First-Order Objective-Function-Free Optimization Algorithms and Their Complexity

S. Gratton*, S. Jerad† and Ph. L. Toint‡

6 III 2022

Abstract

A class of algorithms for unconstrained nonconvex optimization is considered where the value of the objective function is never computed. The class contains a deterministic version of the first-order Adagrad method typically used for minimization of noisy function, but also allows the use of second-order information when available. The rate of convergence of methods in the class is analyzed and is shown to be identical to that known for first-order optimization methods using both function and gradients values. The result is essentially sharp and improves on previously known complexity bounds (in the stochastic context) by Défossez *et al.* (2020) and Gratton *et al.* (2022). A new class of methods is designed, for which a slightly worse and essentially sharp complexity result holds. Limited numerical experiments show that the new methods' performance may be comparable to that of standard steepest descent, despite using significantly less information, and that this performance is relatively insensitive to noise.

Keywords: First-order methods, objective-function-free optimization (OFFO), Adagrad, convergence bounds, evaluation complexity, second-order models.

1 Introduction

This paper is concerned with Objective-Function-Free Optimization (OFFO[(1)]) algorithms, which we define as numerical optimization methods in which *the value of the problem's objective function is never calculated*, although we obviously assume that it exists. This is clearly at variance with the large majority of available numerical optimization algorithms, where the objective function is typically evaluated at every iteration, and its value then used to assess progress towards a minimizer and (often) to enforce descent. Dispensing with this information is therefore challenging. As it turns out, first-order OFFO methods (i.e. OFFO methods only using gradients) already exist for some time and have proved popular and useful in fields such

*Université de Toulouse, INP, IRIT, Toulouse, France. Email: serge.gratton@enseeiht.fr. Work partially supported by 3IA Artificial and Natural Intelligence Toulouse Institute (ANITI), French "Investing for the Future - PIA3" program under the Grant agreement ANR-19-PI3A-0004"

†ANITI, Université de Toulouse, INP, IRIT, Toulouse, France. Email: sadok.jerad@enseeiht.fr

‡NAXYS, University of Namur, Namur, Belgium. Email: philippe.toint@unamur.be. Partially supported by ANITI.

(1)This term is coined in contrast to DFO, a well-known acronym for Derivative Free Optimization. OFFO can be viewed as the complement of DFO, since the latter only uses objective function values and no derivatives, while the former only uses derivatives and no objective function values.

as machine learning or sparse optimization, thereby justifying our interest. Algorithms such Adagrad [12], RMSprop [27], Adam [20] or AMSgrad [26] have been proposed and analyzed, recent contributions being [10] and [30] where improved convergence bounds for such methods are discussed. We refer the reader to the interesting paper [10] for a more extensive historical perspective on the developement of convergence theory of these first-order OFFO algorithms.

We take here the point of view that, although methods such as Adagrad and Adam have been defined and used in the context of probabilistic inexact gradient evaluations (such as resulting from sampling in finite-sum problems), they may neverthless be of interest in a deterministic context (i.e. when gradients are computed exactly). Indeed, in addition to being one of the very few existing approaches to OFFO, their deterministic evaluation complexity analysis provides a very useful template for that of their probabilistic counterparts. In particular, lower complexity bounds for the deterministic case immediately apply to the stochastic one. The purpose of the present contribution is thus to explore the deterministic context, the probablistic approach (using linear models only) being covered in a companion paper [17].

Should one be interested in first-order deterministic objective-function-free optimization, another obvious alternative would be to minimize the norm of the gradient (in the least-squares sense or in some other norm) using a derivative-free algorithm. To the best of our knowledge, this approach has not been widely experimented, maybe because it has the drawback of not being biased towards minimizing the underlying objective function: standard DFO algorithms, such as trust-region or adaptive regularization methods using finite-difference approximations, make, in this context, no distinction between minimizers, maximizers and other first-orders points of the original problem. More importantly, convergence of the DFO minimizer would have to occur to a global minimum for the original gradient to approximately vanish. The evaluation complexity of this approach is nevertheless appealing under these strong assumptions, because it was proved in [4] (see also [15] and [6, Section 13.1]) that reaching an ϵ-approximate first-order point (of the gradient's norm) can be obtained in $\mathcal{O}(\epsilon^{-2} + |\log(\epsilon)|)$ evaluations[(2)]. We will compare below this theoretical bound with the new OFFO complexity results obtained in this paper.

