A GLOBALLY CONVERGENT GRADIENT METHOD WITH MOMENTUM

MATTEO LAPUCCI*, GIAMPAOLO LIUZZI†, STEFANO LUCIDI‡, AND MARCO SCIANDRONE ‡

Abstract. In this work, we consider smooth unconstrained optimization problems and we deal with the class of gradient methods with momentum, i.e., descent algorithms where the search direction is defined as a linear combination of the current gradient and the preceding search direction. This family of algorithms includes nonlinear conjugate gradient methods and Polyak's heavy-ball approach, and is thus of high practical and theoretical interest in large-scale nonlinear optimization. We propose a general framework where the scalars of the linear combination defining the search direction are computed simultaneously by minimizing the approximate quadratic model in the 2 dimensional subspace. This strategy allows us to define a class of gradient methods with momentum enjoying global convergence guarantees and an optimal worst-case complexity bound in the nonconvex setting. Differently than all related works in the literature, the convergence conditions are stated in terms of the Hessian matrix of the bi-dimensional quadratic model. To the best of our knowledge, these results are novel to the literature. Moreover, extensive computational experiments show that the gradient methods with momentum here presented outperform classical conjugate gradient methods and are (at least) competitive with the state-of-art method for unconstrained optimization, i.e, L-BFGS method.

Key words. Nonconvex optimization, momentum, global convergence, complexity bound

MSC codes. 90C26, 90C30, 90C60

1. Introduction.

In this work we consider unconstrained optimization problems

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

where $f:\mathbb{R}^n\to\mathbb{R}$ is a continuously differentiable objective function. We do not assume that the function is convex. We focus on first order descent methods that exploit information from the preceding iteration to determine the search direction and the stepsize at the current one. We will hence refer to *gradient methods with momentum*, i.e., to algorithms defined by an iteration of the generic form

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k) + \beta_k(x_k - x_{k-1}), \tag{1.1}$$

where $\alpha_k > 0$ is the stepsize, and $\beta_k > 0$ is the momentum weight. Partially repeating the previous step has the effect of controlling oscillation and providing acceleration in low curvature regions. All of this can, in principle, be achieved only exploiting already available information: no additional function evaluations are required to be carried out. This feature makes the addition of momentum terms appealing in large-scale settings and, in particular, in the deep learning context [2, 27].

The best-known and most important gradient methods with momentum arguably are:

- Polyak's heavy-ball method [20, 21];
- conjugate gradient methods (see, e.g., [12]).

*Dipartimento di Ingegneria dell'Informazione, Università di Firenze, Firenze, Italy (matteo.lapucci@unifi.it).

†Dipartimento di Ingegneria Informatica, Automatica e Gestionale "Antonio Ruberti", Sapienza Università di Roma, Roma, Italy (liuzzi@diag.uniroma1.it, lucidi@diag.uniroma1.it, sciandrone@diag.uniroma1.it).

Conjugate gradient methods can be described by the updates

$$x_{k+1} = x_k + \alpha_k d_k, \qquad d_{k+1} = -\nabla f(x_{k+1}) + \beta_{k+1} d_k,$$

where α_k is computed by means of a line search, whereas β_k is obtained according to one of many rules from the literature (see, e.g., [12]). The update of conjugate gradient methods can be rewritten as

$$\begin{aligned} x_{k+1} &= x_k + \alpha_k(-\nabla f(x_k) + \beta_k d_{k-1}) \\ &= x_k + \alpha_k \left(-\nabla f(x_k) + \frac{\beta_k}{\alpha_{k-1}}(x_k - x_{k-1})\right) \\ &= x_k - \alpha_k \nabla f(x_k) + \hat{\beta}_k(x_k - x_{k-1}), \end{aligned}$$

so that it can be viewed as a gradient method with momentum according to definition (1.1).

The convergence theory of nonlinear conjugate gradient methods has been a research topic for about 30 years and now it can be considered well-established, while some complexity results have been stated only recently [6, 19]. Several proposed conjugate gradient methods can be considered sound and efficient tools for unconstrained optimization. Recently, a new class of conjugate gradient methods, known as subspace minimization conjugate gradient (SMCG) methods, have been proposed in the literature [18, 25, 28, 29]. We will discuss more in detail this class of approaches later in this work, since they are related to the framework proposed here.

On the other hand, the heavy-ball algorithm is directly described by an iteration of the form (1.1), where α_k and β_k typically are fixed positive values[24]; in principle, suitable constants should be chosen depending on properties of the objective function (e.g., using Lipschitz constant of the gradient or the constant of strong convexity) [9, 15, 21]. In practice, however, this information is often not accessible and thus α_k can be chosen by a line search while the momentum parameter β_k is usually blindly set to some (more or less) reasonable value. Convergence results for the heavy-ball method have been proven in the convex case [9, 20, 23], while the convergence of the method in the nonconvex case is still an open problem.

Thus, algorithmic issues related to the choice of the two parameters α_k and β_k and the theoretical gap related to the convergence in the nonconvex case did not allow, until now, to include the heavy-ball method within the class of sound methods for smooth unconstrained optimization. As a matter of fact, there does not exist any popular software implementation of the method. However, heavy-ball type momentum terms are consistently and effectively used within modern frameworks of stochastic optimization for neural network training [17, 26].

Both conjugate gradient and heavy-ball methods are therefore of practical interest in large-scale nonlinear optimization settings. We hence believe it is worth to focus on the study of the general class of gradient methods with momentum, in order to possibly define convergent algorithms improving the efficiency of standard nonlinear conjugate gradient methods.

To this aim, we draw inspiration from the idea presented in [29], where the search direction is computed by minimizing the approximate quadratic model in the 2 dimensional subspace spanned by the current gradient and the last search direction. According to this approach, the scalars α_k and β_k are not prefixed, but rather they are simultaneously determined by a bidimensional search. We define a general framework of gradient methods with momentum based on a simple Armijo-type line search, and we prove global convergence results under the first-order smoothness assumptions only. We also derive specific algorithms with momentum from the general framework. Furthermore, we provide complexity results, proving for the proposed gradient method with momentum the worst-case complexity bound of $\mathcal{O}(\epsilon^{-2})$, which is optimal for first order algorithms in the nonconvex setting [3]. Up to our knowledge, the presented algorithm is the first framework of gradient methods with momentum having in the nonconvex case both theoretical convergence properties and a complexity bound.

Extensive computational experiments show that the gradient methods with momentum here presented outperform conjugate gradient methods and are (at least) competitive with the state-of-art method for unconstrained optimization, i.e, L-BFGS method.

The rest of the paper is organized as follows: we describe the main idea of the work in section 2, with a focus on some important related works in subsection 2.1; in section 3 we discuss conditions for the proposed method to be well defined. Then, we discuss in section 4 the issue of guaranteeing that the employed search directions are gradient related; in particular, we consider the cases where the property descends from assumptions on either $n \times n$ or 2×2 matrices (subsection 4.1 and subsection 4.2 respectively). In section 5, we finally describe the proposed algorithmic framework for gradient methods with momentum, formalizing convergence and complexity results. In section 6, we then discuss concrete strategies to estimate the matrices that are at the core of the proposed method. In section 7, we report the results of thorough computational experiments empirically showing the potential of the proposed class of approaches. We finally give some concluding remarks in section 8.

2. The main idea. A vast class of iterative algorithms for nonlinear unconstrained optimization (namely linesearch algorithms) can be written in a general form as

$$x_{k+1} = x_k + \alpha_k d_k,$$

where $d_k \in \mathbb{R}^n$ is the search direction and $\alpha_k > 0$ is the stepsize. The most typical rules for the choice of the direction follow a general scheme, which is basically given by the following optimization subproblem:

$$\min_{d\in\mathbb{R}^n} \; \nabla f(x_k)^T d + \frac{1}{2} d^T B_k d, \tag{2.1}$$

where B_k is a suitable symmetric positive definite matrix. By properly choosing B_k, we retrieve standard methods; in particular:

- if $B_k = I$, then we obtain the steepest descent direction $-\nabla f(x_k)$;
- if $B_k = \nabla^2 f(x_k)$, then we obtain Newton's direction $-\left[\nabla^2 f(x_k)\right]^{-1} \nabla f(x_k)$;
- if B_k is a positive definite matrix obtained with suitable update rules, we obtain standard Quasi-Newton updates.

