Adaptive Quasi-Newton and Anderson Acceleration Framework with Explicit Global (Accelerated) Convergence Rates

Damien Scieur D.SCIEUR@SAMSUNG.COM
Samsung SAIL and Mila, Montreal

Abstract

Despite the impressive numerical performance of quasi-Newton and Anderson/nonlinear-acceleration methods, their global convergence rates have remained elusive for over 50 years. This paper addresses this long-standing question by introducing a framework that derives novel and adaptive quasi-Newton or nonlinear/Anderson acceleration schemes. Under mild assumptions, the proposed iterative methods exhibit explicit, non-asymptotic convergence rates that blend those of gradient descent and Cubic Regularized Newton's method. The proposed approach also includes an accelerated version for convex functions. Notably, these rates are achieved adaptively, without prior knowledge of the function's smoothness parameter. The framework presented in this paper is generic, and algorithms such as Newton's method with random subspaces, finite difference, or lazy Hessian can be seen as special cases of this paper's algorithm. Numerical experiments demonstrate the efficiency of the proposed framework, even compared to the L-BFGS algorithm with Wolfe line search.

1. Introduction

Consider the problem of finding the minimizer $x^\star$ of the unconstrained minimization problem

$$f(x^\star) = f^\star = \min_{x\in\mathbb{R}^d} f(x),$$

where d is the problem's dimension, and the function f has a Lipschitz continuous Hessian.

Assumption 1. *The function $f(x)$ has a Lipschitz continuous Hessian with a constant L,*

$$\forall \; y,\, z \in \mathbb{R}^d, \quad \|\nabla^2 f(z) - \nabla^2 f(y)\| \leq L\|z-y\|. \tag{1}$$

In this paper, $\|.\|$ stands for the maximal singular value of a matrix and for the ℓ_2 norm for a vector. Many twice-differentiable problems like logistic or least-squares regression satisfy Assumption 1.

The Lipschitz continuity of the Hessian is crucial when analyzing second-order algorithms, as it extends the concept of smoothness to the second order. The groundbreaking work by Nesterov et al. [52] has sparked a renewed interest in second-order methods, revealing the remarkable convergence rate improvement of Newton's method on problems satisfying Assumption 1 when augmented with cubic regularization. For instance, if the problem is also convex, accelerated gradient descent typically achieves $O(\frac{1}{t^2})$, while accelerated second-order methods achieve $O(\frac{1}{t^3})$. Recent advancements have further pushed the boundaries, achieving even faster convergence rates of up to $\mathcal{O}(\frac{1}{t^{7/2}})$ through the utilization of hybrid methods [49, 15] or direct acceleration of second-order methods [50, 30, 46].

Unfortunately, second-order methods are not scalable, particularly in high-dimensional problems common in machine learning. The limitation is that exact second-order methods require solving a linear system that involves the Hessian of the function f. This motivated alternative approaches that balance the efficiency of second-order methods and the scalability of first-order methods, such as Quasi-Newton methods or Nonlinear acceleration methods (which are equivalent to quasi-Newton methods, see [26]).

Quasi-Newton (qN) methods efficiently minimize differentiable functions by iteratively updating an approximate Hessian matrix using previous gradient information, effectively balancing computational efficiency and optimization accuracy. This approach makes them highly suitable for large-scale optimization problems across diverse fields, providing an appealing combination of speed and effectiveness in finding optimal solutions. For instance, ℓ-BFGS is a widely used and effective optimization method for unconstrained functions (for instance, `fminunc` from Matlab), and is often considered as a state-of-the-art method in many applications [1].

1.1. Contributions

Despite the impressive numerical performance of quasi-Newton methods and nonlinear acceleration schemes, there are currently no satisfying global explicit convergence rates. In fact, global convergence cannot be guaranteed without using either exact or Wolfe-line search techniques. This raises the following long-standing question **that has remained unanswered for over 50 years**:

What are the non-asymptotic global convergence rates of quasi-Newton and Anderson/nonlinear acceleration methods?

This paper provides a partial answer by introducing generic updates that are novel quasi-Newton methods or regularized nonlinear acceleration schemes with cubic regularization. In particular, to the author's knowledge, the method presented in this paper is the first to satisfy those desiderata:

1. The assumptions for the theoretical analysis are simple and verifiable (sec 3.1),
2. The algorithm is suitable for large-scale problems, as for a fixed memory N, its per-iteration cost is linear in the dimension,
3. The algorithm exhibits **explicit, global and non-asymptotic convergence rates** that interpolate the one of first order and second order methods (more details in appendix D):
   - Non-convex problems (Theorem 2): $\min_{i\leq t} \|\nabla f(x_i)\| \leq O(t^{-\frac{2}{3}} + t^{-\frac{1}{3}})$,
   - (Star-)convex problems (Theorems 3 and 4): $f(x_t) - f^\star \leq O(t^{-2} + t^{-1})$,
   - Accelerated rate on convex problems (Theorem 5): $f(x_t) - f^\star \leq O(t^{-3} + t^{-2})$,

4. The algorithm **is adaptive to the problem's constants** (algorithms 4 and 7): both accelerated and classical methods require only an initial estimate of the Lipchitz constant,

5. Is competitive with l-BFGS (Section 6).

Currently, the l-BFGS algorithm is probably at its peak in terms of engineering achievement, given its robust and highly efficient performance. The challenge is that further numerical improvements or finding fast rates without arming the numerical convergence may be increasingly hard or impossible. Hence, to achieve the previous points, this paper explores a new paradigm by *rethinking from scratch the framework underlying qN methods*. The goal is to ensure a theoretical convergence rate while keeping the incredible numerical performance of current qN schemes.

Current limitations Some previous work already tempted to find rates for qN methods, but often violates at least of the previous point: **1)** the analysis requires non-verifiable assumptions, **2)** the algorithm is not suitable for large-scale problems as the per-iteration cost is at least $O(d^2)$, **3)** the rates are locals or do not interpolate between first and second order rates, **4)** the algorithm requires unknown, critical hyper-parameters. A more in-depth analysis of previous work can be found in appendix C.

Violates 1: For instance, the ARC method [16, 17] or proximal qN methods [**carti**, 82, 59] show accelerated rates for quasi-Newton under similar assumptions as this paper. Still, the authors state that the convergence rate is derived under a non-verifiable assumption, and their rates do not rely on or exploit the accuracy of second-order approximations.

Violates 2: Recent research on quasi-Newton updates has unveiled explicit and non-asymptotic rates of convergence [56, 58, 57, 47, 48]. Nonetheless, these analyses suffer from several significant drawbacks, such as assuming an infinite memory size and/or requiring access to the Hessian matrix. In addition, the rates are only valid locally.

Violates 3: By using online algorithms and the Monteiro-Svaiter acceleration technique, [44] achieves accelerated rates $O(\min\{\frac{1}{t^2},\frac{1}{t^{2.5}}\})$ for qN methods, but despite being full-memory algorithms, they do not match the $O(1/t^3)$ accelerated rate of second order method, and also use a full $d\times d$ matrix, which does not scale well in high dimension.

Violates 4: Kamzolov et al. [45] introduced an adaptive regularization technique combined with cubic regularization, but the method relies on knowing L in Assumption 1.

Note that in most of the previous work, **a (wolfe) line search algorithm** (often in addition with other techniques, like secant equation filtering or re-scaling) is required to ensure global convergence. Without such line search, the performance of qN method is usually poor or divergent, even on a simple quadratic case in two dimensions [55].

2. Rethinking From Scratch Quasi-Newton Methods

This section presents the sketch of the ideas introduced in this paper. The starting point is the cubic upper bound on the objective function f, and the quadratic upper bound on the gradient variation, derived using Assumption 1 [52],

$$\|\nabla f(y)-\nabla f(x)-\nabla^2 f(x)(y-x)\| \leq \tfrac{L}{2}\|y-x\|^2, \tag{2}$$

$$\left|f(y)-f(x)-\nabla f(x)(y-x)-\tfrac{1}{2}(y-x)^T\nabla^2 f(x)(y-x)\right| \leq \tfrac{L}{6}\|y-x\|^3, \tag{3}$$

which holds for all $x, y \in \mathbb{R}^d$ [52]. Minimizing (3) over y leads to the cubic regularization of Newton's method [52].

The main steps to derive this paper's algorithms are as follows: **1)** The minimization will be contained to a subspace of dimension $N \leq d$, reducing the per-iteration computation cost. **2)** As for quasi-Newton methods, the Hessian (in the subspace) will be approximated using differences of gradients. **3)** From the previous points, an upper bound for the objective function and the gradient norm will be constructed, leading to a type-I and type-II method. **4)** To ensure convergence, the direction of the subspace will be chosen such that the direction spans the gradient (deterministic), or, spans a portion of the gradient in expectation (random subspace).

Due to space limitation, all the details are presented in appendix A. In the end, at each iteration t, the algorithm updates a matrix of directions D_t and a matrix of gradient differences G_t, defined as

$$D_t = \left[\frac{y_1^{(t)} - z_1^{(t)}}{\|y_1^{(t)} - z_1^{(t)}\|_2}, \ldots, \frac{y_N^{(t)} - z_N^{(t)}}{\|y_N^{(t)} - z_N^{(t)}\|_2}\right], \quad G_t = \left[\ldots, \frac{\nabla f(y_i^{(t)}) - \nabla f(z_i^{(t)})}{\|y_i^{(t)} - z_i^{(t)}\|_2}, \ldots\right], \tag{4}$$

where $y_i^{(t)}$ and $z_i^{(t)}$ have been chosen carefully, such that D_t is orthogonal (see e.g. algorithm 1). Then, it computes the *error vector* ε_t defined as

$$\varepsilon_t \stackrel{\text{def}}{=} [e_1^{(t)}, \ldots, e_N^{(t)}], \quad \text{and} \quad e_i^{(t)} \stackrel{\text{def}}{=} \|y_i^{(t)} - z_i^{(t)}\| + 2\|z_i^{(t)} - x\|. \tag{5}$$

This vector estimates the approximation error of estimating the product $\nabla f(x_t)D_t$ by G_t. Then, the algorithm constructs the matrix H_t

$$H_t \stackrel{\text{def}}{=} \tfrac{G_t^T D_t + D_t^T G_t + \mathrm{I}L\|D_t\|\|\varepsilon_t\|}{2},$$

which can be viewed as an approximation with finite differences of the Hessian $\nabla^2 f(x_t)$ in the subspace spanned by the column of D_t. Finally, the next iterate x_{t+1} is obtained as

$$x_{t+1} = x_t + D_t\alpha_t, \tag{6}$$

where α minimizes the following upper bound, over $\alpha \in \mathbb{R}^N$, (see algorithms 3 and 4)

$$f(x_{t+1}) \leq f(x_t) + \nabla f(x_t)^T D_t\alpha + \tfrac{\alpha^T H_t \alpha}{2} + \tfrac{L\|D_t\alpha\|^3}{6}. \tag{Type-I bound}$$

3. Rates of Convergences for the Type-I method

3.1. Assumptions

This section lists the important assumptions on the function f. Some subsequent results require an upper bound on the radius of the sub-level set of f at $f(x_0)$.

Assumption 2. *The radius of the sub-level set* $\{x : f(x) \leq f(x_0)\}$ *is bounded by* $\mathrm{R} < \infty$.

To ensure the convergence toward $f(x^\star)$, some results require f to be star-convex or convex.

Assumption 3. *The function* f *is star convex if, for all* $x \in \mathbb{R}^d$ *and* $\forall \tau \in [0, 1]$,

$$f((1-\tau)x + \tau x^\star) \leq (1-\tau)f(x) + \tau f(x^\star).$$

Assumption 4. *The function* f *is convex if, for all* $y, z \in \mathbb{R}^d$, $f(y) \geq f(z) + \nabla f(z)(y - z)$.

3.2. Rates of Convergence

When f satisfies Assumption 1, algorithm 3 ensures a minimal function decrease at each step.

Theorem 1. *Let f satisfy Assumption 1. Then, at each iteration $t \geq 0$, algorithm 3 achieves*

$$f(x_{t+1}) \leq f(x_t) - \tfrac{M_{t+1}}{12}\|x_{t+1}-x_t\|^3, \quad M_{t+1} < \max\left\{2L \; ; \; \tfrac{M_0}{2^t}\right\}. \tag{7}$$

Under some mild assumptions, algorithm 3 converges to a critical point for non-convex functions, and converges to an optimum when the function is star-convex.

Theorem 2. *Let f satisfy Assumption 1, and assume that f is bounded below by f^*. Let Requirements 1b to 3 hold, and $M_t \geq M_{\min}$. Then, algorithm 3 starting at x_0 with M_0 achieves*

$$\min_{i=1,\dots,t} \|\nabla f(x_i)\| \leq \max\left\{ \frac{3L}{t^{2/3}} \left(12\frac{f(x_0)-f^\star}{M_{\min}}\right)^{2/3} \; ; \; \left(\frac{C_1}{t^{1/3}}\right)\left(12\frac{f(x_0)-f^\star}{M_{\min}}\right)^{1/3} \right\},$$

where $C_1 = \delta L\left(\frac{\kappa+2\kappa^2}{2}\right) + \max_{i\in[0,t]} \|(I-P_i)\nabla^2 f(x_i) P_i\|$.

Theorem 3. *Assume f satisfy Assumptions 1 to 3. Let Requirements 1b to 3 hold. Then, algorithm 3 starting at x_0 with M_0 achieves, for $t \geq 1$,*

$$f(x_t) - f^\star \leq 6\frac{f(x_0)-f^\star}{t(t+1)(t+2)} + \frac{1}{(t+1)(t+2)}\frac{L(3R)^3}{2} + \frac{1}{t+2}\frac{C_2(3R)^2}{4},$$

where $C_2 \overset{def}{=} \delta L \frac{\kappa+2\kappa^2}{2} + \max_{i\in[0,t]} \|\nabla^2 f(x_i) - P_i \nabla^2 f(x_i) P_i\|$.

Finally, the next theorem shows that when algorithm 3 random directions (that satisfies Requirement 1a), then $f(x_t)$ also converges in expectation to $f(x^\star)$ when f is convex.

Theorem 4. *Assume f satisfy Assumptions 1, 2 and 4. Let Requirements 1a, 2 and 3 hold. Then, in expectation over the matrices D_i, algorithm 3 starting at x_0 with M_0 achieves, for $t \geq 1$,*

$$\mathbb{E}_{D_t}[f(x_t) - f^\star] \leq \frac{1}{1+\frac{1}{4}\left[\frac{N}{d}t\right]^3}(f(x_0)-f^\star) + \frac{1}{\left[\frac{N}{d}t\right]^2}\frac{L(3R)^3}{2} + \frac{1}{\left[\frac{N}{d}t\right]}\frac{C_3(3R)^2}{2},$$

where $C_3 \overset{def}{=} \delta L \frac{\kappa+2\kappa^2}{2} + \frac{(d-N)}{d}\max_{i\in[0,t]} \|\nabla^2 f(x_i)\|$.

For space limitation reasons, the accelerated algorithm 3 is presented in section appendix B, see algorithms 6 and 7. Indeed, while theoretically more interesting, the algorithm performs poorly numerically - probably because it trades off some adaptivity for better worst-case convergence rates.

Theorem 5. *Assume f satisfy Assumptions 1, 2 and 4. Let Requirements 1b to 3 hold. Then, the accelerated algorithm 7 starting at x_0 with M_0 achieves, for $t \geq 1$,*

$$f(x_t) - f^\star \leq C_4 \tfrac{(3R)^2}{(t+3)^2} + 9\max\{M_0 \; ; \; 2L\}\left(\tfrac{3R}{t+3}\right)^3 + \frac{\frac{\tilde\lambda^{(1)}R^2}{2} + \frac{\tilde\lambda^{(2)}R^3}{6}}{(t+1)^3}.$$

where $\tilde\lambda^{(1)} = 0.5\cdot\delta\left(L\kappa + M_1\kappa^2\right) + \|\nabla^2 f(x_0) - P_0\nabla^2 f(x_0) P_0\|, \qquad \tilde\lambda^{(2)} = M_1 + L,$

$$C_4 = 30\cdot\kappa_D\left(\delta\max\{4L, M_0\} + \max_{i=0\dots t}\|(I-P_i)\nabla f(x_i)P_i)\|\right)$$

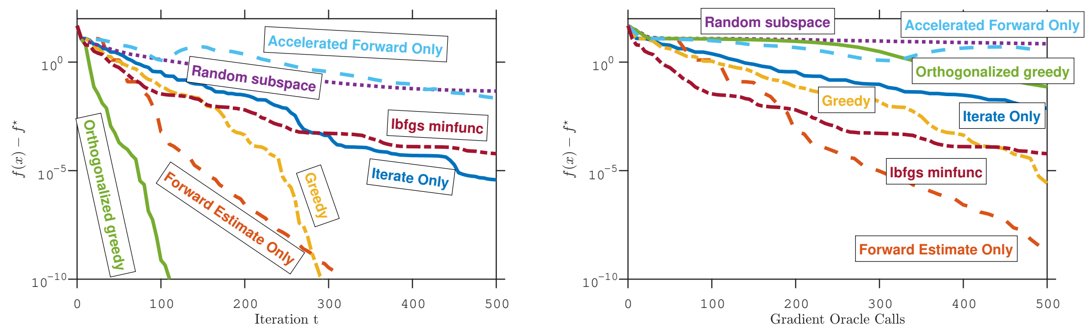

Figure 1: Comparison between the type-1 methods proposed in this paper and the optimized implementation of ℓ-BFGS from `minFunc` [60] with default parameters, except for the memory size. All methods use a memory size of $N = 25$.

The rates in Theorems 2 to 5 combine the ones of cubic regularized Newton's method and gradient descent (or coordinate descent, as in Theorem 4) for functions with Lipschitz-continuous Hessian. As C_1, C_2, C_3, and C_4 decrease, the rates approach those of cubic Newton (See appendix D).

4. Numerical Experiments

This section compares the methods generated by this paper's framework to the fine-tuned ℓ-BFGS algorithm from `minFunc` [60], see creffig:test. More experiments are conducted in appendix G. The tested methods are the Type-I iterative algorithms (algorithm 3 with the techniques from appendix A.4). The step size of the forward estimation was set to $h = 10^{-9}$, and the condition number κ_{D_t} is maintained below $\kappa = 10^9$ with the iterates only and Greedy techniques. The accelerated algorithm 7 is used only with the *Forward Estimates Only* technique. The compared methods are evaluated on a logistic regression problem on the Madelon UCI dataset [37].

Regarding the number of iterations, the greedy orthogonalized version outperforms the others due to the orthogonality of directions (resulting in a condition number of one) and the meaningfulness of previous gradients/iterates. However, in terms of gradient oracle calls, the recommended method, *orthogonal forward iterates only*, achieves the best performance by striking a balance between the cost per iteration (only two gradients per iteration) and efficiency (small and orthogonal directions, reducing theoretical constants). Surprisingly, the accelerated method's performance is suboptimal, possibly because it tightens the theoretical analysis, diminishing its inherent adaptivity.

5. Conclusion, Limitation, and Future work

This paper introduces a generic framework for developing novel quasi-Newton and Anderson/Nonlinear acceleration schemes, offering a global convergence rate in various scenarios, including accelerated convergence on convex functions, with minimal assumptions and design requirements.

The current approach requires an additional gradient step for the *forward estimate*, as discussed in Section A.4. However, this forward estimate is crucial in enabling the algorithm's adaptivity.

In future research, although unsuitable for large-scale problems, the method presented in this paper can achieve super-linear convergence rates, as with infinite memory, they would be as fast as cubic Newton methods. Utilizing the average-case analysis framework from existing literature, such as [54, 65, 24, 19, 53], could also improve the constants in Theorems 2 and 3 to match those in Theorem 4. Furthermore, exploring convergence rates for type-2 methods, which are believed to be effective for variational inequalities, is a worthwhile direction.

Algorithm 1 "Orthogonal Forward Estimate Only" Update

Require: First-order oracle f, step-size h, matrices D_{t-1}, G_{t-1}, Y_{t-1}, Z_{t-1}, new point x_t.

If # columns of D_{t-1}, G_{t-1}, Y_{t-1}, Z_{t-1} is larger than N, **then** remove their first column.
Compute $g_t = \nabla f(x_t)$, then compute $d_t = -\frac{\tilde{d}}{\|\tilde{d}\|}$, where $\tilde{d} = g_t - D_{t-1}(D_{t-1}^T g_t)$.
Compute $x_{t+\frac{1}{2}} = x_t + hd_t$, the *orthogonal forward estimate*.
Update $Y_t = [Y_{t-1}, x_{t+\frac{1}{2}}]$, $Z_t = [Z_{t-1}, x_t]$, $D_t = [D_{t-1}, d_t]$, $G_t =$ (9), $\varepsilon =$ (11) .
return $\nabla f(x_t)$, D_t, G_t, Y_t, Z_t, ε_t.

Algorithm 2 "Orthogonal Random Directions" Update

Require: First-order oracle for f, step-size h, memory N, new point x_t.

Generates N random orthonormal directions, e.g., $[D_t,] = \texttt{qr}(\texttt{Rand}(d, N))$.
Create matrices $Z_t = [x_t, \ldots, x_t]$, $Y_t = Z_t + hD_t$, then update $G_t =$ (9), $\varepsilon = [h, \ldots, h]$.
return $\nabla f(x_t)$, D_t, G_t, Y_t, Z_t, ε_t.

Algorithm 3 Generic Iterative Type-I Method

Require: First-order oracle f, initial iterate and smoothness x_0, M_0, # of iterations T.

```
for t = 0, ..., T − 1 do
    Update Y_t, Z_t, D_t, G_t, and ε_t (see appendix A.4).
    x_{t+1}, M_{t+1} ← [algorithm 4](f, G_t, D_t, ε_t, x_t, (M_t/2))
end for
return x_T
```

Algorithm 4 Type-I Subroutine with Backtracking Line-search

Require: First-order oracle for f, matrices G, D, vector ε, iterate x, initial smoothness M_0.

1: Initialize $M \leftarrow \frac{M_0}{2}$
2: **do**
3: $\quad M \leftarrow 2M \quad$ and $\quad H \leftarrow \frac{G^T D + D^T G}{2} + \mathrm{I}_N \frac{M\|D\|\|\varepsilon\|}{2}$
4: $\quad \alpha^\star \leftarrow \arg\min_\alpha f(x) + \nabla f(x)^T D\alpha + \frac{1}{2}\alpha^T H\alpha + \frac{M\|D\alpha\|^3}{6}$
5: $\quad x_+ \leftarrow x + D\alpha$
6: **while** $f(x_+) \geq f(x) + \nabla f(x)^T D\alpha^\star + \frac{1}{2}[\alpha^\star]^T H\alpha^\star + \frac{M\|D\alpha^\star\|^3}{6}$
7: **return** x_+, M

Algorithm 5 Type-II Subroutine with Backtracking Line-search

Same as algorithm 4, but minimize and check the upper bound (Type-II bound) instead of (Type-I bound) on lines 4 and 6.

References

[1] Hari Om Aggrawal and Jan Modersitzki. “Hessian Initialization Strategies for ℓ-BFGS Solving Non-linear Inverse Problems”. In: *International Conference on Scale Space and Variational Methods in Computer Vision*. Springer. 2021, pp. 216–228.

[2] Donald G Anderson. “Iterative procedures for nonlinear integral equations”. In: *Journal of the ACM (JACM)* 12.4 (1965), pp. 547–560.

[3] Kimon Antonakopoulos, Ali Kavis, and Volkan Cevher. “Extra-Newton: A First Approach to Noise-Adaptive Accelerated Second-Order Methods”. In: *arXiv preprint arXiv:2211.01832* (2022).

[4] Claude Brezinski. “Application de l’ε-algorithme à la résolution des systèmes non linéaires”. In: *Comptes Rendus de l’Académie des Sciences de Paris* 271.A (1970), pp. 1174–1177.

[5] Claude Brezinski. “Sur un algorithme de résolution des systèmes non linéaires”. In: *Comptes Rendus de l’Académie des Sciences de Paris* 272.A (1971), pp. 145–148.

[6] Claude Brezinski and Michela Redivo–Zaglia. “The genesis and early developments of Aitken’s process, Shanks’ transformation, the ε–algorithm, and related fixed point methods”. In: *Numerical Algorithms* 80.1 (2019), pp. 11–133.

[7] Claude Brezinski, Michela Redivo-Zaglia, and Yousef Saad. “Shanks sequence transformations and Anderson acceleration”. In: *SIAM Review* 60.3 (2018), pp. 646–669.

[8] Claude Brezinski and M Redivo Zaglia. *Extrapolation methods: theory and practice*. Elsevier, 1991.

[9] Claude Brezinski et al. “Shanks and Anderson-type acceleration techniques for systems of nonlinear equations”. In: *arXiv:2007.05716* (2020).

[10] Charles G Broyden. “The convergence of a class of double-rank minimization algorithms: 2. The new algorithm”. In: *IMA journal of applied mathematics* 6.3 (1970), pp. 222–231.

[11] Charles George Broyden. “The convergence of a class of double-rank minimization algorithms 1. general considerations”. In: *IMA Journal of Applied Mathematics* 6.1 (1970), pp. 76–90.

[12] Richard H Byrd and Jorge Nocedal. “A tool for the analysis of quasi-Newton methods with application to unconstrained minimization”. In: *SIAM Journal on Numerical Analysis* 26.3 (1989), pp. 727–739.

[13] Richard H Byrd, Jorge Nocedal, and Ya-Xiang Yuan. “Global convergence of a class of quasi-Newton methods on convex problems”. In: *SIAM Journal on Numerical Analysis* 24.5 (1987), pp. 1171–1190.

[14] Marco Canini and Peter Richtárik. “Direct nonlinear acceleration”. In: *Operational Research* 2192 (2022), p. 4406.

[15] Yair Carmon et al. “Recapp: Crafting a more efficient catalyst for convex optimization”. In: *International Conference on Machine Learning*. PMLR. 2022, pp. 2658–2685.

[16] Coralia Cartis, Nicholas IM Gould, and Philippe L Toint. "Adaptive cubic regularisation methods for unconstrained optimization. Part I: motivation, convergence and numerical results". In: *Mathematical Programming* 127.2 (2011), pp. 245–295.

[17] Coralia Cartis, Nicholas IM Gould, and Philippe L Toint. "Adaptive cubic regularisation methods for unconstrained optimization. Part II: worst-case function-and derivative-evaluation complexity". In: *Mathematical programming* 130.2 (2011), pp. 295–319.

[18] Andrew R Conn, Nicholas IM Gould, and Ph L Toint. "Convergence of quasi-Newton matrices generated by the symmetric rank one update". In: *Mathematical programming* 50.1-3 (1991), pp. 177–195.

[19] Leonardo Cunha et al. "Only tails matter: Average-Case Universality and Robustness in the Convex Regime". In: 2022.

[20] Alexandre d'Aspremont, Damien Scieur, Adrien Taylor, et al. "Acceleration methods". In: *Foundations and Trends® in Optimization* 5.1-2 (2021), pp. 1–245.

[21] William C Davidon. "Variable metric method for minimization". In: *SIAM Journal on Optimization* 1.1 (1991), pp. 1–17.

[22] Nikita Doikov, El Mahdi Chayti, and Martin Jaggi. "Second-order optimization with lazy Hessians". In: *arXiv preprint arXiv:2212.00781* (2022).

[23] Nikita Doikov, Peter Richtárik, et al. "Randomized block cubic Newton method". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 1290–1298.