Contributions: In this paper, we

1. interpret the deterministic counterpart of Adagrad as a first-order trust-region method and use this interpretation to extend it to potentially use second-order information,
2. provide, for this extended method, an essentially sharp global[(3)] bound on the gradient's norm as a function of the iteration counter, which is identical to that known for first-order optimization methods using both function and gradients values,
3. use the developed framework to define a further class of first-order OFFO methods, for which an essentially sharp complexity result is also provided,
4. present some numerical experiments indicating that OFFO methods may indeed be competitive with steepest descent in efficiency and reliability.

Note: When this paper was completed, the authors became aware of the paper [16], where an adaptive trust-region algorithm is proposed in which the objective function is not evaluated,

[(2)] As is standard for two real positive sequences $\{a_k\}$ and $\{b_k\}$, we say that $a_k = \mathcal{O}(b_k)$ if and only if $\lim_{k\to\infty}(a_k/b_k)$ is finite.

[(3)] I.e., valid at every iteration.

with a global rate of convergence similar to that presented below. However, while this algorithm is also a member of the class we study here, it is motivated differently (no relation to Adagrad is mentioned in [16]) and involves very different theoretical arguments (in particular, the question of sharpness for the complexity bound is not considered).

The paper is organized as follows. Section 2 introduces the new "trust-region minded" class of algorithms ASTR1 and the global rate of convergence of a subclass containing the deterministic Adagrad method are studied in Section 3. Section 4 then introduces a new ASTR1 subclass and analyzes its global rate of convergence. Some numerical illlustration is provided in Section 5. Conclusions are finally outlined in Section 6.

2 A class of first-order minimization methods

We consider the problem

$$\min_{x\in\mathbb{R}^n} f(x) \tag{2.1}$$

where f is a smooth function from $\mathbb{R}^n$ to $\mathbb{R}$. In particular, we will assume in what follows that

AS.1: the objective function $f(x)$ is continuously differentiable;

AS.2: its gradient $g(x) \stackrel{\rm def}{=} \nabla^1_x f(x)$ is Lipschitz continuous with Lipschitz constant $L \geq 0$, that is

$$\|g(x) - g(y)\| \leq L\|x - y\|$$

for all $x, y \in \mathbb{R}^n$, where, unless otherwise specified, $\|\cdot\| = \|\cdot\|_2$ is the Euclidean norm;

AS.3: there exists a constant $\kappa_g \geq 1$ such that, for all x, $\|g(x)\|_\infty \leq \kappa_g$;

AS.4: there exists a constant $f_{\rm low}$ such that, for all x, $f(x) \geq f_{\rm low}$.

AS.1, AS.2 and AS.4 are standard for the complexity analysis of optimization methods seeking first-order critical points, AS.4 guaranteeing in particular that the problem is well-posed. AS.3 is common in the analysis of stochastic OFFO methods (see [28, 10, 30, 17], for instance).

The class of methods of interest here are iterative and generate a sequence of iterates $\{x_k\}_{k\geq 0}$. The move from an iterate to the next directly depends on the gradient at x_k and algorithm-dependent *scaling factors* $\{w_k = w(x_0, \ldots, x_k)\}$ whose main purpose is to control the move's magnitude. In our analysis, we will assume that

AS.5: for each $i \in \{1, \ldots, n\}$ there exists a constant $\varsigma_i \in (0, 1]$ such that, $w_{i,k} \geq \varsigma_i$ for all $k \geq 0$.

Since scaling factors are designed to control the length of the step, they are strongly reminiscent of the standard mechanism of the much studied trust-region optimization methods (see [8] for an extensive coverage and [29] for a more recent survey). In trust-region algorithms, a model of the objective function at an iterate x_k is built, typically using a truncated Taylor series, and a step s_k is chosen that minimizes this model with a *trust-region*, that is a region where the model is assumed to represent the true objective function sufficiently well. This region is a ball around the current iterate, whose radius is updated adaptively from iteration to iteration, based on the quality of the prediction of the objective function value at the trial

point $x_k + s_k$. For methods using gradient only, the model is then chosen as the first two terms of the Taylor's expansion of f at the iterate x_k. Although, we are interested here in methods where the objective function's value is not evaluated, and therefore cannot be used to accept/reject iterates and update the trust-region radius, a similar mechanism may be designed, this time involving the scaling factors $\{w_k\}$. The resulting algorithm, which we call ASTR1 (for Adaptively Scaled Trust Region using 1rst order information) is stated on this page.

