rangle \leq f(x_k) - \beta t_k \frac{\|\nabla f(x_k)\|^2}{\|\nabla^2 f(x_k)\| + \mu_k} \tag{7}$$

where $\|\nabla^2 f(x_k)\|$ is the operator norm induced by the Euclidean norm in $\mathbb{R}^n$. In particular, the sequence $(f(x_k))$ is strictly decreasing.

Proof. The first inequality follows from the stopping condition for the Armijo line search (Steps **7**-9). To prove the second inequality, observe that since $\nabla^2 f(x_k) + \mu_k I$ is positive definite,

$$\begin{aligned} -\langle s_k, \nabla f(x_k) \rangle &= \langle \nabla f(x_k), (\nabla^2 f(x_k) + \mu_k I)^{-1} \nabla f(x_k) \rangle \\ &\geq \frac{\|\nabla f(x_k)\|^2}{\lambda_{\max}(\nabla^2 f(x_k)) + \mu_k} \end{aligned}$$

where $\lambda_{\max}(\cdot)$ is the (signed) largest-eigenvalue function. □

Corollary 2.4. *If $\bar{x} \in \mathbb{R}^n$ is an accumulation point of (x_k), then*

$$\sum_{k=0}^{\infty} \beta t_k \frac{\|\nabla f(x_k)\|^2}{\|\nabla^2 f(x_k)\| + \mu_k} \leq f(x_0) - f(\bar{x}) < \infty. \tag{8}$$

Proof. Since $\bar{x} \in \mathbb{R}^n$ is an accumulation point there exists a subsequence $(x_{k_j})_{j \in \mathbb{N}}$ which converges to $\bar{x}$. Direct use of Proposition 2.3 shows that for any $m \geq 0$,

$$\sum_{k=0}^{m} \beta t_k \frac{\|\nabla f(x_k)\|^2}{\|\nabla^2 f(x_k)\| + \mu_k} \leq f(x_0) - f(x_{m+1}).$$

Since f is continuous, $(f(x_{k_j}))$ converges to $f(\bar{x})$. To end the proof, take the lim sup on both sides of the above inequality and observe that $(f(x_k))$, being strictly decreasing, converges to $f(\bar{x})$. □

3 Convergence analysis

In this section we analyze convergence properties of Algorithm 1. We show that if the sequence generated by the algorithm converges then its limit point is stationary. We also establish local quadratic convergence in a neighborhood of a stationary point with positive definite Hessian.

If an upper bound for L is available, Algorithm 1 may be initialized so that all steps are "full" quasi-Newton steps, that is, $t_k = 1$ for all k.

Proposition 3.1. *If $\delta \geq L$, then $t_k = 1$ for all k.*

Proof. If $\delta \geq L$, Proposition 2.2, $L_k \geq L$ for all k and the conclusion follows from Proposition 2.1. $\square$

Even if L is available, it may be "too pessimistic" to use $\delta = L$, because the global Lispchitz constant may be much greater than the Lipschitz constant of the Hessian restricted to the line segment $[x_k, x_k + s_k]$. So, it is reasonable to consider using $\delta \ll L$ and to analyze the algorithm without the assumption $\delta \geq L$.

As in most quasi-Newton methods, it is important to analyze the iterations of Algorithm 1 in which unitary steps are used. With this aim we define the set of iterations with unitary steps:

$$\mathcal{K} = \{k \mid t_k = 1\}. \tag{9}$$

By Proposition 2.2 this set is infinite. This fact will be used for proving the next result.

Lemma 3.2. *If the sequence (x_k) is bounded, then it has a stationary accumulation point. Moreover, if this sequence converges, its limit is stationary.*

Proof. To prove the first statement, suppose that (x_k) is bounded. Since $\mathcal{K}$ is infinite, there exists a subsequence $(x_{k_j})_{j\in\mathbb{N}}$ and $\bar{x} \in \mathbb{R}^n$ such that

$$\lim_{j\to\infty} x_{k_j} = \bar{x}, \qquad k_j \in \mathcal{K}, \;\; j = 1, 2, \ldots.$$

Using Corollary 2.4 and the fact that all terms in the series on (8) are non-negative, we conclude that

$$\sum_{j=1}^{\infty} \beta \frac{\|\nabla f(x_{k_j})\|^2}{\|\nabla^2 f(x_{k_j})\| + \mu_{k_j}} = \sum_{j=1}^{\infty} \beta t_{k_j} \frac{\|\nabla f(x_{k_j})\|^2}{\|\nabla^2 f(x_{k_j})\| + \mu_{k_j}} \leq f(x_0) - f(\bar{x}) < \infty,$$

where the first equality follows from the definition of the set $\mathcal{K}$. Since $(\nabla^2 f(x_{k_j}))$ and (μ_{k_j}) are bounded, it follows from the above equation that $(\nabla f(x_{k_j}))$ converges to 0 and hence $\bar{x}$ is stationary.

The second statement of the lemma follows trivially from the first one. $\square$

Theorem 3.3. *If $\delta > 0$, then all accumulation points of the sequence (x_k) are stationary.*

Proof. For any $k \in \mathbb{N}$ and $t \in [0, 1]$, by Lemma 1.1 with $x = x_k$ and $s = ts_k$, and the definition of s_k, we have

$$\begin{aligned}
f(x_k + ts_k) &\leq f(x_k) + t\langle s_k, \nabla f(x_k)\rangle + \frac{t^2}{2}\langle s_k, \nabla^2 f(x_k) s_k\rangle + t^3 \frac{L}{6}\|s_k\|^3 \\
&= f(x_k) + t\langle s_k, \nabla f(x_k)\rangle - \frac{t^2}{2}\langle s_k, \nabla f(x_k)\rangle - \frac{t^2}{2}\mu_k\|s_k\|^2 + t^3\frac{L}{6}\|s_k\|^3 \\
&= f(x_k) + \beta t\langle s_k, \nabla f(x_k)\rangle + \left(\frac{1}{2} - \beta\right) t\langle s_k, \nabla f(x_k)\rangle + \frac{t}{2}(1-t)\langle s_k, \nabla f(x_k)\rangle - \\
&\quad - \frac{t^2}{2}\mu_k\|s_k\|^2 + t^3\frac{L}{6}\|s_k\|^3.
\end{aligned}$$

