Limited memory gradient methods for unconstrained optimization

Giulia Ferrandi[1] and Michiel E. Hochstenbach[1]

[1]Department of Mathematics and Computer Science, TU Eindhoven, PO Box 513, Eindhoven, 5600 MB, The Netherlands.

Contributing authors: g.ferrandi@tue.nl; m.e.hochstenbach@tue.nl;

Abstract

The limited memory steepest descent method (Fletcher, 2012) for unconstrained optimization problems stores a few past gradients to compute multiple stepsizes at once. We review this method and propose new variants. For strictly convex quadratic objective functions, we study the numerical behavior of different techniques to compute new stepsizes. In particular, we introduce a method to improve the use of harmonic Ritz values. We also show the existence of a secant condition associated with LMSD, where the approximating Hessian is projected onto a low-dimensional space. In the general nonlinear case, we propose two new alternatives to Fletcher's method: first, the addition of symmetry constraints to the secant condition valid for the quadratic case; second, a perturbation of the last differences between consecutive gradients, to satisfy multiple secant equations simultaneously. We show that Fletcher's method can also be interpreted from this viewpoint.

Keywords: limited memory steepest descent, unconstrained optimization, secant condition, low-dimensional Hessian approximation, Rayleigh–Ritz extraction, Lyapunov equation.

AMS Classification: 65K05 , 90C20 , 90C30 , 65F15 , 65F10

1 Introduction

We study the limited memory steepest descent method (LMSD), introduced by Fletcher [1], in the context of unconstrained optimization problems for a continuously

differentiable function f:

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

The iteration for a steepest descent scheme reads

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \beta_k\,\mathbf{g}_k \;=\; \mathbf{x}_k - \alpha_k^{-1}\,\mathbf{g}_k,$$

where $\mathbf{g}_k = \nabla f(\mathbf{x}_k)$ is the gradient, $\beta_k > 0$ is the steplength, and its inverse $\alpha_k = \beta_k^{-1}$ is usually chosen as an approximate eigenvalue of an (average) Hessian. We refer to [2, 3] for recent reviews on various steplength selection procedures.

The key idea of LMSD is to store the latest $m > 1$ gradients, and to compute (at most) m new stepsizes for the following iterations of the gradient method. We first consider the strictly convex quadratic problem

$$\min_{\mathbf{x}\in\mathbb{R}^n} \tfrac{1}{2}\,\mathbf{x}^T\mathbf{A}\mathbf{x} - \mathbf{b}^T\mathbf{x} \tag{1}$$

where $\mathbf{A}$ is a symmetric positive definite (SPD) matrix with eigenvalues $0 < \lambda_1 \leq \cdots \leq \lambda_n$, and $\mathbf{b} \in \mathbb{R}^n$. Fletcher points out that the m most recent gradients $\mathbf{G} = [\,\mathbf{g}_1 \;\ldots\; \mathbf{g}_m\,]$ form a basis for an m-dimensional Krylov subspace of $\mathbf{A}$. (Although $\mathbf{G}$ will change during the iterations, for convenience, and without loss of generality, we label the first column as $\mathbf{g}_1$.) Then, m approximate eigenvalues of $\mathbf{A}$ (Ritz values) are computed from the low-dimensional representation of $\mathbf{A}$, a projected Hessian matrix, in the subspace spanned by the columns of $\mathbf{G}$, and used as m inverse stepsizes. For $m = 1$, the proposed method reduces to the steepest descent method with Barzilai–Borwein stepsizes [4].

LMSD shares the property with L-BFGS (see, e.g., [5, Ch. 7]), the limited memory version of BFGS, that $2m$ past vectors are stored, of the form $\mathbf{s}_{k-1} = \mathbf{x}_k - \mathbf{x}_{k-1}$ and $\mathbf{y}_{k-1} = \mathbf{g}_k - \mathbf{g}_{k-1}$. While LMSD is a first-order method which incorporates some second-order information in its simplest form (the stepsize), L-BFGS is a quasi-Newton method, which exploits the $\mathbf{s}$-vectors and $\mathbf{y}$-vectors to provide an additive rank-m update of a tentative approximate inverse Hessian (typically a multiple of the identity matrix). Compared to BFGS, at each iteration, the L-BFGS method computes the action of the approximate inverse Hessian, without storing the entire matrix and using $\mathcal{O}(mn)$ operations (see, e.g., [5, Ch. 7]). As we will see in Section 5, the cost of m LMSD iterations is approximately $\mathcal{O}(m^2 n)$, meaning that the costs of the two algorithms are comparable.

There are several potential benefits of LMSD. First, as shown in [1], there are some problems for which LMSD performs better than L-BFGS. Secondly, to the best of our knowledge and as stated in [5, Sec. 6.4], there are no global convergence results for quasi-Newton methods applied to non-convex functions. Liu and Nocedal [6] have proved the global superlinear convergence of L-BFGS only for (twice continuously differentiable) uniformly convex functions. On the contrary, as a gradient method endowed with line search, LMSD converges globally for continuously differentiable functions (see [7, Thm. 2.1], for the convergence of gradient methods combined with nonmonotone line search). Finally, and quite importantly, we note that the idea of LMSD can be readily extended to other types of problems: to name a few, it has been

used in the scaled spectral projected gradient method [8] for constrained optimization problems, and in a stochastic gradient method [9].

Summary of the state of the art and our contributions. In the quadratic case (1), the projected Hessian matrix can be computed from the Cholesky decomposition of $\mathbf{G}^T\mathbf{G}$ (cf. [1, Eq. (19)] and Section 2) without involving any extra matrix-vector product with $\mathbf{A}$. Although this procedure is memory and time efficient, it is also known to be potentially numerically unstable (cf., e.g., the discussion in [10]) because of the computation of the Gramian matrix $\mathbf{G}^T\mathbf{G}$, especially in our context of having an ill-conditioned $\mathbf{G}$. Therefore, we consider alternative ways to obtain the projected Hessian in Section 2.1; in particular, we propose to use the pivoted QR decomposition of $\mathbf{G}$ (see, e.g., [11, Algorithm 1]), or its SVD, and compare the three methods.

In addition, we show that, in the quadratic case, there is a least squares secant condition associated with LMSD. Indeed, in Section 2.3 we prove that the projected Hessian, obtained via one of these three decompositions, is similar to the solution to $\min_{\mathbf{B}} \|\mathbf{Y} - \mathbf{S}\mathbf{B}\|$, where $\|\cdot\|$ denotes the Frobenius norm of a matrix, and $\mathbf{S} = [\,\mathbf{s}_1 \ \dots \ \mathbf{s}_m\,]$ and $\mathbf{Y} = [\,\mathbf{y}_1 \ \dots \ \mathbf{y}_m\,]$ store the m most recent $\mathbf{s}$-vectors and $\mathbf{y}$-vectors, respectively.

Since $\mathbf{Y} = \mathbf{A}\mathbf{S}$ for quadratic functions, the obtained stepsizes are inverse eigenvalues of a projection of the Hessian matrix $\mathbf{A}$. In the general nonlinear case (i.e., for a non-quadratic function f), one can still reproduce the small matrix in [1, Eq. (19)], since the Hessian is not needed explicitly in its computation. However, there is generally not a clear interpretation of the stepsizes as approximate inverse eigenvalues of a certain Hessian matrix. Also, the obtained eigenvalues might even be complex.

To deal with this latter problem, Fletcher proposes a practical symmetrization of [1, Eq. (19)], but, so far, a clear *theoretical justification* for this approach seems to be lacking. To address this issue, we rely on Schnabel's theorem [12, Thm. 3.1] to connect Fletcher's symmetrization to a perturbation of the $\mathbf{Y}$ matrix, of the form $\widetilde{\mathbf{Y}} = \mathbf{Y} + \Delta\mathbf{Y}$. This guarantees that the eigenvalues of the symmetrized matrix [1, Eq. (19)] correspond to a certain symmetric matrix $\mathbf{A}_+$ that satisfies multiple secant equations $\widetilde{\mathbf{Y}} = \mathbf{A}_+\mathbf{S}$ as in the quadratic case. The matrix $\mathbf{A}_+$ can be interpreted as an approximate Hessian in the current iterate.

In the same line of thought, we also exploit one of the perturbations $\widetilde{\mathbf{Y}}$ proposed by Schnabel [12] in the LMSD context. Although the idea of testing different perturbations of $\mathbf{Y}$ is appealing, a good perturbation may be expensive to compute, compared to the task of getting m new stepsizes. Therefore, we explore a different approach based on the modification of the least squares secant condition of LMSD. The key idea is to add a *symmetry constraint* to the secant condition:

$$\min_{\mathbf{B}=\mathbf{B}^T} \|\mathbf{Y} - \mathbf{S}\mathbf{B}\|.$$

Interestingly, the solution to this problem corresponds to the solution of a *Lyapunov equation* (see, e.g., [13]). This secant condition provides a smooth transition from the strictly convex quadratic case to the general case, and its solution has real eigenvalues by construction.

Along with discussing both the quadratic and the general case, we study the computation of *harmonic Ritz values*, which are also considered by Fletcher [1] and Curtis and Guo [14, 15]. For the quadratic case, in Section 2.2, we show that there are some nice symmetries between the computation of the Ritz values of $\mathbf{A}$ by exploiting a basis for the matrix of gradients $\mathbf{G}$, and the computation of the *inverse* harmonic Ritz values of $\mathbf{A}$ by means of $\mathbf{Y}$. Our implementation is different from Fletcher's, but the two approaches show similar performance in the quadratic experiments of Section 5.1. In general, LMSD with harmonic Ritz values appears to show a less favorable behavior than LMSD with Ritz values. Therefore, in Section 2.2, we present a way to improve the quality of the harmonic Ritz values, by taking an extra Rayleigh quotient of the harmonic Ritz vectors. This is based on the remarks in, e.g., [16, 17].

Outline. The rest of the paper is organized as follows. We first focus on the strictly convex quadratic problem (1) in Section 2. We review the LMSD method, as described by Fletcher [1], and present new ways to compute the approximate eigenvalues of the Hessian. We also give a secant condition for the low-dimensional Hessian of which we compute the eigenvalues. We move to the general unconstrained optimization problems in Section 3, where we give a theoretical foundation to Fletcher's symmetrized matrix [1, Eq. (19)], and show how to compute new stepsizes from the secant equation for quadratics, by adding symmetry constraints. A third new approach based on [12] is also proposed. In both Sections 2 and 3, particular emphasis is put on the issue of (likely) numerical rank-deficiency of $\mathbf{G}$ (or $\mathbf{Y}$, when computing the harmonic Ritz values). Section 4 reports the LMSD algorithms for strictly convex quadratic problems, as in [1], and for general continuously differentiable functions, as in [2]. Related convergence results are also recalled. Finally, numerical experiments on both strictly convex quadratics and general unconstrained problems are presented in Section 5; conclusions are drawn in Section 6.

2 Limited memory BB1 and BB2 for quadratic problems

We review Fletcher's limited memory approach [1] for strictly convex quadratic functions (1), and study some new theoretical and computational aspects. Common choices for the steplength in gradient methods for quadratic functions are the Barzilai–Borwein (BB) stepsizes [4]

$$\beta_k^{\mathrm{BB1}} = \frac{\mathbf{g}_{k-1}^T \mathbf{g}_{k-1}}{\mathbf{g}_{k-1}^T \mathbf{A}\, \mathbf{g}_{k-1}}, \qquad \beta_k^{\mathrm{BB2}} = \frac{\mathbf{g}_{k-1}^T \mathbf{A}\, \mathbf{g}_{k-1}}{\mathbf{g}_{k-1}^T \mathbf{A}^2\, \mathbf{g}_{k-1}}. \tag{2}$$

The inverse stepsizes $\alpha_k^{\mathrm{BB1}} = (\beta_k^{\mathrm{BB1}})^{-1}$ and $\alpha_k^{\mathrm{BB2}} = (\beta_k^{\mathrm{BB2}})^{-1}$ are the standard and the harmonic Rayleigh quotients of $\mathbf{A}$, evaluated at $\mathbf{g}_{k-1}$, respectively. Therefore, they provide estimates of the eigenvalues of $\mathbf{A}$. The key idea of LMSD is to produce $m > 1$ approximate eigenvalues from an m-dimensional space simultaneously, hopefully capturing more information compared to that from a one-dimensional space. One hint about why considering $m > 1$ may be favorable is provided by the well-known Courant–Fischer Theorem and Cauchy's Interlace Theorem (see, e.g., [18, Thms. 10.2.1

and 10.1.1]). For two subspaces $\mathcal{V}$, $\mathcal{W}$ with $\mathcal{V} \subseteq \mathcal{W}$, we have

$$\max_{\mathbf{z}\in\mathcal{V},\,\|\mathbf{z}\|=1} \mathbf{z}^T\mathbf{A}\mathbf{z} \leq \max_{\mathbf{z}\in\mathcal{W},\,\|\mathbf{z}\|=1} \mathbf{z}^T\mathbf{A}\mathbf{z} \leq \max_{\|\mathbf{z}\|=1} \mathbf{z}^T\mathbf{A}\mathbf{z} = \lambda_n.$$

Therefore, a larger search space may result in better approximations to the largest eigenvalue of $\mathbf{A}$. Similarly, a larger subspace may better approximate the smallest eigenvalue, as well as the next-largest and the next-smallest values.

We now show why m consecutive gradients form a basis of a Krylov subspace of $\mathbf{A}$. It is easy to check that, given the stepsizes $\beta_1, \ldots, \beta_m$ corresponding to the m most recent gradients, each gradient can be expressed as follows:

$$\mathbf{g}_k = \prod_{i=1}^{k-1} (I - \beta_i \mathbf{A})\, \mathbf{g}_1, \quad k = 1, \ldots, m. \tag{3}$$

Therefore all m gradients belong to the Krylov subspace of degree m (and of dimension at most m)

$$\mathbf{g}_k \in \mathcal{K}_m(\mathbf{A}, \mathbf{g}_1) = \operatorname{span}\{\mathbf{g}_1,\, \mathbf{A}\,\mathbf{g}_1,\, \ldots,\, \mathbf{A}^{m-1}\mathbf{g}_1\}.$$

Moreover, under mild assumptions, the columns of $\mathbf{G}$ form a basis for $\mathcal{K}_m(\mathbf{A}, \mathbf{g}_1)$. This result is mentioned by Fletcher [1]; here we provide an explicit proof.