Algorithm 2.1: ASTR1

Step 0: Initialization. A starting point x_0 is given. Constants $\kappa_{\rm B} \geq 1$ and $\tau \in (0, 1]$ are also given. Set $k = 0$.

Step 1: Define the TR. Compute $g_k = g(x_k)$ and define

$$\Delta_{i,k} = \frac{|g_{i,k}|}{w_{i,k}} \tag{2.2}$$

where $w_k = w(x_0, \ldots, x_k)$.

Step 2: Hessian approximation. Select a symmetric Hessian approximation B_k such that

$$\|B_k\| \leq \kappa_{\rm B}. \tag{2.3}$$

Step 3: GCP. Compute a step s_k such that

$$|s_{i,k}| \leq \Delta_{i,k} \quad (i \in \{1, \ldots, n\}), \tag{2.4}$$

and

$$g_k^T s_k + \tfrac{1}{2} s_k^T B_k s_k \leq \tau \left(g_k^T s_k^Q + \tfrac{1}{2} (s_k^Q)^T B_k s_k^Q \right), \tag{2.5}$$

where

$$s_{i,k}^L = -\mathrm{sgn}(g_{i,k}) \Delta_{i,k}, \tag{2.6}$$

$$s_k^Q = \gamma_k s_k^L, \tag{2.7}$$

with

$$\gamma_k = \begin{cases} \min \left[1, \dfrac{|g_k^T s_k^L|}{(s_k^L)^T B_k s_k^L} \right] & \text{if} \quad (s_k^L)^T B_k s_k^L > 0, \\ 1 & \text{otherwise.} \end{cases} \tag{2.8}$$

Step 4: New iterate. Define

$$x_{k+1} = x_k + s_k, \tag{2.9}$$

increment k by one and return to Step 1.

The algorithm description calls for some comments.

1. Observe that we allow the use of second-order information by effectively defining a quadratic model
$$g_k^T s + \tfrac{1}{2} s^T B_k s \tag{2.10}$$
where B_k can of course be chosen as the true second-derivative matrix of f at x_k (provided it remains bounded to satisfy (2.3)) or any approximation thereof. Choosing $B_k = 0$ results in a purely first-order algorithm.

   The condition (2.3) on the Hessian approximations is quite weak, and allows in particular for a variety of quasi-Newton approaches, limited-memory or otherwise. In a finite-sum context, sampling bounded Hessians is also possible.

2. Conditions (2.5)–(2.8) define a "generalized Cauchy point", much in the spirit of standard trust-region methodology (see [8, Section 6.3] for instance), where the quadratic model (2.10) is minimized in (2.8) along a good first-order direction (s_k^L) to obtain a "Cauchy step" s_k^Q. Any step s_k can then be accepted provided it remains in the trust region (see (2.4)) and enforces a decrease in the quadratic model which is a least a fraction τ of that achieved at the Cauchy step (see (2.5)).

3. At variance with many existing trust-region algorithms, the radius Δ_k of the trust-region (2.2) is not recurred adaptively from iteration to iteration depending on how well the quadratic model predicts function values, but is directly defined as a scaled version of the local gradient. This is not without similarities with the trust-region method proposed by [13], which corresponds to a scaling factor equal to $\|g_k\|^{-1}$.

4. As stated, the ASTR1 algorithm does not include a termination rule, but such a rule can easily be introduced by terminating the algorithm in Step 1 if $\|g_k\| \leq \epsilon$, where $\epsilon > 0$ is a user-defined first-order accuracy threshold.

The algorithm being defined, the first step of our analysis is to derive a fundamental property of objective-function decrease, valid for all choices of the scaling factors satisfying AS.5.

