A Multi-Secant Limited-Memory BFGS Method

F.V. Gubarev

pSeven SAS,
42 Av. du Général de Croutte, 31100 Toulouse, France
fedor.gubarev@pseven.io

Abstract

We develop multi-secant BFGS-like quasi-Newton updating scheme, which adaptively selects the number of imposed secant conditions and naturally preserves positivity of approximated Hessian. Compact representation and respective limited-memory formulation are also derived. Numerical stability is assured via unconventional damping technique, which symmetrically handles coordinate and gradient differences. Practical relevance of proposed method is demonstrated via extensive numerical tests.

1 Introduction

Consider unconstrained optimization problem $\min_{x\in R^n} f(x)$ with twice continuously differentiable function f having gradient $g = \nabla f$. In iterative context[1] it is standard to assemble m vectors of recent coordinate $s_k = x_k - x_{k-1}$ and gradient $y_k = g_k - g_{k-1}$ differences into the matrices

$$S_m = [s_{k-m+1}, ..., s_k]\,, \qquad Y_m = [y_{k-m+1}, ..., y_k]\,,$$

so that one can write

$$\nabla^2 f\ S_m \ = \ Y_m \ + \ O(||S_m||_F^2)\,, \tag{1}$$

where Frobenius norm is used for concreteness. Close to the solution objective Hessian is expected to be positive definite and the prime subject of quasi-Newton methods is to construct positive approximation B to $\nabla^2 f$ even away from yet to be discovered solution. The only data available for this is encoded in (1), however, these relations are not well-defined because of unknown correction terms. Then there are basically two ways to proceed: one could either attempt to neglect all the corrections and impose (at least a part of) secant conditions $BS_m = Y_m$ exactly or try to treat $BS_m \approx Y_m$ in a least squares sense.

The first approach remains self-consistent only for $m = 1$ (single secant equation), because it demands overlap matrix $O_m \equiv S_m^T Y_m$ to be symmetric positive definite (SPD). Moreover, even with no positivity requirement it is well-known [1] that O_m-symmetry is necessary and sufficient for the existence of symmetric linear operator obeying multiple exact secant equations. In general, neither symmetry nor positivity of overlap are possible. It is not surprising then that major efforts and achievements were concentrated on $m = 1$ case, where symmetry is automatic and positivity might be handled with specific damping schemes. Practically, the most successful quasi-Newton methods are considered to be of celebrated Broyden-Fletcher-Goldfarb-Shanno (BFGS) type [2]-[5] (see, e.g., Refs. [6]-[9]; more recent paper [10] provides further references) and its limited-memory (LBFGS) counterparts [11]-[13], supplemented

[1] Whenever possible we systematically omit iteration index k.

with Powell's damping [14] recipe. Nevertheless, the quest of suitable approaches to (near) exact multi-secant conditions is still alive (see [15]-[18], for instance). In fact, already in [1] previous y-vectors were deformed to get required symmetry/positivity of the overlap, while maintaining most recent secant relation exactly. In the same vein, Ref. [19] (see also [20]) enforced O-symmetry and exactness of only last secant equation, but significantly altered other secants and obtained rather unusual Hessian approximation scheme. The second strategy is pursued in [21], where multiple secants are accounted for via a priori chosen weights in variational approach. Refs. [22, 23] seem to follow [21], but propose to handle most recent relation exactly, while accounting others via similar weights. See also [24] for conceptually analogous treatment of inexact multiple secants.

However, none of above approaches look fully satisfactory either because of ad hoc introduced modification rules for overlap matrix and displacements vectors or due to the explicit appearance of uncertain penalization weights. Ideally, we would like to have self-contained scheme to account for available secant information with no additional parameters apart from those needed to assure numerical stability. It follows from above that none of secant relations are to be imposed exactly, this is not possible with non-infinitesimal steps and no ad hoc manipulations. Hence secant conditions are to be enforced approximately via the penalty terms in suitable variational formulation, which, in turn, is to be constructed in accord with affine invariance principles extensively discussed in [8]. Section 2 demonstrates that with appropriate penalties weight factors could be uniquely fixed and resulting scheme becomes a closed form, self-consistent updating procedure free of any external parameters. In particular, it naturally preserves positive definiteness and minimally violates secant equations in a sense that BS_m and Y_m differ only by right orthogonal rotation derived from current overlap O_m. Furthermore, secant relations automatically become exact when overlap happens to be symmetric positive definite. Structurally, obtained scheme is similar to BFGS update (hence is referred to as MS-BFGS) and reduces to it in the limit of single secant pair with positive curvature. Nevertheless, we might optionally want particular secant condition to be exact from very beginning. This generalization is possible and doesn't notably complicate MS-BFGS scheme, which then starts to resemble nested application of BFGS-like updates.

Numerical stability of MS-BFGS scheme requires to assure sufficient non-singularity of overlap matrix at every update. This becomes subtle in iterative context because previous displacements are already used in approximations and should not be altered. It seems that the only solution is to promote the number of considered secants to be a dynamical quantity: once overlap regularity cannot be assured the number of secant relations sequentially drops down to even one, where conventional damping reasoning applies. In particular, it is legitimate to employ Powell's recipe [14], however, we disfavor it because of explicit asymmetry with respect to coordinate and gradient differences. Instead, we develop in Section 3 manifestly symmetric and affine invariant damping scheme, derived from required boundness of approximated Hessian.

Structural similarity with BFGS updates alludes to consider compact representation of MS-BFGS. It turns out that corresponding derivation is straightforward albeit technically notorious, nevertheless, resulting recurrent equations are relatively simple and akin to those of BFGS. It is then not difficult to develop limited-memory counterpart of MS-BFGS scheme (MS-LBFGS for short), which we discuss in details in Section 4 and which generalizes LBFGS to multi-secants settings.

Finally, to assess practical significance of our developments we performed extensive numerical study of implemented algorithms. First, we embedded MS-LBFGS updating scheme in not much involved Armijo line search based optimization method (see, e.g., Refs. [7]-[9],[25]). Then we analyze the performance of resulting algorithm depending on the parameters of underlying approximation scheme and compare it with a particular baseline. This task requires to carefully select representative set of test problems and the performance measures, which to our taste are to be based on evaluation budget expended either to reach the final convergence or to push objective value below given threshold. We developed a number of respective quality metrics, which we believe are indeed sufficiently representative. As for the test cases, we

considered virtually all unconstrained problems from CUTEst collection [26],[27]. Furthermore, to increase reliability and statistics each test case was once replicated to provide randomized initial point (this is, in fact, not stricty necessary and does not alter the conclusions). As for the baseline algorithm, we considered well-known L-BFGS-B implementation both for its conceptual similarity and ease of adaptation. Brief summary of obtained results is as follows (full details are to be found in Section 5). In all considered cases genuine multi-secant algorithms exhibit significantly better performances, although it is not possible to fairly rank their relative efficiencies. Several methods of MS-LBFGS family utilize, in fact, single secant condition. These reveal subleading performances in all our benchmarks independently on the details of actual implementation (there are a few substantially different variants). Finally, selected baseline appears to be inferior to all MS-LBFGS methods, but perhaps this should not be taken at face value. Ultimately, L-BFGS-B differs from other examined algorithms in too many aspects and its lower efficiency might be caused by some unaccounted reasons.

2 Variational Derivation

In this section we assume m coordinate/gradient displacements are assembled in columns of full rank matrices S_m, Y_m and drop corresponding subscript. Moreover, some initial symmetric positive-definite (SPD) approximations B_0, $H_0 = B_0^{-1}$ to Hessian and inverse Hessian are given. The purpose is to find "improved" SPD matrix $H = B^{-1}$, which accounts for multiple inexact secant relations $HY \approx S$ and is closest to initial guess H_0. Moreover, we will treat particular pair (s, y) specially and require respective secant equation to be exact, $Hy = s$ (at the time being specific origin of (s, y) is irrelevant).

Following [28] (see, e.g., [29, 30] for more references) closeness of B to B_0 is conveniently quantified by strictly convex (in SPD case) function $\mathrm{Tr}\{BH_0 - \ln(BH_0)\}$, while secant conditions are naturally accounted for by penalty term proportional to $||\, W^{1/2}\,(HY - S)\,||_F^2$, where W is some SPD metric, the choice of which is severely restricted by invariance requirement with respect to affine coordinate transformations [8]. Certainly, W is not uniquely determined and following [31] we select $W = B$, which gives rise to the following optimization problem

$$\min_{B^T=B>0} \mathrm{Tr}\{\, B\,H_0 - \ln(B\,H_0) + \alpha\,(S^T B S + Y^T H Y)\, + \lambda^T (Bs - y) + (s^T B - y^T)\lambda\}\,,$$

where $H = B^{-1}$, α is yet undetermined positive parameter and additional exact secant equation is accounted for with vector-valued Lagrange multiplier λ. As noted in [31] (for more details see [28, 32] and references therein) considered problem is convex, hence solution to it could be found from respective optimality conditions

$$H \;=\; \tilde{H}_0 \;+\; \alpha\,(\,SS^T \;-\; UU^T\,)\,, \qquad U = HY\,, \qquad \tilde{H}_0 = H_0 + s\lambda^T + \lambda s^T\,. \tag{2}$$

Closed form expression for H is then derived in two steps: first, defining $G = Y^T H Y$, $G_0 = Y^T H_0 Y$, $U_0 = H_0 Y$, it follows from (2) that

$$G \;=\; \tilde{G}_0 \;+\; \alpha\,(\,O^T O - G^2\,)\,, \qquad \tilde{G}_0 = G_0 + ab^T + ba^T\,,$$

$$a = Y^T s\,, \qquad b = Y^T \lambda\,,$$

solution of which is

$$G \;=\; \frac{1}{2\alpha}(\,\Big[1 + 4\alpha\tilde{G}_0 + (2\alpha)^2 O^T O\Big]^{1/2} \;-\; 1\,)\,.$$

Second, from (2) we also have

$$U \;=\; \tilde{U}_0 \;+\; \alpha(SO - UG)\,, \qquad \tilde{U}_0 \;=\; U_0 + sb^T + \lambda a^T\,,$$

which gives

$$U \;=\; (\tilde{U}_0 \;+\; \alpha\, SO)\,(1+\alpha\, G)^{-1}\,. \tag{3}$$

Plugging everything back to (2) we obtain

$$H \;=\; F \;-\; \alpha\, F\, Y\, \frac{1}{(1+\alpha G)^2}\, Y^T\, F\,, \qquad F \;=\; \tilde{H}_0 + \alpha\, S S^T\,, \tag{4}$$

which in fact defines multi-secant BFGS-like (MS-BFGS for short) update for arbitrary positive value of penalty parameter α. It is then natural to consider the limit $\alpha \to +\infty$ because one wants to impose secant relations as precisely as possible and there is no preferred α-parameter scale. Derivation is straightforward: assume O-matrix non-singularity and define kernel matrices

$$K_L = \sqrt{O^T O}\,, \qquad K_R = \sqrt{O O^T}\,,$$

which are seen to obey $K_L O^{-1} K_R \;=\; O^T$, $K_L O^T \;=\; O^T K_R$ and other similar relations (for instance, $O^{-1} K_R = K_L^{-1} O^T$ is orthogonal). Only $O(\alpha)$ and $O(1)$ terms are to be kept in $(1+\alpha G)^{-2}$, for which we have

$$\begin{aligned} 1+\alpha G \;&=\; \alpha K_L + (1+M)/2 + o(1)\,, \qquad M \;=\; K_L^{-1/2} \tilde{G}_0 K_L^{-1/2}\,, \\ (1+\alpha G)^{-2} \;&=\; \alpha^{-2} K_L^{-1} \left[1 - \frac{1}{\alpha} K_L^{-1/2} (1+M) K_L^{-1/2}\right] K_L^{-1} + o(1)\,. \end{aligned}$$

Then leading order contribution in (4) is seen to vanish identically, while other terms assemble to

$$H \;=\; \tilde{H}_0 - \tilde{U}_0 K_L^{-2} O^T S^T - S O K_L^{-2} \tilde{U}_0^T + SO(...) O^T S^T\,, \tag{5}$$

where dots denote $K_L^{-3/2}(1+M)K_L^{-3/2}$. Trivially, we have $O K_L^{-2} = O^{-T}$, $K_L^{-2} O^T = O^{-1}$ and hence (5) could be represented as

$$H = P^T \tilde{H}_0 P \;+\; SO(...) O^T S^T \;-\; S O^{-T} G_0 O^{-1} S^T\,, \tag{6}$$

where we introduced an oblique projector $P = 1 - Y O^{-1} S^T$. Remaining pieces could easily be simplified to transform (6) into

$$H = P^T \tilde{H}_0 P \;+\; S \left[K_R^{-1} \;+\; \tilde{a}\tilde{b}^T \;+\; \tilde{b}\tilde{a}^T\right] S^T\,, \tag{7}$$

$$\tilde{a} \;=\; O^{-T} Y^T s\,, \qquad \tilde{b} = O^{-T} Y^T \lambda\,. \tag{8}$$

2.1 MS-BFGS without Exact Secant Equation

In case exact secant relation $Hy = s$ is not to be maintained, the above could be summarized as follows: proposed BFGS-like update in case multiple secants $HY \approx S$ are to be kept as close as possible reads

$$H \;=\; P^T\, H_0\, P \;+\; S\, K_R^{-1}\, S^T\,, \tag{9}$$

$$P = 1 \;-\; Y O^{-1} S^T\,, \quad O \;=\; S^T Y\,, \qquad K_R = \sqrt{O O^T}\,.$$

Respective update of Hessian B is straightforward to derive

$$B \;=\; B_0 \;-\; (B_0 S)\, \frac{1}{S^T B_0 S} (B_0 S)^T \;+\; Y\, K_L^{-1}\, Y^T\,, \qquad K_L = \sqrt{O^T O}\,, \tag{10}$$

which follows from generic observation that for $H = P^T H_0 P + S K^{-1} S^T$ its inverse is given by $B = H^{-1} = B_0 - (B_0 S)(S^T B_0 S)^{-1}(B_0 S)^T + Y O^{-1} K O^{-T} Y^T$. A few comments are now in order:

- As is usually the case, "homogeneous" term involving H_0 is irrelevant for secant equations, it is annihilated when forming the product HY. On the other hand, oblique projector P is purely geometric object (see, e.g., [33]-[35]), in particular, it is independent on bases chosen in range spaces of S, Y.