Proposition 1 *Suppose the gradient $\mathbf{g}_1$ does not lie in an ℓ-dimensional invariant subspace, with $\ell < m$, of the SPD matrix $\mathbf{A}$. If $\beta_k \neq 0$ for all $k = 1, \ldots, m-1$, the vectors $\mathbf{g}_1, \ldots, \mathbf{g}_m$ are linearly independent.*

Proof In view of the assumption, the set $\{\mathbf{g}_1, A\,\mathbf{g}_1 \ldots, A^{m-1}\mathbf{g}_1\}$ is a basis for $\mathcal{K}_m(A, \mathbf{g}_1)$. In fact, from (3),

$$[\,\mathbf{g}_1 \;\; \mathbf{g}_2 \;\; \cdots \;\; \mathbf{g}_m\,] = [\,\mathbf{g}_1 \;\; A\mathbf{g}_1 \;\; \ldots \;\; A^{m-1}\mathbf{g}_1\,] \begin{bmatrix} 1 & \times & \times & \times & & \times \\ & -\beta_1 & \times & \times & & \times \\ & & \beta_1\beta_2 & \times & & \times \\ & & & \ddots & & \times \\ & & & & & (-1)^m \prod_{i=1}^{m-1} \beta_i \end{bmatrix}. \tag{4}$$

Up to a sign, the determinant of the rightmost matrix in this equation is $\beta_1^{m-1}\beta_2^{m-2}\cdots\beta_{m-1}$, which is nonzero if and only if the stepsizes are nonzero. Therefore, $\mathbf{g}_1, \ldots, \mathbf{g}_m$ are linearly independent. □

This result shows that m consecutive gradients of a quadratic function are linearly independent in general; in practice, this formula suggests that small β_i may quickly cause ill conditioning. Numerical rank-deficiency of $\mathbf{G}$ is an important issue in the LMSD method and will be considered in the computation of a basis for $\operatorname{span}(\mathbf{G})$ in Section 2.1.

For the following discussion, we also relate $\mathbf{S}$ and $\mathbf{Y}$ to the Krylov subspace $\mathcal{K}_m(\mathbf{A}, \mathbf{g}_1)$.

Proposition 2 *If* $\mathbf{G}$ *is a basis for* $\mathcal{K}_m(A, \mathbf{g}_1)$*, then*

(i) the columns of $\mathbf{S}$ *also form a basis for* $\mathcal{K}_m(\mathbf{A}, \mathbf{g}_1)$*;*
(ii) the columns of $\mathbf{Y}$ *form a basis for* $\mathbf{A}\,\mathcal{K}_m(\mathbf{A}, \mathbf{g}_1)$*.*

Proof The thesis immediately follows from the relations

$$\mathbf{S} = -\mathbf{G}\mathbf{D}^{-1}, \quad \mathbf{Y} = -\mathbf{A}\mathbf{G}\mathbf{D}^{-1}, \qquad \mathbf{D} = \operatorname{diag}(\alpha_1, \ldots, \alpha_m), \tag{5}$$

where the $\alpha_i = \beta_i^{-1}$ are the latest m inverse stepsizes, ordered from the oldest to the most recent. Note that $\mathbf{D}$ is nonsingular. □

Given a basis for $\mathcal{K}_m(\mathbf{A}, \mathbf{g}_1)$ (or $\mathbf{A}\,\mathcal{K}_m(\mathbf{A}, \mathbf{g}_1)$), one can approximate some eigenpairs of $\mathbf{A}$ from this subspace. The procedure is known as the Rayleigh–Ritz extraction method (see, e.g., [18, Sec. 11.3]) and is recalled in the next section.

2.1 The Rayleigh–Ritz extraction

We formulate the standard and harmonic Rayleigh–Ritz extractions in the context of LMSD methods for strictly convex quadratic functions. Let $\mathcal{S}$ be the subspace spanned by the columns of $\mathbf{S}$, and $\mathcal{Y}$ be the subspace spanned by the columns of $\mathbf{Y}$. Fletcher's main idea [1] is to exploit the Rayleigh–Ritz method on the subspace $\mathcal{S}$. We will now review and extend this approach.

We attempt to extract m promising approximate eigenpairs from the subspace $\mathcal{S}$. Therefore, such approximate eigenpairs can be represented as $(\theta_i, \mathbf{S}\mathbf{c}_i)$, with nonzero $\mathbf{c}_i \in \mathbb{R}^m$, for $i = 1, \ldots, m$. The (standard) Rayleigh–Ritz extraction imposes a Galerkin condition:

$$\mathbf{A}\,\mathbf{S}\,\mathbf{c} - \theta\,\mathbf{S}\,\mathbf{c} \perp \mathcal{S}. \tag{6}$$

This means that the pairs $(\theta_i, \mathbf{c}_i)$ are the eigenpairs of the $m \times m$ pencil $(\mathbf{S}^T\mathbf{Y}, \mathbf{S}^T\mathbf{S})$. The θ_i are called *Ritz values*. In the LMSD method, we have $\mathcal{S} = \mathcal{K}_m(\mathbf{A}, \mathbf{g}_1)$ (see Proposition 2). Note that for $m = 1$, the only approximate eigenvalue reduces to the Rayleigh quotient α^{BB1} (2). Ritz values are bounded by the extreme eigenvalues of $\mathbf{A}$, i.e., $\theta_i \in [\lambda_1, \lambda_n]$. This follows from Cauchy's Interlace Theorem [18, Thm. 10.1.1], by choosing an orthogonal basis for $\mathcal{S}$. This inclusion is crucial to prove the global convergence of LMSD for quadratic functions [1].

Although the matrix of gradients $\mathbf{G}$ (or $\mathbf{S}$) already provides a basis for $\mathcal{S}$, from a numerical point of view it may not be ideal to exploit it to compute the Ritz values, since $\mathbf{G}$ is usually numerically ill conditioned. Therefore, we recall Fletcher's approach [1] to compute a basis for $\mathcal{S}$, and then propose two new variants: via a pivoted QR and via an SVD. Fletcher starts by a QR decomposition $\mathbf{G} = \mathbf{Q}\mathbf{R}$, discarding the oldest gradients whenever $\mathbf{R}$ is numerically singular. Then $\mathbf{Q}$ is an orthogonal basis for a possibly smaller space $\mathcal{S} = \operatorname{span}([\mathbf{g}_{m-s+1}, \ldots, \mathbf{g}_m])$, with $s \leq m$. The product $\mathbf{A}\mathbf{G}$ can be computed from the gradients without additional multiplications by $\mathbf{A}$, in view of

$$\mathbf{A}\mathbf{G} = -\mathbf{Y}\mathbf{D} = [\,\mathbf{G} \;\; \mathbf{g}_{m+1}\,]\,\mathbf{J}, \qquad \text{where} \quad \mathbf{J} = \begin{bmatrix} \alpha_1 & & \\ -\alpha_1 & \ddots & \\ & \ddots & \alpha_m \\ & & -\alpha_m \end{bmatrix}. \tag{7}$$

Here, the relation $\mathbf{y}_{k-1} = \mathbf{g}_k - \mathbf{g}_{k-1}$ is used. Then the $s \times s$ low-dimensional representation of $\mathbf{A}$ can be written in terms of $\mathbf{R}$:

$$\mathbf{T} := \mathbf{Q}^T \mathbf{A} \mathbf{Q} = [\, \mathbf{R} \;\; \mathbf{r} \,]\, \mathbf{J}\, \mathbf{R}^{-1}, \tag{8}$$

where $\mathbf{r} = \mathbf{Q}^T \mathbf{g}_{m+1}$. It is clear that $\mathbf{T}$ is symmetric; it is also tridiagonal in view of the fact that it is associated with a Krylov relation for a symmetric matrix (see also Fletcher [1]). Since $\mathbf{r}$ is also the solution to $\mathbf{R}^T \mathbf{r} = \mathbf{G}^T \mathbf{g}_{m+1}$, the matrix $\mathbf{Q}$ is in fact not needed to compute $\mathbf{r}$. For this reason, Fletcher concludes that the Cholesky decomposition $\mathbf{G}^T \mathbf{G} = \mathbf{R}^T \mathbf{R}$ is sufficient to determine $\mathbf{T}$ and its eigenvalues. Standard routines raise an error when $\mathbf{G}^T \mathbf{G}$ is numerically not SPD (numerically having a zero or tiny negative eigenvalue). If this happens, the oldest gradients are discarded (if necessary one by one in several steps), and the Cholesky decomposition is repeated.

Instead of discarding the oldest gradients $\mathbf{g}_1, \ldots, \mathbf{g}_{m-s}$, we will now consider a new variant by selecting the gradients in the following way. We carry out a pivoted QR decomposition of $\mathbf{G}$, i.e., $\mathbf{G}\widehat{\mathbf{\Pi}} = \widehat{\mathbf{Q}}\widehat{\mathbf{R}}$, where $\widehat{\mathbf{\Pi}}$ is a permutation matrix that iteratively picks the column with the maximal norm after each Gram–Schmidt step [11]. As a consequence, the diagonal entries of $\widehat{\mathbf{R}}$ are ordered nonincreasingly in magnitude. (In fact, we can always ensure that these entries are positive, but since standard routines may output negative values, we consider the magnitudes.)

The pivoted QR approach is also a rank-revealing factorization, although generally less accurate than the SVD (see, e.g., [11]). Let $\widehat{\mathbf{R}}_G$ be the first $s \times s$ block of $\widehat{\mathbf{R}}$ for which $|\widehat{r}_i| > \mathsf{thresh} \cdot |\widehat{r}_1|$, where $\widehat{r}_i$ is the ith diagonal element of $\widehat{R}$ and $\mathsf{thresh} > 0$. A crude approximation to its condition number is $\kappa(\widehat{\mathbf{R}}_G) \approx |\widehat{r}_1| \,/\, |\widehat{r}_s|$. Although this approximation may be quite imprecise, the alternative to repeatedly compute $\kappa(\widehat{\mathbf{R}}_G)$ by removing the last column and row of the matrix at each iteration might take up to $\mathcal{O}(m^4)$ work, which, even for modest values of m, may be unwanted.

The approximation subspace for the eigenvectors of $\mathbf{A}$ is now $\mathcal{S} = \mathrm{span}(\widehat{\mathbf{Q}}_G)$, with $\mathbf{G}\widehat{\mathbf{\Pi}}_G = \widehat{\mathbf{Q}}_G \widehat{\mathbf{R}}_G$, where $\widehat{\mathbf{\Pi}}_G$ and $\widehat{\mathbf{Q}}_G$ are the first s columns of $\widehat{\mathbf{\Pi}}$ and $\widehat{\mathbf{Q}}$, respectively. The upper triangular $\widehat{\mathbf{R}}$ can be partitioned as follows:

$$\widehat{\mathbf{R}} = \begin{bmatrix} \widehat{\mathbf{R}}_G & \widehat{\mathbf{R}}_{12} \\ \mathbf{0} & \widehat{\mathbf{R}}_{22} \end{bmatrix}. \tag{9}$$

As in (8), we exploit (7) to compute the projected Hessian

$$\begin{aligned} \mathbf{B}^{\mathrm{QR}} &:= \widehat{\mathbf{Q}}_G^T \mathbf{A} \widehat{\mathbf{Q}}_G = \widehat{\mathbf{Q}}_G^T \mathbf{A}\, \widehat{\mathbf{G}}\, \widehat{\mathbf{\Pi}}_G \widehat{\mathbf{R}}_G^{-1} = \widehat{\mathbf{Q}}_G^T \,[\, \widehat{\mathbf{Q}}\widehat{\mathbf{R}}\widehat{\mathbf{\Pi}}^{-1} \;\; \mathbf{g}_{m+1} \,]\, \mathbf{J} \widehat{\mathbf{\Pi}}_G \widehat{\mathbf{R}}_G^{-1} \\ &= [\, [\widehat{\mathbf{R}}_G \;\; \widehat{\mathbf{R}}_{12}]\, \widehat{\mathbf{\Pi}}^{-1} \;\; \widehat{\mathbf{Q}}_G^T \mathbf{g}_{m+1} \,]\, \mathbf{J}\, \widehat{\mathbf{\Pi}}_G\, \widehat{\mathbf{R}}_G^{-1}. \end{aligned} \tag{10}$$

Note that, compared to Fletcher's approach, this decomposition removes the unwanted gradients all at once, while in [1] the Cholesky decomposition is repeated every time the $\mathbf{R}$ matrix is numerically singular. Fletcher's $\mathbf{T}$ (8) is a specific case of (10), where $\widehat{\mathbf{\Pi}} = \widehat{\mathbf{\Pi}}_G$ is the identity matrix, and $\widehat{\mathbf{R}}_G$ is the whole $\widehat{\mathbf{R}}$, but where $\mathbf{G}$ only contains $[\mathbf{g}_{m-s+1}, \ldots, \mathbf{g}_m]$.

As the second new variant, we exploit an SVD decomposition $\mathbf{G} = \mathbf{U}\,\mathbf{\Sigma}\,\mathbf{V}^T$, where $\mathbf{\Sigma}$ is $m \times m$, to get a basis for $\mathcal{S}$. An advantage of an SVD is that this provides a natural way to reduce the space by removing the singular vectors corresponding to singular values below a certain tolerance. We decide to retain the $s \leq m$ singular values for which $\sigma_i \geq \mathsf{thresh} \cdot \sigma_1$, where σ_1 is the largest singular value of $\mathbf{G}$. Therefore we consider the truncated SVD $\mathbf{G} \approx \mathbf{G}_1 = \mathbf{U}_G\,\mathbf{\Sigma}_G\,\mathbf{V}_G^T$, where the matrices on the right-hand side are $n \times s$, $s \times s$, and $s \times m$, respectively. Then the approximation subspace becomes $\mathcal{S} = \operatorname{span}(\mathbf{U}_G)$, and we compute the corresponding $s \times s$ representation of $\mathbf{A}$. Since $\mathbf{G}_1\mathbf{V}_G = \mathbf{G}\mathbf{V}_G$, and $\mathbf{U}_G = \mathbf{G}_1\mathbf{V}_G\mathbf{\Sigma}_G^{-1}$, we have, using the expression for $\mathbf{AG}$ (7),

$$\begin{aligned}\mathbf{B}^{\mathrm{SVD}} &= \mathbf{U}_G^T\,\mathbf{A}\mathbf{U}_G = \mathbf{U}_G^T\,\mathbf{A}\mathbf{G}\,\mathbf{V}_G\,\mathbf{\Sigma}_G^{-1} = \mathbf{U}_G^T\,[\,\mathbf{U}\,\mathbf{\Sigma}\,\mathbf{V}^T \quad \mathbf{g}_{m+1}\,]\,\mathbf{J}\,\mathbf{V}_G\,\mathbf{\Sigma}_G^{-1}. \\ &= [\,\mathbf{\Sigma}_G\mathbf{V}_G^T \quad \mathbf{U}_G^T\,\mathbf{g}_{m+1}\,]\,\mathbf{J}\,\mathbf{V}_G\,\mathbf{\Sigma}_G^{-1}. \end{aligned} \tag{11}$$

We remark that, by construction, both $\mathbf{B}^{\mathrm{SVD}}$ and $\mathbf{B}^{\mathrm{QR}}$ are SPD. Due to the truncation of the decompositions of $\mathbf{G}$ in both the pivoted QR and SVD techniques, the subspace $\mathcal{S}$ will generally not be a Krylov subspace, in contrast to Fletcher's method. Still, of course, one can also expect to extract useful information from a non-Krylov subspace.