Gradient methods with momentum can be considered in the above framework by adding into (2.1) a suitable constraint on d, i.e.,

$$\begin{aligned} &\min_{d,\alpha,\beta} \; g_k^T d + \frac{1}{2} d^T B_k d \\ &\text{s.t. } \; d = -\alpha g_k + \beta s_k, \end{aligned} \tag{2.2}$$

where $g_k = \nabla f(x_k)$ and $s_k = x_k - x_{k-1}$. Then, substituting the constraint into the objective function of (2.2), the problem reduces to

$$\min_{\alpha,\beta} \phi(\alpha, \beta) \tag{2.3}$$

where

$$\phi(\alpha,\beta) = -\alpha\|g_k\|^2 + \beta g_k^T s_k + \frac{1}{2}\alpha^2 g_k^T B_k g_k + \frac{1}{2}\beta^2 s_k^T B_k s_k - \alpha\beta g_k^T B_k s_k,$$

or equivalently

$$\phi(\alpha,\beta) = \begin{bmatrix} -\|g_k\|^2 \\ g_k^T s_k \end{bmatrix}^T \begin{bmatrix} \alpha \\ \beta \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}^T \begin{bmatrix} g_k^T B_k g_k & -g_k^T B_k s_k \\ -g_k^T B_k s_k & s_k^T B_k s_k \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}. \tag{2.4}$$

We can denote the 2×2 matrix in the above equation as

$$H_k = \begin{bmatrix} H_{11} & H_{12} \\ H_{12} & H_{22} \end{bmatrix} = \begin{bmatrix} g_k^T B_k g_k & -g_k^T B_k s_k \\ -g_k^T B_k s_k & s_k^T B_k s_k \end{bmatrix} = P_k^T B_k P_k, \tag{2.5}$$

where we omit the dependence of H_{ij} on k for the sake of notation simplicity and we set $P_k = [-g_k \;\; s_k]$. Letting $u = [\alpha \;\; \beta]^T$, we can express the problem as

$$\min_u \; \frac{1}{2} u^T P_k^T B_k P_k u + g_k^T P_k u, \tag{2.6}$$

or equivalently as

$$\min_u \; \frac{1}{2} u^T H_k u + g_k^T P_k u. \tag{2.7}$$

Once a solution $u_k = [\alpha_k \;\; \beta_k]^T$ of (2.6) is determined, we define

$$d_k = -\alpha_k g_k + \beta_k s_k$$

and, provided that d_k is a descent direction, we set

$$x_{k+1} = x_k + \eta_k d_k, \tag{2.8}$$

where η_k can be determined, for example, by an Armjio-type line search.

Now consider the two-dimensional function

$$\psi_k(\alpha,\beta) = f(x_k - \alpha g_k + \beta s_k)$$

and assume that f is twice continuously differentiable. We have

$$\psi_k(0,0) = f(x_k), \qquad \nabla\psi_k(0,0) = \begin{bmatrix} -\|g_k\|^2 \\ g_k^T s_k \end{bmatrix},$$
$$\nabla^2\psi_k(0,0) = \begin{bmatrix} g_k^T \nabla^2 f(x_k) g_k & -g_k^T \nabla^2 f(x_k) s_k \\ -g_k^T \nabla^2 f(x_k) s_k & s_k^T \nabla^2 f(x_k) s_k \end{bmatrix}.$$

The quadratic Taylor polynomial for $\psi_k(\alpha,\beta)$ centered at $(0,0)$ is thus

$$f(x_k) + \begin{bmatrix} -\|g_k\|^2 \\ g_k^T s_k \end{bmatrix}^T \begin{bmatrix} \alpha \\ \beta \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}^T \begin{bmatrix} g_k^T \nabla^2 f(x_k) g_k & -g_k^T \nabla^2 f(x_k) s_k \\ -g_k^T \nabla^2 f(x_k) s_k & s_k^T \nabla^2 f(x_k) s_k \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}. \tag{2.9}$$

The matrix H_k defined by (2.5) can then be viewed as a matrix approximating $\nabla^2\psi_k(0,0)$. From this point of view, H_k could in fact be any 2×2 matrix (independent of the $n \times n$ matrix B_k) related to the approximation of the quadratic Taylor polynomial of $\psi_k(\alpha,\beta) = f(x_k - \alpha g_k + \beta s_k)$, i.e.,

$$f(x_k) + \begin{bmatrix} -\|g_k\|^2 \\ g_k^T s_k \end{bmatrix}^T \begin{bmatrix} \alpha \\ \beta \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}^T \begin{bmatrix} H_{11} & H_{12} \\ H_{12} & H_{22} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}. \tag{2.10}$$

Setting $u = [\alpha \;\; \beta]^T$, the minimization problem of (2.10) leads again to problem (2.7), being H_k a generic 2×2 matrix with suitable properties.

From a theoretical point of view, the following issues must me considered:

(a) we shall state conditions ensuring that the two-dimensional subproblem (2.6) admits solution;

(b) we shall analyze under which conditions the obtained search direction d_k, coupled with line search techniques, allows to prove global convergence properties and complexity results for the iterative scheme (2.8).

In the following, we will state conditions concerning either the $n \times n$ matrix B_k appearing in (2.6) or directly the 2×2 matrix H_k in (2.7) (with no explicit connection to B_k), in order to satisfy the above theoretical requirements.

2.1. Related Works. We now briefly analyze some important works directly related to our approach.

Subspace minimization CG Subspace minimization conjugate gradient (SMCG) methods, for which we refer the reader, for example, to the very recent paper [18], consider the two-dimensional subproblem (2.3)-(2.4) by assuming that the $n \times n$ matrix B_k satisfies the secant equation

$$B_k s_k = g_k - g_{k-1}.$$

The issue deserving attention becomes that of suitably managing the term $g_k^T B_k g_k = \rho_k$. Several strategies have been proposed in connection with this issue. Some global convergence results are stated, while complexity results are not known. However, as clearly written in [18], *"A question is naturally to be asked: can one develop an efficient SMCG method without determining the parameter ρ_k?"*. By our work, we give a positive answer to this question.

Common-directions methods The framework discussed in [14] considers (2.6)-(2.8) in a more general setting, i.e., by assuming that P_k is a general $n \times m_k$ matrix, with $m_k \geq 1$, containing at least a search direction which satisfies an angle condition. The columns of P_k are required to be linearly independent. Convergence results are stated by assuming that $\{B_k\}$ is a sequence of uniformly positive definite symmetric matrices bounded above. Then, gradient methods with momentum fits this framework provided that g_k and s_k are linearly independent. The convergence analysis relies on the properties of the $n \times n$ matrix B_k.

Our main theoretical contribution, with respect to all the works on SMCG methods and to [14], concerns the definition of conditions on the 2×2 matrix H_k sufficient to ensure that the sequence of search directions $\{d_k\}$ is *gradient-related* and hence to state convergence and complexity results for the proposed gradient methods with momentum. We remark that the theoretical focus on the low-dimensional matrix H_k is fundamental to design convergent algorithms for large-scale optimization regardless of the $n \times n$ matrix B_k. This issue yields relevant implications, deeply analyzed and discussed later, from theoretical, algorithmic and computational point of views.

3. Existence of a solution for the two dimensional problem. In this section we state conditions on the matrices B_k and H_k sufficient to ensure that the subproblem (2.7) admits solution. The condition on H_k is rather simple to state.

PROPOSITION 3.1. *Consider problem* (2.7)*. If H_k is a symmetric positive definite matrix, then problem* (2.7) *admits an optimal solution.*

Proof. The quadratic objective function is strictly convex and hence the problem admits a unique solution. ∎

Now, assume H_k is obtained, starting from an $n \times n$ matrix B_k, according to (2.5). We can state the following condition.

PROPOSITION 3.2. *Consider problem* (2.2)*. If g_k and s_k are non-zero vectors and B_k is a symmetric positive definite matrix, then problem* (2.2) *admits an optimal solution.*

Proof. Let H_k be the matrix defined in (2.5). Let us consider problem (2.7) as equivalent reformulation of problem (2.2), i.e.,

$$\min_u q(u) = \frac{1}{2} u^T P_k^T B_k P_k u + g_k^T P_k u.$$

Let us first suppose that g_k and s_k are linearly independent. By definition, $H_k = P_k^T B_k P_k$. Since B_k is positive definite, and $P_k = [-g_k \quad s_k]$ is full rank, H_k is positive definite; then, by Proposition 3.1, problem (2.7) admits solution.

On the other hand, let us assume that g_k and s_k are linearly dependent. In this case, we can write $s_k = \sigma g_k$ and it results that the Hessian matrix of the quadratic function $q(u)$ is symmetric positive semi-definite. However, we can show that system

$$P_k^T B_k P_k u = -P_k^T g_k \tag{3.1}$$

admits solution, i.e., that there exists at least a point $\bar{u}$ such that $\nabla q(\bar{u}) = 0$, and hence that $\bar{u}$ is a global minimizer of the convex function $q(u)$. Indeed, we have

$$P_k^T B_k P_k = \begin{bmatrix} g_k^T B_k g_k & -\sigma g_k^T B_k g_k \\ -\sigma g_k^T B_k g_k & \sigma^2 g_k^T B_k g_k \end{bmatrix}, \qquad -P_k^T g_k = \begin{bmatrix} \|g_k\|^2 \\ -\sigma \|g_k\|^2 \end{bmatrix}.$$

By the positive definiteness of B_k, we have that $g_k^\top B_k g_k > 0$ and we can then write that

$$1 = \operatorname{rank}\left(P_k^T B_k P_k\right) = \operatorname{rank}\left(\begin{bmatrix} P_k^T B_k P_k & -P_k^T g_k \end{bmatrix}\right).$$

This implies that system (3.1) admits a solution, which is also a solution of problem (2.7) and thus (2.2). ∎

4. Properties of the search directions. In this section we consider again subproblem (2.2) by means of the equivalent reformulations (2.6)-(2.7) and we assume that it admits solution $u_k = [\alpha_k \quad \beta_k]^T$, i.e., that at least one assumption of the preceding section either on B_k or on H_k holds. We are interested in studying further conditions on the above matrices to ensure that the obtained search directions

$$d_k = -\alpha_k g_k + \beta_k s_k$$

are gradient-related according to the following well-known definition [4, 5].