- Despite of explicitly taken limit $\alpha \to +\infty$ secant equations are fulfilled only approximately

$$H\,Y \;=\; S\,\Omega\,, \qquad \Omega \;=\; K_R^{-1}\,O\,, \qquad \Omega^T\Omega = \Omega\,\Omega^T = 1\,, \tag{11}$$

  where mismatch is encoded in orthogonal matrix Ω. Only when overlap $O = S^TY$ happens to be symmetric positive definite we have $\Omega = 1$ and exact secant relations are recovered. Note that range spaces of $H\,Y$ and S coincide, which might be helpful in both the analysis and practical implementations of MS-BFGS algorithms.

- MS-BFGS update (9) always produces SPD matrix provided that overlap O is non-singular. In this respect (9) significantly differs from usual BFGS scheme even in $m = 1$ case: it is sensitive to absolute value $|s^Ty|$ only, treating negative and positive curvatures equally. Importance of such sign-blindness in single secant context was noted already in, e.g., Refs. [36, 37] (see also [31]). In multi-secant settings there is no single curvature to be looked for and apparently the kernels K_R, K_L are the only natural generalizations.

Sufficient nonsingularity of overlap matrix O is crucial for both derivation and numerical stability of the update (9). In turn, there are two sources of (near) singularity: either we have (almost) rank deficient S and/or Y matrices or their range spaces, $\mathcal{R}(S)$ and $\mathcal{R}(Y)$, happen to be (near) orthogonal. First issue does not arise in conventional $m = 1$ case and is specific to multi-secant treatment; second effect might as well be encountered in usual single secant context. As far as rank-deficiencies are concerned, from practical viewpoint it is natural to assume that we have already constructed B/H approximations, such that m secant equations (11) with respect to most recent iterates are fulfilled. Next we are given a pair s, y to update current approximations, however, these are almost linear dependent with columns of S and/or Y. Consider degeneracy in coordinate space: for quadratic underlying function update with $s \in \mathcal{R}(S)$ might be skipped because B action on $\mathcal{R}(S)$ is already known. However, generically update omission is undesirable, one wants to fully utilize most recent information. The only remedy then is to filter out older S-columns until augmented matrix $[S; s]$ becomes of full rank. Therefore, it seems impossible to always maintain exactly m secant equations, depending on underlying iterative algorithm and details of its implementation it might be necessary to diminish the number of served secants down to $m = 1$, where linear independence becomes automatic. Corresponding adaptive scheme to ensure overlap non-singularity is detailed in Section 3.

2.2 MS-BFGS with Exact Secant Equation

Consider the possibility to additionally maintain exact secant equality $Hy = s$, which in turn determines Lagrange multiplier λ via Eqs. (6)-(8). It is natural to take

$$s \;=\; Se_m\,, \qquad y = Ye_m\,, \tag{12}$$

meaning that we insist on exact quasi-Newton relation for last seen (most recent) coordinate/gradient displacements (e_m denotes last column of unit $m \times m$ matrix). Practically, ansatz (12) is very convenient because $P^TS = 0$ and hence we could substitute $\tilde{H}_0 \to H_0$ in (6). Furthermore, since $PY = 0$ only

inhomogeneous (not involving H_0) term is relevant for $Hy = s$ relation, which then is seen to be equivalent to

$$\left[K_R^{-1} \;+\; e_m \tilde{b}^T \;+\; \tilde{b} e_m^T \right] O\, e_m \;=\; e_m \,, \qquad \tilde{b} = O^{-T} b \,, \qquad b = Y^T \lambda \,, \tag{13}$$

where we used $\tilde{a} = e_m$ as it follows from (8) and (12). Solution for b is established in standard way: left-multiply (13) with $(Oe_m)^T$ to get last b-component $b_{[m]} = (1 - K_{L\,[m,m]}/O_{[m,m]})/2$, where bracketed subindices denote indicated vector/matrix element. Then equation (13) gives b-vector explicitly

$$\tilde{b} \;=\; \frac{1}{O_{[m,m]}} \left[(1 - b_{[m]}) e_m - O^{-T} K_L e_m \right] .$$

Plugging this back to (7), using $K_R^{-1} = O^{-T} K_L O^{-1}$ and simplifying we have for terms in square brackets in (7)

$$K_R^{-1} + \tilde{a}\tilde{b}^T + \tilde{b}\tilde{a}^T = [\, 1 - o_m\, e_m^T / O_{[m,m]}]^T\, K_R^{-1}\, [\, 1 - o_m\, e_m^T / O_{[m,m]}] + e_m e_m^T / O_{[m,m]} \,,$$

where $o_m = Oe_m$ denotes last column of O-matrix. Introducing respective projector $\pi = 1 - o_m e_m^T / O_{[m,m]}$ we see that

$$H \;=\; P^T H_0 P \;+\; S\, \tilde{K}_R^{-1}\, S^T \,, \qquad \tilde{K}_R^{-1} \;=\; \pi^T K_R^{-1} \pi + e_m e_m^T / O_{[m,m]} \,, \tag{14}$$

where the structure of operator in inhomogeneous term resembles that of BFGS update and hence its inverse could easily be found

$$\tilde{K}_R \;=\; K_R \;-\; (K_R e_m)(K_R e_m)^T / K_{R\,[m,m]} \;+\; o_m o_m^T / O_{[m,m]} \,.$$

As expected, updated inverse Hessian (14) remains positive if and only if $O_{[m,m]} = s^T y > 0$. Updating rules for Hessian B require to consider $\tilde{K}_L^{-1} \equiv O^{-1} \tilde{K}_R O^{-T}$ (see comments just below (10)), which is seen to be given by

$$\tilde{K}_L^{-1} \equiv O^{-1} \tilde{K}_R O^{-T} = K_L^{-1} \;-\; (K_L^{-1} O^T e_m)(K_L^{-1} O^T e_m)^T / K_{R\,[m,m]} \;+\; e_m e_m^T / O_{[m,m]} \,.$$

It again has the structure of BFGS update, hence its inverse immediately follows

$$\tilde{K}_L \;=\; \tilde{\pi}^T K_L \tilde{\pi} \;+\; \tilde{o}_m \tilde{o}_m^T / O_{[m,m]} \,, \qquad \tilde{\pi} = 1 - e_m \tilde{o}_m^T / O_{[m,m]} \,, \qquad \tilde{o}_m = O^T e_m \,.$$

To summarize, enforced exactness of most recent secant relation does not change the functional form of MS-BFGS updates (9), (10), but requires to use altered kernels

$$\begin{aligned} \tilde{K}_R &= K_R - (K_R e_m)(K_R e_m)^T \,/\, (e_m^T K_R e_m) \;+\; o_m o_m^T \,/\, (e_m^T o_m) \,, \qquad (15) \\ K_R &\;=\; \sqrt{OO^T} \,, \qquad o_m \;=\; O e_m \,, \\ \tilde{K}_L &= \tilde{\pi}^T K_L \tilde{\pi} \;+\; \tilde{o}_m \tilde{o}_m^T \,/\, (e_m^T \tilde{o}_m) \,, \qquad (16) \\ K_L &\;=\; \sqrt{O^T O} \,, \quad \tilde{\pi} = 1 - e_m \tilde{o}_m^T / (e_m^T \tilde{o}_m) \,, \quad \tilde{o}_m = O^T e_m \,, \end{aligned}$$

which are obtained via conventional BFGS updates of original K_R/K_L matrices with specific choice of respective displacements pairs.

3 MS-BFGS Damping

Usual BFGS update includes inhomogeneous term $ss^T/s^T y$, which makes update well-defined and positive definite only if $s^T y$ is (sufficiently) positive. A priori, sufficiency could be quantified in at least two different

ways: we could either require $s^T y \geq \varepsilon_1 |s||y|$ or $s^T y \geq \varepsilon_2 s^T B s$, where $\varepsilon_{1,2}$ are predefined (small) constants and B is current SPD approximation to underlying Hessian. First variant seems to be inappropriate because it is not invariant with respect to affine coordinate transformations (although it is widely used in SR1 schemes due to the lack of positivity in this context). Second one is the base of commonly used Powell's damping scheme [14], in which one tests required inequality and if necessary modifies gradient displacement according to

$$y \;\rightarrow\; (1-\theta)\, y + \theta\, B\, s\,, \tag{17}$$

where θ is such that violated inequality is cured. This damping scheme ensures that BFGS update is well-defined and produces SPD approximation.

In multi-secants context we need similar facility to ensure sufficient non-singularity of the overlap $O_m = S_m^T Y_m$. Positivity of MS-BFGS approximations is of no concern, updates (9), (10) produce positive matrices by construction. Hence it suffices to consider boundness of B, for which analysis of Ref. [38] applies directly. Namely, consider first the determinant of MS-BFGS updated approximation $\tilde{B}$:

$$\det \tilde{B} = \det B \;\; \det[1 - Q_s Q_s^T + U_y U_y^T]\,,$$

$$Q_s = \sqrt{B} S_m \,(S_m^T B S_m)^{-1/2}\,, \qquad U_y = \sqrt{H} Y_m K_L^{-1/2}\,,$$

where B (H) is Hessian (inverse Hessian) before the update. It is crucial that Q_s is orthonormal, $Q_s^T Q_s = 1$, because from the identity

$$\det[1 + UU^T - VV^T] = \det[(1+U^TU)(1-V^TV) + (1+U^TU)(V^TU)(1+U^TU)^{-1}(U^TV)]$$

valid for tall-skinny V, U of the same geometry, we conclude that

$$\det \tilde{B} = \det B \; \frac{\det[O_m K_L^{-1} O_m^T]}{\det[S_m^T B S_m]} = \det B \; \frac{\det[K_R]}{\det[S_m^T B S_m]}\,.$$

Therefore, to properly bound lower eigenvalues of $\tilde{B}$ we require

$$\det[K_R] \geq \varepsilon_s \; \det[S_m^T B S_m]\,, \tag{18}$$

where ε_s is a small positive constant. In case of single secant update, $m = 1$, inequality (18) almost reduces to that of Powell's scheme, $|s^T y| \geq \varepsilon_s\, s^T B s$, albeit with explicit MS-BFGS specific modulus. Hence we expect ε_s to be of commonly accepted order ~ 0.1. Practical aspects of (18) are to be discussed shortly.

Large $\tilde{B}$-eigenvalues are to be monitored by $\mathrm{Tr}[\tilde{B}]$, for which we have

$$\mathrm{Tr}[\tilde{B}] = \mathrm{Tr}[B\,(1 - Q_s Q_s^T)] + \mathrm{Tr}[Y_m^T Y_m\, K_L^{-1}] \leq \mathrm{Tr}[B] \;+\; \mathrm{Tr}[Y_m^T Y_m]\,\mathrm{Tr}[K_L^{-1}]\,.$$

It is natural then to require $\mathrm{Tr}[Y_m^T Y_m]\,\mathrm{Tr}[K_L^{-1}] \leq 1/\varepsilon_y$, however, appearance of manifestly non-invariant term $\mathrm{Tr}[Y^T Y]$ looks awkward. It would be better to write $\mathrm{Tr}[Y_m^T H Y_m]$ instead, but inequality then requires the replacement $\varepsilon_y \rightarrow \varepsilon_y\, \mathrm{Tr}[B]$. Hopefully, it seems that ε_y might be taken sufficiently small so that its rescaling with bounded $\mathrm{Tr}[B]$ could be absorbed into redefinition of ε_y parameter. Therefore, we're going to regulate large $\tilde{B}$ eigenvalues via

$$(\mathrm{Tr}[K_L^{-1}])^{-1} \geq \varepsilon_y\, \mathrm{Tr}[Y_m^T\, H\, Y_m] \tag{19}$$

with parametrically small constant ε_y (say, $\sim 10^{-3}$), more accurate value of which is to be determined experimentally.

Practical implementation of (18) and (19) looks as follows. We noted already that there might be no possibility to exactly maintain predefined number m of secant relations. In iterative context coordinate/gradient displacements s, y appear sequentially, with previous pairs being already accommodated into current approximations. Hence there seems to be no option to alter previously seen displacement vectors and relations (18), (19) provide a mean to dynamically select the number of secant conditions to be served. Namely, proposed number of secants and respective last seen displacements are to be examined via (18), (19) and accepted only if both inequalities are fulfilled. Otherwise, oldest displacements pair is to be skipped and test of inequalities continues down to $m = 1$. It then remains to discuss how to deal with limiting $m = 1$ case.

For single s, y pair relations (18), (19) reduce to inequality

$$|s^T y| \geq \max\{\varepsilon_s \, (s^T B s) \, , \, \varepsilon_y \, (y^T H y) \, \} \, . \tag{20}$$

Conventionally, conformance with it is achieved via alternation of y-vector only (cf. (17)), however, (20) looks totally symmetric with respect to coordinate and gradient displacements. Therefore it seems natural to consider symmetric deformation

$$s \to (1 - \theta_s) \, s \pm \theta_s H y \, , \qquad y \to (1 - \theta_y) \, y \pm \theta_y B s \, , \tag{21}$$

to ensure (20) and to provide minimal distortion of s and y which, from the principle of affine invariance, is to be quantified by $(\Delta s^T B \Delta s) + (\Delta y^T H \Delta y) \propto \theta_s^2 + \theta_y^2$. Certainly, sign is to be in accord with the sign of initial product $s^T y$ and is to be positive if we're going to impose $s^T y > 0$ (no modulus on the left hand side of (20)). Note that advantage of somewhat similar in spirit "double damping" approach was already noted in Ref. [39]. Resulting box-bounded[2] $0 \leq \theta_{s,y} \leq 1/2$ auxiliary optimization problem possesses quadratic objective and two quadratic inequality constraints, however, it still admits efficient solution (see Section 5 for details).

