CUBIC REGULARIZATION METHOD BASED ON MIXED FACTORIZATIONS FOR UNCONSTRAINED MINIMIZATION*

E. G. BIRGIN† AND J. M. MARTÍNEZ‡

Abstract. Newton's method for unconstrained optimization, subject to proper regularization or special trust-region procedures, finds first-order stationary points with precision ε employing, at most, $O(\varepsilon^{-3/2})$ functional and derivative evaluations. However, the computer work per iteration of the best-known implementations may need several factorizations per iteration or may use rather expensive matrix decompositions. In this paper, we introduce a method that, preserving most features of the regularization approach, uses only one cheap factorization per iteration, as well as the same number of gradient and Hessian evaluations. We prove complexity and convergence results, even in the case in which the Hessians of the subproblems are far from being Hessians of the objective function. We also present fairly successful and fully reproducible numerical experiments and we make available the corresponding software.

Key words. Smooth unconstrained minimization, Bunch-Parlett-Kaufman factorizations, regularization, Newton-type methods.

AMS subject classifications. 90C30, 65K05, 49M37, 90C60, 68Q25.

1. Introduction.

The unconstrained optimization problem, given by

$$\text{Minimize} f(x) \text{ subject to } x \in \mathbb{R}^n,$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a continuous function, is a classical problem of numerical mathematics. In recent years, the interest in the development of new efficient algorithms was enhanced by applications in machine learning, statistical learning, and big-data analysis. See, for example, [18, 20, 29, 36].

Following [35], many Newton-like algorithms were developed in the last few years for which worst-case evaluation complexity proofs have been given. Newton-like algorithms based on regularization [6, 7, 8, 9, 10, 19, 27, 33, 35] or non-standard trust regions [14, 31] enjoy worst-case evaluation complexity $O(\varepsilon^{-3/2})$, which means that the number of functional evaluations that are necessary to obtain a gradient with norm smaller than ε is bounded above by a constant times $\varepsilon^{-3/2}$. Extensions in which Hölder, instead of Lipschitz, conditions are assumed were given in [26, 11, 32].

In 2017, the papers [33], [2, 3], and [36] introduced methods in which the number of factorizations is equal to the number of (successful) iterations. Both [3] and [36] rely on line searches, although special iterations need the computation of the leftmost eigenvalue of the current Hessian. In [33] the spectral QDQ^T factorization is used to mimic the Levenberg-Marquardt path and special terms to approximate third-order derivatives are employed.

Line-search methods are, of course, very attractive. In general, the line search follows the Newton direction $\{-t\nabla^2 f(x)^{-1}\nabla f(x), 0 \le t \le 1\}$, trying firstly $t \approx 1$. When the unitary step is admissible quadratic convergence is generally obtained [16] and, in practice, convergence is very fast. Moreover, at each iteration (or "successful

*Submitted to the editors February 6, 2018.

Funding: Supported by FAPESP (grants 2013/05475-7, 2013/07375-0, and 2016/01860-1) and CNPq (grants 309517/2014-1 and 303750/2014-6).

†Dept. of Computer Science, Institute of Mathematics and Statistics, University of São Paulo, Rua do Matão, 1010, Cidade Universitária, 05508-090, São Paulo, SP, Brazil. egbirgin@ime.usp.br

‡Dept. of Applied Mathematics, Institute of Mathematics, Statistics, and Scientific Computing, State University of Campinas, 13083-859, Campinas, SP, Brazil. martinez@ime.unicamp.br

iteration" using the trust-region terminology [12]) only one Hessian factorization is needed whereas several functional evaluations may be necessary to obtain sufficient decrease. However, when the steplength t becomes small, the advantages of the Newton direction tend to disappear because the Newton direction is not optimal under a small norm constraint. For example, a very ill-conditioned Hessian obviously affects the accuracy of the unitary Newton step, and, in the line search approach, such inaccuracy remains active all along the backtracking procedure since, roughly speaking, the whole direction (not only its size) may be "wrong".

This fact motivated the introduction of trust-region and regularization methods [12, 21, 30, 37], which preserve Newton steps when they are acceptable in terms of functional reduction, but rely on (close to) steepest descent steps for obtaining trial points near the current iterate. Unfortunately, in the classical trust-region approach one needs more than one Cholesky decomposition for computing each trial point [34]. This is the cost of solving a "trust-region subproblem" before each functional evaluation. A similar cost is involved in the solution of cubic regularization subproblems employing an algorithm introduced by Cartis, Gould, and Toint [9, 10, 25].

In this paper, we aim to introduce an efficient method with the trust-region (or regularization) flavor in the sense that when the "backtracking" process is activated (that is, when the regularization parameter is increased), not only the trial point becomes closer to the current point, but also the direction becomes closer to a conservative gradient-like direction. More precisely, the goal of the present work is to introduce a method for unconstrained minimization that (a) employs only cheap (Cholesky-like) factorizations, with no eigenvalue calculation of the whole Hessian approximation at all; (b) employs only one factorization per iteration; (c) enjoys $O(\varepsilon^{-3/2})$ evaluation complexity for first-order optimality and $O(\varepsilon^{-3})$ for second-order optimality when the Hessian is Lipschitz-continuous; and (d) enjoys convergence to first-order optimality and admits a suitable complexity analysis in the case that neither Lipschitz nor Hölder conditions hold for gradients or Hessians.

For these purposes, we introduce Mixed Factorizations based on Bunch-Parlett-Kaufman decompositions in order to define a new algorithm in which full eigenvalue decompositions are not used and the number of factorizations, whose difficulty is similar to Cholesky decompositions, is equal to the number of (successful) iterations. Cubic scaled regularizations allow us to prove $O(\varepsilon^{-3/2})$ complexity using Lipschitz-continuity of the Hessian, as expected. Moreover, in the present research we addressed the problem of proving convergence and complexity for the situation in which the Hessians of the quadratic models are not accurate Hessian approximations, without Lipschitz (or even Hölder) assumptions on first- or second-order derivatives. For this case, we prove that gradients as small as desired are obtained with complexity bounds that, as expected, are worse than the ones that may be obtained with Lipschitz or Hölder assumptions but indicate that efficient practical implementations may be obtained. We have in mind the feasibility problems that arise in constrained optimization, in which the Hessian of the objective function is discontinuous on the boundary of the feasible region.

The rest of this paper is organized as follows. In Section 2, we define Mixed Factorizations, we describe the Mixed Factorization based on Bunch-Parlett-Kaufman (BPK) decompositions, and we introduce the unconstrained minimization algorithm which is the main subject of the paper. The definition of the algorithm refers to an arbitrary Mixed Factorization, although in practical terms we focus on the one based on BPK. At each iteration, the introduced algorithm solves a cubic regularized subproblem, the solution of which is described in Section 3. BPK decompositions were used by Gould and Nocedal to define a method with a variable trust region norm which employs only one factorization per iteration, as the method presented in the present paper. See [23] and [12, §7.7]. In Section 4, we begin proving convergence in the sense that $\liminf \|\nabla f(x^k)\| = 0$ for the sequence generated by the minimization algorithm using only differentiability of the objective function. Existence and uniform continuity of first-order derivatives is assumed but not Hölder conditions. The convergence proof involves complexity arguments, in the sense that it is preceded by the proof that given an arbitrary $\varepsilon > 0$, the number of consecutive iterations at which $\|\nabla f(x^k)\|_\infty \geq \varepsilon$ cannot exceed a quantity N_ε that depends on ε, besides algorithm constants and characteristics of the problem. Obviously, N_ε does not have the form $c\varepsilon^{-q}$, as in the case in which Lipschitz or Hölder assumptions are made. We finish the section proving first-order complexity $O(\varepsilon^{-3/2})$ and second-order complexity $O(\varepsilon^{-3})$ in the case that Lipschitz-continuity of the Hessian is assumed. In Section 5, we report experiments involving all the unconstrained problems of the CUTEst collection [24]. In order to guarantee reproducibility of the results, the codes that implement the algorithms and these experiments are available in http://www.ime.usp.br/~egbirgin/. In Section 6 we derive conclusions and we sketch lines for future research.

Notation. $g(x)$ denotes the gradient of f at x. We say that a matrix $Q \in \mathbb{R}^{n\times n}$ is orthonormal if QQ^T is the Identity. The i-th component of a vector v is denoted v_i or $[v]_i$. If $v \in \mathbb{R}^n$, we denote $|v|$ the vector whose components are $|v_1|, \dots |v_n|$. $\|\cdot\|$ denotes the Euclidean norm.

2. Mixed factorizations and minimization algorithm.

Let $H \in \mathbb{R}^{n\times n}$ be a symmetric matrix. If $H = MDM^T$, where M is non-singular and D is diagonal, we say that MDM^T is a Mixed Factorization of H. In practice, we expect that systems of the form $Mv = b$ or $M^T v = b$ should be easy to solve, say, involving at most $O(n^2)$ flops. The case in which MDM^T is the spectral decomposition of H, being M orthonormal, was considered in [6, 33] in the context of cubic regularization methods for unconstrained minimization. In this paper, we are mainly interested in the Mixed Factorization described below, which is based in the Bunch-Parlett-Kaufman decomposition.

Given a symmetric matrix H, we denote by $P_{\rm bpk} L_{\rm bpk} D_{\rm bpk} L_{\rm bpk}^T P_{\rm bpk}^T$ its Bunch-Parlett-Kaufman factorization [22]. Then, $P_{\rm bpk}$ is a permutation, $L_{\rm bpk}$ is lower-triangular with unitary diagonal, and $D_{\rm bpk}$ is a block-diagonal matrix with 1×1 and 2×2 blocks called $[D_{\rm bpk}]_1, [D_{\rm bpk}]_2, \dots$ here. For each 2×2 block $[D_{\rm bpk}]_i$, we compute Q_i and $D_i \in \mathbb{R}^{2\times 2}$, where Q_i is orthonormal, D_i is diagonal, and $[D_{\rm bpk}]_i = Q_i D_i Q_i^T$. Therefore, we may write

$$H = MDM^T,$$

where M is the product of $P_{\rm bpk} L_{\rm bpk}$ times a finite number of orthonormal matrices (as many as 2×2 blocks in $D_{\rm bpk}$) that affect only two rows of D, and D is diagonal. The matrix M does not need to be explicitly computed, being only necessary to store the permutation $P_{\rm bpk}$, the lower-triangular matrix $L_{\rm bpk}$, and the 2×2 orthonormal matrices Q_i used to diagonalize the 2×2 blocks $[D_{\rm bpk}]_i$ of $D_{\rm bpk}$.

It is worth noting that, since all the 2×2 orthonormal matrices Q_i can be stored together in a single n-dimensional array, if the strict lower-triangle of H is overwritten with $L_{\rm bpk}$ and the 2×2 orthonormal matrices are saved in the diagonal of H then only two additional n-dimensional arrays (one for the permutation matrix $P_{\rm bpk}$ and another for the diagonal matrix D) are needed to store the whole BPK-based MDM^T Mixed Factorization. The computation of this BPK-based Mixed Factorization of H involves, in the dense case, $n^3/6+O(n^2)$ sums and products, whereas the computation of the spectral QDQ^T factorization involves $(2/3)n^3 + O(n^2)$ operations. Therefore, computing this BPK-based Mixed Factorization is, neglecting the $O(n^2)$ terms, four times less expensive than computing the spectral QDQ^T factorization when the matrix H is dense. Moreover, the BPK-based Mixed Factorization can also explore the sparsity of H.

In the current section as well as in Sections 3 and 4, MDM^T denotes an arbitrary Mixed Factorization of a symmetric matrix H. The BPK-based Mixed Factorization described above and the spectral QDQ^T factorization used in [6] and [33] are particular cases that will be considered in the numerical experiments.

Algorithm 2.1. Let $\alpha > 0$, $x^0 \in \mathbb{R}^n$, $\kappa \geq 2$, and $\sigma_{\text{bles}} \geq \sigma_{\min} > 0$ be given. Set $\sigma \leftarrow 0$ and $\sigma_{\text{big}} \leftarrow \sigma_{\text{bles}}$.

Step 1. Compute a symmetric $n \times n$ matrix H_k and its Mixed Factorization $M_k D_k M_k^T$.

Step 2. Consider the problem

$$\text{Minimize } g(x^k)^T s + \frac{1}{2} s^T H_k s + \sigma \|M_k^T s\|_3^3. \tag{1}$$

Step 3. If (1) has no solution (so $\sigma = 0$), choose

$$\sigma_{\text{new}} \in [\sigma_{\min}, \sigma_{\text{big}}], \tag{2}$$

update $\sigma \leftarrow \sigma_{\text{new}}$, and go to Step 2. Otherwise, let $s_{\text{trial}} \in \mathbb{R}^n$ be a solution to (1).

Step 4. Test the sufficient descent condition

$$f(x^k + s_{\text{trial}}) \leq f(x^k) - \alpha \|M_k^T s_{\text{trial}}\|_\infty^3. \tag{3}$$

If (3) is fulfilled, define $s^k = s_{\text{trial}}$, $x^{k+1} = x^k + s^k$, $\sigma_k = \sigma$, update $k \leftarrow k+1$ and $\sigma_{\text{big}} \leftarrow \max\{\sigma_{\text{big}}, \sigma_k\}$, set $\sigma \leftarrow 0$, and go to Step 1. Otherwise, define

$$\sigma_{\text{new}} \in \begin{cases} [\sigma_{\min}, \sigma_{\text{big}}], & \text{if } \sigma = 0, \\ [2\sigma, \kappa\sigma], & \text{if } \sigma > 0. \end{cases} \tag{4}$$

update $\sigma \leftarrow \sigma_{\text{new}}$, and go to Step 2.

Algorithm 2.1 computes a single Mixed Factorization per iteration. Therefore, if one uses the BPK-based Mixed Factorization, the linear-algebra work per iteration is the same as the one of line-search implementations of Newton's method based on modified Cholesky factorizations. In the dense case, variable-norm cubic regularization methods [33] use $(2/3)n^3 + O(n^2)$ operations per iteration. Classical trust-region methods [12], and the best-known cubic regularization methods use $O(n^3/6)$ operations per function evaluation (*at least* one Cholesky factorization per function evaluation) and a variable number of functional evaluations per iteration. Note that we adopt here the criterion of calling "iteration" to the whole process that leads from x^k to a trial point at which the functional value decreases satisfying (3), differently from the traditional trust-region terminology for which each function evaluation counts as an iteration and iterations are classified in successful and unsuccessful [12]. In [23], computing M_k as in the BPK-based Mixed Factorization, the subproblem solved for obtaining each trial approximation is

$$\text{Minimize } g(x^k)^T s + \frac{1}{2} s^T H_k s \text{ subject to } s^T M_k |D_k| M_k^T s \leq \Delta, \tag{5}$$

where $|D_k|$ is the diagonal matrix whose entries are the moduli of the entries of D_k. Therefore, in the convex case the constraint of (5) reduces to $s^T H_k s \leq \Delta$. As a consequence, the method based on (5) is a line-search method when H_k is positive definite.

Using the computational environment described in Section 5, when H is dense, the computer time necessary to compute the BPK-based Mixed Factorization is $\approx 10^{-10} n^3$ seconds whereas the time necessary to compute the spectral factorization is about 15 times bigger if n is greater than ≈ 7. This factor reduces as far as n decreases and, when $n = 2$ both times are essentially identical.

