SIAM J. OPTIM.
Vol. 29, No. 2, pp. 1048–1075

A STRUCTURED QUASI-NEWTON ALGORITHM FOR OPTIMIZING WITH INCOMPLETE HESSIAN INFORMATION*

COSMIN G. PETRA†, NAIYUAN CHIANG‡, AND MIHAI ANITESCU§

Abstract. We present a structured quasi-Newton algorithm for unconstrained optimization problems that have unavailable second-order derivatives or Hessian terms. We provide a formal derivation of the well-known Broyden–Fletcher–Goldfarb–Shanno (BFGS) secant update formula that approximates only the missing Hessian terms, and we propose a linesearch quasi-Newton algorithm based on a modification of Wolfe conditions that converges to first-order optimality conditions. We also analyze the local convergence properties of the structured BFGS algorithm and show that it achieves superlinear convergence under the standard assumptions used by quasi-Newton methods using secant updates. We provide a thorough study of the practical performance of the algorithm on the CUTEr suite of test problems and show that our structured BFGS-based quasi-Newton algorithm outperforms the unstructured counterpart(s).

Key words. structured quasi-Newton, secant approximation, BFGS, incomplete Hessian

AMS subject classifications. 90C53, 90C30, 90C06

DOI. 10.1137/18M1167942

1. Introduction and motivation.

We propose an algorithm for solving unconstrained optimization problems of the form

$$\min_{x\in\mathbb{R}^n} f(x), \quad f(x) := k(x) + u(x) \tag{1}$$

for which the Hessian of $k(x)$, $K(x) := \nabla^2 k(x)$, is available, but the Hessian of $u(x)$, $U(x) := \nabla^2 u(x)$, is not available or is expensive to evaluate. Such a situation occurs, for example, when f comprises sums of functions that are either algebraically described (and derivatives can be evaluated) or simulated (and second-order derivatives are costly to compute), as is the case of adjoint-based computations. A notable example is security-constrained operations in power grids, when the security criteria are expressed by constraints on dynamic contingencies [19]. When these transient behavior requirements are expressed by penalty or Lagrangian approaches, the optimization

*Received by the editors February 2, 2018; accepted for publication (in revised form) January 16, 2019; published electronically April 11, 2019. This paper has been created by UChicago Argonne, LLC, Operator of Argonne National Laboratory ("Argonne"). Argonne, a U.S. Department of Energy Office of Science laboratory, is operated under contract DE-AC02-06CH11357. The U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes. Copyright is owned by SIAM to the extent not limited by these rights.

http://www.siam.org/journals/siopt/29-2/M116794.html

Funding: This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. This material was also based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research (ASCR), under contract DE-AC02-06CH11357. The first author also acknowledges partial support from the LDRD Program of Lawrence Livermore National Laboratory under projects 16-ERD-025 and 17-SI-005. The third author acknowledges partial NSF funding through awards FP061151-01-PR and CNS-1545046.

†Corresponding author. Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA 94550 (petra1@llnl.gov).

‡United Technologies Research Center, East Hartford, CT 06118 (chiangn@utrc.utc.com).

§Mathematics and Computer Science Division, Argonne National Laboratory, Lemont, IL 60439 (anitescu@mcs.anl.gov).

problem has a logical two-stage structure [1]. Evaluation of the contribution of the second-stage, transient components involves obtaining the sensitivity information of a differential algebraic equation, for which the forward simulation alone is much costlier than the evaluation of the terms depending only on the first-stage variables. Another example is nonlinear least-squares problems, for which the Hessian is the sum of a term containing only first-order Jacobian information, which is available, and a term containing Hessian information, which is costly to compute and evaluate. On the other hand, we work under the assumption that the gradient $\nabla f(x)$ is available.

The goal of this work is to develop algorithms of a quasi-Newton flavor that are capable of combining the existing Hessian information and secant updates for the unavailable part of the Hessian. The motivation behind this work is that one can reasonably expect that such algorithms will perform better than general quasi-Newton counterparts that do not consider the structure in the Hessian.

Our approach is in the spirit of quasi-Newton secant methods equipped with a linesearch mechanism. Such iterative numerical procedures produce a sequence of iterates $\{x_i\}_{i\geq 0}$ of the form $x_{i+1} = x_i + \alpha_i p_i$, where the quasi-Newton search direction p_i is given by $p_i = B_i^{-1}\nabla f(x_i)$, with B_i being an approximation of the Hessian $\nabla^2 f(x_i)$. The steplength α_i is found by an appropriate search along the direction p_i that ensures the convergence to a stationary point of the gradient, for example, a linesearch based on Wolfe conditions [24].

The secant Hessian approximation B_i is updated during the linesearch algorithm based on a closed-form formula $\mathbb{B}$, namely,

$$B_{i+1} = \mathbb{B}(s_i, y_i, B_i),$$

where s_i and y_i are the changes in the variables and gradient, respectively, and are given by

$$s_i = x_{i+1} - x_i \quad \text{and} \quad y_i = \nabla f(x_{i+1}) - \nabla f(x_i).$$

The formula $\mathbb{B}$ ensures that B_{i+1} is symmetric and satisfies the secant equation

$$B_{i+1}s_i = y_i. \tag{2}$$

In order to derive the update formula $\mathbb{B}$ and to uniquely determine B_{i+1}, additional conditions need to be posed on B_{i+1} [20]. These conditions take the form of the so-called proximity criterion, which requires that B_{i+1} or its inverse, B_{i+1}^{-1}, be the closest approximation in some norm to B_i or B_i^{-1}, respectively. Weighted Frobenius matrix norms have been used to define proximity [20]; they are a convenient choice since they allow an analytical expression for $\mathbb{B}$. The Broyden–Fletcher–Goldfarb–Shanno (BFGS) formulas are obtained by imposing the proximity criterion on the inverse. The BFGS formula for the inverse $H_{i+1} = B_{i+1}^{-1}$ is obtained as the solution to the variational characterization problem

$$\begin{aligned} H_{i+1}^{BFGS} = \operatorname*{argmin}_{H} \quad & \|H - H_i\|_W \\ \text{s.t.} \quad & H_i y_i = s_i, \\ & H = H^T, \end{aligned} \tag{3}$$

as the analytical expression (see [15, Corollary 2.3])

$$H_{i+1}^{BFGS} = (I - \gamma_i y_i s_i^T) H_i (I - \gamma_i s_i y_i^T) + \gamma_i y_i y_i^T. \tag{4}$$

Here the weight matrix W can be chosen as any symmetric positive definite matrix satisfying the secant equation $Wy_i = s_i$. The weighted Frobenius norm $\|A\|_W$ is the Frobenius norm of $W^{1/2}AW^{1/2}$. We observe that the choice of W does not influence the inverse formula (4). Also, we use the notation $\gamma_i = 1/(s_i^T y_i)$. The BFGS update for the Hessian can be obtained by using the Sherman–Morrison–Woodbury formula, and is given by

$$B_{i+1}^{BFGS} = B_i - \frac{B_i s_i s_i^T B_i}{s_i^T B_i s_i} + \frac{y_i y_i^T}{s_i^T y_i}. \tag{5}$$

When the proximity criterion, together with the secant equation and symmetry conditions, is enforced for the matrix B_i, one can similarly obtain the Davidon–Fletcher–Powell (DFP) secant update as the solution to

$$\begin{aligned} B_{i+1}^{DFP} = \underset{B}{\operatorname{argmin}} \quad & \|B - B_i\|_W \\ \text{s.t.} \quad & B_i s_i = y_i, \\ & B = B^T, \end{aligned} \tag{6}$$

in the form of the analytical expression $B_{i+1}^{DFP} = (I - \gamma_i y_i s_i^T) B_i (I - \gamma_i s_i y_i^T) + \gamma_i y_i y_i^T$. The weight matrix W is chosen to satisfy $Ws_i = y_i$ and to be positive definite.

In this paper we first derive in section 2 *structured* counterparts to the BFGS formulas (4) and (5). The formal derivation of the structured formulas uses the apparatus of Güler, Gurtuna, and Shevchenko [15] to derive two structured BFGS formulas coming from two different perspectives, least-squares and trace-determinant variational characterizations, on BFGS updates.

We then investigate in section 3 linesearch globalization strategies. We find that for one of the structured BFGS formulas, one can always find a subset of the Wolfe points that ensures the positive definiteness of the structured update; furthermore, such Wolfe points can be identified at low computational cost. For the other structured BFGS update, we suggest using a standard inertia perturbation (by a multiple of identity) of the Hessian approximation. As a result, both approaches are globally convergent to a first-order stationary point, as we discuss in section 3.2.

Structured BFGS updates have been investigated before, for example, for nonlinear least-squares problems [9, 6, 17, 2] and for Lagrangian functions in sequential quadratic programming (SQP) methods for constrained optimization [22, 16]. To the best of our knowledge, however, no attempts have been made before to provide a globalization strategy in conjunction with this class of BFGS updates. Furthermore, we show that our formal derivation of structured secant formulas (see discussions of the least-squares and trace-determinant variational characterizations for (9) and (10)) is a rigorous tool that requires no intuition or any other type of empirical input in finding structured secant formula updates that are optimal with respect to various variational characterizations.

In section 3.3 we show the local superlinear convergence of the two structured BFGS formulas. We work under the standard assumptions used in analyzing the local convergence rates of certain quasi-Newton secant methods; for example, see the review paper of Dennis and Moré [7]. Our analysis is also standard in the sense that the superlinear convergence is proved by using a characterization (e.g., bounded deterioration property) of superlinear convergence due to Dennis and Moré (see [7, 3]) and it employs intermediary results and techniques from [3, 14, 6].

We also provide a thorough performance evaluation of various computational strategies with the structured BFGS formulas proposed in this paper. This includes a comparison with the unstructured BFGS formula that reveals that the structured quasi-Newton algorithms outperform the unstructured counterparts and confirms that our structured formulas are capable of using the existing, incomplete Hessian information to improve numerical performance.

Notation. Throughout this paper we denote vectors by lowercase Latin letters and matrices by uppercase Latin letters. The symbol $\mathbb{R}$ denotes the set of real numbers. For a given integer $n \geq 2$, $\mathbb{R}^n$ denotes the n-times Cartesian product of $\mathbb{R}^n$. For vectors we use the norm notation $\|\cdot\|$ to refer to the l_2 norm, while for matrices we use $\|\cdot\|_F$ to denote the Frobenius norm, and the weighted Frobenius norm $\|M\|_W$ is defined as $\|W^{-0.5}MW^{-0.5}\|_F$ for some positive definite matrix W. Superscript ^{T} refers to transpose operator. We use $[x]_l$ to denote the lth component of the vector x, and subscripts i (as in x_i) refer to the ith iteration of an iterative numerical scheme. In some places, we drop the iteration subscripts for compactness and simply use x and x_+ (or B and B_+) to indicate the optimization vector (or the BFGS approximation matrix) at the current iterate and the next iteration. The symbol x_* is used for the minimizer of the optimization problem, and subscripts $*$ are used to indicate a quantity that corresponds to or is evaluated at x_*, for example, $B_* = \nabla^2 f(x_*)$ and $\|\cdot\|_* = \|\cdot\|_{B_*}$. Lowercase Greek letters are used for scalars or scalar functions and, in general, α is used for steplengths. Another frequently used shorthand is $\sigma(u,v) = \max\{\|u - x_*\|, \|v - x_*\|\}$.

2. Derivation of structured quasi-Newton secant updates.

In this section we derive "structured" BFGS formulas for structured Hessians $\nabla^2 f(x) = K(x)+U(x)$ of our problem of interest (1). We drop the i indexing and use the subscript $+$ to denote the quantities updated at each iteration (indexed by $i+1$ in the preceding section).

Our goal is to derive structured formulas that use the exact Hessian information $K(x_+)$ and only approximate the missing curvature, $U(x_+)s$, along the direction s. We are specifically looking for *structured* secant update formulas in the form

$$B_+ = K(x_+) + A_+, \tag{7}$$

where A_+ approximates the unknown Hessian $U(x_+)$ in the spirit of the secant equation (2). That is, we require that

$$A_+ s = \bar{y} := \nabla u(x_+) - \nabla u(x). \tag{8}$$

We remark that one also can use an "unstructured" right-hand side in the secant equation (2), that is, require that A_+ satisfy $(K(x_+)+A_+)s = y := \nabla f(x_+) - \nabla f(x)$. Doing so would imply that $A_+ s = \nabla u(x_+) - \nabla u(x) + [\nabla k(x_+) - \nabla k(x) - K(s_+)s]$, which forces A_+ to inadvertently incorporate the approximate curvature $\nabla k(x_+) - \nabla k(x) - K(s_+)s$ corresponding to the term $k(\cdot)$ for which the exact curvature (Hessian) is available. Other structure-exploiting choices for $\bar{y}$ also are possible, for example, for nonlinear least-squares problems [9, 6, 17] and the Lagrangian function in SQP methods for constrained optimization [22, 16].

Similarly to the preceding section, we derive the analytic formulas for A_+ by imposing a proximity criterion together with the symmetry condition and the secant equation. We observe that in contrast to the classical quasi-Newton update, the proximity condition (3) can be formulated in two ways, depending on where $K(\cdot)$ is evaluated. That is, it can be posed as either

$$A_+^M = \underset{X}{\operatorname{argmin}} \|(X + K(x_+))^{-1} - (A^M + K(x))^{-1}\|_W \quad \text{or} \tag{9}$$

$$A_+^P = \underset{X}{\operatorname{argmin}} \|(X + K(x_+))^{-1} - (A^P + K(x_+))^{-1}\|_W. \tag{10}$$

Hereafter the superscripts ^{P} stand for "plus" and refer to the updates obtained using the evaluation $K(x_+)$ in the proximity condition; in contrast, superscripts ^{M} stand for "minus" and denote the updates that use $K(x)$ in the proximity condition. We also denote $B_+^M = K(x_+) + A_+^M$ and $B_+^P = K(x_+) + A_+^P$ as the structured updates of the form (7) corresponding to the above two variational characterizations. We observe that the latter variational characterization seems more reasonable from a least-squares perspective since it uses updated Hessian information $K(x_+)$ and, therefore, A_+^M is not required to approximate the change in the known part of the Hessian.