Lemma 2.1 Suppose that AS.1, AS.2 and AS.5 hold. Then we have that, for all $k \geq 0$,
$$f(x_0) - f(x_{k+1}) \geq \sum_{j=0}^{k} \sum_{i=1}^{n} \frac{g_{i,j}^2}{2\kappa_{\rm B} w_{i,j}} \left[\tau \varsigma_{\min} - \frac{\kappa_{\rm BBL}}{w_{i,j}} \right] \tag{2.11}$$
where $\varsigma_{\min} \stackrel{\rm def}{=} \min_{i \in \{1,\ldots,n\}} \varsigma_i$ and $\kappa_{\rm BBL} \stackrel{\rm def}{=} \kappa_{\rm B}(\kappa_{\rm B} + L)$.

Proof. Using (2.6) and AS.5, we deduce that, for every $j \geq 0$,
$$|g_j^T s_j^L| = \sum_{i=1}^{n} \frac{g_{i,j}^2}{w_{i,j}} = \sum_{i=1}^{n} \frac{w_{i,j} g_{i,j}^2}{w_{i,j}^2} \geq \sum_{i=1}^{n} \frac{\varsigma_i g_{i,j}^2}{w_{i,j}^2} \geq \varsigma_{\min} \|s_j^L\|^2. \tag{2.12}$$
Suppose first that $(s_j^L)^T B_j s_j^L > 0$ and $\gamma_j < 1$. Then, in view of (2.7), (2.8), (2.12) and

(2.3),

$$g_j^T s_j^Q + \tfrac{1}{2}(s_j^Q)^T B_j s_j^Q = \gamma_j g_j^T s_j^L + \tfrac{1}{2}\gamma_j^2 (s_j^L)^T B_j s_j^L = -\frac{(g_j^T s_j^L)^2}{2(s_j^L)^T B_j s_j^L} \leq -\frac{\varsigma_{\min}|g_j^T s_j^L|}{2\kappa_{\rm B}}.$$

Combining this inequality with the first equality in (2.12) then gives that

$$g_j^T s_j^Q + \tfrac{1}{2}(s_j^Q)^T B_j s_j^Q \leq -\frac{\varsigma_{\min}}{2\kappa_{\rm B}} \sum_{i=1}^{n} \frac{g_{i,j}^2}{w_{i,j}}. \tag{2.13}$$

Suppose now that $(s_j^L)^T B_j s_j^L \leq 0$ or $\gamma_j = 1$. Then, using (2.7), (2.13) and (2.6),

$$g_j^T s_j^Q + \tfrac{1}{2}(s_j^Q)^T B_j s_j^Q = g_j^T s_j^L + \tfrac{1}{2}(s_j^L)^T B_j s_j^L \leq \tfrac{1}{2} g_j^T s_j^L < 0$$

and (2.13) then again follows from the bound $\kappa_{\rm B} \geq 1$. Successively using AS.1–AS.2, (2.5), (2.13), (2.3) and (2.2) then gives that, for $j \geq 0$,

$$\begin{aligned} f(x_{j+1}) &\leq f(x_j) + g_j^T s_j + \tfrac{1}{2} s_j^T B_j s_j - \tfrac{1}{2} s_j^T B_j s_j + \tfrac{1}{2} L \|s_j\|^2 \\ &\leq f(x_j) + \tau \left(g_j^T s_j^Q + \tfrac{1}{2}(s_j^Q)^T B_j s_j^Q \right) + \tfrac{1}{2}(\kappa_{\rm B} + L)\|s_j\|^2 \\ &\leq f(x_j) - \sum_{i=1}^{n} \frac{\tau \varsigma_{\min} g_{i,j}^2}{2\kappa_{\rm B} w_{i,j}} + \tfrac{1}{2}(\kappa_{\rm B} + L) \sum_{i=1}^{n} \Delta_{i,j}^2 \\ &\leq f(x_j) - \sum_{i=1}^{n} \frac{\tau \varsigma_{\min} g_{i,j}^2}{2\kappa_{\rm B} w_{i,j}} + \tfrac{1}{2}(\kappa_{\rm B} + L) \sum_{i=1}^{n} \frac{g_{i,j}^2}{w_{i,j}^2} \end{aligned} \tag{2.14}$$

Summing up this inequality for $j \in \{0, \ldots, k\}$ then yields (2.11). □

Armed with Lemma 2.1, we are now in position to specify particular choices of the scaling factors $w_{i,k}$ and derive the convergence properties of the resulting variants of ASTR1.