3. Solving the subproblem. In this section we consider the solution of subproblem (1) at Step 2 of Algorithm 2.1. Writing

$$y = M_k^T s \quad \text{and} \quad \underline{g} = M_k^{-1} g(x^k), \tag{6}$$

subproblem (1) is equivalent to

$$\text{Minimize } \underline{g}^T y + \frac{1}{2} y^T D_k y + \sigma \sum_{i=1}^{n} |y_i|^3. \tag{7}$$

Problem (7) is entirely separable and its solution may be obtained trivially, solving one-dimensional quadratic equations.

Consider first the case $\sigma = 0$. In this case, problem (7) consists of minimizing $\underline{g}^T y + \frac{1}{2} y^T D_k y$. If some entry of D_k is negative, this problem has no solutions. If some entry is zero and the corresponding entry of $\underline{g}$ is not zero, the problem is unsolvable too. So, the problem is solvable only when all the entries of D_k are non-negative and, for each null entry $(D_k)_{ii}$, the corresponding $\underline{g}_i$ is null too. In the latter case, there are infinitely many values of y_i that solve the i-th one-dimensional sub-subproblem and we may choose $y_i = 0$ in the algorithm. When $(D_k)_{ii}$ is positive the i-th one-dimensional sub-subproblem has only one solution, given by $y_i = -\underline{g}_i/(D_k)_{ii}$.

Now consider the case in which $\sigma > 0$. Clearly, problem (7) can be decomposed into n different one-dimensional problems with unknowns $y_1, \ldots, y_n$. In order to simplify the notation, let us write $y = y_i$, $\underline{g} = \underline{g}_i$, and $d = (D_k)_{ii}$, so that each one-dimensional subproblem has the form

$$\text{Minimize } \underline{g} y + d y^2/2 + \sigma |y|^3. \tag{8}$$

If $\underline{g} \leq 0$ and $y < 0$ we clearly have that the objetive function of (8), evaluated at y is not smaller than the same objective function evaluated at $-y$. Therefore, when $\underline{g} \leq 0$, (8) admits a non-negative minimizer. Therefore, if $\underline{g} \leq 0$, problem (8) is equivalent to

$$\text{Minimize } \underline{g} y + d y^2/2 + \sigma y^3 \text{ subject to } y \geq 0. \tag{9}$$

A solution to (9) can be obtained annihilating the derivative with respect to y, i.e. solving the equation

$$3\sigma y^2 + dy + \underline{g} = 0,$$

and considering the non-negative root given by

$$y = \frac{\sqrt{d^2 - 12\sigma \underline{g}} - d}{6\sigma} = \frac{\sqrt{d^2/\sigma - 12\underline{g}}}{6\sqrt{\sigma}} - \frac{d}{6\sigma}. \tag{10}$$

Note that, in (10), y decreases as a function of d. Analogously, if $\underline{g} \geq 0$, the minimizer of (8) must be non-positive, so that the problem becomes

$$\text{Minimize } \underline{g}y + dy^2/2 - \sigma y^3 \text{ subject to } y \leq 0, \tag{11}$$

whose solution is given by the non-positive root of its objective function derivative, i.e.

$$y = -\frac{\sqrt{d^2 + 12\sigma\underline{g}} - d}{6\sigma} = -\frac{\sqrt{d^2/\sigma + 12\underline{g}}}{6\sqrt{\sigma}} + \frac{d}{6\sigma}. \tag{12}$$

Analogously to (10), in this case y increases as a function of d. Note that, if $\underline{g} = 0$, problem (8) is symmetric and, therefore, we can choose between the non-negative solution (10) and the non-positive solution (12), that in this case reduces to $y = 0$ if $d \geq 0$ and to $y = \mp d/(3\sigma)$ if $d < 0$.

Summing up, regarding (8), we have that (i) if $\sigma = d = \underline{g} = 0$ then its objective function is the null function and any $y \in \mathbb{R}$ is a solution; (ii) if $\sigma = 0$ and $d > 0$, its solution is given by $y = -\underline{g}/d$; and (iii) if $\sigma > 0$ then its solution is given by

$$y = \text{sg}(\underline{g}) \left[-\frac{\sqrt{d^2 + 12\sigma|\underline{g}|} - d}{6\sigma} \right] = \text{sg}(\underline{g}) \left[-\frac{\sqrt{d^2/\sigma + 12|\underline{g}|}}{6\sqrt{\sigma}} + \frac{d}{6\sigma} \right], \tag{13}$$

where $\text{sg}(a) \in \{-1, 1\}$ represents the signal of $a \in \mathbb{R}$, and $|y|$ decreases as a function of d. In the remaining cases (iv) $\sigma = 0$ and $d < 0$ and (v) $\sigma = 0$, $d = 0$, and $\underline{g} \neq 0$, the problem has no solution. The relation (13) shows that if σ is big or $d = 0$ then $y \approx O(1/\sqrt{\sigma})$; while, when σ is small, if $d < 0$ then $y \approx O(1/\sigma)$ and if $d > 0$ then y approaches the solution to (8).

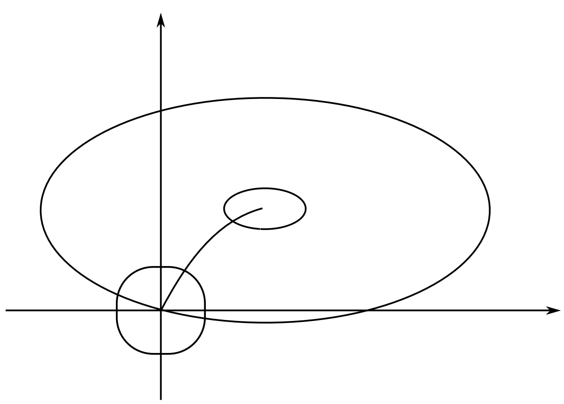

FIG. 1. *Solutions to (7) for $\sigma \geq 0$.*

Figure 1 represents the solutions to (7) for $\sigma \geq 0$ in a problem where D_k is positive definite and $n = 2$. The ellipses are level sets of the quadratic objective function $\underline{g}^T y + \frac{1}{2} y^T D_k y$, with $\underline{g} = (-12.5, -50)^T$ and $D_k = \text{diag}(12.5, 50)$. The minimizer of the quadratic (center of the ellipses) is $(1, 1)^T$ and the convex region centered at the origin represents the ball $\|y\|_3 \leq 0.43$. The curve that joins the origin with the center of the ellipses is the set of solutions to (7) for $\sigma \geq 0$. According to (13), the points of this curve with abscissas 1, 0.5, 0.25, 0.1, 0.01, and 0 correspond to $\sigma = 0$, 8.33, 50, 375, 41250, and ∞, respectively. The tangent vector to the curve of solutions to (7) is proportional to $(\mathrm{sg}(\underline{g}_1)|\underline{g}_1|^{1/2}, \ldots \mathrm{sg}(\underline{g}_n)|\underline{g}_n|^{1/2})^T$, being $(3.54, 7.07)^T$ in Figure 1. This is the steepest descent direction associated with the norm $\|\cdot\|_3$. Comparing this steepest descent direction with the one corresponding to $\|\cdot\|_2$ we see that the components smaller than 1 (in modulus) are increased whereas the components bigger than 1 are decreased. Moreover, the moduli of all the components of the steepest direction with respect to $\|\cdot\|_p$ tend to 1 as p tends to infinity. In some sense this feature reveals a tendency to independence with respect to scaling of variables.

4. Convergence and complexity.

Algorithm 2.1 was conceived regarding economy of calculations in Newton-like methods with regularization that exhibit worst-case complexity $O(\varepsilon^{-3/2})$. In the usual approach, the matrix H_k is the Hessian of the objective function f or a close approximation in the sense of [9] and [10]. However, for many reasons, we may wish to employ different quadratic models for which no guarantee of closeness to the true Hessian is guaranteed. Sometimes, the Hessian of the objective function does not exist or it is discontinuous on some regions of $\mathbb{R}^n$. This is the case of the objective functions in the subproblems that arise in some Penalty and Augmented Lagrangian methods for constrained optimization. See [5]. It is natural, therefore, to ask for the convergence and, perhaps, complexity properties of the algorithm in those cases. This is the first question addressed in this section. We are going to prove that the decrease of the objective function at each iteration at which the gradient norm is greater than ε is bigger than a strictly positive quantity that only depends on ε, besides characteristics of the problem and parameters of the algorithm. For proving this result we will not rely on Lipschitz or Hölder conditions, neither for the Hessian, nor for the gradient. Of course, the complexity result that arises in this case is not of the type $O(\varepsilon^{-q})$ since the assumptions used here are far weaker than the ones that are necessary for proving strong complexity theorems. However, they are useful to show that, in spite of big inaccuracies of Hessian computations, or even disregarding such computations at all, we still maintain the essential theoretical properties that are inherent in unconstrained minimization algorithms.

Proposition 4.1 is a technical result that relates the size of a computed trial increment $s_k(\sigma)$ to the gradient at x^k and the matrices M_k and D_k.

PROPOSITION 4.1. *For all $\sigma > 0$, let $s_k(\sigma)$ be a solution to (1). Then $s_k(\sigma)$ is a solution to*

$$\text{Minimize } g(x^k)^T s + \frac{1}{2} s^T H_k s \text{ subject to } \|M_k^T s\|_3 \le \|M_k^T s_k(\sigma)\|_3. \tag{14}$$

If $\sigma > 0$, and $M_k D_k M^T$ is a Mixed Factorization of H_k,

$$\|M_k^T s_k(\sigma)\|_\infty \le \frac{\sqrt{\|D_k\|^2/\sigma + 12\|M_k^{-1}\|\|g(x^k)\|}}{6\sqrt{\sigma}} + \frac{\|D_k\|}{6\sigma}, \tag{15}$$

$$\|s_k(\sigma)\|_\infty \le \|M_k^{-T}\|_\infty \left[\frac{\sqrt{\|D_k\|^2/\sigma + 12\|M_k^{-1}\|\|g(x^k)\|}}{6\sqrt{\sigma}} + \frac{\|D_k\|}{6\sigma} \right], \tag{16}$$

and

$$\lim_{\sigma\to\infty} \|s_k(\sigma)\| = 0. \tag{17}$$

Moreover,

$$\|M_k^T s_k(\sigma)\| \geq \|M_k^T s_k(\sigma)\|_3 \geq \|M_k^T s_k(\sigma)\|_\infty \geq \frac{\sqrt{\|D_k\|^2+12\sigma\|g(x^k)\|_\infty/\|M_k\|_\infty}-\|D_k\|}{6\sigma} \tag{18}$$

and

$$\|s_k(\sigma)\| \geq \frac{1}{\|M_k^T\|}\frac{\sqrt{\|D_k\|^2+12\sigma\|g(x^k)\|_\infty/\|M_k\|_\infty}-\|D_k\|}{6\sigma}. \tag{19}$$

Finally, for all i such that $(D_k)_{ii} < 0$,

$$\|M_k^T s_k(\sigma)\| \geq \|M_k^T s_k(\sigma)\|_3 \geq \|M_k^T s_k(\sigma)\|_\infty \geq \frac{|(D_k)_{ii}|}{3\sigma} \tag{20}$$

and

$$\|s_k(\sigma)\| \geq \frac{|(D_k)_{ii}|}{3\|M_k^T\|\sigma}. \tag{21}$$

Proof: Assume that s is a feasible point of (14). By the definition of $s_k(\sigma)$, we have that

$$g(x^k)^T s_k(\sigma) + \frac{1}{2}s_k(\sigma)^T H_k s_k(\sigma) + \sigma\|M_k^T s_k(\sigma)\|_3^3 \leq g(x^k)^T s + \frac{1}{2}s^T H_k s + \sigma\|M_k^T s\|_3^3.$$

Then, (14) follows from $\|M_k^T s\|_3^3 \leq \|M_k^T s_k(\sigma)\|_3^3$. The inequality (15) follows from (13) with $y = M_k^T s_k(\sigma)$, (16) follows from (15), and (17) is a consequence of (16).

Since $\sigma > 0$, from (13), we deduce that $(D_k)_{ii} < 0$ implies that

$$|y_i| \geq \frac{|(D_k)_{ii}|}{3\sigma}.$$

This implies (20) and (21). If $\|g(x^k)\| = 0$, (18) and (19) follow trivially. In any case, since $|y|$ decreases as a function of d, by (13), we have that

$$|[M_k^T s_k(\sigma)]_i| \geq \frac{\sqrt{\|D_k\|^2+12\sigma|[M_k^{-1}g(x^k)]_i|}-\|D_k\|}{6\sigma}$$

for all $i = 1,\ldots,n$; and, applying this inequality to the component that maximizes the modulus of $[M_k^{-1}g(x^k)]_i$, we get

$$\|M_k^T s_k(\sigma)\|_\infty \geq |[M_k^T s_k(\sigma)]_i| \geq \frac{\sqrt{\|D_k\|^2+12\sigma\|[M_k^{-1}g(x^k)\|_\infty}-\|D_k\|}{6\sigma}.$$

Thus, (18) is obtained and

$$\|s_k(\sigma)\| \geq \frac{1}{\|M_k^T\|}\frac{\sqrt{\|D_k\|^2+12\sigma\|[M_k^{-1}g(x^k)\|_\infty}-\|D_k\|}{6\sigma}.$$

This completes the proof. □

The bounding results proved in Proposition 4.1 are condensed in the following corollary.

COROLLARY 4.1. *Assume that* $c_{\rm bound} > 0$ *is such that*

$$c_{\rm bound} \geq \max\left\{\|D_k\|, \|M_k\|, \|M_k^{-1}\|, \|M_k^{-T}\|_\infty, \|M_k\|_\infty, \|g(x^k)\|_1\right\}. \tag{22}$$

Then,

$$\|M_k^T s_k(\sigma)\|_\infty \leq \frac{\sqrt{c_{\rm bound}^2/\sigma + 12 c_{\rm bound}\|g(x^k)\|}}{6\sqrt{\sigma}} + \frac{c_{\rm bound}}{6\sigma} \tag{23}$$

and

$$\|s_k(\sigma)\|_\infty \leq c_{\rm bound}\left[\frac{\sqrt{c_{\rm bound}^2/\sigma + 12 c_{\rm bound}\|g(x^k)\|}}{6\sqrt{\sigma}} + \frac{c_{\rm bound}}{6\sigma}\right]. \tag{24}$$

Moreover, if $\varepsilon \geq 0$ *is such that* $\|g(x^k)\|_\infty \geq \varepsilon$,

$$\|M_k^T s_k(\sigma)\| \geq \|M_k^T s_k(\sigma)\|_3 \geq \|M_k^T s_k(\sigma)\|_\infty \geq \frac{\sqrt{c_{\rm bound}^2 + 12\sigma\varepsilon/c_{\rm bound}} - c_{\rm bound}}{6\sigma} \tag{25}$$

and

$$\|s_k(\sigma)\| \geq \frac{1}{c_{\rm bound}} \frac{\sqrt{c_{\rm bound}^2 + 12\sigma\varepsilon/c_{\rm bound}} - c_{\rm bound}}{6\sigma}. \tag{26}$$

Finally, if $\varepsilon_2 > 0$ *and* $i \in \{1, \ldots, n\}$ *is such that* $(D_k)_{ii} \leq -\varepsilon_2$,

$$\|M_k^T s_k(\sigma)\| \geq \|M_k^T s_k(\sigma)\|_3 \geq \|M_k^T s_k(\sigma)\|_\infty \geq \frac{\varepsilon_2}{3\sigma} \tag{27}$$

and

$$\|s_k(\sigma)\| \geq \frac{\varepsilon_2}{3 c_{\rm bound}\sigma}. \tag{28}$$

Proof: The inequalities (23), (24), (25), (26), (27), and (28) follow from (15), (16), (18), (19), (20), and (21) using elementary algebraic properties. ☐

Now we wish to prove that the method is well defined. This means that, given x^k such that $g(x^k)$ does not vanish, after a finite number of steps, we find a point that satisfies the sufficient descent condition (3). In addition, we will compute the minimum reduction that is obtained at each iteration k where $g(x^k)$ is not null. Later, this result is used to compute the maximal number of iterations that may occur in which the gradient norm is bigger than a given quantity and the convergence of the gradient norms to zero. We use the only assumption that f is differentiable at x^k.