[24] Carles Domingo-Enrich, Fabian Pedregosa, and Damien Scieur. "Average-case acceleration for bilinear games and normal matrices". In: *arXiv preprint arXiv:2010.02076* (2020).

[25] John R Engels and Hector J Martinez. "Local and superlinear convergence for partially known quasi-Newton methods". In: *SIAM Journal on Optimization* 1.1 (1991), pp. 42–56.

[26] Haw-Ren Fang and Yousef Saad. "Two classes of multisecant methods for nonlinear acceleration". In: *Numerical Linear Algebra with Applications* 16.3 (2009), pp. 197–221.

[27] Roger Fletcher. "A new approach to variable metric algorithms". In: *The computer journal* 13.3 (1970), pp. 317–322.

[28] Roger Fletcher and Michael JD Powell. "A rapidly convergent descent method for minimization". In: *The computer journal* 6.2 (1963), pp. 163–168.

[29] William F Ford and Avram Sidi. "Recursive algorithms for vector extrapolation methods". In: *Applied numerical mathematics* 4.6 (1988), pp. 477–489.

[30] Alexander Gasnikov et al. "Near optimal methods for minimizing convex functions with lipschitz p-th derivatives". In: *Conference on Learning Theory*. PMLR. 2019, pp. 1392–1393.

[31] Eckart Gekeler. "On the solution of systems of equations by the epsilon algorithm of Wynn". In: *Mathematics of Computation* 26.118 (1972), pp. 427–436.

[32] Saeed Ghadimi, Han Liu, and Tong Zhang. "Second-order methods with cubic regularization under inexact information". In: *arXiv preprint arXiv:1710.05782* (2017).

[33] Hiva Ghanbari and Katya Scheinberg. "Proximal quasi-Newton methods for regularized convex optimization with linear and accelerated sublinear convergence rates". In: *Computational Optimization and Applications* 69 (2018), pp. 597–627.

[34] Donald Goldfarb. "A family of variable-metric methods derived by variational means". In: *Mathematics of computation* 24.109 (1970), pp. 23–26.

[35] Robert Gower et al. "Rsn: Randomized subspace newton". In: *Advances in Neural Information Processing Systems* 32 (2019).

[36] Andreas Griewank and Ph L Toint. "Local convergence analysis for partitioned quasi-Newton updates". In: *Numerische Mathematik* 39.3 (1982), pp. 429–448.

[37] Isabelle Guyon. "Design of experiments of the NIPS 2003 variable selection benchmark". In: *NIPS 2003 workshop on feature extraction and feature selection*. Vol. 253. 2003.

[38] Isabelle Guyon et al. "Design and analysis of the causation and prediction challenge". In: *Causation and Prediction Challenge*. PMLR. 2008, pp. 1–33.

[39] Filip Hanzely et al. "Stochastic subspace cubic Newton method". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 4027–4038.

[40] K Jbilou and H Sadok. "Vector extrapolation methods. Applications and numerical comparison". In: *Journal of Computational and Applied Mathematics* 122.1-2 (2000), pp. 149–165.

[41] Khalide Jbilou and Hassane Sadok. "Analysis of some vector extrapolation methods for solving systems of linear equations". In: *Numerische Mathematik* 70.1 (1995), pp. 73–89.

[42] Khalide Jbilou and Hassane Sadok. "Some results about vector extrapolation methods and related fixed-point iterations". In: *Journal of Computational and Applied Mathematics* 36.3 (1991), pp. 385–398.

[43] Ruichen Jiang, Qiujiang Jin, and Aryan Mokhtari. "Online Learning Guided Curvature Approximation: A Quasi-Newton Method with Global Non-Asymptotic Superlinear Convergence". In: *arXiv preprint arXiv:2302.08580* (2023).

[44] Ruichen Jiang and Aryan Mokhtari. "Accelerated Quasi-Newton Proximal Extragradient: Faster Rate for Smooth Convex Optimization". In: *arXiv preprint arXiv:2306.02212* (2023).

[45] Dmitry Kamzolov et al. "Accelerated Adaptive Cubic Regularized Quasi-Newton Methods". In: *arXiv preprint arXiv:2302.04987* (2023).

[46] Dmitry Kovalev and Alexander Gasnikov. "The first optimal acceleration of high-order methods in smooth convex optimization". In: *arXiv preprint arXiv:2205.09647* (2022).

[47] Dachao Lin, Haishan Ye, and Zhihua Zhang. "Explicit convergence rates of greedy and random quasi-Newton methods". In: *Journal of Machine Learning Research* 23.162 (2022), pp. 1–40.

[48] Dachao Lin, Haishan Ye, and Zhihua Zhang. "Greedy and random quasi-newton methods with faster explicit superlinear convergence". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 6646–6657.

[49] Renato DC Monteiro and Benar Fux Svaiter. "An accelerated hybrid proximal extragradient method for convex optimization and its implications to second-order methods". In: *SIAM Journal on Optimization* 23.2 (2013), pp. 1092–1125.

[50] Yurii Nesterov. "Accelerating the cubic regularization of Newton's method on convex problems". In: *Mathematical Programming* 112.1 (2008), pp. 159–181.

[51] Yurii Nesterov. *Introductory lectures on convex optimization.* Springer, 2004.

[52] Yurii Nesterov and Boris T Polyak. "Cubic regularization of Newton method and its global performance". In: *Mathematical Programming* 108.1 (2006), pp. 177–205.

[53] Courtney Paquette et al. "Halting Time is predictable for large models: A universality property and average-case analysis". In: *Foundations of Computational Mathematics* (2022).

[54] Fabian Pedregosa and Damien Scieur. "Acceleration through spectral density estimation". In: *Proceedings of the 37th International Conference on Machine Learning (ICML).* 2020.

[55] MJD Powell. "How bad are the BFGS and DFP methods when the objective function is quadratic?" In: *Mathematical Programming* 34 (1986), pp. 34–47.

[56] Anton Rodomanov and Yurii Nesterov. "Greedy quasi-Newton methods with explicit superlinear convergence". In: *SIAM Journal on Optimization* 31.1 (2021), pp. 785–811.

[57] Anton Rodomanov and Yurii Nesterov. "New results on superlinear convergence of classical quasi-Newton methods". In: *Journal of optimization theory and applications* 188 (2021), pp. 744–769.

[58] Anton Rodomanov and Yurii Nesterov. "Rates of superlinear convergence for classical quasi-Newton methods". In: *Mathematical Programming* (2021), pp. 1–32.

[59] Katya Scheinberg and Xiaocheng Tang. "Practical inexact proximal quasi-Newton method with global complexity analysis". In: *Mathematical Programming* 160 (2016), pp. 495–529.

[60] Mark Schmidt. "minFunc: unconstrained differentiable multivariate optimization in Matlab". In: *Software available at http://www. cs. ubc. ca/˜ schmidtm/Software/minFunc. htm* (2005).

[61] Robert B Schnabel. *Quasi-Newton Methods Using Multiple Secant Equations.* Tech. rep. COLORADO UNIV AT BOULDER DEPT OF COMPUTER SCIENCE, 1983.

[62] Damien Scieur. "Generalized framework for nonlinear acceleration". In: *arXiv preprint arXiv:1903.08764* (2019).

[63] Damien Scieur, Alexandre d'Aspremont, and Francis Bach. "Regularized nonlinear acceleration". In: *Advances in Neural Information Processing Systems (NIPS).* 2016.

[64] Damien Scieur, Alexandre d'Aspremont, and Francis Bach. "Regularized nonlinear acceleration". In: *Mathematical Programming* (2020).

[65] Damien Scieur and Fabian Pedregosa. "Universal Asymptotic Optimality of Polyak Momentum". In: *Proceedings of the 37th International Conference on Machine Learning (ICML).* 2020.

[66] Damien Scieur et al. “Generalization of Quasi-Newton methods: application to robust symmetric multisecant updates”. In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2021, pp. 550–558.

[67] Damien Scieur et al. “Online Regularized Nonlinear Acceleration”. In: *arXiv:1805.09639* (2018).

[68] David F Shanno. “Conditioning of quasi-Newton methods for function minimization”. In: *Mathematics of computation* 24.111 (1970), pp. 647–656.

[69] Avram Sidi. “Convergence and stability properties of minimal polynomial and reduced rank extrapolation algorithms”. In: *SIAM Journal on Numerical Analysis* 23.1 (1986), pp. 197–209.

[70] Avram Sidi. “Efficient implementation of minimal polynomial and reduced rank extrapolation methods”. In: *Journal of Computational and Applied Mathematics* 36.3 (1991), pp. 305–337.

[71] Avram Sidi. “Extrapolation vs. projection methods for linear systems of equations”. In: *Journal of Computational and Applied Mathematics* 22.1 (1988), pp. 71–88.

[72] Avram Sidi. “Minimal polynomial and reduced rank extrapolation methods are related”. In: *Advances in Computational Mathematics* 43.1 (2017), pp. 151–170.

[73] Avram Sidi. *Vector extrapolation methods with applications*. SIAM, 2017.

[74] Avram Sidi. “Vector extrapolation methods with applications to solution of large systems of equations and to PageRank computations”. In: *Computers & Mathematics with Applications* 56.1 (2008), pp. 1–24.

[75] Avram Sidi and Jacob Bridger. “Convergence and stability analyses for some vector extrapolation methods in the presence of defective iteration matrices”. In: *Journal of Computational and Applied Mathematics* 22.1 (1988), pp. 35–61.

[76] Avram Sidi and Yair Shapira. “Upper bounds for convergence rates of acceleration methods with initial iterations”. In: *Numerical Algorithms* 18.2 (1998), pp. 113–132.

[77] Andrzej Stachurski. “Superlinear convergence of Broyden’s bounded θ-class of methods”. In: *Mathematical Programming* 20.1 (1981), pp. 196–212.

[78] Alex Toth and CT Kelley. “Convergence analysis for Anderson acceleration”. In: *SIAM Journal on Numerical Analysis* 53.2 (2015), pp. 805–819.

[79] Evgenij E Tyrtyshnikov. “How bad are Hankel matrices?” In: *Numerische Mathematik* 67.2 (1994), pp. 261–269.

[80] Homer F Walker and Peng Ni. “Anderson acceleration for fixed-point iterations”. In: *SIAM Journal on Numerical Analysis* 49.4 (2011), pp. 1715–1735.

[81] Zhe Wang et al. “A Note on Inexact Condition for Cubic Regularized Newton’s Method”. In: *arXiv preprint arXiv:1808.07384* (2018).

[82] Zengxin Wei et al. “The superlinear convergence of a modified BFGS-type method for unconstrained optimization”. In: *Computational optimization and applications* 29 (2004), pp. 315–332.

[83] Hiroshi Yabe, Hideho Ogasawara, and Masayuki Yoshino. “Local and superlinear convergence of quasi-Newton methods based on modified secant conditions”. In: *Journal of Computational and Applied Mathematics* 205.1 (2007), pp. 617–632.

[84] Hiroshi Yabe and Naokazu Yamaki. “Local and superlinear convergence of structured quasi-Newton methods for nonlinear optimization”. In: *Journal of the Operations Research Society of Japan* 39.4 (1996), pp. 541–557.

[85] Junzi Zhang, Brendan O’Donoghue, and Stephen Boyd. “Globally convergent type-I Anderson acceleration for nonsmooth fixed-point iterations”. In: *SIAM Journal on Optimization* 30.4 (2020), pp. 3170–3197.

Supplementary Materials

Appendix A. Rethinking From Scratch Quasi-Newton Methods.

A.1. First Ingredient: Subspace Approximation

Minimizing the upper bound (3) is costly in high dimension, as this requires an eigenvalue decomposition of the Hessian $\nabla^2 f(x)$ [52]. Instead, let D_t be some $N \times d$ matrix of directions (the construction of D_t will be defined later in appendix A.4). By constraining the update $x_{t+1} - x_t$ in the span of directions D_t, i.e.,

$$x_{t+1} = x_t + D_t \alpha_t, \tag{8}$$

where α_t is a vector of N coefficients, the minimization problem simplifies into

$$\alpha_t = \arg\min_{\alpha \in \mathbb{R}^N} f(x_t) + \nabla f(x_t) D_t \alpha + \tfrac{1}{2}(D_t\alpha)^T \nabla^2 f(x_t) D_t \alpha + \tfrac{L}{6}\|D_t \alpha\|^3.$$

The complexity of minimizing this upper bound is only $O(N^2 d + N^3)$ operations, where N is the number of columns of D_t (see appendix F).

A.2. Second Ingredient: Multisecant Approximation of the Hessian

Typically, (limited-memory) quasi-Newton methods approximate the Hessian using the properties of the *secant equation*,

$$\nabla^2 f(x_i)(x_i - x_{i-1}) \approx \nabla f(x_i) - \nabla f(x_{i-1}),$$

for the last N pairs of iterates. Usually, the updates are done recursively, i.e., by updating an approximation of the Hessian one secant equation at a time.

Instead, this paper approximates the Hessian using all the secant equations at once. Let the directions D_t and their associated normalized gradient difference G_t be defined as

$$D_t = \left[\frac{y_1^{(t)} - z_1^{(t)}}{\|y_1^{(t)} - z_1^{(t)}\|_2}, \ldots, \frac{y_N^{(t)} - z_N^{(t)}}{\|y_N^{(t)} - z_N^{(t)}\|_2} \right], \quad G_t = \left[\ldots, \frac{\nabla f(y_i^{(t)}) - \nabla f(z_i^{(t)})}{\|y_i^{(t)} - z_i^{(t)}\|_2}, \ldots \right]. \tag{9}$$

where the points $y_i^{(t)}$, $z_i^{(t)}$ are defined as follow:

$$Y_t = [y_1^{(t)}, \ldots, y_N^{(t)}], \quad Z_t = [z_1^{(t)}, \ldots, z_N^{(t)}]. \tag{10}$$

For instance, l-BFGS uses $Y_t = [x_{t-N}, \ldots, x_{t-1}]$ and $Z_t = [x_{t-N+1}, \ldots, x_t]$ (which will **not** be the case in this paper, see appendix A.4). Intuitively, the matrix G_t is a finite difference approximation of the Hessian-matrix product $\nabla^2 f(x) D$. More precisely, the next theorem states a bound on the approximation error of this product as a function of the *error vector* ε_t,

$$\varepsilon_t \stackrel{\text{def}}{=} [e_1^{(t)}, \ldots, e_N^{(t)}], \quad \text{and} \quad e_i^{(t)} \stackrel{\text{def}}{=} \|y_i^{(t)} - z_i^{(t)}\| + 2\|z_i^{(t)} - x\|. \tag{11}$$

Theorem 6. *Let the function f satisfy Assumption 1. Let the matrices D, G be defined as in (10) and vector ε as in (11). Then, for all $w \in \mathbb{R}^d$ and $\alpha \in \mathbb{R}^N$*

$$-\tfrac{L\|w\|}{2}|\alpha|^T \varepsilon_t \le w^T(\nabla^2 f(x) D_t - G_t)\alpha \le \tfrac{L\|w\|}{2}|\alpha|^T \varepsilon_t, \tag{12}$$

$$\|w^T(\nabla^2 f(x) D_t - G_t)\| \le \tfrac{L\|w\|}{2}\|\varepsilon_t\|. \tag{13}$$

Proof sketch The detailed proof can be found in appendix H. The main idea of the proof is as follows. From (2) with $y = y_i$ and $z = z_i$, and Assumption 1, ($\cdot^{(t)}$ is removed for clarity),

$$\frac{\|\nabla f(y_i) - \nabla f(z_i) - \nabla^2 f(x)(y_i - z_i)\|}{\|y_i - z_i\|} \leq \frac{L}{2}\|y_i - z_i\| + \|\nabla^2 f(x) - \nabla^2 f(z)\| \leq \frac{L}{2} e_i.$$

The *first* term in e_i bounds the error of (2), while the *second* comes from the distance between (2) and the current point x where the Hessian is estimated. Then, it suffices to combine the inequalities with coefficients α to obtain Theorem 6.

A.3. Third Ingredient: Objective Function and Gradient Norm Upper bounds

Since the approximation error between $\nabla^2 f(x)D$ and G can be explicitly bounded, by carefully replacing the term $\nabla^2 f(x)D\alpha$ in eqs. (2) and (3) by $G\alpha$, alongside with an appropriate regularization, leads to the **type-I** and **type-II** bounds.

Theorem 7. *Let the function f satisfy Assumption 1. Let x_{t+1} be defined as in (8), the matrices D_t, G_t be defined as in (10) and ε_t be defined as in (11). Then, for all $\alpha \in \mathbb{R}^N$,*

$$f(x_{t+1}) \leq f(x_t) + \nabla f(x_t)^T D_t\alpha + \tfrac{\alpha^T H_t \alpha}{2} + \tfrac{L\|D_t\alpha\|^3}{6}, \qquad \text{(Type-I bound)}$$

$$\|\nabla f(x_{t+1})\| \leq \|\nabla f(x_t) + G_t\alpha\| + \tfrac{L}{2}\Big(|\alpha|^T \varepsilon_t + \|D_t\alpha\|^2\Big), \qquad \text{(Type-II bound)}$$

where $H_t \overset{def}{=} \frac{G_t^T D_t + D_t^T G_t + \mathrm{I}L\|D_t\|\|\varepsilon_t\|}{2}$.

The proof can be found in appendix H. Minimizing eqs. (Type-I bound) and (Type-II bound) leads to algorithms 4 and 5, respectively, whose constant L is replaced by a parameter M, found by backtracking line-search. Type-I methods often refer to algorithms that aim to minimize the function value $f(x)$, while in contrast, type-II methods minimize the gradient norm $\|\nabla f(x)\|$ [26, 85, 14]. See algorithms 4 and 5 for the implementation details.

Solving the sub-problems In algorithms 4 and 5, the coefficients α are computed by solving a minimization sub-problem in $O(N^3 + Nd)$ (appendix F), where N is much smaller than d.

- **In algorithm 4**, the subproblem can be solved easily by a convex problem in two variables, which involves an eigenvalue decomposition of the matrix $H \in \mathbb{R}^{N\times N}$ [52].
- **In algorithm 5**, the subproblem can be cast into a linear-quadratic problem of $O(N)$ variables and constraints that can be solved efficiently with SDP solvers (e.g., `SDPT3`).

Link with qN updates and Anderson Acceleration algorithms 4 and 5 are strongly related to known quasi-Newton methods and Anderson Acceleration technique, see **??**.

A.4. Fourth Ingredient: Direction Update Rules

One critical theoretical property in the analysis is how the gradient $\nabla f(x_t)$ is aligned with the directions D_t. Since D_t is part of the algorithm design, a careful update can ensure that D_t satisfy interesting theoretical properties.

Below are some assumptions on how to update Y_t, Z_t, D_t, called **requirements**. While not overly restrictive, naive methods such as keeping only previous iterates will not satisfy those.

All convergence results rely on *one* of these conditions on the projector onto $\textbf{span}(D_t)$,

$$P_t \overset{\text{def}}{=} D_t(D_t^T D_t)^{-1} D_t^T. \tag{14}$$

1a. *For all t, the projector P_t of the stochastic matrix D_t satisfies $\mathbb{E}[P_t] = \frac{N}{d}\boldsymbol{I}$.*

1b. *For all t, the projector P_t satisfies $P_t\nabla f(x_t) = \nabla f(x_t)$.*

The first condition guarantees that, in expectation, the matrix D_t spans partially the gradient $\nabla f(x_t)$, since $\mathbb{E}[P_t\nabla f(x_t)] = \frac{N}{d}\nabla f(x_t)$. The second condition requires the possibility to move towards the current gradient when taking the step $x + D\alpha$.

In addition, it is required that the norm of $\|\varepsilon_t\|$ does not grow too quickly, hence the next assumption.

2. *For all t, the relative error $\frac{\|\varepsilon_t\|}{\|D_t\|}$ is bounded by δ.*

The Requirement 2 is also non-restrictive, as it simply prevents taking secant equations at $y_i - z_i$ and $z_i - x_i$ too far apart. Most of the time, δ satisfies the crude bound $\delta \leq O(\|x_0 - x^\star\|)$.

Finally, the condition number of the matrix D also has to be bounded.

3. *For all t, the matrix D_t is full-column rank,i.e., $D_t^T D_t$ is invertible. In addition, its condition number $\kappa_{D_t} \overset{def}{=} \sqrt{\|D_t^T D_t\|\|(D_t^T D_t)^{-1}\|}$ is bounded by κ.*

It is possible to ensure the condition with $\kappa = 1$ if the directions are orthogonal.

A.4.1. "Orthogonal Forward Estimate Only" Update Rule (Recommended)

The *"Orthogonal Forward Estimate Only"* update maintains D_t orthonormal, i.e., $D_t^T D_t = I$ for all t, while ensuring that $\nabla f(x_t)$ belongs to the span of columns of D_t (see algorithm 1). Those condition are satisfied thanks to an intermediate iterate $x_{t+\frac{1}{2}}$ that will be used to estimate $\nabla^2 f(x_t)\nabla f(x_t)$, which is called the **orthogonal forward estimate**,

$$x_{t+\frac{1}{2}} = x_t - h\left(\nabla f(x_t) - \tilde{D}_{t-1}\left(\tilde{D}_{t-1}^T \nabla f(x_t)\right)\right),$$

where $h > 0$ is a small stepsize, and $\tilde{D}_{t-1}$ is simply the matrix D_{t-1} whose first column has been removed if its number of columns equals N. This forward estimate corresponds to a step of gradient descent projected onto the orthogonal space spanned by the columns of $\tilde{D}_{t-1}$. This projection step is cheap since the orthogonality of D_t is maintained over the iterations.

After computing the forward estimate, it suffices to update the matrices Y_t, Z_t as, respectively, a moving history of the previous forward iterates and previous iterates,

$$Y_t = [x_{t-N+\frac{3}{2}}, \ldots, x_{t+\frac{1}{2}}], \qquad Z_t = [x_{t-N+1}, \ldots, x_t],$$

then compute the matrix D_t and G_t following (9), see algorithm 1 for the detailed implementation.

This method present several advantages: it ensure good theoretical performance, especially since $\kappa = 1$ (see Theorem 8), at the cost of only one extra gradient evaluation.

Theorem 8. *The "orthogonal forward estimate only" update described in algorithm 1 satisfies Requirements 1b and 3 with* $\kappa = 1$.

A.4.2. "Random Orthogonal Directions" update rule

The "Random Orthogonal Direction" update consists in creating a batch of N random orthogonal direction at each iteration, such that

$$\mathbb{E}[D_t D_t^T] = \frac{N}{d} I.$$

For instance, D_t could be the Q matrix of a `qr` decomposition of a random $N \times d$ matrix (complexity: $O(N^2 d)$), or even simpler, be an aggregation of random canonical vectors (see e.g. [39]).

Afterward, it remains to update the matrices Y_t, Z_t, G_t as follow,

$$Z_t = [x_t, \dots, x_t], \quad Y_t = Z_t + hD_t, \quad G_t = (9).$$

See algorithm 2 for the detailed implementation. The major advantage of this approach is that $\kappa = 1$ and $\delta = \sqrt{N} \cdot h$. However, N additional gradient computations are required to create the matrix G_t.

A.4.3. Other Matrix Updates: Pruning or Orthogonalization

It is possible to create other kind of matrix updates, for instance, the *Iterates only* (stores only the last forward estimate and previous iterates) or *Greedy* (stores all previous forward estimates *and* iterates) strategies, detailed below:

$$Y_t = [x_{t+\frac{1}{2}}, x_t, x_{t-1}, \dots, x_{t-N+2}], \quad Z_t = [x_t, x_{t-1}, \dots, x_{t-N+1}] \qquad \text{(Iterates only)}$$

$$Y_t = [x_{t+\frac{1}{2}}, x_t, x_{t-\frac{1}{2}}, \dots, x_{t-\frac{N+2}{2}}], \quad Z_t = [x_t, x_{t-\frac{1}{2}}, \dots, x_{t-\frac{N+1}{2}}] \qquad \text{(Greedy)}$$

However, it is impossible to ensure that the directions in D_t will be orthogonal, hence κ in Requirement 3 might be huge. Nevertheless, it is possible to bound the condition number by pruning or via an orthogonalization procedure.

Pruning. It suffices to check the condition number of D_t, then prune the columns of Y_t, Z_t, D_t, and G_t until κ is sufficiently small, for instance, until $\kappa \leq 10^3$. Note that, by the nature of those matrices, their condition number grows quickly [79, 63], hence the number of resulting column might be small.

Orthogonalization From the matrices Y_t, Z_t, the matrix D_t is computed as $D_t = \texttt{qr}(Z_t - Y_t)$. Then, the rest of the procedure follows the same steps as the "Random Orthogonal Directions" rule.

The pruning strategy is cheaper than the orthogonalization, at the cost of losing control on how large the history is. The orthogonalization technique present the same advantages as the "Random Orthogonal Directions" rule, but the directions taken might me more relevant than random ones.

Appendix B. Accelerated Algorithm

This section introduces algorithm 7, an accelerated variant of algorithm 3 for convex functions, designed using the estimate sequence technique from [50]. It consists in iteratively building a function $\Phi_t(x)$, that reads

$$\Phi_t(x) = \frac{1}{\sum_{i=0}^t b_i}\left(\sum_{i=0}^t b_i\left(f(x_i) + \nabla f(x_i)(x-x_i)\right) + \lambda_t^{(1)}\frac{\|x-x_0\|^2}{2} + \lambda_t^{(2)}\frac{\|x-x_0\|^3}{6}\right).$$

The parameters $b_i \geq 0$, $\lambda_t^{(1)}$, $\lambda_t^{(2}$ and the iterates X_t are designed by theory to ensure the following properties,

$$B_t f(x_t) \leq \min_x \phi_t(x), \qquad \phi(x) \leq B_t f(x) + \frac{\tilde{\lambda}^{(1)} + \lambda_t^{(1)}}{2}\|x-x_0\|^2 + \frac{\tilde{\lambda}^{(2)} + \lambda_t^{(2)}}{6}\|x-x_0\|^3,$$

where $B_t = \sum_{i=0}^t b_i$ and $\tilde{\lambda}^{(1)}$, $\tilde{\lambda}^{(2)}$ are constants determined by the theory.

Once the parameters are set, the accelerated algorithm operates as follow:

1. The accelerated algorithm combines linearly v_t, the optimum of Φ_t, and the previous iterate x_t.
2. It uses a slight modified version of algorithm 4, see algorithm 6.
3. There is a distinction between small and large step sizes, identifying which λ needs to be updated. The step size is considered "large" if it resembles a cubic-Newton step.

Algorithm 6 Type-I subroutine with backtracking for the accelerated method

Require: First-order oracle f, matrices G, D, vector ε, iterate x, initial smoothness parameter M_0

Initialize $M \leftarrow \frac{M_0}{2}$, `ExitFlag` $\leftarrow$ `None`
Define $\gamma_M \overset{\text{def}}{=} \frac{\kappa_D}{\|D\|}\left(\frac{3}{2}\|\varepsilon\| + 2\frac{(\|I-P)G\|}{M}\right)$
do
  $M \leftarrow 2\cdot M$ and $H_\gamma \leftarrow \frac{G^TD+D^TG}{2} + D^TD\frac{M\gamma_M}{2}$
  $\alpha^* \leftarrow \arg\min_\alpha f(x) + \nabla f(x)^T D\alpha + \frac{1}{2}\alpha^T H_\gamma \alpha + \frac{M\|D\alpha\|^3}{6}$
  $x_+ \leftarrow x + D\alpha$
  if $\frac{2}{3^{3/4}}\frac{\|\nabla f(x_+)\|^{3/2}}{\sqrt{M}} \leq -\nabla f(x_+)^T D\alpha$ **then**
    `ExitFlag` $\leftarrow$ `LargeStep`
  end if
  if $\frac{\|\nabla f(x_+)\|^2}{M(\gamma_M + \|D\alpha\|)} \leq -\nabla f(x_+)^T D\alpha$ `And` $\|D\alpha\| \leq (\sqrt{3}-1)\gamma_M$ **then**
    `ExitFlag` $\leftarrow$ `SmallStep`
  end if
while `ExitFlag` is `None`

return x_+, α, M, γ_M, `ExitFlag`

Algorithm 7 Adaptive Accelerated Type-I Iterative Algorithm

Require: First-order oracle f, initial iterate and smoothness x_0, M_0, number of iterations T.