On the other hand, the trace-determinant variational characterization [4, 11, 15] indicates that solving (9) and (10) is in fact analogous to enforcing the eigenvalues of $(A_+^M + K(x_+))^{-1}(A^M + K(x))$ and $(A_+^P + K(x_+))^{-1}(A^P + K(x_+))$ to be as close to 1 as possible. In this respect, the update based on (9) would be more suitable since B_+^M is more likely to inherit the spectral properties (e.g., positive definiteness) of $B^M = A^M + K(x)$; in contrast, B_+^P has spectral properties similar to $A^P + K(x_+) = B^P + K(x_+) - K(x)$, which is a matrix that is not necessarily positive definite when $K(x_+) - K(x)$ is large.

Based on the discussion of the preceding two paragraphs, we are uncertain which of the variational characterizations (9) and (10) has superior properties. For this reason we consider both updates in this paper.

Before deriving the structured BFGS formulas, we observe that $\bar{y} = A_+ s$ can be equivalently written as $\bar{y} = (H_+^{-1} - K(x_+))s$, where H_+ denotes the inverse of B_+^M or B_+^P. This leads to the "structured" inverse secant condition

$$H_+(\bar{y} + K(x_+)s) = s.$$

Therefore (9) and (10) with the symmetry condition and the inverse secant equation become

$$\begin{aligned} H_+ = \underset{Y}{\operatorname{argmin}} \quad & \|Y - H\|_W \\ \text{s.t.} \quad & Y(\bar{y} + K(x_+)s) = s, \\ & Y = Y^T, \end{aligned} \tag{11}$$

where H is either $(B^M)^{-1}$ or $(B^P)^{-1}$ (corresponding to (9) or (10), respectively). The solution to this variational characterization is (see [15, Corollary 2.5])

$$H_+ = (I - \bar{\gamma} s(\bar{y} + K(x_+)s)^T) H (I - \bar{\gamma}(\bar{y} + K(x_+)s)s^T) + \bar{\gamma} s s^T, \tag{12}$$

where we define

$$\bar{\gamma} = \left[(\bar{y} + K(x_+)s)^T s\right]^{-1}. \tag{13}$$

The BFGS update for the Hessian matrix can be obtained again by using the Sherman–Morison–Woodbury formula, namely,

$$H_+^{-1} = H^{-1} - \frac{H^{-1} s s^T H^{-1}}{s^T H^{-1} s} + \bar{\gamma}(\bar{y} + K(x_+)s)(\bar{y} + K(x_+)s)^T.$$

This allows us to write the BFGS update corresponding to (9) as

$$\begin{aligned} A_+^M &= A^M + K(x) - K(x_+) \\ &\quad - \frac{(A^M + K(x))ss^T(A^M + K(x))}{s^T(A^M + K(x))s} + \bar{\gamma}(\bar{y} + K(x_+)s)(\bar{y} + K(x_+)s)^T \end{aligned}$$

and the BFGS update corresponding to (10) as

$$A_+^P = A^P - \frac{(A^P + K(x_+))ss^T(A^P + K(x_+))}{s^T(A^P + K(x_+))s} + \bar{\gamma}(\bar{y} + K(x_+)s)(\bar{y} + K(x_+)s)^T.$$

To simplify notation, we introduce

$$\mathbb{B}(s, y, M) = -\frac{Mss^TM}{s^TMs} + \frac{yy^T}{y^Ty}.$$

We can then express A_+^M and A_+^P as

$$A_+^M = A^M + K(x) - K(x_+) + \mathbb{B}(s, \bar{y} + K(x_+)s, A^M + K(x)), \tag{14}$$

$$A_+^P = A^P + \mathbb{B}(s, \bar{y} + K(x_+)s, A^P + K(x_+)). \tag{15}$$

The corresponding structured BFGS updates for the Hessian matrix become

$$B_+^M = A_+^M + K(x_+) = B^M + \mathbb{B}(s, \bar{y} + K(x_+)s, B^M), \tag{16}$$

$$\begin{aligned} B_+^P = A_+^P + K(x_+) &= B^P + K(x_+) - K(x) \\ &\quad + \mathbb{B}(s, \bar{y} + K(x_+)s, B^P + K(x_+) - K(x)). \end{aligned} \tag{17}$$

Note on structured DFP update formulas. One can use this formal approach to derive a DFP structured update. The structured counterpart of the DFP variational form (6) would be

$$\begin{aligned} \min_{Y} \quad & \|Y - A\|_W \\ \text{s.t.} \quad & Ys = \bar{y}, \\ & Y = Y^T, \end{aligned}$$

which has the solution

$$A_+^{DFP} = (I - \bar{\gamma}'\bar{y}s^T)A^{DFP}(I - \bar{\gamma}'s\bar{y}^T) + \bar{\gamma}'\bar{y}\bar{y}^T,$$

where $\bar{\gamma}'$ is the scalar $(s^T\bar{y})^{-1}$. We observe that the corresponding structured DFP update $B_+^{DFP} = A_+^{DFP} + K(x_+)$ does not explicitly incorporate the approximation B^{DFP}. Therefore, it is difficult to investigate the properties of the Hessian approximation (e.g., positive definiteness, bounded deterioration) that are needed in our linesearch approach. Trust-region methods can be, for example, an alternative avenue for structured DFP updates; we defer such investigation to future work.

3. A linesearch quasi-Newton algorithm for the structured updates.

In this section we propose and analyze a linesearch algorithm that employs the structured BFGS updates B^M and B^P introduced in the preceding section. We first investigate the hereditary positive definiteness of the structured updates and, motivated by this investigation, propose a linesearch algorithm derived from the Wolfe conditions enhanced by an additional condition to ensure positive definiteness of the update away from a neighborhood of the solution. Furthermore, we show that the algorithm converges to stationary points, and we prove that the algorithm is locally superlinearly convergent.

3.1. Hereditary positive definiteness. The positive definiteness of the BFGS update is needed to ensure that the quasi-Newton direction is a descent direction [20]. The BFGS *unstructured* updates remain positive definite (p.d.) as long as a set of conditions, namely, the Wolfe conditions, are satisfied by the linesearch [20]. As we show in this section, this is not necessarily the case for our *structured* BFGS updates B^M and B^P. For the B^M update we propose a linesearch mechanism that ensures the positive definiteness of the update throughout the algorithm. For the B^P update such modification is not possible, and we will ensure positive definiteness by using the traditional approach of inertia/eigenvalue regularization.

3.1.1. Structured B^M update. Provided B^M is p.d., a sufficient condition for B_+^M to be p.d. is $\bar{\gamma} > 0$. To prove this, we first observe that (12) allows us to write for B_+^M that

$$\left(B_+^M\right)^{-1} = \left[I - \bar{\gamma} s(\bar{y} + K(x_+)s)\right]^T \left(B^M\right)^{-1} \left[I - \bar{\gamma}(\bar{y} + K(x_+)s)s^T\right] + \bar{\gamma} s s^T,$$

which implies that

$$\begin{aligned} u^T \left(B_+^M\right)^{-1} u = &\left[u - \bar{\gamma}(s^T u)(\bar{y} + K(x_+)s)\right]^T \left(B^M\right)^{-1} \\ &\cdot \left[u - \bar{\gamma}(s^T u)(\bar{y} + K(x_+)s)\right] + \bar{\gamma}(s^T u)^2 \geq 0 \end{aligned} \tag{18}$$

for all vectors $u \in \mathbb{R}^n$ since B^M is p.d. Furthermore, under the assumption that $\bar{\gamma} > 0$, equality holds if and only if $u - \bar{\gamma}(s^T u)(\bar{y} + K(x_+)s) = 0$ and $s^T u = 0$, which implies that $u = 0$. Therefore the inequality is strict whenever $u \neq 0$, showing that B_+^M is positive definite.

In general $\bar{\gamma} = 1/\left[(\bar{y} + K(x_+)s)^T s\right]$ is not necessarily positive. To see this, we use the mean value theorem for the function $t \mapsto s^T \nabla u(x + t(x_+ - x))$ on $[0, 1]$ to write $s^T \bar{y} = s^T(\nabla u(x_+) - \nabla u(x)) = s^T \nabla^2 u(\bar{x})(x_+ - x) = s^T \nabla^2 u(\bar{x}) s$, where $\bar{x} \in [x, x_+]$; this implies that

$$\bar{\gamma} = 1/\left[s^T(U(\bar{x}) + K(x_+))s\right], \tag{19}$$

which can be negative since $U(\bar{x}) + K(x_+)$ is not necessarily p.d. However, if there exist $x_+ = x + \alpha x$ for which $s^T \nabla^2 f(x_+) s^T = s^T(U(x_+) + K(x_+))s > 0$, the positiveness of $\bar{\gamma}$ can be ensured provided $\bar{x}$ is close to x_+. Below we show that $s^T \nabla^2 f(x_+) s^T > 0$ holds for a subset of x_+ satisfying the Wolfe conditions (22) and (23) below (see also [20, Chapter 3]), and we enforce the latter condition in the linesearch algorithm. We introduce a simplifying notation and denote

$$x_+ = x_+(\alpha) = x + \alpha p, \quad \phi(\alpha) = f(x + \alpha p).$$

In other words, ϕ is f on the search direction p. Observe that

$$\phi'(\alpha) = \nabla f(x + \alpha p)^T p, \tag{20}$$

$$\phi''(\alpha) = (p)^T \nabla^2 f(x + \alpha p) p. \tag{21}$$

The following lemma shows that, for the purpose of leading to a $\bar{\gamma} > 0$, a subset of points satisfying the Wolfe conditions [20] also satisfies the inequality $\phi''(\alpha) > 0$.

LEMMA 1. *If p is a descent direction, namely, $\phi'(0) < 0$, and if $\phi(\alpha)$ is bounded from below on $[0, \infty)$, then there exists an interval of points α^* such that $\phi''(\alpha^*) > 0$*

and $x + \alpha^* p$ *satisfies the Wolfe conditions:*

$$\phi(\alpha^*) \le \phi(0) + c_1 \alpha^* \phi'(0), \tag{22}$$

$$\phi'(\alpha^*) \ge c_2 \phi'(0), \tag{23}$$

where $0 < c_1 < c_2 < 1$.

Proof. First observe that there is an $\alpha_1 > 0$ for which $\phi'(\alpha_1) = c_1 \phi'(0)$. To see this, consider $l_1(\alpha) = \phi(0) + c_1 \alpha \phi'(0)$. Since $\phi'(0) < 0$, $l_1(\alpha)$ is unbounded; however, $l_1(0) = \phi(0)$ and $l_1'(0) > \phi'(0)$ implies that there is at least one positive α where l_1 and ϕ intersect. We denote the smallest such value by α_0. Therefore, we have $\phi(\alpha_0) = l_1(\alpha_0)$, or, equivalently, $\phi(\alpha_0) - \phi(0) = c_1 \alpha_0 \phi'(0)$. By the mean value theorem we have that there exists $\alpha_1 \in (0, \alpha_0)$ such that $\phi(\alpha_0) - \phi(0) = \alpha_0 \phi'(\alpha_1)$. The last two equalities show that $\phi'(\alpha_1) = c_1 \phi'(0)$.

Since $\phi'(0) < c_2 \phi'(0) < c_1 \phi'(0) = \phi'(\alpha_1)$, one can similarly show that there exists $\alpha_2 \in (0, \alpha_1)$ such that $\phi'(\alpha_2) = c_2 \phi'(0)$. Let α_2 denote the largest such value.

The mean value theorem applied to $\phi'(\alpha)$ on $[\alpha_2, \alpha_1]$ indicates that there exists $\alpha^* \in [\alpha_2, \alpha_1]$ such that $(\alpha_1 - \alpha_2)\phi''(\alpha^*) = \phi'(\alpha_1) - \phi'(\alpha_2)$, which implies $\phi''(\alpha^*) > 0$.

It remains to prove that α^* satisfies Wolfe conditions (22) and (23). The sufficient decrease condition (22) is satisfied in fact by any $\alpha \in (0, \alpha_0)$ ($l_1(\alpha)$ dominates $\phi(\alpha)$ on $(0, \alpha_0)$, as pointed out above) and hence by α^* also. To show that the curvature condition (23) is satisfied, we note that $\phi'(\alpha^*) \le \phi'(\alpha_2) = c_2 \phi'(0)$ cannot hold since it would imply that $\phi'(\cdot)$ takes the value $c_2 \phi'(0)$ on (α^*, α_1), contradicting the maximality α_2 in this respect. Therefore $\phi'(\alpha^*) > c_2 \phi'(0)$, which proves (23). ∎

According to Lemma 1, the Hessian is p.d. (and thus $\bar\gamma > 0$) along the direction p at some steplength α^* if $\bar x$ from (19) is close to $x_+ = x + \alpha^* p$. This suggests a modification of the secant equation (11), namely, that it should be applied at $x + \alpha_* p$ and x_+ instead of at x and x_+, for some α_* close enough to α^*; a continuation argument can be used to argue that the BFGS update B_+^M becomes p.d. with this modification. This observation is formalized in Theorem 2 below. Before that step, we introduce the following notation:

$$\bar y(\alpha_1, \alpha_2) = \nabla u(x + \alpha_1 p) - \nabla u(x + \alpha_2 p), \tag{24}$$

$$s(\alpha_1, \alpha_2) = (x + \alpha_1 p) - (x + \alpha_2 p) = (\alpha_1 - \alpha_2) p, \tag{25}$$

$$\bar\gamma(\alpha_1, \alpha_2) = \left[(\bar y(\alpha_1, \alpha_2) + K(x + \alpha_1 p) s(\alpha_1, \alpha_2))^T s(\alpha_1, \alpha_2) \right]^{-1}. \tag{26}$$

Here α_1 and α_2 are positive numbers. Observe that $\bar y$, s, and $\bar\gamma$ correspond to (24), (25), and (26) for $\alpha_1 = \alpha^*$ and $\alpha_2 = 0$.