Using the fact that $\mu_k \geq 0$, the first inequality at item 2, the second one at item 1 and the second inequality at item 2 of Proposition 1.2, we obtain

$$\begin{aligned} f(x_k + ts_k) &\leq f(x_k) + \beta t\langle s_k, \nabla f(x_k)\rangle + \left(\frac{1}{2} - \beta\right) t\langle s_k, \nabla f(x_k)\rangle + t^3\frac{L}{6}\|s_k\|^3 \\ &\leq f(x_k) + \beta t\langle s_k, \nabla f(x_k)\rangle - \left(\frac{1}{2} - \beta\right) t(\lambda_k + \mu_k)\|s_k\|^2 + t^3\frac{L}{6}\|s_k\|^3 \\ &= f(x_k) + \beta t\langle s_k, \nabla f(x_k)\rangle + t\frac{\|s_k\|^2}{6}\left(Lt^2\|s_k\| - 6\left(\frac{1}{2} - \beta\right)(\lambda_k + \mu_k)\right) \\ &\leq f(x_k) + \beta t\langle s_k, \nabla f(x_k)\rangle + t\frac{\|s_k\|^2}{6(\lambda_k + \mu_k)}\left(Lt^2\|\nabla f(x_k)\| - 6\left(\frac{1}{2} - \beta\right)(\lambda_k + \mu_k)^2\right). \end{aligned} \tag{10}$$

But, as $L_k \geq \delta$, we have

$$\begin{aligned} \lambda_k + \mu_k \geq \lambda_k + \mu_k^- &\geq \frac{\sqrt{(\lambda_k)^2 + 4\delta\|\nabla f(x_k)\|} + \lambda_k}{2} \\ &\geq \frac{\sqrt{(\lambda_k)^2 + 4\delta\|\nabla f(x_k)\|} - |\lambda_k|}{2} \\ &= \frac{2\delta\|\nabla f(x_k)\|}{\sqrt{(\lambda_k)^2 + 4\delta\|\nabla f(x_k)\|} + |\lambda_k|}. \end{aligned}$$

Suppose that $\bar{x} \in \mathbb{R}^n$ is an accumulation point of (x_k). This means that there exists a subsequence (x_{k_j}) which converges to $\bar{x}$. In particular, this subsequence is bounded and there exists $M > 0$ such that

$$\sqrt{(\lambda_{k_j})^2 + 4\delta\|\nabla f(x_{k_j})\|} + |\lambda_{k_j}| \leq M < \infty, \qquad \text{for all } j \in \mathbb{N}.$$

Consequently, $\lambda_{k_j} + \mu_{k_j} \geq 2\delta\|\nabla f(x_{k_j})\|/M$ and

$$Lt^2\|\nabla f(x_{k_j})\| - 6\left(\frac{1}{2} - \beta\right)(\lambda_{k_j} + \mu_{k_j})^2 \leq \|\nabla f(x_{k_j})\|\left(Lt^2 - \gamma\|\nabla f(x_{k_j})\|\right) \tag{11}$$

where

$$\gamma = 6\left(\frac{1}{2} - \beta\right)\left(\frac{2\delta}{M}\right)^2 > 0.$$

Combining (10) with (11), we have that

$$f(x_{k_j} + ts_{k_j}) \leq f(x_{k_j}) + \beta t\langle s_{k_j}, \nabla f(x_{k_j})\rangle, \quad \text{for all } \ t \in \left[0, \sqrt{\gamma\|\nabla f(x_{k_j})\|/L}\right] \cap [0, 1].$$

Therefore, by Steps **6**-10 of the algorithm,

$$t_{k_j} \geq \min\left\{1, \frac{\sqrt{\gamma\|\nabla f(x_{k_j})\|/L}}{2}\right\}. \tag{12}$$

Using Corollary 2.4 and the fact that all terms in the series on (8) are non-negative, we conclude that

$$\sum_{j=1}^{\infty} \beta t_{k_j} \frac{\|\nabla f(x_{k_j})\|^2}{\|\nabla^2 f(x_{k_j})\| + \mu_{k_j}} \leq f(x_0) - f(\bar{x}) < \infty.$$

Since (x_{k_j}) is bounded, $\|\nabla^2 f(x_{k_j})\| + \mu_{k_j}$ is also bounded and

$$\sum_{j=1}^{\infty} t_{k_j} \|\nabla f(x_{k_j})\|^2 < \infty$$

which combined with (12) shows that $\|\nabla f(x_{k_j})\|$ converges to zero. □

For proving the quadratic convergence of the sequence (x_k), under suitable regularity conditions, we need the following two technical results. The first one provides a bound for the distance to a stationary point after an iteration with unitary stepsize.

Lemma 3.4. *If $x_\star \in \mathbb{R}^n$ is a stationary point of f, then for all $k \in \mathcal{K}$*

$$\|x_{k+1} - x_\star\| \leq \frac{1}{\lambda_k + \mu_k} \left(\frac{L}{2} \|x_k - x_\star\|^2 + \mu_k \|x_k - x_\star\| \right). \tag{13}$$

Proof. By the algorithm, we have for all $k \in \mathcal{K}$

$$\begin{aligned} x_{k+1} - x_\star &= x_k - x_\star - (\nabla^2 f(x_k) + \mu_k I)^{-1} \nabla f(x_k) \\ &= \left(\nabla^2 f(x_k) + \mu_k I\right)^{-1} \left(\left(\nabla^2 f(x_k) + \mu_k I\right)(x_k - x_\star) - \nabla f(x_k) \right). \end{aligned}$$

As $x_\star$ is a stationary point, using also triangle inequality, and Lemma 1.1 with $x = x_k$ and $s = x_\star - x_k$, we have

$$\begin{aligned} \left\| \left(\nabla^2 f(x_k) + \mu_k I\right)(x_k - x_\star) - \nabla f(x_k) \right\| &\leq \left\| \nabla f(x_\star) - \nabla f(x_k) - \nabla^2 f(x_k)(x_\star - x_k) \right\| + \mu_k \|x_k - x_\star\| \\ &\leq \frac{L}{2} \|x_k - x_\star\|^2 + \mu_k \|x_k - x_\star\|. \end{aligned}$$

It follows from (5) that $\| \left(\nabla^2 f(x_k) + \mu_k I\right)^{-1} \| = (\lambda_k + \mu_k)^{-1}$, which combined with the above results yields the desired inequality. □

The next result shows that the stepsize is one close to a stationary point in which the Hessian is positive definite.

Lemma 3.5. *Suppose that $x_\star \in \mathbb{R}^n$ is a stationary point of f, $\nabla^2 f(x_\star)$ is positive definite, and let $\lambda_\star$ be the smallest eigenvalue of $\nabla^2 f(x_\star)$. Then there exists a neighborhood V of $x_\star$ such that*

$$\lambda_k > \frac{\lambda_\star}{2} \quad \textit{and} \quad t_k = 1$$

whenever $x_k \in V$.