DEFINITION 4.1. *A sequence of search directions $\{d_k\}$ is gradient-related to the sequence of solutions $\{x_k\}$ if there exist $c_1 > 0$ and $c_2 > 0$ such that, for all k, we have*

$$g_k^T d_k \le -c_1 \|g_k\|^2, \qquad \|d_k\| \le c_2 \|g_k\|. \tag{4.1}$$

As already said, the above property is a requirement sufficient to guarantee that the sequence generated according to the following scheme

$$x_{k+1} = x_k + \eta_k d_k,$$

where η_k is the stepsize computed by an Armjio-type line search, is globally convergent to stationary points, Moreover, under this assumption the algorithm can be proved to have a worst case iteration and function evaluations complexity bound of $\mathcal{O}(\epsilon^{-2})$ to reach a solution $\bar{x}$ with $\|\nabla f(\bar{x})\| \leq \epsilon$. Note that this bound is actually tight for first-order methods under standard first-order smoothness assumptions.

4.1. Conditions on B_k. First we state the following result concerning the $n \times n$ matrix B_k. Note that, from here onward, we will denote by $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$ the minimum and maximum eigenvalues respectively of a symmetric matrix A.

PROPOSITION 4.2. *Let d_k be defined by $d_k = -\alpha_k g_k + \beta_k s_k$, where $[\alpha_k \quad \beta_k]^T$ is solution of* (2.6).

Let $\{B_k\} \subseteq \mathbb{R}^{n\times n}$ be the sequence of symmetric matrices defining problems (2.6) and assume that there exist scalars $0 < \eta_1 \leq \eta_2$ such that

$$\eta_1 \leq \lambda_{\min}(B_k) \leq \lambda_{\max}(B_k) \leq \eta_2 \tag{4.2}$$

holds for all k. Then the direction d_k satisfies the following conditions:

$$g_k^T d_k \leq -\frac{\eta_1}{\eta_2^2}\|g_k\|^2 \tag{4.3}$$

$$\frac{1}{\eta_2}\|g_k\| \leq \|d_k\| \leq \frac{2}{\eta_1}\|g_k\|. \tag{4.4}$$

Proof. Let us recall problem (2.6)

$$\min_u \frac{1}{2}u^T P_k^T B_k P_k u + g_k^T P_k u,$$

where $d = P_k u$, $P_k = [-g_k \quad s_k]$ and $u = [\alpha\ \beta]^T$.

From Proposition 3.2 we have that there exists at least a solution u_k of problem (2.2), satisfying the following linear system

$$P_k^T B_k P_k u_k = -P_k^T g_k. \tag{4.5}$$

Multiplying both the members in (4.5) by u_k^T, we obtain $u_k^T P_k^T B_k P_k u_k = -u_k^T P_k^T g_k$, i.e.,

$$d_k^T B_k d_k = -g_k^T d_k, \tag{4.6}$$

being $d_k = P_k u_k$. Now, consider the first row of system (4.5), i.e.,

$$-g_k^T B_k d_k = \|g_k\|^2.$$

Using (4.6) and recalling assumption (4.2), we can write

$$\|g_k\|^2 = |g_k^T B_k d_k| \leq \|g_k\|\|B_k\|\|d_k\| \leq \lambda_{\max}(B_k)\|g_k\|\|d_k\| \leq \eta_2\|g_k\|\|d_k\|,$$

and hence we obtain

$$\|d_k\| \geq \frac{1}{\eta_2}\|g_k\|. \tag{4.7}$$

Recalling (4.6), using (4.2) and (4.7), it follows

$$-g_k^T d_k \geq \lambda_{\min}(B_k)\|d_k\|^2 \geq \eta_1\|d_k\|^2 \geq \frac{\eta_1}{\eta_2^2}\|g_k\|^2. \tag{4.8}$$

Considering again (4.6) we can also write

$$\|g_k\|\|d_k\| \geq -g_k^T d_k = d_k^T B_k d_k \geq \lambda_{\min}(B_k)\|d_k\|^2 \geq \eta_1 \|d_k\|^2,$$

so that we have

$$\|d_k\| \leq \frac{1}{\eta_1}\|g_k\|. \tag{4.9}$$

Then, (4.7), (4.8) and (4.9) prove the thesis of the proposition. ∎

4.2. Conditions on H_k. In the previous section, problem (2.2) was rewritten as

$$\min_{\alpha,\beta} \frac{1}{2} \begin{bmatrix}\alpha\\ \beta\end{bmatrix}^T P_k^T B_k P_k \begin{bmatrix}\alpha\\ \beta\end{bmatrix} + g_k^T P_k \begin{bmatrix}\alpha\\ \beta\end{bmatrix},$$

having set $P_k = \begin{bmatrix}-g_k & s_k\end{bmatrix}$.
Then, Proposition 4.2 shows that condition (4.2) on the eigenvalues of the matrices B_k guarantees that the solution $[\alpha_k \ \ \beta_k]^T$ of (2.7) produces directions $d_k = -\alpha_k g_k + \beta_k s_k$ that are gradient-related. However, checking or ensuring that a sequence of $n \times n$ matrices $\{B_k\}$ have uniformly bounded eigenvalues can be extremely challenging for large-dimensional optimization problems.

Problem (2.7) can be rewritten in the following form:

$$\min_{\alpha,\beta} \frac{1}{2} \begin{bmatrix}\alpha\\ \beta\end{bmatrix}^T H_k \begin{bmatrix}\alpha\\ \beta\end{bmatrix} + \begin{bmatrix}-\|g_k\|^2\\ g_k^T s_k\end{bmatrix}^T \begin{bmatrix}\alpha\\ \beta\end{bmatrix}. \tag{4.10}$$

In this section we analyze conditions on the 2×2 matrix H_k sufficient to guarantee suitable properties to the direction d_k.

This analysis is fundamental for two reasons. The first one is that, as said before, these "direct" conditions on the low-dimensional matrix can be checked and ensured (possibly by suitable modifications) regardless of the $n \times n$ matrix B_k. The second reason is that they can be employed in connection with any 2×2 matrix defining a quadratic model — see (2.10) — to be minimized for determining the values α_k and β_k that characterize the update rule of gradient methods with momentum.

Following the latter point, we assume that H_k is any symmetric positive definite matrix. Then, (2.5) is a particular case of H_k provided that B_k is positive definite and that g_k and s_k are linearly independent.

A possible approach to define conditions to impose on the sequence of matrices $\{H_k\}$ might, in principle, draw inspiration from Proposition 4.2, concerning the sequence of matrices $\{B_k\}$. First, we therefore state the following theorem involving the sequence of matrices $\{H_k\}$.

PROPOSITION 4.3. *Let d_k be defined by $d_k = -\alpha_k g_k + \beta_k s_k$, where $[\alpha_k \ \ \beta_k]^T$ is the solution of (4.10).*

Let $\{H_k\} \subseteq \mathbb{R}^{2\times 2}$ be the sequence of symmetric matrices defining problems (4.10) and assume that there exist scalars $0 < \hat{c}_1 \leq \hat{c}_2$ such that

$$\hat{c}_1 \leq \lambda_{\min}(H_k) \leq \lambda_{\max}(H_k) \leq \hat{c}_2 \tag{4.11}$$

holds for all k.

Then the direction d_k satisfies the following conditions:

$$g_k^T d_k \leq -\frac{1}{\hat{c}_2}\|g_k\|^4 \tag{4.12}$$

$$\frac{1}{\hat{c}_2}\|g_k\|^3 \leq \|d_k\| \leq \frac{1}{\hat{c}_1}(\|g_k\| + \|s_k\|)(\|g_k\|^2 + |g_k^T s_k|). \tag{4.13}$$

Proof. We know that α_k and β_k are such that:

$$\begin{bmatrix}\alpha_k \\ \beta_k\end{bmatrix} = H_k{}^{-1}\begin{bmatrix}\|g_k\|^2 \\ -g_k^T s_k\end{bmatrix}. \tag{4.14}$$

Multiplying both sides of the above equation by $\begin{bmatrix}\|g_k\|^2 & -g_k^T s_k\end{bmatrix}$ we obtain:

$$\alpha_k\|g_k\|^2 - \beta_k g_k^T s_k = \begin{bmatrix}\|g_k\|^2 & -g_k^T s_k\end{bmatrix} H_k^{-1}\begin{bmatrix}\|g_k\|^2 \\ -g_k^T s_k\end{bmatrix}. \tag{4.15}$$

We note that $-g_k^T d_k = \alpha_k\|g_k\|^2 - \beta_k g_k^T s_k$; thus, by using (4.11) and (4.15) we have:

$$-g_k^T d_k \geq \lambda_{min}(H_k^{-1})\left(\|g_k\|^4 + (g_k^T s_k)^2\right),$$

which implies:

$$g_k^T d_k \leq -\frac{1}{\hat{c}_2}\|g_k\|^4,$$

and thus proves (4.12).

The previous inequality and the Schwarz inequality imply

$$\|d_k\| \geq \frac{1}{\hat{c}_2}\|g_k\|^3$$

which proves the first inequality of (4.13).

By using again (4.14), we can get an upper bound on the norm of d_k:

$$\begin{aligned}
\|d_k\| &= \left\|\begin{bmatrix}-g_k & s_k\end{bmatrix}\begin{bmatrix}\alpha_k \\ \beta_k\end{bmatrix}\right\| = \left\|\begin{bmatrix}-g_k & s_k\end{bmatrix} H_k{}^{-1}\begin{bmatrix}\|g_k\|^2 \\ -g_k^T s_k\end{bmatrix}\right\| \\
&\leq \left\|H_k{}^{-1}\right\| \left\|\begin{bmatrix}-g_k & s_k\end{bmatrix}\right\| \left\|\begin{bmatrix}\|g_k\|^2 \\ -g_k^T s_k\end{bmatrix}\right\| \\
&\leq \frac{1}{\hat{c}_1}(\|g_k\| + \|s_k\|)(\|g_k\|^2 + |g_k^T s_k|),
\end{aligned}$$

which proves the second relation of (4.13) and completes the proof. ∎

According to the above result, suitable descent properties of the obtained search directions hold and would in fact be sufficient, coupled with the employment of a standard Armjio-type line search, to define a globally convergent gradient method with momentum.