ASSUMPTION A1. *We say that this assumption holds at an iterate* x^k *generated by Algorithm 2.1 if there exists a non-decreasing function* $\varphi : \mathbb{R}_+ \to \mathbb{R}_+$ *such that* φ *is continuous at the origin,* $\varphi(0) = 0$, *and, for all* $s \in \mathbb{R}^n$,

$$f(x^k + s) \leq f(x^k) + g(x^k)^T s + \varphi(\|s\|_\infty)\|s\|_\infty. \tag{29}$$

The non-decreasing assumption for φ is not restrictive. In fact, if Assumption A1 holds for a function φ that may decrease, it also holds replacing $\varphi(t)$ with $\sup\{\varphi(v), 0 \leq v \leq t\}$ which is obviously non-decreasing. A sufficient condition for the fulfillment of Assumption A1 is the continuity of the partial first derivatives at

x^k. See [13, Vol.II, p.41]. Neither Lipschitz nor Hölder conditions on the derivatives are necessary for this purpose. Under a Lipschitz condition, (29) holds with $\varphi(t) = ct$ and, more generally, under a Hölder condition it holds with $\varphi(t) = ct^\beta$ with $\beta \in (0, 1]$. At Assumption A1, we do not restrict the size of the norm of the increment s. The reason is that possible big differences between f and its linear approximation when $\|s\|$ is large may be represented by a suitable function φ. For example, the function $f(x) = x^4$ does not satisfy (29) if $x^k = 0$ and $\varphi(t) = t^2$ because (29) would fail for $|s| > 1$, but it does defining $\varphi(t) = \max\{t^4, t^2\}$.

For proving Theorem 4.1, we will assume that $g(x^k)$ is not null. If $\|g(x^k)\| = 0$ and some entry of D_k is negative the iteration may not be well defined, in the sense that the sufficient condition (3) may not hold, independently of the size of the regularization parameter. This would happen, for example, if x^k is a minimizer. However, as we will see later, if H_k is the true Hessian at x^k, the iteration is well defined even if $g(x^k) = 0$ and some $(D_k)_{ii}$ is zero.

THEOREM 4.1. *Assume that, at iteration k, Assumption A1 holds, $\|g(x^k)\|_\infty \geq \varepsilon > 0$, $M_k D_k M_k^T$ is a Mixed Factorization of H_k, and $c_{\rm bound} > 0$ satisfies (22). Then, there exists $\bar\sigma \geq \max\{\sigma_{\min}, 1\}$, only dependent on ε, $c_{\rm bound}$, and algorithmic parameters, such that, if $s_k(\sigma)$ is a solution to (1) and $\sigma \geq \bar\sigma$, we have that*

$$f(x^k + s_k(\sigma)) \leq f(x^k) - \frac{1}{4\sqrt{3\sigma} c_{\rm bound}^{3/2}} \varepsilon^{3/2} \tag{30}$$

and

$$f(x^k + s_k(\sigma)) \leq f(x^k) - \alpha \|M_k^T s_k(\sigma)\|_\infty^3. \tag{31}$$

Proof: For simplicity let us call $x = x^k$, $g = g(x^k)$, $H = H(x^k)$, $D = D_k$, $s(\sigma) = s_k(\sigma)$, and $M = M_k$. Recall that, by definition, M is nonsingular. Subproblem (1) is then given by

$$\text{Minimize } g^T s + \frac{1}{2} s^T H s + \sigma \|M^T s\|_3^3. \tag{32}$$

Defining, as in (6),

$$y = M^T s \tag{33}$$

and

$$\underline{g} = M^{-1} g, \tag{34}$$

we have that (32) is equivalent to

$$\text{Minimize } \underline{g}^T y + \frac{1}{2} y^T D y + \sigma \|y\|_3^3. \tag{35}$$

If $\sigma > 0$ and $y(\sigma)$ is a solution to (35), for $i = 1, \ldots, n$, since $|[y(\sigma)]_i|$ decreases as a function of D_{ii}, by (13) and (22), we have that

$$\begin{aligned} |[y(\sigma)]_i| &= \left| \frac{\sqrt{D_{ii}^2 + 12\sigma|\underline{g}_i|} - D_{ii}}{6\sigma} \right| = \frac{\sqrt{D_{ii}^2 + 12\sigma|\underline{g}_i|} - D_{ii}}{6\sigma} = \frac{\sqrt{D_{ii}^2/\sigma + 12|\underline{g}_i|}}{6\sqrt{\sigma}} - \frac{D_{ii}}{6\sigma} \\ &\geq \frac{\sqrt{c_{\rm bound}^2/\sigma + 12|\underline{g}_i|}}{6\sqrt{\sigma}} - \frac{c_{\rm bound}}{6\sigma} \geq \frac{\sqrt{12|\underline{g}_i|}}{6\sqrt{\sigma}} - \frac{c_{\rm bound}}{6\sigma}. \end{aligned} \tag{36}$$

Therefore, since $\underline{g}_i\,[y(\sigma)]_i \leq 0$ for all i, by (36), (34), and (22),

(37)
$$\begin{aligned}
\underline{g}^T y(\sigma) &= -\sum_{i=1}^{n} |\underline{g}_i||[y(\sigma)]_i| \leq -\sum_{i=1}^{n} |\underline{g}_i| \left(\frac{\sqrt{12|\underline{g}_i|}}{6\sqrt{\sigma}} - \frac{c_{\text{bound}}}{6\sigma} \right) \\
&= -\frac{1}{\sqrt{3\sigma}} \left(\sum_{i=1}^{n} |\underline{g}_i|^{3/2} \right) + \frac{c_{\text{bound}}}{6\sigma} \|\underline{g}\|_1 \leq -\frac{1}{\sqrt{3\sigma}} \|\underline{g}\|_\infty^{3/2} + \frac{c_{\text{bound}}}{6\sigma} \|\underline{g}\|_1 \\
&\leq -\frac{1}{\sqrt{3\sigma}} \left(\frac{\|g\|_\infty}{c_{\text{bound}}} \right)^{3/2} + \frac{c_{\text{bound}}^3}{6\sigma}.
\end{aligned}$$

Thus, by $\|g\|_\infty \geq \varepsilon$,

$$\underline{g}^T y(\sigma) \leq \frac{1}{\sqrt{\sigma}} \left(-\frac{1}{\sqrt{3} c_{\text{bound}}^{3/2}} \varepsilon^{3/2} + \frac{c_{\text{bound}}^3}{6\sqrt{\sigma}} \right). \tag{38}$$

Taking

$$\bar{\sigma} \geq \frac{c_{\text{bound}}^9}{3\varepsilon^3} \tag{39}$$

and $\sigma \geq \bar{\sigma}$, we obtain that

$$\frac{c_{\text{bound}}^3}{6\sqrt{\sigma}} \leq \frac{1}{2} \left(\frac{1}{\sqrt{3} c_{\text{bound}}^{3/2}} \varepsilon^{3/2} \right).$$

Therefore, by (38) and under the bound (39),

$$\underline{g}^T y(\sigma) \leq -\frac{1}{2\sqrt{3\sigma} c_{\text{bound}}^{3/2}} \varepsilon^{3/2} \tag{40}$$

and, since, by (33) and (34), $\underline{g}^T y(\sigma) = g^T s(\sigma)$,

$$g^T s(\sigma) \leq -\frac{1}{2\sqrt{3\sigma} c_{\text{bound}}^{3/2}} \varepsilon^{3/2}. \tag{41}$$

Therefore, by Assumption A1 and (41), for $\sigma \geq \bar{\sigma}$,

$$f(x + s(\sigma)) - f(x) \leq -\frac{1}{2\sqrt{3\sigma} c_{\text{bound}}^{3/2}} \varepsilon^{3/2} + \varphi(\|s(\sigma)\|_\infty) \|s(\sigma)\|_\infty.$$

Then, by (24) and assuming

$$\sigma \geq \bar{\sigma} \geq \sigma_{\min}, \tag{42}$$

we have that

$$f(x + s(\sigma)) - f(x) \leq -\frac{1}{2\sqrt{3\sigma} c_{\text{bound}}^{3/2}} \varepsilon^{3/2} + \varphi(\nu(\sigma)) \nu(\sigma), \tag{43}$$

where

$$\nu(\sigma) = c_{\text{bound}} \left[\frac{\sqrt{c_{\text{bound}}^2 / \sigma_{\min} + 12 c_{\text{bound}} \|g\|}}{6\sqrt{\sigma}} + \frac{c_{\text{bound}}}{6\sigma} \right].$$

Defining

$$c_{\rm aux} = c_{\rm bound}\left[\frac{\sqrt{c_{\rm bound}^2/\sigma_{\min} + 12 c_{\rm bound}^2}}{6} + \frac{c_{\rm bound}}{6}\right], \tag{44}$$

if, in addition to (39),

$$\sigma \ge \bar\sigma \ge 1 \tag{45}$$

we obtain that $\sigma \ge \sqrt{\sigma}$ and, thus, by (22) and (43), we have that

$$f(x+s(\sigma)) - f(x) \le \frac{1}{\sqrt{\sigma}}\left(-\frac{1}{2\sqrt{3}c_{\rm bound}^{3/2}}\varepsilon^{3/2} + c_{\rm aux}\,\varphi\left(\frac{c_{\rm aux}}{\sqrt{\sigma}}\right)\right). \tag{46}$$

Let $t_{\rm aux} > 0$ be such that, for all $t \in [0, t_{\rm aux}]$,

$$c_{\rm aux}\varphi(t) \le \frac{1}{2}\left(\frac{1}{2\sqrt{3}c_{\rm bound}^{3/2}}\varepsilon^{3/2}\right). \tag{47}$$

Taking

$$\bar\sigma \ge \left(\frac{c_{\rm aux}}{t_{\rm aux}}\right)^2 = \left(c_{\rm bound}\left[\frac{\sqrt{c_{\rm bound}^2/\sigma_{\min} + 12c_{\rm bound}^2} + c_{\rm bound}}{6t_{\rm aux}}\right]\right)^2, \tag{48}$$

we obtain that, for all $\sigma \ge \bar\sigma$,

$$\frac{c_{\rm aux}}{\sqrt{\sigma}} \le t_{\rm aux}.$$

Thus, for all $\sigma \ge \bar\sigma$, (30) follows from (46) and (47).

By (22) and (23), for $\sigma \ge 1$,

$$\begin{aligned}\|M^T s(\sigma)\|_\infty &\le \frac{\sqrt{c_{\rm bound}^2/\sigma + 12c_{\rm bound}\|g\|}}{6\sqrt{\sigma}} + \frac{c_{\rm bound}}{6\sqrt{\sigma}}\\ &\le \frac{\sqrt{c_{\rm bound}^2 + 12c_{\rm bound}^2}}{6\sqrt{\sigma}} + \frac{c_{\rm bound}}{6\sqrt{\sigma}} = \left(\frac{\sqrt{13}+1}{6}\right)\frac{c_{\rm bound}}{\sqrt{\sigma}} < \frac{c_{\rm bound}}{\sqrt{\sigma}}.\end{aligned}$$

Therefore,

$$\alpha\|M^T s(\sigma)\|_\infty^3 < \alpha\frac{c_{\rm bound}^3}{\sigma^{3/2}}. \tag{49}$$

Taking

$$\bar\sigma \ge \alpha\frac{4\sqrt{3}c_{\rm bound}^{9/2}}{\varepsilon^{3/2}}, \tag{50}$$

we have that, for all $\sigma \ge \bar\sigma$,

$$\alpha\frac{c_{\rm bound}^3}{\sigma^{3/2}} \le \frac{1}{4\sqrt{3\sigma}c_{\rm bound}^{3/2}}\varepsilon^{3/2}$$

and (31) follows for all $\sigma \ge \bar\sigma$ from (30) and (49). This completes the proof. (The $\bar\sigma$ in the thesis must satisfy (39,42,45,48,50).) □

COROLLARY 4.2. *Assume that, at iteration k, Assumption A1 holds, $\|g(x^k)\|_\infty > 0$, $M_k D_k M_k^T$ is a Mixed Factorization of H_k, and $c_{\text{bound}} > 0$ satisfies (22). Then the k-th iteration finishes with the fulfillment of (3) after a finite number of increases of the regularization parameter σ.*

THEOREM 4.2. *Assume that, for all $k = 0, 1, \ldots, k_\varepsilon$, Assumption A1 holds, $M_k D_k M_k^T$ is a Mixed Factorization of H_k, $c_{\text{bound}} > 0$ satisfies (22), $\|g(x^k)\|_\infty \geq \varepsilon > 0$, and $f(x^k) > f_{\text{target}}$. Then, k_ε is not bigger than $(f(x^0) - f_{\text{target}})$ times a positive quantity that only depends on ε, c_{bound}, and algorithmic parameters.*

Proof: For all $k \in \{0, 1, \ldots, k_\varepsilon\}$ the assumptions of Theorem 4.1 hold. Then, by Corollary 4.2,

$$f(x^{k+1}) = f(x^k + s^k) \leq f(x^k) - \alpha \|M_k^T s^k\|_\infty^3 \tag{51}$$

for all $k = 0, 1, \ldots, k_\varepsilon$. Moreover, by (2), (4), the updating rule of σ_{big} in Algorithm 2.1, and Theorem 4.1,

$$\sigma_k \leq \sigma_{\text{big}} = \max\{\sigma_{\text{bles}}, \sigma_0, \sigma_1, \ldots, \sigma_{k-1}\} \leq \max\{\sigma_{\text{bles}}, \kappa\bar{\sigma}\},$$

where $\bar{\sigma}$, that only depends on ε, c_{bound}, and algorithmic parameters, is given in Theorem 4.1. Therefore, defining $\hat{\sigma} = \max\{\sigma_{\text{bles}}, \kappa\bar{\sigma}\}$, by (25),

$$\|M_k^T s^k\|_\infty \geq \frac{\sqrt{c_{\text{bound}}^2 + 12\hat{\sigma}\varepsilon / c_{\text{bound}}} - c_{\text{bound}}}{6\hat{\sigma}} = \frac{c_{\text{bound}}}{6\hat{\sigma}} \left[\sqrt{1 + 12\hat{\sigma}\varepsilon / c_{\text{bound}}^3} - 1\right],$$

for all $k = 0, 1, \ldots, k_\varepsilon$, and, by (51),

$$f(x^{k+1}) \leq f(x^k) - \alpha \left(\frac{c_{\text{bound}}}{6\hat{\sigma}} \left[\sqrt{1 + 12\hat{\sigma}\varepsilon / c_{\text{bound}}^3} - 1\right]\right)^3. \tag{52}$$

This implies that, for all $k = 0, 1, \ldots, k_\varepsilon$, $f(x^k) - f(x^{k+1})$ is bounded below by a positive quantity that only depends on ε, c_{bound}, and algorithmic parameters. This completes the proof. □

Theorem 4.2 says that, given $\varepsilon > 0$ and $f_{\text{target}} \in \mathbb{R}$, after $K(\varepsilon)$ iterations Algorithm 2.1 finds a point x^k such that $\|g(x^k)\|_\infty \leq \varepsilon$ or $f(x^k) \leq f_{\text{target}}$. The cost of that process is, at most, the computation of $K(\varepsilon)$ gradients and functional values at the iterates x^k, plus the computation of $K(\varepsilon)$ factorizations, plus the computation of functional values at the rejected trial points. Now, at each iteration, according to Theorem 4.1, the number of rejected trial points $s_k(\sigma)$ is, at most, the maximal number of increases of σ that starts not smaller than $\sigma_{\min}$ and goes up to $\hat{\sigma} = \max\{\sigma_{\text{bles}}, \kappa\bar{\sigma}\}$ at most. Recall that $\hat{\sigma}$ only depends on ε, c_{bound}, and algorithmic parameters. This number cannot exceed $\log_2(\hat{\sigma}/\sigma_{\min})$. Therefore, for finding a solution with gradient precision ε, we need at most $K(\varepsilon)$ gradient evaluations and factorizations plus $K(\varepsilon)(1 + \log_2(\hat{\sigma}/\sigma_{\min}))$ functional evaluations.