Damping with modified update rules (15), (16) requires slight generalization. Formally, relations (18), (19) remain the same, but now altered kernels $\tilde{K}_R$, $\tilde{K}_L$ are to be substituted. Since these are related to K_R, K_L via usual BFGS updates, similar technique allows to derive

$$\det[\tilde{K}_R] = \det[K_R] \, O_{[m,m]} / K_{R\,[m,m]} \, , \qquad \mathrm{Tr}[\tilde{K}_L^{-1}] \leq \mathrm{Tr}[K_L^{-1}] + 1/O_{[m,m]} \, , \tag{22}$$

and right hand sides of these relations are to be used in (18), (19) in case last secant equation is maintained exactly. However, both (15) and (16) silently assume that $O_{[m,m]} = s^T y$ is sufficiently positive, which has to be assured first via (20) without absolute value operator and (21) taken with positive sign. Therefore, application order of damping scheme changes: it starts with most recent displacements pair and only then inequalities (18), (19) with replacements $K_{R,L} \to \tilde{K}_{R,L}$ together with (22) determine the number of secant relations to be served next.

4 Compact Representation and Limited-Memory Version

With dense matrices update rules (9), (10) fully specify MS-BFGS scheme. It is accomplished by collecting most recent linear independent m coordinate/gradient displacements s_k, y_k ("moving" or "active" window of size m) in S_m and Y_m matrices and by applying updates iteratively. Note that in general m is floating number bounded above by predefined maximal number of served secants. It is of interest to develop

[2]Formal upper bound 1 is halved because $\theta_{s,y} = 1/2$ is feasible for $\varepsilon_{s,y} \leq 1$; we argue below that, in fact, $\varepsilon_{s,y} < 1/2$.

corresponding compact representation (limited memory context), which is the purpose of this section. For convenience we repeat the key MS-BFGS update using slightly different notations

$$H = P_m^T\, H_0\, P_m + S_m\, K_m^{-1}\, S_m^T\,, \tag{23}$$

$$P_m = 1\; -\; Y_m O_m^{-1} S_m^T\,, \qquad O_m \;=\; S_m^T Y_m\,,$$

where kernel K_m is $K_R = (O_m O_m^T)^{1/2}$, but for now it is assumed arbitrary SPD matrix of proper size. For obvious reasons, the first and second term on the right hand side of (23) is referred to as "homogeneous" and "inhomogeneous".

Operationally, multi-secant limited-memory BFGS (MS-LBFGS for short) approximation procedure looks as follows. There are two predefined parameters: M is the maximal number of secant equations one would like to maintain; L is the maximal number of (s, y) pairs ever kept in memory (memory length), which are conveniently collected in matrices S, Y with at most L columns. These parameters naturally obey $M \le L$, moreover, the above S_m and Y_m are appropriate subsets of columns in S, Y. Then MS-LBFGS is the same as MS-BFGS, but utilizes at most L recent S/Y records (older ones are to be discarded) to update initial H_0 and to represent resulting approximation entirely in terms of S, Y and H_0. This is achieved via explicit unrolling of recursions (23) using a few logically distinct constituents, which we sequentially discuss.

4.1 Gathering First Secant Equations

Given initial approximation H_0 one sequentially gets first m pairs (s_k, y_k), $k = 1, ..., m$, which are accumulated in updates (23). Specifically, H_0 is initially updated with very first pair (single secant equation holds afterwards), then S/Y matrices are augmented with (s_2, y_2) and update (23) is applied to previous outcome (so that two secant equations hold) and so on. Until m pairs are collected, S/Y matrices grow in columns and hence describe a sequence of nested linear spaces. Denoting indices of participating vectors with $[1, ..., k]$ we then have $S_{[1,...,k-1]}^T\, P_{[1,...,k]} = 0$, $P_{[1,...,k-1]}\, P_{[1,...,k]} = P_{[1,...,k]}$, and therefore for $1 \le k \le m$ inverse Hessian is approximated as

$$H_{\mathbf{k}} \;=\; P_{\mathbf{k}}^T\, H_0\, P_{\mathbf{k}} \;+\; S_{\mathbf{k}}\, K_{\mathbf{k}}^{-1}\, S_{\mathbf{k}}^T\,,$$

where $\mathbf{k} = [1, ..., k]$ denotes currently used set of (s, y)-pairs. Note that at this initial stage updates are, in fact, independently applied to initial matrix H_0.

4.2 Product of Overlapping Projectors

Subsequent updates include products of overlapping projectors of the type

$$P_{[1,...,m]}\, P_{[2,...,m+1]} \;=\; P_{\mathbf{1},\mathbf{2}}\, P_{\mathbf{2},\mathbf{3}}\,,$$

where bold numbers refer to subsets of indices (here $\mathbf{1} = [1]$, $\mathbf{3} = [m+1]$, $\mathbf{2} = [2, ..., m]$) and correspond to usual block notations. Index set $\mathbf{2}$ is common to both projectors, while $\mathbf{1}$ and $\mathbf{3}$ are distinct. Direct evaluation reveals that

$$P_{\mathbf{1},\mathbf{2}}\, P_{\mathbf{2},\mathbf{3}} \;=\; \Pi_{\mathbf{1},\mathbf{2},\mathbf{3}} \;=\; 1 \;-\; Y_{\mathbf{1},\mathbf{2},\mathbf{3}}\, X_{\mathbf{1},\mathbf{2},\mathbf{3}}^{-1}\, S_{\mathbf{1},\mathbf{2},\mathbf{3}}^T\,, \tag{24}$$

where the matrix $X_{\mathbf{1},\mathbf{2},\mathbf{3}}$ is the same as total overlap $O_{\mathbf{1},\mathbf{2},\mathbf{3}}$, but with altered $(\mathbf{3}, \mathbf{1})$ block. Specifically, omitting common subscript $\mathbf{1}, \mathbf{2}, \mathbf{3}$ and using block notations we have

$$[\, X\,]_{\mathbf{3},\mathbf{1}} \;=\; O_{\mathbf{3},\mathbf{2}}\, O_{\mathbf{2},\mathbf{2}}^{-1}\, O_{\mathbf{2},\mathbf{1}}\,. \tag{25}$$

Qualitatively, modification of $(\mathbf{3},\mathbf{1})$ block is well understood: product (24) contains no direct overlaps between subsets **3** and **1** and $X_{\mathbf{3},\mathbf{1}}$ corresponds to "effective interaction" via common subset **2**.

To proceed we need to consider $\Pi_{\mathbf{1},\mathbf{2}}\,P_{\mathbf{2},\mathbf{3}}$ with non-intersecting sets of indices $\mathbf{1},\mathbf{2},\mathbf{3}$. In MS-LBFGS context **2** is of size $m-1$, **3** contains single index (newcoming displacements) and **1** is arbitrary (past indices). By induction matrix X entering $\Pi_{\mathbf{1},\mathbf{2}}$ has block structure

$$X_{\mathbf{1},\mathbf{2}} \;=\; \left[\begin{array}{cc} O_{\mathbf{1},\mathbf{1}} & O_{\mathbf{1},\mathbf{2}} \\ X_{\mathbf{2},\mathbf{1}} & O_{\mathbf{2},\mathbf{2}} \end{array}\right]. \tag{26}$$

Derivation below utilizes the following conventions:

- Matrices in large square brackets are of block size 3×3, blocks refer to subsets **1**, **2**, **3** (and in this order), empty places stand for identically zero blocks. Each matrix includes submatrix of block-size 2×2, placed at different corners. It is convenient then to represent such matrices as being of formal block size 2×2, submatrix placement unambiguously defines actual matrix content;
- In order to avoid notational ambiguities, total overlap matrices for different index sets are denoted by M, for instance, $M_{\mathbf{2},\mathbf{3}} = S^T_{\mathbf{2},\mathbf{3}} Y_{\mathbf{2},\mathbf{3}}$ is of block size 2×2 with blocks $O_{\mathbf{i},\mathbf{j}} = S^T_{\mathbf{i}} Y_{\mathbf{j}}$, $\mathbf{i},\mathbf{j} = \mathbf{2},\mathbf{3}$.

We have:

$$\Pi_{\mathbf{1},\mathbf{2}}\,P_{\mathbf{2},\mathbf{3}} \;=\; (1 - [Y_{\mathbf{1}}Y_{\mathbf{2}}Y_{\mathbf{3}}] \left[\begin{array}{cc} X^{-1}_{\mathbf{1},\mathbf{2}} & \\ & \end{array}\right] [S_{\mathbf{1}}S_{\mathbf{2}}S_{\mathbf{3}}]^T) \cdot (1 - [Y_{\mathbf{1}}Y_{\mathbf{2}}Y_{\mathbf{3}}] \left[\begin{array}{cc} & \\ & M^{-1}_{\mathbf{2},\mathbf{3}} \end{array}\right] [S_{\mathbf{1}}S_{\mathbf{2}}S_{\mathbf{3}}]^T). \tag{27}$$

Omitting trivial identity in (27) the matrix between $[Y_{\mathbf{1}}Y_{\mathbf{2}}Y_{\mathbf{3}}]$ and $[S_{\mathbf{1}}S_{\mathbf{2}}S_{\mathbf{3}}]^T$ after left and right multiplication with block-diagonal $[X_{\mathbf{1},\mathbf{2}},1]$ and $[1,M_{\mathbf{2},\mathbf{3}}]$ matrices reads

$$\left[\begin{array}{cc} 1_{2\times 2} & \\ & \end{array}\right] \left[\begin{array}{cc} 1_{1\times 1} & \\ & M_{\mathbf{2},\mathbf{3}} \end{array}\right] + \left[\begin{array}{cc} X_{\mathbf{1},\mathbf{2}} & \\ & 1_{1\times 1} \end{array}\right] \left[\begin{array}{cc} & \\ & 1_{2\times 2} \end{array}\right] - \left[\begin{array}{cc} 1_{2\times 2} & \\ & \end{array}\right] M_{\mathbf{1},\mathbf{2},\mathbf{3}} \left[\begin{array}{cc} & \\ & 1_{2\times 2} \end{array}\right], \tag{28}$$

where block geometry of unit operators were made explicit. Working out each term we have in 3×3 block notations

$$(28) = \left[\begin{array}{ccc} 1 & & \\ & O_{\mathbf{2},\mathbf{2}} & O_{\mathbf{2},\mathbf{3}} \\ & & \end{array}\right] + \left[\begin{array}{ccc} & O_{\mathbf{1},\mathbf{2}} & \\ & O_{\mathbf{2},\mathbf{2}} & \\ & & 1 \end{array}\right] - \left[\begin{array}{ccc} & O_{\mathbf{1},\mathbf{2}} & O_{\mathbf{1},\mathbf{3}} \\ & O_{\mathbf{2},\mathbf{2}} & O_{\mathbf{2},\mathbf{3}} \\ & & \end{array}\right] = \left[\begin{array}{ccc} 1 & & O_{\mathbf{1},\mathbf{3}} \\ & O^{-1}_{\mathbf{2},\mathbf{2}} & \\ & & 1 \end{array}\right]^{-1}.$$

Restoring previously used left/right multipliers we conclude that

$$\Pi_{\mathbf{1},\mathbf{2}}\,P_{\mathbf{2},\mathbf{3}} \;=\; 1 \;-\; Y_{\mathbf{1},\mathbf{2},\mathbf{3}}\,X^{-1}_{\mathbf{1},\mathbf{2},\mathbf{3}}\,S^T_{\mathbf{1},\mathbf{2},\mathbf{3}}\,, \tag{29}$$

where (momentarily mixing up implicit and explicit 3×3 block notations)

$$X_{\mathbf{1},\mathbf{2},\mathbf{3}} \;=\; \left[\begin{array}{ccc} 1 & & \\ & M_{\mathbf{2},\mathbf{3}} & \end{array}\right] \left[\begin{array}{ccc} 1 & & O_{\mathbf{1},\mathbf{3}} \\ & O^{-1}_{\mathbf{2},\mathbf{2}} & \\ & & 1 \end{array}\right] \left[\begin{array}{cc} X_{\mathbf{1},\mathbf{2}} & \\ & 1 \end{array}\right]$$

and $X_{\mathbf{1},\mathbf{2}}$ is from (26). Straightforward evaluation reveals that

$$X_{\mathbf{1},\mathbf{2},\mathbf{3}} \;=\; \left[\begin{array}{ccc} O_{\mathbf{1},\mathbf{1}} & O_{\mathbf{1},\mathbf{2}} & O_{\mathbf{1},\mathbf{3}} \\ X_{\mathbf{2},\mathbf{1}} & O_{\mathbf{2},\mathbf{2}} & O_{\mathbf{2},\mathbf{3}} \\ O_{\mathbf{3},\mathbf{2}}O^{-1}_{\mathbf{2},\mathbf{2}}X_{\mathbf{2},\mathbf{1}} & O_{\mathbf{3},\mathbf{2}} & O_{\mathbf{3},\mathbf{3}} \end{array}\right], \tag{30}$$

which is essentially the same as (25) and admits straightforward recurrent definition of the product of projectors when current "active window" is advanced. Focusing on MS-LBFGS updates with single new (s, y) pair it means that the product $\Pi_{(...)}P_{(...)}$ remains of Π-type, Eq. (24), but with updated X-matrix. Specifically, X is to be augmented with new row and column reflecting overlaps of newcoming s, y displacements with previously seen. However, algebraic structure of considered products forbids appearance of direct overlap between most recent coordinate displacement s and gradient differences beyond current "active window". Respective $(\mathbf{3}, \mathbf{1})$-block of X-matrix represents "effective overlap" mediated by already seen and still active displacements. Limited memory context restricts maximal allowed X-matrix size: once its order becomes $L \times L$ older records are to be discarded (as well as the corresponding columns of S and Y), although simple-minded deletion of one row and column at a time might not be, in fact, appropriate. Detailed discussion of proper memory truncation in MS-LBFGS scheme is postponed till Section 4.4.

In the particular case of constantly single secant condition ($m = 1$ always) equation (30) greatly simplifies. Block **3** becomes one-dimensional, while common block **2** disappears. In this case X-matrix reduces to upper-triangular part of total overlap O in full agreement with well-known results [11].