3 An Adagrad-like variant of ASTR1 using second-order models

We first consider a choice of scaling factors directly derived from the definition of the Adagrad algorithm [12]. For given $\varsigma \in (0, 1]$, $\vartheta \in (0, 1]$ and $\mu \in (0, 1)$, define, for all $i \in \{1, \ldots, n\}$ and for all $k \geq 0$,

$$w_{i,k} \in \left[\sqrt{\vartheta}\, v_{i,k}, v_{i,k} \right] \quad \text{where} \quad v_{i,k} \stackrel{\rm def}{=} \left(\varsigma + \sum_{\ell=0}^{k} g_{i,\ell}^2 \right)^{\mu}. \tag{3.1}$$

The Adagrad scaling factors are recovered by $\mu = \frac{1}{2}$ and $\vartheta = 1$, and ASTR1 with (3.1) and $B_k = 0$ is then the standard (deterministic) Adagrad method. The ϑ parameter is introduced for felxibility, in particular allowing non-monotone scaling factors. Without loss of generality, we assume that $\kappa_g^2 \geq g_{i,0}^2 + \varsigma$ for $i \in \{1, \ldots, n\}$.

Before stating the global rate of convergence of the variant of ASTR1 using (3.1), we first prove a crucial lemma, partly inspired by [28, 10].

Lemma 3.1 Let $\{a_k\}_{k\geq 0}$ be a non-negative sequence, $\alpha > 0$, $\xi > 0$ and define, for each $k \geq 0$, $b_k = \sum_{j=0}^{k} a_j$. Then if $\alpha \neq 1$,

$$\sum_{j=0}^{k} \frac{a_j}{(\xi + b_j)^\alpha} \leq \frac{1}{(1-\alpha)}((\xi + b_k)^{1-\alpha} - \xi^{1-\alpha}). \tag{3.2}$$

Otherwise (i.e. if $\alpha = 1$),

$$\sum_{j=0}^{k} \frac{a_j}{(\xi + b_j)} \leq \log\left(\frac{\xi + b_k}{\xi}\right). \tag{3.3}$$

Proof. Consider first the case where $\alpha \neq 1$ and note that $\frac{1}{(1-\alpha)}x^{1-\alpha}$ is then a non-decreasing and concave function on $(0, +\infty)$. Setting $b_{-1} = 0$ and using these properties, we obtain that, for $j \geq 0$,

$$\begin{aligned}\frac{a_j}{(\xi + b_j)^\alpha} &\leq \frac{1}{1-\alpha}\left((\xi + b_j)^{1-\alpha} - (\xi + b_j - a_j)^{1-\alpha}\right) \\ &\leq \frac{1}{1-\alpha}\left((\xi + b_j)^{1-\alpha} - (\xi + b_{j-1})^{1-\alpha}\right).\end{aligned}$$

We then obtain (3.2) by summing this inequality for $j \in \{0, \ldots, k\}$.

Suppose now that $\alpha = 1$, we then use the concavity and non-decreasing character of the logarithm to derive that

$$\frac{a_j}{(\xi + b_j)^\alpha} = \frac{a_j}{(\xi + b_j)} \leq \log(\xi + b_j) - \log(\xi + b_j - a_j) \leq \log(\xi + b_j) - \log(\xi + b_{j-1}).$$

The inequality (3.3) then again follows by summing for $j \in \{0, \ldots, k\}$. □

From (3.2), we also obtain that, for $\alpha < 1$,

$$\sum_{j=0}^{k} \frac{a_j}{(\xi + b_j)^\alpha} \leq \frac{1}{(1-\alpha)}(\xi + b_k)^{1-\alpha} \tag{3.4}$$

while, for $\alpha > 1$,

$$\sum_{j=0}^{k} \frac{a_j}{(\xi + b_j)^\alpha} \leq \frac{\xi^{1-\alpha}}{(\alpha - 1)}. \tag{3.5}$$

Note that both the numerator and the denominator of the right-hand side of (3.2) tend to zero when α tends to one. Applying l'Hospital rule, we then see that this right-hand side tends to the right-hand side of (3.3) and the bounds on $\sum_{j=0}^{k} a_j/(\xi + b_j)^\alpha$ are therefore continuous at $\alpha = 1$.