Since LMSD with Ritz values can be seen as an extension of a gradient method with BB1 stepsizes, it is reasonable to look for a limited memory extension of the gradient method with BB2 stepsizes. The harmonic Rayleigh–Ritz extraction is a suitable tool to achieve this goal.

2.2 The harmonic Rayleigh–Ritz extraction

The use of harmonic Ritz values in the context of LMSD has been mentioned by Fletcher [1, Sec. 7], and further studied by Curtis and Guo [14]. While the Rayleigh–Ritz extraction usually finds good approximations for exterior eigenvalues, the harmonic Rayleigh–Ritz extraction has originally been introduced to approximate eigenvalues close to a target value in the interior of the spectrum. A natural way to achieve this is to consider a Galerkin condition for $\mathbf{A}^{-1}$:

$$\mathbf{A}^{-1}\mathbf{Y}\widetilde{\mathbf{c}} - \widetilde{\theta}^{-1}\mathbf{Y}\widetilde{\mathbf{c}} \perp \mathcal{Y}, \tag{12}$$

which leads to the eigenpairs $(\widetilde{\theta}_i^{-1}, \widetilde{\mathbf{c}}_i)$ of the pair $(\mathbf{Y}^T\mathbf{S},\, \mathbf{Y}^T\mathbf{Y})$. However, since $\mathbf{A}^{-1}$ is usually not explicitly available or too expensive to compute, one may choose a subspace of the form $\mathcal{Y} = \mathbf{A}\,\mathcal{S}$ (see, e.g., [16]). This simplifies the Galerkin condition:

$$\mathbf{A}\mathbf{S}\,\widetilde{\mathbf{c}} - \widetilde{\theta}\,\mathbf{S}\,\widetilde{\mathbf{c}} \perp \mathbf{A}\,\mathcal{S}.$$

The eigenvalues $\widetilde{\theta}_i$ from this condition are called *harmonic Ritz values*. In the limited memory extension of BB2 we set $\mathcal{Y} = \mathbf{A}\,\mathcal{K}_m(\mathbf{A}, \mathbf{g}_1)$, and we know that $\mathbf{Y}$ is a basis for $\mathcal{Y}$ from Proposition 2. Harmonic Ritz values are also bounded by the extreme eigenvalues of $\mathbf{A}$: $\widetilde{\theta}_i \in [\lambda_1, \lambda_n]$; see, e.g., [19, Thm. 2.1]. It is easy to check that the (memory-less) case $m = 1$ corresponds to the computation of the harmonic Rayleigh quotient α^{BB2}.

We have just observed that the Galerkin condition for the harmonic Ritz values can be formulated either in terms of $\mathbf{Y}$ or $\mathbf{S}$. The latter way is presented in the references [1, 14], which again look for a basis of $\mathcal{S}$ by means of a QR decomposition of $\mathbf{G}$. Following the line of [1], the aim is to find the eigenvalues of

$$(\mathbf{Q}^T\mathbf{A}\mathbf{Q})^{-1}\,\mathbf{Q}^T\mathbf{A}^2\mathbf{Q} =: \mathbf{T}^{-1}\,\mathbf{P}, \tag{13}$$

where $\mathbf{G} = \mathbf{Q}\mathbf{R}$. Since $\mathbf{Q}^T\mathbf{A}^2\mathbf{Q}$ involves the product $[\,\mathbf{G}\quad \mathbf{g}_{m+1}\,]^T[\,\mathbf{G}\quad \mathbf{g}_{m+1}\,]$, we determine the Cholesky decomposition of this matrix, to write [1, Eq. (30)]

$$\mathbf{P} = \mathbf{R}^{-T}\mathbf{J}^T \begin{bmatrix}\mathbf{R} & \mathbf{r}\\ \mathbf{0} & \rho\end{bmatrix}^T \begin{bmatrix}\mathbf{R} & \mathbf{r}\\ \mathbf{0} & \rho\end{bmatrix} \mathbf{J}\mathbf{R}^{-1}, \tag{14}$$

where $\mathbf{R}$ is the Cholesky factor of $\mathbf{G}^T\mathbf{G}$, and $\mathbf{r}$ is as in (8). Both $\mathbf{T}$ and $\mathbf{P}$ are symmetric; moreover, while $\mathbf{T}$ is tridiagonal, $\mathbf{P}$ is pentadiagonal. If $\mathbf{G}$ is rank deficient, the oldest gradients are discarded.

Given the similar roles of $\mathbf{S}$ for $\mathbf{A}$ in (6) and of $\mathbf{Y}$ for $\mathbf{A}^{-1}$ in (12), we now consider new alternative ways to find the harmonic Ritz values of $\mathbf{A}$, based on the decomposition of either $\mathbf{Y}$ or $\mathbf{Y}^T\mathbf{Y}$. The aim is to get an $s \times s$ representation of $\mathbf{A}^{-1}$, as we did for $\mathbf{A}$ in Section 2.1. In this context, we need the following (new) relation:

$$\mathbf{A}^{-1}\mathbf{Y} = -\mathbf{G}\mathbf{D}^{-1} = [\,\mathbf{Y}\quad -\mathbf{g}_{m+1}\,]\,\widetilde{\mathbf{J}}, \quad \text{where} \quad \widetilde{\mathbf{J}} = \begin{bmatrix}1 & & \\ \vdots & \ddots & \\ 1 & \cdots & 1\\ 1 & \cdots & 1\end{bmatrix}\mathbf{D}^{-1}. \tag{15}$$

As for (7), this follows from the definition $\mathbf{y}_{k-1} = \mathbf{g}_k - \mathbf{g}_{k-1}$.

We start with the pivoted QR of $\mathbf{Y}$, i.e., $\mathbf{Y}\check{\mathbf{\Pi}} = \check{\mathbf{Q}}\check{\mathbf{R}}$. As in Section 2.1, we truncate the decomposition based on the diagonal values of $\check{\mathbf{R}}$, and obtain $\mathbf{Y}\check{\mathbf{\Pi}}_Y = \check{\mathbf{Q}}_Y\check{\mathbf{R}}_Y$, with

$$\check{\mathbf{R}} = \begin{bmatrix}\check{\mathbf{R}}_Y & \check{\mathbf{R}}_{12}\\ \mathbf{0} & \check{\mathbf{R}}_{22}\end{bmatrix}.$$

Then we project $\mathbf{A}^{-1}$ onto $\mathcal{Y} = \mathrm{span}(\mathbf{Q}_Y)$ to obtain

$$\begin{aligned}\mathbf{H}^{\mathrm{QR}} &= \check{\mathbf{Q}}_Y^T\mathbf{A}^{-1}\check{\mathbf{Q}}_Y = \check{\mathbf{Q}}_Y^T\mathbf{A}^{-1}\,\check{\mathbf{Y}}\,\check{\mathbf{\Pi}}_Y\check{\mathbf{R}}_Y^{-1} = \check{\mathbf{Q}}_Y^T\,[\,\mathbf{Y}\quad -\mathbf{g}_{m+1}\,]\,\widetilde{\mathbf{J}}\,\check{\mathbf{\Pi}}_Y\check{\mathbf{R}}_Y^{-1}\\ &= [\,[\check{\mathbf{R}}_Y\ \check{\mathbf{R}}_{12}]\,\check{\mathbf{\Pi}}^{-1}\quad -\check{\mathbf{Q}}_Y^T\mathbf{g}_{m+1}\,]\,\widetilde{\mathbf{J}}\,\check{\mathbf{\Pi}}_Y\check{\mathbf{R}}_Y^{-1}.\end{aligned} \tag{16}$$

The matrix $\mathbf{H}^{\mathrm{QR}}$ is also symmetric and delivers the reciprocals of harmonic Ritz values; its expression is similar to (10). An approach based on the Cholesky decomposition of $\mathbf{Y}^T\mathbf{Y} = \widetilde{\mathbf{R}}^T\widetilde{\mathbf{R}}$ may also be derived:

$$\mathbf{H}^{\mathrm{CH}} = [\,\widetilde{\mathbf{R}}\quad \widetilde{\mathbf{r}}\,]\,\widetilde{\mathbf{J}}\,\widetilde{\mathbf{R}}^{-1}, \tag{17}$$

with $\widetilde{\mathbf{r}}$ solution to $\widetilde{\mathbf{R}}^T\widetilde{\mathbf{r}} = -\mathbf{Y}^T\mathbf{g}_{m+1}$.

As for the Ritz values, SVD is another viable option. Consider the truncated SVD of $\mathbf{Y}$: $\mathbf{Y}_1 = \mathbf{U}_Y\,\mathbf{\Sigma}_Y\,\mathbf{V}_Y^T$, where $\mathbf{\Sigma}_Y$ is $s\times s$. Since $\mathbf{Y}_1\mathbf{V}_Y = \mathbf{Y}\mathbf{V}_Y$, by using similar arguments as in the derivation of (11), we get the following low-dimensional representation of $\mathbf{A}^{-1}$:

$$\begin{aligned}\mathbf{H}^{\mathrm{SVD}} &= \mathbf{U}_Y^T\mathbf{A}^{-1}\mathbf{U}_Y = \mathbf{U}_Y^T\mathbf{A}^{-1}\mathbf{Y}\,\mathbf{V}_Y\,\mathbf{\Sigma}_Y^{-1} = \mathbf{U}_Y^T\,[\,\mathbf{Y}\;\; -\mathbf{g}_{m+1}\,]\,\widetilde{\mathbf{J}}\,\mathbf{V}_Y\,\mathbf{\Sigma}_Y^{-1}\\ &= [\,\mathbf{\Sigma}_Y\mathbf{V}_Y^T\;\; -\mathbf{U}_Y^T\mathbf{g}_{m+1}\,]\,\widetilde{\mathbf{J}}\,\mathbf{V}_Y\,\mathbf{\Sigma}_Y^{-1}. \end{aligned} \tag{18}$$

Note that, in contrast to $\mathbf{T}^{-1}\mathbf{P}$, the matrix $\mathbf{H}^{\mathrm{SVD}}$ is symmetric and gives the reciprocals of harmonic Ritz values. In addition, the expression for $\mathbf{H}^{\mathrm{SVD}}$ is similar to the one for $\mathbf{B}^{\mathrm{SVD}}$ in (11).

To conclude the section, we mention the following technique, which is new in the context of LMSD. For the solution of eigenvalue problems, it has been observed (e.g., by Morgan [16]) that harmonic Ritz values sometimes do not approximate eigenvalues well, and it is recommended to use the Rayleigh quotients of harmonic Ritz vectors instead. This means that we use $\mathbf{S}\widetilde{\mathbf{c}}_i$ as approximate eigenvectors, and their Rayleigh quotients $\widetilde{\mathbf{c}}_i^T\mathbf{S}^T\mathbf{A}\mathbf{S}\widetilde{\mathbf{c}}_i$ as approximate eigenvalues. This fits nicely with Fletcher's approach: in fact, once we have the eigenvectors $\widetilde{\mathbf{c}}_i$ of $\mathbf{T}^{-1}\mathbf{P}$ (13), we compute their corresponding Rayleigh quotients as $\widetilde{\mathbf{c}}_i^T\mathbf{T}\widetilde{\mathbf{c}}_i$. We remark that, in the one-dimensional case, this procedure reduces to the gradient method with BB1 stepsizes, instead of the BB2 ones.

In Section 5.1 we compare and comment on the different strategies to get both the standard and the harmonic Ritz values. We will see how the computation of the harmonic Rayleigh quotients can result in a lower number of iterations of LMSD, although computing extra Rayleigh quotients involves some additional work in the m-dimensional space.

2.3 Secant conditions for LMSD

We finally show that the low-dimensional representations of the Hessian matrix $\mathbf{A}$ (or its inverse) satisfy a certain secant condition. This result is new in the context of LMSD, and will be the starting point of one of our extensions of LMSD to general unconstrained optimization problems in Section 3. Recall from [4] that the BB stepsizes (2) satisfy a secant condition each, in the least squares sense:

$$\alpha^{\mathrm{BB1}} = \operatorname*{argmin}_{\alpha} \|\mathbf{y} - \alpha\,\mathbf{s}\|, \qquad \alpha^{\mathrm{BB2}} = \operatorname*{argmin}_{\alpha} \|\alpha^{-1}\,\mathbf{y} - \mathbf{s}\|.$$

We now give a straightforward extension of these conditions to the limited memory variant of the steepest descent. We show that there exist $m \times m$ matrices that satisfy a secant condition and share the same eigenvalues as the two pencils $(\mathbf{S}^T\mathbf{Y},\, \mathbf{S}^T\mathbf{S})$, $(\mathbf{Y}^T\mathbf{S},\, \mathbf{Y}^T\mathbf{Y})$. In the quadratic case, when $\mathbf{Y} = \mathbf{A}\mathbf{S}$, the following results correspond to [18, Thm. 11.4.2] and [20, Thm. 4.2], respectively.

Proposition 3 *Let* $\mathbf{S},\, \mathbf{Y} \in \mathbb{R}^{n\times m}$ *be full rank, with* $n \geq m$, *and let* $\mathbf{B},\, \mathbf{H} \in \mathbb{R}^{m\times m}$.

(i) *The unique solution to* $\min_{\mathbf{B}} \|\mathbf{Y} - \mathbf{S}\mathbf{B}\|$ *is* $\mathbf{B} = (\mathbf{S}^T\mathbf{S})^{-1}\mathbf{S}^T\mathbf{Y}$.

(ii) *The unique solution to* $\min_{\mathbf{H}} \|\mathbf{Y}\mathbf{H} - \mathbf{S}\|$ *is* $\mathbf{H} = (\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T\mathbf{S}$.

(iii) *In the quadratic case* (1)*, the eigenvalues of* $\mathbf{B}$ *are the Ritz values of* $\mathbf{A}$ *and the eigenvalues of* $\mathbf{H}$ *are the* inverse *harmonic Ritz values of* $\mathbf{A}$.

Proof The stationarity conditions for the overdetermined least squares problem $\min_{\mathbf{B}} \|\mathbf{Y} - \mathbf{S}\mathbf{B}\|$ are the normal equations $\mathbf{S}^T(\mathbf{Y} - \mathbf{S}\mathbf{B}) = \mathbf{0}$. Since $\mathbf{S}$ is of full rank, $\mathbf{S}^T\mathbf{S}$ is nonsingular, and thus $\mathbf{B} = (\mathbf{S}^T\mathbf{S})^{-1}\mathbf{S}^T\mathbf{Y}$. Part (ii) follows similarly, by exchanging the role of $\mathbf{S}$ and $\mathbf{Y}$. Since $\mathbf{B}$ and $(\mathbf{S}^T\mathbf{Y}, \mathbf{S}^T\mathbf{S})$ have the same eigenvalues, part (iii) easily follows. The same relation holds for the eigenvalues of $\mathbf{H}$ and the eigenvalues of the pencil $(\mathbf{Y}^T\mathbf{S}, \mathbf{Y}^T\mathbf{Y})$. □

Proposition 3 is a good starting point to extend LMSD for solving general unconstrained optimization problems.