Proof. Using Proposition 1.2 item 1, the facts that $\beta \in (0, 1/2)$ and $\mu_k \geq 0$, and Proposition 1.2 item 2, we have

$$\begin{aligned} f(x_k + s_k) &\leq f(x_k) + \frac{1}{2}\langle s_k, \nabla f(x_k)\rangle + \frac{\|s_k\|^2}{6}\left(L\|s_k\| - 3\mu_k\right) \\ &= f(x_k) + \beta\langle s_k, \nabla f(x_k)\rangle + \left(\frac{1}{2} - \beta\right)\langle s_k, \nabla f(x_k)\rangle + \frac{\|s_k\|^2}{6}\left(L\|s_k\| - 3\mu_k\right) \\ &\leq f(x_k) + \beta\langle s_k, \nabla f(x_k)\rangle - \left(\frac{1}{2} - \beta\right)(\lambda_k + \mu_k)\|s_k\|^2 + \frac{\|s_k\|^2}{6}\left(L\|s_k\| - 3\mu_k\right) \\ &\leq f(x_k) + \beta\langle s_k, \nabla f(x_k)\rangle + \left(L\|s_k\| - 6\left(\frac{1}{2} - \beta\right)\lambda_k\right)\frac{\|s_k\|^2}{6} \\ &\leq f(x_k) + \beta\langle s_k, \nabla f(x_k)\rangle + \left(\frac{L}{\lambda_k + \mu_k}\|\nabla f(x_k)\| - 6\left(\frac{1}{2} - \beta\right)\lambda_k\right)\frac{\|s_k\|^2}{6}, \end{aligned} \tag{14}$$

for *all* $k \in \mathbb{N}$.

Suppose that $x_\star \in \mathbb{R}^n$ is a stationary point of f and $\nabla^2 f(x_\star)$ is positive definite, i.e,

$$\nabla f(x_\star) = 0 \quad \text{and} \quad \lambda_{\min}(\nabla^2 f(x_\star)) = \lambda_\star > 0.$$

Since $f \in \mathcal{C}^2$ and $\beta \in (0, 1/2)$, there exists a neighborhood V of $x_\star$ such that

$$\|\nabla f(x)\| \leq \frac{3}{2L}\left(\frac{1}{2} - \beta\right)\lambda_\star^2 \quad \text{and} \quad \lambda_{\min}(\nabla^2 f(x)) \geq \frac{\lambda_\star}{2} \quad \text{for all } \ x \in V.$$

Assume that $x_k \in V$. Then, the two above inequalities hold for $x = x_k$ and, since $\mu_k \geq 0$, we have

$$\begin{aligned} \frac{L}{\lambda_k + \mu_k}\|\nabla f(x_k)\| - 6\left(\frac{1}{2} - \beta\right)\lambda_k &\leq \frac{L}{\lambda_k}\|\nabla f(x_k)\| - 6\left(\frac{1}{2} - \beta\right)\lambda_k \\ &\leq \frac{2L}{\lambda_\star}\|\nabla f(x_k)\| - 3\left(\frac{1}{2} - \beta\right)\lambda_\star \leq 0. \end{aligned}$$

Combining this inequality with (14) and using Steps **6**-10 we conclude that $t_k = 1$. □

Now we are ready to establish local quadratic convergence under suitable assumptions.

Theorem 3.6. *Suppose that $x_\star \in \mathbb{R}^n$ is a stationary point of f and $\nabla^2 f(x_\star)$ is positive definite. Then there exists $r > 0$ such that the sequence (x_k) converges to $x_\star$ with q-order at least $3/2$ whenever $x_{k_0} \in B(x_\star, r)$ for some $k_0 \in \mathbb{N}$. Moreover, if $\mu_k = \mu_k^-$ for all $k \in \mathbb{N}$, then the convergence is q-quadratic.*

Proof. Let V be as in Lemma 3.5. There exists $r_0 > 0$ such that $B(x_\star, r_0) \subset V$. Suppose that

$$x_k \in B(x_\star, r_0).$$

By Lemma 3.5

$$\lambda_k > \frac{\lambda_\star}{2} \quad \text{and} \quad t_k = 1, \tag{15}$$

where $\lambda_\star > 0$ is the smallest eigenvalue of $\nabla^2 f(x_\star)$. Using the above relations, Lemma 3.4 and the inequality $\mu_k \geq 0$ we have

$$\begin{aligned}\|x_{k+1} - x_\star\| &\leq \frac{1}{\lambda_k + \mu_k}\left(\frac{L}{2}\|x_k - x_\star\|^2 + \mu_k\|x_k - x_\star\|\right) \\ &\leq \frac{2}{\lambda_\star}\left(\frac{L}{2}\|x_k - x_\star\|^2 + \mu_k\|x_k - x_\star\|\right). \end{aligned} \tag{16}$$

From the first inequality of Lemma 1.1 with $x = x_\star$ and $s = x_k - x_\star$; the assumption that $x_\star$ is a stationary point; and the triangle inequality it follows that

$$\begin{aligned}\|\nabla f(x_k)\| &\leq \|\nabla^2 f(x_\star)(x_k - x_\star)\| + \frac{L}{2}\|x_k - x_\star\|^2 \\ &\leq \left(\|\nabla^2 f(x_\star)\| + \frac{L}{2}\|x_k - x_\star\|\right)\|x_k - x_\star\|. \end{aligned} \tag{17}$$

By Proposition 2.2, $L_k \leq \bar{L}$ with $\bar{L} = \max\{L_0, 2L\}$. Therefore, using also the inequality $\mu_k \leq \mu_k^+$, the fact that $\lambda_k > 0$ implied by the first inequality in (15), and (17) we have

$$\begin{aligned}\mu_k &\leq \sqrt{\bar{L}\|\nabla f(x_k)\|} \\ &\leq \sqrt{\bar{L}\left(\|\nabla^2 f(x_\star)\| + \frac{L}{2}\|x_k - x_\star\|\right)}\ \|x_k - x_\star\|^{1/2}. \end{aligned} \tag{18}$$

Using (18) in (16) and the assumption $x_k \in B(x_\star, r_0)$ we conclude that

$$\|x_{k+1} - x_\star\| \leq \left(\frac{L}{\lambda_\star}\sqrt{r_0} + \frac{2}{\lambda_\star}\sqrt{\bar{L}\left(\|\nabla^2 f(x_\star)\| + \frac{L}{2}r_0\right)}\right)\|x_k - x_\star\|^{3/2} = \alpha\|x_k - x_\star\|^{3/2}, \tag{19}$$

where

$$\alpha = \frac{L}{\lambda_\star}\sqrt{r_0} + \frac{2}{\lambda_\star}\sqrt{\bar{L}\left(\|\nabla^2 f(x_\star)\| + \frac{L}{2}r_0\right)}.$$

To prove the first part of the theorem, define $r = \min\{r_0, 1/(4\alpha^2)\}$ and observe that if $x_{k_0} \in B(x_\star, r)$ then (19) holds for $k = k_0$, the sequence converges to $x_\star$ and the convergence rate is order of $3/2$.