THEOREM 2. *Let* α^* *denote one of the steplengths given by Lemma* 1 *and let* $x_+ = x + \alpha^* p$. *Then there exists an* α_* *close to* α^* *such that the modified update*

$$B_+^M = A_+^M + K(x_+) = B^M + \mathbb{B}\left(s(\alpha^*, \alpha_*), \bar y(\alpha^*, \alpha_*) + K(x_+) s(\alpha^*, \alpha_*), B^M\right) \tag{27}$$

is p.d., where

$$A_+^M = A^M - \frac{\left[A^M + K(x_+)\right] s(\alpha^*, \alpha_*) s(\alpha^*, \alpha_*)^T \left[A^M + K(x_+)\right]}{s(\alpha^*, \alpha_*)^T \left[A^M + K(x_+)\right] s(\alpha^*, \alpha_*)} + \bar\gamma(\alpha^*, \alpha_*) \left[\bar y(\alpha^*, \alpha_*) + K(x_+) s(\alpha^*, \alpha_*)\right] \left[\bar y(\alpha^*, \alpha_*) + K(x_+) s(\alpha^*, \alpha_*)\right]^T.$$

Proof. Since $\phi''(\alpha^*) > 0$ by Lemma 1, one can easily see that for any $\hat{\alpha}$

$$s(\alpha^*, \hat{\alpha})^T \left[U(x_+) + K(x_+)\right] s(\alpha^*, \hat{\alpha}) > 0. \tag{28}$$

On the other hand, one can write based on the mean value theorem, similarly to obtaining (19), that for any $\alpha_* \neq \alpha^*$

$$\bar{\gamma}(\alpha^*, \alpha_*) = 1/\left[s(\alpha^*, \alpha_*)^T \left[U(x + \hat{\alpha}p) + K(x + \alpha^* p)\right] s(\alpha^*, \alpha_*)\right], \tag{29}$$

where $\hat{\alpha}$ is in the open interval with endpoints α_* and α^*.

Then (28), (29), and the continuity of $U(\cdot)$ imply that $\bar{\gamma}(\alpha^*, \alpha_*) > 0$ (and thus B^M_+ is p.d.) provided that α_* (and hence $\hat{\alpha}$) is close to α^*. ◻

The numerical linesearch procedure would be responsible for computing the endpoints α_* and α^* that maintain the positive definiteness of the structured update (27). We show in Appendix A that the state-of-the-art linesearch algorithm of Moré and Thuente (MT) [18] needs only a slight modification of the termination criteria (to include $\bar{\gamma}(\alpha^*, \alpha_*) > 0$) to *provably* find a Wolfe point α^* and a second secant endpoint α_* such that update (27) is p.d. The secant endpoint α_* is one of the endpoints of the so-called uncertainty interval employed by the MT linesearch and, therefore, requires no function or gradient in addition to the one already performed by the MT linesearch. However, the modified linesearch may take additional iterations whenever the positive curvature condition $\bar{\gamma}(\alpha^*, \alpha_*) > 0$ is not satisfied at the Wolfe point that would be normally returned by the standard MT linesearch. A bit surprisingly, this occurred for only a few test problems, which occasionally required one or two additional searches to satisfy the curvature condition. Also, we remark that the known part of the Hessian $K(\cdot)$ needs to be applied once to a vector at each iteration of the linesearch in order to compute $\bar{\gamma}(\alpha^*, \alpha_*)$ based on (26); other than this, the computation of $\bar{\gamma}(\alpha^*, \alpha_*)$ in the additional stopping criterion reuses function or gradient values and requires only low-cost vector-vector operations.

3.1.2. Structured B^P_+ update. For the structured B^P_+, the situation is different from that for the B^M. We can write, similarly to B^M, that

$$\begin{aligned} u^T (B^P_+)^{-1} u = u^T \left(H(x_+) + A^P_+\right)^{-1} u &= \left[u - \bar{\gamma}(s^T u)(\bar{y} + K(x_+)s)\right]^T \\ &\cdot \left(B^P - K(x) + K(x_+)\right)^{-1} \left[u - \bar{\gamma}(s^T u)(\bar{y} + K(x_+)s)\right] + \bar{\gamma}(s^T u)^2, \end{aligned} \tag{30}$$

which can be negative since $B^P - K(x) + K(x_+)$ is not necessarily p.d. even if B^P is p.d.

To ensure the positive definiteness of the structured B^P_+ update, we use a regularization approach that consists of adding a positive multiple of the identity, namely, we replace $B^P_+ = A^P_+ + K(x_+)$ with

$$\widetilde{B}^P_+ = A^P_+ + K(x_+) + \sigma I \tag{31}$$

when computing the search direction, where $\sigma > 0$ is chosen such that $\widetilde{B}^P_+$ is p.d.

This regularization approach is widely used in Newton-like methods based on linesearches [20] and involves trying successive, increasingly large nonnegative values of σ (starting with $\sigma = 0$) until $\widetilde{B}^P_+$ becomes p.d. In the context of our structured quasi-Newton linesearch, one can easily compute an initial trial diagonal perturbation σ based on the observation that the positive definiteness of $\widetilde{B}^P_+$ implies that $s^T \widetilde{B}^P_+ s$

should be strictly positive. Since $s^T \widetilde{B}_+^P s = s^T(A_+^P + K(x_+) + \sigma I)s = (\bar{y} + K(x_+)s)^T s + \sigma s^T s$, it makes sense to choose the initial σ such that $(\bar{y} + K(x_+)s)^T s + \sigma s^T s \geq \varepsilon > 0$ whenever $(\bar{y} + K(x_+)s)^T s \leq 0$, where $\varepsilon > 0$ is a safeguard to avoid division by small numbers. In particular, such an initial σ would be $\sigma = \max\{0, (\varepsilon - (\bar{y} + K(x_+)s)^T s)/\|s\|^2\}$. This initial choice of σ requires no additional function evaluations and has negligible computational cost since $\bar{y} + K(x_+)s$ has been already computed for $\bar{\gamma}$. This strategy avoids an unnecessary initial matrix factorization corresponding to $\sigma = 0$ when $(\bar{y} + K(x_+)s)^T s \leq 0$ during the linesearch. For example, this occurred in our numerical experiments for one test problem in 30 linesearches. However, the initial value of σ does not necessarily ensure that $\widetilde{B}_+^P$ is p.d., and subsequent increases of σ may be required, which we do by doubling σ, similarly to IPOPT's inertia regularization heuristic [23].

3.2. Global convergence. The satisfaction of the Wolfe conditions (22) and (23) at each iteration implies that the sequence of iterates has a limit point x_* that satisfies $\|\nabla f(x_*)\| = 0$ [8, Theorem 6.3] (also see [24, 25]).

3.3. Local superlinear convergence. Of great interest in computational practice is also how fast the algorithm converges. In this section we prove that the two structured BFGS updates we derived in section 2, together with the Wolfe-based linesearch mechanism of section 3, have *local* convergence properties specific to quasi-Newton methods with secant updates, for example, q-superlinear convergence. Specifically, we show in this section that the iterates produced by our algorithm satisfy

$$\lim_{i \to \infty} \frac{\|x_{i+1} - x_*\|}{\|x_i - x_*\|} = 0.$$

We work under the following "standard" assumptions that have been used in many publications dedicated to the analysis of local convergence rates of certain quasi-Newton secant methods; for example, see the review paper [7].

ASSUMPTION 3. *There exists a local minimizer x_* to problem* (1).

ASSUMPTION 4. *The objective f is of class C^2, and the Hessians $\nabla^2 f$ and $\nabla^2 k$ are locally Lipschitz continuous at x_*; that is, there exist $L \geq 0$, $L_\kappa \geq 0$, and a neighborhood $\mathcal{N}_1(x_*)$ of x_* such that for any $x \in \mathcal{N}_1(x_*)$*

$$\|\nabla^2 f(x) - \nabla^2 f(x_*)\| \leq L\|x - x_*\|, \tag{32}$$

$$\|\nabla^2 k(x) - \nabla^2 k(x_*)\| \leq L_\kappa\|x - x_*\|. \tag{33}$$

Under this assumption, u is also C^2, and its Hessian is locally Lipschitz. We denote the Lipschitz constant by L_u. From the triangle inequality it follows that $L_u \leq L + L_\kappa$.

ASSUMPTION 5. *The Hessian is continuously bounded from below and above in a neighborhood $\mathcal{N}_2(x_*)$ of x_*; that is, there exist $m > 0$ and $M > 0$ such that*

$$m\|x\|^2 \leq x^T \nabla^2 f(x_*) x \leq M\|x\|^2 \quad \forall x \in \mathcal{N}_2(x_*). \tag{34}$$

Our analysis is also standard in the sense that the superlinear convergence is proved by using a characterization of superlinear convergence due to Dennis and Moré [7], namely, Theorem 6 below. In fact, the remainder of this section essentially shows that our structured BFGS updates satisfy the hypotheses of the Dennis–Moré characterization of superlinear convergence.

THEOREM 6 (see [8, Theorem 3.1]). *Let B_i be a sequence of nonsingular matrices, and consider updates of the form $x_{i+1} = x_i - B_i^{-1}\nabla f(x_i)$ such that x_i remains in a neighborhood of x_*, $x_i \neq x_*$ for all $i \geq 0$, and converges to x_*. Also, let $B_* = \nabla^2 f(x_*)$. Then the sequence $\{x_i\}$ converges superlinearly to x_* if and only if*

$$\lim_{i\to\infty} \frac{\|(B_i - B_*)(x_{i+1} - x_i)\|}{\|x_{i+1} - x_i\|} = 0. \tag{35}$$

For our linesearch approach, we note that the Wolfe conditions (22) and (23) hold with unit steps ($\alpha = 1$) in the limit, according to a well-known result [8, Theorem 6.4]. Since the (modified) MT [18] linesearch always starts with unit steps, it follows that in a neighborhood of x_* our algorithm uses updates of the form $x_{i+1} = x_i - B_i^{-1}\nabla f(x_i)$. In what follows, the intersection of this neighborhood with $\mathcal{N}_2(x_*)$ is denoted by $\mathcal{N}_3(x_*)$. Therefore, condition (35) is equivalent to

$$\lim_{i\to\infty} \frac{\|(B_i - B_*)s_i\|}{\|s_i\|} = 0, \tag{36}$$

a result that we prove in the remainder of this section. We remark that the remaining conditions required by Theorem 6 are satisfied. In particular, our updates remain p.d., and the convergence was proved in section 3.2.

Key to showing (36) is proving a "bounded deterioration" property for B_+^M and B_+^P, namely, Theorem 11. This property implies q-linear convergence, which in turn is needed to show (36) [3].

In what follows we denote $\sigma(u, v) = \max\{\|u - x_*\|, \|v - x_*\|\}$ and $B_* = \nabla^2 f(x_*)$.

LEMMA 7. *Let x and x_+ be in $\mathcal{N}_3(x_*)$. Under Assumptions 3–5, the following inequalities hold:*

(i) $\|(\bar{y} + K(x_+)s) - B_*s\| \leq (L + 2L_\kappa)\sigma(x, x_+)\|s\|$,

(ii) $\|\bar{y} + K(x_+)s\| \leq (M + (L + 2L_\kappa)\sigma(x, x_+))\|s\|$,

(iii) $s^T(\bar{y} + K(x_+)s) \leq (M + (L + 2L_\kappa)\sigma(x, x_+))\|s\|^2$, *and*

(iv) $s^T(\bar{y} + K(x_+)s) \geq [m - (L + 2L_\kappa)\sigma(x, x_+)]\,\|s\|^2$,

where $s = x_+ - x$ and, as before, $\bar{y} = \nabla u(x_+) - \nabla u(x)$.

Proof. (i) We first write

$$\begin{aligned}\bar{y} + K(x_+)s - B_*s &= \nabla u(x_+) - \nabla u(x) + K(x_+)s - K(x_*)s - U(x_*)s \\ &= \int_0^1 [U((1-t)x + tx_+)s]dt - U(x_*)s \;+\; [K(x_+) - K(x_*)]s.\end{aligned}$$

Then, standard norm inequalities and Assumption 5 imply that

$$\begin{aligned}\|(\bar{y} + K(x_+)s) - B_*s\| &\leq \int_0^1 [L_u(1-t)\|x - x_*\| + L_u t\|x_+ - x_*\|]\,\|s\|dt + L_\kappa\|x_+ - x_*\|\|s\| \\ &\leq \int_0^1 L_u\sigma(x, x_+)\|s\|dt + L_\kappa\|x_+ - x_*\|\|s\| \\ &\leq L_u\sigma(x, x_+)\|s\| + L_\kappa\sigma(x, x_+)\|s\| \\ &\leq (L + 2L_\kappa)\sigma(x, x_+)\|s\|.\end{aligned}$$

(ii) The inequality follows from the triangle inequality, inequality (i), and Assumption 5, namely, $\|\bar{y} + K(x_+)s\| \leq \|\bar{y} + K(x_+)s - B_*s\| + \|B_*s\| \leq (M + (L + 2L_\kappa)\sigma(x, x_+))\|s\|$.