However, as we detail below, the obtained sequence of directions $\{d_k\}$ is not gradient-related according to Definition 4.1 and, hence, an $\mathcal{O}(\epsilon^{-2})$ complexity bound cannot be ensured.

Remark 4.4. We observe that when the matrix H_k is given by (2.5), it tends to become the null matrix as x_k approaches a stationary point. Hence, uniform boundedness conditions on the eigenvalues of H_k are in contrast with the matrices deriving from (2.2).
Even more in general, under the assumptions of Proposition 4.3, it is actually not possible to ensure that the obtained sequence of directions $\{d_k\}$ is gradient-related. Indeed, consider the cases where $g_k^T s_k = 0$. The inequality (4.13) implies:

$$|g_k^T d_k| \leq \|g_k\|\|d_k\| \leq \frac{1}{\hat{c}_1}(\|g_k\| + \|s_k\|)\|g_k\|^3,$$

and hence, by assuming that $\{s_k\}$ is bounded, for sufficiently small values of $\|g_k\|$, the direction d_k does not satisfy the first requirement of (4.1).

By the next proposition, we finally state suitable conditions on the sequence of matrices $\{H_k\}$, that also take into account the sequences of vectors involved, i.e., $\{g_k\}$ and $\{s_k\}$. These conditions are sufficient to ensure that the obtained sequence of search directions $\{d_k\}$ is gradient-related.

PROPOSITION 4.5. *Let $\{H_k\} \subseteq \mathbb{R}^{2\times 2}$ be the sequence of symmetric matrices defining problems (4.10) and assume that there exist scalars $0 < c_1 \leq c_2$ such that*

$$c_1 \leq \lambda_{\min}(D_k^{-1} H_k D_k^{-1}) \leq \lambda_{\max}(D_k^{-1} H_k D_k^{-1}) \leq c_2 \tag{4.16}$$

holds for all k, where

$$D_k = \begin{bmatrix} \|g_k\| & 0 \\ 0 & \|s_k\| \end{bmatrix}.$$

Let d_k be defined by $d_k = -\alpha_k g_k + \beta_k s_k$, where $[\alpha_k \ \ \beta_k]^T$ is solution of problem (4.10). Then, the direction d_k satisfies the following conditions:

$$g_k^T d_k \leq -\frac{1}{c_2}\|g_k\|^2 \tag{4.17}$$

$$\frac{1}{c_2}\|g_k\| \leq \|d_k\| \leq \frac{2}{c_1}\|g_k\|. \tag{4.18}$$

Proof. For simplicity, let us introduce the following matrix:

$$\tilde{H}_k = D_k^{-1} H_k D_k^{-1} = \begin{bmatrix} \frac{(H_{11})_k}{\|g_k\|^2} & \frac{(H_{12})_k}{\|s_k\|\,\|g_k\|} \\ \frac{(H_{12})_k}{\|s_k\|\,\|g_k\|} & \frac{(H_{22})_k}{\|s_k\|^2} \end{bmatrix}$$

Then, problem (4.10) can be rewritten as:

$$\min_{\alpha,\beta} \frac{1}{2} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}^T D_k \tilde{H}_k D_k \begin{bmatrix} \alpha \\ \beta \end{bmatrix} + \begin{bmatrix} -\|g_k\|^2 \\ g_k^T s_k \end{bmatrix}^T \begin{bmatrix} \alpha \\ \beta \end{bmatrix},$$

and its optimal solutions α_k and β_k are such that

$$D_k \tilde{H}_k D_k \begin{bmatrix} \alpha_k \\ \beta_k \end{bmatrix} = \begin{bmatrix} \|g_k\|^2 \\ -g_k^T s_k \end{bmatrix}, \tag{4.19}$$

from which we have:

$$D_k \begin{bmatrix} \alpha_k \\ \beta_k \end{bmatrix} = \tilde{H}_k^{-1} D_k^{-1} \begin{bmatrix} \|g_k\|^2 \\ -g_k^T s_k \end{bmatrix},$$

namely:

$$\begin{bmatrix} \alpha_k \|g_k\| \\ \beta_k \|s_k\| \end{bmatrix} = \tilde{H}_k^{-1} \begin{bmatrix} \|g_k\| \\ -\frac{g_k^T s_k}{\|s_k\|} \end{bmatrix}.$$

Multiplying both sides of the above equation by $\begin{bmatrix} \|g_k\| & -\frac{g_k^T s_k}{\|s_k\|} \end{bmatrix}$ we obtain:

$$\alpha_k \|g_k\|^2 - \beta_k g_k^T s_k = \begin{bmatrix} \|g_k\| & -\frac{g_k^T s_k}{\|s_k\|} \end{bmatrix} \tilde{H}_k^{-1} \begin{bmatrix} \|g_k\| \\ -\frac{g_k^T s_k}{\|s_k\|} \end{bmatrix}. \tag{4.20}$$

Recalling (4.16) and that $-g_k^T d_k = \alpha_k \|g_k\|^2 - \beta_k g_k^T s_k$, we have from (4.20):

$$-g_k^T d_k \geq \lambda_{min}(\tilde{H}^{-1}) \left(\|g_k\|^2 + \left(\frac{g_k^T s_k}{\|s_k\|} \right)^2 \right),$$

and thus

$$g_k^T d_k \leq -\frac{1}{c_2} \|g_k\|^2. \tag{4.21}$$

Then, Schwarz inequality implies:

$$\|g_k\| \leq c_2 \, \|d_k\|. \tag{4.22}$$

Now, multiplying both terms of equality (4.19) by $\begin{bmatrix} \alpha_k & \beta_k \end{bmatrix}$ we obtain:

$$\begin{bmatrix} \alpha_k \|g_k\| \\ \beta_k \|s_k\| \end{bmatrix}^T \tilde{H}_k \begin{bmatrix} \alpha_k \|g_k\| \\ \beta_k \|s_k\| \end{bmatrix} = -g_k^T(-\alpha g_k + \beta_k s_k) = -g_k^T d_k,$$

from which we get:

$$\begin{aligned} -g_k^T d_k &\geq \lambda_{min} \left(\tilde{H}_k \right) (\alpha_k^2 \|g_k\|^2 + \beta_k^2 \|s_k\|^2) \\ &\geq c_1 (\alpha_k^2 \|g_k\|^2 + \beta_k^2 \|s_k\|^2) \\ &\geq \frac{c_1}{2} \|d_k\|^2. \end{aligned}$$

Then, by using Schwarz inequality we get:

$$\|g_k\| \|d_k\| \geq |g_k^T d_k| \geq \frac{c_1}{2} \|d_k\|^2$$

and hence

$$\|d_k\| \leq \frac{2}{c_1} \|g_k\|. \tag{4.23}$$

Now (4.21), (4.22) and (4.23) imply (4.17) and (4.18) and, hence, the proof is complete. □

Remark 4.6. It is important to note that assumption (4.16) is not difficult to satisfy. In fact, given any sequence of symmetric matrices $\{\hat{H}_k\} \subseteq \mathbb{R}^{2\times 2}$ for which there exist scalars $0 < \tilde{c}_1 \leq \tilde{c}_2$ such that, for all k,

$$\tilde{c}_1 \leq \lambda_{\min}(\hat{H}_k) \leq \lambda_{\max}(\hat{H}_k) \leq \tilde{c}_2, \tag{4.24}$$

a sequence of matrices $\{H_k\}$ satisfying (4.16) is obtained by choosing:

$$H_k = \begin{bmatrix} \|g_k\|(\hat{H}_{11})_k \|g_k\| & \|g_k\|\hat{(}H_{12})_k \|s_k\| \\ \|s_k\|(\hat{H}_{21})_k \|g_k\| & \|s_k\|(\hat{H}_{22})_k \|s_k\| \end{bmatrix}.$$

5. Algorithmic model. In this section we propose an algorithmic framework for gradient methods with momentum which exploits the theoretical analysis carried out in the previous sections. The idea is to define a first-order algorithm with both strong theoretical guarantees in the nonconvex setting and the possibility of computationally exploiting eventual second-order information on the minimization problem. Before formally presenting the algorithm, we summarize the key steps:

(a) define a quadratic subproblem

$$\min_{\alpha,\beta} \frac{1}{2} \begin{bmatrix}\alpha \\ \beta\end{bmatrix}^T H_k \begin{bmatrix}\alpha \\ \beta\end{bmatrix} + \begin{bmatrix}-\|g_k\|^2 \\ g_k^T s_k\end{bmatrix}^T \begin{bmatrix}\alpha \\ \beta\end{bmatrix}. \tag{5.1}$$

where H_k is a 2×2 symmetric matrix;

(b) once computed a solution $[\alpha_k \ \beta_k]^T$ of (5.1), provided one exists, a check on the obtained search direction

$$d_k = -\alpha_k g_k + \beta_k s_k$$

is performed in order to ensure the gradient-related property of the sequence $\{d_k\}$;

(c) if the test is satisfied, then a standard Armjio-type line search is performed along d_k; otherwise, a suitable modification of H_k based on (4.16) is introduced and, again, steps (a) (with the modified H_k) and (b) (without the check on d_k) are performed, as well as the Armjio-type line search along the obtained d_k.