To prove the second part of the theorem, suppose that $x_{k_0} \in B(x_\star, r)$. We already know that under this assumption the sequence converges to $x_\star$. If $\mu_k = \mu_k^-$ for all $k \in \mathbb{N}$, then using again the bound $L_k \leq \bar{L}$, the concavity of the square root, the first inequality of (15) and (17) we have, for k large enough,

$$\begin{aligned}\mu_k \leq \frac{\sqrt{(\lambda_k)^2 + 4\bar{L}\|\nabla f(x_k)\|} - \lambda_k}{2} &\leq \frac{\bar{L}}{\lambda_k}\|\nabla f(x_k)\| \\ &\leq \frac{2\bar{L}}{\lambda_\star}\left(\|\nabla^2 f(x_\star)\| + \frac{L}{2}r\right)\|x_k - x_\star\|. \end{aligned}$$

Using the above inequalities in (16) and defining

$$\tilde{\alpha} = \frac{L}{\lambda_\star} + \frac{4\bar{L}\|\nabla^2 f(x_\star)\|}{\lambda_\star^2} + \frac{2L\bar{L}}{\lambda_\star^2}r,$$

we conclude that

$$\|x_{k+1} - x_\star\| \leq \tilde{\alpha}\|x_k - x_\star\|^2,$$

which completes the proof. □

4 Numerical experiments

The tests were performed using the `gfortran` compiler (32-bits), version gcc-4.6, in a notebook Sony Vaio VGN-SR140E, Intel Centrino 2, 2.26GHz, RAM of 3 Gb and Cache of 3Mb. Algorithm 1 was run with the input $\beta = 10^{-2}$, $\eta = 0.25$, $L_0 = 10^{-6}$ and $\delta = 10^{-16}$.

The routines `dsaupd` and `dseupd` of `ARPACK` [16] were used to compute the smallest eigenvalue of the Hessians. `ARPACK` is a sofware package that implements the *implicitly restarted Arnoldi method*, a particular variant of Krylov subspace projection methods. It combines the implicitly shifted QR mechanism with an ℓ-step Arnoldi (or Lanczos) factorization to obtain a truncated form of the implicitly shifted QR-iteration. `ARPACK` is suitable for large-scale problems, as the implicit restarting provides a means to approximate a few eigenvalues with user specified properties in space proportional to nn_e, where n_e is the number of eigenvalues sought, and n is the problem size. Moreover, its reverse communication interface offers a convenient way to exchange information with application codes without imposing a structure on the user's matrix or the way a matrix-vector product is accomplished. In the implementation, we have essentially used the parameter choices suggested in the example that comes with the code, except for the length of the Arnoldi factorization (`ncv`), that was increased from 20 to 30, and the maximum number of Arnoldi iterations allowed (`maxitr`), that was decreased from 300 to 200.

The parameter μ_k was set either as the lower bound μ_k^- or as the upper bound μ_k^+. The step direction s_k was computed using the Cholesky factorization routines `dpotf2` and `dpotrs` of `LAPACK`[1]. The implementation also rested upon `BLAS`[2]. Although, theoretically, the matrix $\nabla^2 f(x_k) + \mu_k I$ should be positive definite, a safeguard was adopted to ensure positive definiteness of the regularized Hessian in the presence of a poor estimate for its smallest eigenvalue λ_k, as described next. We set $\Delta\mu = \frac{\|\nabla^2 f(x_k)\|_F}{100\sqrt{n}}$, where $\|\cdot\|_F$ is the Frobenius norm and n is the problem dimension. In case the Cholesky factorization fails, the successively more aggressive shifts

$$\nabla^2 f(x_k) + (\mu_k + j^2\Delta\mu)I, \ j = 1, 2, \ldots$$

are applied until a successful Cholesky factorization is produced. As corroborated by the numerical results, extra factorizations are an exception and not a rule. Moreover, the impact of computing the smallest eigenvalue is somehow attenuated in the calculation of the step direction s_k. The adopted line search strategy combines the classic quadratic and cubic interpolation, as in [7].

The test problems are 36 selected medium-sized ($100 \leq n \leq 1024$) unconstrained instances from the `CUTEr` collection, which include ill-conditioned problems (highlighted with $^{(*)}$ in the tables of the Appendix), analyzed in [14] and [9] as well. To put our approach in perspective, the same set of test problems were addressed by the benchmark modular code `uncmin`, developed by Schnabel, Koontz and Weiss [22] (available within the `CUTEr` platform).

[1] `http://www.netlib.org/lapack/`
[2] `http://www.netlib.org/blas/`

Summing up, there are three strategies under analysis, which are:

- S_1: `uncmin`;
- S_2: Algorithm 1 with $\mu_k \equiv \mu_k^-$;
- S_3: Algorithm 1 with $\mu_k \equiv \mu_k^+$.

To standardize the comparative results, the following choices were made upon `uncmin`: the analytic gradient and Hessian were used; the interpolation-based line search was adopted, and no scaling were performed. It is worth mentioning that the Newtonian linear systems of `uncmin` are solved by the *modified Cholesky factorization* of Gill and Murray (cf. [10, Algorithm MC, p.111]), that produces a convenient diagonal shift to obtain a safely positive definite modified Hessian. The stopping criteria of the Algorithm 1 were the following:

AS - Absolute stationarity: $\|\nabla f(x_k)\| \leq 10^{-6}$

RS - Relative stationarity: $\|\nabla f(x_k)\| \leq 10^{-15}\|\nabla f(x_0)\|$

SI - Successive iterates within tolerance: $\|s_k\| \leq 10^{-12}$

MF - Maximum number of 10000 functional evaluations exceeded.

Tables 1, 2 and 3 (see Appendix) contain the numerical output for each strategy, namely, the problem name; its dimension (n); the number of performed iterations (#Iter) and function evaluations (#Fun); the objective function value at the last iterate (f^*); the norm of the gradient at the last iterate ($\|g^*\|$); the demanded CPU time in seconds (CPU), and the reason for stopping (exit). In Table 1, the reasons for stopping are similar to those of Algorithm 1, being denoted by RS (relative stationarity), SI (successive iterates within tolerance), and MI (maximum number of 10000 iterations exceeded). Tables 2 and 3 also include a column with the number of performed Cholesky factorizations per iteration (#Chol/#Iter).