(iii) The inequality follows from the Cauchy–Schwarz inequality and inequality (ii).

(iv) Similar to the proof of inequality (i) we write

$$\begin{aligned} s^T \left[U(x_*)s - \bar{y}\right] &= s^T \left[U(x_*)s - \int_0^1 U((1-t)x + tx_+)sdt\right] \\ &= s^T \int_0^1 \left[U(x_*) - U((1-t)x + tx_+)\right] sdt \\ &\le \|s\| \int_0^1 \left[(1-t)L_u\|x_* - x\| + tL_u\|x_* - x_+\|\right] \|s\|dt \\ &= L_u\sigma(x, x_+)\|s\|^2 \le (L + L_\kappa)\sigma(x, x_+)\|s\|^2. \end{aligned}$$

This allows us to use Assumptions 4 and 5 to write

$$\begin{aligned} m\|s\|^2 &\le s^T B_* s \\ &= s^T(\bar{y} + K(x_+))s + s^T\left[U(x_*)s - \bar{y}\right] + s^T(K(x_*) - K(x_+))s \\ &\le s^T(\bar{y} + K(x_+))s + (L + L_\kappa)\sigma(x, x_+)\|s\|^2 + L_\kappa\|x_* - x_+\|\|s\|^2 \\ &\le s^T(\bar{y} + K(x_+))s + (L + 2L_\kappa)\sigma(x, x_+)\|s\|^2. \end{aligned}$$

This completes the proof of inequality (iv). ∎

We introduce the "theoretical" update

$$B' = B + \mathbb{B}(s, B_*s, B) = B - \frac{Bss^TB}{s^TBs} + \frac{B_*ss^TB_*}{s^TB_*s} \tag{37}$$

that uses the curvature at the optimal solution along the current update direction, namely, B_*s, instead of the structured curvature $\bar{y} + K(x_+)s$. This theoretical update plays an important role in establishing the bounding inequalities needed to prove the bounded deterioration of B_+^M and B_+^P. The theoretical update of B^M is denoted by $B^{M\prime}$.

To simplify the proofs, we also transform the B^P update into a slightly different form and notation. In particular, we observe from (17) that the B_+^P formula can be written as $B_+^P = \bar{B}^P + \mathbb{B}(s, \bar{y} + K(x_+)s, \bar{B}^P)$, where $\bar{B}^P = B^P + K(x_+) - K(x)$. One can easily see that a recurrence similar to (17) holds:

$$\bar{B}_+^P = B_+^P + K(x_{++}) - K(x_+) = \bar{B}^P + \mathbb{B}(s, \bar{y} + K(x_+)s, \bar{B}^P) + [K(x_{++}) - K(x_+)]. \tag{38}$$

This new notation simplifies the use of the theoretical update in the proofs, as will become apparent later in this section. We denote the theoretical update of $\bar{B}^P$ by $\bar{B}^{P\prime}$.

LEMMA 8 (see [6, Lemma 2.1]). *Consider a symmetric matrix M and vectors u, z such that $u^Tu = 1$ and $u^TMu = (u^Tz)$. If we define $M_+ = M + uu^T - zz^T$, then*

$$\|M_+ - I\|_F^2 = \|M - I\|_F^2 - \left[(1 - z^Tz)^2 + 2\left(z^TMz - (z^Tz)^2\right)\right]. \tag{39}$$

Moreover, if M is also positive definite, if $u = v/\|v\|$, and if $z = Mv/\sqrt{v^TMv}$ for some $v \neq 0$, then

$$z^TMz \ge (z^Tz)^2 \tag{40}$$

and

$$\|M_+ - I\|_F \le \|M - I\|_F. \tag{41}$$

The following results are a direct consequence of Lemma 8 and reveal interesting properties of B'. Here we denote $\|M\|_* \triangleq \|M\|_{B_*} = \|B_*^{-0.5} M B_*^{-0.5}\|_F$.

LEMMA 9. *Let B denote B^M or $\bar{B}^P$, and let B' denote $B^{M'}$ or $\bar{B}^{P'}$, respectively, and consider $z = \frac{B_*^{-0.5}Bs}{\sqrt{s^TBs}}$. Then one can write*
 (i) $\|B' - B_*\|_*^2 = \|B - B_*\|_*^2 - \left[(1 - z^Tz)^2 + 2\left(z^TB_*^{-0.5}BB_*^{-0.5}z - (z^Tz)^2\right)\right]$,
 (ii) $z^TB_*^{-0.5}BB_*^{-0.5}z \geq (z^Tz)^2$, *and*
 (iii) $\|B' - B_*\|_* \leq \|B - B_*\|_*$.

Proof. We first observe that

$$\|B' - B_*\|_*^2 = \left\| B_*^{-0.5}BB_*^{-0.5} - I + \frac{B_*^{0.5}ss^TB_*^{0.5}}{s^TB_*s} - \frac{B_*^{-0.5}Bss^TBB_*^{-0.5}}{s^TBs} \right\|_F^2. \tag{42}$$

The inequality (i) follows from (39) by taking $M = B_*^{-0.5}BB_*^{-0.5}$ and $u = B_*^{0.5}s/\sqrt{s^TB_*s}$ in Lemma 8 and by observing that the lemma's condition $u^TMz = u^Tz$ is fulfilled.

Similarly, inequality (ii) follows from (40) of Lemma 8 with $v = B_*^{0.5}s \neq 0$ since one can easily verify that the conditions $u = v/\|v\|$ and $z = Mv/\sqrt{v^TMv}$ are satisfied. Furthermore, since $\|B - B_*\|_* = \|B_*^{-0.5}BB_*^{-0.5} - I\|_F$, inequality (iii) is just (41). ∎

We now bound the variation in our updates from the theoretical updates.

LEMMA 10. *If Assumptions* 3–5 *hold, then*
 (i) *there exists a constant $C_1^M > 0$ such that $\|B_+^M - B^{M'}\|_* \leq C_1^M\sigma(x, x_+)$ for any x, x_+ in a nontrivial neighborhood $\mathcal{N}_4(x_*)$ included in $\mathcal{N}_3(x_*)$;*
 (ii) *there exists a constant $\bar{C}_1^P > 0$ such that $\|B_+^P - \bar{B}^{P'}\|_* \leq C_1^P\sigma(x, x_+)$ for any x, x_+ in a nontrivial neighborhood of x_*, to which, for the sake of brevity, we also refer as $\mathcal{N}_4(x_*)$, included in $\mathcal{N}_3(x_*)$.*

As a consequence of (ii),
 (iii) $\|\bar{B}_+^P - \bar{B}^{P'}\|_* \leq \bar{C}_1^P\sigma(x, x_+) + L_\kappa\sigma(x_+, x_{++})$ *for any x, x_+, and x_{++} in $\mathcal{N}_4(x_*)$.*

Proof. It is more illustrative to first prove (ii). We first observe that $B_+^P - \bar{B}^{P'}$ is a rank-two update since

$$B_+^P - \bar{B}^{P'} = \frac{(\bar{y} + K(x_+)s)(\bar{y} + K(x_+)s)^T}{(\bar{y} + K(x_+)s)^Ts} - \frac{B_*ss^TB_*}{s^TB_*s}.$$

Then we perform the following manipulation:

$$\begin{aligned} B_+^P - B^{P'} &= \frac{(\bar{y} + K(x_+)s)(\bar{y} + K(x_+)s - B_*s)^T}{(\bar{y} + K(x_+)s)^Ts} + \frac{(\bar{y} + K(x_+)s - B_*s)s^TB_*}{s^TB_*s} \\ &\quad + \frac{(\bar{y} + K(x_+)s)s^TB_*}{(\bar{y} + K(x_+)s)^Ts} - \frac{(\bar{y} + K(x_+)s)s^TB_*}{s^TB_*s} \\ &= \frac{(\bar{y} + K(x_+)s)(\bar{y} + K(x_+)s - B_*s)^T}{(\bar{y} + K(x_+)s)^Ts} + \frac{(\bar{y} + K(x_+)s - B_*s)s^TB_*}{s^TB_*s} \\ &\quad + (\bar{y} + K(x_+)s)s^TB_*\frac{\left(B_*s - (\bar{y} + K(x_+)s)\right)^Ts}{s^TB_*s \cdot (\bar{y} + K(x_+)s)^Ts}. \end{aligned}$$

Using the fact that $\|uv^T\|_F = \|u\|\|v\|$ and the standard norm inequalities, we write

$$\|B_+^P - B^{P'}\|_F \le \frac{\|\bar{y} + K(x_+)s\|\|\bar{y} + K(x_+)s - B_* s\|}{|(\bar{y} + K(x_+)s)^T s|} + \frac{\|(\bar{y} + K(x_+)s - B_* s)\|\|B_* s\|}{s^T B_* s}$$

$$+ \|\bar{y} + K(x_+)s\|\|B_* s\| \frac{\|\bar{y} + K(x_+)s - B_* s\|\|s\|}{s^T B_* s \cdot |(\bar{y} + K(x_+)s)^T s|}. \tag{43}$$

Furthermore, we use Assumption 5 and the inequalities of Lemma 7 to obtain

$$\begin{aligned}\|B_+^P - B^{P'}\|_F \le& \frac{[M + (L + 2L_\kappa)\sigma(x, x_+)]}{[m - (L + 2L_\kappa)\sigma(x, x_+)]}(L + 2L_\kappa)\sigma(x, x_+) \\ &+ \frac{M}{m}(L + 2L_\kappa)\sigma(x, x_+) \\ &+ \frac{[M + (L + 2L_\kappa)\sigma(x, x_+)]\, M}{m\,[m - (L + 2L_\kappa)\sigma(x, x_+)]}(L + 2L_\kappa)\sigma(x, x_+).\end{aligned}$$

In this inequality we assume that $\mathcal{N}_3(x_*)$ was shrunk to $\mathcal{N}_4(x_*)$ to ensure $m - (L + 2L_\kappa)\sigma(x, x_+) \ge \frac{1}{2}m > 0$. More specifically, $\mathcal{N}_4(x_*)$ should contain a ball of radius $\epsilon_4 = \frac{1}{2}\frac{m}{L+2L_\kappa}$ around x_* since this implies that $m - (L + 2L_\kappa)\sigma(x, x_+) \ge m - (L + 2L_\kappa)\epsilon_4 = \frac{1}{2}m > 0$. We obtain

$$\|B_+^P - B^{P'}\|_F \le \tilde{C}_1\, \sigma(x, x_+), \tag{44}$$

where $\tilde{C}_1 = \left[\frac{M+\frac{1}{2}m}{\frac{1}{2}m} + \frac{M}{m} + \frac{(M+\frac{1}{2}m)M}{\frac{1}{2}m^2}\right](L + 2L_\kappa)$.

Since $\|B_+^P - B^{P'}\|_* \le \|B_*^{-0.5}\|^2\|B_+^P - B^{P'}\|_F \le \frac{1}{m}\|B_+^P - B^{P'}\|_F$, we conclude that (ii) holds with $C_1^P = \tilde{C}_1/m$.

(i) The proof for this case is a bit more involved than for (ii) because the expression (27) of B_+^M uses the left "secant" endpoint α_* that is not necessarily zero as it is in the case for (ii). We recall that we work with $\alpha^* = 1$ as per the discussion following Theorem 6 and that $\alpha_* \in [0, 1)$.

First we observe that the expression (27) of B_+^M is exactly

$$\begin{aligned}B_+^M = B^M &- \frac{B^M s(1,\alpha_*)s(1,\alpha_*)^T B^M}{s(1,\alpha_*)^T B^M s(1,\alpha_*)} \\ &+ \frac{[\bar{y}(1,\alpha_*) + K(x_+)s(1,\alpha_*)]\,[\bar{y}(1,\alpha_*) + K(x_+)s(1,\alpha_*)]^T}{[\bar{y}(1,\alpha_*) + K(x_+)s(1,\alpha_*)]^T s(1,\alpha_*)}\end{aligned}$$

and that the second term in the right-hand side can be simplified to $\frac{B^M ss^T B^M}{s^T B^M s}$ since $s(1,\alpha_*) = (1-\alpha_*)s$. This shows that $B_+^M - B^{M'}$ is also a rank-two update in the form

$$B_+^M - B^{M'} = \frac{[\bar{y}(1,\alpha_*) + K(x_+)s(1,\alpha_*)]\,[\bar{y}(1,\alpha_*) + K(x_+)s(1,\alpha_*)]^T}{[\bar{y}(1,\alpha_*) + K(x_+)s(1,\alpha_*)]^T s(1,\alpha_*)} - \frac{B_* ss^T B_*}{s^T B_* s}.$$

Using again the fact that $s(1,\alpha_*) = (1-\alpha_*)s$, the second term can be manipulated to obtain

$$\begin{aligned}B_+^M - B^{M'} = &\frac{[\bar{y}(1,\alpha_*) + K(x_+)s(1,\alpha_*)]\,[\bar{y}(1,\alpha_*) + K(x_+)s(1,\alpha_*)]^T}{[\bar{y}(1,\alpha_*) + K(x_+)s(1,\alpha_*)]^T s(1,\alpha_*)} \\ &- \frac{B_* s(1,\alpha_*)s(1,\alpha_*)^T B_*}{s(1,\alpha_*)^T B_* s(1,\alpha_*)}.\end{aligned}$$

The above rank-two update has a form identical to that of the rank-two update from the proof of (ii) with $x + \alpha_* s$ replacing x. Consequently, the remainder of the proof is similar to that of (ii); the only remarkable difference is that Lemma 7 should be applied to $x + \alpha_* s$ and x_+ instead of to x and x_+. Consequently, it follows that $\|B_+^M - B^{M'}\|_* \le C_1^M \sigma(x + \alpha_* s, x_+)$ for some $C_1^M > 0$. This proves (i) since $\sigma(x + \alpha_* s, x_+) \le \sigma(x, x_+)$.