From Theorem 4.2 the following convergence theorem holds.

THEOREM 4.3. *Assume that the sequence $\{x^k\}$ is generated by Algorithm 2.1 and, for all $k = 0, 1, 2, \ldots$, $\|g(x^k)\| > 0$, Assumption A1 holds, and c_{bound} satisfies (22). Then,*

$$\lim f(x^k) = -\infty \ \text{ or } \ \liminf \|g(x^k)\| = 0.$$

Proof: Since the sequence $\{f(x^k)\}$ is strictly decreasing, if this sequence does not go to $-\infty$, then it is bounded below. Taking $f_{\rm target}$ as a lower bound of $\{f(x^k)\}$, by Theorem 4.2, we have that for all $\varepsilon > 0$ there exists k_ε such that, for some $k \geq k_\varepsilon$, $\|g(x^k)\|_\infty \leq \varepsilon$. This implies the thesis. □

In Theorem 4.3, we proved that, if $\{f(x^k)\}$ is bounded below, $\liminf \|g(x^k)\| = 0$. The reason why we cannot prove that $\lim \|g(x^k)\| = 0$ is associated with the updating rule for $\sigma_{\rm new}$ in (2) and in (4) when $\sigma = 0$. Note that, according to (2) and (4), we could choose $\sigma_{\rm new} \leftarrow \sigma_k$ at every iteration, in such a way that the decrease at each iteration could tend to zero at the iterates at which $\|g(x^k)\|_\infty \geq \varepsilon$. In this way, it remains valid that the maximal number of *consecutive* iterations at which $\|g(x^k)\|_\infty \geq \varepsilon$ is bounded but this bound is not valid anymore for non-consecutive iterations. We will see later that, in practice, we try to choose $\sigma_{\rm new} \leftarrow \sigma_k/2$ in (2) and in (4) when $\sigma = 0$. In the following theorem we prove that convergence to zero of the whole sequence $\{\|g(x^k)\|\}$ occurs if we use a slightly more restrictive choice of σ at the beginning of each iteration.

THEOREM 4.4. *Assume that the sequence $\{x^k\}$ is generated by Algorithm 2.1 and, for all $k = 0, 1, 2, \dots$, $\|g(x^k)\| > 0$, Assumption A1 holds, and $c_{\rm bound}$ satisfies (22). Moreover, assume that in (2) and in (4) when $\sigma = 0$, we impose the condition*

$$\sigma_{\rm new} \leq \sigma_{\rm safe}, \tag{53}$$

where $\sigma_{\rm safe} \geq \sigma_{\min} > 0$ is a new given parameter of the algorithm. Then,

$$\lim f(x^k) = -\infty \ \textit{or} \ \lim \|g(x^k)\| = 0.$$

Proof: Suppose that $\lim f(x^k) > -\infty$. By Theorem 4.2, at each iteration k, (52) holds replacing $\hat{\sigma}$ with $\max\{\sigma_{\rm safe}, \bar{\sigma}_k\}$, where $\bar{\sigma}_k$ corresponds to the value of $\bar{\sigma}$ given by Theorem 4.1 for iteration k. Let $\varepsilon > 0$ be arbitrary. Then, at each iteration where $\|g(x^k)\| \geq \varepsilon$ we obtain a functional decrease bounded away from zero. This implies that the number of iterations at which $\|g(x^k)\|_\infty \geq \varepsilon$ is finite. This implies that $\lim \|g(x^k)\|_\infty = 0$. □

Given the present state of the art of Cubic Regularization methods for unconstrained optimization, the final results of this section are far from being surprising. In particular, they can be essentially obtained as consequences of results obtained in [33], with additional care in the determination of the complexity constants. We state them here because the proofs become more simple thanks to the employment of Proposition 4.1. We will use the following assumption, which holds whenever $\nabla^2 f(x)$ satisfies a Lipschitz condition on a sufficient large open and convex set that includes all the iterates x^k and trial points $x^k + s$ generated by Algorithm 2.1.

ASSUMPTION A2. *We say that this assumption holds at an iterate x^k generated by Algorithm 2.1 if there exists $\gamma > 0$ such that for all $s \in \mathbb{R}^n$ such that $x^k + s$ lies in an open and convex set that contains x^k and the trial points generated by Algorithm 2.1,*

$$f(x^k + s) \leq f(x^k) + g(x^k)^T s + \frac{1}{2} s^T \nabla^2 f(x^k) s + \gamma \|s\|_\infty^3 \tag{54}$$

and

$$\big\| g(x^k + s) - \big(g(x^k) + \nabla^2 f(x^k) s\big) \big\| \leq \gamma \|s\|_\infty^2. \tag{55}$$

THEOREM 4.5. *Assume that, at every iteration k of Algorithm 2.1, Assumption A2 holds, $H_k = \nabla^2 f(x^k)$, $M_k D_k M_k^T$ is a Mixed Factorization of H_k, and $c_{\text{bound}} > 0$ satisfies (22). Then, there exists $\tilde\sigma \geq \sigma_{\min}$, that only depends on parameters of the algorithm and characteristics of the problem, such that, if $s_k(\sigma)$ is a solution to (1) and $\sigma \geq \tilde\sigma$, we have that*

$$f(x^k + s_k(\sigma)) - f(x^k) \leq -\alpha \|M_k^T s_k(\sigma)\|_\infty^3. \tag{56}$$

Moreover,

$$f(x^{k+1}) \leq f(x^k) - \alpha c \|g(x^{k+1})\|^{3/2}, \tag{57}$$

where c is a constant that only depends on parameters of the algorithm and characteristics of the problem.

Proof: On the one hand, by (54), (22), and since $s_k(\sigma)$ is a solution to (1) with $H_k = \nabla^2 f(x^k)$, we have that

$$\begin{aligned}
f(x^k + s_k(\sigma)) &\leq f(x^k) + g(x^k)^T s_k(\sigma) + \frac{1}{2} s_k(\sigma)^T H_k s_k(\sigma) + \gamma \|s_k(\sigma)\|_\infty^3 \\
&= f(x^k) + g(x^k)^T s_k(\sigma) + \frac{1}{2} s_k(\sigma)^T H_k s_k(\sigma) + \gamma \|s_k(\sigma)\|_\infty^3 \\
&\qquad + \sigma \|M_k^T s_k(\sigma)\|_3^3 - \sigma \|M_k^T s_k(\sigma)\|_3^3 \\
&\leq f(x^k) - \sigma \|M_k^T s_k(\sigma)\|_3^3 + \gamma \|s_k(\sigma)\|_\infty^3 \\
&\leq f(x^k) - \sigma \|M_k^T s_k(\sigma)\|_\infty^3 + \gamma \|s_k(\sigma)\|_\infty^3 \\
&\leq f(x^k) - \sigma \|M_k^T s_k(\sigma)\|_\infty^3 + \gamma \|M_k^{-T}\|_\infty^3 \|M_k^T s_k(\sigma)\|_\infty^3 \\
&= f(x^k) + \left(-\sigma + \gamma \|M_k^{-T}\|_\infty^3\right) \|M_k^T s_k(\sigma)\|_\infty^3 \\
&\leq f(x^k) + \left(-\sigma + \gamma c_{\text{bound}}^3\right) \|M_k^T s_k(\sigma)\|_\infty^3.
\end{aligned}$$

Therefore, defining

$$\tilde\sigma = \max\{\sigma_{\min}, \alpha + \gamma c_{\text{bound}}^3\},$$

we have that (56) follows for all $\sigma \geq \tilde\sigma$.

On the other hand, for all $\sigma \geq 0$, by (55), and since $s_k(\sigma)$ is a solution to (1) with $H_k = \nabla^2 f(x^k)$, we have that

$$\begin{aligned}
\|g(x^k + s_k(\sigma))\|_\infty &\leq \left\|g(x^k) + \nabla^2 f(x^k) s_k(\sigma)\right\|_\infty + \gamma \|s_k(\sigma)\|_\infty^2 \\
&\leq \left\| g(x^k) + \nabla^2 f(x^k) s_k(\sigma) + \sigma \left(\nabla \|M_k^T s\|_3^3\right)\big|_{s_k(\sigma)} \right\|_\infty \\
&\qquad + \sigma \left\| \left(\nabla \|M_k^T s\|_3^3\right)\big|_{s_k(\sigma)} \right\|_\infty + \gamma \|s_k(\sigma)\|_\infty^2 \\
&= \sigma \left\| \left(\nabla \|M_k^T s\|_3^3\right)\big|_{s_k(\sigma)} \right\|_\infty + \gamma \|s_k(\sigma)\|_\infty^2.
\end{aligned} \tag{58}$$

By the definition of Algorithm 2.1, we have that $x^{k+1} = x^k + \sigma_k s^k$ with $s^k = s_k(\sigma_k)$ and σ_k satisfying $\sigma_k \leq \max\{\sigma_{\text{bles}}, \kappa\tilde\sigma\}$. Therefore, by (58) and (22), since

$$\nabla \|M_k^T s\|_3^3 = 3 M_k \left([M_k^T s]_1^2 \operatorname{sg}([M_k^T s]_1), \ldots, [M_k^T s]_n^2 \operatorname{sg}([M_k^T s]_n)\right)^T$$

and, in consequence,

$$\left\|\left(\nabla\|M_k^T s\|_3^3\right)\right\|_\infty \leq 3\|M_k\|_\infty \|M_k^T s\|_\infty^2,$$

we have that

$$\begin{aligned}
\|g(x^{k+1})\|_\infty &\leq \max\{\sigma_{\text{bles}}, \kappa\tilde{\sigma}\} \left\| \left(\nabla\|M_k^T s\|_3^3\right)\big|_{s^k} \right\|_\infty + \gamma\|s^k\|_\infty^2 \\
&\leq \max\{\sigma_{\text{bles}}, \kappa\tilde{\sigma}\} \left\| \left(\nabla\|M_k^T s\|_3^3\right)\big|_{s^k} \right\|_\infty + \gamma\|s^k\|_\infty^2 \\
&\leq \max\{\sigma_{\text{bles}}, \kappa\tilde{\sigma}\} 3\|M_k\|_\infty \|M_k^T s^k\|_\infty^2 + \gamma\|M_k^{-T}\|_\infty \|M_k^T s^k\|_\infty^2 \\
&\leq \left(3\max\{\sigma_{\text{bles}}, \kappa(\alpha + \gamma c_{\text{bound}}^3\} c_{\text{bound}} + \gamma c_{\text{bound}}\right) \|M_k^T s^k\|_\infty^2 \\
&= \bar{c}\|M_k^T s^k\|_\infty^2,
\end{aligned}$$

where $\bar{c} = 3\max\{\sigma_{\text{bles}}, \kappa(\alpha + \gamma c_{\text{bound}}^3\} c_{\text{bound}} + \gamma c_{\text{bound}}$ is a constant that depends on c_{bound}, γ, and the algorithmic constants α, κ, and σ_{bles}. Therefore,

$$\|g(x^{k+1})\|_\infty^{3/2} \leq \bar{c}^{3/2}\|M_k^T s^k\|_\infty^3.$$

Thus, (57) follows from (56). □

Note that, in Theorem 4.5, the well-definiteness of iteration k holds independently of assumptions on the gradient norms and the matrix D_k. This observation is important for the following theorem, where a second-order complexity result is proved. In Theorem 4.6, we prove that, given $\varepsilon_2 > 0$, the number of iterations at which there exists an entry of D_k smaller than $-\varepsilon_2$ is bounded by a multiple of ε_2^{-3}. It is interesting to observe that this fact is independent of the factorization used. Using the spectral QDQ^T factorization, this result is standard because the entries of D_k are the eigenvalues of H_k. However the result holds in the general case because the positive-definiteness of H_k is well represented by the positiveness of the entries of D_k for every Mixed Factorization $M_k D_k M_k^T$. The complexity proof in this case is substantially different, and more simple, than the one given in [33] for a similar case.

THEOREM 4.6. *Assume that, at every iteration k of Algorithm 2.1, Assumption A2 holds, $H_k = \nabla^2 f(x^k)$, $M_k D_k M_k^T$ is a Mixed Factorization of H_k, and $c_{\text{bound}} > 0$ satisfies (22). Then, given $\varepsilon > 0$ and $f_{\text{target}} \in \mathbb{R}$, the number of iterations k such that*

$$\|g(x^{k+1})\|_\infty \geq \varepsilon \text{ or } f(x^{k+1}) > f_{\text{target}}$$

is bounded above by

$$\left(f(x^0) - f_{\text{target}}\right) \varepsilon^{-3/2}$$

times a constant that only depends on parameters of the algorithm and characteristics of the problem. Moreover, given $\varepsilon_2 > 0$, the number of iterations k such that there exists $i \in \{1, \dots, n\}$ with $(D_k)_{ii} \leq -\varepsilon_2$ is bounded above by

$$\left(f(x^0) - f_{\text{target}}\right) \varepsilon_2^{-3}$$

times a constant that only depends on parameters of the algorithm and characteristics of the problem.

Proof: The first part of the thesis follows directly from (57). For the second part, observe that, if $(D_k)_{ii} \leq -\varepsilon_2$, then, by (20),

$$\|M_k^T s^k\|_\infty = \|M_k^T s_k(\sigma_k)\|_\infty \geq \frac{|(D_k)_{ii}|}{3\sigma_k} \geq \frac{\varepsilon_2}{3\max\{\sigma_{\text{bles}}, \kappa\tilde{\sigma}\}},$$

where $\tilde{\sigma}$ is the one defined at Theorem 4.5. Therefore, the second part of the thesis follows from (56). ☐

Theorem 4.6 reports the maximal number of iterations that are necessary to obtain a gradient smaller than ε or a positive semidefinite Hessian up to tolerance ε_2, respectively, under Assumption A2. The complexity analysis is completed computing the total number of functional evaluations. By Theorem 4.4, the maximal number of rejected trial points at each iteration of the algorithm does not exceed $\log_2(\kappa\tilde{\sigma}/\sigma_{\min})$, where $\tilde{\sigma}$ does not depend on ε. Combining this computation with the theorems above we obtain the expected result that precision ε on the gradient demands at most $O(\varepsilon^{-3/2})$ iterations and function evaluations whereas precision ε_2 on the positive semidefiniteness of the Hessian demands at most $O(\varepsilon_2^{-3})$ iterations and function evaluations.