3 General nonlinear functions

When the objective function f is a general continuously differentiable function, the Hessian is no longer constant through the iterations, and not necessarily positive definite. In general, there is no SPD approximate Hessian such that multiple secant equations hold (that is, an expression analogous to $\mathbf{Y} = \mathbf{A}\mathbf{S}$ in the quadratic case). This is clearly stated by Schnabel [12, Thm. 3.1].

Theorem 4 *Let* $\mathbf{S}$, $\mathbf{Y}$ *be full rank. Then there exists a symmetric (positive definite) matrix* $\mathbf{A}_+$ *such that* $\mathbf{Y} = \mathbf{A}_+\mathbf{S}$ *if and only if* $\mathbf{Y}^T\mathbf{S}$ *is symmetric (positive definite).*

By inspecting all the expressions derived in Sections 2.1 and 2.2, we observe that only $\mathbf{G}$ and $\mathbf{Y}$ are needed to compute the $m \times m$ matrices of interest for LMSD. However, given that $\mathbf{Y}^T\mathbf{S}$ is in general not symmetric, Theorem 4 suggests that we cannot interpret these matrices as low-dimensional representations of some Hessian matrices.

We propose two ways to restore the connection with Hessian matrices. In Section 3.1, we exploit a technique proposed by Schnabel [12] for quasi-Newton methods. It consists of perturbing $\mathbf{Y}$ to make $\mathbf{Y}^T\mathbf{S}$ symmetric. We show that Fletcher's method can also be interpreted in this way. In Section 3.2, we introduce a second method which does not aim at satisfying multiple secant equations at the same time, but finds the solution to the least squares secant conditions of Proposition 3 by imposing symmetry constraints.

3.1 Perturbation of Y to solve multiple secant equations

In the context of quasi-Newton methods, Schnabel [12] proposes to perturb the matrix $\mathbf{Y}$ of a quantity $\Delta\mathbf{Y} = \widetilde{\mathbf{Y}} - \mathbf{Y}$ to obtain an SPD $\widetilde{\mathbf{Y}}^T\mathbf{S}$. With this strategy, we implicitly obtain a certain SPD approximate Hessian $\mathbf{A}_+$ such that $\widetilde{\mathbf{Y}} = \mathbf{A}_+\mathbf{S}$. We then refer to Sections 2.1 and 2.2 to compute either the Ritz values or the harmonic Ritz values

of the approximate Hessian $\mathbf{A}_+$. Although we only have $\widetilde{\mathbf{Y}}$ at our disposal, and not $\mathbf{A}_+$, this is all that is needed; the procedures in Section 2 do not need to know $\mathbf{A}_+$ explicitly. In addition, Proposition 3 is still valid, after replacing $\mathbf{Y}$ with $\widetilde{\mathbf{Y}}$. We remark that, for our purpose, just a symmetric $\widetilde{\mathbf{Y}}^T\mathbf{S}$ may also be sufficient, since we usually discard negative Ritz values.

In Section 5 we test one possible way of computing $\Delta\mathbf{Y}$, as proposed in [12], and the Ritz values of the associated low-dimensional representation of $\mathbf{A}_+$. This application is new in the context of LMSD. The perturbation is constructed as follows: first, consider the strict negative lower triangle $\mathbf{L}$ of $\mathbf{Y}^T\mathbf{S} - \mathbf{S}^T\mathbf{Y} = -\mathbf{L} + \mathbf{L}^T$, and suppose $\mathbf{S}$ is of full rank. (If not, remove the oldest $\mathbf{s}$-vectors until the condition is satisfied.) Then $\mathbf{Y}^T\mathbf{S} + \mathbf{L}$ is symmetric. Schnabel [12] solves the underdetermined system $\Delta\mathbf{Y}^T\mathbf{S} = \mathbf{L}$, which has $\Delta\mathbf{Y} = \mathbf{S}(\mathbf{S}^T\mathbf{S})^{-1}\mathbf{L}^T$ as minimum norm solution. By Theorem 4, there exists a symmetric $\mathbf{A}_+$ such that $\widetilde{\mathbf{Y}} = \mathbf{A}_+\mathbf{S}$. Now let us consider the QR decomposition of $\mathbf{G}$, which is of full rank since $\mathbf{S}$ is also of full rank. Similar to (7) we know that $\mathbf{A}_+\mathbf{G} = -\widetilde{\mathbf{Y}}\mathbf{D}$. Moreover, we recall that $\mathbf{S} = -\mathbf{G}\mathbf{D}^{-1}$, and that $\mathbf{Y}\mathbf{D} = -[\,\mathbf{G}\;\; \mathbf{g}_{m+1}\,]\,\mathbf{J}$ from (7). Therefore, we obtain the following low-dimensional representation of $\mathbf{A}_+$:

$$\begin{aligned}\mathbf{Q}^T\mathbf{A}_+\mathbf{Q} &= \mathbf{Q}^T\mathbf{A}_+\mathbf{G}\mathbf{R}^{-1} = -\mathbf{Q}^T\widetilde{\mathbf{Y}}\mathbf{D}\mathbf{R}^{-1} = -\mathbf{Q}^T(\mathbf{Y}+\Delta\mathbf{Y})\,\mathbf{D}\mathbf{R}^{-1} \\ &= [\,\mathbf{R}\;\; \mathbf{r}\,]\,\mathbf{J}\mathbf{R}^{-1} + \mathbf{R}(\mathbf{R}^T\mathbf{R})^{-1}\mathbf{D}\mathbf{L}^T\mathbf{D}\mathbf{R}^{-1}, \end{aligned} \tag{19}$$

where $\mathbf{r}$ is the solution to $\mathbf{R}^T\mathbf{r} = \mathbf{G}^T\mathbf{g}_{m+1}$ as in (8). This means that (19) can be computed by means of the Cholesky decomposition of $\mathbf{G}^T\mathbf{G}$ only; the factor $\mathbf{Q}$ is not needed.

We now give a new interpretation of Fletcher's extension of LMSD to general nonlinear problems [1, Sec. 4], in terms of a specific perturbation of $\mathbf{Y}$. Fletcher notices that the matrix $\mathbf{T}$ (8) is an upper Hessenberg matrix and can still be computed from the matrix of gradients, but, because of Theorem 4, there is no guarantee that $\mathbf{T}$ corresponds to a low-dimensional representation of a symmetric approximate Hessian matrix. Since the eigenvalues of $\mathbf{T}$ might be complex, Fletcher proposes to enforce $\mathbf{T}$ to be tridiagonal by replacing its strict upper triangular part with the transpose of its strict lower triangular part. We now show that this operation in fact corresponds to a perturbation of the $\mathbf{Y}$ matrix. To the best of our knowledge, this result is new.

Proposition 5 *Let $\mathbf{T}$ be as in (8) and consider its decomposition $\mathbf{T} = \mathbf{L} + \mathbf{\Lambda} + \mathbf{U}$, where $\mathbf{L}$ ($\mathbf{U}$) is strictly lower (upper) triangular and $\mathbf{\Lambda}$ is diagonal. Moreover, let $\mathbf{G}$ be full rank and $\mathbf{G} = \mathbf{Q}\mathbf{R}$ its QR decomposition. If*

$$\Delta\mathbf{Y} = \mathbf{Q}\,(\mathbf{U} - \mathbf{L}^T)\,\mathbf{D}\mathbf{R}^{-1},$$

then $\widetilde{\mathbf{Y}}^T\mathbf{S} = (\mathbf{Y}+\Delta\mathbf{Y})^T\mathbf{S}$ is symmetric and there exists a symmetric $\mathbf{A}_+$ such that $\widetilde{\mathbf{Y}} = \mathbf{A}_+\mathbf{S}$ and $\mathbf{Q}^T\mathbf{A}_+\mathbf{Q} = \mathbf{L} + \mathbf{\Lambda} + \mathbf{L}^T$.

Proof First, we prove that $\mathbf{S}^T\widetilde{\mathbf{Y}}$ is symmetric. By replacing the expression for $\Delta\mathbf{Y}$ and exploiting the QR decomposition of $\mathbf{G}$, we get

$$\mathbf{S}^T\widetilde{\mathbf{Y}} = -\mathbf{D}^{-1}\mathbf{R}^T\big(-[\,\mathbf{R}\;\; \mathbf{r}\,]\,\mathbf{J}\mathbf{R}^{-1} + \mathbf{U} - \mathbf{L}^T\big)\,\mathbf{R}\mathbf{D}^{-1}$$

$$= -\mathbf{D}^{-1}\mathbf{R}^T\big(-(\mathbf{L}+\mathbf{\Lambda}+\mathbf{U})+\mathbf{U}-\mathbf{L}^T\big)\,\mathbf{R}\mathbf{D}^{-1}$$
$$= \mathbf{D}^{-1}\mathbf{R}^T\big(\mathbf{L}+\mathbf{\Lambda}+\mathbf{L}^T\big)\,\mathbf{R}\mathbf{D}^{-1}.$$

Therefore $\mathbf{S}^T\widetilde{\mathbf{Y}}$ is symmetric; Theorem 4 implies that there exists a symmetric $\mathbf{A}_+$ such that $\widetilde{\mathbf{Y}} = \mathbf{A}_+\mathbf{S}$. From this secant equation, it follows that

$$\mathbf{Q}^T\mathbf{A}_+\mathbf{Q} = -\mathbf{Q}^T\widetilde{\mathbf{Y}}\mathbf{D}\mathbf{R}^{-1} = \big(\mathbf{L}+\mathbf{\Lambda}+\mathbf{L}^T\big)\,\mathbf{R}\mathbf{D}^{-1}(\mathbf{D}\mathbf{R}^{-1}) = \mathbf{L}+\mathbf{\Lambda}+\mathbf{L}^T.$$

□

From this proposition, we are able to provide an upper bound for the spectral norm of the perturbation $\Delta\mathbf{Y}$:

$$\|\Delta\mathbf{Y}\|_2 \leq (\min_i \beta_i \cdot \sigma_m(\mathbf{R}))^{-1}\,\|\mathbf{T}-(\mathbf{L}+\mathbf{\Lambda}+\mathbf{L}^T)\|_2,$$

where $\sigma_{\min}(\mathbf{R})$ is the smallest singular value of $\mathbf{R}$ and $\min_i \beta_i$ is the smallest stepsize among the latest m steps. This suggests that the size of the perturbation $\Delta\mathbf{Y}$ is determined not only by the distance between $\mathbf{T}$ and its symmetrization, as expected, but also by the conditioning of $\mathbf{R}$: if $\mathbf{R}$ is close to singular, the upper bound may be large.

We would like to point out the following important open and intriguing question. While Schnabel solves $\Delta\mathbf{Y}^T\mathbf{S} = \mathbf{L}$ to symmetrize $\mathbf{Y}^T\mathbf{S}$, and Fletcher's update is described in Proposition 5, there may be other choices for the perturbation matrix $\Delta\mathbf{Y}$ that, e.g., have a smaller $\Delta\mathbf{Y}$ in a certain norm. However, obtaining these perturbations might be computationally demanding, compared to the task of getting m new stepsizes. In the cases we have analyzed, the lower-dimensional $\mathbf{A}_+$ can be obtained from the Cholesky decomposition of $\mathbf{G}^T\mathbf{G}$ at negligible cost.

Given the generality of Schnabel's Theorem 4, another possibility that may be explored is a perturbation of $\mathbf{S}$, rather than $\mathbf{Y}$, to symmetrize $\mathbf{S}^T\mathbf{Y}$. This would be a natural choice for computing the harmonic Ritz values given a basis for $\mathbf{Y}$. In this situation, the matrix binding $\mathbf{S}$ and $\mathbf{Y}$ would play the role of an approximate inverse Hessian. A thorough investigation is out of the scope of this paper.

3.2 Symmetric solutions to the secant equations

In this subsection, we explore a second and alternative extension of LMSD. We start from the secant condition of Proposition 3 for a low-dimensional matrix $\mathbf{B}$. The key idea is to impose *symmetry constraints* to obtain real eigenvalues from the solutions to the least squares problems of Proposition 3. Even if the hypothesis of Theorem 4 is not met, this method still fulfills the purpose of obtaining new stepsizes for the LMSD iterations.

The following proposition gives the stationarity conditions to solve the two modified least squares problems. Denote the symmetric part of a matrix by $\mathrm{sym}(\mathbf{A}) := \frac{1}{2}(\mathbf{A}+\mathbf{A}^T)$.

Proposition 6 *Let* $\mathbf{S}$, $\mathbf{Y} \in \mathbb{R}^{n\times m}$ *be full rank, with* $n \geq m$, *and* $\mathbf{B}$, $\mathbf{H} \in \mathbb{R}^{m\times m}$.

(i) The solution to $\min_{\mathbf{B}=\mathbf{B}^T} \|\mathbf{Y}-\mathbf{S}\mathbf{B}\|$ *satisfies* $\mathrm{sym}(\mathbf{S}^T\mathbf{S}\,\mathbf{B}-\mathbf{S}^T\mathbf{Y}) = \mathbf{0}$.

(ii) The solution to $\min_{\mathbf{H}=\mathbf{H}^T} \|\mathbf{Y}\mathbf{H}-\mathbf{S}\|$ *satisfies* $\mathrm{sym}(\mathbf{Y}^T\mathbf{Y}\mathbf{H}-\mathbf{Y}^T\mathbf{S})=\mathbf{0}$.

Proof If $\mathbf{B}$ is symmetric, it holds that

$$\|\mathbf{Y}-\mathbf{S}\mathbf{B}\|^2 = \mathrm{tr}(\mathbf{B}\,\mathbf{S}^T\mathbf{S}\,\mathbf{B} - 2\,\mathrm{sym}(\mathbf{S}^T\mathbf{Y})\,\mathbf{B} + \mathbf{Y}^T\mathbf{Y}),$$

where $\mathrm{tr}(\cdot)$ denotes the trace of a matrix. Differentiation leads to the following stationarity condition for $\mathbf{B}$:

$$\mathbf{S}^T\mathbf{S}\,\mathbf{B} + \mathbf{B}\,\mathbf{S}^T\mathbf{S} = 2\,\mathrm{sym}(\mathbf{S}^T\mathbf{Y}), \tag{20}$$

which is a Lyapunov equation. Since $\mathbf{S}$ is of full rank, its Gramian matrix is positive definite. This implies that the spectra of $\mathbf{S}^T\mathbf{S}$ and $-\mathbf{S}^T\mathbf{S}$ are disjoint, and therefore the equation admits a unique solution (see, e.g., [13] for a review of the Lyapunov equation and properties of its solution). Part (ii) follows similarly. □

It is easy to check that, for $m=1$, $\mathbf{B}$ in part (i) reduces to α^{BB1} and $\mathbf{H}$ in part (ii) to β^{BB2}. Compared to Fletcher's $\mathbf{T}$ matrix (8), the symmetric solutions $\mathbf{B}$ and $\mathbf{H}$ will generally give a larger residual (since they are suboptimal for the unconstrained secant conditions), but they enjoy the benefit that their eigenvalues are guaranteed to be real.