$\lambda_0^{(1)} \leftarrow 0$, $\lambda_0^{(2)} \leftarrow 0$

Initialize G_0, D_0, ε_0 (see appendix A.4)

$\{x_1,\, M_1\} \leftarrow$ `[algorithm 4]`$(f, G_0, D_0, \varepsilon_0, x_0, M_0)$

Initialize $\ell_0^{(0)} = f(x_1), \quad \ell_0^{(1)} = 0$

for $t = 1, \ldots, T-1$ **do**

Update G_t, D_t, ε_t (see appendix A.4)

Set $b_t \leftarrow \frac{(t+1)(t+2)}{2}$, $B_t \leftarrow \frac{t(t+1)(t+2)}{6}$, $\beta_t \leftarrow \frac{3}{t+3}$.

Update $\ell_t^{(0)} \leftarrow \ell_{t-1}^{(0)} + b_{t-1}[f(x_t) - \nabla f(x_t)^T x_t], \quad \ell_t^{(1)} \leftarrow \ell_{t-1}^{(1)} + b_{t-1}\nabla f(x_t)$

do

`ValidBound` $\leftarrow$ `True`

Set $v_t \leftarrow \arg\min_v \phi_t(v)$ (See proposition 1).

Let $y_t \leftarrow \frac{3}{t+3} v_t + \frac{t}{t+3} x_t$

$$\{x_{t+1}, \alpha_t,\, M_{t+1},\, \gamma_t, \texttt{ExitFlag}\} \leftarrow \texttt{[Alg.6]}(f, G_t, D_t, \varepsilon_t, y_t, \frac{M_t}{2})$$

`%% Check if the next` ϕ `is still a lower bound for` $B_t f(x_{t+1})$

Define $\phi_+(x) = \phi_t(x) + b_t[f(x_{t+1} + \nabla f(x_{t+1})(x - x_{t+1})]$.

Set $v_+ \leftarrow \arg\min_v \phi_+(v)$ (See proposition 1).

if $\Phi_+(v_+) \leq B_t f(x_{t+1})$ **then** `%% Parameters adjustment if needed`

`ValidBound` $\leftarrow$ `False` `%% Unsuccessful iteration:` $\phi_{t+1}(v_{t+1}) \geq f(x_{t+1})$.

if `ExitFlag` is `LargeStep` **then**

If $\lambda_t^{(2)} = 0$ **then** $\lambda_t^{(2)} \leftarrow \dfrac{4}{\sqrt{3}} \dfrac{b_{t+1}^3}{B_t^2} M_{t+1}$. **Else,** $\lambda_t^{(2)} \leftarrow 2\lambda_t^{(2)}$.

else `%% Exitflag is SmallStep`

If $\lambda_t^{(1)} = 0$ **then** $\dfrac{b_{t+1}^2}{B_t} M_{t+1}\left(\gamma_t + \|D_t \alpha_t\|\right)$. **Else,** $\lambda_t^{(1)} \leftarrow 2\lambda_t^{(1)}$.

end if

else

$\{\lambda_{t+1}^{(1)}, \lambda_{t+1}^{(2)}\} \leftarrow \{\lambda_t^{(1)}, \lambda_t^{(2)}\}$ `%% Successful iteration`

end if

while `ValidBound is False`

end for

return x_T

Proposition 1. *Let v_t be the minimizer of*

$$\phi_t(v) = \ell_t^{(0)} + \left[\ell_t^{(1)}\right]^T v + \frac{\lambda_t^{(1)}}{2}\|v - x_0\|^2 + \frac{\lambda_t^{(2)}}{6}\|v - x_0\|^3.$$

where $\lambda_t^{(1)} \geq 0$, $\lambda_t^{(2)} \geq 0$. *Let* $r_t = \|v_t - x_0\|$. *Then,*

$$r_t = \|v_t - x_0\| = \begin{cases} 0 & \textit{if } \lambda_t^{(1)} = \lambda_t^{(2)} = 0 \\ \frac{\|\ell_t^{(1)}\|}{\lambda_t^{(1)}} & \textit{if } \lambda_t^{(1)} > 0 \textit{ and } \lambda_t^{(2)} = 0 \\ \frac{-\lambda_t^{(1)} + \sqrt{[\lambda_t^{(1)}]^2 + 2\lambda_t^{(2)}\|\ell_k\|}}{\lambda_2^{(2)}} & \textit{if } \lambda_t^{(2)} > 0 \end{cases}$$

$$v_t = \arg\min \Phi_t(x) = x_0 - r_t \frac{\ell_t^{(1)}}{\|\ell_t^{(1)}\|}$$

Appendix C. Related work

C.1. Inexact, Subspace, and Stochastic Newton Methods

Instead of explicitly computing the Hessian matrix and the Newton step, inexact methods compute an approximation using sampling [3], inexact Hessian computation [32, 22], or random subspaces [23, 35, 39]. These approaches substantially reduce per-iteration costs without significantly compromising the convergence rate. The convergence speed in such cases often represents an interpolation between the rates observed in gradient descent methods and (cubic) Newton's method.

C.2. Nonlinear and Anderson Acceleration

Nonlinear acceleration techniques, including Anderson acceleration [2], have a long standing history [4, 5, 31]. Driven by their promising empirical performance, they recently gained interest in their convergence analysis [71, 29, 70, 42, 76, 74, 80, 78, 63, 72, 73, 7, 67, 9, 64]. In essence, Anderson acceleration is an optimization technique that enhances convergence by extrapolating a sequence of iterates using a combination of previous gradients and corresponding iterates. Comprehensive reviews and analyses of these techniques can be found in notable sources such as [42, 8, 41, 40, 6, 20]. However, these methods do not generalize well outside quadratic minimization and their convergence rate can only be guaranteed asymptotically when using a line-search or regularization techniques [69, 75, 63].

C.3. Quasi-Newton Methods

Quasi-Newton schemes are renowned for their exceptional efficiency in continuous optimization. These methods replace the exact Hessian matrix (or its inverse) in Newton's step with an approximation updated iteratively during the method's execution. The most widely used algorithms in this category include DFP [21, 28] and BFGS [68, 34, 27, 11, 10]. Most of the existing convergence results predominantly focus on the asymptotic super-linear rate of convergence [77, 36, 13, 12, 18, 25, 84, 82, 83]. However, recent research on quasi-Newton updates has unveiled explicit and non-asymptotic rates of convergence [56, 58, 57, 47, 48]. Nonetheless, these analyses suffer from several significant drawbacks, such as assuming an infinite memory size and/or requiring access to the Hessian matrix. These limitations fundamentally undermine the essence of quasi-Newton methods, typically designed to be Hessian-free and maintain low per-iteration cost through their low memory requirement and low-rank structure.

C.4. Close Related Work

C.4.1. (Accelerated) Quasi-Newton with Secant Inexactness

Recently, Kamzolov et al. [45] introduced an adaptive regularization technique combined with cubic regularization, with global, explicit (accelerated) convergence rates for any quasi-Newton method. Based on the secant inexactness inequality, the technique introduces a quadratic regularization whose parameter is found by a backtracking line search. However, this algorithm relies on prior knowledge of the Lipschitz constant specified in Assumption 1. Unfortunately, the paper does not provide an adaptive method to find jointly the Lipschitz constant as well, as it is *a priory* too costly to know which parameter to update. This aspect makes the method impractical in real-world scenarios.

C.4.2. ARC: Adaptive Regularization algorithm using Cubics

In [16, 17] is proposed a generic framework for inexact cubic regularized Newton's steps,

$$x_{t+1} = \min_x f(x_t) + \nabla f(x_t)(x - x_t) + \frac{1}{2}(x - x_t)H_t(x - x_t) + \frac{M_t}{6}\|x - x_t\|^3,$$

where H_t is assumed to be an approximation of the Hessian $\nabla^2 f(x_t)$. However, the theoretical analysis presents numerous problems, in particular, the assumption that the norm of the current step bounds the approximation

$$\|\nabla^2 f(x_t) - H_t\| \leq C\|x_{t+1} - x_t\|,$$

for some constant C. Follow up works, such as [81], relaxed this assumption into

$$\|\nabla^2 f(x_t) - H_t\| \leq C\|x_t - x_{t-1}\|,$$

which is much weaker since it can be verified while computing the step x_{t+1}. Nevertheless, those are assumptions on the matrix H_t, but those works do not explicitly construct such a matrix. Even worse - the assumption might not be met in practice, especially if H_t is a subspace estimation of the matrix $\nabla^2 f(x_t)$.

C.4.3. Proximal Quasi-Newton Methods

The work of [59, 33] combined qN methods with proximal schemes and provided sublinear and accelerated convergence rates. However, the rates in [59] are based on a technical assumption [59, Assumption 2], for which the authors commented that "*Exploring different conditions on the Hessian approximations that ensure Assumption 2 is a subject of a separate study*", and acknowledge in their conclusion that "*Our framework does not rely on or exploit the accuracy of second-order information, and hence we do not obtain fast local convergence rates.*"

In a follow-up work, [33] proposed accelerated convergence rates under similar assumptions. However, the authors acknowledge the following: "*In our numerical results, we construct H_k via L-BFGS and ignore condition $\sigma_{k+1}H_{k+1} \preceq \sigma_k H_k$, since enforcing it in this case causes a very rapid decrease in σ. It is unclear, however, if a practical version of Algorithm 5, based on L-BFGS Hessian approximation, can be derived, which may explain why the accelerated version of our algorithm does not represent any significant advantage.*" In addition, their theoretical convergence results are based on an upper bound on the sequence σ_k, which current qN schemes cannot ensure.

C.4.4. Proximal Extragradient Quasi-Newton Methods with Online Estimation

Based on the technique in [43], [44] developed a novel quasi-Newton method with the global accelerated rate of convergence of $O(\min\{\frac{1}{t^2}; \frac{\sqrt{d\log t}}{t^{2.5}}\})$. The main ideas are as follows: the authors used the framework of inexact proximal method from [49], used an online algorithm to estimate the Hessian, and then solved a linear system involving this approximation using conjugate gradients.

The paper focuses on a different regime than this study: [44] explicitly show that it is possible to break the $O(\frac{1}{t^2})$ barrier for first order methods using full memory qN methods but this implies storing a full $d \times d$ matrix, and using it in a linear system, leading to per-iteration complexities of at least $O(d^2)$.

From a practical point of view, the algorithm requires numerous hyperparameters such as α_1, α_2, β,..., whose impact on the efficiency is rather unclear. Moreover, numerically, the algorithm improves over Nesterov's acceleration but is slower than l-BFGS on toy experiments.

Appendix D. Known rates of convergence and Comparison

D.1. (Accelerated) Gradient Descent

This section study the rate of gradient decent when function is smooth (i.e., has Lipschitz continuous gradients):

$$f(y) \leq f(x) + \nabla f(x)(y-x) + \frac{\mathcal{L}}{2}\|y-x\|^2, \tag{15}$$

Note that the class of functions considered in this paper is *not* the class of smooth functions. However, if the function satisfies Assumption 1, the Lipchits constant can be bounded as

$$\mathcal{L} \leq \|\nabla^2 f(x)\| + LR \qquad \text{for all } x \in \{x : f(x) \leq f(x_0)\}. \tag{16}$$

The rates of plain gradient descent and its accelerated version read [51] (after replacing $\mathcal{L}$)

$$\min_{0\leq i\leq t} \|\nabla f(x_i)\| \leq \sqrt{\frac{\left[\|\nabla^2 f(x)\| + LR\right](f(x_0)-f^\star)}{t+1}}, \qquad \text{(plain, non-convex)} \tag{17}$$

$$f(x_t) - f(x^\star) \leq \left[\|\nabla^2 f(x)\| + LR\right]\frac{2}{t+4}R^2, \qquad \text{(plain, convex)} \tag{18}$$

$$f(x_t) - f(x^\star) \leq \left[\|\nabla^2 f(x)\| + LR\right]\frac{4}{(t+2)^2}R^2. \qquad \text{(accelerated)} \tag{19}$$

D.2. (Accelerated) Cubic Regularized Newton's Method

When the function has a Lipschitz-continuous Hessian, the cubic regularized Newton method and its accelerated version converge with the following rates [52, 50, 39]:

$$\min_{0\leq i\leq t} \|\nabla f(x_i)\| \leq \frac{16L}{9}\left(\frac{3(f(x_0)-f^\star)}{2tM_{\min}}\right)^{2/3}, \qquad \text{(plain, non-convex)} \tag{20}$$

$$f(x_t) - f(x^\star) \leq \quad 9L\frac{R^3}{(t+4)^2}, \qquad \text{(plain, convex)} \tag{21}$$

$$\mathbb{E}[f(x_t)] - f(x^\star) \leq \left(\frac{d-N}{N}\right)\frac{\mathcal{L}(3R)^2}{2t} + \left(\frac{d}{N}\right)^2\frac{L(3R)^3}{3t^2} + O\left(\frac{1}{t^3}\right), \quad \text{(Random Subspace, convex)} \tag{22}$$

$$f(x_t) - f(x^\star) \leq L\frac{14R^3}{t(t+1)(t+2)}. \qquad \text{(accelerated)} \tag{23}$$

D.3. Relation Between Parameters

Given that this paper does not make the assumption of Lipchitz-continuous gradients, it becomes necessary to establish connections between various quantities to facilitate the comparison of rates. To streamline the notation, all numeric constants are substituted with the big O notation, and the subsequent equations are derived for the "orthogonal forward estimate only" update rule, hence $\|D\| = 1$ and $\kappa = 1$.

Relation between δ and R. The constant δ represents the upper bound on the relative error (see Requirement 2):

$$\forall t, \quad \frac{\|\varepsilon_t\|}{\|D_t\|} \leq \delta.$$

For a fixed memory, and assuming h small, since ε is the norm between iterates, δ is upper-bounded as

$$\delta \leq O(R). \tag{24}$$

Relation between the different C_i and $\mathcal{L}$ The C_1, C_2, and C_4 in Theorems 2, 3 and 5 quantifies the estimation error of $D_t^T \nabla^2 f(x_t) D_t$ by H_t in (Type-I bound) into two terms:

$$C_i \leq O\big(\delta L + \max_{i \leq t} \|(I - P_i)\nabla^2 f(x_i)\|\big).$$

The first term is the error caused by approximating $\nabla^2 f(x) D_t$ by G_t, and the second is the subspace approximation error of $\nabla^2 f(x_t)$ in the span of the columns of D_t.

Intuitively, the constants C_i can be seen as an approximation of an upper bound on $\mathcal{L}$ in a neighborhood of size δ. This is similar to (16) but the norm of the Hessian is taken in a subspace, hence the C_i's are smaller. Indeed, using (24), in the worst case, if all iterates satisfies $\|x_i - x^\star\| < R$,

$$C_i = O(RL + \max_{i \leq t} \|(I - P_i)\nabla^2 f(x_i)\|). \tag{25}$$

Other updates Note that eqs. (24) and (25) are valid only for the *"orthogonal forward estimate only"* update rule. If the random orthogonal forward estimate, or the orthogonalization of the "greedy" or "iterates only" update rules were used, the results would have been

$$\delta = O(h), \qquad C_i = O(hL + \max_{i \leq t} \|(I - P_i)\nabla^2 f(x_i)\|),$$

where h is small. However, the comparison with gradient descent or Newton's method wouldn't have been fair as the orthogonalization update rules requires N additional gradient calls.

D.4. Comparing rates of convergence

Non convex The rate from Theorem 2 reads

$$\min_{i=1,\dots,t} \|\nabla f(x_i)\| \leq \max\left\{ \frac{3L}{t^{2/3}} \left(12 \frac{f(x_0) - f^\star}{M_{\min}}\right)^{2/3} \; ; \; \left(\frac{C_1}{t^{1/3}}\right) \left(12 \frac{f(x_0) - f^\star}{M_{\min}}\right)^{1/3} \right\},$$

where $C_1 = \frac{3\delta L}{2} + \max_{i \in [0,t]} \|(I - P_i)\nabla^2 f(x_i) P_i\|$. In the case where C_1 is small, the rate matches exactly (20). In the other case, using the approximation from (25),

$$\min_{i=1,\dots,t} \|\nabla f(x_i)\| \leq \left(\frac{O(RL + \max_{i \leq t} \|(I - P_i)\nabla^2 f(x_i)\|)}{t^{1/3}} \right) \left(12 \frac{f(x_0) - f^\star}{M_{\min}}\right)^{1/3}$$

which differs significantly from (17), as the rate is $O(\frac{1}{\sqrt{t}})$. However, this might be an artifact of the theoretical analysis, since the function was not assumed to be smooth.

Star convex After using the approximation from (25), the rate from Theorem 3 reads

$$f(x_t) - f^\star \leq O\left(\frac{f(x_0)-f^\star}{t^3}\right) + O\left(\frac{LR^3}{t^2}\right) + O\left(\frac{[RL + \max_{i\leq t}\|(I-P_i)\nabla^2 f(x_i)\|]R^2}{t}\right) \tag{26}$$

The term in t^{-2} is *exactly* the one from (21), while the term is t^{-1} has the same dependency in R^3 compared to (18). However, $\|(I-P)\nabla^2 f(x_i)\|$ could be much smaller than $\|\nabla^2 f(x)\|$.

Convex with random coordinates or random subspace The rate from Theorem 4 reads

$$\mathbb{E}_{D_t}[f(x_t) - f^\star] \leq \frac{1}{1+\frac{1}{4}\left[\frac{N}{d}t\right]^3}(f(x_0)-f^\star) + \frac{1}{\left[\frac{N}{d}t\right]^2}\frac{L(3R)^3}{2} + \frac{1}{\left[\frac{N}{d}t\right]}\frac{[O(\delta L) + \frac{(d-N)}{d}\max\limits_{i\in[0,t]}\|\nabla^2 f(x_i)\|](3R)^2}{2}.$$

The rate is similar to (22), up to an additional $O(\delta L/t)$ term. This extra term comes from the estimation of the Hessian with finite difference, while the method presented in [39] uses exact Hessian-vector products.

Convex, accelerated rates After using the approximation from (25), and ignoring the terms $\tilde{\lambda}^{(1)}$, $\tilde{\lambda}^{(2)}$ for clarity, the rate from Theorem 5 reads

$$f(x_t) - f^\star \leq [RL + \max_{i=0\ldots t}\|(I-P_i)\nabla f(x_i)\|]\frac{(3R)^2}{(t+3)^2} + 9\max\{M_0\ ;\ 2L\}\left(\frac{3R}{t+3}\right)^3$$

The rate is exactly a combination of (23) and (19), but the constant associated to the $1/t^2$ rate is smaller in practice: (24) is a conservative bound and $\|(I-P_i)\nabla^2 f(x)\| \leq \|\nabla^2 f(x)\|$.

Appendix E. Link with quasi-Newton and Anderson/Nonlinear Acceleration

This section presents the fundamentals of Anderson/nonlinear acceleration (appendix E.1), quasi-Newton schemes (appendix E.2), and their relationship with the method proposed in this paper (appendix E.3).

E.1. Anderson Acceleration and Nonlinear Acceleration

Anderson acceleration, also known as nonlinear acceleration, is a powerful technique that enhances the convergence speed of fixed point iterations and optimization algorithms. Initially developed for solving linear systems, Anderson acceleration has gained popularity due to its effectiveness in accelerating iterative methods, including the ones in optimization. The method leverages previous iterations to construct an improved estimate of the objective function's minimizer.

The Anderson acceleration algorithm employs the following approximation to compute weights:

$$\nabla f\left(\sum_{i=0}^{N} \beta_i x_i\right) \approx \sum_{i=0}^{N} \beta_i \nabla f(x_i), \quad \sum_{i=0}^{N} \beta_i = 1.$$

When the function f is quadratic, this approximation becomes an equality. The underlying idea is as follows: since the optimum satisfies $\nabla f(x^\star) = 0$,

$$\sum_{i=0}^{N} \beta_i \nabla f(x_i) \approx 0 \quad \Rightarrow \nabla f\left(\sum_{i=0}^{N} \beta_i x_i\right) \approx 0 \quad \Rightarrow \sum_{i=0}^{N} \beta_i x_i \approx x^\star.$$

The Anderson acceleration steps are thus given by

$$x_{t+1} = \sum_{i=0}^{N} \beta_i^\star x_{t-i+1}, \quad \beta^\star = \arg\min_{\beta} \| \sum_{i=0}^{N} \beta_i \nabla f(x_{t-i+1}) \|^2$$

Over the past decades, the ideas behind Anderson acceleration have been refined. For example, the constraint can be eliminated by considering the step $x_{t+1} - x_t$ instead:

$$\begin{aligned} x_{t+1} - x_t &= \left(\sum_{i=0}^{N} \beta_i x_{t-i+1}\right) - x_t \\ &= \sum_{i=0}^{N} \tilde{\beta}_i x_{t-i+1}. \end{aligned}$$

The vector $\tilde{\beta}_i$ has the property that its sum equals zero. Hence, it can be rewritten as

$$\begin{aligned} x_{t+1} - x_t &= \sum_{i=1}^{N} \alpha_i (x_{t-i+1} - x_{t-i}) \\ \alpha &= \arg\min_{\alpha} \left\| \nabla f(x_t) + \sum_{i=1}^{N} \alpha_i (\nabla f(x_{t-i+1}) - \nabla f(x_{t-i})) \right\| \end{aligned}$$

where $\alpha \in \mathbb{R}^N$ has no constraint. By writing $d_i = x_{t-i+1} - x_{t-i}$, $g_i = \nabla f(x_{t-i+1}) - \nabla f(x_{t-i})$, and $D = [d_t, \dots, d_{t-N+1}]$, $G = [g_t, \dots, g_{t-N+1}]$, the step becomes

$$x_{t+1} - x_t = D_t\alpha, \quad \alpha = \arg\min_{\alpha} \|\nabla f(x_t) + G_t\alpha\|.$$

However, this version of Anderson acceleration is non-convergent because there is no contribution from $\nabla f(x_t)$ in the step $x_{t+1} - x_t$. The most popular solution to this problem is introducing a *mixing parameter* that combines gradient steps, resulting in the following expression:

$$x_{t+1} = x_t - h\nabla f(x_t) + (D - hG)\alpha, \quad \alpha = \arg\min_{\alpha} \|\nabla f(x_t) + G\alpha\|. \qquad \text{(AA Type II)}$$

Following a similar idea, recent works have introduced a type I variant of the algorithm [26, 80, 85, 14] that minimizes the function value instead of the gradient norm:

$$x_{t+1} = x_t - h\nabla f(x_t) + (D - hG)\alpha, \quad \alpha = \arg\min f(x_t) + \nabla f(x_t)D_t\alpha + \frac{1}{2}\alpha^T D_t^T G_t\alpha, \qquad \text{(AA Type I)}$$

By incorporating regularization [63, 14], globalization techniques [85], or performing a line search on the parameter h, the algorithm converges towards $x^\star$.

E.2. Single-secant and Multisecant Quasi-Newton Methods

Quasi-Newton methods, such as the Broyden-Fletcher-Goldfarb-Shanno (BFGS) method, approximate the Hessian matrix to solve unconstrained optimization problems efficiently. These methods avoid the expensive computation of the exact Hessian by using iterative updates based on previous iterates and gradients of the objective function.

This section focuses on other commonly used quasi-Newton methods: the Davidon-Fletcher-Powell (DFP) and Broyden type-1 and type-2 updates.

E.2.1. The Ideas Behind Single-Secant and Multisecant Hessian Approximation

In quasi-Newton methods, the Hessian approximation is updated using the *secant equation*, which relates the gradients and Hessian at two different points. For a twice continuously differentiable function, the secant equation is given by:

$$\nabla f(y) - \nabla f(x) = \nabla^2 f(\xi)(y - x),$$

where ξ is a point on the line segment connecting x and y. This equation serves as the basis for updating the Hessian approximation.

Based on this remarkable identity, quasi-Newton methods update an approximation of the Hessian B_t or its inverse H_t such that the approximation satisfies

$$\nabla f(x_t) - \nabla f(x_{t-1}) = B_t(x_t - x_{t-1}), \quad H_t\left(\nabla f(x_t) - \nabla f(x_{t-1})\right) = x_t - x_{t-1}.$$

What distinguishes the different updates is how to fix the remaining degrees of freedom. For instance, the simple SR-1 method updates H_t such that

$$\min_H \|H - H_{t-1}\|_F \quad : H = H^T, \;\; H\left(\nabla f(x_t) - \nabla f(x_{t-1})\right) = x_t - x_{t-1}. \qquad (27)$$

Those methods are called *single-secant* as they update H_t only one secant equation at a time. Hence, in general, H_t only satisfies the latest secant equation.

Multisecant updates, on the other hand, approximate the Hessian using a batch of secant equations. By introducing matrices $D_t = [x_{t-N+1} - x_{t-N}, \ldots, x_t - x_{t-1}]$ and $G_t = [\nabla f(x_{t-N+1}) - \nabla f(x_{t-N}), \ldots, \nabla f(x_t) - \nabla f(x_{t-1})]$, the multisecant updates satisfy

$$G_t = B_t D_t, \quad \text{or} \quad H_t G_t = D_t.$$

Unfortunately, when imposing symmetry, it is impossible to satisfy multiple secants at a time [61]. However,it is possible to enforce symmetry while approximating the secant equation in a least square sense [62, 66].

When symmetry is not imposed, the solution for B_t and H_t can be obtained as:

$$B_t = G_t[D_t]^\dagger + B_0(I - D_t D_t^\dagger), \quad H_t = D_t[G_t]^\dagger + H_0(I - G_t G_t^\dagger), \tag{28}$$

where B_0 and H_0 are the initial approximations, and $[A]^\dagger$ denotes the pseudo-inverse of matrix A. Different choices of pseudo-inverse lead to different methods.

The inversion of B_t can be computed using the Woodbury matrix identity, which provides an efficient way to compute the inverse. The update for B_t^{-1} is given by:

$$B_t^{-1} = B_0^{-1}\left(I - G_t\left(D_t^\dagger B_0^{-1} G_t\right)^{-1} D_t^\dagger B_0^{-1}\right) + D_t\left(D_t^\dagger B_0^{-1} G_t\right)^{-1} D_t^\dagger B_0^{-1}.$$

This update is equivalent to the update for H_t, given that

$$B_0^{-1} = H_0, \quad \text{and} \quad G_t^\dagger = \left(D_t^\dagger B_0^{-1} G_t\right)^{-1} D_t^\dagger B_0^{-1}. \tag{29}$$

In summary, quasi-Newton methods update the Hessian approximation using the secant equation. Single-secant methods update the approximation using the secant equation one by one, while multisecant methods use a batch of secant equations. The choice of updating strategy and pseudo-inverse affects the behavior of the method.