THEOREM 4.7. *Assume that, at every iteration k of Algorithm 2.1, Assumption A2 holds, $H_k = \nabla^2 f(x^k)$, $M_k D_k M_k^T$ is a Mixed Factorization of H_k, and $c_{\text{bound}} > 0$ satisfies (22). Then, the sequence $\{x^k\}$ given by Algorithm 2.1 is well defined,*

$$\lim \|g(x^k)\| = 0 \text{ and } \lim \min\{0, (D_k)_{11}, \ldots, (D_k)_{nn}\} = 0. \tag{59}$$

Proof: By Theorem 4.5, given $\varepsilon > 0$, the number of iterations for which $\|g(x^k)\|_\infty > \varepsilon$ is finite. Therefore, $\lim \|g(x^k)\| = 0$. Analogously, the number of iterations for which $\min\{0, \min\{(D_k)_{ii}, i = 1, \ldots, n\}\} > \varepsilon$ is finite. Therefore, $\lim \min\{0, \min\{(D_k)_{ii}, i = 1, \ldots, n\}\} = 0$. ☐

5. Numerical experiments.

We implemented Algorithm 2.1 in Fortran 90 employing the BPK-based MDM^T Mixed Factorization as well as the Mixed Factorization based on the QDQ^T spectral decomposition. The Bunch-Parlett-Kaufman factorization was computed with subroutine `dsytrf_rk` from Lapack [1] for dense matrices. Moreover, we also considered a sparse version using subroutine MA57 from HSL [38]. For the Mixed Factorization based on the QDQ^T spectral decomposition we used subroutine `dsyev` from Lapack.

At Step 3, we choose $\sigma_{\text{new}} = \max\{\sigma_{\min}, \frac{1}{2}\sigma_{\text{lnn}}\}$, where σ_{lnn} is the latest non-null σ_ℓ, $\ell = 0, \ldots, k-1$, and $\sigma_{\text{lnn}} = 0$ if $\sigma_0 = \cdots = \sigma_{k-1} = 0$. At Step 4, the value of σ_{new} is chosen in the same way if $\sigma = 0$; while $\sigma_{\text{new}} = \kappa\sigma$, otherwise. However, there are two situations in which, if the value of σ_{new} was computed using σ_{lnn} (at Step 3 or 4), it may be redefined. In first place, if $\sigma_{\text{new}} > \sigma_{\min}$ and $\|s_k(\sigma_{\text{new}})\| < \sqrt{\epsilon_{\text{mach}}} \max\{1, \|x^k\|\}$ then we redefine $\sigma_{\text{new}} = \sigma_{\min}$. In second place, if $\sigma_{\text{new}} = \sigma_{\min}$ and $\|s_k(\sigma_{\text{new}})\| > \max\{1, \|x^k\|\}$ then σ_{new} is redefined to the first value in $\{10\sigma_{\min}, 10^2\sigma_{\min}, \ldots\}$, limited to σ_{bles}, such that $\|s_k(\sigma_{\text{new}})\| \leq \max\{1, \|x^k\|\}$. The first possible modification has the purpose of avoiding stagnation due to a large value of σ inherited from previous iterations; while the second possible modification aims to reduce the influence of the arbitrary parameter $\sigma_{\min}$.

In the numerical experiments, we arbitrarily considered $\alpha = 10^{-8}$, $\kappa = 10$, $\sigma_{\min} = 10^{-8}$, and $\sigma_{\text{bles}} = 10^8$; while ϵ_{mach} is the machine ϵ, i.e. the smallest $\epsilon > 0$ such that

$1 + \epsilon \neq 1$. As stopping criterion we considered the condition

$$\|g(x^k)\|_\infty \leq \varepsilon \tag{60}$$

with $\varepsilon = 10^{-8}$. As it will be seen in the numerical experiments, in a few cases, the method may also stop by any of the following alternative stopping criteria:

1. $\|g(x^{k-\ell})\|_\infty < \sqrt{\varepsilon}$ for all $0 \leq \ell < 100$;
2. $\|g(x^{k-\ell})\|_\infty < \varepsilon^{1/4}$ for all $0 \leq \ell < 1{,}000$;
3. $\|g(x^{k-\ell})\|_\infty < \varepsilon^{1/8}$ for all $0 \leq \ell < 5{,}000$;
4. $s_{\rm trial}$ is the Newton step, $x^k + s_{\rm trial}$ does not satisfy the sufficient descent condition (3), but $\|s_{\rm trial}\| \leq \sqrt{\varepsilon}$ and $\|g(x^k + s_{\rm trial})\|_\infty \leq \varepsilon$;
5. $s_{\rm trial}$ is the Newton step, $x^k + s_{\rm trial}$ does not satisfy the sufficient descent condition (3), and $\|s_{\rm trial}\| \leq \sqrt{\varepsilon}$;
6. $f(x^k) \leq f_{\rm target}$;
7. $x^k + s_{\rm trial}$ does not satisfy the sufficient descent condition (3) but $f(x^k + s_{\rm trial}) \leq f_{\rm target}$;
8. $x^k = x^{k+1}$ and $f(x^k) \leq f(x^k \pm h_i e_i)$ with $h_i = \epsilon_{\rm mach} \max\{1, |x_i^k|\}$ for all $1 \leq i \leq n$;
9. $f(x^k) = f(x^{k-\ell})$ for all $0 \leq \ell < 10$.

In the numerical experiments, the fulfillment of (60) will be reported as "STOP=0"; while the other cases will be reported making reference to the number in the enumeration above. In cases 4 and 7, the method returns $x^k + s_{\rm trial}$ as an approximation to a solution. In all the other cases it returns x^k. It should be noted that the stopping criteria above were chosen in such a way that the method never stops by another stopping criterion such as maximum of iterations, maximum of functional evaluations, or a limit in the CPU time.

The Fortran 90 implementation of Algorithm 2.1 is freely available at http://www.ime.usp.br/~egbirgin/. Interfaces for solving user-defined problems coded in Fortran 90 as well as problems from the CUTEst [24] collection are available. All tests reported below were conducted on a computer with 3.5 GHz Intel Core i7 processor and 16GB 1600 MHz DDR3 RAM memory, running OS X Yosemite (version 10.10.5). Codes were compiled by the GFortran compiler of GCC (version 7.2.0) with the -O3 optimization directive enabled.

5.1. Bunch-Parlett-Kaufman-based versus spectral-based mixed factorization. In this section we analyze the behavior of Algorithm 2.1 in conection with the BPK-based and the spectral-based mixed factorizations. We considered all the 87 unconstrained minimization problems from the CUTEst collection [24]. The same dimensions chosen in [30, 6] were preserved (most of the problems have $n = 1{,}000$ variables), since, in this section, we are using dense linear algebra subroutines for computing both MDM^T Mixed Factorizations. These problems correspond to *all* the unconstrained problems from the CUTEst collection with available second-order derivatives.

For a given problem, let f_1 and f_2 be the value of the objective function at the final iterate delivered by Algorithm 2.1 with the BPK-based and the spectral-based MDM^T Mixed Factorizations, respectively. Following [4], we say that the two methods found *equivalent solutions* if

$$\frac{f_i - f_{\rm best}}{\max\{1, |f_{\rm best}|\}} \leq 10^{-8} \text{ for } i = 1, 2, \tag{61}$$

where $f_{\rm best} = \min\{f_1, f_2\}$. The 87 problems will be separated into two sets. Set 1 will be given by 59 problems in which the two methods found equivalent solutions and stopped both satisfying the same stopping criterion SC with SC $\in \{0, 4, 6, 7\}$. Set 2 will contain the remaining 28 problems. Problems in Set 1 will be used to analyze the efficiency of the methods; while problems in Set 2 will be observed with an eye on robustness. Tables 1 and 2 display detailed information regarding the performance of Algorithm 2.1 in problems on sets 1 and 2, respectively.

For analyzing the efficiency of the methods on the 59 problems on Set 1, we used performance profiles [17]. See Figure 2. By definition of the performance profiles and the way in which the problems were selected, all curves reach the value 1 at the right-hand-side of the graphic. Thus, these pictures evaluate efficiency only. As expected, the picture in the top of Figure 2 shows that the variant of Algorithm 2.1 that uses the spectral-based MDM^T Mixed Factorization uses less functional evaluations; while the picture in the bottom of Figure 2 shows that the variant of Algorithm 2.1 that uses the BPK-based MDM^T Mixed Factorization is much faster.

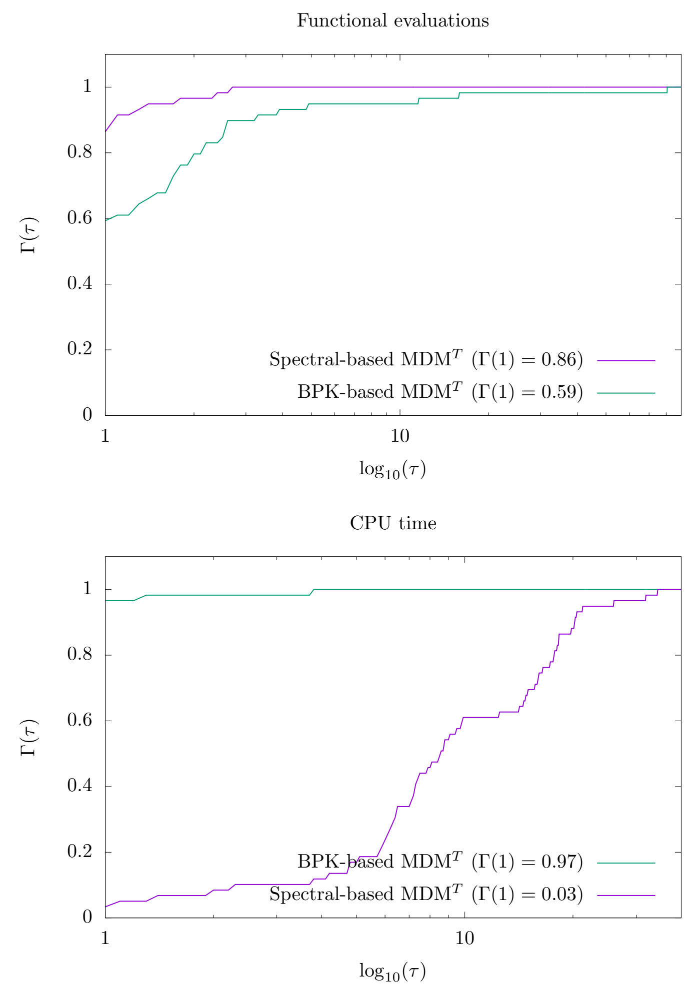

FIG. 2. *Performance profiles considering 59 problems in which the two versions of Algorithm 2.1 found equivalent solutions and stopped satisfying the same stopping criteria related to a small gradient (criteria 0 or 4) or related to achieving a target functional value (criteria 6 and 7).*

Problem	n	BPK-based MDM^T						Spectral-based MDM^T					
		$f(x^*)$	$\|g(x^*)\|_\infty$	#it	#f	Time	SC	$f(x^*)$	$\|g(x^*)\|_\infty$	#it	#f	Time	SC
ARGLINA	500	5.00000D+02	1.6D−13	1	2	3.63	0	5.00000D+02	1.5D−12	1	2	2.90	0
ARWHEAD	1000	0.00000D+00	1.2D−12	6	7	0.62	0	0.00000D+00	1.2D−12	6	7	5.41	0
BDQRTIC	1000	3.98382D+03	2.2D−13	10	11	1.11	0	3.98382D+03	2.2D−13	10	11	18.28	0
BRYBND	1000	2.22994D−18	8.8D−09	10	12	1.07	0	5.05988D−27	1.8D−11	12	16	21.64	0
CRAGGLVY	1000	3.36423D+02	5.3D−15	15	16	1.58	4	3.36423D+02	5.1D−15	15	16	25.15	4
CURLY10	1000	-1.00316D+05	9.5D−13	12	13	1.26	0	-1.00316D+05	7.4D−13	20	23	32.73	0
CURLY20	1000	-1.00316D+05	1.8D−12	9	10	1.01	0	-1.00316D+05	1.8D−12	20	24	32.15	0
CURLY30	1000	-1.00316D+05	3.5D−12	9	10	1.07	0	-1.00316D+05	4.0D−11	23	27	36.77	0
DIXMAANA	900	1.00000D+00	0.0D+00	6	9	0.48	0	1.00000D+00	1.1D−20	6	7	7.10	0
DIXMAANB	900	1.00000D+00	1.7D−10	40	50	3.57	0	1.00000D+00	9.5D−19	10	13	13.34	0
DIXMAANC	900	1.00000D+00	1.3D−23	23	27	2.32	0	1.00000D+00	7.3D−17	9	11	11.76	0
DIXMAAND	900	1.00000D+00	9.9D−10	28	34	2.93	0	1.00000D+00	4.4D−14	17	21	21.27	0
DIXMAANE	900	1.00000D+00	7.2D−23	9	10	0.73	0	1.00000D+00	1.0D−10	7	8	9.08	0
DIXMAANF	900	1.00000D+00	9.4D−15	23	26	2.11	0	1.00000D+00	1.3D−10	11	13	15.74	0
DIXMAANG	900	1.00000D+00	5.4D−11	26	29	2.45	0	1.00000D+00	6.2D−16	17	21	24.13	0
DIXMAANH	900	1.00000D+00	8.6D−12	34	39	2.97	0	1.00000D+00	1.0D−08	15	18	21.07	0
DIXMAANI	900	1.00000D+00	2.9D−12	17	18	1.37	0	1.00000D+00	1.4D−15	9	10	11.72	0
DIXMAANJ	900	1.00000D+00	2.0D−14	31	35	2.81	0	1.00000D+00	1.0D−10	14	16	20.59	0
DIXMAANK	900	1.00000D+00	1.9D−15	29	33	2.54	0	1.00000D+00	4.5D−09	16	20	23.09	0
DIXMAANL	900	1.00000D+00	2.7D−09	29	33	2.49	0	1.00000D+00	1.2D−10	16	19	23.41	0
DIXON3DQ	1000	0.00000D+00	0.0D+00	1	2	0.12	0	1.19650D−21	1.6D−13	1	2	1.93	0
DQDRTIC	1000	0.00000D+00	0.0D+00	1	2	0.11	0	0.00000D+00	0.0D+00	1	2	0.67	0
DQRTIC	1000	2.23542D−10	4.3D−09	34	35	3.56	0	2.23542D−10	4.3D−09	34	35	22.23	0
EDENSCH	1000	6.00328D+03	1.5D−10	12	13	1.25	0	6.00328D+03	1.5D−10	12	13	22.79	0
EIGENALS	420	1.40870D−24	6.2D−11	414	568	8.81	0	1.98607D−21	7.4D−10	133	219	16.98	0
EIGENBLS	420	1.07055D−16	4.4D−09	204	283	3.94	0	4.82353D−23	6.2D−12	120	174	16.54	0
EIGENCLS	462	1.47145D−27	3.8D−14	283	369	6.99	0	2.31494D−20	2.2D−10	84	115	15.61	0
ENGVAL1	1000	1.10819D+03	1.3D−12	8	9	0.86	0	1.10819D+03	1.3D−12	8	9	15.22	0
FLETCBV2	1000	-5.01429D−01	8.4D−09	1	2	0.11	0	-5.01429D−01	8.4D−09	1	2	1.98	0
FLETCHCR	1000	1.29422D−30	4.4D−14	1442	1882	167.11	0	1.71563D−23	5.8D−12	1438	1942	2491.96	0
FMINSRF2	961	1.00000D+00	1.9D−15	58	107	6.13	0	1.00000D+00	3.3D−12	74	138	120.93	0
FMINSURF	961	1.00000D+00	2.9D−09	110	195	14.66	0	1.00000D+00	1.5D−12	74	131	123.33	0
GENHUMPS	1000	6.21560D−26	1.3D−13	3022	3938	340.37	0	1.65481D−24	8.0D−13	1544	2015	1619.88	0
GENROSE	1000	1.00000D+00	5.4D−13	707	1037	75.25	0	1.00000D+00	1.1D−09	624	1053	1066.10	0
HILBERTB	500	5.38430D−26	1.3D−13	1	2	0.05	0	2.32994D−25	7.4D−13	1	2	0.24	0
LIARWHD	1000	9.44338D−26	2.3D−11	12	13	1.25	0	9.44338D−26	2.3D−11	12	13	10.00	0
MODBEALE	1000	6.44647D−28	5.4D−13	24	26	2.38	0	5.94274D−16	3.4D−09	8	10	14.06	0
MOREBV	1000	7.32887D−13	4.7D−11	1	2	0.12	0	7.33318D−13	4.7D−11	1	2	1.74	0
MSQRTALS	1024	1.48095D−25	7.8D−14	308	450	48.27	0	1.76366D−18	6.7D−10	32	39	67.16	0
MSQRTBLS	1024	5.11171D−18	4.6D−09	336	475	52.17	0	2.04811D−22	1.9D−11	26	30	54.93	0
NCB20B	1000	1.67601D+03	2.3D−11	42	77	4.29	0	1.67601D+03	3.9D−09	15	16	27.68	0
NONDIA	1000	1.78727D−26	4.2D−11	6	7	0.56	0	1.78727D−26	4.2D−11	6	7	4.90	0
NONDQUAR	1000	3.18493D−13	9.5D−09	22	23	2.06	0	3.18493D−13	9.5D−09	22	23	41.43	0
OSCIPATH	500	9.99967D−01	1.3D−12	2	3	0.03	0	9.99967D−01	1.3D−12	2	3	0.47	0
PENALTY1	1000	9.68618D−03	2.4D−13	41	51	4.91	0	9.68618D−03	3.0D−09	40	50	35.65	0
POWELLSG	1000	3.29204D−10	8.7D−09	20	21	1.93	0	3.29204D−10	8.7D−09	20	21	11.80	0
POWER	1000	1.42811D−12	7.4D−09	33	34	3.81	0	1.42811D−12	7.4D−09	33	34	65.76	0
QUARTC	1000	2.23542D−10	4.3D−09	34	35	3.28	0	2.23542D−10	4.3D−09	34	35	20.67	0
SCHMVETT	1000	-2.99400D+03	1.9D−13	3	4	0.28	0	-2.99400D+03	1.9D−13	3	4	5.72	0
SPARSINE	1000	3.44543D−18	1.1D−09	1244	1613	123.38	0	9.86306D−16	6.2D−09	19	20	33.14	0
SPARSQUR	1000	4.49227D−11	7.6D−09	22	23	2.28	0	4.49227D−11	7.6D−09	22	23	40.36	0
SROSENBR	1000	3.34168D−18	1.9D−09	8	18	0.75	0	3.89179D−18	1.4D−09	8	18	4.81	0
TESTQUAD	1000	0.00000D+00	0.0D+00	1	2	0.10	0	0.00000D+00	0.0D+00	1	2	0.58	0
TOINTGSS	1000	1.00000D+01	1.9D−15	1	2	0.10	0	1.00000D+01	2.5D−13	1	2	1.82	0
TQUARTIC	1000	1.35757D−24	1.1D−10	1	2	0.10	0	3.34915D−23	5.0D−10	1	2	0.78	0
TRIDIA	1000	2.37990D−26	2.3D−12	1	2	0.10	0	9.77167D−23	8.3D−11	1	2	2.13	0
VARDIM	1000	1.96586D−23	8.9D−09	80	81	11.32	0	1.96586D−23	8.9D−09	82	83	110.58	0
VAREIGVL	1000	2.55125D−26	8.9D−13	17	23	1.98	0	1.87621D−26	6.9D−14	8	9	14.07	0
WOODS	1000	1.00361D−26	1.6D−13	39	52	3.95	0	4.50820D−20	1.6D−10	38	52	22.77	0