We remark that symmetry constraints also appear in the secant conditions of the BFGS method, and in the symmetric rank-one update (see, e.g., [5, Chapter 6]). While in the BFGS method the approximate Hessians are SPD, provided that the initial approximation is SPD, in the rank-one update method it is possible to get negative eigenvalues. The fundamental difference between LMSD and these methods is that we do not attempt to find an approximate $n\times n$ Hessian matrix.

Even while we do not approximate the eigenvalues of some Hessian, as in the quadratic case of Section 2, it is possible to establish bounds for the extreme eigenvalues of the solutions to the Lyapunov equations of Proposition 6, provided that $\mathrm{sym}(\mathbf{S}^T\mathbf{Y})$ is positive definite. The following result is a direct consequence of [21, Cor. 1].

Proposition 7 *Given the solution* $\mathbf{B}$ *to* (20)*, let* $\lambda_1(\mathbf{B})$ *(*$\lambda_n(\mathbf{B})$*) be the smallest (largest) eigenvalue of* $\mathbf{B}$*. If* $\mathbf{S}$ *is of full rank and* $\mathrm{sym}(\mathbf{S}^T\mathbf{Y})$ *is positive definite, then*

$$[\lambda_1(\mathbf{B}),\,\lambda_n(\mathbf{B})] \subseteq [\lambda_1((\mathbf{S}^T\mathbf{S})^{-1}\,\mathrm{sym}(\mathbf{S}^T\mathbf{Y})),\ \lambda_n((\mathbf{S}^T\mathbf{S})^{-1}\,\mathrm{sym}(\mathbf{S}^T\mathbf{Y}))].$$

If there exists an SPD matrix $\mathbf{A}_+$ *such that* $\mathbf{Y}=\mathbf{A}_+\mathbf{S}$*, then* $[\lambda_1(\mathbf{B}),\,\lambda_n(\mathbf{B})] \subseteq [\lambda_1(\mathbf{A}_+),\,\lambda_n(\mathbf{A}_+)]$.

Proof The first statement directly follows from [21, Cor. 1]. From this we have

$$\lambda_1(\mathbf{B}) \geq -\lambda_1^{-1}(-\mathbf{S}^T\mathbf{S}\,(\mathrm{sym}(\mathbf{S}^T\mathbf{Y}))^{-1}), \quad \lambda_n(\mathbf{B}) \leq -\lambda_n^{-1}(-\mathbf{S}^T\mathbf{S}\,(\mathrm{sym}(\mathbf{S}^T\mathbf{Y}))^{-1}).$$

The thesis follows from the fact that, given a nonsingular matrix $\mathbf{\Lambda}$ with positive eigenvalues, the following equality holds for the largest and the smallest eigenvalue: $\lambda_i(-\mathbf{\Lambda}^{-1}) = -1/\lambda_i(\mathbf{\Lambda})$, where $i=1,\,n$.

When $\mathbf{Y}=\mathbf{A}_+\mathbf{S}$, from Cauchy's Interlace Theorem, the spectrum of $\mathbf{B}$ lies in $[\lambda_1(\mathbf{A}_+),\lambda_n(\mathbf{A}_+)]$. □

An analogous result can be provided for the matrix $\mathbf{H}$ of Proposition 6(ii).

3.3 Solving the Lyapunov equation while handling rank deficiency

The solution to the Lyapunov equation (20) is unique, provided that $\mathbf{S}$ is of full rank. In this section, we propose three options if $\mathbf{S}$ is (close to) rank deficient. As in Section 2, we discuss approaches using a Cholesky decomposition, a pivoted QR factorization, and a truncated SVD. By using the decompositions exploited in Section 2.1 we can either discard some $\mathbf{s}$-vectors (and their corresponding $\mathbf{y}$-vectors) or solve the Lyapunov equation onto the space spanned by the first right singular vectors of $\mathbf{S}$.

In the Cholesky decomposition and the pivoted QR decomposition we remove some of the $\mathbf{s}$-vectors and the corresponding $\mathbf{y}$-vectors, if needed. As we have seen in Section 2.1, in the Cholesky decomposition we discard past gradients until the Cholesky factor $\mathbf{R}$ of $\mathbf{G}^T\mathbf{G}$ is sufficiently far from singular. In this new context of Lyapunov equations, we additionally need the relation (cf. (7))

$$\mathbf{Y} = [\,\mathbf{G} \;\; \mathbf{g}_{m+1}\,]\,\mathbf{K}, \qquad \text{where} \quad \mathbf{K} = \begin{bmatrix} -1 & & \\ 1 & \ddots & \\ & \ddots & -1 \\ & & 1 \end{bmatrix}. \tag{21}$$

Then we can easily compute the matrices present in (20):

$$\mathbf{S}^T\mathbf{S} = \mathbf{D}^{-1}\mathbf{R}^T\mathbf{R}\mathbf{D}^{-1}, \qquad \mathbf{S}^T\mathbf{Y} = -\mathbf{D}^{-1}\mathbf{R}^T[\,\mathbf{R} \;\; \mathbf{r}\,]\,\mathbf{K}. \tag{22}$$

In the pivoted QR, we keep only the columns $\mathbf{G}\widehat{\mathbf{\Pi}}_G$ and $\mathbf{Y}\widehat{\mathbf{\Pi}}_G$, as in Section 2. Let $\mathbf{D}_G$ be the diagonal matrix that stores the inverse stepsizes corresponding to $\mathbf{G}\widehat{\mathbf{\Pi}}_G$. Then

$$\begin{aligned}
\widehat{\mathbf{\Pi}}_G^T\,\mathbf{S}^T\mathbf{S}\,\widehat{\mathbf{\Pi}}_G &= \mathbf{D}_G^{-1}\,\widehat{\mathbf{R}}_G^T\widehat{\mathbf{R}}_G\,\mathbf{D}_G^{-1}, \\
\widehat{\mathbf{\Pi}}_G^T\,\mathbf{S}^T\mathbf{Y}\,\widehat{\mathbf{\Pi}}_G &= -\mathbf{D}_G^{-1}[\,\widehat{\mathbf{R}}_G^T\,[\widehat{\mathbf{R}}_G \;\; \widehat{\mathbf{R}}_{12}]\,\widehat{\mathbf{\Pi}}^{-1} \;\; \widehat{\mathbf{\Pi}}_G^T\mathbf{G}^T\mathbf{g}_{m+1}\,]\,\mathbf{K}\widehat{\mathbf{\Pi}}_G,
\end{aligned}$$

where $\widehat{\mathbf{R}}_G$ and $\widehat{\mathbf{R}}_{12}$ come from the block representation of $\widehat{\mathbf{R}}$ (9).

The third approach involves the SVD of $\mathbf{S}$, instead of the SVD of $\mathbf{G}$ as in Section 2.1. This is due to the fact that the solution to the Lyapunov equation (20) for $\mathbf{S}$ and $\mathbf{Y}$ is not directly related to the solution to the Lyapunov equation for $\mathbf{S}\mathbf{\Lambda}$ and $\mathbf{Y}\mathbf{\Lambda}$, for a nonsingular $\mathbf{\Lambda}$. From the SVD $\mathbf{S} = \widehat{\mathbf{U}}\widehat{\mathbf{\Sigma}}\widehat{\mathbf{V}}^T$, we get

$$\widehat{\mathbf{V}}\widehat{\mathbf{\Sigma}}^2\widehat{\mathbf{V}}^T\,\mathbf{B} + \mathbf{B}\,\widehat{\mathbf{V}}\widehat{\mathbf{\Sigma}}^2\widehat{\mathbf{V}}^T = 2\,\mathrm{sym}(\mathbf{S}^T\mathbf{Y}).$$

To simplify this equation, consider the truncated SVD $\mathbf{S}_1 = \mathbf{U}_S\mathbf{\Sigma}_S\mathbf{V}_S^T$, where $\mathbf{\Sigma}_S$ is $s \times s$, and multiply by $\mathbf{V}_S^T$ on the left and by $\mathbf{V}_S$ on the right. Since $\mathbf{V}_S^T\widehat{\mathbf{V}}\mathbf{\Sigma}^2\widehat{\mathbf{V}}^T = \mathbf{\Sigma}_S^2\mathbf{V}_S^T$,

$$\mathbf{\Sigma}_S^2\,\mathbf{B}_S + \mathbf{B}_S\,\mathbf{\Sigma}_S^2 = 2\,\mathrm{sym}(\mathbf{\Sigma}_S\mathbf{U}_S^T\mathbf{Y}\mathbf{V}_S), \tag{23}$$

where $\mathbf{B}_S = \mathbf{V}_S^T\mathbf{B}\mathbf{V}_S$ is the projection of $\mathbf{B}$ onto $\mathbf{V}_S$. Moreover, from (21) we get

$$\mathbf{U}_S^T\mathbf{Y} = [\,-\mathbf{\Sigma}_S\mathbf{V}_S^T\mathbf{D} \;\; \mathbf{U}_S^T\mathbf{g}_{m+1}\,]\,\mathbf{K}.$$

We remark that it is appropriate to control the truncation of the SVD by the condition number of the coefficient matrix $\mathbf{S}^T\mathbf{S}$, which is $\kappa^2(\mathbf{S})$.

The previous discussion on the three decompositions can also be extended to the secant equation of Proposition 6(ii), to compute the matrix $\mathbf{H}$ and use its eigenvalues directly as stepsizes. Several possibilities may be explored by decomposing either $\mathbf{G}$ or $\mathbf{Y}$ as in Section 2.2 for the harmonic Ritz values. We will not discuss any further details regarding all these methods, but in the experiments in Section 5 we will present results obtained with the Cholesky factorization of $[\,\mathbf{G} \;\; \mathbf{g}_{m+1}\,]^T[\,\mathbf{G} \;\; \mathbf{g}_{m+1}\,]$ as expressed in (14). Then for the quantities in Proposition 6(ii) we have:

$$\mathbf{Y}^T\mathbf{Y} = \mathbf{K}^T \begin{bmatrix} \mathbf{R} & \mathbf{r} \\ \mathbf{0} & \rho \end{bmatrix}^T \begin{bmatrix} \mathbf{R} & \mathbf{r} \\ \mathbf{0} & \rho \end{bmatrix} \mathbf{K},$$

and the matrix $\mathbf{Y}^T\mathbf{S}$ can be obtained from (22).

We note that all Lyapunov equations in this section are of the form $\mathbf{E}^T\mathbf{E}\mathbf{B} + \mathbf{B}\mathbf{E}^T\mathbf{E} = \mathbf{F}$. We describe a practical solution approach. Consider the truncated SVD $\mathbf{E} \approx \mathbf{U}_E\mathbf{\Sigma}_E\mathbf{V}_E^T$, where the singular values in $\mathbf{\Sigma}_E$ satisfy $\sigma_i^2(\mathbf{E}) \geq \mathsf{thresh}\cdot\sigma_1^2(\mathbf{E})$. In case we exploit the Cholesky decomposition or the pivoted QR, an extra truncated SVD might still be appropriate, since these two decompositions do not provide an accurate estimate of $\kappa^2(\mathbf{E})$. By left and right multiplication by $\mathbf{V}_E$, we obtain an expression analogous to (22):

$$\mathbf{\Sigma}_E^2\,\mathbf{B}_E + \mathbf{B}_E\,\mathbf{\Sigma}_E^2 = \mathbf{V}_E^T\mathbf{F}\mathbf{V}_E,$$

where $\mathbf{B}_E = \mathbf{V}_E^T\mathbf{B}\mathbf{V}_E$. Since $\mathbf{\Sigma}_E$ is diagonal, the solution to this Lyapunov equation can be easily found by elementwise division (cf. [13, p. 388]):

$$[\mathbf{B}_E]_{ij} = [\mathbf{V}_E^T\mathbf{F}\mathbf{V}_E]_{ij} \;/\; (\sigma_i^2(\mathbf{E}) + \sigma_j^2(\mathbf{E})).$$

We notice that, in the SVD approach (22), the solution can be found directly from this last step. In addition, we remark that, for the scope of LMSD, it is not necessary to find the solution $\mathbf{B}$ to the original Lyapunov equation.

4 Algorithms and convergence results

In this section we present the LMSD method for strictly convex quadratic functions and general continuously differentiable functions. As mentioned in Section 2, the key idea of both algorithms is to store either the m most recent gradients or $\mathbf{y}$-vectors, to compute up to $s \leq m$ new stepsizes, according to the procedures described in Sections 2–3. These stepsizes are then used in (up to) s consecutive iterations of a gradient method; this group of iterations is referred to as a *sweep* [1].

In Algorithm 1, we report the LMSD method for strictly convex quadratic functions as proposed in [1]. Algorithm 2 is a slight variation of [2, Alg. 2]. Our new approaches differ from these mainly in the way we determine the new stepsizes. Other minor differences will be discussed in the rest of the section.

In both algorithms, we plug in the stepsizes in increasing order, but there is no theoretical guarantee that this choice is optimal in some sense. From a theoretical

viewpoint, the ordering of the stepsizes is irrelevant in a gradient method for strictly convex quadratic functions, as is apparent from (3). In practice, due to rounding errors and other additions to the implementation (such as, e.g., Lines 7–13 of Algorithm 1 and Lines 10 and 13 of Algorithm 2), the stepsize ordering is relevant for both the quadratic and the nonlinear case. For the quadratic case, Fletcher [1] suggests that choosing the increasing order improves the chances of a monotone decrease in both the function value and the gradient norm. Nevertheless, his argument is based on the knowledge of s exact eigenvalues of $\mathbf{A}$ [22].

4.1 Strictly convex quadratic functions

The LMSD method for quadratic functions (1) is described in Algorithm 1, which corresponds to [1, Algorithm "A Ritz sweep algorithm"]. This routine is a gradient method without line search. Particular attention is put into the choice of the stepsize: whenever the function value increases compared to the initial function value of the sweep f_{ref}, Fletcher resets the iterate and computes a new point by taking a Cauchy step (cf. Algorithm 1, Line 9). This ensures that the next function value will not be higher than the current f_{ref}, since the Cauchy step is the solution to the exact line search $\min_\beta f(\mathbf{x}_k - \beta\,\mathbf{g}_k)$. Additionally, every time we take a Cauchy step, or the norm of the current gradient has increased compared to the previous iteration, we clear the stack of stepsizes and compute new (harmonic) Ritz values. At each iteration, a new gradient or $\mathbf{y}$-vector is stored, depending on the method chosen to approximate the eigenvalues of $\mathbf{A}$ (cf. Section 2).