E.2.2. Davidon-Fletcher-Powell (DFP) Formula

The DFP formula is a Quasi-Newton update rule used to iteratively refine an approximation of the inverse Hessian matrix. It is defined as follows:

$$H_t = H_{t-1} + \frac{d_t d_t^T}{d_t^T g_t} - \frac{H_{t-1} g_t g_t^T H_{t-1}}{g_t^T H_{t-1} g_t}, \tag{30}$$

In the above equation, $g_t = \nabla f(x_t) - \nabla f(x_{t-1})$ represents the difference in gradients, and $d_t = x_t - x_{t-1}$ denotes the difference in parameter values. The DFP formula updates the matrix H_t using a rank-two matrix such that it remains symmetric and positive definite.

E.2.3. Multisecant Broyden Methods

The multisecant Broyden methods utilize the update equation from (28), where $A^\dagger$ is chosen as the Moore-Penrose pseudo-inverse of A, given by $A^\dagger = (A^T A)^{-1} A$. In this equation, B_0 and H_0 are scaled identity matrices. After simplification, the two types of updates can be expressed as follows:

$$B_t^{-1} = D_t\left(D_t^\dagger G_t\right)^{-1} D_t^\dagger + B_0^{-1}\left(I - G_t\left(D_t^\dagger G_t\right)^{-1} D_t^\dagger\right), \tag{31}$$

$$H_t = D_t(G_t^T G_t)^{-1} G_t^T + H_0\left(I - G_t\left(G_t^T G_t\right)^{-1} G_t^T\right). \tag{32}$$

Both updates are quite similar, differing mainly in the choice of the pseudo-inverse of the matrix G.

E.2.4. Link with Anderson Acceleration

The connection between quasi-Newton methods and Anderson Acceleration is strong, as, for instance, Broyden methods and Anderson acceleration are equivalent. To illustrate this, let's closely examine the update of α in (AA Type I):

$$\begin{aligned}
&x_{t+1} = x_t - h\nabla f(x_t) + (D_t - hG_t)\alpha, \quad \alpha = \arg\min f(x_t) + \nabla f(x_t) D_t\alpha + \frac{1}{2}\alpha^T D_t^T G_t \alpha \\
\Leftrightarrow &x_{t+1} = x_t - h\nabla f(x_t) + (D_t - hG_t)\alpha, \quad \alpha : D_t^T \nabla f(x_t) + D_t^T G_t \alpha = 0 \\
\Leftrightarrow &x_{t+1} = x_t - h\nabla f(x_t) + (D_t - hG_t)\alpha, \quad \alpha : \alpha = -(D_t^T G_t)^{-1} D_t^T \nabla f(x_t) \\
\Leftrightarrow &x_{t+1} = x_t - h\nabla f(x_t) - (D_t - hG_t)(D_t^T G_t)^{-1} D_t^T \nabla f(x_t). \\
\Leftrightarrow &x_{t+1} = x_t - \left(D_t(D_t^T G_t)^{-1} D_t^T + h\left(I - G_t(D_t^T G_t)^{-1} D_t^T\right)\right)\nabla f(x_t)
\end{aligned}$$

The above step is precisely the quasi-Newton step $x_{t+1} = x_t - B_t^{-1}\nabla f(x_t)$, where B_t^{-1} corresponds to the Broyden update given by Equation 31, with $B_0^{-1} = hI$. A similar reasoning can be applied to Equation 32.

When considering the single-secant updates, following the same reasoning as in Section 3 leads to the same conclusion for the SR-1 and DFP updates.

This result is expected since the approximations H_t or B_t^{-1} satisfy the single or multisecant equation:

$$H_t G_t = D_t.$$

This indicates that the matrix H_t maps vectors from the span of previous gradients to the span of previous directions. This observation justifies the construction in (8).

E.3. Links with Algorithms 4 and 5

Both Algorithms 4 and 5 can be viewed as quasi-Newton and Anderson/nonlinear acceleration schemes. The update formulas are

$$\min_\alpha f(x_t) + \nabla f(x_t)^T D_t\alpha + \frac{\alpha^T H_t \alpha}{2} + \frac{M\|D_t\alpha\|^3}{6}, \quad H_t \stackrel{\text{def}}{=} \frac{G_t^T D_t + D_t^T G_t + \mathrm{I}M\|D_t\|\|\varepsilon_t\|}{2}. \tag{Type I}$$

$$\min_\alpha \|\nabla f(x_t) + G_t\alpha\| + \frac{M}{2}\Big(\sum_{i=1}^N |\alpha_i|[\varepsilon_t]_i + \|D_t\alpha\|^2\Big), \tag{Type II}$$

The resemblance with Anderson/nonlinear acceleration is strong, as the objective function is similar. If the function is quadratic, $L = 0$ and therefore M can also be set to 0; hence, the coefficients α are *exactly* the type I and type II Anderson steps eqs. (AA Type I) and (AA Type II).

The same idea holds when compared to quasi-Newton methods. In both cases, the optimal solution $\alpha^\star$ can be written implicitly:

$$\alpha^\star = -\left(H_t + \frac{M D_t^T D_t \|D_t\alpha^\star\|}{6}\right)^{-1} D_t^T \nabla f(x_t), \qquad \text{(Type I - solution)}$$

$$\alpha^\star = -\left(G_t^T G_t + \tilde{M} D_t^T D_t\right)^{-1} \left(G_t^T \nabla f(x) + \frac{\tilde{M}\|\varepsilon_t\|}{2} \partial(|\alpha^\star|)\right), \qquad \text{(Type II - solution)}$$

where $\tilde{M} \stackrel{\text{def}}{=} \|\nabla f(x_t) + G_t\alpha\| M$ and $\partial(|\alpha^\star|)$ is a subgradient of $|\alpha^*|$. The step then reads

$$x_{t+1} = x_t + D\alpha^\star \qquad \text{(Generic step)}$$

$$x_{t+1} = x_t - D_t\left(H_t + \frac{M D_t^T D_t \|D_t\alpha^\star\|}{6}\right)^{-1} D_t^T \nabla f(x_t), \qquad \text{(Type I - step)}$$

$$x_{t+1} = x_t - D_t\left(G_t^T G_t + \tilde{M} D_t^T D_t\right)^{-1} \left(G_t^T \nabla f(x) + \frac{\tilde{M}\|\varepsilon_t\|}{2} \partial(|\alpha^\star|)\right), \qquad \text{(Type II - step)}$$

Type I is a quasi-Newton step with a symmetrization of $G^T D$ and a regularization. In contrast, the type II step can be seen as a quasi-Newton method with a regularization on $G^\dagger$, with a correction term on the gradient. Therefore the Hessian approximation reads

$$B_t^{-1} = D_t\left(H_t + \frac{M D_t^T D_t \|D_t\alpha^\star\|}{6}\right)^{-1} D^T, \quad H_t = D_t\left(G_t^T G_t + \tilde{M} D_t^T D_t\right)^{-1} G_t^T.$$

Again, when the objective function is quadratic, $L = 0$ and therefore $M = 0$. Moreover, when f is quadratic, the matrix multiplication $D^T G$ satisfies $D^T G + G^T D = 2D^T G$ as $D^T G$ becomes symmetric. Hence,

$$x_{t+1} = x_t - D_t\left(D_t^T G_t\right)^{-1} D_t^T \nabla f(x_t), \qquad \text{(Type I - quadratic)}$$

$$x_{t+1} = x_t - D_t\left(G_t^T G_t\right)^{-1} G_t^T \nabla f(x_t), \qquad \text{(Type II quadratic)}$$

The steps are *exactly* the type I and type II multisecant Broyden methods from eqs. (31) and (32), with the only difference that there is no initialization H_0 or B_0.

Appendix F. Solving the sub-problems

Solving the Type 1 Subproblem The Type 1 subproblem is a well-studied problem that involves minimizing a specific objective function. A method proposed by [52] has proven to be efficient for solving this problem. The method utilizes eigenvalue decomposition on a matrix to find the optimal solution. In this paper, the matrix involved in this problem is relatively small, therefore eigenvalue decomposition is not a concern even for large-scale problems. The subproblem aims to determine the norm of the solution, and this can be achieved through solving one nonlinear equation using bisection or secant method.

Solving the Type 2 Subproblem The Type 2 subproblem can be formulated as a Second-Order Cone Program (SOCP). The objective function of this subproblem consists of three terms: a norm term, a sum of absolute values term, and a quadratic term. The norm term can be transformed using singular value decomposition, and the sum of absolute values term can be expressed as with linear constraints. The quadratic term can be simplified using a rotated quadratic cone. By utilizing these techniques, the Type 2 subproblem can be effectively solved using existing SOCP solvers.

F.1. Solving the Type 1 Subproblem

The Type 1 subproblem can be expressed as follows:

$$\min_{\alpha} \nabla f(x) D\alpha + \frac{1}{2}\alpha^T H \alpha + \frac{M}{6}\|D\alpha\|^3,$$

where H is symmetric but not necessarily positive definite. This problem has been well-studied, and [52] proposed an efficient method to solve it using eigenvalue decomposition on the matrix H. Although eigenvalue decomposition may be challenging for large-scale problems, it is not a concern here since $H \in \mathbb{R}^{N\times N}$, with a relatively small N (e.g., $N = 25$ in the experiments).

In essence, the subproblem involves determining the norm of the solution $r = \|\alpha\|$. This can be accomplished through a simple bisection on the following system of nonlinear equations:

$$\left(H + \frac{MD^T Dr}{2} I\right)\alpha = -D^t \nabla f(x), \quad \|\alpha\| = r, \quad r \geq -\lambda_{\min}(H). \tag{33}$$

Interestingly, this problem is equivalent to the following formulation, as shown in Proposition 2:

$$\left(\Lambda + \frac{Mr}{2} I\right)\tilde{\alpha} = -V^T (D^T D)^{-1/2} D^t \nabla f(x),\ \|\alpha\| = r,\ r \geq -\lambda_{\min}(H),\ \tilde{\alpha} = V^T (D^T D)^{1/2}\alpha, \tag{34}$$

which involves the eigenvalue decomposition $(D^T D)^{-1/2} H (D^T D)^{-1/2} = V\Lambda V^T$.

Proposition 2. *Problems (33) and (34) are equivalent.*

Proof. The first step is to split $D^TD = (D^TD)^{1/2}(D^TD)^{1/2}$ and then employ an eigenvalue decomposition on $(D^TD)^{-1/2}H(D^TD)^{-1/2} = V\Lambda V^T$ (where V is orthonormal due to the symmetry of the matrix):

$$\begin{aligned}
&\left(H + \frac{MD^TDr}{2}I\right)\alpha = -D^t\nabla f(x) \\
\Leftrightarrow &(D^TD)^{1/2}\left((D^TD)^{-1/2}H(D^TD)^{-1/2} + \frac{Mr}{2}I\right)(D^TD)^{1/2}\alpha = -D^t\nabla f(x) \\
\Leftrightarrow &(D^TD)^{1/2}V\left(\Lambda + \frac{Mr}{2}I\right)V^T(D^TD)^{1/2}\alpha = -D^t\nabla f(x) \\
\Leftrightarrow &\left(\Lambda + \frac{Mr}{2}I\right)V^T(D^TD)^{1/2}\alpha = -V^T(D^TD)^{-1/2}D^t\nabla f(x) \\
\Leftrightarrow &\left(\Lambda + \frac{Mr}{2}I\right)\tilde{\alpha} = -V^T(D^TD)^{-1/2}D^t\nabla f(x).
\end{aligned}$$

□

Once the eigenvalue decomposition is performed, the subproblem (34) becomes relatively simple since it involves solving a diagonal system of equations for a fixed value of r. The main objective is to find an interval $[r_{\min}, r_{\max}]$ that encompasses the optimal value $r = \|\alpha\|$. Once this interval is identified, a straightforward bisection or secant method can be employed to obtain the optimal solution.

Finding initial bounds Starting with $r_{\min} = \max\{0, -\lambda_{\min(H)}\}$ and $r_{\max} = \max\{2r_{\min}, 1\}$,

$$\text{do } r_{\max} \leftarrow 2r_{\max} \quad \text{while } \|\tilde{\alpha}\| \geq r_{\max}.$$

where $\tilde{\alpha} = -\left(\Lambda + \frac{Mr_{\max}}{2}I\right)^{-1}V^T(D^TD)^{-1/2}D^t\nabla f(x)$. Increasing $r_{\max}$ increases the regularization, hence reduces the norm of $\tilde{\alpha}$.

Finding α After $r^\star$ has been found such that $|r^\star - \|\tilde{\alpha}\||$ is sufficiently small, the best α is simply

$$\alpha = (D^TD)^{-1/2}V\tilde{\alpha} = -(D^TD)^{-1/2}V\left(\Lambda + \frac{Mr^\star}{2}I\right)^{-1}V^T(D^TD)^{-1/2}D^t\nabla f(x).$$

In the case where the diagonal matrix is not invertible, which happens when $r^\star = r_{\min}$, it suffices to use the pseudo-inverse instead.

Note that D^TD is an $N \times N$ matrix, where N is small, therefore, computing its inverse is inexpensive. Moreover, when D is orthogonal, $D^TD = I$, therefore there is no need to invert it. In addition, $(\Lambda + \frac{Mr^\star}{2}I)^{-1}$ can be computed in $O(N)$ complexity since the matrix is diagonal.

F.2. Solving the Type 2 Subproblem

The Type 2 subproblem is given by:

$$\min_{\alpha} \underbrace{\|\nabla f(x) + G\alpha\|}_{\textbf{(a)}} + \frac{L}{2}\Big(\underbrace{\sum_{i=1}^{N} |\alpha_i|\varepsilon_i}_{\textbf{(b)}} + \underbrace{\|D\alpha\|^2}_{\textbf{(c)}}\Big). \tag{35}$$

Although it may not be immediately apparent, this subproblem can be formulated as a Second-Order Cone Program (SOCP) with $O(N)$ variables and constraints.

F.2.1. Fundamentals of SOCP

SOCP solvers handle the following conic problems:

$$\begin{aligned}
&\min_{x,t_i,\omega_i} c_0 x + \sum_i c_i[t_i;\omega_i] \quad \text{subject to} \\
&A_0 x + \sum_{i=1}^{k} A_i[t_i;\omega_i] = b \\
&x \geq 0 \\
&(t_i,\omega_i) \in \mathcal{K}_i \quad \Leftrightarrow t_i \geq \|\omega_i\|, \ \ t \geq 0.
\end{aligned} \tag{SOCP Standard Matrix Form}$$

Here, k represents the number of cones, and the cone $\mathcal{K}$ refers to the second-order cone, also known as the *Lorenz* cone.

A useful transformation is the *rotated quadratic cone*, defined as follows:

$$[a,b,c] \in \mathcal{K}_q \quad \Leftrightarrow \quad 2ab \geq \|c\|^2.$$

The rotated quadratic cone can be reformulated as a second-order cone using a linear transformation:

$$\text{if} \quad \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & I_K \end{bmatrix} \begin{bmatrix} t \\ \omega^{(0)} \\ \omega \end{bmatrix} \quad \text{then} \quad (t,[\omega^{(0)};\omega]) \in \mathcal{K} \quad \Leftrightarrow \quad [a,b,c] \in \mathcal{K}_q.$$

Thanks to this transformation, the rotated quadratic cone can be included in SOCP solvers.

F.2.2. SOCP Formulation of the Type 2 Subproblem

The SOCP of (35) is composed of the three terms **a**, **b**, and **c**.

Term (a) Let $U_G\Sigma_G V_G^T$ be the singular value decomposition of G. Write $P_G = U_G U_G^T$ as the projector onto the columns of G. Then,

$$\begin{aligned}
\|\nabla f(x) + R\alpha\| &= \|P_G \nabla f(x) + P_G G\alpha + (I - P_G)\nabla f(x)\| \\
&= \sqrt{\|P_G \nabla f(x) + R\alpha\|^2 + \|(I - P_G)\nabla f(x)\|^2} \\
&= \sqrt{\left\|U_G \left(U_G^T \nabla f(x) + \Sigma_G V_G^T \alpha\right)\right\|^2 + \|(I - P_G)\nabla f(x)\|^2} \\
&= \sqrt{\left\|U_G^T \nabla f(x) + \Sigma_G V_G^T \alpha\right\|^2 + \|(I - P_G)\nabla f(x)\|^2}
\end{aligned}$$

Let the vector $\omega_1 = \left[U_G^T \nabla f(x) + \Sigma_G V\alpha;\ \|(I - P_G)\nabla f(x)\|\right]$. Hence,

$$\|\nabla f(x) + G\alpha\| = \min_{t_1,\,\alpha,\,\omega_1} t_1 : (t_1, \omega_1) \in \mathcal{K}_L, \quad \omega_1 = \left[U_G^T \nabla f(x) + \Sigma_G V\alpha;\ \|(I - P_G)\nabla f(x)\|\right].$$

Term (b) This term is standard in linear programming. Let $\alpha = \alpha_+ - \alpha_-$, with $\alpha_+, \alpha_- \geq 0$,

$$\sum_{i=1}^{N} |\alpha_i|\varepsilon_i = \sum_{i=1}^{N} (\alpha_+ + \alpha_-)\varepsilon_i.$$

Term (c) Let $U_D \Sigma_D V_D^T$ be the singular value decomposition of D. Using the rotated cone, the constraint can be written as

$$2t_3 b \geq \|U_D \Sigma_D V_D \alpha\|^2 = \|\Sigma_D V_D \alpha\|^2, \quad b = \frac{1}{2}.$$

Using the transformation into a Lorenz cone, this is equivalent to

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \Sigma_D V_D^T \end{bmatrix} \begin{bmatrix} t_3 \\ b \\ \alpha \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & I_k \end{bmatrix} \begin{bmatrix} t_2 \\ \omega_2^{(0)} \\ \omega_2 \end{bmatrix}, \quad b = \frac{1}{2}, \quad (t_2,\, [\omega_2^{(0)},\, \omega_2]) \in \mathcal{K}.$$

Simplification. Note that, since $b = \frac{1}{2}$, the value can be immediately replaced. Same idea with t_3: the constraint is written as

$$t_3 = \frac{t_2 + \omega_2^{(0)}}{\sqrt{2}}, \quad t_3 \geq 0.$$

Since, by construction, $t_2 \geq \omega_2^{(0)}$ and $t_2 \geq 0$, t_3 always satisfies the condition, which means both t_3 and its constraint can be removed. The constraints thus simplify into

$$\begin{bmatrix} \frac{1}{2} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \Sigma_D V_D^T \end{bmatrix} \begin{bmatrix} \alpha \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & I_k \end{bmatrix} \begin{bmatrix} t_2 \\ \omega_2^{(0)} \\ \omega_2 \end{bmatrix}, \quad (t_2,\, [\omega_2^{(0)},\, \omega_2]) \in \mathcal{K}.$$

Final formulation Gathering all terms, the final SOCP formulation reads

$$
\begin{aligned}
\text{minimize} \quad & t_1 + \frac{L}{2}\Big((\alpha_+ + \alpha_-)^T \varepsilon + t_2\Big) \\
\text{subject to} \quad & \omega_1 = \Big[U_G^T \nabla f(x) + \Sigma_G V_G^T \alpha \; ; \; \|(I - P_G)\nabla f(x)\|\Big], \\
& \alpha_+, \alpha_- \geq 0 \\
& \alpha = \alpha_+ - \alpha_- \\
& \begin{bmatrix} \mathbf{0}_{1\times N} & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ \Sigma_D V_D^T & \mathbf{0}_{N\times 1} & \mathbf{0}_{N\times 1} & -I_N \end{bmatrix} \begin{bmatrix} \alpha \\ t_2 \\ \omega_2^{(0)} \\ \omega_2 \end{bmatrix} = \begin{bmatrix} 0 \\ -\frac{1}{2} \\ \mathbf{0}_{N\times 1} \end{bmatrix} \\
& (t_1, \omega_1) \in \mathcal{K}, \quad (t_2, [\omega_2^{(0)}; \omega_2]) \in \mathcal{K}_L, \quad t_2 \geq 0.
\end{aligned}
$$

Standard matrix formulation The SOCP can be written under the standard matrix form (SOCP Standard Matrix Form). Let the variables

$$
\alpha_+,\, \alpha_- \geq 0, \quad (t_1, \omega_1) \in \mathcal{K}_1, \quad (t_2, [\omega_2^{(0)} \omega_2]) \in \mathcal{K}_2,
$$

where t_1, t_2, and $\omega_2^{(0)}$ are scalars, ω_2, α_+, and α_- are vectors of size N, and ω_1 is a vector of size $N+1$. The SOCP matrices read

$$
\begin{aligned}
c_0 &= \begin{bmatrix} \frac{L\varepsilon^T}{2} & \frac{L\varepsilon^T}{2} \end{bmatrix} \quad c_1 = \begin{bmatrix} 1 & \mathbf{0}_{1\times N+1} \end{bmatrix} \quad c_2 = \begin{bmatrix} \frac{L}{2\sqrt{2}} & \frac{L}{2\sqrt{2}} & \mathbf{0}_{1\times N} \end{bmatrix} \\
A_0 &= \begin{bmatrix} -\Sigma_G V_G^T & \Sigma_G V_G^T \\ \mathbf{0}_{2\times N} & \mathbf{0}_{2\times N} \\ \Sigma_D V_D^T & -\Sigma_D V_D^T \end{bmatrix} \\
A_1 &= \begin{bmatrix} \mathbf{0}_{N+1\times 1} & I_{N+1\times N+1} \\ \mathbf{0}_{N+1\times 1} & \mathbf{0}_{N+1\times N+1} \end{bmatrix} \\
A_2 &= \begin{bmatrix} \mathbf{0}_{N+1\times 1} & \mathbf{0}_{N+1\times 1} & \mathbf{0}_{N+1\times N} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \mathbf{0}_{1\times N} \\ \mathbf{0}_{N\times 1} & \mathbf{0}_{N\times 1} & -I_{N\times N} \end{bmatrix} \\
b &= \begin{bmatrix} \nabla f(x)^T U_G & \|(I - P_R)\nabla f(x)\| & -\frac{1}{2} & \mathbf{0}_{N\times 1} \end{bmatrix}^T.
\end{aligned}
$$

This completes the SOCP formulation of the type 2 subproblem.

Appendix G. Additional Numerical Experiments

This section presents additional numerical experiments.

Methods The methods compared are the type 1 and type 2 steps with the following strategies: *Iterate only*, *Forward estimate only*, *Greedy* (refer to appendix A.4), and the accelerated type 1 method with the strategy *forward estimate only.* The batch methods are not included as they perform poorly regarding the number of Oracle calls. The baseline is the l-BFGS method from `minFunc` [60].

Method parameters In all experiments, the memory of the methods is set to $N = 25$ and the h for the forward estimates is set to $h = 10^{-9}$. The parameters of the l-BFGS are left untouched except for the memory. The initial point is $x_0 = \nabla f(0_d)$.

Functions The minimized problems are square loss with cubic regularization, logistic loss with small quadratic regularization, and the generalized Rosenbrock function. The regularization parameter of the square loss is set to $1e-3$ times the norm of the Hessian, and the regularization of the logistic loss is set to $1e-10$ times the square norm of the feature matrix.

Dataset The datasets for the square and the logistic loss are Madelon [37], Sido0 [38], and Marti2 [38] datasets.

Post-processing The dataset matrix is normalized by its norm, then a vector of ones is concatenated to the data matrix.

G.1. Initial Parameter for the Backtracking Line search

The backtracking line search was used in all experiments. The estimation of the initial value M_0 (see (36)) is based on the following observation. Since the function satisfies Assumption 1,

$$\|\nabla f(y) - \nabla f(x) - \nabla^2 f(x)(y-x))\| \leq \frac{L}{2}\|y-x\|^2,$$

for some x, y. Hence, the parameter L can be estimated as

$$L \approx 2\frac{\|\nabla f(y) - \nabla f(x) - \nabla^2 f(x)(y-x))\|}{\|y-x\|^2}.$$

Now, define

$$s_h \overset{\text{def}}{=} h\nabla f(x_0),$$

for some small h and $\tau > 1$, and let $x = x_0$ and $y = x_0 + s_{\tau h}$. Indeed, if h is small, then

$$\tau\left[\nabla f(x_0 + s_h) - \nabla f(x_0)\right] \approx \tau \nabla^2 f(x) s_h = \nabla^2 f(x) s_{\tau h}.$$

Therefore,

$$\|\nabla f(x_0+s_{\tau h}) - \nabla f(x_0) - \tau\left[\nabla f(x_0 + s_h) - \nabla f(x_0)\right]\| \approx \|\nabla f(x_0+s_{\tau h}) - \nabla f(x_0) - \nabla^2 f(x) s_{\tau h}\|,$$

and hence, the Lipchitz constant can be estimated as

$$M_0 = \frac{2}{\|s_{\tau h}\|^2} \|\nabla f(x_0 + s_{\tau h}) - \nabla f(x_0) - \tau \left[\nabla f(x_0 + s_h) - \nabla f(x_0)\right] \|. \tag{36}$$

In the experiments, h is the same as the algorithm, and $\tau = 10$. Various choices of τ, h have been tested without significantly impacting the numerical convergence.

G.2. Scalability w.r.t. Dimension and Memory

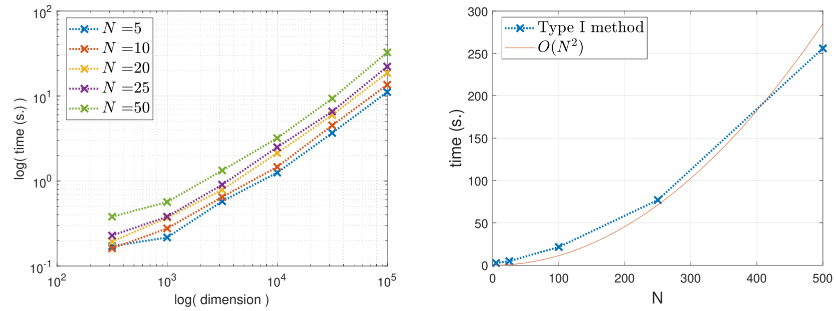

Figure 2: Scaling of the Type 1 method with the "orthogonal forward estimates only" updates rules w.r.t. N and d to minimize a random logistic regression function. As predicted by the theory, the scaling is linear in the dimension and quadratic w.r.t. N. The proposed method is suitable for large-scale problems, as it can quickly solve problems with $d \approx 10^6$.

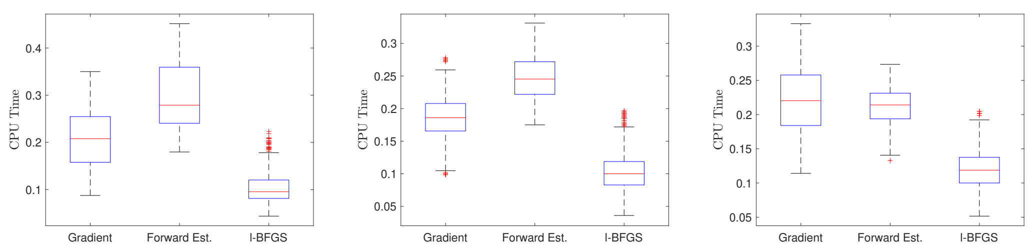

Figure 3: Distribution of the per-iteration time for three methods. The memory parameter of l-BFGS and the type I method is set to (left to right) $N = 5, 25, 100$. The time required by the l-BFGS algorithm increases slightly when N grows, and the per-iteration computation time is approximately two times faster than the type I method. Surprisingly, the total computation time of the type-1 method remains constant for different N because the condition in the backtracking line search is more often satisfied. Note that the $\times 2$ factor between l-BFGS and the type 1 method is expected since the type 1 method requires at least 2 gradient calls.