TABLE 1

Two versions of Algorithm 2.1 applied to 59 unconstrained problems in the CUTEst collection in which both variants found equivalent solutions and stopped satisfying the same stopping criteria related to a small gradient (criteria 0 or 4) or related to achieving a target functional value (criteria 6 and 7).

Table 2 shows the details of the final iterates found by the two versions of Algorithm 2.1 on problems in Set 2. As a whole, the BPK-based version obtained smaller functional values than the spectral-based version in 11 problems whereas the spectral-based method got better funcional values in 7 problems. CPU time was smaller in the BPK-based version in 26 out of the 28 problems. Therefore, these experiments confirm that using non-expensive BPK-based Mixed Factorizations has practical advantages over the employment of QDQ^T factorizations in the context of Algorithm 2.1.

Problem	n	BPK-based MDM^T						Spectral-based MDM^T					
		$f(x^*)$	$\|g(x^*)\|_\infty$	#it	#f	Time	SC	$f(x^*)$	$\|g(x^*)\|_\infty$	#it	#f	Time	SC
ARGLINC	500	2.51125D+02	1.3D+01	4	5	9.17	7	2.51125D+02	1.4D−01	3	4	7.39	6
BOX	1000	-1.77371D+02	1.0D−13	14	26	1.49	4	-1.77371D+02	1.8D−13	11	18	9.45	0
BROWNAL	1000	1.00000D+00	1.9D−10	6	7	173.98	0	4.60178D−22	4.7D−10	5	6	138.34	0
BROYDN7D	1000	4.42450D+02	2.6D−11	42	52	4.46	0	3.61880D+02	4.6D−13	17	22	30.44	0
CHAINWOO	1000	7.42428D+01	2.8D−11	215	279	23.71	0	1.00000D+00	2.4D−12	52	80	123.05	0
COSINE	1000	-9.99000D+02	3.0D−10	5	6	0.52	0	-7.95559D+02	1.5D−02	6178	11770	4260.02	3
EG2	1000	-9.98947D+02	2.6D−07	16	353	1.84	9	-9.98947D+02	4.0D−09	5	7	3.88	0
EXTROSNB	1000	2.00727D−08	8.4D−05	1126	1698	119.81	2	2.07573D−08	8.2D−06	1120	1703	707.22	2
FLETCBV3	1000	-2.86635D+07	3.1D−02	4999	6501	533.40	3	-2.85189D+07	3.1D−02	4999	6501	3601.99	3
FLETCHBV	1000	-1.42629D+10	3.0D+06	4	5	0.46	6	-1.42612D+10	3.0D+06	4	5	6.24	6
FREUROTH	1000	1.21470D+05	1.3D−11	23	25	2.70	0	1.21470D+05	1.3D−06	28	553	53.76	9
INDEF	1000	-1.04532D+10	1.5D+01	35	37	3.78	7	-2.39122D+17	1.9D+02	4	5	7.02	7
MANCINO	1000	6.73178D−15	4.5D−04	31	36	848.99	5	6.75368D−15	4.5D−04	32	38	899.14	5
NCB20	1010	9.21210D+02	2.5D−14	92	119	9.67	0	9.27932D+02	3.8D−10	24	30	38.89	0
NONCVXU2	1000	2.31710D+03	2.8D−09	2020	2626	199.39	0	2.31894D+03	1.2D−13	180	240	327.17	0
NONCVXUN	1000	2.34905D+03	6.0D−11	2445	3171	242.67	0	2.32032D+03	2.7D−06	266	310	471.17	1
NONMSQRT	1024	8.99049D+01	3.5D−04	1897	410715	271.38	8	8.99049D+01	5.3D−03	2876	393328	2161.30	8
OSCIGRAD	1000	6.60719D−24	2.0D−08	11	15	1.03	5	1.34111D−23	2.9D−08	15	4051	32.70	5
PENALTY2	1000	3.95036D+82	1.3D+67	155	3423	19.19	8	1.01277D+83	4.7D+67	21	2253	37.45	8
PENALTY3	200	3.98575D+04	1.5D−07	647	1459	14.57	9	9.95470D−04	1.3D−07	26	286	1.16	9
SBRYBND	1000	3.66569D−27	1.4D−06	23	35	2.17	5	4.48886D−27	1.8D−06	28	115	59.16	5
SCOSINE	1000	-9.92319D+02	2.0D+13	87	32910	9.78	8	-9.96057D+02	1.1D+13	174	1321	184.22	9
SCURLY10	1000	-1.00316D+05	2.9D−08	64	73	6.39	9	-1.00316D+05	5.7D−08	147	182	227.01	9
SCURLY20	1000	-1.00316D+05	2.8D−08	64	73	6.20	9	-1.00316D+05	1.1D−07	69	82	102.79	9
SCURLY30	1000	-1.00316D+05	4.4D−08	64	73	6.44	9	-1.00316D+05	1.0D−07	71	84	106.64	9
SENSORS	1000	-2.10896D+05	2.6D−12	88	112	61.13	0	-2.09377D+05	4.9D−12	13	16	29.69	0
SINQUAD	1000	-2.94250D+05	1.0D−09	15	19	1.44	0	-2.94250D+05	4.4D−10	12	15	10.25	4
SPMSRTLS	1000	5.60850D−02	8.9D−11	47	61	4.51	0	4.34751D−16	2.0D−15	13	15	24.57	0

TABLE 2

Two versions of Algorithm 2.1 applied to 28 unconstrained problems in the CUTEst collection in which at least one of the following situations ocurred: (a) Non-equivalent solutions were found, (b) Stopping ocurred satisfying different stopping criteria, or (c) At least one of the versions finished satisfying a criterion different from 0, 4, 6, or 7.

5.2. Comparison against CurviH [15]. In this section, we perform a comparison between Algorithm 2.1 with the dense BPK-based mixed factorization and the method introduced in [15], named CurviH in the present work. At each iteration, CurviH performs a curvilinear search along the path defined by $x^k-(H_k+\sigma I)^{-1}g(x^k)$, $\sigma \geq 0$, stopping the search when an approximate minimizer of f along this path is reached. The matrix H_k is the true Hessian of f when k is a multiple of q and a quasi-Newton approximation otherwise. For computing the search path, the method employs the factorization $H_k = Q_k T_k Q_k^T$, where Q_k is orthonormal and T_k is tridiagonal. Therefore, the successive trial points are computed solving tridiagonal systems. When k is not a multiple of q, only the tridiagonal matrix is updated using a PSB (Powell-Symmetric-Broyden) formula whereas the orthonormal factor remains unmodified. We used the default value $q = 3$, as recommended in the documentation of CurviH, as well as the default values for all the other parameters of the method. Details of the performance of the method on the 87 problems of the CUTEst collection being considered can be found in Table 3. The method has three stopping criteria given by SC=0 meaning "convergence has been achieved"; SC=1 meaning "maximum number of function evaluations exceeded"; and SC=2 meaning "failure to converge". In the numerical experiments reported in Table 3, the criterion related to convergence was replaced by (60). The results obtained preserving the original stopping criterion related to convergence, given by $\max_{i=1,\dots,n} \left\{[g(x^k)]_i \max\{1, |x_i^k|\}\right\} \leq \varepsilon \max\{1, |f(x^k)|\}$, can be found in http://www.ime.usp.br/~egbirgin/.

Once again, the 87 problems will be divided into two sets to perform the comparison. In Set 1 we include the 57 problem in which both methods found equivalent solutions and stopped with a small sup-norm of the gradient (i.e. the final iterates satisfy (61) and (60)). Set 1 will be used to compare the efficiency of the methods. Set 2, composed by the remaining 30 problems will be used to evaluate their robustness. Efficiency will be evaluated with the help of performance profiles. See Figure 3. The figure shows that Algorithm 2.1 with the BPK-based mixed factorization is much more efficient than CurviH when the number of functional evaluations or the CPU time are used as a performance measurement. Analyzing the remaining 30 problems in Set 2, we can say that: (a) Algorithm 2.1 found a small gradient in 14 cases; while CurviH found a small gradient in 8 problems; (b) they both found equivalent functional values in 10 problems; (c) in 9 out of the 10 problems in which both methods found equivalent solutions, Algorithm 2.1 was faster; (d) Algorithm 2.1 found smaller values in 8 problems and CurviH found smaller values in another 8 problems; (e) in one case both methods identified that f is unbounded from below; and (f) in the remaining 3 cases CurviH reached the CPU time limit of one hour. Summing up, there is no meaningful differences in the robustness of the methods; while Algorithm 2.1 is much more efficient.

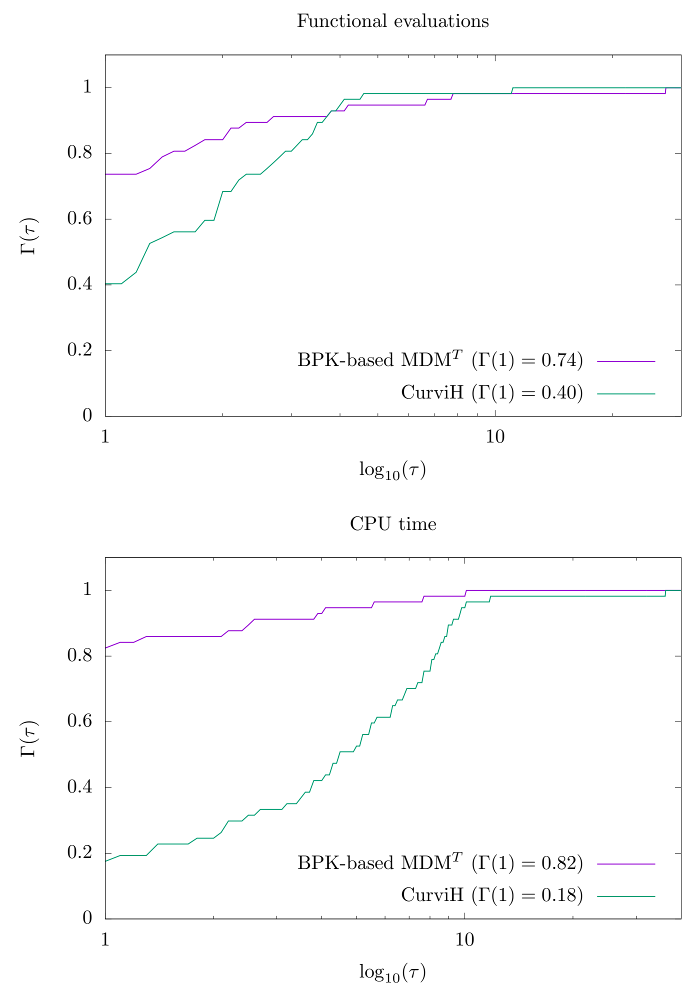

Fig. 3. *Performance profiles considering 57 problems in which Algorithm 2.1 with the BPK-based mixed factorization and CurviH found equivalent solutions and stopped satisfying the same stopping criteria related to a small gradient.*

5.3. Advantages of exploiting sparsity. We finish the numerical experiments by comparing the performance of Algorithm 2.1 with the dense and the sparse implementations of the BPK-based Mixed Factorization. On the one hand, we expect to illustrate the magnitude of the reduction of CPU time. On the other hand, we would like to check the influence of the MA57 pivoting strategy, that takes into account sparsity issues, on the overall stability of the method. It is worth noting that the numerical experiments that will be shown here were obtained using MA57 with the relative pivoting tolerance parameter $u = 0.5$ (its default value is 0.01), recommended for "problems requiring greater than average numerical care". (Preliminary numerical experiments with the default value for parameter u showed a big increase in the number of iterations and functional evaluations with respect to the results obtained with the dense BPK-based Mixed Factorization.) Details of the performance of the method on the 87 problems of the CUTEst collection being considered can be found in Table 4; while a comparison between Algorithm 2.1 with the dense and the sparse implementations of the BPK-based Mixed Factorization can be seen in Figure 4. The graphics in Figure 4 take into account the 61 problems in which the dense and the sparse version found equivalent solutions and stopped satisfying the same stopping criterion related to a small gradient or a target functional value. The graphic in the top shows that the number of iterations and functional evaluations is mostly the same in both versions, with a slight advantage of the dense version; while the graphic in the bottom shows that, in the considered set of problems, the sparse version is at least two orders of magnitude faster in most cases. Analyzing the remaining 26 problems, we can see that (a) they both found equivalent solutions in 6 problems; (b) they both reach the target functional value -10^{10} in a single problem (INDEF) that appears to be unbounded from below; (c) the dense version found a small functional value in 14 problems; and (d) the sparse version found a smaller functional value in 5 problems. Since the set of problems in which the dense version found a better functional value includes 6 problems of the same family (CURLY10, CURLY20, CURLY30, SCURLY10, SCURLY20, and SCURLY30) in which the difference appears in the fifth decimal place,we conclude that there is no meaningful difference in the robustness of the methods.