Theorem 3.2 Suppose that AS.1–AS.4 hold and that the ASTR1 algorithm is applied to problem (2.1) with its scaling given by (3.1). If we define

$$\Gamma_0 \stackrel{\rm def}{=} f(x_0) - f_{\rm low}, \quad \kappa_1 \stackrel{\rm def}{=} \frac{\kappa_g^{2\mu}\kappa_{\rm B}}{\tau\varsigma^{\mu}\sqrt{\vartheta}} \quad \text{and} \quad \kappa_2 \stackrel{\rm def}{=} \frac{n\kappa_1(\kappa_{\rm B}+L)}{\vartheta},$$

then,

(i) if $0 < \mu < \frac{1}{2}$,

$$\operatorname*{average}_{j\in\{0,\ldots,k\}} \|g_j\|^2 \leq \min\left[\frac{\kappa_3}{k+1},\ \frac{\kappa_4}{(k+1)^{1-\mu}} + \frac{\kappa_5}{(k+1)^{\mu}}\right] \tag{3.6}$$

with

$$\kappa_3 = \max\left\{\varsigma,\ \left[\frac{4\,n\,\kappa_{\rm BBL}}{(1-2\mu)\tau\varsigma^{\mu}\vartheta^{\frac{3}{2}}}\right]^{\frac{1}{\mu}},\ \left[\frac{2^{2\mu}\vartheta(1-2\mu)\Gamma_0}{n(\kappa_{\rm B}+L)}\right]^{\frac{1}{1-2\mu}}\right\}, \tag{3.7}$$

$$\kappa_4 = 2\kappa_1\Gamma_0 \quad \text{and} \quad \kappa_5 = \left(\frac{\kappa_2}{1-2\mu}\right)\left[(\varsigma+\kappa_g^2)^{1-2\mu} - \varsigma^{1-2\mu}\right];$$

(ii) if $\mu = \frac{1}{2}$,

$$\operatorname*{average}_{j\in\{0,\ldots,k\}} \|g_j\|^2 \leq \min\left[\frac{\kappa_6}{k+1}, \frac{\kappa_4}{\sqrt{k+1}} + \kappa_2\,\frac{\log\left(1+(k+1)\varsigma^{-1}\kappa_g^2\right)}{\sqrt{k+1}}\right] \tag{3.8}$$

with

$$\kappa_6 = \max\left\{\varsigma, \frac{1}{2}e^{\frac{2\Gamma_0\vartheta}{n(\kappa_B+L)}}, \frac{1}{2}\left(\frac{8n\kappa_{\rm B}(\kappa_{\rm B}+L)}{\tau\sqrt{\varsigma}\vartheta^{\frac{3}{2}}}\right)^2\left|W_{-1}\left(-\frac{\tau\sqrt{\varsigma}\vartheta^{\frac{3}{2}}}{8n\kappa_{\rm B}(\kappa_{\rm B}+L)}\right)\right|^2\right\}, \tag{3.9}$$

where W_{-1} is the second branch of the Lambert function [9];

(iii) if $\frac{1}{2} < \mu < 1$,

$$\operatorname*{average}_{j\in\{0,\ldots,k\}} \|g_j\|^2 \leq \min\left[\frac{\kappa_7}{k+1}, \frac{\kappa_8}{(k+1)^{1-\mu}}\right] \tag{3.10}$$

with

$$\kappa_7 \stackrel{\rm def}{=} \left[\frac{2^{1+\mu}\kappa_{\rm B}}{\tau\varsigma^{\mu}\sqrt{\vartheta}}\left(\Gamma_0 + \frac{n(\kappa_{\rm B}+L)\varsigma^{1-2\mu}}{2\vartheta(2\mu-1)}\right)\right]^{\frac{1}{1-\mu}} \text{and} \quad \kappa_8 \stackrel{\rm def}{=} 2\kappa_1\Gamma_0 + \left(\frac{\kappa_2\varsigma^{1-2\mu}}{2\mu-1}\right). \tag{3.11}$$

Proof. We see from (3.1) that $w_{i,k}$ verifies **AS.5**. We may thus use Lemma 2.1.