4.3 Inhomogeneous Terms

Sequential application of MS-BFGS basic update (23) requires also to consider inhomogeneous terms, which do not refer explicitly to initial approximation matrix H_0. In the notations of previous section expression to be examined is

$$P^T_{\mathbf{2,3}} \left(S_{\mathbf{1,2}} \, K^{-1}_{\mathbf{1,2}} \, S^T_{\mathbf{1,2}} \right) P_{\mathbf{2,3}} \;+\; S_{\mathbf{2,3}} \, K^{-1}_{\mathbf{2,3}} \, S^T_{\mathbf{2,3}} \,, \tag{31}$$

where the kernels $K_{\mathbf{1,2}}$, $K_{\mathbf{2,3}}$ are temporally taken to be arbitrary invertible matrices of appropriate size. Since

$$S^T_{\mathbf{1,2}} \, P_{\mathbf{2,3}} \;=\; \left[\begin{array}{c} S^T_{\mathbf{1}} \;-\; [\, O_{\mathbf{12}} \,,\, O_{\mathbf{13}}] \, M^{-1}_{\mathbf{2,3}} \, [S_{\mathbf{2}} \,,\, S_{\mathbf{3}}]^T \\ 0 \end{array} \right] , \tag{32}$$

in terms of new basis

$$\tilde{S}_{\mathbf{1}} = S_{\mathbf{1}} \,, \qquad [\tilde{S}_{\mathbf{2}}, \tilde{S}_{\mathbf{3}}]^T = M^{-1}_{\mathbf{2,3}} \, [S_{\mathbf{2}}, S_{\mathbf{3}}]^T \tag{33}$$

expression (31) may be written as

$$(31) \;=\; \tilde{S}_{\mathbf{1,2,3}} \left(\left[\begin{array}{c} 1 \\ -O^T_{\mathbf{1,2}} \\ -O^T_{\mathbf{1,3}} \end{array} \right] K^{-1}_{\mathbf{1,2}} \left[\begin{array}{ccc} 1 & -O_{\mathbf{1,2}} & -O_{\mathbf{1,3}} \end{array} \right] \;+\; \left[\begin{array}{cc} & \\ & \tilde{K}^{-1}_{\mathbf{2,3}} \end{array} \right] \right) \tilde{S}^T_{\mathbf{1,2,3}} \,, \tag{34}$$

where $\tilde{K}^{-1}_{\mathbf{2,3}} = M^T_{\mathbf{2,3}} K^{-1}_{\mathbf{2,3}} M_{\mathbf{2,3}}$ and block matrix notations are similar to previously used. Introducing non-singular upper triangular matrix

$$\hat{R} \;=\; \left[\begin{array}{ccc} 1 & O_{\mathbf{1,2}} & O_{\mathbf{1,3}} \\ & 1 & \\ & & 1 \end{array} \right] , \tag{35}$$

we see that expression in round brackets of (34) equals to

$$\hat{R}^{-T} \left[\begin{array}{cc} d_{\mathbf{1,1}} & \\ & \tilde{K}^{-1}_{\mathbf{2,3}} \end{array} \right] \hat{R}^{-1} \,,$$

where $d_{\mathbf{1,1}} = [\, K^{-1}_{\mathbf{1,2}} \,]_{\mathbf{1,1}}$ denotes top-left block of $K^{-1}_{\mathbf{1,2}}$. Going back to original basis we then have for (31)

$$(31) \;=\; S_{\mathbf{1,2,3}} \left[\begin{array}{cc} 1 & \\ & M^{-T}_{\mathbf{2,3}} \end{array} \right] \hat{R}^{-T} \left[\begin{array}{cc} d_{\mathbf{1,1}} & \\ & \tilde{K}^{-1}_{\mathbf{2,3}} \end{array} \right] \hat{R}^{-1} \left[\begin{array}{cc} 1 & \\ & M^{-1}_{\mathbf{2,3}} \end{array} \right] S^T_{\mathbf{1,2,3}} \,,$$

which could be further simplified via

$$\hat{R}^{-1} \begin{bmatrix} 1 & \\ & M_{\mathbf{2,3}}^{-1} \end{bmatrix} = \begin{bmatrix} 1 & \\ & M_{\mathbf{2,3}}^{-1} \end{bmatrix} \begin{bmatrix} 1 & x_{\mathbf{2,3}} \\ & 1 \end{bmatrix}^{-1} ,$$

$$x_{\mathbf{2,3}} = [\, O_{\mathbf{1,2}} \,,\, O_{\mathbf{1,3}} \,] \cdot M_{\mathbf{2,3}}^{-1} . \tag{36}$$

To summarize, considered inhomogeneous term takes the form

$$(31) = S_{\mathbf{1,2,3}} \, R_{\mathbf{1,2,3}}^{-T} \, D_{\mathbf{1,2,3}} \, R_{\mathbf{1,2,3}}^{-1} \, S_{\mathbf{1,2,3}}^{T} , \tag{37}$$

where block-diagonal $D_{\mathbf{1,2,3}}$ and upper-triangular $R_{\mathbf{1,2,3}}$ matrices are

$$D_{\mathbf{1,2,3}} = \begin{bmatrix} d_{\mathbf{1,1}} & \\ & K_{\mathbf{2,3}}^{-1} \end{bmatrix} , \qquad R_{\mathbf{1,2,3}} = \begin{bmatrix} 1 & x_{\mathbf{2,3}} \\ & 1_{2\times 2} \end{bmatrix} , \tag{38}$$

with row block-vector $x_{\mathbf{2,3}}$ from (36).

To finish derivation and to conclude with recurrent formulae we consider generalization of (31)

$$P_{\mathbf{2,3}}^{T} \, (S_{\mathbf{1,2}} \, R_{\mathbf{1,2}}^{-T} \, D_{\mathbf{1,2}} \, R_{\mathbf{1,2}}^{-1} \, S_{\mathbf{1,2}}^{T}) \, P_{\mathbf{2,3}} \; + \; S_{\mathbf{2,3}} \, K_{\mathbf{2,3}}^{-1} \, S_{\mathbf{2,3}}^{T} , \tag{39}$$

where

$$R_{\mathbf{1,2}} = \begin{bmatrix} r_{\mathbf{1,1}} & * \\ & * \end{bmatrix} , \qquad D_{\mathbf{1,2}} = \begin{bmatrix} d_{\mathbf{1,1}} & * \\ * & * \end{bmatrix} ,$$

with unit upper triangular $r_{\mathbf{1,1}}$ (stars denote arbitrary entries). Eq. (32) as well as new basis definition (33) remain the same, while $\hat{R}$-matrix (35) is modified only slightly: its top-left block becomes $r_{\mathbf{1,1}}$, which allows then to perform identical to previous transformations. The overall conclusion is that (39) equals to the same expression (37),

$$(39) = S_{\mathbf{1,2,3}} \, R_{\mathbf{1,2,3}}^{-T} \, D_{\mathbf{1,2,3}} \, R_{\mathbf{1,2,3}}^{-1} \, S_{\mathbf{1,2,3}}^{T} , \tag{40}$$

with the same block-diagonal D, slightly changed R and the same row block-vector $x_{\mathbf{2,3}}$, eq. (36)

$$D_{\mathbf{1,2,3}} = \begin{bmatrix} d_{\mathbf{1,1}} & \\ & K_{\mathbf{2,3}}^{-1} \end{bmatrix} , \qquad R_{\mathbf{1,2,3}} = \begin{bmatrix} r_{\mathbf{1,1}} & x_{\mathbf{2,3}} \\ & 1_{2\times 2} \end{bmatrix} . \tag{41}$$

Prior to discussing particularities of representation (40) and of corresponding recurrent formulae (41), it is worth to specialize them to the most relevant case of positive definite kernels, for which equations above could be further simplified. Indeed, let us consider "reversed Cholesky" decomposition $K = r\, r^T$ (with upper triangular r) of all involved kernels. Then D-matrix factorizes into analogous product of block-diagonal triangular factors, which in turn could be absorbed into R-matrices. Therefore, for positive kernels compact representation of inhomogeneous terms involves only R-matrices of the form

$$R_{\mathbf{1,2,3}} = \begin{bmatrix} r_{\mathbf{1,1}} & x_{\mathbf{2,3}} \, r_{\mathbf{2,3}} \\ & r_{\mathbf{2,3}} \end{bmatrix} , \tag{42}$$

where $K_{\mathbf{2,3}} = r_{\mathbf{2,3}} \, r_{\mathbf{2,3}}^{T}$ and $x_{\mathbf{2,3}}$ is given by (36).

Further specializing to unit cardinality of subset $\mathbf{3}$, ultimate summary for this section is as follows: for SPD kernels inhomogeneous terms in MS-LBFGS updating scheme could be written as $S \, R^{-T} \, R^{-1} \, S^T$, where S denotes currently accumulated coordinate displacements. Recurrences for R-matrix start with r-factor of initial kernel, $K = rr^T$; at every update its size grows by one and its m right-most columns are

re-evaluated according to (42): bottom right $m \times m$ block gets r-factor of newcoming kernel decomposition, while everything above this block is replaced with the product of (36) and the same r-factor.

It is worth to examine how the above representation simplifies in case only single secant equation is maintained ($m = 1$ always) and, in particular, how it reduces to well-known LBFGS formulation. First, all involved kernels become positive numbers, $K = |s^T y| \equiv |\gamma|$, corresponding r-factors being $\sqrt{|\gamma|}$. Second, overlapping index set **2** disappears, while subset **3** turns into single index (say, k) of most recent iterate. In this case, matrix $x_{\mathbf{2},\mathbf{3}}$ degenerates to single column vector, $x_{\mathbf{2},\mathbf{3}} \to S_{\mathbf{1}}^T y_k \,/\, \gamma_k$ and in (42) we have

$$R_{\mathbf{1},k} = \begin{bmatrix} R_{\mathbf{1}} & S_{\mathbf{1}}^T y_k \\ & \gamma_k \end{bmatrix} \begin{bmatrix} 1 & \\ & \sqrt{|\gamma_k|}/\gamma_k \end{bmatrix} . \tag{43}$$

It is seen that representation $SR^{-T}R^{-1}S^T$ in $m = 1$ case alludes to restore diag$[\,|\gamma_k|\,]$ matrix between R-factors and to modify R-matrices themselves: as is hinted by (43) and is easily confirmed by inductive arguments R-factor becomes upper triangular part of total overlap $O = S^T Y$. Moreover, positivity requirement $\gamma_i = (s_i^T y_i) > 0$, $\forall i$ equates appeared diagonal matrix to the diagonal part of the same overlap and everything reduces to well-known LBFGS formulation.

Finally, it remains to mention minor adjustments required if one wants to impose last seen secant equation exactly, Section 2.2. Notable simplification arises due to the close connection between $r\,r^T$ factorization of kernels K_R and $\tilde{K}_R$. Indeed, having computed $K_R = r\,r^T$ and using $r^T e_m \propto e_m$ we get in (15)

$$\tilde{K}_R = r\,(1 - e_m e_m^T + v_m v_m^T)\,r^T\,, \qquad v_m = r^{-1} o_m \,/\, \sqrt{O_{[m,m]}}\,.$$

In turn, $r\,r^T$ decomposition of the expression in round brackets is trivial, respective r-factor is unit matrix with last column replaced with v_m. Then r-factor for $\tilde{K}_R$ turns out to be the same r matrix, in which last column is substituted with $rv_m = o_m \,/\, \sqrt{O_{[m,m]}}$.

4.4 MS-LBFGS: Further Details and Summary

Up to this point we specifically avoided to discuss memory truncation within compact representation of MS-BFGS. In this respect all we did up to now is relatively simple rewrite of the same basic MS-BFGS update (23). Order of relevant matrices S, Y, X and R grows by one at each iteration and it is tempting to simply discard oldest record once respective size reaches maximal allowed. However, with dynamically selected number of served secants this might not be appropriate. Qualitative indication of this comes from the fact that kernel K_R is complicated non-linear function of involved overlaps $s_i^T y_j$ and includes pair-wise correlations between all displacement pairs in currently "active" set of secants. In this sense kernels are non-local with memory extending up to m steps backward in iterations. With naive memory truncation inhomogeneous term $SR^{-T}R^{-1}S^T$ might include pieces of kernels used previously and hence might implicitly refer to displacement pairs already gone from available memory.

To illustrate, let's consider a sequence of two updates (23) with respective numbers of imposed secant conditions m_1 and m_2 with $m_2 \leq m_1$ (trivial case $m_2 = m_1 + 1$ is excluded). Problematic top-left block of resulting $(m_1 + 1) \times (m_1 + 1)$ R-matrix (42) is of order $m_1 - m_2 + 1$ and unless memory reduction totally truncates it, dependence on formally removed displacements survives via left out matrix elements of previously used kernel K_1. Note that the issue does not arise for $m_1 = m_2$, otherwise, we need to ensure proper location of memory cut (number of oldest records to be removed). It is easily seen that memory truncation is allowed only at the start of some previously imposed block of secant conditions. Helpful illustration of this is provided by diagrammatic representation of multi-secant updates, Fig. 1, where horizontal solid lines represent particular update (line length is in accord with the number of imposed secant conditions), updates are ordered from top to bottom and black points denote s, y pairs with convention that

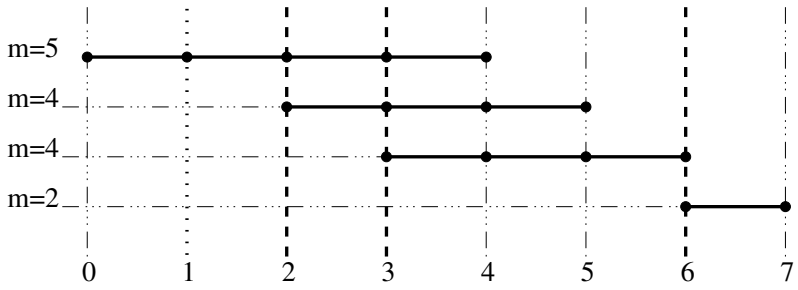

Figure 1: Diagrammatic representation of MS-LBFGS updating scheme. Solid horizontal lines depict sequence of performed updates ordered from top to bottom, line length reflects the number of imposed secant conditions. Iteration index is on horizontal axis, black dots denote s, y pairs and are identified along each vertical line. Memory truncation in limited-memory context is allowed only at the positions of vertical dashed lines.

points on the same vertical line are identified (these correspond to the same displacements participating differently at sequential updates). Right end-points of solid lines shift to south-east at constant slant reflecting the fact we gain s, y displacements one at a time, while their left ends move with different slopes because of varying number of served secants. Allowed locations of memory cuts are displayed with dashed vertical lines.