6. Conclusions. We introduced a new method for Unconstrained Optimization that, at each iteration, performs only one factorization, whose cost is similar to Cholesky decomposition, preserving $O(\varepsilon^{-3/2})$ complexity for first-order optimality and $O(\varepsilon^{-3})$ complexity for second-order optimality if the Hessian is Lipschitz continuous. Moreover, the introduced method convergences to first-order critical points under the only assumption of uniform continuity of first derivatives. The computation of trial points at each iteration does not need additional factorizations. The convergence and complexity theories cover a number of alternative algorithms. In particular the non-Lipschitzian results allows one to consider arbitrary Hessian approximations without connection with true Hessians at all. The Linear Algebra work per iteration is similar to the Linear Algebra work involved in a Newtonian line-search method, although the search direction changes each time a trial point is rejected, as in Trust-Region and Regularization algorithms.

We performed experiments in which, besides Mixed Factorizations based on the Bunch-Parlett-Kaufman decomposition, we used the analogous iteration scheme with Spectral Factorizations, which are significantly more expensive than BPK-based factorizations and can not exploit sparsity. The objective of these experiments was to test whether the stability differences between those factorizations could cause significative

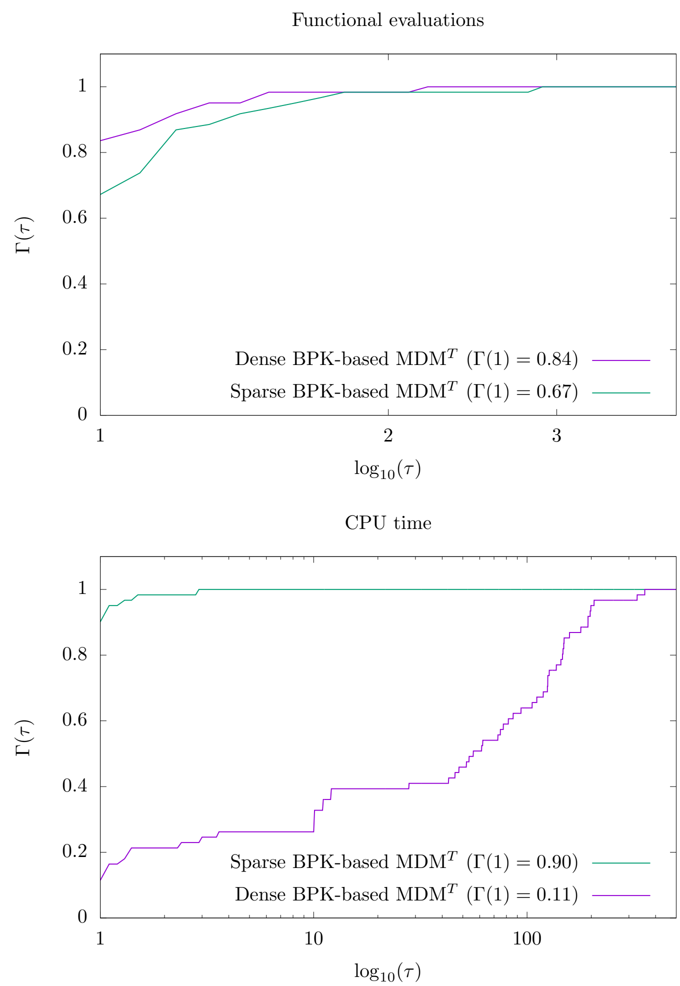

FIG. 4. *Performance profiles considering the 61 problems in which Algorithm 2.1 with the dense and the sparse implementations of the BPK-based mixed factorization found equivalent solutions and stopped satisfying the same stopping criteria related to a small gradient (criteria 0 or 4) or related to achieving a target functional value (criteria 6 and 7).*

differences in the performance of the algorithm. The results of these experiments have been conclusive: In terms of functional evaluations the algorithm with the Spectral Factorization is slightly better than the one with BPK-based Mixed Factorizations, but the second is much better than the first in terms of computer time. In terms of robustness, there are no meaningful differences between those algorithms.

Among other improvement paths for the new algorithm we may mention: (i) employment of non-monotone strategies along the lines of [28] and many other authors; (ii) using, at adequate iterations, the same factorization as in the previous one, instead of a new factorization; (iii) updating the Hessian approximation using quasi-Newton corrections; and (iv) for huge and very huge problems, use Hessian approximations with very simple structures (diagonal, tridiagonal, band). Moreover, considering the potential good behavior of the new method in cases where the Hessian does not exist, we have in mind the application to subproblems of Penalty and Augmented Lagrangian algorithms.

Acknowledgment: The authors would like to thank H. D. Scolnik for providing an updated version of the method introduced in [15]. We also would like to thank N. I. M. Gould for point out the work [23] and also for point out the usage of subroutine MA57_get_factors in the sparse implementation of the BPK-based Mixed Factorization.

REFERENCES

[1] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK Users' Guide*, Society for Industrial and Applied Mathematics, Philadelphia, PA, third ed., 1999, doi:10.1137/1.9780898719604.

[2] E. Bergou, Y. Diouane, and S. Gratton, *A line-search algorithm inspired by the adaptive cubic regularization framework, with a worst-case complexity*, Computational Optimization and Applications, 68 (2017), pp. 533–554, doi:10.1007/s10589-017-9929-2.

[3] E. Bergou, Y. Diouane, and S. Gratton, *A line-search algorithm inspired by the adaptive cubic regularization framework, with a worst-case complexity* $\wr(\varepsilon^{-3/2})$, tech. report, 2017. Available at http://www.optimization-online.org/DB-HTML/2017/06/6083.html.

[4] E. G. Birgin and J. M. Gentil, *Evaluating bound-constrained minimization software*, Computational Optimization and Applications, 53 (2012), pp. 347–373, doi:10.1007/s10589-012-9466-y.

[5] E. G. Birgin and J. M. Martínez, *Practical Augmented Lagrangian Methods for Constrained Optimization*, vol. 10 of Fundamentals of Algorithms, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2014, doi:10.1137/1.9781611973365.

[6] E. G. Birgin and J. M. Martínez, *The use of quadratic regularization with a cubic descent condition for unconstrained optimization*, SIAM Journal on Optimization, 27 (2017), pp. 1049–1074, doi:10.1137/16M110280X.

[7] E. G. Birgin and J. M. Martínez, *On regularization and active-set methods with complexity for constrained optimization*, SIAM Journal on Optimization, (2018). To appear.

[8] C. Cartis, N. I. M. Gould, and P. L. Toint, *On the complexity of steepest descent, newton's and regularized newton's methods for nonconvex unconstrained optimization*, SIAM Journal on Optimization, 20 (2010), pp. 2833–2852, doi:10.1137/090774100.

[9] C. Cartis, N. I. M. Gould, and P. L. Toint, *Adaptive cubic regularization methods for unconstrained optimization. Part I: motivation, convergence and numerical results*, Mathematical Programming, 127 (2011), pp. 245–295, doi:10.1007/s10107-009-0286-5.

[10] C. Cartis, N. I. M. Gould, and P. L. Toint, *Adaptive cubic regularization methods for unconstrained optimization. Part II: worst-case function and derivative complexity*, Mathematical Programming, 130 (2011), pp. 295–319, doi:10.1007/s10107-009-0337-y.

[11] C. Cartis, N. I. M. Gould, and P. L. Toint, *Universal regularization methods – varying the power, the smoothness and the accuracy*, Optimization Methods and Software, (2017). To appear.

[12] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Trust Region Methods*, Society for Industral and Applied Mathematics, Philadelphia, PA, 2000, doi:10.1137/1.9780898719857.

[13] R. Courant and F. John, *Introduction to Calculus and Analysis*, Wiley, New York, 1974.

[14] F. E. Curtis, D. P. Robinson, and M. Samadi, *A trust-region algorithm with a worst-case iteration complexity of* $O(\varepsilon^{-3/2})$, Mathematical Programming, 162 (2017), pp. 1–32, doi:10.1007/s10107-016-1026-2.

[15] J. E. Dennis Jr., N. Echebest, M. T. Guardarucci, J. M. Martínez, H. D. Scolnik, and C. Vacchino, *A curvilinear search using tridiagonal secant updates for unconstrained optimization*, SIAM Journal on Optimization, 1 (1991), pp. 333–357, doi:10.1137/0801022.

[16] J. E. Dennis Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Society for Industral and Applied Mathematics, Philadelphia, PA, 1996, doi:10.1137/1.9781611971200.

[17] E. D. Dolan and J. J. Moré, *Benchmarking optimization software with performance profiles*, Mathematical Programming, 91 (2002), pp. 201–213, doi:10.1007/s101070100263.

[18] K. Duraisamy, *Data-enabled, physics-constrained predictive modeling of complex systems*, SIAM News, 50 (July/August 2017), p. 12.

[19] J. P. Dussault, *ARC_q: a new adaptive regularization by cubics*, Optimization Methods and Software, (2017), doi:10.1080/10556788.2017.1322080. To appear.

[20] M. A. Figueiredo, R. D. Nowak, and S. J. Wright, *Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems*, IEEE Journal of Selected Topics in Signal Processing, 1 (2007), pp. 586–597, doi:10.1109/JSTSP.2007.910281.

[21] D. M. Gay, *Computing optimal locally constrained steps*, SIAM Journal on Scientific and Statistical Computing, 2 (1981), pp. 186–197, doi:10.1137/0902016.

[22] G. H. Golub and C. F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore and London, 3rd. ed., 1996.

[23] N. I. M. Gould and J. Nocedal, *The modified absolute-value factorization norm for trust-region minimization*, in High Performance Algorithms and Software in Nonlinear Optimization, R. De Leone, A. Murli, P. M. Pardalos, and G. Toraldo, eds., Kluwer Academic Publishers, 1998, pp. 225–241, doi:10.1007/978-1-4613-3279-4_15.

[24] N. I. M. Gould, D. Orban, and P. L. Toint, *CUTEst: a constrained and unconstrained testing environment with safe threads for mathematical optimization*, Computational Optimization and Applications, 60 (2014), pp. 545–557, doi:10.1007/s10589-014-9687-3.

[25] N. I. M. Gould, M. Porcelli, and P. L. Toint, *Updating the regularization parameter in the adaptive cubic regularization algorithm*, Computational Optimization and Applications, 53 (2012), pp. 1–22, doi:10.1007/s10589-011-9446-7.

[26] G. N. Grapiglia and Y. Nesterov, *Regularized newton methods for minimizing functions with Hölder continuous hessians*, SIAM Journal on Optimization, 27 (2017), pp. 478–506, doi:10.1137/16M1087801.

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

[28] L. Grippo, F. Lampariello, and S. Lucidi, *A nonmonotone line search technique for newtons method*, SIAM Journal on Numerical Analysis, 23 (1986), pp. 707–716, doi:10.1137/0723046.

[29] D. Gunopulos, T. Hofmann, D. Malerba, and M. Vazirgiannis, eds., *Machine Learning and Knowledge Discovery in Databases, Part III*, vol. 6913 of Lecture Notes in Artificial Intelligence, Springer-Verlag Berlin Heidelberg, 2011, doi:10.1007/978-3-642-23808-6.

[30] E. W. Karas, S. A. Santos, and B. F. Svaiter, *Algebraic rules for quadratic regularization of Newton's method*, Computational Optimization and Applications, 60 (2015), pp. 343–376, doi:10.1007/s10589-014-9671-y.

[31] S. Lu, Z. Wei, and L. Li, *A trust region algorithm with adaptive cubic regularization methods for nonsmooth convex minimization*, Computational Optimization and Applications, 51 (2012), pp. 551–573, doi:10.1007/s10589-010-9363-1.

[32] J. M. Martínez, *On high-order model regularization for constrained optimization*, SIAM Journal on Optimization, 27 (2017), pp. 2447–2458, doi:10.1137/17M1115472.

[33] J. M. Martínez and M. Raydan, *Cubic-regularization counterpart of a variable-norm trust-region method for unconstrained minimization*, Journal of Global Optimization, 68 (2017), pp. 367–385, doi:10.1007/s10898-016-0475-8.

[34] J. J. Moré and D. C. Sorensen, *Computing a trust region step*, SIAM Journal on Scientific and Statistical Computing, 4 (1983), pp. 553–572, doi:10.1137/0904038.

[35] Y. Nesterov and B. T. Polyak, *Cubic regularization of Newton's method and its global performance*, Mathematical Programming, 108 (2006), pp. 177–205, doi:10.1007/s10107-006-0706-8.

[36] C. W. Royer and S. J. Wright, *Complexity analysis of second-order line-search algorithms for smooth nonconvex optimization*, tech. report, 2017. arXiv:1706.03131v2 [math.OC].

[37] Z.-H. Wang and Y.-X. Yuan, *A subspace implementation of quasi-newton trust region methods for unconstrained optimization*, Numerische Mathematik, 104 (2006), pp. 241–269, doi:10.1007/s00211-006-0021-6.

[38] *HSL. A collection of fortran codes for large scale scientific computation*, http://www.hsl.rl.ac.uk/.