The proposed Algorithmic Model is described in Algorithm 1. Notice that the initial tentative stepsize $\eta = 1$ is optimal according to the quadratic model used to define the search direction d_k; thus, the unit step will often be a good step even for the true objective and satisfy the Armijo sufficient decrease condition. From a computational perspective, this allows to possibly save several backtracking steps and, consequently, function evaluations.

The theoretical properties of the proposed framework for gradient methods with momentum are stated in the following proposition.

PROPOSITION 5.1. *Assume that $\mathcal{L}_0 = \{x \in \mathbb{R}^n \mid f(x) \leq f(x_0)\}$ is a compact set. Then, Algorithm 1 either stops in a finite number of iterations ν producing a point x_ν which is stationary for f, i.e. $\nabla f(x_\nu) = 0$, or it produces an infinite sequence $\{x_k\}$ that admits limit points, each one being a stationary point for f. Furthermore, if the gradient ∇f is Lipschitz continuous on $\mathbb{R}^n$, we have that Algorithm 1 requires at most $\mathcal{O}(\epsilon^{-2})$ iterations, function and gradient evaluations to attain*

$$\|\nabla f(x_k)\| \leq \epsilon_k.$$

Proof. From the steps the algorithm and Proposition 4.5 we have that $\{d_k\}$ is a sequence of gradient-related directions. Since the Armijo line search is employed within Algorithm 1, the results follow from [1, Proposition 1.2.1] and by [5]. ∎

We now focus on the issue of modifying a given 2×2 matrix H_k in order to satisfy condition (4.16) as required at step 11 of Algorithm 1. Suppose that a symmetric H_k matrix has been defined at step 5, but either problem problem (5.1) does not admit solution or the test at step 8 is not satisfied, i.e., the value of *gr_dir_found* remains False. Then, step 11 must be performed and a new matrix H_k must be constructed modifying as least as possible the given matrix defined at step 5.

Algorithm 1: `Gradient Method with Momentum (GMM)`

1 Input: $x_0 \in \mathbb{R}^n$, $\gamma \in (0,1)$, $\delta \in (0,1)$, $c_1 > 0$, $c_2 > 0$.
2 Set $k \leftarrow 0$
3 **while** $\|\nabla f(x_k)\| > 0$ **do**
    `/* Compute the search direction */`
4   Set ***gr_dir_found*** $\leftarrow$ `False`
5   Define a 2×2 symmetric matrix H_k
6   **if** *problem* (5.1) *admits solution* $[\alpha_k \ \ \beta_k]^T$ **then**
7   Set $d_k \leftarrow -\alpha_k g_k + \beta_k s_k$
8   **if** $g_k^T d_k \le -c_1 \|g_k\|^2$ ***and*** $\|d_k\| \le c_2 \|g_k\|$ **then**
9   Set ***gr_dir_found*** $\leftarrow$ `True`
10   **if** ***gr_dir_found*** $=$ *False* **then**
11   Define a new 2×2 symmetric matrix H_k satisfying condition (4.16)
12   Compute α_k and β_k by solving (5.1)
13   Set $d_k \leftarrow -\alpha_k g_k + \beta_k s_k$
    `/* Perform Armijo line search along` d_k `*/`
14   Set $\eta \leftarrow 1$
15   **while** $f(x_k + \eta d_k) > f(x_k) + \gamma \eta d_k^\top \nabla f(x_k)$ **do**
16   Set $\eta \leftarrow \delta \eta$
17   Set $\eta_k \leftarrow \eta$, $x_{k+1} \leftarrow x_k + \eta_k d_k$
18   Set $k \leftarrow k + 1$

Let us denote by H_k^0 the matrix defined at step 5 and by H_k the new matrix defined at step 11. We can proceed as follows:

- Let $\hat{H}_k$ be obtained by a modified Cholesky factorization [1] applied to $D_k^{-1} H_k^0 D_k^{-1}$, where
$$D_k = \begin{bmatrix} \|g_k\| & 0 \\ 0 & \|s_k\| \end{bmatrix};$$
- Set $H_k = D_k \hat{H}_k D_k$, i.e.,
$$H_k = \begin{bmatrix} \|g_k\|(\hat{H}_{11})_k\|g_k\| & \|g_k\|(\hat{H}_{12})_k\|s_k\| \\ \|s_k\|(\hat{H}_{21})_k\|g_k\| & \|s_k\|(\hat{H}_{22})_k\|s_k\| \end{bmatrix}.$$

The boundedness of $\{H_k^0\}$ and the properties of the modified Cholesky factorization imply that (4.24) of Remark 4.6 holds, so that, according to the same remark, the matrix H_k is such that condition (4.16) is satisfied.

We conclude this section by stating a theoretical result showing that, under suitable assumptions on the objective function and a choice of the matrix H_k actually related to the Hessian $\nabla^2 f(x_k)$, steps 11-13 are never executed for k sufficiently large, that is, the test at step 8 is always satisfied.

PROPOSITION 5.2. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a twice continuously differentiable function. Suppose that*
$$H_k = \begin{bmatrix} g_k^T \nabla^2 f(x_k) g_k & -g_k^T \nabla^2 f(x_k) s_k \\ -g_k^T \nabla^2 f(x_k) s_k & s_k^T \nabla^2 f(x_k) s_k \end{bmatrix}$$
is the matrix defined at step 5, and let $\{x_k\}$ be the sequence generated by Algorithm 1. Assume that $\{x_k\}$ converges to x^*, where $\nabla f(x^*) = 0$ and the Hessian matrix

$\nabla^2 f(x^*)$ *is positive definite. Furthermore, assume that the constants of the test at step 8 are chosen in a such a way that:*

$$c_1 \leq \theta^3 \frac{\lambda_{\min}(\nabla^2 f(x^*))}{\lambda_{\max}(\nabla^2 f(x^*))^2}, \qquad c_2 \geq \frac{2}{\theta\lambda_{\min}(\nabla^2 f(x^*))}.$$

where $\theta \in (0,1)$*. Then, for* k *sufficiently large the test at step 8 is satisfied.*

Proof. By continuity there exists a neighborhood $\mathcal{B}(x^*)$ of x^* such that if $x_k \in \mathcal{B}(x^*)$ we have:

$$\theta\lambda_{\min}(\nabla^2 f(x^*)) \leq \lambda_{\min}(\nabla^2 f(x_k)), \qquad \lambda_{\max}(\nabla^2 f(x_k)) \leq \frac{1}{\theta}\lambda_{\max}(\nabla^2 f(x^*)).$$

Then the thesis follows from Proposition 4.2 by setting

$$B_k = \nabla^2 f(x_k), \quad \eta_1 = \theta\lambda_{\min}(\nabla^2 f(x_k)), \quad \eta_2 = \frac{1}{\theta}\lambda_{\max}(\nabla^2 f(x_k))$$

and the proof is complete. ∎

6. Computation of H_k. The core of the general framework lies in how a sequence of 2×2 matrices $\{H_k\}$ can be determined to ensure an efficient computational behavior of the algorithm, as well as exploiting Proposition 4.2 or Proposition 4.5 to guarantee sound theoretical properties.
Regarding the first issue, the following two strategies can be adopted:

i) define a suitable $n \times n$ matrix B_k and compute the matrix H_k by using (2.5);
ii) define a suitable 2×2 matrix H_k independent on any $n \times n$ matrix.

Concerning strategy i), in large-scale optimization problems the use and storage of the $n \times n$ matrix B_k can be computationally too expensive, if not downright prohibitive. Therefore, to take into account this issue, we propose two approaches described in subsection 6.1 and subsection 6.3. A technique related to strategy ii) is presented in subsection 6.2. Summarizing, we propose three techniques to compute the matrix H_k, although other approaches could be exploited within the general framework we have presented.

6.1. Approximating Hessian-vector products by finite differences of gradients. We draw inspiration by the Truncated Newton methods approach [11, 7], where the explicit management of the Hessian matrix $\nabla^2 f(x_k)$ is not required, but rather the Hessian-vector product $\nabla^2 f(x_k)d$ is directly handled, with $d \in R^n$.

Consider H_k defined by (2.5), i.e.,

$$H_k = \begin{bmatrix} g_k^T B_k g_k & -g_k^T B_k s_k \\ -g_k^T B_k s_k & s_k^T B_k s_k \end{bmatrix},$$

The elements of H_k can be obtained estimating the two matrix-vector products $B_k g_k$ and $B_k s_k$ by finite difference approximation, namely by the following vectors (where $\xi > 0$ is a suitably small parameter):

$$\frac{\nabla f(x_k + \xi g_k/\|g_k\|) - g_k}{\xi/\|g_k\|} \approx \nabla^2 f(x_k) g_k,$$
$$\frac{\nabla f(x_k + \xi s_k/\|s_k\|) - g_k}{\xi/\|s_k\|} \approx \nabla^2 f(x_k) s_k.$$

In this way it is possible to consistently construct H_k without the need of handling an $n \times n$ matrix. The price to pay consists in two additional evaluations of the n dimensional gradient of f, $\nabla f(x)$.

6.2. Hessian estimation in subspace by interpolation. In this subsection, an alternative way to compute the matrix H_k is proposed that avoids the need of additional n dimensional gradient evaluations.