Algorithm 1 LMSD for strictly convex quadratic functions [1]

Input: Function $f(\mathbf{x}) = \frac{1}{2}\,\mathbf{x}^T\mathbf{A}\mathbf{x} - \mathbf{b}^T\mathbf{x}$ with $\mathbf{A}$ SPD, initial guess $\mathbf{x}_0$, initial stepsize $\beta_0 > 0$, tolerance tol

Output: Approximation to minimizer $\operatorname{argmin}_{\mathbf{x}} f(\mathbf{x})$

1: $\mathbf{g}_0 = \nabla f(\mathbf{x}_0), \quad f_{\text{ref}} = f(\mathbf{x}_0)$
2: $j = 0, \quad s = 1$ # s is the stack size
3: **for** $k = 0, 1, \ldots$
4: $\quad \nu_k = \beta_j, \quad j = j + 1$
5: $\quad \mathbf{x}_{k+1} = \mathbf{x}_k - \nu_k\,\mathbf{g}_k$
6: $\quad$ **if** $\|\mathbf{g}_{k+1}\| \le \mathsf{tol} \cdot \|\mathbf{g}_0\|$, **return**, **end**
7: $\quad$ **if** $f(\mathbf{x}_{k+1}) \ge f_{\text{ref}}$
8: $\quad\quad$ Reset $\mathbf{x}_{k+1} = \mathbf{x}_k$, clear the stack
9: $\quad\quad$ Reset $\beta_1 = \mathbf{g}_k^T\mathbf{g}_k \,/\, \mathbf{g}_k^T\mathbf{A}\mathbf{g}_k$ # Cauchy stepsize
10: $\quad\quad$ **continue**
11: $\quad$ **else**
12: $\quad\quad$ **if** $\|\mathbf{g}_{k+1}\| \ge \|\mathbf{g}_k\|$, clear the stack, **end**
13: $\quad$ **end**
14: $\quad$ **if** empty stack **or** $j > s$
15: $\quad\quad$ Compute stack of $s \le m$ new stepsizes β_j, ordered increasingly
16: $\quad\quad$ $j = 1$, $f_{\text{ref}} = f(\mathbf{x}_{k+1})$, **end**
17: **end**

It is possible to implement LMSD without controlling the function value of the iterates or the gradient norm, as in [15]. Here Curtis and Guo also show the R-linear convergence of the method. However, in our experiments, we have noticed that this latter implementation converges slower than Fletcher's (for quadratic problems).

To the best of our knowledge, an aspect that has not been discussed yet is the presence of rounding errors in the low-dimensional representation of the Hessian. Except for (13), all the obtained matrices are symmetric, but their expressions are not. Therefore, in a numerical setting, it might happen that a representation of the Hessian is not symmetric. This may result in negative or complex eigenvalues; for this reason, we enforce symmetry by taking the symmetric part of the projected Hessian, i.e., $\mathbf{B} \leftarrow \frac{1}{2}(\mathbf{B}+\mathbf{B}^T)$, which is the symmetric matrix nearest to $\mathbf{B}$. In the Cholesky decomposition, we replace the upper triangle of $\mathbf{T}$ with the transpose of its lower triangle, in agreement with Fletcher's choice for the unconstrained case (cf. [1] and Section 3.1). In both situations, we discard negative eigenvalues, which may still arise.

In practice, we observe that the non-symmetry of a projected Hessian appears especially in problems with large $\kappa(\mathbf{A})$, for a relatively large choice of m (e.g., $m = 10$) and a small value of thresh (e.g., $\mathsf{thresh} = 10^{-10}$). In this situation, the Cholesky decomposition seems to produce a non-symmetric projected Hessian more often than pivoted QR or SVD. This is likely related to the fact that the Cholesky decomposition of an ill-conditioned Gramian matrix leads to a more inaccurate $\mathbf{R}$ factor (cf. Section 1). In addition, the symmetrized $\mathbf{T}$ seems to generate negative eigenvalues more often than the Hessian representations obtained via pivoted QR and SVD. However, these aspects may not directly affect the performance of LMSD. As we will see in Section 5.1, the adoption of different decompositions does not seem to influence the speed of LMSD.

We finally note that for smaller values of m, such as $m = 5$, the projected Hessian tends to be numerically symmetric even for a small thresh. In fact, fewer gradients form a better condition matrix, because of the following argument. First, we have $\sigma_i^2(\mathbf{G}) = \lambda_{m-i+1}(\mathbf{G}^T\mathbf{G})$, for $i = 1, \ldots, m$. Since the Gramian matrix of the $s \leq m$ most recent gradients $[\mathbf{g}_{m-s+1}, \ldots, \mathbf{g}_m]$ is a submatrix of $\mathbf{G}^T\mathbf{G}$, from Cauchy's Interlace Theorem (see, e.g., [18, Thms. 10.2.1 and 10.1.1]), we get that $\sigma_{\min}(\mathbf{G}) \leq \sigma_{\min}([\mathbf{g}_{m-s+1}, \ldots, \mathbf{g}_m])$ and $\sigma_{\max}(\mathbf{G}) \geq \sigma_{\max}([\mathbf{g}_{m-s+1}, \ldots, \mathbf{g}_m])$. This proves that $\kappa([\mathbf{g}_{m-s+1}, \ldots, \mathbf{g}_m]) \leq \kappa(\mathbf{G})$.

4.2 General nonlinear functions

We now review the limited memory steepest descent for general unconstrained optimization problems, as implemented in [2] and reported in Algorithm 2. Compared to the gradient method for strictly convex quadratic functions, LMSD for general nonlinear functions has more complications. In Section 3 we have proposed two alternative ways to find a set of real eigenvalues to use as stepsizes. However, we may still get negative eigenvalues. This problem also occurs in classical gradient methods, when $\mathbf{s}_k^T\mathbf{y}_k < 0$: in this case, the standard approach is to replace any negative stepsize with a positive one. In LMSD, we keep $s \leq m$ positive eigenvalues and discard the negative ones. If all eigenvalues are negative, we restart from $\beta_k = \max(\min(\|\mathbf{g}_k\|_2^{-1}, 10^5),\ 1)$ as in [7]. Moreover, as in [2], only the latest s gradients are kept. As an alternative to

this strategy, we also mention the more elaborated approach of Curtis and Guo [14], which involves the simultaneous computation of Ritz and harmonic Ritz values.

The line search of LMSD in [2] is inspired by Algorithm 1 for quadratic functions. Once new stepsizes have been computed, at each sweep we produce a new iterate starting from the smallest stepsize in the stack. The reference function value f_{ref} for the Armijo sufficient decrease condition is the function value at the beginning of the sweep, as in Algorithm 1. We note that this Armijo type of line search appropriately replaces the exact line search of Algorithm 1, i.e., the choice of the Cauchy stepsize when a nonmonotone behavior (with respect to f_{ref}) is observed. The stack of stepsizes is cleared whenever the current steplength needs to be reduced to meet the sufficient decrease condition, or when the new gradient norm is larger than the previous one. This requirement is also present in Algorithm 1. Notice that, since we terminate the sweep whenever a backtracking step is performed, starting from the smallest stepsizes decreases the likelihood of ending a sweep prematurely. In contrast with [2], we keep storing the past gradients even after clearing the stack. This choice turns out to be favorable for the experiments in Section 5.2.

Algorithm 2 LMSD for general nonlinear functions [2]

Input: Continuously differentiable function f, initial guess $\mathbf{x}_0$, initial stepsize $\nu_0 > 0$, tolerance tol; safeguarding parameters $\beta_{\max} > \beta_{\min} > 0$; line search parameters c_{ls}, $\sigma_{\text{ls}} \in (0, 1)$

Output: Approximation to minimizer $\operatorname{argmin}_{\mathbf{x}} f(\mathbf{x})$

$\mathbf{g}_0 = \nabla f(\mathbf{x}_0), \quad \beta_1 = \nu_0, \quad f_{\text{ref}} = f(\mathbf{x}_0)$
$j = 0, \quad s = 1$ # s is the stack size
for $k = 0, 1, \ldots$
  $\nu_k = \max(\beta_{\min},\ \min(\beta_j,\ \beta_{\max}))$
  if $f(\mathbf{x}_k - \nu_k \mathbf{g}_k) \leq f_{\text{ref}} - c_{\text{ls}}\, \nu_k\, \|\mathbf{g}_k\|^2$
    $\mathbf{x}_{k+1} = \mathbf{x}_k - \nu_k \mathbf{g}_k$
  else
    while $f(\mathbf{x}_k - \nu_k\, \mathbf{g}_k) > f_{\text{ref}} - c_{\text{ls}}\, \nu_k\, \|\mathbf{g}_k\|^2$ **do** $\nu_k = \sigma_{\text{ls}}\, \nu_k$ **end**
    $\mathbf{x}_{k+1} = \mathbf{x}_k - \nu_k \mathbf{g}_k$
    clear the stack
  end
  if $\|\mathbf{g}_{k+1}\| \leq \mathsf{tol} \cdot \|\mathbf{g}_0\|$, **return**, **end**
  if $\|\mathbf{g}_{k+1}\| \geq \|\mathbf{g}_k\|$, clear the stack **end**
  $j = j + 1$
  if empty stack **or** $j > s$
    Compute stack of $s \leq m$ new stepsizes $\beta_j > 0$, ordered increasingly
    Store only last s vectors of $\mathbf{G}$
    $j = 1, \quad f_{\text{ref}} = f(\mathbf{x}_{k+1})$
  end
end

We remark that, by construction, all new function values within a sweep are smaller than f_{ref}. Therefore, the line search strategy adopted in [2] can be seen as a nonmonotone line search strategy [23]. Given the uniform bounds imposed on the sequence of stepsizes, the result of global convergence for a gradient method with nonmonotone line search [7, Thm. 2.1] also holds for Algorithm 2.

5 Numerical experiments

We explore the several variants of LMSD, for the quadratic and for the general unconstrained case. We compare LMSD with a gradient method with ABB_{min} stepsizes [24]. As claimed by Fletcher [1], we have observed that LMSD may indeed perform better than L-BFGS on some problems. However, in the majority of our test cases, L-BFGS, as implemented in [25], converges faster than LMSD, in terms of number of function (and gradient) evaluations, and computational time. The comparison with another gradient method seems fairer to us than the comparison with a second-order method, and therefore we will not show L-BFGS in our study. Nevertheless, as discussed in Section 1, we recall the two main advantages of considering LMSD methods: the possibility to extend its idea to problems beyond unconstrained optimization (see, e.g., [8, 9]), and the less stringent requirements on the objective functions to guarantee the global convergence of the method.

5.1 Quadratic functions

The performance of the LMSD method may depend on several choices: the memory parameter m, whether we compute Ritz or harmonic Ritz values, and how we compute a basis for either $\mathcal{S}$ or $\mathcal{Y}$. This section studies how different choices affect the behavior of LMSD in the context of strictly convex quadratic problems (1).

We consider quadratic problems by taking the Hessian matrices from the SuiteSparse Matrix Collection [26]. These are 103 SPD matrices with a number of rows n between 10^2 and 10^4. From this collection we exclude only mhd1280b, nd3k, nos7. The vector $\mathbf{b}$ is chosen so that the solution of $\mathbf{A}\mathbf{x} = \mathbf{b}$ is $\mathbf{x}^* = \mathbf{e}$, the vector of all ones. For all problems, the starting vector is $\mathbf{x}_0 = 10\,\mathbf{e}$, and the initial stepsize is $\beta_0 = 1$. The algorithm stops when $\|\mathbf{g}_k\| \leq \mathsf{tol} \cdot \|\mathbf{g}_0\|$ with $\mathsf{tol} = 10^{-6}$, or when $5 \cdot 10^4$ iterations are reached. We compare the performance of LMSD with memory parameters $m = 3$, 5, 10 with the ABB_{min} gradient method [24]. Its stepsize is defined as

$$\beta_k^{\text{ABB}_{\text{min}}} = \begin{cases} \min\{\beta_j^{\text{BB2}} \mid j = \max\{1, k-m\}, \ldots, k\}, & \text{if } \beta_k^{\text{BB2}} < \eta\, \beta_k^{\text{BB1}}, \\ \beta_k^{\text{BB1}}, & \text{otherwise,} \end{cases}$$

where $m = 5$ and $\eta = 0.8$. Since the performance of ABB_{min} depends less on the choice of m than LMSD, we only show $m = 5$ for ABB_{min}. Among many possible stepsize choices, we compare LMSD with ABB_{min} because the latter method behaves better than classical BB stepsizes on quadratic problems (see, e.g., [27]).

We recall that one ABB_{min} step requires the computation of three inner products of cost $\mathcal{O}(n)$ each. An LMSD sweep is slightly more expensive, involving operations of order $m^2 n$ and (much less important) m^3, but it is performed approximately once

every m iterations. These costs correspond to the decomposition of either $\mathbf{G}$ or $\mathbf{Y}$, the computation of the projected Hessian matrices and their eigenvalues. We also remark that, while pivoted QR and SVD require $\mathcal{O}(m^2n)$ operations, the Cholesky decomposition is $\mathcal{O}(m^3)$, but is preceded by the computation of a Gramian matrix, with cost $\mathcal{O}(m^2n)$.

We consider two performance metrics: the number of gradient evaluations (NGE) and the computational time. The number of gradient evaluations also includes the iterations that had to be restarted with a Cauchy step (cf. Algorithm 1, Line 9). Our experience indicates that computational time may depend significantly on the chosen programming language, and therefore should not be the primary choice in the comparison of the methods. Nevertheless, it is included as an indication, because it takes into account the different costs of an LMSD sweep and m iterations of a gradient method.

The comparison of different methods is made by means of the performance profile [28], as it is implemented in Python's library perfprof. Briefly speaking, the cost of each algorithm per problem is normalized, so that the winning algorithm has cost 1. This quantity is called *performance ratio*. Then we plot the proportion of problems that have been solved within a certain performance ratio. An infinite cost is assigned whenever a method is not able to solve a problem to the tolerance within the maximum number of iterations.

We compare the performance of LMSD where the stepsizes are computed as summarized in Table 1. In the first comparison we only consider methods that involve a

Table 1 Strategies to compute the new stack of stepsizes in LMSD methods for quadratic functions. RQ refers to the computation of Rayleigh quotients from the harmonic Ritz vectors. H stands for "harmonic", the letters G (Y) indicate whether a decomposition has been used to implicitly compute a basis for $\mathbf{G}$ ($\mathbf{Y}$).

Method	Description	Matrix
LMSD-G [1]	Cholesky on $\mathbf{G}^T\mathbf{G}$ to compute the inverse Ritz values of $\mathbf{A}$	$\mathbf{T}$ (8)
LMSD-G-QR	Pivoted QR on $\mathbf{G}$ to compute the inverse Ritz values of $\mathbf{A}$	$\mathbf{B}^{\mathrm{QR}}$ (10)
LMSD-G-SVD	SVD on $\mathbf{G}$ to compute the inverse Ritz values of $\mathbf{A}$	$\mathbf{B}^{\mathrm{SVD}}$ (11)
LMSD-HG [1]	$\mathbf{A} \cdot \mathrm{span}(\mathbf{G})$ to compute the inverse harmonic Ritz values of $\mathbf{A}$	$\mathbf{T}^{-1}\mathbf{P}$ (13)
LMSD-HG-RQ	$\mathbf{A} \cdot \mathrm{span}(\mathbf{G})$ to find the harmonic Ritz vectors $\mathbf{A}$ and compute their inverse Rayleigh quotients (cf. end of Sec. 2.2)	$\mathbf{T}^{-1}\mathbf{P}$ (13)
LMSD-HY	Cholesky on $\mathbf{Y}^T\mathbf{Y}$ to compute the Ritz values of $\mathbf{A}^{-1}$	$\mathbf{H}^{\mathrm{CH}}$ (17)

Cholesky decomposition for simplicity. The Cholesky routine raises an error any time the input matrix is not SPD; therefore no tolerance thresh for discarding old gradients needs to be chosen. The performance profiles for this first experiment are shown in Figure 1 for $m \in \{3, 5, 10\}$, in the performance range $[1,\ 3]$. As m increases all methods improve, both in terms of gradient evaluations and computational time.