(iii) By definition, $\bar{B}_+^P = B_+^P + K(x_{++}) - K(x_+)$ (see (38)). This allows us to write, based on Assumption 4 and the triangle inequality, that

$$\|\bar{B}_+^P - \bar{B}^{P'}\|_* \le \|B_+^P - \bar{B}^{P'}\|_* + L_\kappa \|x_{++} - x_+\|,$$

which, based on (ii), proves (iii). As before, we used the fact that $\|x_{++} - x_+\| \le \sigma(x_+, x_{++})$. ∎

THEOREM 11. *Under Assumptions* 3–5*, there exist constants* $C_1^M > 0$ *and* $C_2^P > 0$ *independent of* x *and* x_+ *such that*

$$\|B_+^M - B_*\|_* \le \|B^M - B_*\|_* + C_1^M \sigma(x, x_+), \tag{45}$$

$$\|B_+^P - B_*\|_* \le \|B^P - B_*\|_* + C_2^P \sigma(x, x_+) \tag{46}$$

for any $x, x_+ \in \mathcal{N}_4(x_*)$.

Proof. To obtain (45), we use the triangle inequality together with (i) of Lemma 10 and (iii) of Lemma 9:

$$\|B_+^M - B_*\|_* \le \|B_+^M - B^{M'}\|_* + \|B^{M'} - B_*\|_* \le C_1^M \sigma(x, x_+) + \|B^M - B_*\|_*.$$

To prove (46), we follow an argument similar to the one above:

$$\|B_+^P - B_*\|_* \le \|B_+^P - \bar{B}^{P'}\|_* + \|\bar{B}^{P'} - B_*\|_* \le C_1^P \sigma(x, x_+) + \|\bar{B}^P - B_*\|_*.$$

The last term can be bounded based on the definition of $\bar{B}^P$ and using the triangle inequality and Assumption 4:

$$\|\bar{B}^P - B_*\|_* = \|B^P + K(x_+) - K(x) - B_*\|_* \le \|B^P - B_*\|_* + L_\kappa \sigma(x, x_+).$$

Therefore (46) follows with $C_2^P = C_1^P + L_\kappa$. ∎

The bounded deterioration property proved above implies q-linear convergence, namely, Theorem 12 below. This is only an intermediary step, which is needed to prove the asymptotic convergence (36) of the search direction by using the structured updates B_+^M and B_+^P to the Newton direction in Theorem 14.

Theorem 12 is known to hold (for example, see [3, Theorem 3.2]) for any BFGS update sequence that satisfies a bounded deterioration property, such as the one we proved in Theorem 11. Hence, we state the result without a proof. We still work under Assumptions 3–5.

THEOREM 12 (see [3, Theorem 3.2]). *Consider updates* $x_{i+1} = x_i - B_i^{-1}\nabla f(x_i)$*, where* B_i *are symmetric and satisfy a bounded deterioration property of* (45) *or* (46)*. Then for each* $r \in (0, 1)$*, there are positive constants* $\epsilon(r)$ *and* $\delta(r)$ *such that for* $\|x_0 - x_*\| \le \epsilon(r)$ *and* $\|B_0 - B_*\|_* \le \delta(r)$*, the sequence* $\{x_i\}$ *converges to* x_**. Furthermore,*

$$\|x_{i+1} - x_*\| \le r\|x_i - x_*\| \tag{47}$$

for each $i \ge 0$*. In addition, the sequences* $\{\|B_i\|\}$ *and* $\{\|B_i^{-1}\|\}$ *are uniformly bounded.*

To prove the asymptotic convergence (36), we need the following result that is similar to the bounded deterioration property but involves the theoretical updates. We drop the "+" subscripts and use "i" indexes for the remainder of this section. We remark that $\sigma(x_i, x_{i+1}) = \max\{\|x_i - x_*\|, \|x_{i+1} - x_*\|\} = \|x_i - x_*\|$ and $\|x_i - x_*\| \leq \epsilon(r)r^i$ for all $i \geq 0$ when one works under the conditions of Theorem 12. In what follows we assume without loss of generality that $\epsilon(r)$ from Theorem 12 is restricted so that $x_0 \in \mathcal{N}_4(x_*)$.

LEMMA 13. *Under the conditions of Theorem 12, for any given $r \in (0,1)$ there exist constants $C_2^M(r) > 0$ and $\bar{C}_2^P(r) > 0$ independent of the sequence $\{x_i\}$ such that*

$$\|B_{i+1}^M - B_*\|_*^2 \leq \|B_i^{M'} - B_*\|_*^2 + C_2^M(r) \cdot r^i, \tag{48}$$

$$\|\bar{B}_{i+1}^P - B_*\|_*^2 \leq \|\bar{B}_i^{P'} - B_*\|_*^2 + \bar{C}_2^P(r) \cdot r^i. \tag{49}$$

Proof. Based on the triangle inequality and (i) of Lemma 10, we first write

$$\begin{aligned}\|B_{i+1}^M - B_*\|_* &\leq \|B_{i+1}^M - B_i^{M'}\|_* + \|B_i^{M'} - B_*\|_* \\ &\leq C_1^M \sigma(x_i, x_{i+1}) + \|B_i^{M'} - B_*\|_* \leq C_1^M \epsilon(r) r^i + \|B_i^{M'} - B_*\|_*.\end{aligned} \tag{50}$$

On the one hand, based on (iii) of Lemma 9, this inequality implies that

$$\|B_{i+1}^M - B_*\|_* \leq C_1^M \epsilon(r) r^i + \|B_i^M - B_*\|_*,$$

which, in turn, inductively implies that

$$\|B_i^M - B_*\|_* \leq \|B_0^M - B_*\|_* + C_1^M \epsilon(r)(r^{i-1} + \cdots + 1) \leq \delta(r) + C_1^M \epsilon(r) \frac{1 - r^i}{1 - r}. \tag{51}$$

On the other hand, squaring (50) implies

$$\|B_{i+1}^M - B_*\|_*^2 \leq \|B_i^{M'} - B_*\|_*^2 + 2C_1^M \epsilon(r) r^i \|B_i^{M'} - B_*\|_* + (C_1^M)^2 \epsilon^2(r) r^{2i},$$

which, by (iii) of Lemma 9 and inequality (51), implies that

$$\|B_{i+1}^M - B_*\|_*^2 \leq \|B_i^{M'} - B_*\|_*^2 + r^i \left[2C_1^M \epsilon(r) \left(\delta(r) + C_1^M \epsilon(r) \frac{1 - r^i}{1 - r} \right) + (C_1^M)^2 \epsilon^2(r) r^i \right].$$

We remark that the quantity inside the brackets in the second term in this inequality is bounded from above by

$$C_2^M(r) = 2C_1^M \epsilon(r) \left(\delta(r) + C_1^M \epsilon(r) \frac{1}{1 - r} \right) + (C_1^M)^2 \epsilon^2(r),$$

which completes the proof of (48).

The proof of (49) is almost identical. The slight difference is the inequality (50), which translates into

$$\|\bar{B}_{i+1}^P - B_*\|_* \leq (\bar{C}_1^P + L_\kappa r)\epsilon(r) r^i + \|\bar{B}_i^{P'} - B_*\|_* \leq (\bar{C}_1^P + L_\kappa)\epsilon(r) r^i + \|\bar{B}_i^{P'} - B_*\|_*$$

since (iii) of Lemma 10 needs to be used. Relation (49) holds with

$$\bar{C}_2^P(r) = 2(\bar{C}_1^P + L_\kappa)\epsilon(r) \left(\delta(r) + (\bar{C}_1^P + L_\kappa)\epsilon(r) \frac{1}{1 - r} \right) + (\bar{C}_1^P + L_\kappa)^2 \epsilon^2(r). \qquad \square$$

We are now in a position to prove the final result.

THEOREM 14. *Under Assumptions* 3–5 *the update sequences* $\{B_i^M\}$ *and* $\{B_i^P\}$ *satisfy the asymptotic convergence limit* (36)*, provided that* x_0 *and* B_0 *are sufficiently close to* x_* *and* B_**, namely,* $\|x_0 - x_*\| \leq \epsilon(r)$ *and* $\|B_0 - B_*\|_* \leq \delta(r)$*, where* $\epsilon(r)$ *and* $\delta(r)$ *are given by Theorem* 12 *for some* $r \in (0, 1)$*.*

Proof. The proof has two parts.

(i) We first prove that the limit (36) holds for $\{B_i^M\}$ and $\{\bar{B}_i^P\}$.

(ii) We then use the convergence of $\{\bar{B}_i^P\}$ and a continuity argument to show that $\{B_i^P\}$ has a similar convergence behavior.

Part (i) of the proof relies on Lemma 13 and, for this reason, is identical for both $\{B_i^M\}$ and $\{\bar{B}_i^P\}$ sequences. B_i denotes either B_i^M or $\bar{B}_i^P$. Lemma 13 implies

$$\|B_i - B_*\|_*^2 - \|B_i' - B_*\|_*^2 \leq \|B_i - B_*\|_*^2 - \|B_{i+1} - B_*\|_*^2 + C_2 r^i,$$

where C_2 is one of the constants provided by Lemma 13. We sum this over i, and, based on Theorem 12, we obtain

$$\sum_{i=0}^{\infty} \left[\|B_i - B_*\|_*^2 - \|B_i' - B_*\|_*^2\right] \leq \|B_0 - B_*\|_*^2 + C_2 \sum_{i=0}^{\infty} r^i < \infty.$$

Therefore $\lim_{i\to\infty} \left[\|B_i - B_*\|_*^2 - \|B_i' - B_*\|_*^2\right] = 0$. Then Lemma 9 implies that $z_i^T z_i$ and $z_i^T B_*^{-0.5} B_i B_*^{-0.5} z_i$ converge to 1. These are exactly

$$\gamma_i := \frac{s_i^T B_i B_*^{-1} B_i s_i}{s_i^T B_i s_i} \to 1 \quad \text{and} \quad \theta_i := \frac{s_i^T B_i B_*^{-1} B_i B_*^{-1} B_i s_i}{s_i^T B_i s_i} \to 1.$$

Furthermore, we observe that

$$\frac{s_i^T (B_i - B_*) B_*^{-1} B_i B_*^{-1} (B_i - B_*) s_i}{s_i^T B_i s_i} = \gamma_i - 2\theta_i + 1 \to 0 \text{ as } i \to \infty.$$

However, this expression is exactly

$$\frac{\|B_i^{0.5} B_*^{-1} (B_i - B_*) s_i\|^2}{\|B_i^{0.5} s_i\|^2} \to 0,$$

which, by the boundedness of $\{B_i\}$ (Theorem 12) and of B_* (Assumption 5), implies the asymptotic convergence limit (36):

$$\frac{\|(B_i - B_*) s_i\|}{\|s_i\|} \to 0.$$

This completes the proof of part (i).

For the proof of part (ii), we observe that B_i^P is by definition a $K(x_{i+1}) - K(x_i)$ perturbation of $\bar{B}_i^P$. Then Theorem 12 and the continuity of K given by Assumption 4 imply that $\{B_i^P\}$ also satisfies the limit above. ∎

3.4. Hereditary (strong) positive definiteness. We proved that the inverses of the updates B^M and B^P are bounded. This proof implies that the updates are p.d. and their eigenvalues are bounded away from zero. The consequence is that asymptotically, regularizations and extra factorization are not needed for the updates, a desirable property since it reduces the computational cost.

3.5. Discussion for quadratic functions. We investigate the behavior of our linesearch quasi-Newton method for strongly convex quadratic functions since this may provide insight into the behavior for the general nonquadratic case. For such functions the known and unknown Hessian parts are constant and denoted by K and U, and $Q := \nabla^2 f(x) = K + U$ is p.d. We immediately observe that our structured update formulas (16) and (17) are identical and reduce to the unstructured BFGS formula

$$B_+ = \mathbb{B}(s, y, B), \tag{52}$$

where y is the variation in the gradient for the (unstructured) function and is defined in (2). Nevertheless, our linesearch algorithm starts with a different initial BFGS approximation, namely, with $K + A_0$, where A_0 is the initial approximation for U. Consequently, our algorithm will produce different iterations from those of an unstructured counterpart. The number of iterations of our algorithm is given by the following result.

PROPOSITION 15. *The quasi-Newton structured BFGS algorithm with an exact linesearch and A_0 as the initial approximation will generate a sequence of iterates identical to that of the conjugate gradient method (CG) preconditioned with $K + A_0$.*

Proof. Formulas like (52) produce a certain conjugacy property among the quasi-Newton iterates for strongly convex functions when *exact* linesearches are used. This translates into iterations identical to the preconditioned CG preconditioned with B_0; see [20, Theorem 6.4] and the discussion afterwards. Our proposition follows from the fact that $B_0 = K + A_0$ in the case of our structured algorithm. □

Let us consider the situation when the unknown Hessian U is of rank r. We take $A_0 = 0$. The preconditioned CG iteration count will be at most the number of distinct eigenvalues of the preconditioned matrix $(K + A_0)^{-1}(K + U) = I + K^{-1}U$, which is bounded from above by $r + 1$. This points toward a dramatically improved performance of the structured quasi-Newton method over the unstructured counterpart (which would "blindly" start at a multiple of identity and would take n iterations [20, Theorem 6.4]).