Specific requirements on allowed memory truncations could also be illustrated from viewpoint of homogeneous terms (29), (30). Consider the example of Fig. 1, which represents four-fold application of (23). Damping scheme of Section 3 ensured that all involved overlap matrices are non-singular and overall update is well-defined. Suppose now that memory cut is applied at the position of dotted line (index 1). Then very first update incorporates overlap $O_{[1,...,4]}$, for which non-singularity cannot be assured, its minimal singular value drops below that of $O_{[0,...,4]}$ and could become arbitrary small (see, for instance, [33]). To the contrary, memory truncation at allowed left-most dashed line (index 2) results in the product $P_{[2,...,4]}P_{[2,...,5]}$, which is identically equal to $P_{[2,...,5]}$ and no near singularity appears.

Finally, it remains to discuss appropriate selection of initial approximation matrix H_0. As is usual in limited-memory approaches H_0 could be promoted to depend on current iteration (see, e.g., [40], [11], [7]) or even on recently accumulated history and its construction might be quite sophisticated [15],[41]. Here we focus on conventional scalar initialization, $H_0 = \gamma_*$, defined by the requirement to most closely reproduce intended secant relations (11)

$$\gamma_* = \arg\min_\gamma \, ||\gamma\, Y_m - S_m\, \Omega_m||_F^2 \,.$$

Perhaps, it is remarkable that solution to this problem $\gamma_* = ||r_m^{-1} O_m||_F^2 / ||Y_m||_F^2$ is always positive and in single secant limit reduces to $|s^T y|/|y|^2$, generalizing conventional expression $s^T y/|y|^2$, well-known in BFGS context. Here, r_m is r-factor of kernel K_R rr^T-decomposition, $\sqrt{O_m O_m^T} = r_m r_m^T$. Certainly, there is an alternative way to estimate scalar factor via $\min_\gamma \, ||\gamma^{-1}\, S_m - Y_m\, \Omega_m^T||_F^2$, which queries minimal mismatch of the same intended secant relations, but represented differently. Respective scale $\gamma^* = ||S_m||_F^2 / ||r_m||_F^2$, being analogous to familiar but impractical guess $|s|^2/s^T y$ in common single-secant methods, obeys $\gamma_* \leq \gamma^*$ and is not helpful for similar reason: it significantly underestimates relevant curvatures and causes severe performance degradation.

To summarize, multi-secant limited-memory BFGS (MS-LBFGS) updating scheme is characterized by two integers $M \leq L$, where M is the maximal number of secant conditions to be maintained, L is the maximal number of (s, y)-pairs to be kept in memory. Inverse Hessian approximation looks similar to

usual LBFGS formulae and is given by

$$H \;=\; \Pi^T\, H_0\, \Pi \;+\; S\, R^{-T}\, R^{-1}\, S^T\,, \qquad \Pi \;=\; 1 \;-\; Y\, X^{-1}\, S^T\,, \tag{44}$$

where:

- Tall-skinny matrices S and Y of at most L columns represent, perhaps, corrected (see Section 3) most recent coordinate and gradient displacements;
- Square and upper triangular matrices X and R of maximal size $L \times L$ are updated iteratively, see Sections 4.2, 4.3.

Similar to classic LBFGS there is an equivalent representation

$$H \;=\; H_0 \;+\; [\, S\,;\, H_0 Y\,] \begin{bmatrix} R^{-T}R^{-1} \;+\; X^{-T}(Y^T H_0 Y)X^{-1} & -X^{-T} \\ -X^{-1} & 0 \end{bmatrix} [\, S\,;\, H_0 Y\,]^T\,, \tag{45}$$

from which explicit form of corresponding Hessian approximation might also be derived

$$B \;=\; B_0 \;+\; [\, B_0 S\,;\, Y\,] \begin{bmatrix} -\,S^T B_0 S & Q \\ Q^T & C \end{bmatrix}^{-1} [\, B_0 S\,;\, Y\,]^T\,, \tag{46}$$

$$Q \;=\; X - S^T Y\,, \qquad C = (\, R^{-1} X\,)^T\, R^{-1} X\,.$$

Treatment of matrix inversion in (46) is no way different from that of LBFGS and hence is not discussed.

5 Numerical Experiments

In order to assess practical relevance of MS-LBFGS scheme we performed extensive numerical experiments, thorough discussion of which is the purpose of this section. To evaluate operational significance of any quasi-Newton approximation method, it has to be embedded into particular unconstrained optimization algorithm of quasi-Newton type. In turn, enclosing optimizer is to be sufficiently advanced to allow potential benefits of underlying quasi-Newton scheme to be revealed and yet not to be overly sophisticated to admit treatment of unveiled (dis)advantages as being due to approximation methodology itself and not due to optimization algorithm complexity. Therefore, we implemented rather standard Armijo line-search method (see, e.g., Refs. [7], [8], [25]), supplemented with a few notable tweaks. First, with yet trivial Hessian approximation (it occurs either at initial iteration or right after enforced reset, which happens once on failed line search) we utilize Goldstein rules instead of plain Armijo to get better step scaling along steepest descent direction. In turn, implemented line search could ever fail for two reasons: either computed direction happens to be not a descent or backtracked step size becomes numerically unacceptable (base and trial points are numerically the same). Fortunately, neither of these causes are actually relevant: the former had never been observed in all performed experiments, the latter turned out to be vanishingly rare. Thus, Goldstein rules are utilized, in fact, at very first iteration only, all subsequent line searches are based on Armijo acceptance criterion supplemented with guarded quadratic guessing and unit initial step size. Second, we implemented simple backtracking safeguard in case underlying problem responses with infinite or numerically undefined value(s). Finally, successful convergence of overall algorithm is governed by smallness of attained gradient only, no matter how small objective decrease or coordinate displacement are. Specifically, let g_0 (g) be objective gradient at initial (current) location, then optimization algorithm terminates with success status if

$$|g|_\infty \;\leq\; \min(\, \max[\, \varepsilon_g\, \max\{1, |g_0|_\infty\},\, \varepsilon_g^{min}\,],\, \varepsilon_g^{max}\,)\,, \tag{47}$$

where default values of indicated parameters are $\varepsilon_g = 10^{-8}$, $\varepsilon_g^{min} = 10^{-4}$, $\varepsilon_g^{max} = 1$. Only in a few cases these parameters were hardened to force convergence of various algorithm instances to roughly the same solution. Remaining stopping criteria with non-fatal statuses include maximal allowed number of iterations and line search failures mentioned earlier. As will be explained shortly, utilized performance measures are sufficiently robust and do not necessitate gradient convergence, hence we could treat these cases as successful. Algorithm structure implies that the term "iterations" means, in fact, the number of gradient evaluations, which by default is upper bounded by $I_{max} = 10^4$. Remaining parameters ε_s, ε_y are inherent to MS-LBFGS damping scheme and are explained in (18)-(20). Although their default values $\varepsilon_s = 10^{-2}$, $\varepsilon_y = 10^{-3}$ are not firmly established (in particular, ε_s seems to significantly differ from similar parameter in Powell's damping scheme) attempted tuning revealed no prominent sensitivity and we stick with these defaults in what follows.

Particular details are to be provided on the solution of MS-LBFGS damping subproblem (20), (21). Prime simplifying assertion here is that once the condition (20) requires non-trivial damping (21), optimal $\theta_{s,y}$ parameters are to be strictly positive. In this case there remain at most three distinct activity patterns of two constraints (20) and optimal point is to be selected from the solutions of respective optimality conditions. Although in each case subproblem remains quadratic in both objective and constraints, due to its analytic nature and tiny dimensionality (damped) Newton method performs extremely well. To justify made assertion we note that in terms of new variables $a_0 = \sqrt{\varepsilon_s B}\, s$, $b_0 = \pm\sqrt{\varepsilon_y H}\, y$ and $x = [x_a, x_b] = [\theta_s, \theta_y]$ MS-LBFGS damping subproblem becomes

$$\min_{x\in R^2_+} |x|^2 \qquad \text{s.t.} \quad a^T b \geq \varepsilon \, \max[\, |a|^2 \,,\, |b|^2 \,]$$

$$a = a_0 + x_a \Delta \,, \qquad b = b_0 - x_b \Delta \,, \qquad \Delta = b_0 - a_0 \,,$$

where $\varepsilon = \sqrt{\varepsilon_s \varepsilon_y}$ and sign selection is in accord with that in (21). In generic case activity of both constraints determines solution unambiguously and there are no reasons to expect either coordinate to vanish. Otherwise, infinitesimal coordinate displacements $dx_{a,b}$ tangential to the equality $a^T b = \varepsilon |a|^2$, for instance, obey

$$(1-\varepsilon)\, dx_b + (|b|^2/|a|^2 + \varepsilon(1-2\varepsilon))\, dx_a \;=\; 0$$

with strictly positive coefficients for $\varepsilon < 1/2$. Then optimality conditions imply $x_a \propto x_b$ with positive proportionality factor, hence none of them vanish on the solution (origin is known to be infeasible). Note that in the original formulation (20) these arguments entail an upper bound $\varepsilon_s, \varepsilon_y < 1/2$.

Numerical assessment of MS-LBFGS updating scheme necessitates thorough discussion of relevant performance measures and selection of particular baseline to compare with. As for the latter, the natural choice is well-known (and de facto standard in the field) LBFGS algorithm [42]-[45] both for its methodological similarity and ease of adaptation to the convergence criteria described above. In fact, the only tweak we introduced into original L-BFGS-B (version 3.0) fortran code is the slight alternation of termination test with respect to relative change of objective function: instead of non-strict inequality we inserted strict one, which allows to effectively cancel this test in accord with our intention.

In turn, performance measures of prime importance are related to the required for convergence number of function and gradient evaluations. However, the issue to be resolved is that MS-LBFGS utilizes Armijo line search and doesn't need gradients at intermediate points, while L-BFGS-B is based on (strong) Wolfe conditions, which demand derivatives at every trial location. In particular, for L-BFGS-B the counts of function and gradient calls are always the same. Due to this we decided to concentrate exclusively on the number N of gradient evaluations, rationale being that in normal circumstances derivatives are most computationally demanding. For MS-LBFGS it counts the performed iterations and fairly reflects expended computational resources. For L-BFGS-B these metrics are, perhaps, biased, but yet qualitatively

correct. In either case L-BFGS-B is included into analysis as an appropriate baseline to get reasonable estimates of the numbers involved with no intention to make comprehensive comparison.

We consider three different families of performance measures quantified via conventional performance profiles [46] and slight variation of data profiles [47] (see, e.g., Refs. [48], [49] for overview and further references). First performance measure to be considered is built over the number of gradient evaluations $N_{s,p}$ required for algorithm (solver) s to converge on problem p with convergence criteria detailed above. It directly fits the framework of performance profiles and is defined as the probability of solver s to be within factor $\tau \geq 1$ worse than the best one with respect to selected performance measure

$$\mathcal{P}_g(\tau) \;=\; \mathrm{Prob}_{p\in\tilde{P}}[\, N_{s,p} \;\leq\; \tau\, N_p^{min}\,]\,, \qquad \tau \geq 1\,, \tag{48}$$

where $N_p^{min} = \min_s N_{s,p}$ and $\tilde{P}$ is some representative subset of optimization problems. Note that comparison of evaluation counts makes sense only when considered algorithms converge to (at least roughly) the same solution. Therefore, we should require that for all problems in $\tilde{P}$ objective values $f_{s,p}$ attained by compared solvers must not significantly differ,

$$f_p^{max} - f_p^{min} \leq \epsilon \max[\,|f_p^{max}|\,,\,1\,]\,, \qquad f_p^{min,max} = (\min,\max)_s\, f_{s,p}\,, \tag{49}$$

where unity on right hand size is conventional guard against too small objective magnitudes and we take $\epsilon = 10^{-2}$ by default. In fact, Eq. (49) defines the selection of subset $\tilde{P}$ from the total set P of test problems and, perhaps, this is a prime weakness of the performance measure (48). Indeed, requirement (49) might significantly restrict the number of suitable problems and limit selection to only relatively simple instances.

To avoid this we introduce another family of performance indicators, which could be qualitatively motivated as follows. For any given solver-problem pair (s,p) we define solver's trajectory $\hat{f}_{s,p}[k]$ to be the graph of minimal attained objective value as a function of evaluation count. This is a non-increasing function on $0 \leq k \leq N_{s,p}$, which starts at initial value f_p^0 and ends up with $f_{s,p}$. Then for any given level $l \leq f_p^0$ we ask at which minimal index $k_{s,p}^l$ trajectory of solver crosses l,

$$k_{s,p}^l \;=\; \arg\min_k(\, \hat{f}_{s,p}[k] \leq l\,)\,, \tag{50}$$

with the convention that $k_{s,p}^l$ is some (very) large positive number for $l < f_{s,p}$. Qualitative picture behind is the "intermediate" rate of objective decrease achieved by solver s on problem p and it applies regardless of finally attained $f_{s,p}$. For reasonable choices of level l best solver is expected to deliver minimal $k_{s,p}^l$. Moreover, (50) considered as a performance measure seems to be fair enough even when various solvers converge to different solutions (although it might be tenable to take $l \geq f_p^{max}$ in this case): intermediate rate of decrease is to large extent independent upon final convergence point and selection rules like (49) become unnecessary. However, definition (50) is not yet suitable to introduce respective performance profile: involved level value depends strongly on problem instance, is not invariant with respect to additive/multiplicative objective function redefinition and hence averaging as in (48) makes no sense. The remedy is to define l in terms of problem-specific data, concretely we consider parameterized family

$$l(\mu) \;=\; f_p^{min} \;+\; (0.1)^\mu\,(f_p^0 - f_p^{min})\,, \qquad \mu \geq 0\,, \tag{51}$$

with understanding that relatively large μ-values are most interesting/important. Corresponding performance profile

$$\mathcal{P}_l(\tau|\mu) \;=\; \mathrm{Prob}_{p\in P}[\, k_{s,p}^{l(\mu)} \;\leq\; \tau\, k_{p,min}^{l(\mu)}\,]\,, \qquad k_{p,min}^{l(\mu)} = \min_s k_{p,s}^{l(\mu)}\,, \qquad \tau \geq 1 \tag{52}$$

is the probability of solver s to attain μ-parameterized objective value within the budget τ times larger than that of the best solver. As usual, magnitude of the profile (52) at $\tau = 1$ is the probability of solver s to be the first in falling below given objective level $l(\mu)$.