G.3. Influence of h

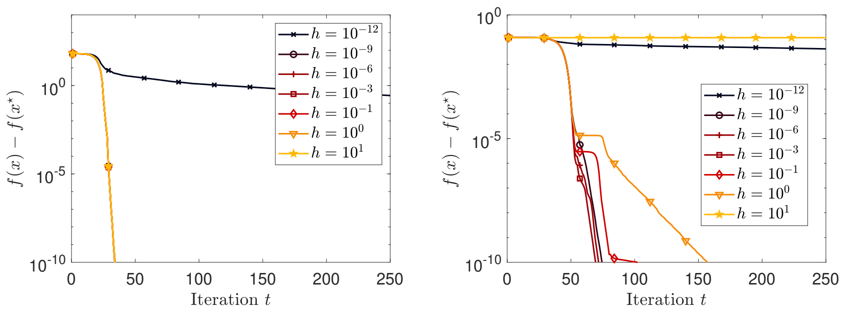

Figure 4: Influence of the step size h to compute the forward estimate $x_{+\frac{1}{2}}$ in the "orthogonal forward estimates only" updates rules on the Madelon dataset to minimize a (left) quadratic and (right) a logistic loss. The range of acceptable h is rather large. For instance, this range is $[10^{-9}, 10^{-1}]$ when minimizing the logistic loss.

G.4. Impact of the memory parameter N

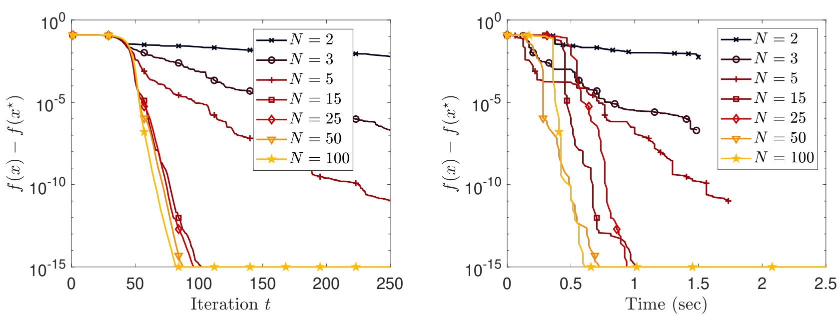

Figure 5: Impact of the memory size N on the convergence rate of the type 1 method with the "Orthogonal forward estimate" update rule to minimize a logistic loss on the Madelon dataset. Left: number of iterations versus suboptimality, right: time versus suboptimality. Overall, it is always better to increase the memory parameter in terms of the number of iterations, but there is an effect of diminishing returns.

G.5. Nonconvex optimization

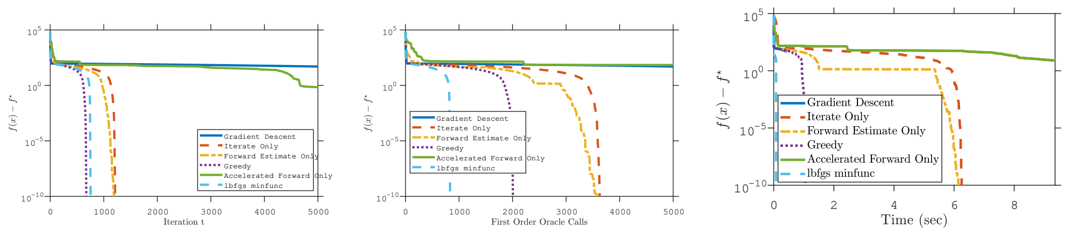

Figure 6: Comparison of type 1 methods on the Generalized Rosenbrock function in $\mathbb{R}^{100}$.

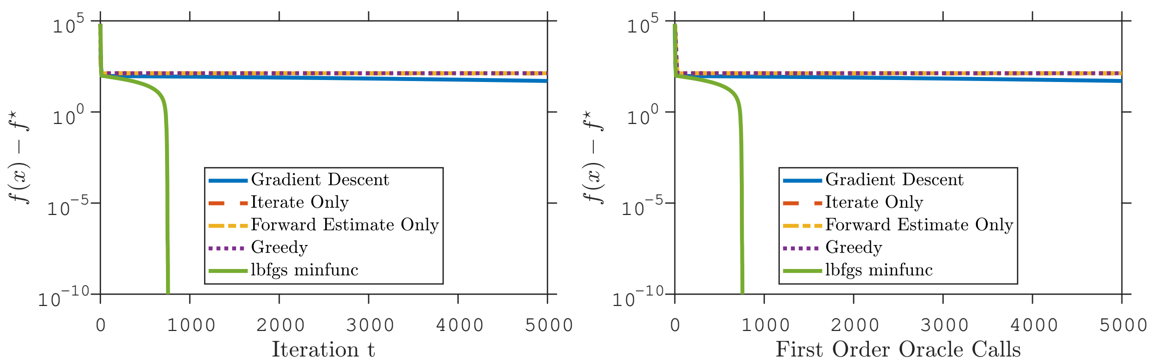

Figure 7: Comparison of type 2 methods on the Generalized Rosenbrock function in $\mathbb{R}^{100}$

G.6. Comparison of Type 1 Methods on Convex Problems

G.6.1. Square loss and cubic regularization

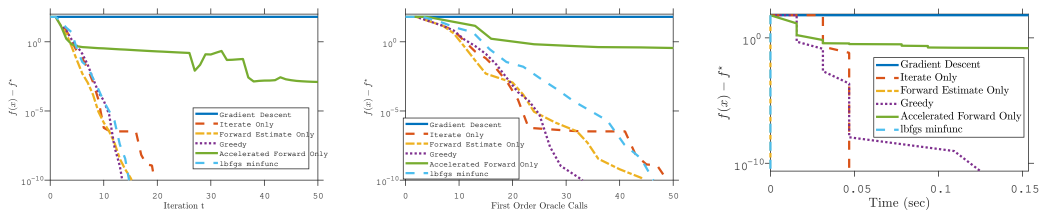

Figure 8: Comparison of type 1 methods: Square loss and cubic regularization on Madelon dataset

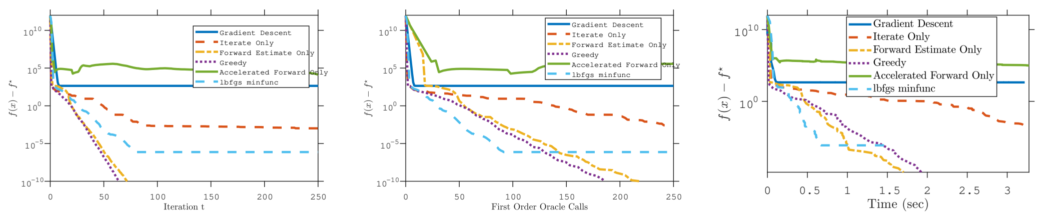

Figure 9: Comparison of type 1 methods: Square loss and cubic regularization on sido0 dataset

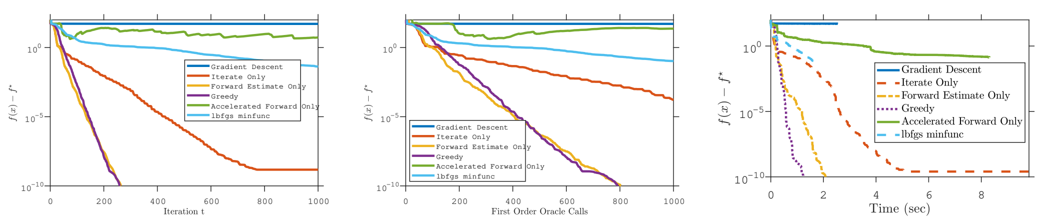

Figure 10: Comparison of type 1 methods: Square loss and cubic regularization on marti2 dataset

G.6.2. Logistic regression

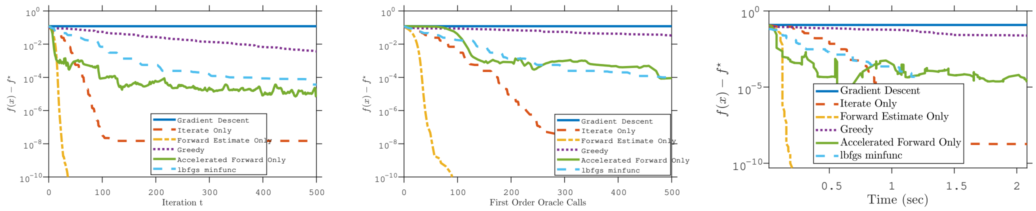

Figure 11: Comparison of type 1 methods: Logistic loss and cubic regularization on Madelon dataset

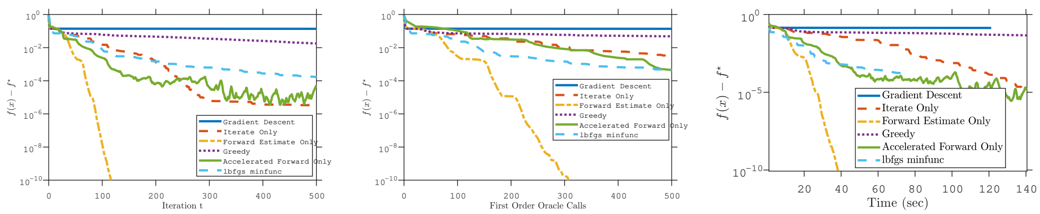

Figure 12: Comparison of type 1 methods: Logistic loss and cubic regularization on sido0 dataset

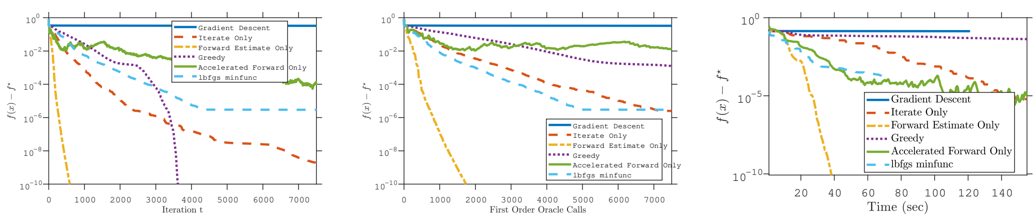

Figure 13: Comparison of type 1 methods: Logistic loss and cubic regularization on marti2 dataset

G.7. Comparison of Type 2 Methods on Convex Problems

The type-2 method was not the focus of this study. Its prototypical implementation is rather slow, hence, the time VS suboptimality graph are not showed.

G.7.1. Square loss and cubic regularization

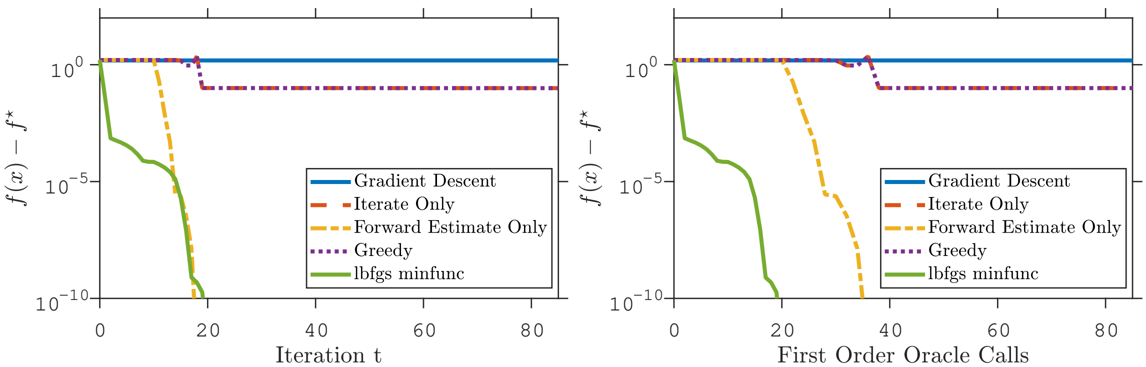

Figure 14: Comparison of type 2 methods: Square loss and cubic regularization on Madelon dataset

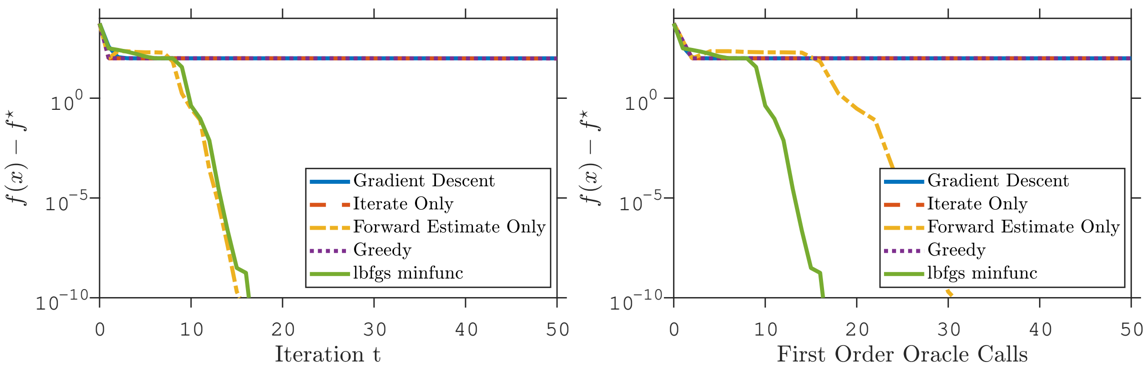

Figure 15: Comparison of type 2 methods: Square loss and cubic regularization on sido0 dataset

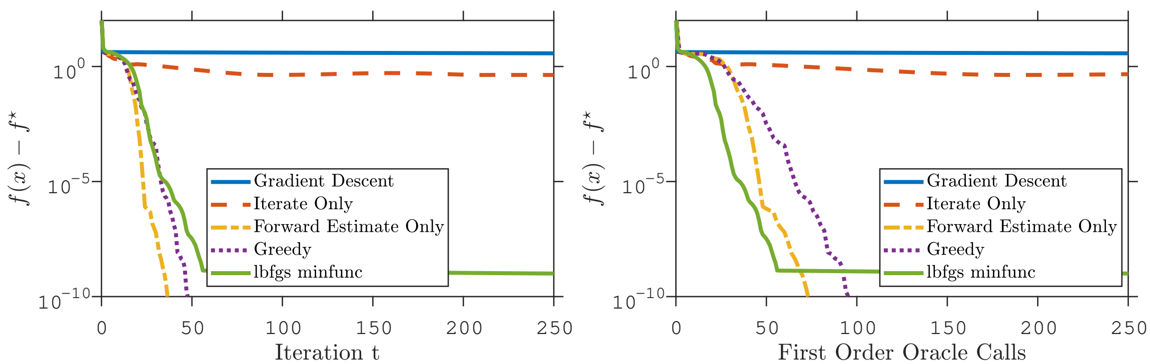

Figure 16: Comparison of type 2 methods: Square loss and cubic regularization on marti2 dataset

G.7.2. Logistic regression

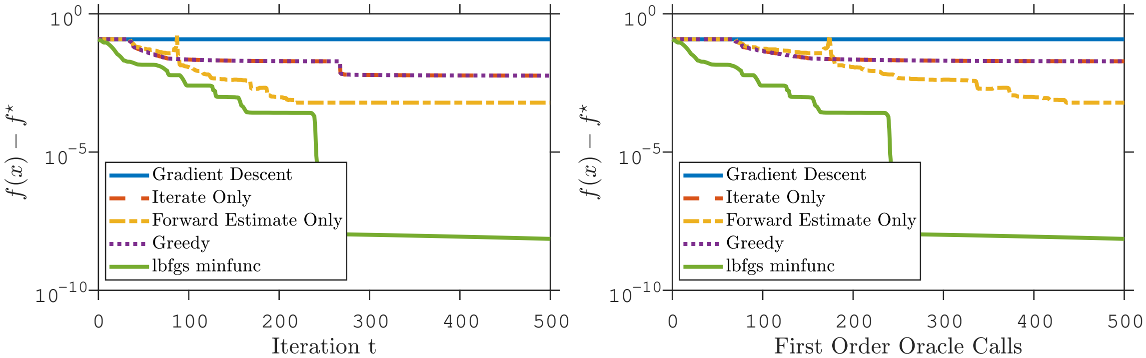

Figure 17: Comparison of type 2 methods: Logistic loss and cubic regularization on Madelon dataset

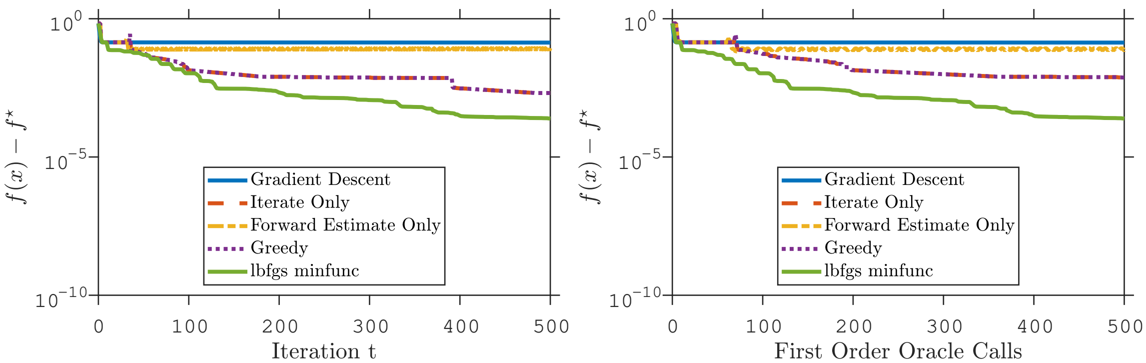

Figure 18: Comparison of type 2 methods: Logistic loss and cubic regularization on sido0 dataset

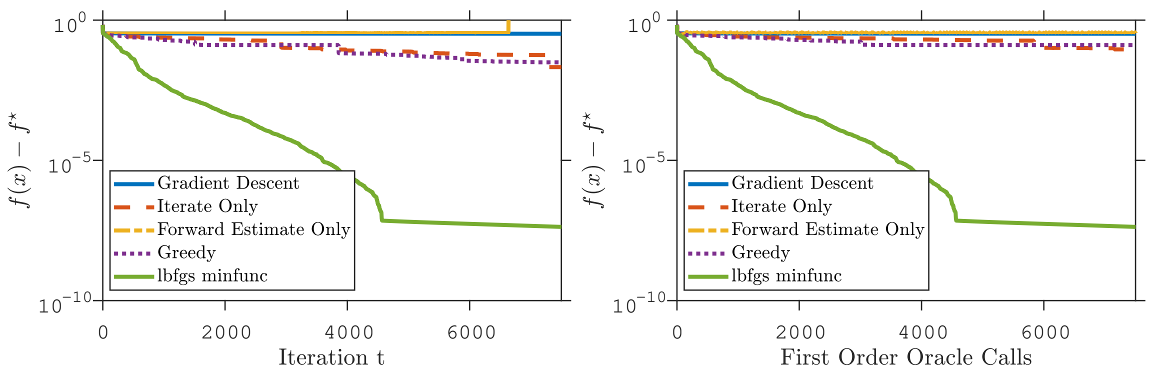

Figure 19: Comparison of type 2 methods: Logistic loss and cubic regularization on marti2 dataset

Appendix H. Missing proofs

In this section, when not needed, the subscript t has been removed for clarity. The following definitions simplify the notations:

$$D_\dagger = (D^T D)^{-1} D^T, \tag{37}$$

$$D_\dagger^T = D(D^T D)^{-1}, \tag{38}$$

$$\kappa_D = \|D_\dagger\| \|D\|, \tag{39}$$

Note that the pseudo inverse $D_\dagger$ exists under Requirement 3. Note that

$$D_\dagger D = I, \qquad D D_\dagger = P_D = P.$$

H.1. Technical Result: Hessian Approximation

This section presents technical results related to the approximation of the Hessian $\nabla^2 f(x)$. To simplify notations, let the matrices H_0 and $\tilde{H}_0$ be

$$H_0 = \frac{D^T R + R^T D}{2}, \qquad \tilde{H}_0 = D_\dagger^T H_0 D_\dagger = \frac{PGD_\dagger + D_\dagger^T G^T P}{2}. \tag{40}$$

Intuitively, $\tilde{H}_0$ is the Hessian approximation, while H_0 is the approximation of the quadratic form $D^T \nabla^2 f(x) D$.

Proposition 3 (Subspace Hessian Approximation Error)**.** *Assume D satisfies Requirement 1b. Then, the following holds:*

$$\left\| \left(\tilde{H}_0 - P \nabla^2 f(x) P \right) D\alpha \right\| \leq \frac{L}{2} \|D_\dagger\| \|\varepsilon\| \|D\alpha\|$$

Proof. Since $D^T D_\dagger = D_\dagger^T D^T = P$, $D_\dagger D = \mathrm{I}$, $PD = D$, $\|P\| = 1$, and using (40),

$$\begin{aligned}
& \left\| \left[\frac{PGD_\dagger + D_\dagger^T G^T P}{2} - P\nabla^2 f(x) P \right] D\alpha \right\| \\
\leq \;& \frac{1}{2} \left(\left\| (PGD_\dagger - P\nabla^2 f(x) P) D\alpha \right\| + \left\| (D_\dagger^T G^T P - P\nabla^2 f(x) P) D\alpha \right\| \right) \\
\leq \;& \frac{1}{2} \left(\left\| G\alpha - \nabla^2 f(x) D\alpha \right\| + \|D_\dagger\| \left\| (G^T - D^T \nabla^2 f(x)) D\alpha \right\| \right)
\end{aligned}$$

Using inequality (12) for the first term and (13) for second gives

$$\left\| \left[\frac{PGD_\dagger + D_\dagger^T G^T P}{2} - P\nabla^2 f(x) P \right] D\alpha \right\| \leq \frac{1}{2} \left(\frac{L}{2} |\alpha|^T \varepsilon + \|D_\dagger\| \frac{L\|D\alpha\|}{2} \|\varepsilon\| \right)$$

Because $|\alpha|^T \varepsilon \leq \|\alpha\| \|\varepsilon\| \leq \|D_\dagger\| \|D\alpha\| \|\varepsilon\|$,

$$\left\| \left[\frac{PGD_\dagger + D_\dagger^T G^T P}{2} - P\nabla^2 f(x) P \right] D\alpha \right\| \leq \frac{L}{2} \|D_\dagger\| \|\varepsilon\| \|D\alpha\|.$$

□

Proposition 4. *[Out-of-subspace Error Estimation] Let the function f satisfy Assumption 1. Let the matrices D, G be defined as in (10) and vector ε as in (11). Then, for all $\alpha \in \mathbb{R}^N$,*

$$\frac{\|(I-P)\nabla^2 f(x) D\alpha\|}{\|D\alpha\|} \leq \left(\|(I-P)G\| + L\|\varepsilon\|\right)\frac{\kappa_D}{\|D\|}.$$

Proof. Indeed, using (13),

$$\begin{aligned}
\|(I-P)\nabla^2 f(x) D\alpha\| &= \|(I-P)(G - G + \nabla^2 f(x) D)\alpha\| \\
&\leq \|(I-P)(\nabla^2 f(x) D - G)\alpha\| + \|(I-P)G\alpha\| \\
&\leq \|(\nabla^2 f(x) D - G)\alpha\| + \|(I-P)G\alpha\| \\
&\leq \|\nabla^2 f(x) D - G\|\|\alpha\| + \|(I-P)G\alpha\| \\
&\leq \left(\frac{L\|\varepsilon\|}{2}\|\alpha\| + \|(I-P)G\alpha\|\right)
\end{aligned}$$

Hence,

$$\frac{\|(I-P)\nabla^2 f(x) D\alpha\|}{\|D\alpha\|} \leq \frac{\left(\frac{L\|\varepsilon\|}{2}\|\alpha\| + \|(I-P)G\alpha\|\right)}{\|D\alpha\|}.$$

Moreover,

$$\begin{aligned}
\frac{\|(I-P)\nabla^2 f(x) D\alpha\|}{\|D\alpha\|} &\leq \left(\frac{L\|\varepsilon\|}{2} + \|(I-P)G\|\right)\|\alpha\|. \\
&\leq \left(\frac{L\|\varepsilon\|}{2} + \|(I-P)G\|\right)\frac{\|\alpha\|}{\|D\alpha\|} \\
&\leq \max_\alpha \left(\frac{L\|\varepsilon\|}{2} + \|(I-P)G\|\right)\frac{\|\alpha\|}{\|D\alpha\|} \\
&= \left(\frac{L\|\varepsilon\|}{2} + \|(I-P)G\|\right)\sigma_{\min}^{-1}(D).
\end{aligned}$$

The desired result follows from the fact that $\kappa_D = \frac{\|D\|}{\sigma_{\min}(D)}$. □

H.2. Technical Results: Cubic Subproblem

This section presents results on the properties of the solution of the cubic subproblem

$$\alpha^\star \stackrel{\text{def}}{=} \arg\min_\alpha \nabla f(x)^T(D\alpha) + \frac{1}{2}(D\alpha)^T \tilde{H}_\Gamma (D\alpha) + \frac{M}{6}\|D\alpha\|^3, \qquad x_+ = x + D\alpha^\star \tag{41}$$

where $\tilde{H}_\Gamma \in \mathbb{R}^{d\times d}$ is a rank N matrix such that

$$\tilde{H} = D_\dagger^T H_\gamma D_\dagger, \qquad \Leftrightarrow \quad H = D^T \tilde{H}_\Gamma D, \qquad H_\gamma = \frac{R^T D + D^T R + \Gamma}{2}, \tag{42}$$

and Γ is a $N\times N$ matrix. For instance, setting $\Gamma = M\|\varepsilon\|\|D\|I$ gives the H in algorithm 4.

Proposition 5. *The first-order and second-order conditions of the subproblem (41) read*

$$D^T\nabla f(x) + H_\Gamma \alpha + \frac{M}{2} D^T D\alpha \|D\alpha\| = 0, \tag{43}$$

$$H_\Gamma + \frac{M}{2} D^T D \|D\alpha\| \succeq 0. \tag{44}$$

Proof. See [50], equation (3.3), and [52], equation (2.7). □

Proposition 6. *Let f satisfies Assumption 1 and $B \in \mathbb{R}^{d\times d}$ be any matrix. Assume the matrix D satisfies Requirement 1b, and α satisfies the first-order condition (43). Let $\tilde{H}_\Gamma$ be defined in (42). Then,*

$$\begin{aligned} \|\nabla f(x) + BD\alpha - \nabla f(x_+)\| &= \|(\tilde{H}_\Gamma - B + \frac{M\|D\alpha\|}{2})D\alpha + \nabla f(x_+)\| && (45)\\ &\leq \frac{L}{2}\|D\alpha\|^2 + \|[B - \nabla^2 f(x)]D\alpha\|. && (46) \end{aligned}$$

Then, the following equation follows from the optimality condition multiplied by $D(D^TD)^{-1}$, writing $P = DD_\dagger = D_\dagger^T D^T$, assuming $P\nabla f(x) = \nabla f(x)$,

$$\nabla f(x) + (\tilde{H}_\Gamma + \frac{M\|D\alpha\|}{2})D\alpha = 0.$$

Replacing $\nabla f(x)$ gives

$$\|\nabla f(x) + BD\alpha - \nabla f(x_+)\| = \| - (\tilde{H}_\Gamma + \frac{M\|D\alpha\|}{2})D\alpha + BD\alpha - \nabla f(x_+)\|,$$

which is the desired result.