One can argue that the unstructured BFGS algorithm can be started with $B_0 = K$ and, thus, would reproduce the behavior of the structured algorithm. While this is a valid argument for the quadratic case, it does not generalize to nonquadratic nonlinear functions. In this case $B_0 = K(x_0)$ may not be p.d. and, hence, not suitable as a general initial choice for the initial approximation B_0.

4. Numerical experiments.

In this section we investigate the performance (number of iterations and function evaluations) of our structured updates and perform comparisons with Newton and unstructured BFGS methods. We also study the behavior of our updates on nonlinear problems that have low-rank missing Hessians $\nabla^2 u(x)$, along the lines of the discussion following Proposition 15.

4.1. Test problems. We use the CUTEr test set [13] specified in AMPL [12]. Of 73 unconstrained problems with nonlinear objectives in CUTEr, we consider 61 problems that have fewer than 5000 variables, a threshold dictated by the limitations of our MATLAB implementation based on dense linear algebra. Of these 61 small- to medium-scale problems we have selected the problems on which IPOPT [23] requires fewer than 1000 iterations to reach optimality. The selection yielded 59 test problems. To mimic the unavailability of Hessian information, we modified the test problems to include a term $u(x)$ in (1) along the lines of the following two experiments.

Experiment 1. Given the CUTEr objective $f(x)$ and a *ratio* parameter $\eta \in [0,1]$, we consider the problem of minimizing $\bar{f}(x) = \eta \cdot k(x) + (1-\eta) \cdot u(x)$, where $k(x) = u(x) = f(x)$. Our structured BFGS algorithms use only $\nabla^2 k(x)$, the unstructured BFGS does not use any Hessian term, and the Newton method uses the full Hessian $\nabla^2 \bar{f}(x)(= \nabla^2 f(x))$.

Experiment 2. Given the CUTEr objective $f(x)$, we set $u(x) = \sum_{l\in\mathcal{L}} \log[([x]_l - [x_*]_l)^2 + 0.25]$, where x_* is the minimizer of $f(x)$, $\mathcal{L}$ is an index set specifying the variables included in $u(x)$, and, for any given x, $[x]_l$ denotes its lth component. In this experiment we minimize $\bar{f}(x) = k(x)+u(x)$, where $k(x) = f(x)$. The term $u(x)$ is a smooth nonlinear nonconvex function having the same minimizer x_* as $f(x)$ (when $\mathcal{L} = \{1,\dots,n\}$). We observe that x_* is a local minimizer of $\bar{f}(x)$ for any subset $\mathcal{L}$ of $\{1,\dots,n\}$. In this experiment, $u(x)$ is taken to be *full-space*, that is, $\mathcal{L} = \{1,\dots,n\}$, as well as *subspace*, that is, $\mathcal{L} \subset \{1,\dots,n\}$ and of low cardinality, $|\mathcal{L}| = O(1)$. The subspace approach is used to study the performance of our structured BFGS formulas on problems for which the unknown Hessian is of low rank and to investigate whether the theoretical result given by Proposition 15 for quadratic objectives is observed empirically for nonquadratic nonlinear objective functions.

When the term $u(x)$ is defined on a subspace, we compute subspace variants of the structured updates (14) and (15) of the form

$$A^M_{i+1} = A^M_i + K^L_i - K^L_{i+1} - \frac{\left[A^M_i + K^L_i\right](s^L_i)(s^L_i)^T\left[A^M_i + K^L_i\right]}{(s^L_i)^T\left[A^M_i + K^L_i\right](s^L_i)} + \gamma^L_i(y^L_i)(y^L_i)^T, \tag{53}$$

$$B^M_{i+1} = K_{i+1} + PA^M_{i+1}P^T, \tag{54}$$

and, respectively,

$$A^P_{i+1} = A^P_i - \frac{\left[A^P_i + K^L_{i+1}\right](s^L_i)(s^L_i)^T\left[A^P_i + K^L_{i+1}\right]}{(s^L_i)^T\left[A^P_i + K^L_{i+1}\right](s^L_i)} + \gamma^L_i(y^L_i)(y^L_i)^T, \tag{55}$$

$$B^P_{i+1} = K_{i+1} + PA^P_{i+1}P^T, \tag{56}$$

where $K^L_i = PK(x^i)P^T$, $s^L_i = Ps_i$, $y^L_i = \bar{y}_i + K^L_{i+1}s^L_i$, and $\gamma^L_i = ((y^L_i)^T(s^L_i))^{-1}$. Also, $P : \mathbb{R}^n \to \mathbb{R}^{|\mathcal{L}|}$ denotes the projection (matrix) from the full space to the subspace considered by $u(x)$. As an alternative to the update (53)–(54) above, a subspace version of the modified formula (27) can also be derived. Theorem 2 would allow us to use the modified MT linesearch of section 3.1.1. However, in our experiments, the update (53)–(54) (equipped with the inertia correction regularization) showed superior performance. For this reason we only report the performance of (53)–(54).

Table 1 lists a summary of the algorithms and experimental setups used in the performance investigation of the structured BFGS algorithms of this paper.

4.2. Implementation details. The numerical experiments were performed on a desktop with a quad-core 2.66 GHz Intel CPU Q8400 and with 3.8 GB memory, using MATLAB R2013a as an environment for implementation. We implemented Newton's method, the (unstructured) BFGS method [20], and structured BFGS methods; all the algorithms use the linesearch algorithm of Moré and Thuente [18] (see Appendix A for a short discussion) from the MATLAB implementation of O'Leary [21], with the exception of *SBFGS_M*, which uses the variant of this linesearch presented in Appendix A. All algorithms are stopped when the norm of the gradient is less than

Table 1

Summary of the six different methods. As discussed in the text, all the methods use the same linesearch mechanism; they differ only in the type of Hessian approximation used.

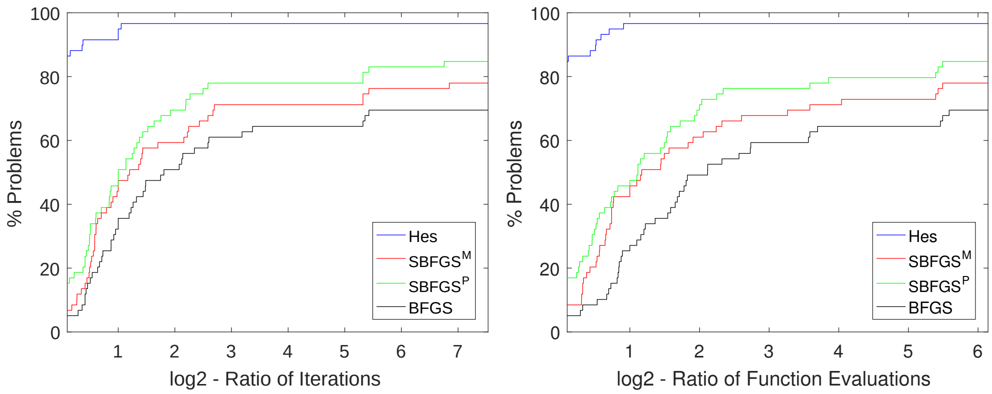
Acronym	Description
Hes	Newton method with exact Hessian. Regularization is required to ensure positive definiteness.
BFGS	Unstructured BFGS. Positive definiteness is guaranteed by the linesearch.
SBFGS_M	*Structured* BFGS with (full-space) update B^M from (27). Positive definiteness is guaranteed by the linesearch.
SubBFGS_M	*Structured* BFGS with *subspace* update B^M from (54). Regularization is required to ensure positive definiteness. (Experiment 2 only)
SBFGS_P	*Structured* BFGS with (full-space) update B^P from (31). Regularization is required to ensure positive definiteness.
SubBFGS_M	*Structured* BFGS with *subspace* update B^P from (56). Regularization is required to ensure positive definiteness. (Experiment 2 only)

10^{-6}, the number of iterations reaches 1000, or an abnormal situation occurs (i.e., ascent directions due to round-off errors, AMPL errors, objective not bounded from below during the linesearch).

In the linesearch we initially set the upper bound on the step to $\alpha_{max} \leftarrow 2$. The linesearch can terminate at the upper bound α_{max}, and the strong Wolfe conditions do not hold. This situation can occur when the objective function is not bounded from below along the search direction and also, in practice, when the search direction is "short" (for example, caused by distorted quasi-Newton approximations). In these situations, we increase $\alpha_{max} \leftarrow 100$ and restart the linesearch to rule out a short search direction. In a couple of instances the second linesearch fails, in which case we conclude that the function is likely unbounded and stop the algorithm.

In order to guarantee that the search direction is of descent, the linesearch procedure requires that the Hessian (for Newton's method) or Hessian secant approximation (for BFGS-based algorithms) is p.d. For Newton's method, whenever the Hessian is not p.d., we use an inertia-correction mechanism present in IPOPT [23], in which we repeatedly factorize and increase the diagonal perturbation until the (perturbed) Hessian becomes p.d. We remark that the unstructured BFGS algorithm does not need inertia regularization since the unstructured update is p.d. at points satisfying the Wolfe conditions. For our (full-space) structured BFGS update B_+^M, the MT linesearch was modified to find a point satisfying the conditions of Theorem 2, as elaborated in Appendix A; for the (full-space) structured BFGS update B_+^P, we use the variant of the inertia-correction technique that is discussed in section 3.1.2. For the subspace structured BFGS updates (54) and (56) we use inertia regularization. When the search direction is not of descent, which can occur for all algorithms because of round-off errors, we terminate with an error message.

The subspace structured updates A_{i+1}^M and A_{i+1}^P given by (53) and (55) are set to zero whenever our implementation detects that there is no change in the gradient of the unknown part and in the variables in the subspace, namely, $s_i^L = 0$ and $y_i^L = 0$ numerically. This occurs in Experiment 2, when $u(x)$ is defined on a subspace and it is a manifestation of the fact that the missing curvature $\nabla^2 u(x)$ is zero.



(a) for # of iterations (b) for # of function evaluations

FIG. 1. *Dolan–Moré performance profiles for Experiment* 1 *with ratio* $\eta = 0.25$.

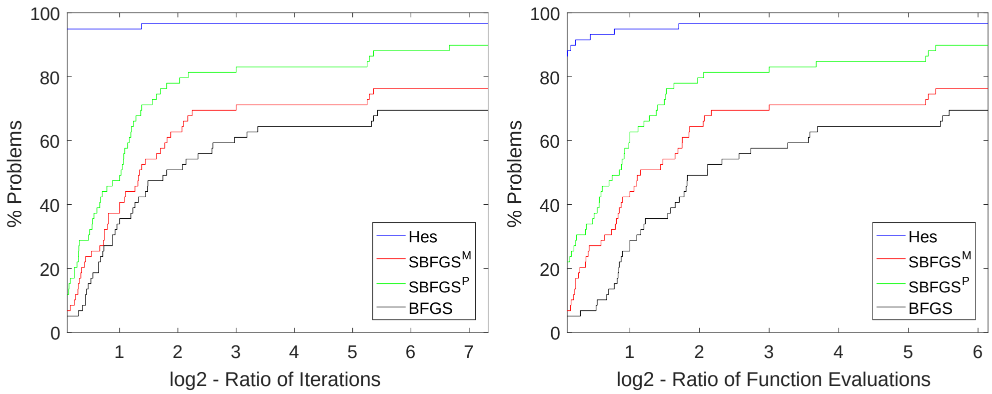
4.3. Discussion of results. We show comparisons of the six algorithms in Figures 1–3. We employ the performance profiling technique of Dolan and Moré [10]. For each algorithm this technique assigns to each problem a performance ratio defined as the ratio of *an algorithm's performance*, namely, the number of iterations ((a) figures) or the number of function evaluations ((b) figures), and *the best performance of any algorithm* on the problem in question. If an algorithm does not solve the problem, then the performance ratio is set to infinity for that problem. The Dolan–Moré performance profile consists of plotting the percentage of problems (y-axis) for which the performance ratio is within a factor τ (x-axis) of the best possible ratio. For example, for $\tau = 1$ ($\log_2(\tau) = 0$), Figure 1 indicates that the Newton method is the best solver on more than 85% and $SBFGS^P$ on a bit more than 15% of the problems in terms of iteration count.[1] As another example, $\tau = 8$ ($\log_2(\tau) = 3$) indicates that $SBFGS^P$ solves a bit more than 75% of the problems in an iteration count that is no more than 8 times the iteration count of the best algorithm; similarly, $SBFGS^M$ and $BFGS$ solve about 70% and 60% within this performance factor.

Figures 1 and 2 show the performance profiles for Experiment 1 for $\eta = 0.25$ and $\eta = 0.75$, respectively. Unsurprisingly, *Hes* (Newton's method) dominates all other methods. However, the structured BFGS methods outperform the unstructured BFGS method in both metrics (the number of iterations and the number of function evaluations). The performance gap between the structured and unstructured BFGS methods seems to be larger for $\eta = 0.75$ than for $\eta = 0.25$, possibly as the result of "more" Hessian information used by the structured updates. Also, we remark that the B^P update slightly outperforms B^M.

We point out that for some classes of problems, for example, those with a large number of variables, the cost of factorizations needed by the inertia regularization technique necessary for B^P can be higher than the difference in the number of iterations/function evaluations over B^M, and thus B^M can outperform B^P in terms of execution time.