Let f_1^*, f_2^*, f_3^* be the objective function value obtained by strategies S_1, S_2, S_3 when applied to a given problem, runned with the default `CUTEr` initial point. We consider that the strategy S_i *found a solution* if

$$\frac{f_i^* - f_{\min}}{\max\{1, |f_{\min}|\}} \leq 0.01,$$

where $f_{\min} = \min\{f_1^*, f_2^*, f_3^*\}$ (cf. [1]). The results corresponding to the *solved problems* are depicted in the performance profiles of Figures 1, 2 and 3, for the number of iterations, the number of function evaluations and the demanded CPU time, respectively. The plots on the right provide a zoomed view of the ones on the left, so that the most efficient strategy for each measure can be better visualized.

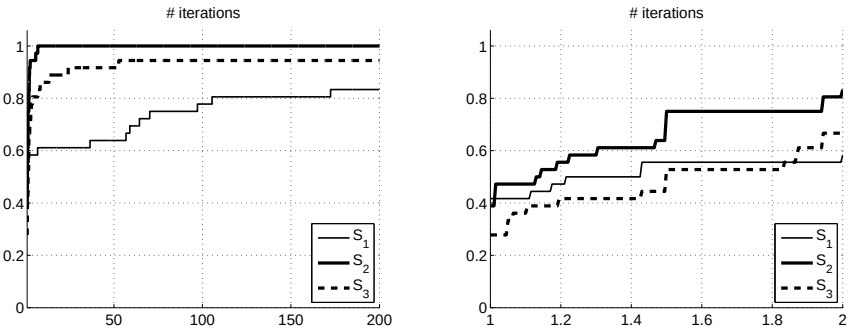

Figure 1: Performance profiles of the number of iterations.

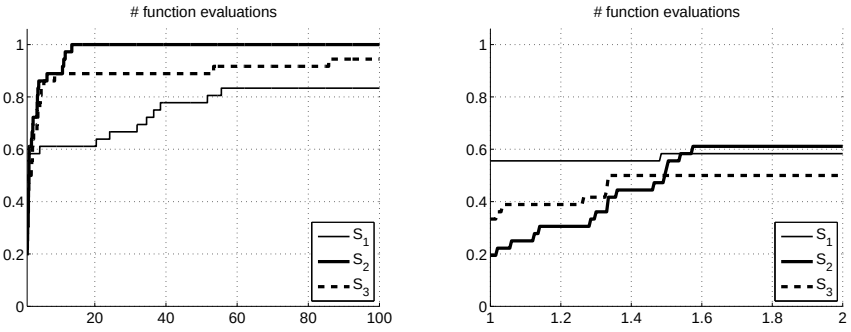

Figure 2: Performance profiles of the number of function evaluations.

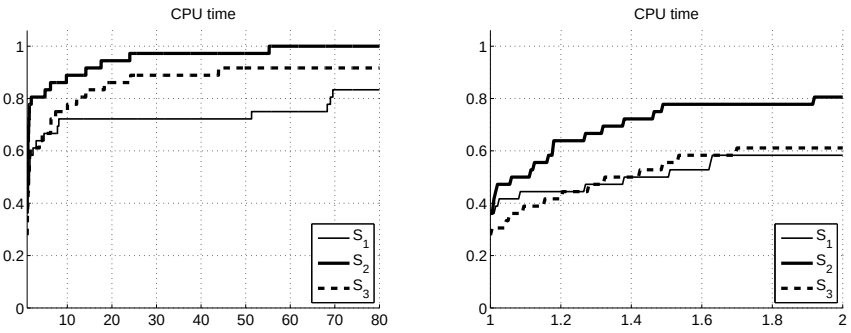

Figure 3: Performance profiles of the demanded CPU time.

The performance profiles for number of iterations given in Figure 1 show that the strategies S_1 and S_2 are competitive in terms of efficiency, whereas S_3 is slightly inferior. When it comes to robustness, strategy S_2 clearly demands much fewer iterations than S_1 and S_3 to solve the same amount of problems. Moreover, S_2 is robust, as it actually solved the whole set of 36 problems. Strategy S_1

solved 83% of the set (30 problems, the exceptions were `COSINE` with the three possible dimensions, `EIGENBLS`, `EIGENCLS` and `NONCVXUN` with $n = 200$) and strategy S_3 solved 94% (34 problems, except `COSINE` with $n = 100$ and 200).

For the measure *number of function evaluations*, shown in Figure 2, strategy S_1 is the most efficient, followed by S_3, being S_2 the least efficient for this set of problems. While S_1 gains in 55% of the problems, S_2 solves the same amount of problems spending around 1.5 times the number of function evaluations used by the best algorithm. On the other hand, S_2 is robust, solving the whole set of problems with much fewer function evaluations than the other two strategies.

In the analysis of the demanded CPU time, from Figure 3 one can see that the efficiency resembles the pattern of the performed iterations: there is a tie between strategies S_1 and S_2, and S_3 is slightly inferior. The robustness of S_2 stands out for this measure as well. From the three profiles it is also possible to infer that the computation of the smallest eigenvalue (in each iteration) by S_2 and S_3 does not burden the effort of these two strategies.

The average number of Cholesky factorizations per iteration (#Chol/#Iter) required by S_2 and S_3 (cf. Tables 2 and 3, resp.) are close to one, and rarely surpasses two, in accordance with the experimental findings of [15]. The corresponding *boxplots* [24], depicted in Figure 4, corroborate this fact, revealing that the strategy S_2 is more effective than S_3, in spite of having six outliers. Summing up, the computation of the smallest eigenvalue demanded by S_2 and S_3 seems to be tuned and worth, so that we have an alternative to the single modified Cholesky factorization employed by each iteration of `uncmin`.

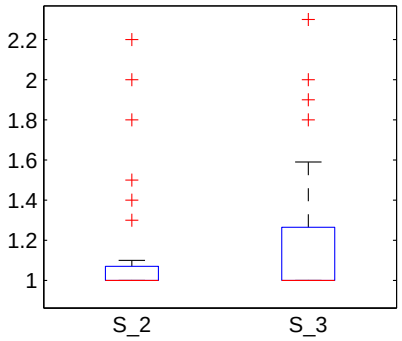

Figure 4: Boxplots for the data of columns $\frac{\#\text{Chol}}{\#\text{Iter}}$ of Tables 2 and 3.