Let us consider the two-dimensional function

$$\psi_k(\alpha, \beta) = f(x_k - \alpha g_k + \beta s_k).$$

Assuming that f is twice continuously differentiable, we have seen — see (2.9) — that the quadratic Taylor polynomial of $\psi_k(\alpha, \beta)$ centered at $(0, 0)$ is

$$f(x_k) + \begin{bmatrix} -\|g_k\|^2 \\ g_k^T s_k \end{bmatrix}^T \begin{bmatrix} \alpha \\ \beta \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}^T \begin{bmatrix} g_k^T \nabla^2 f(x_k) g_k & -g_k^T \nabla^2 f(x_k) s_k \\ -g_k^T \nabla^2 f(x_k) s_k & s_k^T \nabla^2 f(x_k) s_k \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}.$$

Then, the matrix H_k defined by (2.5) can be viewed as a matrix approximating $\nabla^2 \psi_k(0, 0)$, and, in a more general setting, H_k could be any 2×2 matrix (independent of the $n \times n$ matrix B_k). This leads to consider the following approximation of the quadratic Taylor polynomial of $\psi_k(\alpha, \beta) = f(x_k - \alpha g_k + \beta s_k)$:

$$\phi(\alpha, \beta) = f(x_k) + \begin{bmatrix} -\|g_k\|^2 \\ g_k^T s_k \end{bmatrix}^T \begin{bmatrix} \alpha \\ \beta \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}^T \begin{bmatrix} H_{11} & H_{12} \\ H_{12} & H_{22} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}.$$

The three elements defining a symmetric matrix H_k can be determined by imposing the interpolation conditions on three points (α_1, β_1), (α_2, β_2), and (α_3, β_3) different from $(0, 0)$:

$$\begin{aligned} \phi(\alpha_1, \beta_1) &= \psi_k(\alpha_1, \beta_1) = f(x_k - \alpha_1 g_k + \beta_1 s_k), \\ \phi(\alpha_2, \beta_2) &= \psi_k(\alpha_2, \beta_2) = f(x_k - \alpha_2 g_k + \beta_2 s_k), \\ \phi(\alpha_3, \beta_3) &= \psi_k(\alpha_3, \beta_3) = f(x_k - \alpha_3 g_k + \beta_3 s_k), \end{aligned}$$

i.e., by solving the linear system

$$\begin{aligned} &\frac{1}{2} \begin{pmatrix} \alpha_1^2 & 2\alpha_1\beta_1 & \beta_1^2 \\ \alpha_2^2 & 2\alpha_2\beta_2 & \beta_2^2 \\ \alpha_3^2 & 2\alpha_3\beta_3 & \beta_3^2 \end{pmatrix} \begin{pmatrix} H_{11} \\ H_{12} \\ H_{22} \end{pmatrix} = \\ &\qquad \begin{pmatrix} f(x_k - \alpha_1 g_k + \beta_1 s_k) - f(x_k) + \alpha_1 \|g_k\|^2 - \beta_1 g_k^T s_k \\ f(x_k - \alpha_2 g_k + \beta_2 s_k) - f(x_k) + \alpha_2 \|g_k\|^2 - \beta_2 g_k^T s_k \\ f(x_k - \alpha_3 g_k + \beta_3 s_k) - f(x_k) + \alpha_3 \|g_k\|^2 - \beta_3 g_k^T s_k \end{pmatrix}. \end{aligned}$$

Remark 6.1. The described strategy requires three function evaluations. However, considering the point $(\alpha_1, \beta_1) = (0, -1)$, we have

$$f(x_k - \alpha_1 g_k + \beta_1 s_k) = f(x_k - (x_k - x_{k-1}) = f(x_{k-1})$$

Then, by exploiting the information of the past iteration, i.e., the knowledge of $f(x_{k-1})$, the matrix H_k can be built by only two additional function evaluations.

Two reasonable candidate points to evaluate the function at might be, for example, $(\alpha_{k-1}, \beta_{k-1})$ and $(\alpha_{k-1}, 0)$. The interpolation system to obtain the quadratic matrix H_k becomes

$$\begin{aligned} &\frac{1}{2} \begin{pmatrix} 0 & 0 & 1 \\ \alpha_{k-1}^2 & 0 & 0 \\ \alpha_{k-1}^2 & 2\alpha_{k-1}\beta_{k-1} & \beta_{k-1}^2 \end{pmatrix} \begin{pmatrix} H_{11} \\ H_{12} \\ H_{22} \end{pmatrix} = \\ &\qquad \begin{pmatrix} f(x_{k-1}) - f(x_k) + g_k^T s_k \\ f(x_k - \alpha_{k-1} g_k) - f(x^k) + \alpha_{k-1} \|g_k\|^2 \\ f(x_k - \alpha_{k-1} g_k + \beta_{k-1} s_k) - f(x_k) + \alpha_{k-1} \|g_k\|^2 - \beta_{k-1} g_k^T s_k \end{pmatrix}. \end{aligned}$$

6.3. Hessian approximation by a diagonal matrix. The last example of computation of the matrix H_k does not require any additional function or gradient computations. The idea is to consider a diagonal matrix B_k, that is

$$B_k = \begin{bmatrix} (\mu_k)_1 & 0 & \cdots & 0 \\ 0 & (\mu_k)_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & (\mu_k)_n \end{bmatrix}. \tag{6.1}$$

The diagonal elements of the matrix B_k are computed by drawing inspiration from the approach of Barzilai-Borwein methods [22]. The matrix B_k given by (6.1) is the optimal solution of the following problem:

$$\min_B \|B_k s_k - y_k\|^2$$

where

$$y_k = g_k - g_{k-1}, \qquad s_k = x_k - x_{k-1}.$$

This implies that, for $i = 1, \ldots, n$:

$$(\mu_k)_i = (y_k)_i/(s_k)_i.$$

Finally, the elements of the 2×2 matrix H_k are give by:

$$(H_{11})_k = \sum_{i=1}^n (\mu_k)_i (g_k)_i^2, \quad (H_{12})_k = \sum_{i=1}^n (\mu_k)_i (g_k)_i (s_k)_i, \quad (H_{22})_k = \sum_{i=1}^n (\mu_k)_i (s_k)_i^2.$$

7. Computational experiments. In this section, we describe and report the results of thorough computational experiments aimed at assessing the potential of the algorithm proposed in this work. Specifically, we considered to this aim a benchmark of 163 problems from the CUTEst test-suite [10]. In particular, we included in our test set all 183 problems in the CUTEst collection that are 1) unconstrained, 2) regular, 3) with a number of variables n that is greater or equal than 1000 (except for the problems `LUKSAN12LS`, `LUKSAN13LS`, `LUKSAN14LS` having 998 variables) or that is user definable; in the latter case, we set the number of variables to 1000 or as close as possible to 1000. In Table 1 those 183 problems are reported along with their dimensions.

Then, from this set of 183 problems, we removed those problems that are unbounded below, i.e., the problems where at least one solver among the ones considered generates a sequence $\{x_k\}$ such that $f(x_k) \to -\infty$. Those problems are marked in Table 1 with a strike-through.

The code for all the experiments described in this work was written in Python 3.9, only exploiting `numpy` and `scipy` libraries. The software is available at:
`https://github.com/gliuzzi/GMM`.

Regarding Algorithm 1, it has been implemented choosing matrix H_k according to the rules discussed in section 6; we employed the safeguarding technique based on the modified Cholesky factorization, described in section 5, to ensure that Assumption (4.16) holds. As for the parameters, we set $\delta = 0.5$, $\gamma = 10^{-5}$.

The point x_{-1} is set equal to x_0, so that the momentum term is null at the first iteration. As stopping condition for our algorithm, as well as all other methods

TABLE 1
Collection of 183 unconstrained CUTEst problems (n denotes the number of variables). The problems whose name is crossed out have then been removed from the initial selection as they turned out to be unbounded from below.