Certainly, the above construction admits symmetric counterpart: instead of horizontal cross-sections (levels) of trajectories one could query objective values at fixed iteration count. Defining $\hat{f}_{s,p}[k] = f_{s,p}$ for $k > N_{s,p}$ consider (following [50], [51]) relative gain attained by solver s at iteration k

$$\rho_{s,p}[k] = (\hat{f}_{s,p}[k] - \hat{f}^{min}_{k,p})/(\hat{f}^{max}_{k,p} - \hat{f}^{min}_{k,p})\,, \qquad \hat{f}^{min,max}_{k,p} = (\min,\max)_s\, \hat{f}_{s,p}[k]\,,$$

defined to be zero in case $\hat{f}^{max}_{k,p} = \hat{f}^{min}_{k,p}$. For given solver it then make sense to consider the probability of this performance to be smaller than given threshold, however, we first need to suitably parameterize relevant iteration counts. Borrowing the same parameter μ from (51) we introduce

$$k(\mu) = (1 - q^{\mu}) \max_s N_{s,p}\,, \qquad \mu \geq 0\,, \tag{53}$$

where factor $q < 1$ is fixed by the requirement that for minimal reasonable μ-values (say, $\mu \sim 1$) $k(\mu)$ defines minimal sensible portion of N^{min}_p (for instance, $k(1) \sim N^{min}_p/2$). As before, relatively large μ-values are most interesting/important. Therefore, third parameterized family of performance profiles to be considered is

$$\mathcal{P}_s(\tau|\mu) \;=\; \mathrm{Prob}_{p\in P}[\,\rho_{s,p}[k(\mu)] \leq \tau\,]\,, \qquad 0 \leq \tau \leq 1\,. \tag{54}$$

Aimed with all the above, we now turn to the numerical examples. Yet tiny final remark is in order: below we confine ourselves to single memory length parameter $L = 8$, the same is used in L-BFGS-B. As for the maximal number M of served secant equations, which actually defines particular MS-LBFGS method, we stick with $M \in \{\,0,\,1,\,2,\,3,\,4,\,6,\,8\,\}$ and abbreviate respective algorithm flavor as, e.g., L8M3. By convention $M = 0$ variant is the same as $M = 1$, but always ensures curvature positivity (no modulus involved in (20)) and in this respect is closest to conventional LBFGS albeit utilizes different damping scheme. Still a particular distinction is to be made for algorithms, which impose most recent secant relation exactly and which are named the same way, but with star sign appended, e.g., L8M2*. Namely, these methods for $M = 0, 1$ identically coincide with $M = 0$ variant of uniform secants treatment and hence will not be considered.

5.1 Random Quadratics

To get the feeling of numbers involved, let us first consider a distilled example of random quadratic functions. Specifically, in $3 \cdot 10^3$ dimensions we generated about 10^3 diagonal Hessians with condition number $\kappa = 10^6$ and uniformly distributed in $[1;\kappa]$ spectrum. Solution is at zero (no linear term in objective) and initial point is vector of all ones. Each problem was solved with selected algorithms, relevant objective N_f and gradient N_g evaluation counts were averaged over all instances. In all combinations convergence was achieved with small gradient near the true solution so that the comparison of expended evaluations is fair. Moreover, simplicity of this example allows to prescind from virtually all MS-LBFGS complications. Indeed, damping had never been invoked, number of served secant equations was almost always equal to maximal allowed and these equations are fulfilled exactly (overlap matrix is SPD). Thus, the effect of multiple secants is expected to be seen in its purity.

Fig. 2 represents the averaged evaluation counts for MS-LBFGS algorithms without exactly imposed last secant condition together with respective result for L-BFGS-B baseline, for which the numbers of objective and gradient calls coincide. Up to $M \lesssim L/2 = 4$ multi-secant performances remain fairly the same and equal to that of L-BFGS-B. The only thing which seem to distinguish single and multi-secant

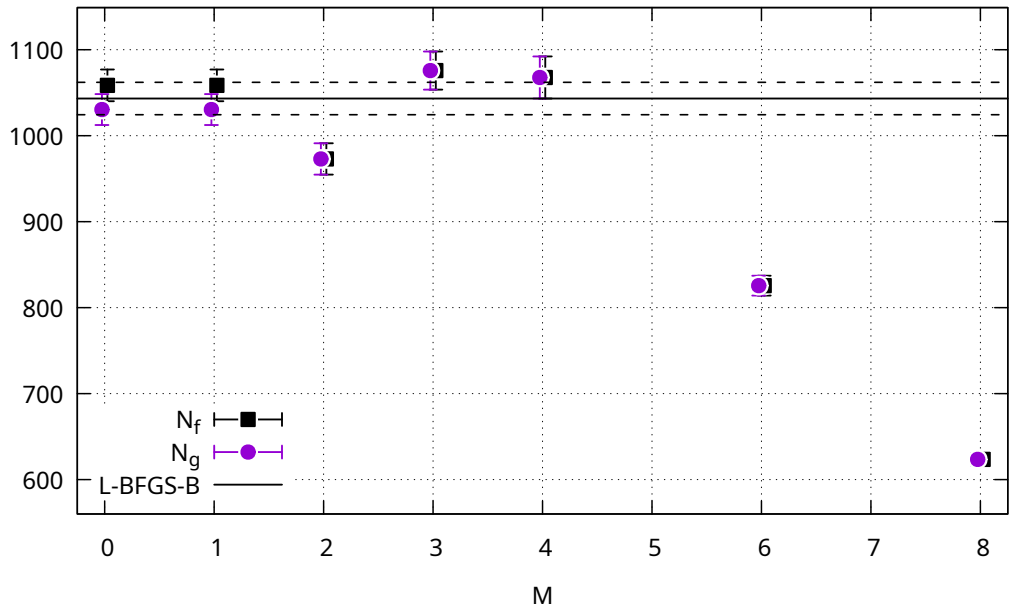

Figure 2: Random quadratics example: averaged number of function N_f and gradient N_g evaluations required for selected set of methods to converge (active stopping criterion was always (47)). Horizontal axis: index M of MS-LBFGS algorithm flavor; vertical axis: mean evaluation counts. Solid horizontal line represents mean number of evaluations for L-BFGS-B (it is the same for functions and gradients), dashed lines depict respective statistical uncertainties.

MS-LBFGS flavors in this range is the gradual alignment of N_f and N_g, which might signify increasingly better Hessian approximation. However, the cases $M = 6, 8$ differ drastically: respective average counts substantially drop down being by factor $\sim 0.8 \div 0.6$ smaller than that for all other algorithms. Taken at face value, this result indicates that MS-LBFGS methods with relatively large number of imposed secant conditions might significantly outperform conventional approximation schemes. It turns out that this qualitative conclusion seems to be correct indeed, as is demonstrated in the next section.

5.2 CUTEst Collection

Extended numerical tests were performed on large collection of unconstrained optimization problems, provided by well-known CUTEst suite [26] available at [27]. In turn, it originates from elder benchmark collections [52], [53] and as such is considered nowadays de facto standard toolset to benchmark optimization software (recent Ref. [54] provides more references on the subject).

Initially, 282 unconstrained optimization problems were identified within CUTEst collection and for all of them we attempted to keep respective parameters at their defaults. However, in several cases we were forced to either totally exclude particular instance or to alter its dimension and possibly replicate it with different number of variables. List of excluded problems (17 in total) with brief explanations is:

- AKIVA, INDEF, PARKCH, STRATEC: numerically unstable (seem not bounded from below, NaN/Inf at initial point);
- FLETCBV2: stationary initial guess;
- DANWOODLS, RAT43LS: L-BFGS-B firmly converge to fake stationary point, hence fair comparison with other algorithms becomes problematic;

	CURLY*	CYCLOOCFLS	DIXON3DQ	INDEFM	SBRYBND	SCOSINE	SCURLY10	SCURLY20 SCURLY30	SSCOSINE
Default	10000	29996	10000	100000	5000	5000	10000	10000	5000
Replaced	100,1000	296,2996	1000	5000,10000	10,50	10	10,100	100	10,100

Table 1: Problems, for which default dimensionality had been replaced with listed dimensions (CURLY* abbreviates three test cases: CURLY10, CURLY20, CURLY30).

	BOXPOWER	BROYDN7D	CHAINWOO	EIGENALS, EIGENBLS	EIGENCLS
Default	20000	5000	4000	2550	2652
Replicated	10000	1000, 10000	1000	110	462
	GENHUMPS	MODBEALE	NONMSQRT	SPINLS	TESTQUAD
Default	5000	20000	4900	1327	5000
Replicated	1000	200	529	862	1000

Table 2: Replicated problems, which were considered both in default and altered dimensions as indicated.

- BA-L16LS, BA-L21LS, BA-L49LS, BA-L52LS, BA-L73LS, FLETCBV3, FLETCHBV, MEYER3, MGH10LS, PENALTY2: high dimensional (in part) problems, for which solution process seems not to enlighten anything.

Furthermore, test cases listed in Table 1 were altered: for each of them instead of default dimension we considered several instances of the same problem with diminished number of variables as indicated. Additionally, in a few cases default problem realization was supplemented with its replicas of different dimensionality, see Table 2. After all these manipulations we were left with 282 - 17 + 8 + 12 = 285 test problems, some of which are still worth to filter out because in MS-LBFGS scheme the number of served secants is always upper bounded by problem dimensionality. Somewhat arbitrary, we decided not to consider test cases with less that 4 variables, which leave us with 212 problems. Finally, to increase available statistics, boost overall reliability and to diminish influence of the initial point (as is provided by CUTEst), we replicated 198 problems with pseudo-random initial guess. Specifically, if x_0 denotes default initial point, then randomized starting coordinates are $x_{0,i} + 0.5\,\xi\,\max(1, |x_{0,i}|)$ with uniformly distributed on $[-1; 1]$ pseudo-random number ξ (respective seed is derived separately for each problem instance). Note that several test cases produced numerically undefined (or close to be undefined) responses at randomized initial location and hence were not replicated this way. Therefore, finally collected test set consists of 410 problems. Among these only in 22 cases it was necessary to alter default values of algorithm parameters presented above. To be precise, we sometimes diminished gradient convergence threshold ε_g^{max} to force attainment of approximately the same solution by different methods and boosted maximal allowed iterations count up to $I_{max} = 2 \cdot 10^4$ in hope to get nominal termination with small gradient.

Performance profiles $\mathcal{P}_g$, eq. (48), for all considered algorithms are plotted on Fig. 3, where left panel refers to MS-LBFGS methods with alike treatment of all secant conditions, while right panel references multi-secant approaches with exactly maintained last secant equation. Due to the selection rule (49) collected statistics is somewhat smaller (only 300 problems participate on Fig. 3), however, it is still sufficiently large to be fairly representative. Only most relevant range $\tau \leq 10$ is displayed on the horizontal axis. Immediate conclusions drawn from Fig. 3) are as follows. First, with respect to $\mathcal{P}_g$ metric all multi-secant based algorithms significantly outperform selected baseline with ratio of performances being $\sim 6 \div 8$ at $\tau \approx 1$ (MS-LBFGS methods are the best in budget in overwhelming number of cases). Second, there is a clearly seen distinction between $M = 0, 1$ and $M \geq 2$ MS-LBFGS flavors: genuine multi-secant algorithms are superior to effectively single-secant methods, no matter what are the particular details of latter implementations. Indeed, left panel of Fig. 3 depicts three (totally) different single-secant approaches, all of which display inferior to $M \geq 2$ results. Third, performances of multi-secant methods

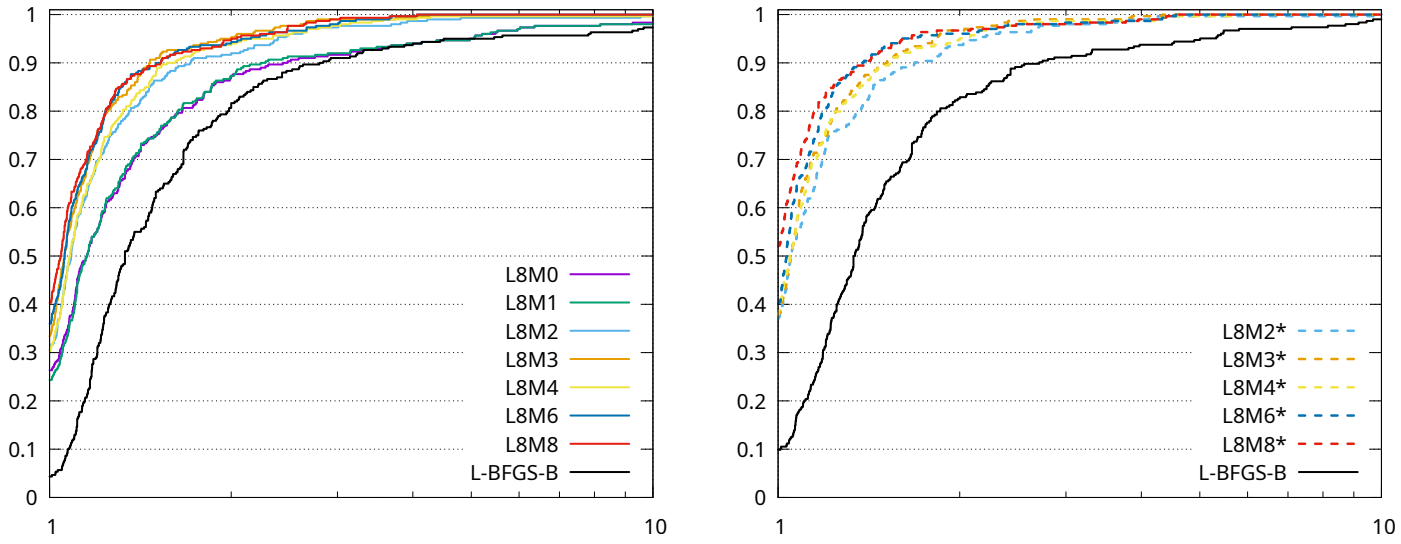

Figure 3: Performance profiles (48) for selected family of MS-LBFGS algorithms and for L-BFGS-B baseline. For all methods memory length is $L = 8$, MS-LBFGS flavors differ by maximal allowed number M of served secant conditions. Left panel: uniform treatment of multiple secants, Section 2.1; right panel: algorithms, which impose last secant condition exactly, Section 2.2; for $M = 0, 1$ these are the same as $M = 0$ on the left and are not displayed for this reason.

are similar although $M = 8$ variants seem to be preferable. Certainly, we aware that performance profiles are to be interpreted with care when discussing relative quality of subleading algorithms [55]. However, it turns out that conclusions drawn from Fig. 3 remain the same even when smaller subsets of different methods are compared. We verified that there is no significant differences in $\mathcal{P}_g$ performances between various multi-secant algorithms regardless of utilized policy to treat last secant relation, while mentioned advantage with respect to single secant approaches remains lucid.