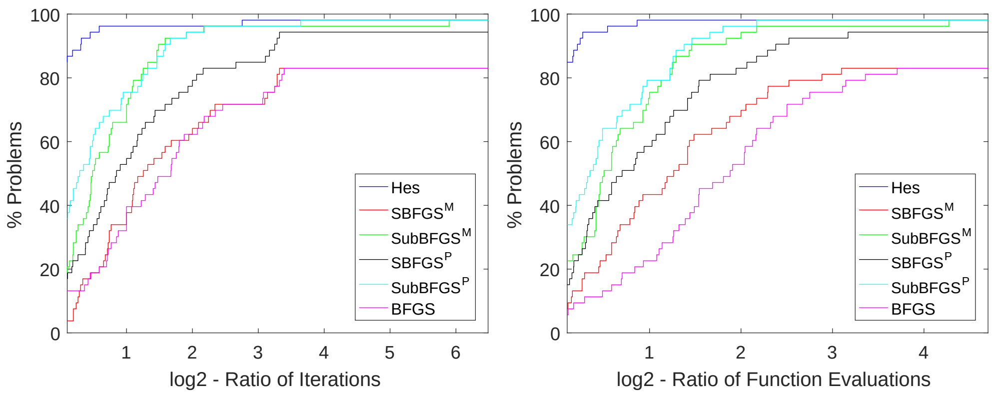
For Experiment 2 we consider subspaces of two (randomly chosen) variables for $u(x)$. Then we compare the six methods on the problems that can be solved by Hes or

[1]The two algorithms have identical performance on a subset of the problems, in case the reader wonders why their total percentage is greater than 100%.



(a) for # of iterations (b) for # of function evaluations

FIG. 2. *Dolan–Moré performance profiles for Experiment* 1 *with ratio* $\eta = 0.75$.



(a) for # of iterations (b) for # of function evaluations

FIG. 3. *Experiment* 2 *with full-space and subspace structured updates* B_+^M *and* B_+^P*, as well as unstructured BFGS and Newton's method.*

$BFGS$, for a total of 53 test problems. Figure 3 shows the Dolan–Moré performance profile for this comparison. We observe that the subspace update strategies perform better than the full-space counterparts, as we anticipated. However, the subspace structured updates B^M and B^P perform about the same. We also observe that, as in Experiment 1, the four structured BFGS approaches perform better than the unstructured BFGS algorithm.

Detailed results for each CUTEr test problem and for each algorithm are displayed in Appendix B in Table 2 for Experiment 1 with $\eta = 0.5$ and in Table 3 for Experiment 2. In these tables, column Stat shows the convergence status: 1 indicates success; 2 denotes that the maximum number of iterations has been reached; and 0 denotes the failure of convergence because of the two conditions discussed in the preceding section, i.e., unboundedness of the objective function along the search direction or the occurrence of an ascent search direction. Columns #Iter and #Func show the numbers of iterations and function evaluations, respectively.

5. Conclusion and further developments. We derived structured secant formulas that incorporate available second-order derivative information. We also analyzed the convergence (global and local) properties of a linesearch algorithm equipped with the structured formulas. In particular we have shown convergence to stationary points and superlinear local convergence properties. Our numerical experiments were aimed at investigating the benefits of incorporating Hessian terms in the secant update formula and, indeed, show that the performance (number of iterations and number of function evaluations) of the unstructured BFGS can be improved; furthermore, the improvement can be considerable when the Hessian terms that are not known or available have low rank.

In this work we have used a straightforward rank-two update recursive computation for the Hessian approximation matrix that results in a *dense* linear system; obviously, this approach has severe computational limitations for large-scale problems. Limited-memory methods equipped with compact (e.g., low-rank) representations for the dense matrix in the idea of [5] is a natural direction of further development. A plausible alternative direction to investigate is the use of iterative methods for the solution of the quasi-Newton linear system since, as we discussed in section 3.5, the known part of the Hessian is a natural preconditioner for the quasi-Newton linear system. We mention that further development is needed to extend or adapt such computational techniques to *constrained* problems.

Appendix A. A variant of Moré–Thuente [18] linesearch for our structured update B^M. Here we show that with a simple modification of its termination criteria, the Moré–Thuente (MT) numerical linesearch procedure [18] is capable of finding the endpoints α_* and α^* that maintain the positive definiteness of the structured update (27). Namely, we show that the proposed variant of MT linesearch finds an α^* that satisfies the Wolfe conditions (22) and (23) and the positive curvature condition $\bar{\gamma}(\alpha^*, \alpha_*) > 0$, which ensures the positive definiteness of the structured update (27) as per Theorem 2.

The MT linesearch aims at finding an acceptable steplength α^* for which the so-called strong Wolfe conditions are satisfied, namely inequality (22) and

$$|\phi'(\alpha^*)| \leq c_2 |\phi'(0)|, \tag{57}$$

where $0 < c_1 < c_2 < 1$. We remark that under the premise that $\phi'(0) < 0$, i.e., the search direction p is of descent, (57) implies (23). The main idea of the MT linesearch algorithm is to generate a sequence of nested closed intervals $\{I_t\}$ and a sequence of steplengths $\alpha_t \in I_t$ such that $\{\alpha_t\}$ converges to a point in

$$T(c_1) = \{\alpha > 0 : \phi(\alpha) \leq \phi(0) + c_1 \alpha \phi'(0),\ |\phi'(\alpha)| \leq c_1 |\phi'(0)|\} \tag{58}$$

or to one of the safeguard bounds α_{min}, in which case $\alpha^* = \alpha_{min} \in T(c_1)$, or α_{max}, in which case the problem is (likely to be) unbounded from below. Once $\alpha_t \in T(c_1)$, we remark that $\alpha^* = \alpha_t$ satisfies the (strong) Wolfe conditions (22) and (57) since $c_1 < c_2$.

At each iteration t of the MT linesearch, a trial steplength is computed in I_t as an approximate minimizer of a quadratic or cubic interpolant of ϕ. The next interval I_{t+1} is either I_t or the subinterval of I_t given by the trial steplength and one of the endpoints of I_t. Coupled with a bisection-based reduction of the length of I_t for the cases when I_t does not shrink during a couple of consecutive iterations, this approach yields a convergent sequence $\{\alpha_t\}$; moreover, this sequence is finite, except for pathological cases [18, Theorem 2.3].

Key to ensuring that $\{\alpha_t\}$ converges to a point in $T(c_1)$ is the fact that $I_t \cap T(c_1)$ is not empty for all t. This is guaranteed by requiring and maintaining certain conditions on the endpoints of each interval I_t. Specified in terms of the function

$$\psi(\alpha) = \phi(\alpha) - \phi(0) - c_1 \alpha \phi'(0), \tag{59}$$

these conditions are given by the following theorem.

THEOREM 16 (see [18, Theorem 2.1]). *Let I be a closed interval with endpoints α_l and α_u satisfying*

$$\psi(\alpha_l) \leq \psi(\alpha_u), \quad \psi(\alpha_l) \leq 0, \quad \text{and} \quad \psi'(\alpha_l)(\alpha_u - \alpha_l) < 0. \tag{60}$$

Then there is an $\alpha^* \in I$ with $\psi(\alpha^*) \leq \psi(\alpha_l)$ and $\psi'(\alpha^*) = 0$. In particular, $\alpha^* \in T(c_1) \cap I$.

Also, a proof that conditions (60) are maintained for each I_t (or the algorithm stops at the safeguards α_{min} or α_{max} or at a point that satisfies strong Wolfe conditions) is provided in Theorem 2.2 of [18].

Our variant of the MT linesearch for the update B^M builds on the fact that the conditions (60) imply that each interval I_t must also contain steplengths in $T(c_1)$ where ϕ has positive curvature, as shown in the following proposition.

PROPOSITION 17. *If the endpoints α_l and α_u of I satisfy* (60)*, then there must exist $\alpha^* \in T(c_1) \cap I$ such that $\phi''(\alpha^*) > 0$.*

Proof. As in the proof of [18, Theorem 2.1], one can show that the global minimizers of ψ on I are in the interior of I. Furthermore, any such global optimizer α_g satisfies $\psi(\alpha_g) < 0$, $\psi'(\alpha_g) = 0$, which imply that $\alpha_g \in T(c_1)$.

We remark that $\psi''(\alpha_g) \geq 0$ by standard second-order necessary conditions for optimality. If at least one global minimizer α_g satisfies $\psi''(\alpha_g) > 0$, then we can take $\alpha^* = \alpha_g$ and the proof is completed (observe that ψ'' is identical to ϕ'').

Otherwise, if all global minimizers α_g satisfy $\psi''(\alpha_g) = 0$, then any (interval) of them can be used to construct a nearby α^* such that $\alpha^* \in T(c_1)$ and $\phi''(\alpha^*) > 0$ using continuation arguments.

First, we observe that ψ' must take positive values between α_g and α_u since $\psi(\alpha_u) > \psi(\alpha_g)$.[2] Then the continuity of ψ' implies that for any given $\varepsilon > 0$ there exists $\delta > 0$ such that $\psi'(\alpha_g) = 0 < \psi'(\alpha) < \varepsilon$ for all $\alpha \in (\alpha_g, \alpha_g + \delta)$.[3] In particular, for $\varepsilon = -c_1\phi'(0) > 0$, this implies that $0 < \psi'(\alpha) < -c_1\phi'(0)$ and, consequently, based on expression (59) of ψ, that $0 < \phi'(\alpha) - c_1\phi'(0) < -c_1\phi'(0)$ for all $\alpha \in (\alpha_g, \alpha_g + \delta)$. One can easily verify that this last inequality implies $|\phi'(\alpha)| \leq c_1|\phi'(0)|$ for all $\alpha \in (\alpha_g, \alpha_g + \delta)$. Second, as mentioned above we have that $\psi(\alpha_g) < 0$, which implies based on (59) and continuity of ψ that $\phi(\alpha) \leq \phi(0) + c_1\alpha\phi'(0)$ for any α in a nonempty subinterval of $(\alpha_g, \alpha_g + \delta)$. We have just proved that any α in this subinterval is in $T(c_1)$. Finally, since α_g is a minimum of ψ,[3] we remark that this subinterval must contain a point α^* at which ψ'' (and hence ϕ'') is positive. ∎

Our variant of the MT algorithm uses the MT trial point α_t as a candidate for α^* and the left endpoint of I_t as a candidate for α_*. The *only* modification of the MT linesearch consists of requiring $\bar{\gamma}(\alpha^*, \alpha_*) > 0$ in addition to the termination criteria of the original MT algorithm.

[2] We assume that $\alpha_l < \alpha_g < \alpha_u$. The case $\alpha_l > \alpha_g > \alpha_u$ can be proved using identical arguments.

[3] In case there is an interval of global minimizers, we take α_g to be the right endpoint of this interval.

Since the length of I_t is driven to zero and Proposition 17 guarantees the existence of an α^* with $\phi''(\alpha^*) > 0$, the steplengths $\alpha_t \in T(c_1)$ generated by the MT algorithm eventually converge to such α^*. Similarly, α_* can be arbitrarily close to α^* since the intervals I_t shrink. Under these conditions, namely $\phi''(\alpha^*) > 0$ and α_* being close to α^*, one can easily prove that the additional stopping criterion $\bar{\gamma}(\alpha^*, \alpha_*) > 0$ is eventually satisfied (see the proof of Theorem 2 for an identical argument). Therefore, we conclude that our variant MT linesearch is a convergent numerical scheme.

Finally, we mention that while our MT linesearch variant is convenient to implement, especially by modifying existing implementations of the MT algorithm (we used the MATLAB implementation by O'Leary [21]), and inherits the robustness of its original counterpart, *structured* linesearch algorithms that make use of the existing Hessian information may have improved performance (e.g., fewer linesearch iterations) and may be worth further investigation.

Appendix B. Performance results. Detailed listings of the performance of our methods for CUTEr problems are given in Table 2 for Experiment 1 and in Table 3 for Experiment 2.

TABLE 2
CUTEr results of the four methods. Compute $u(x)$ by Experiment 1 with $\eta = 0.5$.