Proof. The inequality follows directly from (2),

$$\begin{aligned} \|\nabla f(x) + BD\alpha - \nabla f(x_+)\| &\leq \|\nabla f(x) + \nabla^2 f(x)D\alpha - \nabla f(x_+)\| + \|BD\alpha - \nabla^2 f(x)D\alpha\| \\ &\leq \frac{L}{2}\|D\alpha\|^2 + \|[B - \nabla^2 f(x)]D\alpha\|. \end{aligned}$$

□

Proposition 7. *Assume D satisfies Requirement 1b. Let $\tilde{H}$ be defined in (42). Then, for all $\tilde{\Gamma}$, if*

$$B = \tilde{H}_\Gamma - \frac{1}{2} D_\dagger \tilde{\Gamma} D_\dagger^T$$

in proposition 6, the following holds:

$$\left\| \left(\frac{1}{2} D_\dagger \tilde{\Gamma} D_\dagger^T + \frac{M\|D\alpha\|}{2} \right) D\alpha + \nabla f(x_+) \right\| \leq \frac{L}{2}\|D\alpha\|^2 + \|[B - \nabla^2 f(x)]D\alpha\|, \tag{47}$$

where

$$\|[B - \nabla^2 f(x)]D\alpha\| \leq \|D\alpha\| \left(\frac{L}{2}\|D_\dagger\|\|\varepsilon\| + \frac{\|(I-P)\nabla^2 f(x) D\alpha\|}{\|D\alpha\|} + \frac{1}{2}\left\| D_\dagger(\Gamma - \tilde{\Gamma})D_\dagger \right\| \right)$$

Proof. From proposition 6,

$$\|(\tilde{H}_\Gamma - B + \frac{M\|D\alpha\|}{2})D\alpha + \nabla f(x_+)\| \leq \frac{L}{2}\|D\alpha\|^2 + \|[B - \nabla^2 f(x)]D\alpha\|.$$

Replacing B in the left-hand-side gives

$$\|(\tilde{H}_\Gamma - B + \frac{M\|D\alpha\|}{2})D\alpha + \nabla f(x_+)\| = \|(\frac{D_\dagger \Gamma D_\dagger^T}{2} + \frac{M\|D\alpha\|}{2})D\alpha + \nabla f(x_+)\|$$

Since

$$\nabla^2 f(x)D\alpha = P\nabla^2 f(x)PD\alpha + (I-P)\nabla^2 f(x)PD\alpha,$$

where $P = D(D^TD)^{-1}D^T$, and because $PD = D$, the inequality becomes

$$\begin{aligned}
\|[B - \nabla^2 f(x)]D\alpha\| =& \| \left[\tilde{H}_\Gamma - \frac{1}{2}D_\dagger\tilde{\Gamma}D_\dagger^T - \nabla^2 f(x)\right] D\alpha\| && (48)\\
=& \|[P + (I-P)]\left[\tilde{H}_\Gamma - \frac{1}{2}D_\dagger\tilde{\Gamma}D_\dagger^T - \nabla^2 f(x)\right] PD\alpha\| && (49)\\
\leq& \left\|\left(\tilde{H}_0 - P\nabla^2 f(x)P\right)D\alpha\right\| && (50)\\
&+ \left(\frac{1}{2}\left\|D_\dagger^T(\Gamma - \tilde{\Gamma})D_\dagger\right\| + \frac{\|(I-P)\nabla^2 f(x)D\alpha)\|}{\|D\alpha\|}\right)\|D\alpha\| && (51)
\end{aligned}$$

□

Corollary 1 (Bound depending on $\tilde{\Gamma}$)**.** *In proposition 7,*

- *if* $\tilde{\Gamma} = 0$ *and* $\Gamma = M\|D\|\|\varepsilon\|I$,

$$\left\|\frac{M\|D\alpha\|}{2}D\alpha + \nabla f(x_+)\right\| \leq \frac{L}{2}\|D\alpha\|^2 + \|D\alpha\|\left(\frac{\|\varepsilon\|}{\|D\|}\left(\frac{L + M\kappa_D}{2}\right)\kappa_D + \|(I-P)\nabla^2 f(x)P\|\right) \tag{52}$$

- *if* $\tilde{\Gamma} = \Gamma$,

$$\left\|\left(\frac{1}{2}D_\dagger\Gamma D_\dagger^T + \frac{M\|D\alpha\|}{2}\right)D\alpha + \nabla f(x_+)\right\| \leq \frac{L}{2}\|D\alpha\|^2 + \|D\alpha\|\left(\frac{L}{2}\frac{\|\varepsilon\|}{\|D\|}\kappa_D + \frac{\|(I-P)\nabla^2 f(x)D\alpha\|}{\|D\alpha\|}\right) \tag{53}$$

- *If* $\tilde{\Gamma} = D(M\|D\alpha\|)D^T$ *and* $\Gamma = M\|D\|\|\varepsilon\|I$,

$$\|\nabla f(x_+)\| \leq \frac{L+M}{2}\|D\alpha\|^2 + \|D\alpha\|\left(\frac{\|\varepsilon\|}{\|D\|}\left(\frac{L + M\kappa_D}{2}\right)\kappa_D + \|(I-P)\nabla^2 f(x)P\|\right) \tag{54}$$

H.3. Technical Results: Decrease Guarantees

This section presents two technical results on the minimal decrease of the function f.

Proposition 8. *Let Assumption 1 and Requirements 1b to 3 hold. Then, $\forall y \in \mathbb{R}^d$, algorithm 4 ensures*

$$f(x_+) \leq f(y) + \frac{M+L}{6}\|y-x\|^3 + \frac{\|y-x\|^2}{2}\left(\|\nabla^2 f(x) - P\nabla^2 f(x) P\| + \delta\frac{L\kappa + M\kappa^2}{2}\right)$$

Proof. The output of algorithm 4 ensures that

$$f(x_+) \leq \min_\alpha f(x) + \nabla f(x)^T D\alpha + \frac{1}{2}(D\alpha)^T\nabla^2 f(x) D\alpha + \frac{1}{2}\alpha^T\left(H - D^T\nabla^2 f(x) D\right)\alpha + \frac{M}{6}\|D\alpha\|^3$$

However, by the definition of H (Type-I bound),

$$\begin{aligned}
&\frac{1}{2}\alpha^T\left(H - D^T\nabla^2 f(x) D\right)\alpha \\
\leq&\frac{1}{2}\left(\alpha^T\left(\frac{G^TD + D^TG}{2} - D^T\nabla^2 f(x)D\right)\alpha + \|\alpha\|^2\frac{M\|D\|\|\varepsilon\|}{2}\right) \\
\leq&\frac{1}{2}\left(\alpha^T\left(\frac{G^TD + D^TG}{2} - D^T\nabla^2 f(x)D\right)\alpha + \|D^\dagger\|^2\|D\alpha\|\frac{M\|D\|\|\varepsilon\|}{2}\right) \\
=&\frac{1}{2}\left((D\alpha)^T\left(G - \nabla^2 f(x)D\right)\alpha + \|D^\dagger\|^2\|D\alpha\|\frac{M\|D\|\|\varepsilon\|}{2}\right).
\end{aligned}$$

The last equality comes from the fact that

$$\alpha^T\left(D^TG\right)\alpha = \alpha^T\left(\frac{D^TG + G^TD}{2} + \frac{D^TG - G^TD}{2}\right)\alpha = \alpha^T\left(\frac{D^TG + G^TD}{2}\right)\alpha.$$

Now, using (12) with $w = D\alpha$ gives

$$\frac{1}{2}\alpha^T\left(H - D^T\nabla^2 f(x)D\right)\alpha \leq \frac{L\|D\alpha\|}{4}\sum_{i=1}^N |\alpha_i|\varepsilon_i + \|D^\dagger\|^2\|D\alpha\|\frac{M\|D\|\|\varepsilon\|}{4}.$$

Finally, since

$$\sum_{i=1}^N |\alpha_i|\varepsilon_i \leq \|\alpha\|\|\varepsilon\| \leq \|D^\dagger\|\|D\alpha\|\|\varepsilon\|,$$

the inequality becomes

$$\begin{aligned}
\frac{1}{2}\alpha^T\left(H - D^T\nabla^2 f(x)D\right)\alpha &\leq \frac{\|D\alpha\|^2}{4}\left(L\|D^\dagger\|\|\varepsilon\| + M\|D^\dagger\|^2\|D\|\|\varepsilon\|\right) \\
&= \frac{\|D\alpha\|^2}{4}\frac{\|\varepsilon\|}{\|D\|}\left(L\kappa_D + M\kappa_D^2\right).
\end{aligned}$$

All together,

$$
\begin{aligned}
& f(x_+) \\
\le & \min_\alpha f(x) + \nabla f(x)^T D\alpha + \frac{1}{2}(D\alpha)^T \nabla^2 f(x) D\alpha + \frac{1}{2}\alpha^T \left(H - D^T \nabla^2 f(x) D\right)\alpha + \frac{M}{6}\|D\alpha\|^3 \\
\le & \min_\alpha f(x) + \nabla f(x)^T D\alpha + \frac{1}{2}(D\alpha)^T \nabla^2 f(x) D\alpha + \frac{\|D\alpha\|^2}{4}\frac{\|\varepsilon\|}{\|D\|}\left(L\kappa_D + M\kappa_D^2\right) + \frac{M}{6}\|D\alpha\|^3
\end{aligned}
$$

Now, by Requirement 3, for all y, one can find α such that

$$D\alpha = P(y-x) = DD^\dagger(y-x).$$

Indeed, multiplying both sides by $D^\dagger$ gives

$$\alpha = D^\dagger(y-x).$$

Therefore, the minimum can be written as a function of y instead of α,

$$
\begin{aligned}
f(x_+) \le \min_{y\in\mathbb{R}^d}\ & f(x) + \nabla f(x)^T P(y-x) + \frac{1}{2}(P(y-x))^T \nabla^2 f(x) P(y-x) \\
& + \frac{\|P(y-x)\|^2}{4}\frac{\|\varepsilon\|}{\|D\|}\left(L\kappa_D + M\kappa_D^2\right) + \frac{M}{6}\|P(y-x)\|^3. \qquad (55)
\end{aligned}
$$

Since $P\nabla f(x) = \nabla f(x)$ by Requirement 1b, and using the crude bound $\|P(y-x)\| \le \|y-x\|$,

$$
\begin{aligned}
f(x_+) \le \min_{y\in\mathbb{R}^d}\ & f(x) + \nabla f(x)^T(y-x) + \frac{1}{2}(y-x)^T \nabla^2 f(x)(y-x) \\
& + \frac{1}{2}(y-x)\left[\nabla^2 f(x) - P\nabla^2 f(x) P\right](y-x) \\
& + \frac{\|y-x\|^2}{4}\frac{\|\varepsilon\|}{\|D\|}\left(L\kappa_D + M\kappa_D^2\right) + \frac{M}{6}\|y-x\|^3.
\end{aligned}
$$

Using the lower bound (3),

$$f(x) + \nabla f(x)^T(y-x) + \frac{1}{2}(y-x)^T\nabla^2 f(x)(y-x) - \frac{L}{6}\|y-x\|^3 \le f(y),$$

the crude bound $(y-x)\left[\nabla^2 f(x) - P\nabla^2 f(x)P\right](y-x) \le \|\nabla^2 f(x) - P\nabla^2 f(x) P\|\|y-x\|^2$, and Requirements 2 and 3 lead to the desired result,

$$f(x_+) \le f(y) + \frac{M+L}{6}\|y-x\|^3 + \frac{\|y-x\|^2}{2}\left(\|\nabla^2 f(x) - P\nabla^2 f(x)P\| + \delta\frac{L\kappa + M\kappa^2}{2}\right)$$

□

Proposition 9. *Let Assumption 1 and Requirements 1a, 2 and 3 hold. Then, $\forall y \in \mathbb{R}^d$, algorithm 4 ensures*

$$
\begin{aligned}
\mathbb{E} f(x_+) \le & \left(1 - \frac{N}{d}\right) f(x) + \frac{N}{d} f(y) + \frac{N}{d}\frac{(M+L)}{6}\|y-x\|^3 \\
& + \frac{N}{d}\frac{\|y-x\|^2}{2}\left(\delta\frac{L\kappa + M\kappa^2}{2} + \frac{(d-N)}{d}\|\nabla^2 f(x)\|\right)
\end{aligned}
$$

Proof. The proof is the same as for proposition 8, until equation (55),

$$f(x_+) \leq \min_{y\in\mathbb{R}^d} \; f(x) + \nabla f(x)^T P(y-x) + \frac{1}{2}(P(y-x))^T \nabla^2 f(x) P(y-x)$$
$$+ \frac{\|P(y-x)\|^2}{4}\frac{\|\varepsilon\|}{\|D\|}\left(L\kappa_D + M\kappa_D^2\right) + \frac{M}{6}\|P(y-x)\|^3.$$

With Requirement 1a, the following relations hold (see [39, lemma 5.7])

$$\mathbb{E}[\|P(y-x)\|^2] = (y-x)^T\mathbb{E}[P](y-x) = \frac{N}{d}\|y-x\|^2, \tag{56}$$

$$\mathbb{E}[\|P(y-x)\|^3] \leq \mathbb{E}[\|P(y-x)\|^2]\|y-x\| = \frac{N}{d}\|y-x\|^2, \tag{57}$$

$$\mathbb{E}[(y-x)^T P \nabla^2 f(x) P(y-x)] \leq \frac{N^2}{d^2}(y-x)\nabla^2 f(x)(y-x) + \frac{N(d-N)}{d^2}\|\nabla^2 f(x)\|\|y-x\|^2 \tag{58}$$

Hence, removing the minimum and taking the expectation of (55) gives

$$\begin{aligned}\mathbb{E}f(x_+) \leq& f(x) + \frac{N}{d}\nabla f(x)^T(y-x)\\ &+ \frac{1}{2}\left(\frac{N^2}{d^2}(y-x)\nabla^2 f(x)(y-x) + \frac{N(d-N)}{d^2}\|\nabla^2 f(x)\|\|y-x\|^2\right)\\ &+ \frac{N}{d}\frac{\|y-x\|^2}{4}\frac{\|\varepsilon\|}{\|D\|}\left(L\kappa_D + M\kappa_D^2\right) + \frac{N}{d}\frac{M}{6}\|y-x\|^3.\end{aligned}$$

Using the lower bound from (3)

$$\frac{1}{2}(y-x)\nabla^2 f(x)(y-x) \leq f(y) + \frac{L}{6}\|y-x\|^3 - f(x) - \nabla f(x)(y-x)$$

in the inequality over the expectation gives

$$\begin{aligned}\mathbb{E}f(x_+) \leq& f(x) + \frac{N}{d}\nabla f(x)^T(y-x)\\ &+ \frac{N^2}{d^2}\left(f(y) + \frac{L}{6}\|y-x\|^3 - f(x) - \nabla f(x)(y-x)\right)\\ &+ \frac{1}{2}\frac{N(d-N)}{d^2}\|\nabla^2 f(x)\|\|y-x\|^2\\ &+ \frac{N}{d}\frac{\|y-x\|^2}{4}\frac{\|\varepsilon\|}{\|D\|}\left(L\kappa_D + M\kappa_D^2\right) + \frac{N}{d}\frac{M}{6}\|y-x\|^3.\end{aligned}$$

After simplification,

$$\begin{aligned}\mathbb{E}f(x_+) \leq& \left(1-\frac{N^2}{d^2}\right)f(x) + \frac{N^2}{d^2}f(y) + \frac{N}{d}\left(1-\frac{N}{d}\right)\nabla f(x)^T(y-x)\\ &+ \frac{1}{2}\frac{N(d-N)}{d^2}\|\nabla^2 f(x)\|\|y-x\|^2\\ &+ \frac{N}{d}\frac{\|y-x\|^2}{4}\frac{\|\varepsilon\|}{\|D\|}\left(L\kappa_D + M\kappa_D^2\right) + \left(\frac{N^2 L}{6d^2} + \frac{NM}{6d}\right)\|y-x\|^3.\end{aligned}$$

To simplify the expression, since $N \leq d$,

$$\left(\frac{N^2 L}{6d^2} + \frac{NM}{6d}\right) \|y-x\|^3 \leq \frac{N(M+L)}{6d}\|y-x\|^3.$$

Finally, since the function is convex,

$$\frac{N}{d}\left(1-\frac{N}{d}\right)\nabla f(x)^T(y-x) \leq \frac{N}{d}\left(1-\frac{N}{d}\right)(f(y)-f(x)).$$

From this last relation, Requirement 2 and Requirement 3 comes the desired result,

$$\begin{aligned}\mathbb{E}f(x_+) \leq \left(1-\frac{N}{d}\right)f(x) + \frac{N}{d}f(y) + \frac{N(M+L)}{6d}\|y-x\|^3 \\ + \frac{\|y-x\|^2}{2}\left(\frac{N}{d}\delta\frac{L\kappa + M\kappa^2}{2} + \frac{N(d-N)}{d^2}\|\nabla^2 f(x)\|\right)\end{aligned}$$

□

H.4. Technical Results: Accelerated Algorithm

Notations The following functions define the estimate sequence,

$$\ell_t(x) = \sum_{i=2}^{t} b_{i-1}\left(f(x_i) + \nabla f(x_i)(x-x_i)\right), \tag{59}$$

$$\phi_t(x) = f(x_1) + \ell_t(x) + \frac{\lambda_t^{(1)}}{2}\|x-x_0\|^2 + \frac{\lambda_t^{(2)}}{6}\|x-x_0\|^3 \tag{60}$$

$$\Phi_t(x) = \frac{\phi_t(x)}{B_t}, \tag{61}$$

where $\lambda_t^{(1,2)}$ are non-negative and increasing, and the sequences b_t, B_t are

$$B_t = \frac{t(t+1)(t+2)}{6} = \sum_{i=1}^{t} b_i, \tag{62}$$

$$b_t = \frac{(t+1)(t+2)}{2} = B_{t+1} - B_t. \tag{63}$$

(64)

Moreover, the following quantities will be important later,

$$v_t = \arg\min_x \phi_t(x) = \arg\min_x \Phi_t(x), \tag{65}$$

$$\beta_t = \frac{b_t}{B_{t+1}}, \tag{66}$$

$$y_t = (1-\beta_t)x_t + \beta_t v_t. \tag{67}$$

Lemma 1. *From [50, Lemma 4]. The Bregman divergence of the function $\|x\|^i$ satisfies, for $i \geq 2$,*

$$\|x\|^i - \|y\|^i - \nabla(\|y\|^i)(x-y) \geq \frac{1}{2^{i-2}}\|x-y\|^i.$$

Proposition 10. *The function ϕ_t is lower-bounded by*

$$\phi_t \geq \underbrace{\phi_t(v_t)}_{=\phi_t^\star} + \frac{\lambda_t^{(1)}}{2}\|x - v_t\|^2 + \frac{\lambda_t^{(2)}}{12}\|x - v_t\|^3 \tag{68}$$

where $v_t = \arg\min_x \phi_t(x)$.

Proof. The first order condition on ϕ_t reads,

$$\ell_t' + \nabla\left(\frac{\lambda_t^{(1)}}{2}\|v_t - x_0\|^2 + \frac{\lambda_t^{(2)}}{6}\|v_t - x_0\|^3\right) = 0.$$

Multiplying both sides by $(x - v_t)$ gives

$$\ell_t'(x - v_t) + \nabla\left(\frac{\lambda_t^{(1)}}{2}\|v_t - x_0\|^2 + \frac{\lambda_t^{(2)}}{6}\|v_t - x_0\|^3\right)(x - v_t) = 0.$$

Note that, since ℓ_t is an affine function, $\ell_t'(x - v_t) = \ell_t(x) - \ell_t(v_t)$. Hence,

$$\ell_t(x) - \ell_t(v_t) + \nabla\left(\frac{\lambda_t^{(1)}}{2}\|v_t - x_0\|^2 + \frac{\lambda_t^{(2)}}{6}\|v_t - x_0\|^3\right)(x - v_t) = 0.$$

Finally, adding $\frac{\lambda_t^{(1)}}{2}\|x - x_0\|^2 + \frac{\lambda_t^{(2)}}{6}\|x - x_0\|^3$ on both sides and after reorganizing the terms,

$$\phi_t(x) = \ell_t(v_t) + \frac{\lambda_t^{(1)}}{2}\|x - x_0\|^2 + \frac{\lambda_t^{(2)}}{6}\|x - x_0\|^3 - \nabla\left(\frac{\lambda_t^{(1)}}{2}\|v_t - x_0\|^2 + \frac{\lambda_t^{(2)}}{6}\|v_t - x_0\|^3\right)(x - v_t). \tag{69}$$

From lemma 1 with $x = x - x_0$, $y = v_t - x_0$, and after reorganizing the terms,

$$\|x - x_0\|^i - \nabla(\|v_t - x_0\|^i)(x - v_t) \geq \frac{1}{2^{i-2}}\|x - v_t\|^i + \|v_t - x_0\|^i.$$

Therefore, using the previous inequality with $i = 2$ and $i = 3$, (69) becomes

$$\phi_t(x) \geq \ell_t(v_t) + \frac{\lambda_t^{(1)}}{2}\|v_t - x_0\|^2 + \frac{\lambda_t^{(2)}}{6}\|v_t - x_0\|^3 + \frac{\lambda_t^{(2)}}{2}\|v_t - x\|^2 + \frac{\lambda_t^{(3)}}{12}\|v_t - x\|^3$$

By definition of $\phi_t^\star = \phi_t(v_t)$,

$$\phi_t(x) \geq \phi_t^\star + \frac{\lambda_t^{(1)}}{2}\|v_t - x\|^2 + \frac{\lambda_t^{(2)}}{12}\|v_t - x\|^3.$$

□

Proposition 11. *Let*

$$\gamma = \frac{\kappa_D}{\|D\|}\left(\frac{3}{2}\|\varepsilon\| + 2\frac{\|(I - P)G\|}{M}\right).$$

Then, under the assumptions of proposition 4 the condition

$$\frac{\|f(x_+)\|^2}{M\left(\gamma + \|D\alpha\|\right)} \leq -\nabla f(x)^T D\alpha$$

is guaranteed as long as $M \geq 2L$.

Proof. The starting point is (53) combined with proposition 4:

$$\begin{aligned}\left\|\left(\frac{1}{2}D_\dagger \Gamma D_\dagger^T + \frac{M\|D\alpha\|}{2}\right)D\alpha + \nabla f(x_+)\right\| &\le \frac{L}{2}\|D\alpha\|^2 + \|D\alpha\|\left(\frac{L}{2}\frac{\|\varepsilon\|}{\|D\|}\kappa_D + \frac{\|(I-P)\nabla^2 f(x)D\alpha\|}{\|D\alpha\|}\right)\\ &\le \frac{L}{2}\|D\alpha\|^2 + \|D\alpha\|\left(\frac{L}{2}\frac{\|\varepsilon\|}{\|D\|}\kappa_D + (\|(I-P)G\| + L\|\varepsilon\|)\frac{\kappa_D}{\|D\|}\right)\\ &\le \frac{L}{2}\|D\alpha\|^2 + \|D\alpha\|\left(\frac{3L}{2}\frac{\|\varepsilon\|}{\|D\|}\kappa_D + |(I-P)G\|\frac{\kappa_D}{\|D\|}\right)\end{aligned}$$

To simplify, let $\Gamma = MD\gamma D^T$. Hence,

$$\left\|M\left(\frac{\|D\alpha\|+\gamma}{2}\right)D\alpha + \nabla f(x_+)\right\| \le \frac{L}{2}\|D\alpha\|^2 + \|D\alpha\|\left(\frac{3L}{2}\frac{\|\varepsilon\|}{\|D\|}\kappa_D + \|(I-P)G\|\frac{\kappa_D}{\|D\|}\right)$$

Elevating to the square this inequality gives

$$\begin{aligned}&\left(M\left(\frac{\gamma+\|D\alpha\|}{2}\right)\right)^2\|D\alpha\|^2 + \|\nabla f(x_+)\|^2 + 2\left(M\left(\frac{\gamma+\|D\alpha\|}{2}\right)\right)\nabla f(x_+)^T D\alpha\\ \le\ & \|D\alpha\|^2\left(\frac{L}{2}\|D\alpha\| + \frac{L}{2}\frac{\|\varepsilon\|}{\|D\|}\kappa_D + \frac{\|(I-P)\nabla^2 f(x)D\alpha\|}{\|D\alpha\|}\right)^2.\end{aligned}$$

The desired result holds if the following condition is satisfied,

$$\left(M\left(\frac{\gamma+\|D\alpha\|}{2}\right)\right)^2\|D\alpha\|^2 \ge \|D\alpha\|^2\left(\frac{L}{2}\|D\alpha\| + \frac{3L}{2}\frac{\|\varepsilon\|}{\|D\|}\kappa_D + \frac{\|(I-P)G\|\kappa_D}{\|D\|}\right)^2.$$

After simplification of the squares,

$$M\frac{\gamma+\|D\alpha\|}{2} \ge \frac{L}{2}\|D\alpha\| + \frac{3L}{2}\frac{\|\varepsilon\|}{\|D\|}\kappa_D + \frac{\|(I-P)G\|\kappa_D}{\|D\|}.$$

Replacing γ by its value gives

$$M\frac{\|D\alpha\| + \frac{\kappa_D}{\|D\|}\left(\frac{3}{2}\|\varepsilon\| + 2\frac{\|(I-P)G\|}{M}\right)}{2} \ge \frac{L}{2}\|D\alpha\| + \frac{3L}{2}\frac{\|\varepsilon\|}{\|D\|}\kappa_D + \frac{\|(I-P)G\|\kappa_D}{\|D\|}.$$

The condition is simplified into

$$(M-L)\frac{\|D\alpha\|}{2} + (M-2L)\frac{3}{2}\frac{\|\varepsilon\|\kappa_D}{\|D\|} \ge 0.$$

This condition is implied by $M \ge 2L$. □

Proposition 12. *Under the same assumptions as proposition 7, if $M \ge 2L$, and if*

$$\gamma = \frac{\kappa_D}{\|D\|}\left(\frac{3}{2}\|\varepsilon\| + 2\frac{\|(I-P)G\|}{M}\right) \le \frac{(\sqrt{3}-1)\|D\alpha\|}{4},$$

then

$$\frac{2}{3^{3/4}}\frac{\|\nabla f(x_+)\|^{3/2}}{\sqrt{M}} \le -\nabla f(x_+)^T D\alpha.$$

Proof. The starting point is (53),

$$\left\|M\frac{\|D\alpha\|}{2}D\alpha+\nabla f(x_+)\right\| \leq \frac{L}{2}\|D\alpha\|^2+\|D\alpha\|\left(\frac{L}{2}\frac{\kappa_D\|\varepsilon\|}{\|D\|}+\frac{M\gamma}{2}+\frac{\|(I-P)\nabla^2 f(x)D\alpha\|}{\|D\alpha\|}\right)$$

Therefore, to obtain

$$\left\|M\frac{\|D\alpha\|}{2}D\alpha+\nabla f(x_+)\right\| \leq M\left(\frac{\|D\alpha\|}{4}+\gamma\right)\|D\alpha\|,$$