Problem	n	$f(x^*)$	$\|g(x^*)\|_\infty$	#it	#f	#g	#H	Time	SC
ARGLINA	500	5.0000000000D+02	8.6D-14	1	2	2	1	2.97	0
ARGLINC	500	2.5112518728D+02	1.9D-02	29	192	204	11	26.47	2
ARWHEAD	1000	0.0000000000D+00	1.2D-13	13	14	14	5	5.11	0
BDQRTIC	1000	3.9838179506D+03	3.9D-09	19	22	22	7	6.98	0
BOX	1000	-1.7737059413D+02	1.8D-11	8	15	15	3	3.19	0
BROWNAL	1000	8.2679716579D-22	1.8D-09	7	8	8	3	88.29	0
BROYDN7D	1000	3.4952238029D+02	5.9D-10	67	81	81	23	25.26	0
BRYBND	1000	1.3870201692D-26	5.7D-12	20	24	24	7	8.22	0
CHAINWOO	1000	1.7790185089D+02	3.5D-13	1078	1326	1326	360	495.97	0
COSINE	1000	-9.9900000000D+02	2.8D-10	10	16	16	4	4.41	0
CRAGGLVY	1000	3.3845287719D+02	8.2D-08	38	114	114	14	15.19	2
CURLY10	1000	-1.0031629024D+05	2.4D-11	20	37	37	7	8.08	0
CURLY20	1000	-1.0031629024D+05	5.8D-10	21	38	38	7	8.12	0
CURLY30	1000	-1.0031629024D+05	1.4D-10	23	40	40	8	9.18	0
DIXMAANA	900	1.0000000000D+00	3.0D-12	9	10	10	3	2.49	0
DIXMAANB	900	1.0000000000D+00	1.5D-15	10	12	12	4	3.27	0
DIXMAANC	900	1.0000000000D+00	2.0D-10	11	12	12	4	3.22	0
DIXMAAND	900	1.0000000000D+00	2.6D-11	12	13	13	4	3.12	0
DIXMAANE	900	1.0000000000D+00	1.3D-12	14	18	18	5	3.99	0
DIXMAANF	900	1.0000000000D+00	1.6D-09	26	38	40	9	7.38	0
DIXMAANG	900	1.0000000000D+00	1.2D-11	31	36	39	11	9.28	0
DIXMAANH	900	1.0000000000D+00	1.6D-12	28	32	33	10	7.91	0
DIXMAANI	900	1.0000000000D+00	5.5D-12	33	65	72	11	8.62	0
DIXMAANJ	900	1.0000000000D+00	4.9D-10	55	107	117	19	16.01	0
DIXMAANK	900	1.0000000000D+00	2.0D-10	58	112	125	20	17.12	0
DIXMAANL	900	1.0000000000D+00	7.4D-12	73	135	158	25	21.76	0
DIXON3DQ	1000	1.8436964407D-10	1.9D-09	10	22	23	4	4.34	0
DQDRTIC	1000	0.0000000000D+00	0.0D+00	1	2	2	1	0.02	0
DQRTIC	1000	2.6040660838D-09	8.2D-09	62	122	122	21	0.93	0
EDENSCH	1000	6.0032845920D+03	9.2D-10	14	16	16	5	5.52	0
EG2	1000	-9.9894739330D+02	2.4D-08	87	581	632	31	31.54	2
EIGENALS	420	4.0220012518D-19	1.6D-09	110	150	151	37	3.67	0
EIGENBLS	420	6.2639758091D-19	1.5D-09	533	786	786	178	16.85	0
EIGENCLS	462	4.4624982881D-18	2.8D-10	317	408	408	106	12.56	0
ENGVAL1	1000	1.1081947188D+03	1.5D-11	10	11	11	4	4.25	0
EXTROSNB	1000	1.3693496282D-09	9.7D-09	7994	12030	17351	2665	152.61	0
FLETCBV2	1000	-5.0142903408D-01	8.4D-09	1	2	2	1	0.98	0
FLETCHCR	1000	1.2221291387D-20	3.8D-09	2489	5851	7500	831	853.32	0
FMINSRF2	961	1.0000000016D+00	6.9D-09	79	141	141	27	25.00	0
FMINSURF	961	9.9999999900D-01	9.3D-10	61	95	95	21	19.70	0
FREUROTH	1000	1.2146971011D+05	1.3D-08	12	30	30	5	5.06	2
GENROSE	1000	1.0000000000D+00	1.7D-09	949	2211	2833	317	316.95	0
HILBERTB	500	1.4038161241D-25	8.0D-13	1	2	2	1	0.19	0
INDEF	1000	-1.4864895857D+16	5.9D+01	152	830	859	55	57.65	2
LIARWHD	1000	5.6881470181D-19	6.2D-09	28	29	30	10	10.02	0
MANCINO	1000	6.2579858179D-15	3.9D-04	76	366	366	26	794.77	2
MODBEALE	1000	4.4132104884D-20	3.8D-10	14	16	16	5	5.19	0
MOREBV	1000	1.0844344982D-09	2.5D-09	3	4	4	1	1.07	0
MSQRTALS	1024	3.6170305416D-14	4.6D-09	57	68	68	19	21.96	0
MSQRTBLS	1024	1.8703730634D-18	2.7D-09	52	61	61	18	20.84	0
NCB20	1010	9.1986070003D+02	4.0D-09	165	224	224	56	66.74	0
NCB20B	1000	1.6760112121D+03	8.9D-09	38	53	53	13	13.38	0
NONCVXU2	1000	2.3180574688D+03	8.9D-09	603	880	880	201	199.04	0
NONCVXUN	1000	2.3266274704D+03	6.7D-09	1730	8636	8636	578	619.65	0
NONDIA	1000	3.5722395251D-22	1.9D-11	13	15	17	5	5.16	0
NONDQUAR	1000	2.6147062947D-08	9.6D-09	198	272	272	66	76.76	0
NONMSQRT	1024	8.9906813008D+01	4.5D-07	1465	7023	7033	493	748.04	2
OSCIGRAD	1000	7.7276112636D+04	1.6D-08	21	50	63	8	9.05	2
OSCIPATH	500	9.9996666552D-01	2.1D-08	4	46	48	2	0.28	2
PENALTY1	1000	9.6861754324D-03	2.3D-11	73	97	97	25	33.56	0
PENALTY2	1000	1.4463988820D+83	2.1D+38	1	5	5	1	1.01	2
PENALTY3	200	9.9713897705D-04	3.5D-06	25	160	160	9	0.62	2
POWELLSG	1000	1.0972521352D-10	6.1D-09	35	36	36	12	10.53	0
POWER	1000	4.7635080787D-14	5.5D-09	85	87	92	29	29.08	0
QUARTC	1000	2.6040660838D-09	8.2D-09	62	122	122	21	0.81	0
SBRYBND	1000	6.6415972259D-27	1.9D-06	96	300	317	34	57.97	2
SCHMVETT	1000	-2.9940000000D+03	4.4D-15	4	5	5	2	2.06	0
SCOSINE	1000	-9.9900000000D+02	1.1D-07	172	204	240	59	62.86	2
SCURLY10	1000	-1.0031629024D+05	4.3D-08	174	262	262	59	88.80	2
SCURLY20	1000	-1.0031629024D+05	1.2D-07	126	200	200	43	54.29	2
SCURLY30	1000	-1.0031629024D+05	1.8D-07	105	171	171	36	43.63	2
SENSORS	1000	-2.0054165625D+05	1.7D-08	60	70	70	21	42.35	2
SINQUAD	1000	-2.9425049403D+05	2.1D-05	19	176	176	7	7.48	2
SPARSINE	1000	7.3116629848D-16	5.7D-10	47	59	59	16	16.16	0
SPARSQUR	1000	5.4540486806D-11	8.8D-09	28	29	29	10	10.04	0
SPMSRTLS	1000	4.3475087793D-16	4.6D-13	28	35	35	10	10.03	0
SROSENBR	1000	8.1086291267D-21	7.4D-10	11	13	13	4	2.64	0
TESTQUAD	1000	0.0000000000D+00	0.0D+00	1	2	2	1	0.01	0
TOINTGSS	1000	1.0000000020D+01	2.8D-15	1	2	2	1	0.96	0
TQUARTIC	1000	9.3332139869D-24	1.7D-10	1	2	2	1	1.00	0
TRIDIA	1000	2.0172800372D-26	2.2D-12	1	2	2	1	0.97	0
VARDIM	1000	2.3897900421D-23	9.8D-09	52	62	62	18	27.78	0
VAREIGVL	1000	1.7085711434D-20	4.6D-11	10	11	11	4	4.15	0
WOODS	1000	2.4085500931D-22	4.2D-10	169	237	267	57	46.36	0

TABLE 3

Details of the application of CurviH to the 87 unconstrained problems in the CUTEst collection. Only problems FLETCBV3, FLETCHBV, *and* GENHUMPS *were excluded from the table since the method exceeded a CPU time limit of one hour without satisfying any of the stopping criteria.*

Problem	n	$f(x^*)$	$\|g(x^*)\|_\infty$	#it	#f	Time	SC
ARGLINA	500	5.0000000000D+02	1.6D−13	1	2	2.80	0
ARGLINC	500	2.5112518716D+02	6.8D+00	4	5	9.09	7
ARWHEAD	1000	0.0000000000D+00	1.2D−12	6	7	0.00	0
BDQRTIC	1000	3.9838179506D+03	2.2D−13	10	11	0.01	0
BOX	1000	-1.7737059413D+02	9.2D−10	12	21	0.01	0
BROWNAL	1000	4.3243991418D−19	4.2D−08	14	15	373.97	5
BROYDN7D	1000	4.2360571241D+02	5.2D−09	46	56	0.05	0
BRYBND	1000	3.1652479269D−23	2.3D−11	11	14	0.02	0
CHAINWOO	1000	7.5046422953D+01	3.0D−13	217	286	0.15	0
COSINE	1000	-9.9900000000D+02	3.1D−12	6	7	0.00	0
CRAGGLVY	1000	3.3642314787D+02	5.4D−15	15	16	0.01	4
CURLY10	1000	-1.0031376042D+05	1.5D−12	422	548	0.56	0
CURLY20	1000	-1.0030047885D+05	6.9D−12	490	637	1.56	0
CURLY30	1000	-1.0028340256D+05	1.3D−11	330	429	1.93	0
DIXMAANA	900	1.0000000000D+00	0.0D+00	6	8	0.01	0
DIXMAANB	900	1.0000000000D+00	1.5D−10	28	34	0.02	0
DIXMAANC	900	1.0000000000D+00	1.3D−23	37	44	0.03	0
DIXMAAND	900	1.0000000000D+00	2.9D−15	30	34	0.02	0
DIXMAANE	900	1.0000000000D+00	4.4D−12	9	10	0.01	0
DIXMAANF	900	1.0000000000D+00	7.7D−21	27	30	0.02	0
DIXMAANG	900	1.0000000000D+00	5.7D−10	39	46	0.03	0
DIXMAANH	900	1.0000000000D+00	3.0D−19	35	40	0.02	0
DIXMAANI	900	1.0000000000D+00	4.8D−17	11	12	0.01	0
DIXMAANJ	900	1.0000000000D+00	6.0D−11	38	43	0.03	0
DIXMAANK	900	1.0000000000D+00	3.1D−09	34	38	0.02	0
DIXMAANL	900	1.0000000000D+00	8.6D−09	33	37	0.02	0
DIXON3DQ	1000	0.0000000000D+00	0.0D+00	1	2	0.00	0
DQDRTIC	1000	0.0000000000D+00	0.0D+00	1	2	0.00	0
DQRTIC	1000	2.2354180180D−10	4.3D−09	34	35	0.01	0
EDENSCH	1000	6.0032845920D+03	1.5D−10	12	13	0.01	0
EG2	1000	-9.9894739330D+02	2.7D−07	16	317	0.23	9
EIGENALS	420	3.6996340677D−18	7.7D−09	466	643	9.21	0
EIGENBLS	420	9.3259138939D−16	7.6D−09	223	327	4.00	0
EIGENCLS	462	3.8834451010D−22	2.1D−10	241	330	5.23	0
ENGVAL1	1000	1.1081947188D+03	1.3D−12	8	9	0.01	0
EXTROSNB	1000	2.0419363501D−08	4.1D−05	1125	1659	0.55	2
FLETCBV2	1000	-5.0142903408D−01	8.4D−09	1	2	0.00	0
FLETCBV3	1000	-2.8519222615D+07	3.1D−02	4999	6501	3.27	3
FLETCHBV	1000	-1.4262893412D+10	3.0D+06	4	5	0.00	6
FLETCHCR	1000	1.6636144649D−25	8.9D−13	1470	2021	0.84	0
FMINSRF2	961	9.9999999900D−01	5.4D−15	51	98	0.10	0
FMINSURF	961	9.9999999900D−01	7.8D−12	98	167	10.68	0
FREUROTH	1000	1.2146971011D+05	7.1D−05	35	391	0.05	9
GENHUMPS	1000	1.6915335609D−18	5.0D−10	3080	4016	2.31	0
GENROSE	1000	1.0000000000D+00	6.7D−13	706	966	0.39	0
HILBERTB	500	3.8827140132D−26	1.0D−13	1	2	0.05	0
INDEF	1000	-5.3093486065D+18	2.6D+02	11	12	0.01	7
LIARWHD	1000	9.4433750103D−26	2.3D−11	12	13	0.01	0
MANCINO	1000	6.7326101684D−15	4.4D−04	23	27	620.35	5
MODBEALE	1000	1.1308969338D−21	2.7D−11	19	20	0.02	0
MOREBV	1000	7.3289563806D−13	4.7D−11	1	2	0.00	0
MSQRTALS	1024	1.5355565738D−24	2.4D−12	420	593	62.35	0
MSQRTBLS	1024	3.6774154581D−23	5.7D−12	519	711	75.01	0
NCB20	1010	9.2526099787D+02	2.8D−09	116	151	0.48	0
NCB20B	1000	1.6760112150D+03	7.6D−09	22	36	0.10	0
NONCVXU2	1000	2.3175732714D+03	1.8D−11	3088	4016	40.62	0
NONCVXUN	1000	2.3277730975D+03	4.9D−11	3709	4823	8.35	0
NONDIA	1000	1.7872679188D−26	4.2D−11	6	7	0.00	0
NONDQUAR	1000	3.1849289202D−13	9.5D−09	22	23	0.01	0
NONMSQRT	1024	8.9904972164D+01	3.6D−05	1216	2274	9.02	9
OSCIGRAD	1000	1.4643716141D−23	2.8D−08	13	23	0.01	5
OSCIPATH	500	9.9996666552D−01	1.3D−12	2	3	0.00	0
PENALTY1	1000	9.6861754324D−03	6.2D−11	41	51	4.64	0
PENALTY2	1000	1.0127718562D+83	8.6D+66	26	163	2.97	9
PENALTY3	200	9.9427795410D−04	1.3D−07	69	680	1.82	9
POWELLSG	1000	3.2920404304D−10	8.7D−09	20	21	0.01	0
POWER	1000	1.4281110130D−12	7.4D−09	33	34	3.69	0
QUARTC	1000	2.2354180180D−10	4.3D−09	34	35	0.01	0
SBRYBND	1000	4.6535729448D−27	1.4D−06	15	43	0.02	5
SCHMVETT	1000	-2.9940000000D+03	1.9D−13	3	4	0.00	0
SCOSINE	1000	-9.9187975204D+02	6.5D+13	225	1084	0.20	9
SCURLY10	1000	-1.0002915545D+05	1.2D−07	451	577	0.57	9
SCURLY20	1000	-1.0001207915D+05	3.0D−07	341	433	1.04	9
SCURLY30	1000	-1.0002346335D+05	5.3D−07	295	374	1.63	9
SENSORS	1000	-2.1093750000D+05	6.2D−12	51	67	34.28	0
SINQUAD	1000	-2.9425049403D+05	7.3D−10	15	34	0.01	0
SPARSINE	1000	8.0457143349D−20	8.0D−10	1664	2158	53.43	0
SPARSQUR	1000	4.4922696396D−11	7.6D−09	22	23	0.65	0
SPMSRTLS	1000	2.8806449220D−01	1.5D−15	77	99	0.07	0
SROSENBR	1000	3.8917905294D−18	1.4D−09	8	18	0.00	0
TESTQUAD	1000	8.6876741402D−27	8.9D−11	1	2	0.00	0
TOINTGSS	1000	1.0000000020D+01	2.8D−15	1	2	0.00	0
TQUARTIC	1000	9.0443341587D−25	1.9D−12	1	2	0.00	0
TRIDIA	1000	6.2328146641D−27	2.1D−12	1	2	0.00	0
VARDIM	1000	1.9658584423D−23	8.9D−09	83	84	11.56	0
VAREIGVL	1000	2.2059909687D−23	3.4D−11	48	66	5.56	0
WOODS	1000	3.0845693989D−28	4.4D−14	41	61	0.02	0

TABLE 4

Details of the application of Algorithm 2.1 with the sparse implementation of the BPK-based Mixed Factorization applied to the 87 *unconstrained problems in the CUTEst collection under consideration.*