The method that performs best, both in terms of NGE and computational time, is LMSD-G for $m = 10$. When $m = 5$, LMSD-HG-RQ performs better than LMSD-G in terms of NGE, but it is more computationally demanding. This is reasonable, since LMSD-HG-RQ has to compute m extra Rayleigh quotients; this operation has an

additional cost of m^3, which can be relatively large for some problems in our collection, where $m^3 \approx n$.

LMSD-HG and LMSD-HY perform similarly, since they are two different ways of computing the same harmonic Ritz values. They generally perform worse than the other two methods; in the case $m = 10$, their performances are comparable with those of LMSD-G for $m = 5$.

In Figure 2 we compare LMSD with ABB_{min}. Given the comments to Figure 1, we decide to compute the Ritz values of the Hessian matrix, by decomposing $\mathbf{G}$ in different ways. Specifically, we compare LMSD-G, LMSD-G-QR and LMSD-G-SVD

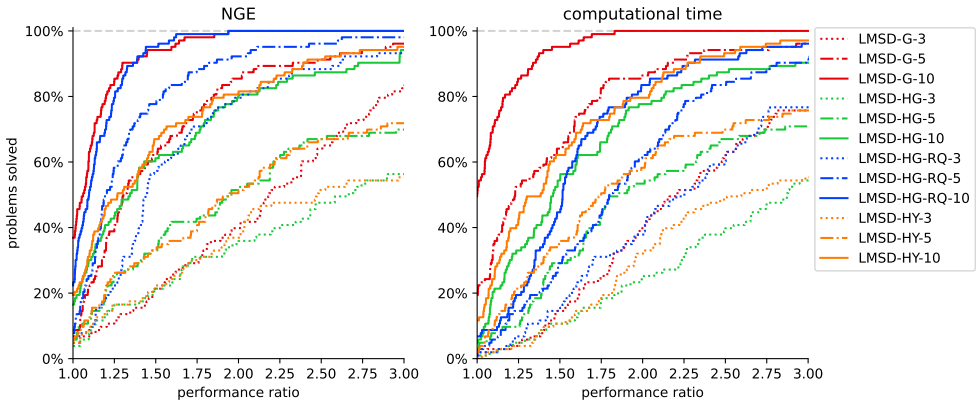

Fig. 1 Performance profile for strictly convex quadratic problems, based on the number of gradient evaluations (left) and computational time (right). Different line types indicate different values for m. Comparison between the computation of Ritz values or harmonic Ritz values.

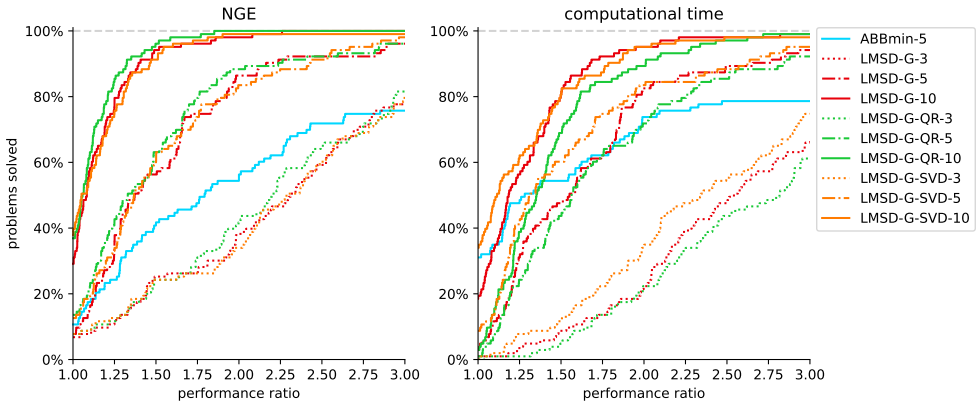

Fig. 2 Performance profile for strictly convex quadratic problems, based on the number of gradient evaluations (left) and computational time (right). Different line types indicate different values for m. Comparison between different decompositions for the matrix $\mathbf{G}$.

(cf. Table 1). The tolerance to decide the memory size $s \leq m$ is set to $\mathsf{thresh} = 10^{-8}$, for both pivoted QR and SVD. Once more, we clearly see that LMSD improves as the memory parameter increases, both in terms of gradient evaluations and computational time. Once m is fixed, the three methods to compute the basis for $\mathcal{S}$ are almost equivalent. LMSD-G-SVD seems to be slightly faster than LMSD-G in terms of computational time, as long as the performance ratio is smaller than 1.5. In our implementation, LMSD-G-QR seems to be more expensive. Compared to ABB_{min}, all LMSD methods with $m = 5, 10$ perform better in terms of gradient evaluations. LMSD-G-SVD, for $m = 10$, appears to be faster than ABB_{min} also in terms of computational time.

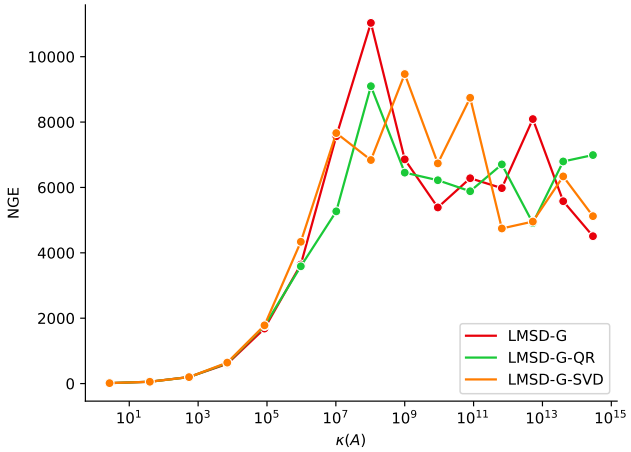

Fig. 3 Condition number of quadratic problems with Hessian matrix $\mathbf{A}$ and corresponding number of gradient evaluations. Different colors indicate different ways of computing the Ritz values of $\mathbf{A}$.

Figure 2 already suggests that different decompositions give approximately equivalent results. In addition, given a problem, it is difficult to recommend a certain decomposition strategy. We illustrate this idea with the following example: consider a family of 15 problems with $\mathbf{A} = \text{diag}(1, \omega, \omega^2, \ldots, \omega^{99})$, where ω assumes 15 values equally spaced in [1.01, 1.4]. Geometric sequences as eigenvalues are quite frequent in the literature; see, e.g., [1, 2]. The starting vector is $\mathbf{x}_0 = \mathbf{e}$, the associated linear system is $\mathbf{A}\mathbf{x} = \mathbf{0}$; the memory parameter is $m = 5$, and each problem is scaled by the norm of the first gradient, so that $\mathsf{tol} = 10^{-7}/\|\mathbf{g}_0\|$. The initial stepsize is $\beta_0 = 0.5$. In Figure 3, we plot the condition number of $\mathbf{A}$ against the number of gradient evaluations. The three methods start to differ already with $\kappa(\mathbf{A}) \approx 10^5$. For a large condition number, there is no clear winner in the performed experiments.

To summarize, when the objective function is strictly convex quadratic, Ritz values seem preferable over harmonic Ritz values. This is emphasized by the improvement of LMSD-HG when taking Rayleigh quotients instead of harmonic Rayleigh quotients. Different decompositions of $\mathbf{G}$ result in mild differences in the performance of LMSD. Even if Cholesky decomposition is the least stable from a numerical point of view, its

instability does not seem to have a clear effect on the performance of LMSD. Finally, we observe that, in all methods, LMSD seems to improve as the memory parameter m increases.

5.2 General unconstrained optimization problems

In this section we want to assess the performance of LMSD for general unconstrained optimization problems, when we choose different methods to compute the stepsizes. These choices are summarized in Table 2. All the methods presented in Section 3

Table 2 Strategies to compute the new stack of stepsizes in LMSD methods for general nonlinear functions. H stands for "harmonic".

Method	Description
LMSD-CHOL [1]	Tridiagonalize $\mathbf{T}$ as in [1] and compute its inverse eigenvalues
LMSD-H-CHOL [14]	Symmetrize $\mathbf{P}^{-1}\mathbf{T}$ as in [14] and compute its eigenvalues
LMSD-LYA	Inverse eigenvalues of the solution to Prop. 6 (i) with Cholesky of $\mathbf{G}^T\mathbf{G}$ to handle rank deficiency
LMSD-LYA-QR	Idem with pivoted QR of $\mathbf{G}$ to handle rank deficiency
LMSD-LYA-SVD	Idem with SVD of $\mathbf{S}$ to handle rank deficiency
LMSD-H-LYA	Eigenvalues of the solution to Prop. 6 (ii) with Cholesky of $[\,\mathbf{G}\ \ \mathbf{g}_{m+1}\,]^T[\,\mathbf{G}\ \ \mathbf{g}_{m+1}\,]$ to handle rank deficiency
LMSD-PERT	Perturb $\mathbf{Y}$ according to [12] to get (19) and compute its inverse eigenvalues

are considered, along with the extension of the harmonic Ritz values computation to the general unconstrained case. This is explained in [14], and indicated as LMSD-H-CHOL. In the quadratic case, the authors point out that the matrix $\mathbf{P}$ (14) can be expressed in terms of $\mathbf{T}$ as $\mathbf{P} = \mathbf{T}^T\mathbf{T} + \boldsymbol{\xi}\boldsymbol{\xi}^T$, where $\boldsymbol{\xi}^T = [\mathbf{0}^T\ \ \rho]\,\mathbf{J}\mathbf{R}^{-1}$. Then, if $\widetilde{\mathbf{T}}$ is the tridiagonal symmetrization of $\mathbf{T}$ as in LMSD-CHOL, the new $\mathbf{P}$ is defined as $\widetilde{\mathbf{P}} = \widetilde{\mathbf{T}}^T\widetilde{\mathbf{T}} + \boldsymbol{\xi}\boldsymbol{\xi}^T$. The new stepsizes are the eigenvalues of $\widetilde{\mathbf{P}}^{-1}\widetilde{\mathbf{T}}$, and are real since $\widetilde{\mathbf{P}}$ is generically SPD, and $\widetilde{\mathbf{T}}$ is symmetric.

All the LMSD methods are tested against the gradient method with nonmonotone line search [7]. The stepsize choice is again $\mathrm{ABB}_{\min}$ with $m = 5$. The nonmonotone line search features a memory parameter $M = 10$; negative stepsizes are replaced by $\beta_k = \max(\min(\|\mathbf{g}_k\|^{-1}, 10^5),\ 1)$, as in [7]. In both algorithms, we set $\beta_{\min} = 10^{-30}$, $\beta_{\max} = 10^{30}$, $c_{\mathrm{ls}} = 10^{-4}$, $\sigma_{\mathrm{ls}} = \frac{1}{2}$, and $\beta_0 = \|\mathbf{g}_0\|^{-1}$. The routine stops when $\|\mathbf{g}_k\| \leq \mathsf{tol} \cdot \|\mathbf{g}_0\|$, with $\mathsf{tol} = 10^{-6}$, or when 10^5 iterations are reached. In LMSD, the memory parameter has been set to $m \in \{3, 5, 7\}$.

We take 31 general differentiable functions from the CUTEst collection [29, 30] and the suggested starting points $\mathbf{x}_0$ therein. The problems are reported in Table 3. Since some test problems are non-convex, we checked whether all gradient methods converged to the same stationary point for different methods. As the performance profile, we may consider three different costs: the number of function evaluations (NFE), the number of iterations, and the computational time. The number of iterations coincides with the number of gradient evaluations for both LMSD and $\mathrm{ABB}_{\min}$.

Before comparing LMSD methods with the $\mathrm{ABB}_{\min}$ gradient method, we discuss the following two aspects of LMSD: the use of different decompositions of either $\mathbf{G}$ or $\mathbf{S}$ in LMSD-LYA, which has been presented in Section 3.3; the number of steps

Table 3 Problems from the CUTEst collection and their sizes.

Problem	n	Problem	n	Problem	n
ARGTRIGLS	200	EIGENBLS	110	MOREBV	5000
CHNROSNB	50	EIGENCLS	462	MSQRTALS	529
COATING	134	ERRINROS	50	MSQRTBLS	529
COSINE	10000	EXTROSNB	1000	NONCVXU2	10000
DIXMAANE1	3000	FLETCHCR	1000	NONCVXUN	10000
DIXMAANF	9000	FMINSURF	1024	NONDQUAR	10000
DIXMAANG	9000	GENHUMPS	5000	SPMSRTLS	10000
DIXMAANH	9000	GENROSE	500	SSBRYBND	5000
DIXMAANJ	9000	LUKSAN11LS	100	TQUARTIC	5000
DIXMAANK	9000	LUKSAN21LS	100		
EIGENALS	110	MODBEALE	2000		

per sweep that are actually used by each LMSD method, in relation with the chosen memory parameter m.

Different decompositions in LMSD-LYA. In the quadratic case, we notice that there is not much difference between the listed decompositions to compute a basis for $\mathcal{S}$. We repeat this experiment with LMSD-LYA, for general unconstrained problems, because the Hessian matrix is not constant during the iterations and therefore the way we discard the past gradients might be relevant. We recall that Cholesky decomposition (LMSD-LYA) discards the oldest gradients first, pivoted QR (LMSD-LYA-QR) selects the gradients in a different order; SVD (LMSD-LYA-SVD) takes a linear combination of the available gradients. For the last two methods, the tolerance to detect linear dependency is set to $\mathsf{thresh} = 10^{-8}$.

Figure 4 shows the three decompositions for $m = 5$. Memory parameters $m = 3, 7$ are not reported as they are similar to the case $m = 5$. The conclusion is the same as for the quadratic case: the decomposition method does not seem to have a large impact on the performance of LMSD. However, for the general case, we remark that while LMSD-LYA solved all problems, both LMSD-LYA-QR and LMSD-LYA-SVD fail to solve one problem each, for all the tested memory parameters. In addition, LMSD-LYA seems more computationally efficient than the other methods. For these two reasons, we continue our analysis by focusing on Cholesky decomposition only.