The following is sufficient,

$$M\left(\frac{\|D\alpha\|}{4}+\gamma\right)\|D\alpha\| \geq \frac{L}{2}\|D\alpha\|^2+\|D\alpha\|\left(\frac{L}{2}\frac{\kappa_D\|\varepsilon\|}{\|D\|}+\frac{M\gamma}{2}+\frac{\|(I-P)\nabla^2 f(x)D\alpha\|}{\|D\alpha\|}\right).$$

Using proposition 4, the condition can be strengthened into

$$\begin{aligned}
&\frac{M}{2}\left(\frac{\|D\alpha\|+\gamma}{2}\right)\|D\alpha\| \\
&\geq \frac{L}{2}\|D\alpha\|^2+\|D\alpha\|\left(\frac{L}{2}\frac{\kappa_D\|\varepsilon\|}{\|D\|}+\frac{M\gamma}{2}+(\|(I-P)G\|+L\|\varepsilon\|)\frac{\kappa_D}{\|D\|}\right) \\
&= \frac{L}{2}\|D\alpha\|^2+\|D\alpha\|\left(\frac{3L}{2}\frac{\kappa_D\|\varepsilon\|}{\|D\|}+\frac{M\gamma}{2}+\|(I-P)G\|\frac{\kappa_D}{\|D\|}\right)
\end{aligned}$$

Defining

$$\frac{\gamma}{2}=\left(\frac{3}{4}\frac{\kappa_D\|\varepsilon\|}{\|D\|}+\frac{\|(I-P)G\|\frac{\kappa_D}{\|D\|}}{M}\right)$$

simplifies the condition into

$$M\left(\frac{\|D\alpha\|}{4}+\gamma\right)\|D\alpha\| \geq \frac{L}{2}\|D\alpha\|^2+\|D\alpha\|\left(M\gamma+\frac{3(L-\frac{M}{2})}{2}\frac{\kappa_D\|\varepsilon\|}{\|D\|}\right)$$

which is satisfied when $M>2L$. Now, assume that

$$\gamma \leq \frac{(\sqrt{3}-1)\|D\alpha\|}{4}.$$

Then,

$$\left\|M\frac{\|D\alpha\|}{2}D\alpha+\nabla f(x_+)\right\| \leq \sqrt{3}\frac{M\|D\alpha\|^2}{4}.$$

Elevating both sides to the square gives

$$\|\nabla f(x_+)\|^2+\frac{3M^2\|D\alpha\|^4}{16} \leq -M\|D\alpha\|\nabla f(x_+)^T D\alpha$$

Writing $r=\|D\alpha\|$,

$$\frac{\|\nabla f(x_+)\|^2}{Mr}+\frac{3Mr^3}{16} \leq -\nabla f(x_+)^T D\alpha.$$

Using

$$\frac{c_1}{r} + c_2 r^3 \geq 4 c_2^{1/4} \left(\frac{c_1}{3}\right)^{3/4},$$

the inequality becomes

$$\begin{aligned} -\nabla f(x_+)^T D\alpha &\geq \frac{M^{1/4}}{2} \frac{\|\nabla f(x_+)\|^{3/2}}{M^{3/4}} \frac{4}{3^{3/4}} \\ &= \frac{2}{3^{3/4}} \frac{\|\nabla f(x_+)\|^{3/2}}{\sqrt{M}}. \end{aligned}$$

□

Proposition 13 (Termination of algorithm 6)**.** *Let f satisfies Assumption 1. Assume that Requirements 1b to 3 holds. Then, once $M \geq 2L$, algorithm 6 terminates with* ***`ExitFlag`*** *equals to either* ***`SmallStep`*** *or* ***`LargeStep`****. Moreover, if $M_0 \leq L$, then the algorithm terminates with $M \leq 4L$. Moreover, if the algorithm terminates with* ***`ExitFlag`*** *equals to* ***`SmallStep`****, then*

$$\|D\alpha\| \leq \frac{4\gamma_M}{\sqrt{3}-1}, \quad \gamma_M = \frac{\kappa_D}{\|D\|}\left(\frac{3}{2}\|\varepsilon\| + 2\frac{\|(I-P)G\|}{M}\right).$$

Proof. Let

$$\gamma_M = \frac{\kappa_D}{\|D\|}\left(\frac{3}{2}\|\varepsilon\| + 2\frac{\|(I-P)G\|}{M}\right).$$

Assume that $M \geq 2L$. If $\gamma_M \leq \frac{(\sqrt{3}-1)\|D\alpha\|}{4}$, then, by proposition 12, the following condition is satisfied:

$$\frac{2}{3^{3/4}} \frac{\|\nabla f(x_+)\|^{3/2}}{\sqrt{M}} \leq -\nabla f(x_+)^T D\alpha.$$

In this case the algorithm terminates with `ExitFlag = LargeStep`. In any case, by proposition 11, the following conditions is always satisfied when $M \geq 2L$:

$$\frac{\|f(x_+)\|^2}{M\left(\gamma + \|D\alpha\|\right)} \leq -\nabla f(x)^T D\alpha.$$

Then, if $\gamma_M \geq \frac{(\sqrt{3}-1)\|D\alpha\|}{4}$, the algorithm terminates with `ExitFlag = SmallStep` (otherwise the algorithm would have been terminated with `ExitFlag = LargeStep`).

Since the algorithm doubles M until one of the two condition is satisfied, in the worst case, $M = 4L$. □

Proposition 14. *If $\lambda_t^{(1)}$ and $\lambda_t^{(2)}$ satisfy*

$$\lambda_t^{(1)} \geq \frac{b_{t+1}^2}{B_t} M_{t+1}\left(\gamma_t + \|D_t\alpha_t\|\right), \quad \lambda_t^{(2)} \geq \frac{4}{\sqrt{3}} \frac{b_{t+1}^3}{B_t^2} M_{t+1},$$

where $\gamma_t = \frac{\kappa_{D_t}}{\|D_t\|}\left(\frac{3}{2}\|\varepsilon_t\| + 2\frac{\|(I-P_t)G_t\|}{M_{t+1}}\right)$. Then, the function ϕ satisfies

$$B_t f(x_t) \leq \phi_t(x), \qquad \phi_t(x) \leq B_t f(x) + \frac{\lambda_t^{(1)} + \tilde{\lambda}^{(1)}}{2}\|x - x_0\|^2 + \frac{\lambda_t^{(2)} + \tilde{\lambda}^{(2)}}{6}\|x - x_0\|^3,$$

where

$$\tilde{\lambda}^{(1)} = \|\nabla f(x_0) - P_0 \nabla f(x_0) P_0\| + \delta\left(\frac{L\kappa + M_1\kappa^2}{2}\right), \quad \tilde{\lambda}^{(2)} = M_1 + L.$$

Proof. The result is proven by recursion. At $t=1$, the condition $B_t f(x_t) \leq \phi_t(x)$ is obviously satisfied since

$$f(x_1) \leq \min_v \phi_1(v) = f(x_1).$$

On the other hand, by proposition 8,

$$\begin{aligned} f(x_1) &\leq \min_x f(x) + \frac{\tilde{\lambda}^{(2)}}{6}\|x - x_0\|^3 + \frac{\tilde{\lambda}^{(1)}}{2}\|x - x_0\|^2 \\ &\leq f(x) + \frac{\tilde{\lambda}^{(2)}}{6}\|x - x_0\|^3 + \frac{\tilde{\lambda}^{(1)}}{2}\|x - x_0\|^2. \end{aligned}$$

Therefore, the second condition holds by definition of ϕ,

$$\begin{aligned} \phi_t &= f(x_1) + \frac{\lambda_t^{(1)}}{2}\|x - x_0\|^2 + \frac{\lambda_t^{(2)}}{6}\|x - x_0\|^3 \\ &\leq \frac{\lambda_1^{(1)} + \tilde{\lambda}^{(1)}}{2}\|x - x_0\|^2 + \frac{\lambda_1^{(2)} + \tilde{\lambda}^{(2)}}{6}\|x - x_0\|^3. \end{aligned}$$

Now, assume $t > 1$, and $B_t f(x_t) \leq \phi_t(x)$. Hence,

$$\begin{aligned} &\min_x \phi_{t+1}(x) \\ &= \min_x \ell_t(x) + b_t\left[f(x_{t+1}) + \nabla f(x_{t+1})(x - x_{t+1})\right] + \frac{\lambda_{t+1}^{(1)}}{2}\|x - x_0\|^2 + \frac{\lambda_{t+1}^{(2)}}{6}\|x - x_0\|^3 \\ &= \min_x \phi_t(x) + b_t\left[f(x_{t+1}) + \nabla f(x_{t+1})(x - x_{t+1})\right] \\ &\quad + \frac{\lambda_{t+1}^{(1)} - \lambda_t^{(1)}}{2}\|x - x_0\|^2 + \frac{\lambda_{t+1}^{(2)} - \lambda_t^{(2)}}{6}\|x - x_0\|^3 \\ &\geq \min_x \phi_t(x) + b_t\left[f(x_{t+1}) + \nabla f(x_{t+1})(x - x_{t+1})\right] \\ &\overset{(68)}{\geq} \min_x \phi_t^\star + \frac{\lambda_t^{(1)}}{2}\|x - v_t\|^2 + \frac{\lambda_t^{(2)}}{12}\|x - v_t\|^3 + b_t\left[f(x_{t+1}) + \nabla f(x_{t+1})(x - x_{t+1})\right] \\ &\geq \min_x B_t f(x_t) + \frac{\lambda_t^{(1)}}{2}\|x - v_t\|^2 + \frac{\lambda_t^{(2)}}{12}\|x - v_t\|^3 + b_t\left[f(x_{t+1}) + \nabla f(x_{t+1})(x - x_{t+1})\right] \\ &\overset{\text{A.4}}{\geq} \min_x B_t f(x_{t+1}) + \nabla f(x_{t+1})(x_t - x_{t+1}) + b_t\left[f(x_{t+1}) + \nabla f(x_{t+1})(x - x_{t+1})\right] \\ &\quad + \frac{\lambda_t^{(1)}}{2}\|x - v_t\|^2 + \frac{\lambda_t^{(2)}}{12}\|x - v_t\|^3 \\ &= \min_x B_{t+1} f(x_{t+1}) + \nabla f(x_{t+1})(B_t x_t + b_t x - B_{t+1} x_{t+1}) + \frac{\lambda_t^{(1)}}{2}\|x - v_t\|^2 + \frac{\lambda_t^{(2)}}{12}\|x - v_t\|^3 \\ &\overset{(67)}{=} \min_x B_{t+1} f(x_{t+1}) + B_{t+1}\nabla f(x_{t+1})(y_t - x_{t+1}) \\ &\quad + b_t \nabla f(x_{t+1})(x - v_t) + \frac{\lambda_t^{(1)}}{2}\|x - v_t\|^2 + \frac{\lambda_t^{(2)}}{12}\|x - v_t\|^3 \end{aligned}$$

The inequality is satisfied if either

$$\textbf{(a)} \quad 0 \leq B_{t+1}\nabla f(x_{t+1})(y_t - x_{t+1}) + b_t \nabla f(x_{t+1})(x - v_t) + \frac{\lambda_t^{(2)}}{12}\|x - v_t\|^3, \quad \text{or}$$

$$\textbf{(b)} \quad 0 \leq B_{t+1}\nabla f(x_{t+1})(y_t - x_{t+1}) + b_t \nabla f(x_{t+1})(x - v_t) + \frac{\lambda_t^{(1)}}{2}\|x - v_t\|^2.$$

It remains now to find *sufficient condition* such that one of the previous inequalities hold.

Define x_{t+1} to be the output of algorithm 6 starting from y_t, hence $y_t - x_{t+1} = -D_t\alpha_t$. The algorithm guarantees that

$$\textbf{(a)} \quad -\nabla f(x_{t+1})^T D_t \alpha_t \geq \frac{2}{3^{3/4}} \frac{\|\nabla f(x_{t+1})\|^{3/2}}{\sqrt{M_{t+1}}} \quad \text{and} \quad \text{or} \tag{70}$$

$$\textbf{(b)} \quad -\nabla f(x_{t+1})^T D_t \alpha_t \geq \frac{\|f(x_{t+1})\|^2}{M_{t+1}\left(\gamma_t + \|D_t\alpha_t\|\right)} \tag{71}$$

Combining the expressions **(a)** and **(b)** leads to the following sufficient conditions:

$$0 \leq B_{t+1}\frac{2}{3^{3/4}} \frac{\|\nabla f(x_{t+1})\|^{3/2}}{\sqrt{M_{t+1}}} + b_t \nabla f(x_{t+1})(x - v_t) + \frac{\lambda_t^{(2)}}{12}\|x - v_t\|^3, \tag{72}$$

$$0 \leq B_{t+1}\frac{\|f(x_{t+1})\|^2}{M_{t+1}\left(\gamma_t + \|D_t\alpha_t\|\right)} + b_t \nabla f(x_{t+1})(x - v_t) + \frac{\lambda_t^{(1)}}{2}\|x - v_t\|^2. \tag{73}$$

Case 1: equation (72). Starting from the first order condition of the minimum of (72) over x,

$$b_t \nabla f(x_{t+1}) + \frac{\lambda_t^{(2)}}{4}\|x - v_t\|(x - v_t) = 0. \tag{74}$$

Multiplying (74) by $(x - v_t)$ gives

$$b_t \nabla f(x_{t+1})(x - v_t) = -\frac{\lambda_t^{(2)}}{4}\|x - v_t\|^3$$

Hence, when x satisfies (74),

$$b_t \nabla f(x_{t+1})(x - v_t) + \frac{\lambda_t^{(2)}}{12}\|x - v_t\|^3 = -\frac{\lambda_t^{(2)}}{6}\|x - v_t\|^3. \tag{75}$$

Going back to (74), after isolating $x - v_t$,

$$(x - v_t) = -\frac{4b_t}{\lambda_t^{(2)}}\nabla f(x_{t+1})\frac{1}{\|x - v_t\|}$$

Therefore, after taking the norm and changing the power,

$$\begin{aligned}
\|x - v_t\|^3 &= \left(\frac{4b_t}{\lambda_t^{(2)}}\|\nabla f(x_{t+1})\|\right)^{3/2}, \\
\Leftrightarrow \frac{\lambda_t^{(2)}}{6}\|x - v_t\|^3 &= \frac{\lambda_t^{(2)}}{6}\left(\frac{4b_t}{\lambda_t^{(2)}}\|\nabla f(x_{t+1})\|\right)^{3/2} \\
&= \frac{4}{3\sqrt{\lambda_t^{(2)}}}\left(b_t\|\nabla f(x_{t+1})\|\right)^{3/2}.
\end{aligned}$$

After using (75) and injecting the minimal value makes the condition (72) stronger:

$$0 \leq B_{t+1}\frac{2}{3^{3/4}}\frac{\|\nabla f(x_{t+1})\|^{3/2}}{\sqrt{M_{t+1}}} - \frac{4}{3\sqrt{\lambda_t^{(2)}}}\left(b_t\|\nabla f(x_{t+1})\|\right)^{3/2}.$$

Hence, if $\lambda_t^{(2)}$ satisfies

$$B_{t+1}\frac{2}{3^{3/4}\sqrt{M_{t+1}}} \geq \frac{4}{3\sqrt{\lambda_t^{(2)}}}b_t^{(3/2)} \quad \Leftrightarrow \quad \lambda_t^{(2)} \geq \frac{4}{\sqrt{3}}\frac{b_t^3}{B_{t+1}^2}M_{t+1}, \tag{76}$$

then (72) is satisfied.

Case 2: equation (73). Starting from the first order condition of the minimum of (73) over x,

$$b_{t+1}\nabla f(x_{t+1}) + \lambda_t^{(1)}(x - v_t). \tag{77}$$

Hence,

$$(x - v_t) = -\frac{b_t\nabla f(x_{t+1})}{\lambda_t^{(1)}}.$$

Injecting the value back in (73) gives

$$B_{t+1}\frac{\|f(x_{t+1})\|^2}{M\left(\gamma_t + \|D_t\alpha_t\|\right)} - b_t^2\frac{\|\nabla f(x_{t+1})\|^2}{\lambda_t^{(1)}} + \frac{1}{2}b_t^2\frac{\|\nabla f(x_{t+1})\|^2}{\lambda_t^{(1)}}.$$

Therefore, if the following condition holds,

$$\frac{B_{t+1}}{2M_{t+1}\left(\gamma_t + \|D_t\alpha_t\|\right)} \geq \frac{b_t^2}{\lambda_t^{(1)}} \quad \Leftrightarrow \quad \lambda_t^{(1)} \geq \frac{b_t^2}{2B_{t+1}}M_{t+1}\left(\gamma_t + \|D_t\alpha_t\|\right),$$

then (73) is satisfied.

□

Proposition 15. *Let f satisfies Assumption 1. Then, under Requirements 1b to 3, $\lambda_t^{(1)}$ and $\lambda_t^{(2)}$ in algorithm 7 are bounded by*

$$\lambda_t^{(1)} \leq 30 \cdot \frac{b_{t+1}^2}{B_t}\kappa_D\left(\delta\max\{4L, M_0\} + \max_{i=0\dots t}\|(I - P_i)\nabla f(x_i)P_i)\|\right) \tag{78}$$

$$\lambda_t^{(2)} \leq \frac{L}{2}\delta + \max_{i=0\dots t}\|(I - P_i)\nabla f(x_i)P_i)\|. \tag{79}$$

Proof. Since algorithm 7 doubles $\lambda_t^{(1)}$, $\lambda_t^{(2)}$ until $\phi_t^\star \geq f(x_{t+1})$, then by proposition 14, both $\lambda_t^{(1)}$, $\lambda_t^{(2)}$ achieves at most

$$\lambda_t^{(1)} \leq 2 \cdot \frac{b_{t+1}^2}{B_t}M_{t+1}\left(\gamma_t + \|D_t\alpha_t\|\right), \quad \lambda_t^{(2)} \leq 2 \cdot \frac{4}{\sqrt{3}}\frac{b_{t+1}^3}{B_t^2}M_{t+1}.$$

There are three cases to distinguish:

1. The algorithm finishes with `ExitFlag = LargeStep`,
2. The algorithm finishes with `ExitFlag = SmallStep`.

Case 1. In this case, $\lambda_{t+1}^{(2)}$ may be updated. By proposition proposition 13, $M_t \leq 4L$ (unless $M_0 \geq 4L$). Hence, $\lambda_t^{(2)}$ is bounded by

$$\lambda_t^{(2)} \leq 2 \cdot \frac{4}{\sqrt{3}} \frac{b_{t+1}^3}{B_t^2} \max\{M_0, 4L\} \leq 5 \frac{b_{t+1}^3}{B_t^2} \max\{M_0, 4L\}.$$

Case 2. In this case, $\lambda_{t+1}^{(1)}$ may be updated. By proposition 13, and by Requirements 2 and 3,

$$\begin{aligned} M_{t+1}\left(\gamma_t + \|D_t\alpha_t\|\right) &\leq \frac{\sqrt{3}+1}{\sqrt{3}-1} M_{t+1}\gamma_t \\ &= \frac{\sqrt{3}+1}{\sqrt{3}-1} \frac{\kappa_{D_t}}{\|D_t\|}\left(\frac{3}{2}\|\varepsilon_t\| M_{t+1} + 2\|(I-P_t)G_t\|\right), \\ &\leq \frac{\sqrt{3}+1}{\sqrt{3}-1}\left(\frac{3}{2}\delta\kappa_D \max\{4L, M_0\} + 2\kappa_D \frac{\|(I-P_t)G_t\|}{\|D_t\|}\right). \end{aligned}$$

In addition, by Theorem 6 and Requirement 2,

$$\begin{aligned} \frac{\|(I-P_t)G_t\|}{\|D_t\|} &\leq \frac{\|(I-P_t)(G_t - \nabla f(x_t)D_t)\| + \|(I-P_t)\nabla f(x_t)D_t)\|}{\|D_t\|} \\ &\leq \frac{\frac{L}{2}\|\varepsilon_t\| + \|(I-P_t)\nabla f(x_t)D_t)\|}{\|D_t\|}, \\ &= \frac{\frac{L}{2}\|\varepsilon_t\| + \|(I-P_t)\nabla f(x_t)P_tD_t)\|}{\|D_t\|}, \\ &\leq \frac{L}{2}\delta + \max_{i=0\dots t}\|(I-P_i)\nabla f(x_i)P_i)\|. \end{aligned}$$

Hence,

$$\begin{aligned} &M_{t+1}\left(\gamma_t + \|D_t\alpha_t\|\right) \\ &\leq \frac{\sqrt{3}+1}{\sqrt{3}-1}\left(\frac{3}{2}\delta\kappa_D \max\{4L, M_0\} + 2\kappa_D\left(\frac{L}{2}\delta + \max_{i=0\dots t}\|(I-P_i)\nabla f(x_i)P_i)\|\right)\right), \\ &\leq \frac{\sqrt{3}+1}{\sqrt{3}-1}\left(\frac{7}{4}\delta\kappa_D \max\{4L, M_0\} + 2\kappa_D \max_{i=0\dots t}\|(I-P_i)\nabla f(x_i)P_i)\|\right). \\ &\leq 7.5\kappa_D\left(\delta \max\{4L, M_0\} + \max_{i=0\dots t}\|(I-P_i)\nabla f(x_i)P_i)\|\right). \end{aligned}$$

Therefore,

$$\begin{aligned} \lambda_t^{(1)} &\leq 2 \cdot \frac{b_{t+1}^2}{B_t} M_{t+1}\left(\gamma_t + \|D_t\alpha_t\|\right) \\ &\leq 30 \cdot \frac{b_{t+1}^2}{B_t}\kappa_D\left(\delta \max\{4L, M_0\} + \max_{i=0\dots t}\|(I-P_i)\nabla f(x_i)P_i)\|\right) \end{aligned}$$

□

H.5. Missing proofs from Sections A and 3

Theorem 6. *Let the function f satisfy Assumption 1. Let the matrices D, G be defined as in (10) and vector ε as in (11). Then, for all $w \in \mathbb{R}^d$ and $\alpha \in \mathbb{R}^N$*

$$-\tfrac{L\|w\|}{2}|\alpha|^T\varepsilon_t \le w^T(\nabla^2 f(x)D_t - G_t)\alpha \le \tfrac{L\|w\|}{2}|\alpha|^T\varepsilon_t, \tag{12}$$

$$\|w^T(\nabla^2 f(x)D_t - G_t)\| \le \tfrac{L\|w\|}{2}\|\varepsilon_t\|. \tag{13}$$

Proof. Using Cauchy-Schwartz with (2) gives that, for all v,

$$v^T\left(\nabla f(y) - \nabla f(z) - \nabla^2 f(z)(y-z)\right) \le \frac{L\|v\|}{2}\|y-z\|^2.$$

Let $v = v_i$, $y = y_i$, and $z = z_i$. By the definition of Y, Z, D, G in (10),

$$v_i^T\left(g_i - \nabla^2 f(z_i)d_i\right) \le \frac{L\|v_i\|}{2}\|d_i\|^2.$$

Introducing $\nabla^2 f(x)$ gives

$$v_i^T\left(g_i - \nabla^2 f(z_i)d_i\right) = v_i^T\left(g_i - \nabla^2 f(x)d_i\right) + v_i^T(\nabla^2 f(z_i) - \nabla^2 f(x))d_i.$$

Since the Hessian is L-Lipchitz-continuous Assumption 1, $(\nabla^2 f(z_i) - \nabla^2 f(x))d_i \le L\|d_i\|\|z_i - x\|$. Therefore, by the definition of ε_i,

$$v_i^T\left(g_i - \nabla^2 f(x)d_i\right) \le \frac{L\|v_i\|\varepsilon_i}{2}. \tag{80}$$

Let $v_i = \text{sign}(\alpha_i)w$. Summing all inequalities multiplied by $|\alpha_i|$ gives the first desired result:

$$w^T\left(G - \nabla^2 f(x)D\right)\alpha \le \frac{L\|w\|\sum_{i=1}^N \varepsilon_i|\alpha_i|}{2}.$$

The second result is rather straightforward, since (80) with $v_i = w$ gives

$$w^T\left(g_i - \nabla^2 f(x)d_i\right) \le \frac{L\|w\|\varepsilon_i}{2}.$$

Therefore,

$$\sqrt{\sum_{i=1}^N \left(w^T\left(g_i - \nabla^2 f(x)d_i\right)\right)^2} \le \|w\|\sqrt{\sum_{i=1}^N \|g_i - \nabla^2 f(x)d_i\|^2} \le \|w\|\sqrt{\sum_{i=1}^N L\varepsilon_i^2} \le \frac{L\|w\|\|\varepsilon\|}{2}.$$

□

Theorem 7. *Let the function f satisfy Assumption 1. Let x_{t+1} be defined as in (8), the matrices D_t, G_t be defined as in (10) and ε_t be defined as in (11). Then, for all $\alpha \in \mathbb{R}^N$,*

$$f(x_{t+1}) \le f(x_t) + \nabla f(x_t)^T D_t\alpha + \tfrac{\alpha^T H_t \alpha}{2} + \tfrac{L\|D_t\alpha\|^3}{6}, \tag{Type-I bound}$$

$$\|\nabla f(x_{t+1})\| \le \|\nabla f(x_t) + G_t\alpha\| + \tfrac{L}{2}\left(|\alpha|^T\varepsilon_t + \|D_t\alpha\|^2\right), \tag{Type-II bound}$$

where $H_t \overset{def}{=} \frac{G_t^T D_t + D_t^T G_t + \mathrm{I}L\|D_t\|\|\varepsilon_t\|}{2}$.

Proof. The inequality (Type-II bound) is a direct consequence of (2) (with $y = x_+$, $z = x$) combined with (13),

$$
\begin{aligned}
& \|\nabla f(x_+) - \nabla f(x) - \nabla^2 f(x) D\alpha\| \leq \frac{L}{2}\|D\alpha\|^2 \\
\Leftrightarrow\ & w^T \left(\nabla f(x_+) - \nabla f(x) - \nabla^2 f(x) D\alpha\right) \leq \frac{L\|w\|}{2}\|D\alpha\|^2 \\
\Leftrightarrow\ & w^T \nabla f(x_+) \leq \frac{L\|w\|}{2}\|D\alpha\|^2 + w^T\left(\nabla f(x) + \nabla^2 f(x) D\alpha\right) \\
\Leftrightarrow\ & w^T \nabla f(x_+) \overset{(12)}{\leq} \frac{L\|w\|}{2}\left(\|D\alpha\|^2 + \sum_{i=1}^N |\alpha_i|\varepsilon_i\right) + w^T\left(\nabla f(x) + G\alpha\right) \\
\Leftrightarrow\ & w^T \nabla f(x_+) \leq \|w\|\left(\frac{L}{2}\left(\|D\alpha\|^2 + \sum_{i=1}^N |\alpha_i|\varepsilon_i\right) + \|\nabla f(x) + G\alpha\|\right)
\end{aligned}
$$

Setting $w = \nabla f(x_+)$ gives (Type-II bound).