Figure 5 shows the logarithm (base 10) of the gradient norm against iterations for both strategies S_2 and S_3 for two typical problems: a well-conditioned (left) and an ill-conditioned one (right). The quadratic rate of convergence of strategy S_2 can be visualized for both problems.

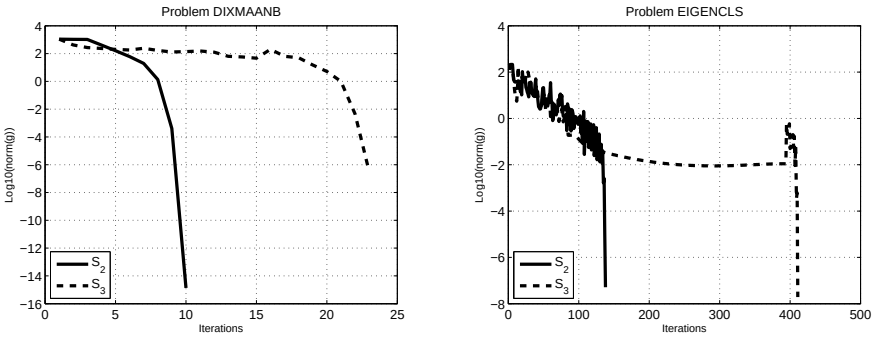

Figure 5: Gradient norm against iterations for problems `DIXMAANB` (left) and `EIGENCLS` (right).

Last but not least, considering specifically the addressed ill-conditioned problems, namely `EIGENALS`, `EIGENBLS`, `EIGENCLS`, `NCB20`, `NONDQUAR` and `SPARSINE`, we should notice that, except for `NONDQUAR`, the strategy S_1 is not as effective as the other two. In fact, the performance of S_2 stands out for `EIGENALS`, `EIGENBLS`, `EIGENCLS` and `SPARSINE` ($n = 300$), indicating the potential regularizing effect of the proposed approach.

5 Final remarks

We have proposed algebraic rules for choosing the quadratic regularizing parameter in a quasi-Newton method for smooth minimization. These rules are based on the analysis of the Taylor second-order approximation of a C^2 function with a Lipschitz continuous Hessian and rests upon two intrinsic quantities: the Hessian's Lipschitz constant and the smallest algebraic eigenvalue of this operator at the current point. We have presented an algorithmic framework for using such rules in combination with the Armijo line search. If the Lipschitz constant is available, these rules guarantee *a priori* that the full quasi-Newton step is accepted. If such a constant is not available, an estimation of it can be dynamically updated. In both cases, the proposed algorithm has quadratic convergence under the usual assumptions for Newton's method.

We have developed a `fortran` implementation of the proposed algorithmic framework taking advantage of the software package `ARPACK` for computing the smallest eigenvalue of the Hessian. The quasi-Newton linear systems were solved using a Cholesky factorization from `LAPACK`, each iteration requiring, in most cases, a single factorization. Preliminary numerical results with medium-sized problems from the `CUTEr` collection were encouraging. A thorough validation requires further numerical experiments.

Future research topics include inexact solutions of the quasi-Newton linear system using conjugate gradient methods. This would allow the use of our approach in the context of large-scale minimization problems. Extension of our scheme to least-squares problems, using its special structure, is the subject of ongoing research.

Acknowledgement. The authors are grateful to José Mario Martínez for his valuable comments and suggestions.

References

[1] E. G. Birgin and J. M. Gentil. Evaluating bound-constrained minimization software. *Comput. Optim. Appl.*, 53(2):347–373, 2012.

[2] C. Cartis, N. I. M. Gould, and P. L. Toint. Evaluation complexity of adaptive cubic regularization methods for convex unconstrained optimization. *Optim. Methods Softw.*, 27(2):197–219, 2012.

[3] C. Cartis, N. I. M. Gould, and Ph. L. Toint. Adaptive cubic regularisation methods for unconstrained optimization. Part I: motivation, convergence and numerical results. *Math. Program.*, 127(2, Ser. A):245–295, 2011.

[4] C. Cartis, N. I. M. Gould, and Ph. L. Toint. Adaptive cubic regularisation methods for unconstrained optimization. Part II: worst-case function- and derivative-evaluation complexity. *Math. Program.*, 130(2, Ser. A):295–319, 2011.

[5] C. Cartis, N. I. M. Gould, and Ph. L. Toint. An adaptive cubic regularization algorithm for nonconvex optimization with convex constraints and its function-evaluation complexity. *IMA J. Numer. Anal.*, 32(4):1662–1695, 2012.

[6] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. *Trust-region methods*. MPS/SIAM Series on Optimization. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000.

[7] J. E. Dennis, Jr. and R. B. Schnabel. *Numerical methods for unconstrained optimization and nonlinear equations*, volume 16 of *Classics in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1996. Corrected reprint of the 1983 original.

[8] R. Fletcher. *Practical methods of optimization*. A Wiley-Interscience Publication. John Wiley & Sons Ltd., Chichester, second edition, 1987.

[9] M. Fuentes, J. Malick, and C. Lemaréchal. Descentwise inexact proximal algorithms for smooth optimization. *Comput. Optim. Appl.*, 53(3):755–769, 2012.

[10] P. E. Gill, W. Murray, and M. H. Wright. *Practical optimization*. Academic Press Inc. [Harcourt Brace Jovanovich Publishers], London, 1981.

[11] S. M. Goldfeld, R. E. Quandt, and H. F. Trotter. Maximization by quadratic hill-climbing. *Econometrica*, 34:541–551, 1966.

[12] N. I. M. Gould, M. Porcelli, and Ph. L. Toint. Updating the regularization parameter in the adaptive cubic regularization algorithm. *Comput. Optim. Appl.*, 53(1):1–22, 2012.

[13] A. Griewank. The modification of Newton's method for unconstrained optimization by bounding cubic terms. Technical Report NA/12, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, 1981.

[14] W. W. Hager and H. Zhang. Self-adaptive inexact proximal point methods. *Comput. Optim. Appl.*, 39(2):161–181, 2008.

[15] M. D. Hebden. An algorithm for minimization using exact second derivatives. Technical Report T.P. 515, AERE Harwell Laboratory, Harwell, Oxfordshire, England, 1973.

[16] R. B. Lehoucq, D. C. Sorensen, and C. Yang. *ARPACK users' guide*, volume 6 of *Software, Environments, and Tools*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1998.

[17] K. Levenberg. A method for the solution of certain non-linear problems in least squares. *Quart. Appl. Math.*, 2:164–168, 1944.