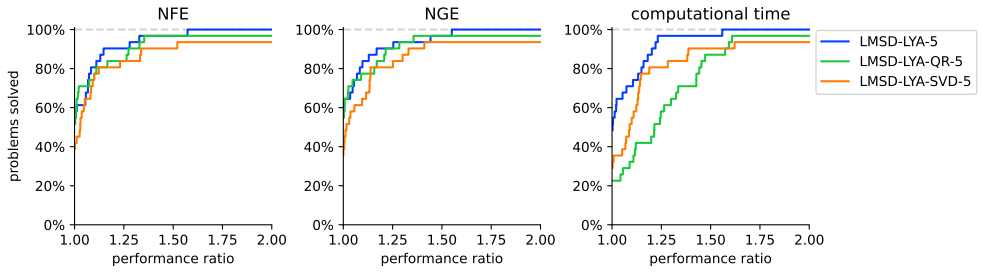

Fig. 4 Performance profile for general unconstrained problems, based on the number of function evaluations, gradient evaluations, and computational time. Comparison between different decompositions for the matrix $\mathbf{G}$ (or $\mathbf{S}$) and $m = 5$.

Average number of stepsizes per sweep. We quantify the efficiency of the various LMSD methods as follows. Ideally, each sweep should provide m new stepsizes, which are supposed to be used in the next m iterations. However, because of the algorithm we adopted, less than m stepsizes are actually employed before the stack is cleared. For each problem and method, we compute the ratio between the number of iterations and the number of sweeps. This gives the average number of stepsizes that are used in each sweep. This value is in $[1, m]$, where the memory parameter m indicates the ideal situation where all the steps are used in a sweep. A method that often uses less than m stepsizes might be inefficient, since the effort of computing new stepsizes (of approximately $\mathcal{O}(m^2 n)$ operations) is not entirely compensated.

The number of iterations per sweep is shown in Figure 5 as a distribution function over the tested problems. An ideal curve should be a step function with a jump in m. For example, when $m = 3$, LMSD-CHOL, i.e., Fletcher's method, tends to use 3 stepsizes on average for approximately 80% of the problems; this is close to the desired situation. When $m = 5$, we notice that LMSD-H-LYA and LMSD-LYA have a similar behavior but for an average smaller than 5. In all cases, LMSD-H-LYA is the curve that shows the lowest average number of steps per sweep. Another interesting behavior is the one of LMSD-PERT, which, for some problems, approaches the largest value m, but, for many others, shows a lower average. In $m = 5, 7$, more than 50% of problems are solved by using only half of the available stepsizes per sweep. This behavior was reflected by the performance profiles of LMSD-PERT: while going from $m = 5$ to $m = 7$, we observed an improvement in terms of the number of function evaluations, but a deterioration in the computational time.

As m increases, the deterioration of the average number of stepsizes per sweep is also visible for the other methods. As already remarked by [1, 2], this suggests that choosing a large value for m does not improve the LMSD methods for general unconstrained problems. This is in contrast with what we have observed in the quadratic case.

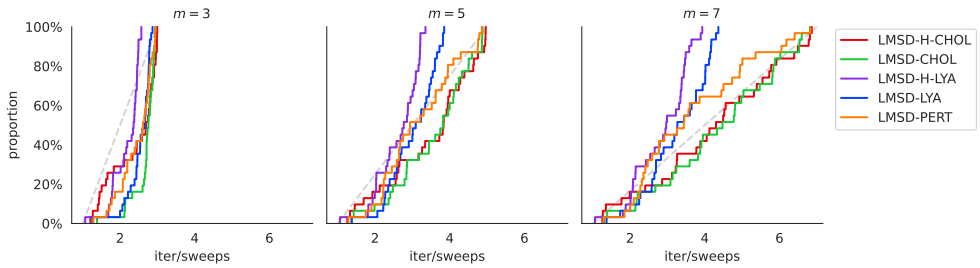

Fig. 5 Cumulative distribution function of the number of iterations per sweep, i.e., the average number of stepsizes per sweep. Curves are based on the tested problems. Straight dashed lines indicate the uniform distribution over $[1, m]$.

Comparison with a gradient method. In what follows, we consider only $m = 5$ for the comparison with ABB_{min}. For LMSD, we do not include $m = 3$ because it

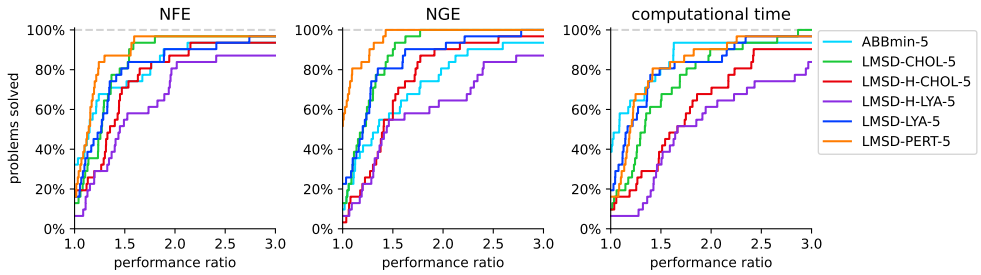

Fig. 6 Performance profile for general unconstrained problems, based on the number of function evaluations, gradient evaluations, and computational time. Comparison between different ways to compute the new stepsizes of a sweep in LMSD, and the gradient method with nonmonotone line search and- the ABB_{min} step.

showed poorer results compared to the simpler nonmonotone gradient method. LMSD for $m = 7$ is not considered since it gives performances similar to $m = 5$, but with a higher computational cost. Results are shown in Figure 6. From the performance profiles related to the computational time, we see that ABB_{min} solves a high proportion of problems with the minimum computational time. LMSD-LYA, LMSD-PERT start competing with the ABB_{min} gradient method when the performance ratio is larger than 1.5.

Regarding the performance profiles for both NFE and NGE, we note a similar pattern: LMSD-PERT has the highest curve; LMSD-LYA and LMSD-CHOL almost overlap for a performance ratio smaller than 1.5; after that, the two curves split, and LMSD-CHOL reaches LMSD-PERT.

The LMSD-PERT method solves 52% of the problems with the minimum number of gradient evaluations. By looking at the performance profile for the NFE, this fact does not seem to be complemented by a low number of function evaluations. Intuitively, this means that LMSD-PERT enters the backtracking procedure more often than the other methods, and it reflects what we have also observed in the central plot of Figure 5. Any time we enter the backtracking procedure, the stack of stepsizes is cleared and the sweep is terminated. Then, the more backtracking we need, the smaller the number of stepsizes per sweep we use.

LMSD-H-CHOL and LMSD-H-LYA, the "harmonic" approaches, perform a little bit worse than the other methods: while LMSD-H-CHOL can still compete with ABB_{min} in terms of NFE and NGE, it performs worse in terms of computational time. LMSD-H-LYA performs generally worse than the other techniques; Figure 5 was already suggesting the poorer quality of the stepsizes of LMSD-H-LYA, which often need backtracking or lead to an increasing gradient norm.

To complete the picture, Table 4 reports two important quantities related to the performance profile: the proportion of problems solved by each method, and the proportion of problems solved with minimum cost, which is not always clearly visible from Figure 6. We notice that ABB_{min} and LMSD-H-LYA fail to solve one of the 31 tested problems. When ABB_{min} succeeds, it solves 32% of problems with minimum NFE and 39% of problems with minimum computational time. LMSD-PERT wins in

terms of NGE. The proportion of problems solved with minimum NFE is the same for LMSD-PERT, LMSD-LYA, and LMSD-H-CHOL.

Table 4 For each method, we report the proportion of solved problems and the proportion of problems solved at minimum cost (performance ratio equal to 1) for different performance measures. The memory parameter is $m = 5$ for all the LMSD methods.

Method	Solved (%)	PR = 1 (%)		
		NFE	NGE	Time
LMSD-PERT	1.00	0.16	0.52	0.16
LMSD-LYA	1.00	0.16	0.23	0.19
LMSD-H-CHOL	1.00	0.16	0.03	0.10
LMSD-CHOL	1.00	0.13	0.13	0.10
ABB_{min}	0.97	0.32	0.10	0.39
LMSD-H-LYA	0.97	0.06	0.06	0.06

6 Conclusions

We have reviewed the limited memory steepest descent method proposed by Fletcher [1], for both quadratic and general nonlinear unconstrained problems. In the context of strictly convex quadratic functions, we have explored pivoted QR and SVD as alternative ways to compute a basis for either the matrix $\mathbf{G}$ (Ritz values) or $\mathbf{Y}$ (harmonic Ritz values). We have also proposed to improve the harmonic Ritz values by computing the Rayleigh quotients of their corresponding harmonic Ritz vectors.

Experiments in Section 5.1 have shown that the type of decomposition has little influence on the number of iterations of LMSD. The choice between Cholesky decomposition, pivoted QR and SVD is problem dependent. These three methods may compete with the ABB_{min} gradient method.

The experiments also suggest that a larger memory parameter improves the performance of LMSD, and Ritz values seem to perform better than harmonic Ritz values. The modification of the harmonic Ritz values (Section 2.2) effectively improves the number of iterations, at the extra expense of (relatively cheap) $\mathcal{O}(m^3)$ work.

In the context of general nonlinear functions, we have given a theoretical foundation to Fletcher's idea [1] (LMSD-CHOL), by connecting the symmetrization of $\mathbf{T}$ (8) to a perturbation of $\mathbf{Y}$. We have proposed another LMSD method (LMSD-PERT) based on a different perturbation given by Schnabel [12] in the area of quasi-Newton methods. An additional modification of LMSD for general functions (LMSD-LYA) has been obtained by adding symmetry constraints to the secant condition of LMSD for quadratic functions. The solution to this problem coincides with the solution to a Lyapunov equation.

In Section 5.2, experiments on general unconstrained optimization problems have shown that, in contrast with the quadratic case, increasing the memory parameter does not necessarily improve the performance of LMSD. This may also be related to the choices made in Algorithm 2, such as the sufficient decrease condition or the criteria to keep or discard old gradients.

Given a certain memory parameter, the aforementioned LMSD methods seem to perform equally well in terms of the number of function evaluations and computational time. They all seem valid alternatives to the nonmonotone gradient method based on ABB_{min} stepsizes, with the caveat that LMSD-PERT and LMSD-LYA tend to not exploit all the stepsizes computed in a sweep, more often than LMSD-CHOL.

A Python code for the LMSD methods and the nonmonotone ABB_{min} is available at github.com/gferrandi/lmsdpy.

Acknowledgments: This work has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 812912. We would also like to thank Nataša Krejić for the inspiring discussions on gradient methods.

References

[1] Fletcher, R.: A limited memory steepest descent method. Math. Program. **135**(1), 413–436 (2012)

[2] Di Serafino, D., Ruggiero, V., Toraldo, G., Zanni, L.: On the steplength selection in gradient methods for unconstrained optimization. Appl. Math. Comput. **318**, 176–195 (2018)

[3] Zou, Q., Magoulès, F.: Delayed gradient methods for symmetric and positive definite linear systems. SIAM Rev. **64**(3), 517–553 (2022)

[4] Barzilai, J., Borwein, J.M.: Two-point step size gradient methods. IMA J. Numer. Anal. **8**(1), 141–148 (1988)

[5] Nocedal, J., Wright, S.J.: Numerical Optimization, 2nd edn. Springer, New York, NY, USA (2006)

[6] Liu, D.C., Nocedal, J.: On the limited memory BFGS method for large scale optimization. Math. Program. **45**, 503–528 (1989)

[7] Raydan, M.: The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem. SIAM J. Optim. **7**(1), 26–33 (1997)

[8] Porta, F., Prato, M., Zanni, L.: A new steplength selection for scaled gradient methods with application to image deblurring. J. Sci. Comp. **65**(3), 895–919 (2015)

[9] Franchini, G., Ruggiero, V., Zanni, L.: Ritz-like values in steplength selections for stochastic gradient methods. Soft Comput. **24**(23), 17573–17588 (2020)

[10] Fukaya, T., Kannan, R., Nakatsukasa, Y., Yamamoto, Y., Yanagisawa, Y.: Shifted Cholesky QR for computing the QR factorization of ill-conditioned matrices. SIAM J. Sci. Comput. **42**(1), 477–503 (2020)

[11] Gu, M., Eisenstat, S.C.: Efficient algorithms for computing a strong rank-revealing QR factorization. SIAM J. Sci. Comput. **17**(4), 848–869 (1996)

[12] Schnabel, R.B.: Quasi-Newton methods using multiple secant equations. Technical Report CU-CS-247-83, Department of Computer Science, University of Colorado, Boulder, USA (1983)

[13] Simoncini, V.: Computational methods for linear matrix equations. SIAM Rev. **58**(3), 377–441 (2016)

[14] Curtis, F.E., Guo, W.: Handling nonpositive curvature in a limited memory steepest descent method. IMA J. Numer. Anal. **36**(2), 717–742 (2016)

[15] Curtis, F.E., Guo, W.: R-linear convergence of limited memory steepest descent. IMA J. Numer. Anal. **38**(2), 720–742 (2018)

[16] Morgan, R.B.: Computing interior eigenvalues of large matrices. Linear Algebra Appl. **154**, 289–309 (1991)

[17] Sleijpen, G.L.G., Eshof, J.: On the use of harmonic Ritz pairs in approximating internal eigenpairs. Linear Algebra Appl. **358**(1-3), 115–137 (2003)

[18] Parlett, B.N.: The Symmetric Eigenvalue Problem. SIAM, Philadelphia, PA (1998)

[19] Beattie, C.: Harmonic Ritz and Lehmann bounds. Electron. Trans. Numer. Anal **7**, 18–39 (1998)

[20] Christof, V.: A note on harmonic Ritz values and their reciprocals. Numer. Linear Algebra Appl. **17**(1), 97–108 (2010)

[21] Yasuda, K., Hirai, K.: Upper and lower bounds on the solution of the algebraic Riccati equation. IEEE Trans. Automat. Contr. **24**(3), 483–487 (1979)

[22] Fletcher, R.: Low storage methods for unconstrained optimization. Lect. Appl. Math. (AMS) **26**, 165–179 (1990)

[23] Grippo, L., Lampariello, F., Lucidi, S.: A nonmonotone line search technique for Newton's method. SIAM J. Numer. Anal. **23**(4), 707–716 (1986)

[24] Frassoldati, G., Zanni, L., Zanghirati, G.: New adaptive stepsize selections in gradient methods. J. Ind. Manag. **4**(2), 299 (2008)

[25] Byrd, R.H., Lu, P., Nocedal, J., Zhu, C.: A limited memory algorithm for bound constrained optimization. SIAM J. Sci. Comput. **16**(5), 1190–1208 (1995)

[26] Davis, T.A., Hu, Y.: The University of Florida sparse matrix collection. ACM Trans. Math. Softw. **38**(1), 1–25 (2011)

[27] Ferrandi, G., Hochstenbach, M.E., Krejić, N.: A harmonic framework for stepsize selection in gradient methods. Comput. Opt. Appl. **85**, 75–106 (2023)

[28] Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. **91**(2), 201–213 (2002)

[29] Fowkes, J., Roberts, R., Bűrmen, A.: PyCUTEst: an open source Python package of optimization test problems. J. Open Source Softw. **7**(78), 4377 (2022)

[30] Gould, N.I.M., Orban, D., Toint, P.L.: CUTEst: a constrained and unconstrained testing environment with safe threads for mathematical optimization. Comput. Optim. Appl. **60**, 545–557 (2015)