Turn now to the quality measures quantified via parameterized performance profiles $\mathcal{P}_l$ and $\mathcal{P}_s$, eqs. (52) and (54), respectively. Somewhat arbitrary we display on Figs. 4, 5 the performance profiles only for particular μ-values 4, 6, 8, with justification that the results depend only slightly on μ in the examined range $3 \leq \mu \leq 10$ so that selected set is sufficiently representative. In turn, considered μ-range seems to be quite conservative. Profiles were constructed from all available problems, in case of $\mathcal{P}_l$ metric depicted range $\tau \leq 10$ is reduced for clarity of illustration: all $\mathcal{P}_l$-profiles asymptotically converge to unity by construction, but their large τ behavior is not much informative. As before, left panels on each figure represent data for MS-LBFGS algorithms with uniform handling of all secant relations, while to the right we plotted profiles for multi-secant methods, in which last secant equation holds exactly. Everywhere solid black lines stand for respective results for L-BFGS-B baseline. It is immediately obvious that here the conclusions are essentially the same as with $\mathcal{P}_g$ metric. All MS-LBFGS methods significantly outperform the baseline, moreover, genuine multi-secant algorithms, $M \geq 2$, exhibit conspicuous advantage with respect to their effectively single-secant cousins. As far as relative performances are concerned, we cannot fairly insist that more utilized secant relations imply more efficient algorithm. At the level of considered metrics all studied multi-secant methods expose similar characteristics.

It is then instructive to investigate whether considered multi-secant methods are at all different. Ultimately, parameter M regulates only the maximal allowed number of secant relations and it might happen that $M \geq 2$ algorithms effectively utilize some smaller number of secants. To address this we plot on Fig. 6 distribution of actually used secant conditions for each method, averaged over all considered problems. As

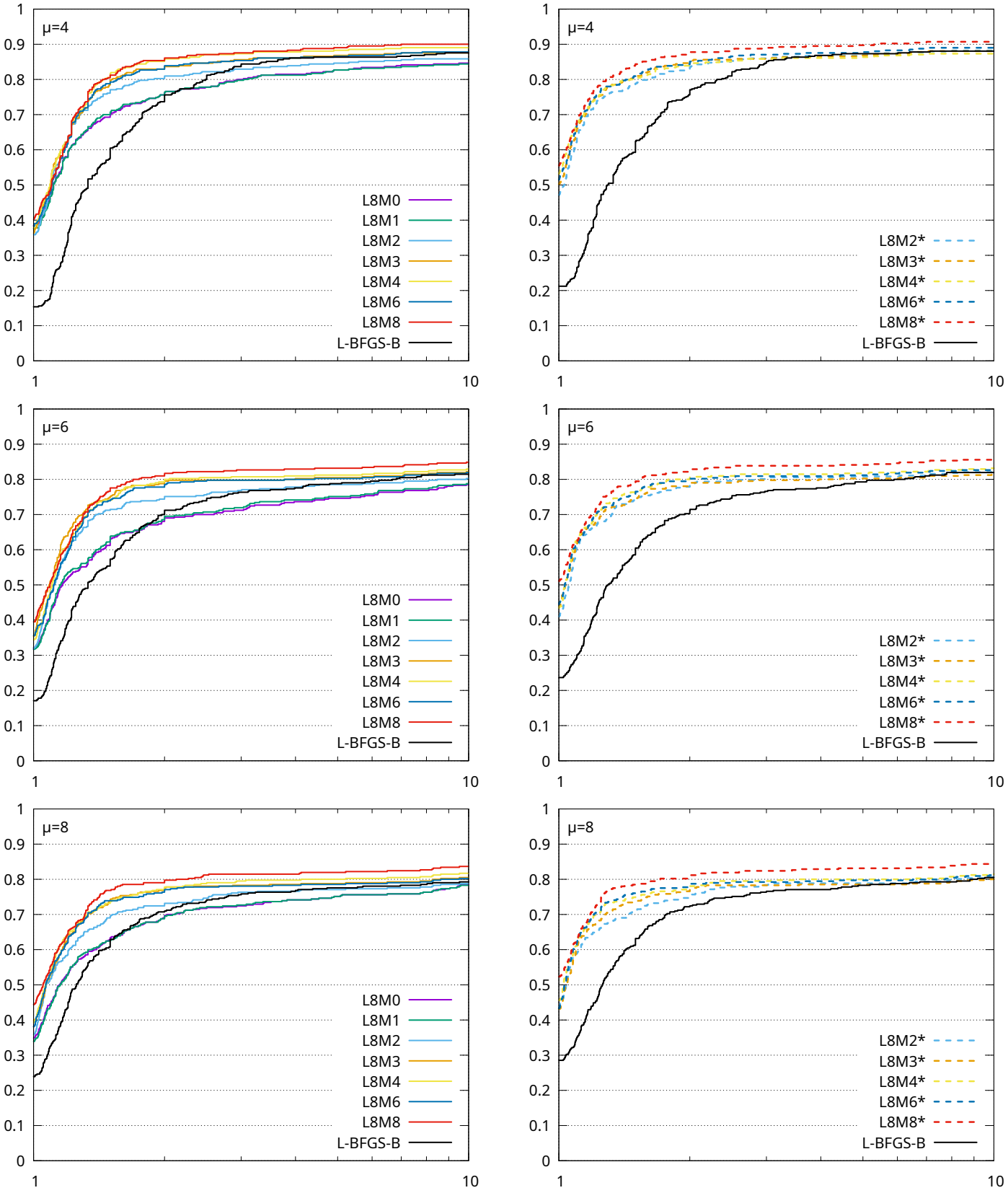

Figure 4: Performance profiles (52), left (right) column depicts methods of Section 2.1 (2.2).

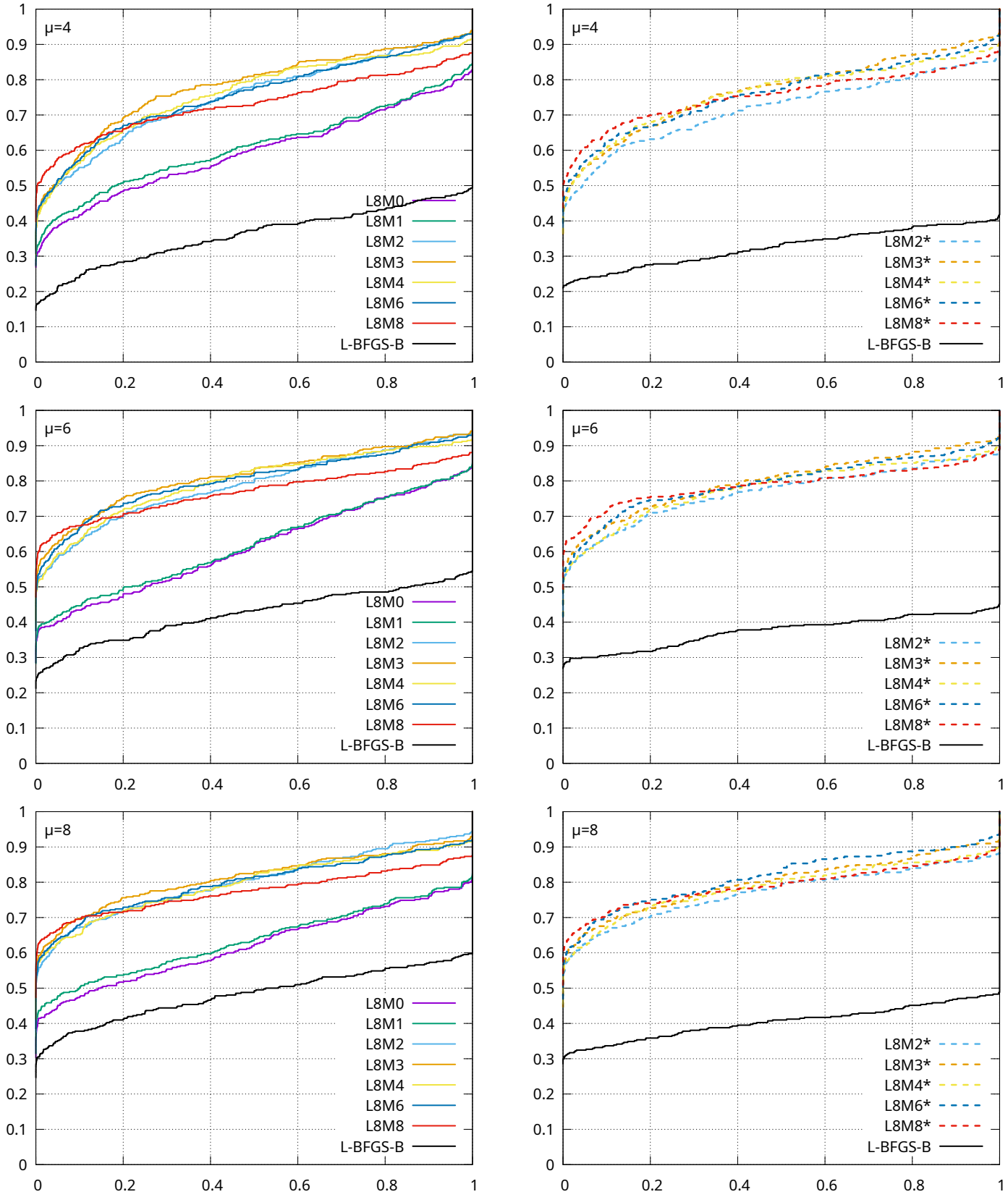

Figure 5: Performance profiles (54), left (right) column depicts methods of Section 2.1 (2.2).

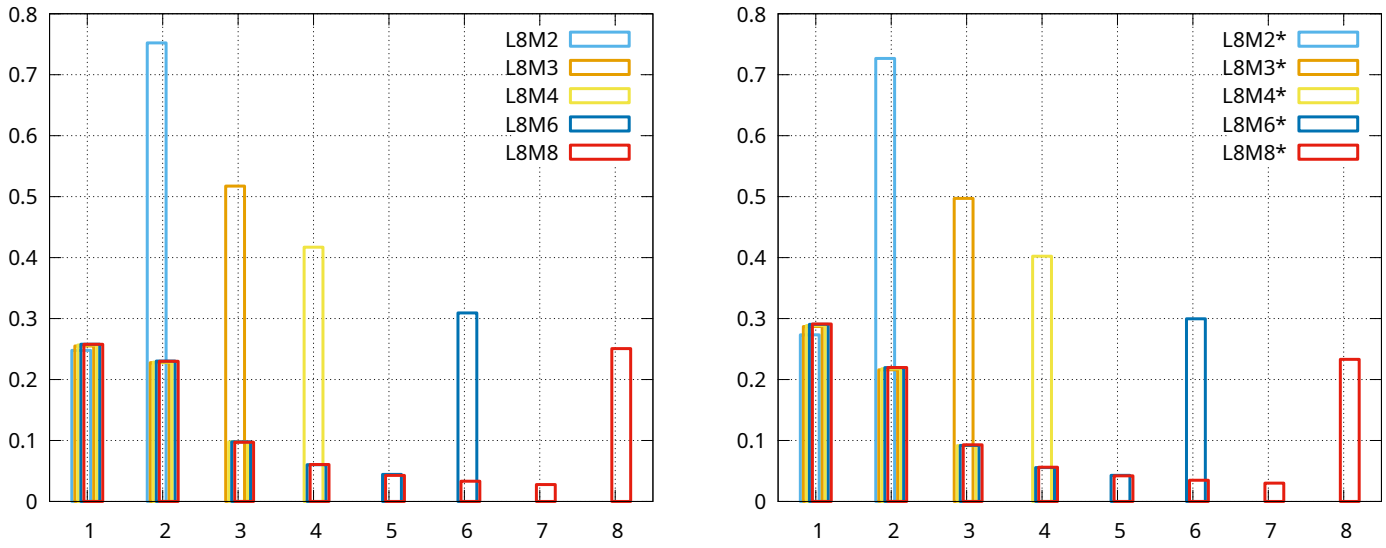

Figure 6: Distributions of actually used number of secant conditions for MS-LBFGS algorithms, averages are taken with respect to all considered problems. For clarity bars are slightly spread horizontally around respective integer values.

it was previously, histograms of the left refer to the algorithms with alike handling of secant conditions, right panel illustrates distributions for the cases with exact last secant equation. It is true that left and right plots are quite similar, however, they do differ and both highlight the same essential point: usage of maximal possible number of secants is significant, respective bins are almost always the highest. This effect is fairly well-pronounced for $M \leq 4$. Its apparent gradual suppression at larger M could be explained by histogram domain enlargement and the fact that secant count could ever increase by at most one at a time, while it is allowed to drop down to unity instantly. Moreover, all distributions qualitatively follow the same pattern of substantial monotonic falloff superposed with single high probability peak at maximal permitted argument value. Therefore, we conclude that multi-secant algorithms of considered family do not dynamically freeze at intermediate number of served secant conditions, to the contrary, maximal allowed number of secants is utilized frequently and this might explain superior performance clearly seen in numerical experiments.