[18] B. Martinet. Régularisation d'inéquations variationnelles par approximations successives. *Rev. Française Informat. Recherche Opérationnelle*, 4(Ser. R-3):154–158, 1970.

[19] J. J. Moré. Recent developments in algorithms and software for trust region methods. In *Mathematical programming: the state of the art (Bonn, 1982)*, pages 258–287. Springer, Berlin, 1983.

[20] Y. Nesterov and B. T. Polyak. Cubic regularization of Newton method and its global performance. *Math. Program.*, 108(1, Ser. A):177–205, 2006.

[21] J. Nocedal and S. J. Wright. *Numerical optimization*. Springer Series in Operations Research and Financial Engineering. Springer, New York, second edition, 2006.

[22] R. B. Schnabel, J. E. Koontz, and B. E. Weiss. A modular system of algorithms for unconstrained minimization. *ACM Trans. Math. Software*, 11(4):419–440, 1985.

[23] A. N. Tikhonov. On the stability of inverse problems. *C. R. (Doklady) Acad. Sci. URSS (N.S.)*, 39:176–179, 1943.

[24] J. W. Tukey. *Exploratory data analysis.* Behavioral Science: Quantitative Methods. 7616. Reading, Massachusetts etc.: Addison-Wesley Publishing Company. XVI, 1977.

[25] M. Weiser, P. Deuflhard, and B. Erdmann. Affine conjugate adaptive Newton methods for nonlinear elastomechanics. *Optim. Methods Softw.*, 22(3):413–431, 2007.

Appendix

The complete computational results are presented next.

Problem	n	#Iter	#Fun	f^*	$\|g^*\|$	CPU	exit
ARWHEAD	1000	6	7	+0.00E+00	1.10E−12	12.8	RS
COSINE	100	10000	10001	−2.10E+01	2.84E−05	23.1	MI
COSINE	200	10000	10001	−4.00E+01	2.40E−05	169.6	MI
COSINE	1000	10000	10001	−1.89E+02	2.87E−05	44450.1	MI
CURLY30	1000	32	35	−1.00E+05	1.40E−04	305.7	RS
DIXMAANA	900	6	7	+1.00E+00	2.70E−20	11.9	RS
DIXMAANB	900	10	12	+1.00E+00	4.80E−09	48.0	RS
DIXMAANG	900	15	16	+1.00E+00	3.39E−09	104.6	RS
DIXMAANH	900	16	17	+1.00E+00	3.41E−09	112.4	RS
DIXMAANL	900	22	24	+1.00E+00	6.67E−09	155.4	RS
DIXON3DQ	1000	1	2	+1.10E−28	8.88E−16	2.2	RS
EIGENALS$^{(*)}$	420	40	41	+3.12E−27	3.01E−13	14.0	RS
EIGENBLS$^{(*)}$	420	10000	10001	+6.39E−02	7.92E−03	8124.4	MI
EIGENCLS$^{(*)}$	462	10000	10011	+4.54E+00	7.08E−02	9117.9	MI
ENGVAL1	1000	7	8	+1.10E+03	2.40E−06	15.0	RS
FMINSRF2	961	17	56	+1.00E+00	2.65E−07	63.5	RS
FREUROTH	1000	6	9	+1.20E+05	1.53E−02	15.0	RS
GENROSE	500	2549	2742	+1.00E+00	2.43E−08	990.1	RS
MSQRTALS	1024	6996	7001	+1.87E−20	7.15E−12	46139.0	RS
MSQRTBLS	1024	6390	6396	+1.30E−15	7.83E−08	42205.0	RS
NCB20$^{(*)}$	210	7171	7180	+3.43E+02	4.16E−05	794.9	RS
NCB20$^{(*)}$	510	5191	5196	+7.11E+02	2.77E−05	13709.1	RS
NONCVXUN	100	2443	2454	+2.33E+02	1.79E−05	5.9	RS
NONCVXUN	200	10000	10009	+4.76E+02	1.64E−03	173.8	MI
NONCVXUN	1000	10000	10015	+2.33E+03	2.00E−03	44289.4	MI
NONDQUAR$^{(*)}$	500	18	19	+1.04E−10	6.17E−07	3.6	RS
NONDQUAR$^{(*)}$	1000	19	20	+4.13E−11	3.66E−07	45.5	RS
PENALTY1	1000	40	43	+9.69E−09	9.73E−07	111.2	RS
SCHMVETT	500	2	3	−1.49E+03	1.33E−04	0.6	RS
SCHMVETT	1000	2	3	−2.99E+03	1.33E−04	6.5	RS
SINQUAD	1000	10	19	−2.94E+05	2.98E−06	41.8	RS
SPARSINE$^{(*)}$	200	2061	2062	+3.15E−11	1.98E−06	36.6	SI
SPARSINE$^{(*)}$	250	47	48	+3.19E−15	1.76E−08	2.6	RS
SPARSINE$^{(*)}$	300	10000	10001	+8.79E−04	2.03E−02	600.6	MI
TOINTGSS	1000	1	2	+1.00E+01	3.64E−15	2.2	RS
WOODS	1000	57	90	+2.79E−13	4.37E−07	125.3	RS

Table 1: Numerical results of strategy S_1 (code `uncmin`). Problems marked with $^{(*)}$ are ill-conditioned.