The inequality (Type-I bound) instead comes from (3) combined with (13). Indeed,

$$
\begin{aligned}
f(x_+) &\leq f(x) + \nabla f(x) D\alpha + \frac{1}{2}(D\alpha)^T \nabla^2 f(x)(D\alpha) + \frac{L}{6}\|D\alpha\|^3 \\
&\overset{(13)}{\leq} f(x) + \nabla f(x) D\alpha + \frac{1}{2}\left((D\alpha)^T G\alpha + \frac{L\|D\alpha\|}{2}\sum_{i=1}^N |\alpha_i|\varepsilon_i\right) + \frac{L}{6}\|D\alpha\|^3
\end{aligned}
$$

It remains to use the followings bounds:

$$
\sum_{i=1}^N |\alpha_i|\varepsilon_i = \alpha^T(\text{sign}(\alpha)\odot\varepsilon) \leq \|\alpha\|\|\varepsilon\|,
$$

$$
\|D\alpha\| \leq \|D\|\|\alpha\|.
$$

All together,

$$
f(x_+) \leq f(x) + \nabla f(x) D\alpha + \frac{1}{2}(D\alpha)^T G\alpha + \frac{L}{4}\|\alpha\|^2\|D\|\|\varepsilon\| + \frac{L}{6}\|D\alpha\|^3
$$

Finally, since $(D\alpha)^T G\alpha$ is a quadratic form, only the symmetric counterpart of $D^T G$ counts. That means, $(D\alpha)^T G\alpha = \alpha^T \frac{D^T G + G^T D}{2}\alpha$. Hence, writing $H = \frac{D^T G + G^T D}{2} + \mathrm{I}\frac{L}{2}\|D\|\|\varepsilon\|$ gives the desired result,

$$
f(x_+) \leq f(x) + \nabla f(x) D\alpha + \frac{\alpha^T H\alpha}{2} + \frac{L}{6}\|D\alpha\|^3.
$$

□

Theorem 1. *Let f satisfy Assumption 1. Then, at each iteration $t \geq 0$, algorithm 3 achieves*

$$
f(x_{t+1}) \leq f(x_t) - \tfrac{M_{t+1}}{12}\|x_{t+1} - x_t\|^3, \quad M_{t+1} < \max\left\{2L\ ;\ \tfrac{M_0}{2^t}\right\}. \tag{7}
$$

Proof. Using (43), at each iteration, after the while loop, the first-order condition of the subroutine algorithm 4 reads

$$D_t^T \nabla f(x_t) + H_t \alpha_{t+1} + \frac{M_{t+1}}{2} D_t^T D_t \alpha_{t+1} \|D_t \alpha_{t+1}\| = 0. \tag{81}$$

The subscript t is dropped for clarity. After multiplying by α,

$$\nabla f(x_t)^T D\alpha + \alpha^T H \alpha + \frac{M}{2}\|D\alpha\|^3 = 0.$$

In addition, multiplying both times by α the second-order condition (44) gives

$$\alpha^T H \alpha \geq -\frac{M}{2}\|D\alpha\|^3.$$

which gives, after replacing it in (81),

$$\nabla f(x_t)^T D\alpha \leq -\frac{M}{2}\|D\alpha\|^3 + \frac{M}{2}\|D\alpha\|^3 = 0. \tag{82}$$

Injecting eqs. (81) and (82) into the while condition of algorithm 4 gives the desired result:

$$\begin{aligned} f(x_+) &\leq f(x) + \nabla f(x)^T D\alpha + \frac{1}{2}\alpha^T H \alpha + \frac{M\|D\alpha\|^3}{6}, \\ &= f(x) - \frac{1}{2}\nabla f(x)^T D\alpha - \frac{M\|D\alpha\|^3}{12} \\ &\leq f(x) - \frac{M\|D\alpha\|^3}{12}. \end{aligned} \tag{83}$$

Where (83) is guaranteed if $M > L$. Therefore, in the worst case, $M < 2L$. Finally, after t iterations, the number of total gradient calls is bounded by $2t + \log_2\left(\frac{M_0}{L}\right)$ as shown in [52]. □

Theorem 2. *Let f satisfy Assumption 1, and assume that f is bounded below by f^*. Let Requirements 1b to 3 hold, and $M_t \geq M_{\min}$. Then, algorithm 3 starting at x_0 with M_0 achieves*

$$\min_{i=1,\dots,t} \|\nabla f(x_i)\| \leq \max\left\{ \frac{3L}{t^{2/3}}\left(12\frac{f(x_0)-f^\star}{M_{\min}}\right)^{2/3} \; ; \; \left(\frac{C_1}{t^{1/3}}\right)\left(12\frac{f(x_0)-f^\star}{M_{\min}}\right)^{1/3} \right\},$$

where $C_1 = \delta L\left(\frac{\kappa+2\kappa^2}{2}\right) + \max_{i\in[0,t]} \|(I-P_i)\nabla^2 f(x_i) P_i\|$.

Proof. The starting inequality is (54):

$$\|\nabla f(x_+)\| \leq \frac{L+M}{2}\|D\alpha\|^2 + \|D\alpha\|\left(\frac{\|\varepsilon\|}{\|D\|}\left(\frac{L+M\kappa_D}{2}\right)\kappa_D + \|(I-P)\nabla^2 f(x) P\|\right).$$

The result is obtained by decomposing the inequality using a maximum,

$$\begin{aligned} &\|\nabla f(x_+)\| \\ &\leq \max\left\{(L+M)\|D\alpha\|^2 \; ; \; 2\|D\alpha\|\left(\frac{\|\varepsilon\|}{\|D\|}\left(\frac{L+M\kappa_D}{2}\right)\kappa_D + \|(I-P)\nabla^2 f(x) P\|\right)\right\}. \end{aligned}$$

In the first case,

$$\|D\alpha\| \geq \sqrt{\frac{\|\nabla f(x_+)\|}{L+M}}, \tag{84}$$

while in the second case,

$$\|D\alpha\| \geq \frac{\|\nabla f(x_+)\|}{\frac{\|\varepsilon\|}{\|D\|}\left(\frac{L+M\kappa_D}{2}\right)\kappa_D + \|(I-P)\nabla^2 f(x)P\|}.$$

Let C_t be defined as

$$C_t = \frac{\|\varepsilon_t\|}{\|D_t\|}\left(\frac{L+M_{t+1}\kappa_{D_t}}{2}\right)\kappa_{D_t} + \|(I-P_t)\nabla^2 f(x_t)P_t\|.$$

Then, using Requirements 2 and 3, and since $M < 2L$ by Theorem 1,

$$C_t \leq C = \delta L\left(\frac{1+2\kappa}{2}\right)\kappa + \max_t \|(I-P_t)\nabla^2 f(x_t)P_t\|$$

Therefore,

$$\|D\alpha\| \geq \frac{\|\nabla f(x_+)\|}{C}. \tag{85}$$

At each iteration t, combining eqs. (84) and (85) into Theorem 1 gives

$$f(x_t) - f(x_{t+1}) \geq \frac{M_{t+1}}{12}\|\underbrace{x_{t+1}-x_t}_{=D_t\alpha_t}\|^3 \geq \frac{M_{t+1}}{12}\min\left\{\left(\frac{\|\nabla f(x_+)\|}{L+M_{t+1}}\right)^{3/2} \;;\; \left(\frac{\|\nabla f(x_+)\|}{C}\right)^3\right\}$$

Therefore,

$$\begin{aligned}
f(x_0) - f^\star &\geq f(x_0) - f(x_t) \\
&= \sum_{i=0}^{t-1} f(x_i) - f(x_{i+1}) \\
&\geq \sum_{i=0}^{t-1}\left(\frac{M_{i+1}}{12}\|x_{i+1}-x_i\|^3\right) \\
&\geq \sum_{i=0}^{t-1}\min_t \frac{M_{i+1}}{12}\left\{\left(\frac{\|\nabla f(x_{i+1})\|}{L+M_{i+1}}\right)^{3/2} \;;\; \left(\frac{\|\nabla f(x_{i+1})\|}{C}\right)^3\right\} \\
&\geq t\min_{i\in[0,t-1]}\frac{M_{i+1}}{12}\min\left\{\left(\frac{\|\nabla f(x_{i+1})\|}{L+M_{i+1}}\right)^{3/2} \;;\; \left(\frac{\|\nabla f(x_{i+1})\|}{C}\right)^3\right\} \\
&\geq t\frac{M_{\min}}{12}\min\left\{\min_{i\in[1,t]}\left(\frac{\|\nabla f(x_i)\|}{3L}\right)^{3/2} \;;\; \min_{i\in[1,t]}\left(\frac{\|\nabla f(x_i)\|}{C}\right)^3\right\}
\end{aligned}$$

After analyzing separately each case of the minimum, either

$$\left(\frac{\min_{i\in[1,t]}\|\nabla f(x_i)\|}{3L}\right)^{3/2} \leq 12\frac{f(x_0)-f^\star}{tM_{\min}} \quad \text{or} \quad \left(\frac{\min_{i\in[1,t]}\|\nabla f(x_{t+1})\|}{C}\right)^3 \leq 12\frac{f(x_0)-f^\star}{tM_{\min}}.$$

It remains to simplify to obtain the desired result,

$$\min_{i=1\ldots t} \|\nabla f(x_i)\| \leq \max\left\{ \frac{3L}{t^{2/3}}\left(12\frac{f(x_0)-f^\star}{M_{\min}}\right)^{2/3} \; ; \; \left(\frac{C}{t^{1/3}}\right)\left(12\frac{f(x_0)-f^\star}{M_{\min}}\right)^{1/3} \right\}.$$

□

Theorem 3. *Assume f satisfy Assumptions 1 to 3. Let Requirements 1b to 3 hold. Then, algorithm 3 starting at x_0 with M_0 achieves, for $t \geq 1$,*

$$f(x_t) - f^\star \leq 6\frac{f(x_0)-f^\star}{t(t+1)(t+2)} + \frac{1}{(t+1)(t+2)}\frac{L(3R)^3}{2} + \frac{1}{t+2}\frac{C_2(3R)^2}{4},$$

where $C_2 \overset{def}{=} \delta L\frac{\kappa+2\kappa^2}{2} + \max_{i\in[0,t]} \|\nabla^2 f(x_i) - P_i\nabla^2 f(x_i)P_i\|.$

Proof. Starting from the inequality in proposition 8,

$$f(x_{t+1}) \leq f(y) + \frac{M_{t+1}+L}{6}\|y-x_t\|^3 + \frac{\|y-x_t\|^2}{2}C_2^{(t)},$$

where

$$C_2^{(t)} = \|\nabla^2 f(x_t) - P_t\nabla^2 f(x_t)P_t\| + \delta\frac{L\kappa + M_{t+1}\kappa^2}{2},$$

and setting $y = (1-\beta_t)x_t + \beta_t x^\star$ and $f(x^\star) = f^\star$ gives

$$f(x_{t+1}) - f^\star \leq f((1-\beta_t)x_t + \beta_t x^\star) - f^\star + \frac{M_{t+1}+L}{6}\beta_t^3\|x_t-x^\star\|^3 + \frac{\beta_t^2\|x_t-x^\star\|^2}{2}C_2^{(t)}.$$

Because the function is star-convex,

$$f(x_{t+1}) - f^\star \leq (1-\beta_t)(f(x_t)-f^\star) + \frac{M_{t+1}+L}{6}\beta_t^3\|x_t-x^\star\|^3 + \frac{\beta_t^2\|x_t-x^\star\|^2}{2}C_2^{(t)}.$$

Since algorithm 4 ensure a decrease in the function value, the iterate x_t satisfies

$$x_t \in \{x : f(x \leq f(x_0))\},$$

and therefore, $\|x_t - x^\star\| \leq R$ by Assumption 2. In addition, $M < 2L$ by Theorem 1. The inequality now becomes

$$(f(x_{t+1}) - f^\star) \leq (1-\beta_t)(f(x_t)-f^\star) + \beta_t^3\frac{LR^3}{2} + \beta_t^2\frac{R^2C_2^{(t)}}{2}. \tag{86}$$

Finally, since $M < 2L$, the scalar C_2^t is bounded over time by C_2:

$$C_2^{(t)} \leq C_2 \overset{\text{def}}{=} \delta L\frac{\kappa+2\kappa^2}{2} + \max_t \|\nabla^2 f(x_t) - P_t\nabla^2 f(x_t)P_t\|.$$

Now, let

- $B_t = \frac{t(t+1)(t+2)}{6}$,

- $b_t : B_t = B_{t-1} + b_t$, hence $b_t = \frac{t(t+1)}{2}$, and
- $\beta_t = \frac{b_{t+1}}{B_{t+1}}$.

Therefore, for $t \geq 1$,

$$1 = \frac{B_t}{B_t} = \frac{B_{t-1}}{B_t} + \frac{b_t}{B_t} = \frac{B_{t-1}}{B_t} + \beta_{t-1} \quad \Rightarrow \quad 1 - \beta_{t-1} = \frac{B_{t-1}}{B_t}.$$

Injecting those relations in (86) gives

$$(f(x_{t+1}) - f^\star) \leq \frac{B_t}{B_{t+1}}(f(x_t) - f^\star) + \left(\frac{b_{t+1}}{B_{t+1}}\right)^3 \frac{LR^3}{2} + \left(\frac{b_{t+1}}{B_{t+1}}\right)^2 \frac{R^2 C_2}{2},$$

hence the recursion

$$\begin{aligned} B_{t+1}(f(x_{t+1}) - f^\star) &\leq B_t(f(x_t) - f^\star) + \frac{b_{t+1}^3}{B_{t+1}^2}\frac{LR^3}{2} + \frac{b_{t+1}^2}{B_{t+1}}\frac{R^2C_2}{2} \\ &\leq B_0(f(x_t) - f^\star) + \sum_{i=0}^{t} \frac{b_{i+1}^3}{B_{i+1}^2}\frac{LR^3}{2} + \sum_{i=0}^{t}\frac{b_{i+1}^2}{B_{i+1}}\frac{R^2C_2}{2}. \end{aligned}$$

$$(f(x_{t+1}) - f^\star) \leq \frac{B_0}{B_{t+1}}(f(x_t) - f^\star) + \frac{\sum_{i=0}^t \frac{b_{i+1}^3}{B_{i+1}^2}}{B_{t+1}}\frac{LR^3}{2} + \frac{\sum_{i=0}^t \frac{b_{i+1}^2}{B_{i+1}}}{B_{t+1}}\frac{R^2C_2}{2}.$$

Therefore, the rate reads By the definition of b_t and B_t,

$$\begin{aligned} \frac{b_{i+1}^3}{B_{i+1}^2} &= \frac{36}{8}\frac{(i+1)^3(i+2)^3}{(i+1)^2(i+2)^2(i+3)^2} = \frac{9}{2}\frac{(i+1)(i+2)}{(i+3)^2} \leq \frac{9}{2}, \\ \frac{b_{i+1}^2}{B_{i+1}} &= \frac{6}{4}\frac{(i+1)^2(i+2)^2}{(i+1)(i+2)(i+3)} = \frac{3}{2}\frac{(i+2)}{(i+3)}(i+1) \leq \frac{3}{2}(i+1). \end{aligned}$$

Hence,

$$\begin{aligned} \frac{\sum_{i=0}^t \frac{b_{i+1}^3}{B_{i+1}^2}}{B_{t+1}} &\leq \frac{\frac{9}{2}(t+1)}{\frac{(t+1)(t+2)(t+3)}{6}} \leq \frac{27}{(t+2)(t+3)}, \\ \frac{\sum_{i=0}^t \frac{b_{i+1}^2}{B_{i+1}}}{B_{t+1}} &\leq \frac{\sum_{i=0}^t \frac{3}{2}(i+1)}{\frac{(t+1)(t+2)(t+3)}{6}} = \frac{\frac{3}{4}(t+2)(t+1)}{\frac{(t+1)(t+2)(t+3)}{6}} = \frac{9}{2(t+3)}. \end{aligned}$$

Shifting from $t+1$ tp t gives the desired result,

$$(f(x_t) - f^\star) \leq 6\frac{f(x_t) - f^\star}{t(t+1)(t+2)} + \frac{1}{(t+1)(t+2)}\frac{L(3R)^3}{2} + \frac{1}{t+2}\frac{C_2(3R)^2}{4}.$$

□

Theorem 4. *Assume f satisfy Assumptions 1, 2 and 4. Let Requirements 1a, 2 and 3 hold. Then, in expectation over the matrices D_i, algorithm 3 starting at x_0 with M_0 achieves, for $t \geq 1$,*

$$\mathbb{E}_{D_t}[f(x_t) - f^\star] \leq \frac{1}{1 + \frac{1}{4}\left[\frac{N}{d}t\right]^3}(f(x_0) - f^\star) + \frac{1}{\left[\frac{N}{d}t\right]^2}\frac{L(3R)^3}{2} + \frac{1}{\left[\frac{N}{d}t\right]}\frac{C_3(3R)^2}{2},$$

where $C_3 \overset{def}{=} \delta L \frac{\kappa + 2\kappa^2}{2} + \frac{(d-N)}{d} \max_{i \in [0,t]} \|\nabla^2 f(x_i)\|.$

Proof. The proof technique is similar to [39]. Starting from proposition 9 with $x = x_t$,

$$\begin{aligned}\mathbb{E}f(x_{t+1}) &\leq \left(1 - \frac{N}{d}\right) f(x_t) + \frac{N}{d} f(y) + \frac{N}{d}\frac{(M_{t+1} + L)}{6}\|y - x_t\|^3 \\ &\quad + \frac{N}{d}\frac{\|y - x_t\|^2}{2}\left(\delta\frac{L\kappa + M_{t+1}\kappa^2}{2} + \frac{(d-N)}{d}\|\nabla^2 f(x_t)\|\right),\end{aligned}$$

where the expectation is taken with $D_0, \ldots, D_{t-1}$ fixed. Using the inequality $M_{t+1} \leq 2L$ gives

$$\mathbb{E}f(x_{t+1}) \leq \left(1 - \frac{N}{d}\right) f(x_t) + \frac{N}{d}\left(f(y) + \frac{\|y - x_t\|^2}{2}C_3 + \frac{L}{2}\|y - x_t\|^3\right)$$

where

$$C_3 \overset{\text{def}}{=} \left(\delta L \frac{\kappa + 2\kappa^2}{2} + \frac{(d-N)}{d} \max_{i \in [0,t]} \|\nabla^2 f(x_i)\|\right).$$

Let $y = \beta_t x^\star + (1 - \beta_t)x_t$, $\beta_t \in [0, 1]$. After using Assumption 4 and Assumption 2,

$$\begin{aligned}\mathbb{E}f(x_{t+1}) &\leq \left(1 - \frac{N}{d}\right) f(x_t) + \frac{N}{d}\left(f\Big(\beta_t x^\star + (1 - \beta_t)x_t\Big) + \beta_t^2\frac{C_3 R^2}{2} + \beta_t^3\frac{LR^3}{2}\right) \\ &\leq \left(1 - \frac{N}{d}\right) f(x_t) + \frac{N}{d}\left(\beta_t f(x^\star) + (1 - \beta_t)f(x_t) + \beta_t^2\frac{C_3 R^2}{2} + \beta_t^3\frac{LR^3}{2}\right) \\ &= \left(1 - \frac{N}{d}\right) f(x_t) + \frac{N}{d}\left(\beta_t f(x^\star) + (1 - \beta_t)f(x_t) + \beta_t^2\frac{C_3 R^2}{2} + \beta_t^3\frac{LR^3}{2}\right), \\ &= \left(1 - \beta_t\frac{N}{d}\right) f(x_t) + \frac{N}{d}\left(\beta_t f(x^\star) + \beta_t^2\frac{C_3 R^2}{2} + \beta_t^3\frac{LR^3}{2}\right).\end{aligned}$$

Hence, the recursion

$$(\mathbb{E}f(x_{t+1}) - f^\star) \leq \left(1 - \beta_t\frac{N}{d}\right)(f(x_t) - f^\star) + \frac{N}{d}\left(\beta_t^2\frac{C_3 R^2}{2} + \beta_t^3\frac{LR^3}{2}\right).$$

Now, define

$$\begin{aligned}&b_t = t^2, \\ &B_t = B_0 + \sum_{i=0}^{t} b_i, \quad B_0 = \frac{4}{3}\left(\frac{d}{N}\right)^3 \\ &\beta_t = \frac{d}{N}\frac{b_{t+1}}{B_{t+1}} \quad \Rightarrow \quad 1 - \frac{N}{d}\beta_t = \frac{B_t}{B_{t+1}}.\end{aligned}$$

Replacing those relations in the recursion gives

$$\begin{aligned}
& B_{t+1}\left(\mathbb{E}f(x_{t+1})-f^\star\right) \\
\leq & B_t(f(x_t)-f^\star)+\frac{N}{dB_{t+1}}\left(\left(\frac{d}{N}\frac{b_{t+1}}{B_{t+1}}\right)^2\frac{C_3R^2}{2}+\left(\frac{d}{N}\frac{b_{t+1}}{B_{t+1}}\right)^3\frac{LR^3}{2}\right) \\
= & B_t(f(x_t)-f^\star)+\frac{d}{N}\frac{b_{t+1}^2}{B_{t+1}}\frac{C_3R^2}{2}+\frac{d^2}{N^2}\frac{b_{t+1}^3}{B_{t+1}^2}\frac{LR^3}{2}
\end{aligned}$$

Expanding the inequality gives

$$B_{t+1}\left(\mathbb{E}f(x_{t+1})-f^\star\right)\leq B_0(f(x_0)-f^\star)+\frac{d}{N}\sum_{t=0}^{t+1}\frac{b_{i+1}^2}{B_{i+1}}\frac{C_3R^2}{2}+\frac{d^2}{N^2}\sum_{t=0}^{t+1}\frac{b_{i+1}^3}{B_{i+1}^2}\frac{LR^3}{2}$$

Since

$$\begin{aligned}
B_t = B_0+\sum_{i=1}^{t} & \geq B_0+\int_0^t x^2\mathrm{d}x = B_0+\frac{t^3}{3} \\
\sum_{i=0}^{t}\frac{b_t^2}{B_t} & \leq \sum_{i=0}^{t}\frac{i^4}{B_0+i^3/3}\leq 3t^2, \\
\sum_{i=0}^{t}\frac{b_t^3}{B_t^2} & \leq \sum_{i=0}^{t}\frac{i^6}{(B_0+i^3/3)^2}\leq 9t,
\end{aligned}$$

the bound becomes

$$B_{t+1}\left(\mathbb{E}f(x_{t+1})-f^\star\right)\leq B_0(f(x_0)-f^\star)+\frac{d}{N}3t^2\frac{C_3R^2}{2}+\frac{d^2}{N^2}9t\frac{LR^3}{2}$$

Dividing both sides by B_{t+1} gives

$$\mathbb{E}f(x_{t+1})-f^\star\leq\frac{B_0}{B_0+\frac{(t+1)^3}{3}}(f(x_0)-f^\star)+\frac{d}{N}\frac{3(t+1)^2}{B_0+\frac{(t+1)^3}{3}}\frac{C_3R^2}{2}+\frac{d^2}{N^2}\frac{9(t+1)}{B_0+\frac{(t+1)^3}{3}}\frac{LR^3}{2}.$$

After the following simplifications,

$$\begin{aligned}
\frac{B_0}{B_0+(t+1)^3/3} &= \frac{1}{1+\frac{(t+1)^3}{3B_0}}=\frac{1}{1+\frac{1}{4}\left(\frac{N}{d}(t+1)\right)^3}, \\
\frac{3(t+1)^2}{B_0+(t+1)^3/3} &= \frac{3}{B_0}\frac{(t+1)^3}{1+\frac{(t+1)^3}{3B_0}}\frac{1}{t+1}\leq\frac{3}{B_0}3B_0\frac{1}{t+1}=\frac{9}{t+1}, \\
\frac{9(t+1)}{B_0+\frac{(t+1)^3}{3}} &= \frac{9}{B_0}\frac{(t+1)^3}{\frac{(t+1)^3}{3B_0}}\frac{1}{(t+1)^2}\leq\frac{9}{B_0}3B_0\frac{1}{(t+1)^2}=\frac{27}{(t+1)^2},
\end{aligned}$$

the inequality finally becomes (after shifting from $t+1$ to t),

$$\mathbb{E}f(x_t)-f^\star\leq\frac{1}{1+\frac{1}{4}\left[\frac{N}{d}t\right]^3}(f(x_0)-f^\star)+\frac{1}{\left[\frac{N}{d}t\right]^2}\frac{L(3R)^3}{2}+\frac{1}{\left[\frac{N}{d}t\right]}\frac{C_3(3R)^2}{2}.$$

□

Theorem 5. *Assume f satisfy Assumptions 1, 2 and 4. Let Requirements 1b to 3 hold. Then, the accelerated algorithm 7 starting at x_0 with M_0 achieves, for $t \geq 1$,*

$$f(x_t) - f^\star \leq C_4 \tfrac{(3R)^2}{(t+3)^2} + 9\max\{M_0\ ;\ 2L\}\left(\tfrac{3R}{t+3}\right)^3 + \tfrac{\frac{\tilde{\lambda}^{(1)}R^2}{2} + \frac{\tilde{\lambda}^{(2)}R^3}{6}}{(t+1)^3}.$$
$$\text{where } \tilde{\lambda}^{(1)} = 0.5\cdot\delta\left(L\kappa + M_1\kappa^2\right) + \|\nabla^2 f(x_0) - P_0\nabla^2 f(x_0)P_0\|, \qquad \tilde{\lambda}^{(2)} = M_1 + L,$$
$$C_4 = 30\cdot\kappa_D\left(\delta\max\{4L, M_0\} + \max_{i=0\dots t}\|(I-P_i)\nabla f(x_i)P_i\|\right)$$

Proof. By construction of $\phi_t(x)$, from proposition 14 and Assumption 2,

$$B_t f(x_t) \leq \min_x \phi_t(x) \tag{87}$$
$$\leq \phi_t(x^\star) \tag{88}$$
$$\leq B_t f(x^\star) + \frac{\lambda_t^{(1)} + \tilde{\lambda}^{(1)}}{2}\|x^\star - x_0\|^2 + \frac{\lambda_t^{(2)} + \tilde{\lambda}^{(2)}}{6}\|x^\star - x_0\|^3 \tag{89}$$
$$\leq B_t f(x^\star) + \frac{\lambda_t^{(1)} + \tilde{\lambda}^{(1)}}{2}R^2 + \frac{\lambda_t^{(2)} + \tilde{\lambda}^{(2)}}{6}R^3 \tag{90}$$
$$\Rightarrow f(x_t) - f^\star \leq \frac{\lambda_t^{(1)} + \tilde{\lambda}^{(1)}}{2B_t}R^2 + \frac{\lambda_t^{(2)} + \tilde{\lambda}^{(2)}}{6B_t}R^3. \tag{91}$$

By proposition 15, the following bounds holds:

$$\lambda_t^{(1)} \leq 30\cdot\frac{b_{t+1}^2}{B_t}\kappa_D\left(\delta\max\{4L, M_0\} + \max_{i=0\dots t}\|(I-P_i)\nabla f(x_i)P_i\|\right),$$
$$\lambda_t^{(2)} \leq 5\frac{b_{t+1}^3}{B_t^2}\max\{M_0, 4L\}.$$

Since $\frac{b_{t+1}}{B_t} = \frac{3}{(t+3)}$,

$$\frac{b_{t+1}^3}{B_t^3} = \frac{3^3}{(t+3)^3}, \qquad \frac{b_{t+1}^2}{B_t^2} = \frac{3^2}{(t+3)^2}. \tag{92}$$

Therefore,

$$f(x_t) - f^\star \leq 30\cdot\kappa_D\left(\delta\max\{4L, M_0\} + \max_{i=0\dots t}\|(I-P_i)\nabla f(x_i)P_i\|\right)\frac{(3R)^2}{(t+3)^2}$$
$$+ 5\max\{M_0, 4L\}\left(\frac{3R}{t+3}\right)^3$$
$$+ \frac{\frac{\tilde{\lambda}^{(1)}R^2}{2} + \frac{\tilde{\lambda}^{(2)}R^3}{6}}{(t+1)^3}.$$

□