	Hes			$BFGS$			$SBFGS^M$			$SBFGS^P$		
Prob	Stat	# Iter	# Func	Stat	# Iter	# Func	Stat	# Iter	# Func	Stat	# Iter	# Func
arglina	1	1	1	1	2	3	1	1	1	1	1	1
arglinb	1	2	2	0	1	71	1	2	3	1	2	3
arglinc	1	2	2	0	1	69	1	2	3	1	2	3
bard	1	7	7	1	19	24	1	19	19	1	21	26
bdqrtic	1	10	10	0	119	290	0	37	107	0	15	58
beale	1	9	10	1	14	18	1	12	12	1	12	12
biggs6	1	38	51	1	72	81	1	34	35	1	60	62
box3	1	8	9	1	20	21	1	21	21	1	19	19
brownal	1	7	7	1	9	15	1	9	13	1	10	13
brownbs	1	8	8	1	11	28	1	11	27	1	13	16
chainwoo	1	90	145	2	1000	3217	2	1000	1088	1	261	336
chnrosnb	1	52	92	1	174	325	1	171	186	1	76	104
cube	1	28	36	1	27	44	1	40	50	1	34	40
deconvu	1	44	50	1	103	157	1	50	65	1	41	46
denschnb	1	6	12	1	8	10	1	7	8	1	7	8
denschnc	1	11	12	1	16	22	1	18	21	1	12	12
denschnd	1	35	52	1	80	116	1	85	89	1	88	90
denschne	1	10	11	1	24	39	1	22	33	1	22	33
denschnf	1	6	6	1	11	26	1	8	8	1	7	7
eigenals	1	23	27	1	97	135	1	108	111	1	104	107
eigenbls	1	118	154	1	520	668	1	523	540	1	170	172
engval2	1	16	20	1	31	46	1	33	38	1	29	34
errinros	1	29	40	1	264	473	0	226	288	1	80	100
expfit	1	6	9	1	11	18	1	12	17	1	12	17
extrosnb	1	0	0	1	0	0	1	0	0	1	0	0
genhumps	1	64	130	1	71	89	1	81	105	1	106	175
genrose	1	894	1370	2	1000	2179	2	1000	1472	1	990	1491
growthls	0	74	151	1	1	9	0	121	185	1	100	108
gulf	1	21	27	1	28	39	1	30	33	1	30	36
hatfldd	1	20	21	1	27	34	1	30	33	1	29	31
hatflde	1	19	21	1	27	37	1	28	31	1	26	30
heart6ls	1	905	1139	2	1000	1361	2	1000	1246	1	691	704
heart8ls	1	74	99	2	1000	1321	2	1000	1228	2	1000	1002
helix	1	10	12	1	28	35	1	24	27	1	23	26
himmelbf	1	10	10	0	30	109	1	32	37	1	29	36
humps	1	127	302	0	0	0	0	11	1042	0	6	1022
kowosb	1	8	18	1	28	32	1	27	29	1	27	27
mancino	1	5	5	0	93	224	1	33	37	1	13	17
meyer3	0	207	298	0	311	535	0	329	467	0	283	327
msqrtals	1	32	55	2	1000	2099	2	1000	1005	2	1000	1003
msqrtbls	1	28	39	2	1000	2109	2	1000	1005	2	1000	1003
nonmsqrt	1	143	202	0	532	699	0	142	177	0	118	123
osborneb	1	19	22	1	53	72	1	50	62	1	63	71
palmer5c	1	1	1	1	6	12	1	13	13	1	13	13
palmer6c	1	1	1	1	43	48	1	43	44	1	43	44
palmer7c	1	1	1	1	40	44	1	39	40	1	39	40
palmer8c	1	1	1	1	41	45	1	40	41	1	40	41
penalty1	1	40	42	1	414	544	1	71	78	1	54	55
penalty2	1	19	20	0	0	0	0	60	111	0	29	74
pfit1ls	1	60	81	1	364	480	1	230	285	1	135	145
pfit2ls	1	179	244	1	541	722	1	72	94	1	43	51
pfit3ls	1	217	290	0	0	0	1	324	411	1	584	595
pfit4ls	1	284	387	0	0	0	1	389	487	1	569	581
rosenbr	1	22	28	1	36	52	1	27	33	1	27	30
s308	1	9	10	1	14	18	1	14	14	1	11	11
sensors	1	30	54	0	190	670	1	43	52	1	32	41
sineval	1	44	61	1	70	101	1	54	69	1	53	61
watson	1	12	12	1	61	80	1	56	56	1	53	53
yfitu	1	38	48	1	63	86	1	50	63	1	46	50

TABLE 3
CUTEr results of the six methods for Experiment 2.

	Hes			*BFGS*			$SBFGS^M$			$SubBFGS^M$			$SBFGS^P$			$SubBFGS^P$		
Prob	Stat	#Iter	# Func	Stat	#Iter	# Func	Stat	#Iter	# Func	Stat	#Iter	# Func	Stat	#Iter	# Func	Stat	#Iter	# Func
arglina	1	5	5	1	5	7	1	6	6	1	5	5	1	6	6	1	5	5
arglinb	1	3	3	1	2	3	1	8	8	0	17	27	1	8	8	0	17	27
arglinc	1	4	4	1	4	6	1	9	9	1	15	18	1	9	9	1	15	18
bard	1	7	7	1	14	19	1	11	11	1	14	14	1	7	7	1	8	8
bdqrtic	1	10	10	0	110	309	0	47	96	1	10	10	1	13	13	1	10	10
beale	1	7	8	1	9	13	1	8	11	1	8	11	1	7	10	1	7	10
biggs6	1	17	20	1	45	58	1	37	48	1	20	30	1	40	45	1	18	27
box3	1	5	5	1	9	11	1	8	8	1	6	6	1	9	9	1	7	7
brownal	1	15	16	1	15	21	1	11	11	1	15	15	1	11	12	1	15	15
brownbs	1	8	8	1	16	33	1	17	32	1	17	32	1	12	13	1	12	13
chainwoo	1	87	136	2	1000	3189	2	1000	1035	1	70	115	1	131	169	1	84	128
chnrosnb	1	54	94	1	169	320	1	204	211	1	53	95	1	57	78	1	56	102
cube	1	21	25	1	17	27	1	28	34	1	28	34	1	23	31	1	23	31
deconvu	1	16	16	1	88	138	1	137	137	1	22	22	1	144	144	1	25	25
denschnb	1	7	9	1	7	9	1	8	8	1	8	8	1	7	9	1	7	9
denschnc	1	10	10	1	17	24	1	17	19	1	17	19	1	10	10	1	10	10
denschnd	1	32	45	1	78	118	1	69	80	1	42	68	1	52	56	1	46	55
denschne	1	9	10	1	25	41	1	18	28	1	13	16	1	16	24	1	11	14
denschnf	1	6	6	1	12	27	1	9	9	1	9	9	1	6	6	1	6	6
eigenals	1	23	27	1	97	135	1	109	112	1	24	28	1	103	104	1	24	28
eigenbls	1	60	76	1	469	601	1	494	502	1	56	68	1	354	354	1	57	71
engval2	1	14	18	1	28	43	1	28	31	1	20	24	1	28	32	1	27	31
errinros	1	22	36	1	231	444	0	238	324	1	25	38	1	25	34	1	26	39
expfit	1	6	9	1	11	19	1	10	14	1	10	14	1	10	14	1	10	14
extrosnb	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
growthls	1	54	71	1	8	39	1	78	89	1	477	754	1	69	82	1	100	123
gulf	1	11	26	1	18	42	1	18	28	1	19	26	1	18	32	1	15	26
hatfldd	1	9	10	1	21	29	1	19	22	1	18	27	1	12	15	1	17	26
hatflde	1	12	18	1	16	22	1	20	22	1	20	24	1	19	20	1	27	34
heart8ls	1	60	94	1	209	303	1	304	345	1	129	200	1	134	171	1	113	176
helix	1	10	12	1	27	35	1	29	30	1	28	33	1	27	28	1	25	28
himmelbf	0	9	67	1	29	45	0	32	101	1	12	17	1	29	33	1	9	12
kowosb	1	5	6	1	16	21	1	15	16	1	5	8	1	15	19	1	5	8
mancino	1	5	5	0	104	276	1	16	16	1	7	7	1	5	5	1	5	5
msqrtals	1	28	40	2	1000	2103	2	1000	1037	1	30	39	2	1000	1006	1	31	41
msqrtbls	1	26	38	2	1000	2112	2	1000	1034	1	27	38	2	1000	1006	1	28	40
nonmsqrt	1	126	172	0	447	575	0	113	135	1	160	227	0	117	142	1	144	202
osborneb	1	17	22	1	52	70	1	64	68	1	15	19	1	52	55	1	16	19
palmer5c	1	3	3	1	11	17	1	8	8	1	7	7	1	8	8	1	7	7
palmer6c	1	4	8	1	41	51	1	40	46	1	11	19	1	40	46	1	11	19
palmer7c	1	4	10	1	38	51	1	38	45	1	11	19	1	38	45	1	11	19
palmer8c	1	4	9	1	40	51	1	39	44	1	12	22	1	39	44	1	12	22
penalty1	1	40	42	1	338	431	0	95	177	1	54	56	1	45	49	1	40	42
penalty2	1	19	20	0	0	0	0	59	118	1	38	39	1	29	30	1	19	20
pfit2ls	1	22	33	1	98	144	1	47	59	1	99	118	1	84	96	1	100	115
pfit3ls	1	42	61	1	190	273	1	53	68	1	100	121	1	90	117	1	133	168
pfit4ls	1	60	87	0	0	0	1	74	93	1	122	154	1	88	115	1	177	208
rosenbr	1	17	20	1	28	43	1	20	24	1	20	24	1	24	27	1	24	27
s308	1	8	8	1	12	20	1	11	12	1	11	12	1	8	8	1	8	8
sensors	1	31	57	0	188	664	1	74	82	1	34	61	1	65	72	1	34	61
sineval	1	16	26	1	54	73	1	40	48	1	40	62	1	39	53	1	38	51
watson	1	11	11	1	56	74	1	53	54	1	23	24	1	46	46	1	14	14
yfitu	1	37	47	1	60	84	1	45	52	1	51	74	1	48	53	1	37	49

Acknowledgment. The authors would like to thank the anonymous reviewers for their suggestions and careful reading, which helped to improve this work substantially.

REFERENCES

[1] S. Abhyankar, V. Rao, and M. Anitescu, *Dynamic security constrained optimal power flow using finite difference sensitivities*, in 2014 IEEE Power and Energy Society General Meeting—Conference & Exposition, IEEE, 2014, pp. 1–5.

[2] K. Amini and A. Ghorbani Rizi, *A new structured quasi-Newton algorithm using partial information on Hessian*, J. Comput. Appl. Math., 234 (2010), pp. 805–811.

[3] C. G. Broyden, J. E. Dennis, Jr., and J. J. Moré, *On the local and superlinear convergence of quasi-Newton methods*, IMA J. Appl. Math., 12 (1973), pp. 223–245.

[4] R. H. Byrd and J. Nocedal, *A tool for the analysis of quasi-Newton methods with application to unconstrained minimization*, SIAM J. Numer. Anal., 26 (1989), pp. 727–739, https://doi.org/10.1137/0726042.

[5] R. H. Byrd, J. Nocedal, and R. B. Schnabel, *Representations of quasi-Newton matrices and their use in limited memory methods*, Math. Programming, 63 (1994), pp. 129–156.

[6] J. E. Dennis, Jr., H. J. Martinez, and R. A. Tapia, *Convergence theory for the structured BFGS secant method with an application to nonlinear least squares*, J. Optim. Theory Appl., 61 (1989), pp. 161–178.

[7] J. E. Dennis, Jr., and J. J. Moré, *A characterization of superlinear convergence and its application to quasi-Newton methods*, Math. Comp., 28 (1974), pp. 549–560.

[8] J. E. Dennis, Jr., and J. J. Moré, *Quasi-Newton methods, motivation and theory*, SIAM Rev., 19 (1977), pp. 46–89, https://doi.org/10.1137/1019005.

[9] J. E. Dennis, Jr., and H. F. Walker, *Convergence theorems for least-change secant update methods*, SIAM J. Numer. Anal., 18 (1981), pp. 949–987, https://doi.org/10.1137/0718067.

[10] E. D. Dolan and J. J. Moré, *Benchmarking optimization software with performance profiles*, Math. Program., 91 (2002), pp. 201–213.

[11] R. Fletcher, *An optimal positive definite update for sparse Hessian matrices*, SIAM J. Optim., 5 (1995), pp. 192–218, https://doi.org/10.1137/0805010.

[12] R. Fourer, D. M. Gay, and B. W. Kernighan, *AMPL: A mathematical programming language*, in Algorithms and Model Formulations in Mathematical Programming, Springer, Berlin, Heidelberg, 1989, pp. 150–151.

[13] N. I. M. Gould, D. Orban, and P. L. Toint, *CUTEr and SifDec: A constrained and unconstrained testing environment, revisited*, ACM Trans. Math. Software, 29 (2003), pp. 373–394.

[14] A. Griewank and P. L. Toint, *Local convergence analysis for partitioned quasi-Newton updates*, Numer. Math., 39 (1982), pp. 429–448.

[15] O. Güler, F. Gürtuna, and O. Shevchenko, *Duality in quasi-Newton methods and new variational characterizations of the DFP and BFGS updates*, Optim. Methods Softw., 24 (2009), pp. 45–62.

[16] J. Huschens, *Exploiting additional structure in equality constrained optimization by structured SQP secant algorithms*, J. Optim. Theory Appl., 77 (1993), pp. 343–358.

[17] L. Lukšan and E. Spedicato, *Variable metric methods for unconstrained optimization and nonlinear least squares*, J. Comput. Appl. Math., 124 (2000), pp. 61–95.

[18] J. J. Moré and D. J. Thuente, *Line search algorithms with guaranteed sufficient decrease*, ACM Trans. Math. Software, 20 (1994), pp. 286–307.

[19] T. B. Nguyen and M. Pai, *Dynamic security-constrained rescheduling of power systems using trajectory sensitivities*, IEEE Trans. Power Syst., 18 (2003), pp. 848–854.

[20] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed., Springer, New York, 2006.

[21] D. P. O'Leary, *A Matlab implementation of a MINPACK line search algorithm by Jorge J. Moré and David J. Thuente*, http://www.cs.umd.edu/~oleary/software, 1991.

[22] R. Tapia, *On secant updates for use in general constrained optimization*, Math. Comp., 51 (1988), pp. 181–202.

[23] A. Wächter and L. T. Biegler, *On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming*, Math. Program., 106 (2006), pp. 25–57.

[24] P. Wolfe, *Convergence conditions for ascent methods*, SIAM Rev., 11 (1969), pp. 226–235, https://doi.org/10.1137/1011036.

[25] P. Wolfe, *Convergence conditions for ascent methods.* II: *Some corrections*, SIAM Rev., 13 (1971), pp. 185–188, https://doi.org/10.1137/1013035.