Problem	n	#Iter	#Fun	$\frac{\#\text{Chol}}{\#\text{Iter}}$	f^*	$\|g^*\|$	CPU	exit
ARWHEAD	1000	6	7	1	+0.00E+00	1.15E−12	11.9	AS
COSINE	100	16	48	1.5	−9.90E+01	1.54E−08	19.9	AS
COSINE	200	34	346	1.1	−1.98E+02	3.66E−07	40.3	AS
COSINE	1000	22	140	2.2	−9.99E+02	4.05E−03	59.7	SI
CURLY30	1000	16	133	1.3	−1.00E+05	3.46E−11	64.1	AS
DIXMAANA	900	9	46	1	+1.00E+00	1.68E−14	15.1	AS
DIXMAANB	900	9	32	2	+1.00E+00	1.37E−15	29.5	AS
DIXMAANG	900	35	187	1	+1.00E+00	4.96E−08	95.4	AS
DIXMAANH	900	35	230	1.02	+1.00E+00	2.20E−07	98.7	AS
DIXMAANL	900	47	262	1	+1.00E+00	5.74E−07	153.8	AS
DIXON3DQ	1000	7	8	1	+2.24E−08	2.10E−08	30.5	AS
EIGENALS$^{(*)}$	420	232	462	1.8	+1.83E−21	2.48E−10	774.3	AS
EIGENBLS$^{(*)}$	420	155	277	1	+1.97E−15	5.15E−08	545.9	AS
EIGENCLS$^{(*)}$	462	137	426	1	+2.21E−17	5.12E−08	388.5	AS
ENGVAL1	1000	8	9	1	+1.11E+03	1.32E−12	15.9	AS
FMINSRF2	961	33	57	1	+1.00E+00	3.77E−08	83.6	AS
FREUROTH	1000	9	35	1.4	+1.21E+05	1.84E−08	22.3	AS
GENROSE	500	375	777	1	+1.00E+00	1.87E−09	437.8	AS
MSQRTALS	1024	73	198	1.1	+9.41E−17	5.88E−09	663.4	AS
MSQRTBLS	1024	92	227	1.04	+3.91E−15	4.58E−09	822.3	AS
NCB20$^{(*)}$	210	68	209	1	+3.45E+02	2.43E−07	98.3	AS
NCB20$^{(*)}$	510	88	238	1.1	+7.15E+02	3.36E−08	231.4	AS
NONCVXUN	100	56	106	1	+2.34E+02	5.29E−08	58.1	AS
NONCVXUN	200	83	145	1	+4.67E+02	2.60E−08	87.3	AS
NONCVXUN	1000	309	470	1	+2.33E+03	5.25E−08	727.0	AS
NONDQUAR$^{(*)}$	500	35	49	1	+8.15E−08	4.60E−07	86.9	AS
NONDQUAR$^{(*)}$	1000	51	75	1	+1.17E−07	7.08E−07	228.0	AS
PENALTY1	1000	28	29	1	+9.73E−03	2.46E−04	69.5	RS
SCHMVETT	500	3	4	1	−1.49E+03	1.24E−11	3.8	AS
SCHMVETT	1000	3	4	1	−2.99E+03	1.77E−11	6.4	AS
SINQUAD	1000	7	45	1	−2.94E+05	1.94E−08	13.9	AS
SPARSINE$^{(*)}$	200	47	223	1	+2.33E−19	1.41E−08	53.5	AS
SPARSINE$^{(*)}$	250	40	204	1	+9.41E−18	8.29E−09	46.5	AS
SPARSINE$^{(*)}$	300	58	260	1	+3.69E−15	4.06E−08	77.1	AS
TOINTGSS	1000	2	3	1	+1.00E+01	3.81E−08	4.1	AS
WOODS	1000	47	95	1	+1.03E−18	1.26E−09	90.9	AS

Table 2: Numerical results of strategy S_2 (Algorithm 1 with $\mu_k \equiv \mu_k^-$). Problems marked with $^{(*)}$ are ill-conditioned.

Problem	n	#Iter	#Fun	$\frac{\#\text{Chol}}{\#\text{Iter}}$	f^*	$\|g^*\|$	CPU	exit
ARWHEAD	1000	6	7	1	+0.00E+00	1.34E−12	11.8	AS
COSINE	100	9318	10000	1.9	−6.49E+01	3.20E−04	7123.7	MF
COSINE	200	8222	10000	1	−1.61E+02	9.10E+00	9409.0	MF
COSINE	1000	18	91	2.3	−9.97E+02	1.49E−08	50.7	AS
CURLY30	1000	17	80	1.3	11.00E+05	2.68E−10	69.9	AS
DIXMAANA	900	9	15	1	+1.00E+00	2.48E−13	15.2	AS
DIXMAANB	900	22	45	1	+1.00E+00	3.37E−07	45.3	AS
DIXMAANG	900	28	45	1	+1.00E+00	1.73E−08	69.3	AS
DIXMAANH	900	30	77	1.2	+1.00E+00	6.85E−10	88.8	AS
DIXMAANL	900	64	209	1.6	+1.00E+00	3.47E−11	261.0	AS
DIXON3DQ	1000	7	8	1	+2.37E−08	2.10E−08	30.4	AS
EIGENALS$^{(*)}$	420	2109	2189	1	+1.04E−20	5.85E−10	7495.7	AS
EIGENBLS$^{(*)}$	420	1287	1360	1.9	+1.03E−15	8.08E−09	5479.5	AS
EIGENCLS$^{(*)}$	462	410	565	1.8	+1.15E−15	1.93E−08	1695.0	AS
ENGVAL1	1000	93	686	1	+1.11E+03	1.92E−06	181.9	SI
FMINSRF2	961	33	57	1	+1.00E+00	3.77E−08	84.0	AS
FREUROTH	1000	11	35	1	+1.00E+00	3.22E−08	22.3	AS
GENROSE	500	371	606	1	+1.00E+00	2.89E−07	431.2	AS
MSQRTALS	1024	72	126	1.12	+2.64E−16	4.64E−09	692.8	AS
MSQRTBLS	1024	91	175	1.4	+6.52E−15	4.43E−08	947.0	AS
NCB20$^{(*)}$	210	75	139	1	+3.44E+02	1.15E−07	103.6	AS
NCB20$^{(*)}$	510	92	163	1.1	+7.12E+02	4.93E−08	198.7	AS
NONCVXUN	100	43	71	1	+2.35E+02	1.07E−07	43.8	AS
NONCVXUN	200	70	107	1	+4.67E+02	7.76E−09	74.1	AS
NONCVXUN	1000	274	414	1	+2.32E+03	7.04E−09	648.5	AS
NONDQUAR$^{(*)}$	500	35	49	1	+8.15E−08	4.60E−07	87.5	AS
NONDQUAR$^{(*)}$	1000	64	94	1	+6.64E−08	7.71E−07	288.0	AS
PENALTY1	1000	28	29	1	+9.73E−03	2.46E−04	68.5	RS
SCHMVETT	500	3	4	1	−1.49E+03	4.19E−09	3.8	AS
SCHMVETT	1000	3	4	1	−2.99E+03	4.49E−09	6.3	AS
SINQUAD	1000	10	24	1	−2.94E+05	1.37E−09	19.8	AS
SPARSINE$^{(*)}$	200	32	101	1	+3.02E−18	2.76E−09	36.7	AS
SPARSINE$^{(*)}$	250	42	118	1	+8.28E−17	1.82E−09	48.4	AS
SPARSINE$^{(*)}$	300	1401	1479	2.0	+4.12E−12	1.72E−07	3384.1	AS
TOINTGSS	1000	4	5	1	+1.00E+01	1.51E−12	8.1	AS
WOODS	1000	56	93	1	+4.08E−17	7.14E−10	109.5	AS

Table 3: Numerical results of strategy S_3 (Algorithm 1 with $\mu_k \equiv \mu_k^+$). Problems marked with $^{(*)}$ are ill-conditioned.