6 Conclusions

In this paper we attempted to revive an old issue of how to self-consistently account for multiple secant conditions within quasi-Newton approximation framework. Potential advantages of multiple secants seem apparent, for this is akin to considering additional data in fitting underdetermined model. However, an immediate problem within quasi-Newton context is that with multiple secants relevant overlap matrix is in general neither symmetric nor positive definite and this is in tension with ultimate goal to have symmetric positive definite (SPD) Hessian approximation. It is true that in the past the issue was addressed many times, however, to our taste none of proposed approaches seem to be fully satisfactory. Generically, there seem to be two main strategies: either to alter the overlap matrix itself until it warrants symmetric positive Hessian or to give up with exact secant equations and to impose them only approximately. Respective weaknesses are evident: the former case inevitably includes rather ad hoc recipes, the latter brings forth an awkward matter of proper selection of weights, which regulate secants exactness. Note that conventional

single secant context eliminates these issues only partially: symmetry becomes automatic, but positivity is still to be assured with appropriate damping schemes.

We argued that there is no inherent inconsistency in quasi-Newton treatment of multiple secant conditions even when overlap is non-symmetric and/or non-positive. In iterative optimization context with finite step sizes overlap is to be asymmetric in general, but this does not preclude neither the existence of underlying SPD Hessian nor the ability to construct SPD approximation to it. Hence the problem is to extract appropriate information from finite coordinate and gradient displacements, which might be assembled to conventional secant equations only in the limit of infinitesimal steps. In this sense inexactness of secant conditions seems natural and unavoidable, thus it alludes to appropriate variational formulation. It turns out that basing on invariance principle suitable Lagrangian with penalized secant equations is easy to guess, moreover, once solution is found penalties could be eliminated analytically. Resulting quasi-Newton scheme possesses, in fact, no external parameters, automatically maintains positivity and looks very similar to conventional BFGS update (hence the name MS-BFGS, reflecting its Multi-Secant nature). Its single-secant version was obtained recently in Ref. [31].

As usual, mere formulation of updating scheme does not imply that it is numerically sound. In single secant iterative context well-definiteness is usually achieved via conditional adjustment (damping) of every newcoming coordinate/gradient displacements pair. With multiple secants things become more complicated because it is impossible/undesirable to alter already accumulated such pairs and modification of only newcoming one doesn't ensure numerical stability. To address this we proposed to select the number actually used secant equations dynamically and to sequentially diminish it down to one once MS update is suspected to become numerically unstable. Then the problem boils down to single secant case, where we suggested alternative damping scheme, which is symmetric with respect to coordinate and gradient differences and is explicitly affine invariant.

The next natural step is to attempt compact representation of MS-BFGS updates and to develop respective limited-memory (MS-LBFGS) scheme. We constructed such representation, which turned out to be structurally reminiscent to well-known compact formulation of BFGS protocol and reduces to it in the limit of single positive curvature secant equation served through all the history. MS-LBFGS updating scheme then follows straightforwardly, however, there are some subtleties not present in single secant context, which we thoroughly discussed.

To assess the practical relevance of proposed MS-LBFGS methodology we implemented a family of corresponding Hessian approximation algorithms, embedded them into not much sophisticated optimization method and performed extensive numerical tests. As is usual with optimization software benchmarking, the problem of prime importance is the selection of large representative set of test problems and the development of adequate and whenever possible complete set of performance measures. The former issue is resolved by using well established CUTEst toolset. As for the latter, observable of paramount importance for us is the expended evaluation budget. Therefore, apart from standard evaluation count criterion, fair application of which might significantly narrow available set of test cases, we suggested to use parameterized families of performance metrics, which reflect the rate of objective decrease at various stages of the solution process. As is coherently revealed by all these performances, MS-LBFGS algorithms significantly outperform their respective single-secant counterparts. Moreover, all implemented methods of MS-LBFGS family are superior to L-BFGS-B baseline although this might not be entirely attributed to advantages of MS-LBFGS Hessian approximation per se (ultimately, here we compare completely different algorithms and exhibited performances might not reduce to only the quality of Hessian approximation). Note, however, that probability of latter event is small as is illustrated by distilled example of random quadratic functions. Nevertheless, to be honest it must not be totally excluded.

References

[1] R.B. Schnabel, *"Quasi-Newton Methods Using Multiple Secant Equations"*, University of Colorado at Boulder report CU-CS-247-83 (1983) [https://doi.org/10.21236/ada131444].

[2] C.G. Broyden, *"A new double-rank minimization algorithm"*, Notices American Math. Soc, 16:670, 1969.

[3] R. Fletcher, *"A new approach to variable metric algorithms"*, The Computer Journal, 13(3):317, 1970.

[4] D. Goldfarb, *"A family of variable metric updates derived by variational means"*, Math. Comp., 24 (109):23–26, 1970.

[5] D.F. Shanno, *"Conditioning of quasi-Newton methods for function minimization"*, Math. Comp., 24 (111):647–656, 1970.

[6] J.E. Dennis, Jr. and J.J. Moré, *"Quasi-Newton Methods, Motivation and Theory"*, SIAM Rev. 19, 46 (1977).

[7] J. Nocedal, S.J. Wright, *"Numerical Optimization"*, Springer, New York, 1999.

[8] R. Fletcher, *"Practical Methods of Optimization"*, (Wiley, Chichester, West Sussex, U.K, 2008).

[9] J. Nocedal, *"Theory of algorithms for unconstrained optimization"*, Acta Numerica, 1 (1992) p. 199-242, doi:10.1017/S0962492900002270

[10] Ph.E. Gill, J.H. Runnoe, *"On Recent Developments in BFGS Methods for Unconstrained Optimization"*, UCSD Center for Computational Mathematics, Technical Report CCoM-22-4, May 1, 2022.

[11] R.H. Byrd, J. Nocedal, R.B. Schnabel, *"Representations of quasi-Newton matrices and their use in limited memory methods"*, Mathematical Programming 63, 129–156 (1994) [https://doi.org/10.1007/BF01582063].

[12] D.C. Liu, J. Nocedal, *"On the limited memory BFGS method for large scale optimization"*, Math. Programming 45, 503 (1989).

[13] J. Nocedal, *"Updating quasi-Newton matrices with limited storage"*, Math. Comput. 35(151), 773–782, 1980 [https://doi.org/10.2307/2006193].

[14] M. J. D. Powell, *"Algorithms for nonlinear constraints that use lagrangian functions"*, Mathematical Programming 14, 224 (1978).

[15] J.B. Erway, M. Rezapour, *"A New Multipoint Symmetric Secant Method with a Dense Initial Matrix"*, arXiv:2107.06321.

[16] M. Lee, Y. Sun, *"Advancing Multi-Secant Quasi-Newton Methods for General Convex Functions"*, arXiv:2504.07330.

[17] Ph. Toint, S. Gratton, *"Multi-Secant Equations, Approximate Invariant Subspaces and Multigrid Optimization"*, Report 07/11, Department of Mathematics, University of Namur, Belgium (2007). [https://optimization-online.org/?p=10277].

[18] J.J. Brust, J.B. Erway, R.F. Marcia, *"Shape-Changing Trust-Region Methods Using Multipoint Symmetric Secant Matrices"*, arXiv:2209.12057.

[19] O.P. Burdakov, J.M. Martínez, E.A. Pilotta, *"A Limited-Memory Multipoint Symmetric Secant Method for Bound Constrained Optimization"*, Annals of Operations Research 117, 51–70 (2002) [https://doi.org/10.1023/A:1021561204463].

[20] O.P. Burdakov, *"Methods of the secant type for systems of equations with symmetric jacobian matrix"*, Numerical Functional Analysis and Optimization 6, (1983) 183.

[21] S. Gratton, V. Malmedy, P.L. Toint, *"Quasi-Newton updates with weighted secant equations"*, Optimization Methods and Software, 30, 748-755 (2015) [https://doi.org/10.1080/10556788.2014.971025].

[22] N. Boutet, R. Haelterman, J. Degroote, *"Secant update version of quasi-Newton PSB with weighted multisecant equations"*, Computational Optimization and Applications, 75, 441 - 466 (2020) [https://doi.org/10.1007/s10589-019-00164-z].

[23] N. Boutet, R. Haelterman, J. Degroote, *"Secant Update generalized version of PSB: a new approach"*, Computational Optimization and Applications, 78, 953 - 982 (2021) [https://doi.org/10.1007/s10589-020-00256-1].

[24] D. Scieur, L. Liu, T. Pumir, N. Boumal, *"Generalization of Quasi-Newton Methods: Application to Robust Symmetric Multisecant Updates"*, ArXiv:2011.03358.

[25] A.F. Izmailov, M.V. Solodov, *"Newton-Type Methods for Optimization and Variational Problems"*, Springer International Publishing, Cham, 2014.

[26] N.I.M. Gould, D. Orban, Ph.L. Toint, *"CUTEst: a constrained and unconstrained testing environment with safe threads for mathematical optimization"*, Computational Optimization and Applications, 60(3):545–557, 2015.

[27] *"The Constrained and Unconstrained Testing Environment with safe threads (CUTEst) for optimization software"*, [https://github.com/ralna/CUTEst].

[28] R. Fletcher, *"A New Variational Result for Quasi-Newton Formulae"*, SIAM J. Optim. 1, 18 (1991).

[29] O. Güler, F. Gürtuna, O. Shevchenko, *"Duality in quasi-Newton methods and new variational characterizations of the DFP and BFGS updates"*, Optimization Methods and Software 24, 45 (2009).

[30] O. Shevchenko, *"Variational Problems in Quasi-Newton Methods"*, (2006).

[31] E. Berglund, J. Zhang, M. Johansson, *"Soft quasi-Newton: guaranteed positive definiteness by relaxing the secant constraint"*, Optimization Methods and Software 40, 783 (2025)

[32] K. Nordström, *"Convexity of the inverse and Moore–Penrose inverse"*, Linear Algebra and Its Applications 434, 1489 (2011).

[33] Gene H. Golub, Charles F. Van Loan, *"Matrix Computations"*, 4th Edition, SIAM Publications Library [https://epubs.siam.org/doi/book/10.1137/1.9781421407944].

[34] I.C.F. Ipsen, C.D. Meyer, *"The Angle Between Complementary Subspaces"*, Am. Math. Mon. 102, 904 (1995).

[35] G.W. Stewart, *"On the Numerical Analysis of Oblique Projectors"*, SIAM J. Matrix Anal. & Appl. 32, 309 (2011).

[36] S. Paternain, A. Mokhtari, A. Ribeiro, *"A Newton-Based Method for Nonconvex Optimization with Fast Evasion of Saddle Points"*, SIAM J. Optim. 29, 343 (2019).

[37] Y. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, Y. Bengio, *"Identifying and Attacking the Saddle Point Problem in High-Dimensional Non-Convex Optimization"*, arXiv:1406.2572

[38] R. H. Byrd, J. Nocedal, *"A Tool for the Analysis of Quasi-Newton Methods with Application to Unconstrained Minimization"*, SIAM J. Numer. Anal. 26, 727 (1989).

[39] D. Goldfarb, Y. Ren, A. Bahamou, *"Practical Quasi-Newton Methods for Training Deep Neural Networks"*, arXiv:2006.08877.

[40] S.G. Nash, J. Nocedal, "A Numerical Study of the Limited Memory BFGS Method and the Truncated-Newton Method for Large Scale Optimization", SIAM J. Optim. 1, 358 (1991).

[41] H.O. Aggrawal, J. Modersitzki, "Hessian Initialization Strategies for L-BFGS Solving Non-Linear Inverse Problems", arXiv:2103.10010.

[42] https://users.iems.northwestern.edu/∼nocedal/lbfgsb.html

[43] R.H. Byrd, P. Lu, J. Nocedal, *"A Limited Memory Algorithm for Bound Constrained Optimization"*, (1995), SIAM Journal on Scientific and Statistical Computing , 16, 5, pp. 1190-1208.

[44] C. Zhu, R.H. Byrd, J. Nocedal, *"L-BFGS-B: Algorithm 778: L-BFGS-B, FORTRAN routines for large scale bound constrained optimization"*, (1997), ACM Transactions on Mathematical Software, Vol 23, Num. 4, pp. 550 - 560.

[45] J.L. Morales, J. Nocedal, *"L-BFGS-B: Remark on Algorithm 778: L-BFGS-B, FORTRAN routines for large scale bound constrained optimization"*, (2011), to appear in ACM Transactions on Mathematical Software.

[46] E. Dolan, J. Moré, *"Benchmarking Optimization Software with Performance Profiles"*, Mathematical Programming, 91(2):201–213, 2002.

[47] J. Moré, S.M. Wild, *"Benchmarking Derivative-Free Optimization Algorithms"*, SIAM Journal on Optimization, 20(1):172–191, 2009.

[48] V. Beiranvand, W. Hare, Y. Lucet, *"Best practices for comparing optimization algorithms"*, Optim Eng 18, 815–848 (2017) [https://doi.org/10.1007/s11081-017-9366-1].

[49] C. Audet, W. Hare, C. Tribes, *"Benchmarking constrained, multi-objective and surrogate-assisted derivative-free optimization methods"*, 29 May 2025, PREPRINT (Version 1) available at Research Square [https://doi.org/10.21203/rs.3.rs-6657064/v1].

[50] M.M. Ali, C. Khompatraporn, Z.B. Zabinsky, *"A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems"*, Journal of Global Optimization, 31(4):635–672, 2005.

[51] A.I.F. Vaz, L.N. Vicente, *"A particle swarm pattern search method for bound constrained global optimization"*, Journal of Global Optimization, 39(2):197–219, 2007.

[52] I. Bongartz, A.R. Conn, N.I.M. Gould, Ph.L. Toint, *"CUTE: Constrained and Unconstrained Testing Environment"*, ACM Transactions on Mathematical Software, 21(1):123–160, 1995.

[53] N.I.M. Gould, D. Orban, Ph.L. Toint, *"CUTEr, a constrained and unconstrained testing environment, revisited"*, ACM Transactions on Mathematical Software, 29(4):373–394, 2003.

[54] S. Gratton, Ph.L. Toint, *"S2MPJ and CUTEst optimization problems for Matlab, Python and Julia"*, arXiv:2407.07812

[55] N. Gould, J. Scott, *"A Note on Performance Profiles for Benchmarking Software"*, ACM Transactions on Mathematical Software, 43 (2016) 1-5.