Problem	n	Problem	n	Problem	n
ARGLINA	1000	EDENSCH	2000	OBSTCLBU	9604
ARGLINB	1000	EIGENALS	2550	ODC	4900
ARGLINC	1000	EIGENBLS	2550	ODNAMUR	11130
ARGTRIGLS	1000	EIGENCLS	2652	OSCIGRAD	100000
BDEXP	5000	ENGVAL1	5000	OSCIPATH	5000
BOX	10000	EXTROSNB	1000	PENALTY1	1000
BOXPOWER	10000	FLETBV3M	5000	PENALTY2	1000
~~BRATU1D~~	5001	FLETCBV2	5000	PENALTY3	1000
BROWNAL	1000	~~FLETCBV3~~	5000	PENTDI	5000
BROYDN3DLS	5000	~~FLETCHBV~~	5000	POWELLBC	1000
BROYDNBDLS	5000	FLETCHCR	1000	POWELLSG	5000
CHEBYQAD	1000	FMINSRF2	5625	POWER	10000
CHNRSNBM	1000	FMINSURF	5625	~~POWERSUM~~	1000
CLPLATEA	4970	FREUROTH	5000	PRICE3	2000
CURLY30	10000	GENHUMPS	5000	PROBPENL	1000
CVXBQP1	10000	GENROSE	1000	QING	1000
CYCLIC3LS	100002	GENROSEB	1000	QRTQUAD	5000
CYCLOOCFLS	29996	GRIDGENA	5560	QUARTC	5000
CYCLOOCTLS	29996	HILBERTA	1000	~~RAYBENDL~~	2046
DEGDIAG	100001	HILBERTB	1000	~~RAYBENDS~~	2046
DEGTRID	100001	~~INDEF~~	5000	~~S368~~	1000
ARWHEAD	5000	INDEFM	100000	SBRYBND	5000
BDQRTIC	5000	INTEQNELS	1002	SCHMVETT	5000
BROYDN7D	5000	JNLBRNG1	9604	SCOSINE	5000
BRYBND	5000	JNLBRNG2	9604	SCURLY10	10000
CHAINWOO	4000	JNLBRNGA	9604	SCURLY20	10000
CLPLATEB	4970	JNLBRNGB	9604	SCURLY30	10000
CLPLATEC	4970	KSSLS	1000	SENSORS	1000
COSINE	10000	LIARWHD	5000	SINEALI	1000
CRAGGLVY	5000	LINVERSE	1999	SINQUAD	5000
CURLY10	10000	LMINSURF	5329	SPARSINE	5000
CURLY20	10000	LUKSAN11LS	1000	SPARSQUR	10000
DEGTRID2	100001	LUKSAN12LS	998	SPIN2LS	1002
~~DIAGIQB~~	1000	LUKSAN13LS	998	SPINLS	1327
~~DIAGIQE~~	1000	LUKSAN14LS	998	SROSENBR	5000
~~DIAGIQT~~	1000	LUKSAN15LS	1000	SSBRYBND	5000
~~DIAGNQB~~	1000	LUKSAN16LS	1000	~~SSC~~	4900
~~DIAGNQE~~	1000	LUKSAN17LS	1000	SSCOSINE	5000
~~DIAGNQT~~	1000	LUKSAN21LS	1000	STRTCHDV	1000
DIAGPQB	1000	LUKSAN22LS	1000	TESTQUAD	5000
DIAGPQE	1000	MANCINO	1000	TOINTGSS	5000
DIAGPQT	1000	MCCORMCK	5000	TORSION1	5184
DIXMAANA1	3000	MODBEALE	20000	~~TORSION2~~	5184
DIXMAANB	3000	MOREBV	5000	TORSION3	5184
DIXMAANC	3000	NCB20	5010	TORSION4	5184
DIXMAAND	3000	NCB20B	5000	TORSION5	5184
DIXMAANE1	3000	~~NCVXBQP1~~	10000	TORSION6	5184
DIXMAANF	3000	~~NCVXBQP2~~	10000	TORSIONA	5184
DIXMAANG	3000	~~NCVXBQP3~~	10000	~~TORSIONB~~	5184
DIXMAANH	3000	NLMSURF	5329	TORSIONC	5184
DIXMAANI1	3000	NOBNDTOR	5184	TORSIOND	5184
DIXMAANJ	3000	NONCVXU2	5000	TORSIONE	5184
DIXMAANK	3000	NONCVXUN	5000	TORSIONF	5184
DIXMAANL	3000	NONDIA	5000	TQUARTIC	5000
DIXMAANM1	3000	NONDQUAR	5000	TRIDIA	5000
DIXMAANN	3000	NONMSQRT	4900	TRIGON1	1000
DIXMAANO	3000	NONSCOMP	5000	TRIGON2	1000
DIXMAANP	3000	OBSTCLAE	9604	VANDANMSLS	1002
DIXON3DQ	10000	OBSTCLAL	9604	VARDIM	1000
DQDRTIC	5000	OBSTCLBL	9604	VAREIGVL	5000
DQRTIC	5000	OBSTCLBM	9604	WOODS	4000

considered in the experimentation, we required $\|\nabla f(x_k)\|_\infty \leq 10^{-3}$; we also set a maximum number of iterations to 5000, so that we consider a failure each run that ends by hitting this threshold.

As a baseline for comparisons, we took into account classical nonlinear conjugate gradient methods [13] and the L-BFGS algorithm [16], which can arguably be considered the state-of-the-art for the solution of smooth unconstrained nonlinear optimization problems. For both algorithms, we considered the very efficient implementations available through the `scipy` library.

The results of the experiments are shown in the form of performance profiles [8]. As performance metrics, we considered both the runtime and the number of iterations.

In the following, we report results for three versions of our algorithm, depending on the technique adopted for the computation of H_k, as detailed in section 6. More in detail, we denote by GMM_1, GMM_2, GMM_3 the algorithm where H_k is computed as described in subsection 6.1, 6.2 and 6.3, respectively.

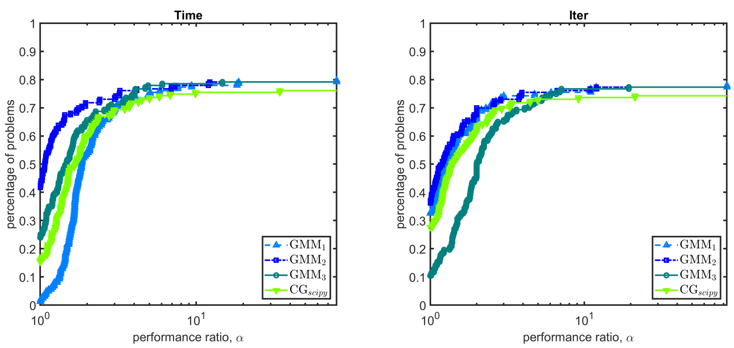

FIG. 1. *Performance profiles on all 163 problems.*

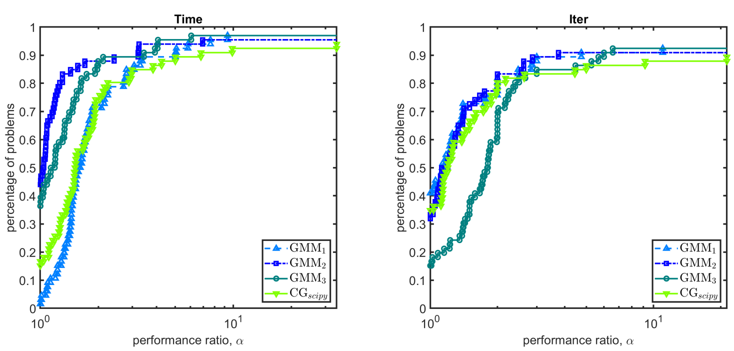

FIG. 2. *Performance profiles on the subset of problems where all the solvers find the same best function value.*

In Figure 1, we report the performance profiles in terms of time and number of iterations for the comparison among GMM_1, GMM_2, GMM_3 and CG_{scipy} on the whole collection of 163 problems. As we can see, the best performing solver is GMM_2 both in terms of time and iterations. We can also say that GMM_3 is the second best solver in terms of time even though it is the worst one in terms of iterations. Both solvers GMM_2 and GMM_3 are better than CG_{scipy} in terms of time. In doing these considerations, we must also take into account that it might happen that solvers converge toward different points. Indeed, in Table 2 we report the number of wins, i.e., the number of problems where each solver finds the best function value.

TABLE 2
Number of wins, i.e., the number of problems where each solver finds the best function value, on all 163 test problems. GMM variants and CG_{scipy} solvers are considered.

Solver	# wins
GMM_1	71
GMM_2	58
GMM_3	26
CG_{scipy}	16

Then, in Figure 2, we report the performance profiles considering the subset of problems where each solver finds the same best function value, i.e.

$$f_i^* - f_L < 10^{-3} \quad \forall\, i \in \mathcal{S},$$

where $\mathcal{S}$ is the set of solvers. From Figure 2, we note that the performance of GMM_1 becomes closer to that of the conjugate gradient method in terms of computational time, whereas GMM_2 and GMM_3 continue to be clearly superior.

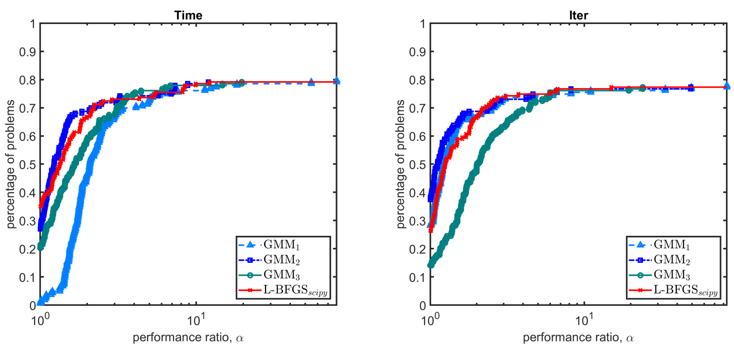

FIG. 3. *Performance profiles for the comparison between GMM_1, GMM_2, GMM_3 and L-BFGS on all the 163 problems.*

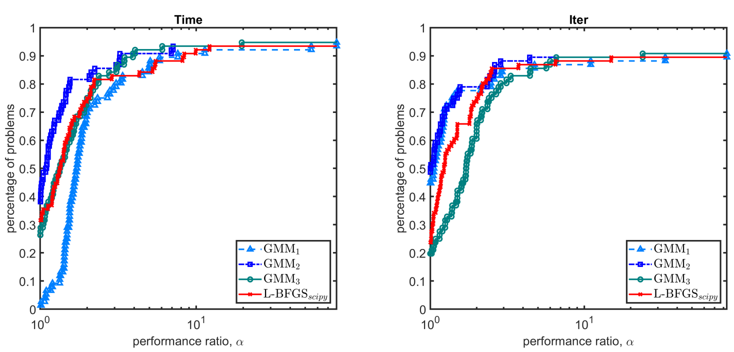

FIG. 4. *Performance profiles for the comparison between GMM_1, GMM_2, GMM_3 and L-BFGS on the subset of problems where all the solvers find the same best function value.*

Now, we consider the comparison between our methods and the state-of-the-art algorithm for unconstrained optimization, namely, L-BFGS [16]. In particular, performance profiles of solvers GMM_1, GMM_2, GMM_3 and L-BFGS on the entire collection of 163 problems are reported in Figure 3. As we can see, our method GMM_2 is quite competitive with L-BFGS in terms of computational time. Again, we have to consider the fact that the different solvers can converge toward points with different function values. In fact, as reported in Table 3, GMM_1 and GMM_2 find a better function value more frequently than L-BFGS. In Figure 4, we thus consider the

TABLE 3
Number of wins, i.e., the number of problems where each solver finds the best function value, on all 163 test problems. GMM variants and L-BFGS solvers are considered.

Solver	# wins
GMM_1	55
GMM_2	40
GMM_3	17
$\mathrm{L\text{-}BFGS}_{scipy}$	33

performance profiles obtained for the four solvers on the subset of problems where all of them find the same function value. As we can see, GMM_2 gained some efficiency and it now slightly dominates L-BFGS in terms of computational time.

Then, for major clarity, we report in Figures 5 and 6 the performance profiles for the comparison between our bet method GMM_2 and L-BFGS.

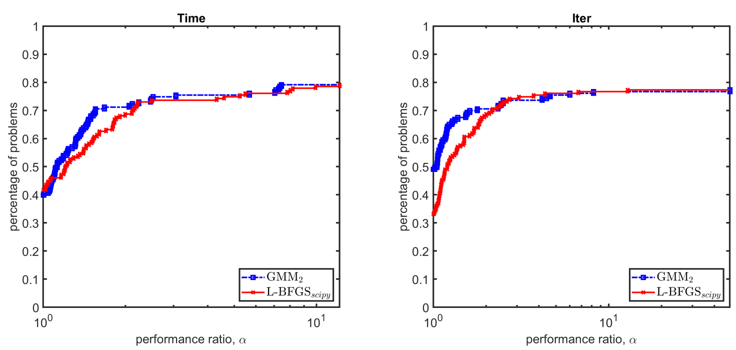

FIG. 5. *Performance profiles for the comparison between GMM_2 and L-BFGS on all the 163 problems.*

From Figures 5 and 6, we can say that GMM_2 is quite competitive if not better than L-BFGS both in terms of computational time and number of iterations. The advantages of using GMM_2 are more evident when the two solvers are compared on the subset of problems where they both find the same function value.

8. Conclusions. In this work, we introduced a general framework of gradient methods with momentum for nonconvex optimization. For the proposed class of algorithms, we proved global convergence and optimal worst-case complexity bounds. The assumptions required to obtain the theoretical results are lighter to check, from a computational perspective, than those required in related works from the literature. This result allowed to devise particularly efficient ways of implementing the proposed method. Thorough computational results we showed that the novel algorithm outperforms standard conjugate gradient methods and is competitive with L-BFGS, the state-of-the-art for nonlinear optimization problems.

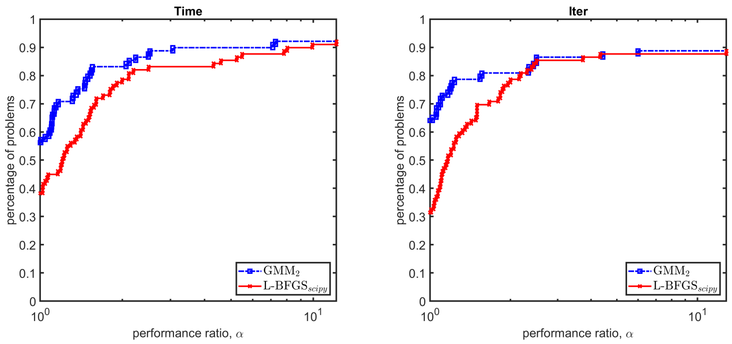

FIG. 6. *Performance profiles for the comparison between GMM_2 and L-BFGS on the subset of problems where all the solvers find the same best function value.*

Future studies might concern improvements of the proposed framework and suitable modifications to make it exploitable in different settings.

Acknowledgments. The authors are very grateful to Prof. Luigi Grippo for the useful discussions at the early stages of this work.

REFERENCES

[1] D. P. Bertsekas, *Nonlinear Programming, third edition*, Athena Scientific, Belmont, 2016.
[2] L. Bottou, F. E. Curtis, and J. Nocedal, *Optimization methods for large-scale machine learning*, SIAM Review, 60 (2018), pp. 223–311.
[3] Y. Carmon, J. C. Duchi, O. Hinder, and A. Sidford, *Lower bounds for finding stationary points I*, Mathematical Programming, 184 (2020), pp. 71–120.
[4] C. Cartis, N. I. M. Gould, and Ph. L. Toint, *Evaluation Complexity of Algorithms for Nonconvex Optimization: Theory, Computation and Perspectives*, SIAM, 2022.
[5] C. Cartis, P. R. Sampaio, and Ph. L. Toint, *Worst-case evaluation complexity of non-monotone gradient-related algorithms for unconstrained optimization*, Optimization, 64 (2015), pp. 1349–1361.
[6] R. Chan-Renous-Legoubin and C. W. Royer, *A nonlinear conjugate gradient method with complexity guarantees and its application to nonconvex regression*, EURO Journal on Computational Optimization, 10 (2022), p. 100044.
[7] R. S. Dembo, S. C. Eisenstat, and T. Steihaug, *Inexact newton methods*, SIAM Journal on Numerical Analysis, 19 (1982), pp. 400–408.
[8] E. D. Dolan and J. J. Moré, *Benchmarking optimization software with performance profiles*, Mathematical Programming, 91 (2002), pp. 201–213.
[9] E. Ghadimi, H. R. Feyzmahdavian, and M. Johansson, *Global convergence of the heavy-ball method for convex optimization*, in 2015 European Control Conference (ECC), IEEE, 2015, pp. 310–315.
[10] N. I. M. Gould, D. Orban, and Ph. L. Toint, *The constrained and unconstrained testing environment with safe threads (CUTEst) for optimization software*, 2019. `https://github.com/ralna/CUTEst`.
[11] L. Grippo, F. Lampariello, and S. Lucidi, *A truncated newton method with nonmonotone line search for unconstrained optimization*, Journal of Optimization Theory and Applications, 60 (1989), pp. 401–419.
[12] L. Grippo and M. Sciandrone, *Introduction to Methods for Nonlinear Optimization*, Springer, 2023.
[13] W. W. Hager and H. Zhang, *A survey of nonlinear conjugate gradient methods*, Pacific Journal of Optimization, 2 (2006), pp. 35–58.
[14] C.-p. Lee, P.-W. Wang, and C.-J. Lin, *Limited-memory common-directions method for large-scale optimization: convergence, parallelization, and distributed optimization*, Mathematical Programming Computation, 14 (2022), pp. 543–591.
[15] L. Lessard, B. Recht, and A. Packard, *Analysis and design of optimization algorithms via*

integral quadratic constraints, SIAM Journal on Optimization, 26 (2016), pp. 57–95.

[16] D. C. Liu and J. Nocedal, *On the limited memory BFGS method for large scale optimization*, Mathematical Programming, 45 (1989), pp. 503–528.

[17] Y. Liu, Y. Gao, and W. Yin, *An improved analysis of stochastic gradient descent with momentum*, Advances in Neural Information Processing Systems, 33 (2020), pp. 18261–18271.

[18] Z. Liu, Y. Ni, H. Liu, and W. Sun, *A new subspace minimization conjugate gradient method for unconstrained minimization*, Journal of Optimization Theory and Applications, 200 (2024), pp. 820–851.

[19] A. Neumaier, M. Kimiaei, and B. Azmi, *Globally linearly convergent nonlinear conjugate gradients without wolfe line search*, Numerical Algorithms, (2024), pp. 1–27.

[20] B. T. Polyak, *Some methods of speeding up the convergence of iteration methods*, USSR Computational Mathematics and Mathematical Physics, 4 (1964), pp. 1–17.

[21] B. T. Polyak, *Introduction to optimization. Optimization software*, Inc., Publications Division, New York, 1 (1987), p. 32.

[22] M. Raydan, *The barzilai and borwein gradient method for the large scale unconstrained minimization problem*, SIAM Journal on Optimization, 7 (1997), pp. 26–33.

[23] S. Saab Jr, S. Phoha, M. Zhu, and A. Ray, *An adaptive polyak heavy-ball method*, Machine Learning, 111 (2022), pp. 3245–3277.

[24] S. Sra, S. Nowozin, and S. J. Wright, *Optimization for machine learning*, Mit Press, 2012.

[25] W. Sun, H. Liu, and Z. Liu, *A class of accelerated subspace minimization conjugate gradient methods*, Journal of Optimization Theory and Applications, 190 (2021), pp. 811–840.

[26] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, *On the importance of initialization and momentum in deep learning*, in International Conference on Machine Learning, PMLR, 2013, pp. 1139–1147.

[27] S. J. Wright and B. Recht, *Optimization for data analysis*, Cambridge University Press, 2022.

[28] Y. Yang, Y. Chen, and Y. Lu, *A subspace conjugate gradient algorithm for large-scale unconstrained optimization*, Numerical Algorithms, 76 (2017), pp. 813–828.

[29] Y.-X. Yuan and J. Stoer, *A subspace study on conjugate gradient algorithms*, ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik, 75 (1995), pp. 69–77.