Moreover, (3.1) and the fact that $\kappa_g^2 \geq g_{i,0}^2 + \varsigma$ for $i \in \{1, \ldots, n\}$ also imply that

$$\varsigma^\mu \sqrt{\vartheta} \leq w_{i,j} \leq \min\left[\kappa_g^{2\mu}(j+1)^\mu, \left(\varsigma + \sum_{\ell=0}^{j} \|g_\ell\|^2\right)^\mu\right] \tag{3.12}$$

for all $j \geq 0$ and all $i \in \{1, \ldots, n\}$. We now deduce from (2.2) and (2.11) that, for $k \geq 0$,

$$f(x_{k+1}) \;\; \leq \;\; f(x_0) - \sum_{j=0}^{k} \frac{\tau \varsigma^\mu \sqrt{\vartheta}\, \|g_j\|^2}{2\kappa_{\rm B} \max\limits_{i \in \{1,\ldots,n\}} w_{i,k}} + \tfrac{1}{2}(\kappa_{\rm B} + L) \sum_{i=1}^{n} \sum_{j=0}^{k} \Delta_{i,j}^2. \tag{3.13}$$

For each $i \in \{1, \ldots, n\}$, we then apply Lemma 3.1 with $a_\ell = g_{i,\ell}^2$, $\xi = \varsigma$ and $\alpha = 2\mu < 1$, and obtain from (2.2) and (3.1) that,

$$\sum_{j=0}^{k} \Delta_{i,j}^2 \leq \frac{1}{\vartheta(1-2\mu)} \left[\left(\varsigma + \sum_{\ell=0}^{k} g_{i,\ell}^2\right)^{1-2\mu} - \varsigma^{1-2\mu}\right] \leq \frac{1}{\vartheta(1-2\mu)} \left(\varsigma + \sum_{\ell=0}^{k} g_{i,\ell}^2\right)^{1-2\mu}. \tag{3.14}$$

Now

$$\sum_{i=1}^{n} \sum_{j=0}^{k} \Delta_{i,j}^2 \leq \sum_{i=1}^{n} \frac{1}{\vartheta(1-2\mu)} \left(\varsigma + \sum_{i=1}^{n} \sum_{\ell=0}^{k} g_{i,\ell}^2\right)^{1-2\mu} \leq \frac{n}{\vartheta(1-2\mu)} \left(\varsigma + \sum_{\ell=0}^{k} \|g_\ell\|^2\right)^{1-2\mu} \tag{3.15}$$

and thus substituting this bound in (3.13) and using AS.4 gives that

$$\sum_{j=0}^{k} \frac{\tau \varsigma^\mu \sqrt{\vartheta} \|g_j\|^2}{2\kappa_{\rm B} \max\limits_{i \in \{1,\ldots,n\}} w_{i,k}} \leq \Gamma_0 + \frac{n(\kappa_{\rm B} + L)}{2\vartheta}(1-2\mu) \left(\varsigma + \sum_{j=0}^{k} \|g_j\|^2\right)^{1-2\mu}. \tag{3.16}$$

Suppose now that

$$\sum_{j=0}^{k} \|g_j\|^2 \geq \max\left\{\varsigma, \left[\frac{2^{2\mu}\vartheta(1-2\mu)\Gamma_0}{n(\kappa_{\rm B} + L)}\right]^{\frac{1}{1-2\mu}}\right\}, \tag{3.17}$$

implying

$$\varsigma + \sum_{j=0}^{k} \|g_j\|^2 \leq 2 \sum_{j=0}^{k} \|g_j\|^2 \;\text{ and }\; \Gamma_0 \leq \frac{n(\kappa_{\rm B} + L)}{2\vartheta(1-2\mu)} \left(2 \sum_{j=0}^{k} \|g_j\|^2\right)^{1-2\mu}.$$

Then, using (3.16) and (3.12),

$$\frac{\tau \varsigma^\mu \sqrt{\vartheta}}{2^{1+\mu}\, \kappa_{\rm B} \left[\sum_{\ell=0}^{k} \|g_\ell\|^2\right]^\mu} \sum_{j=0}^{k} \|g_j\|^2 \leq \frac{2^{1-2\mu}\, n\, (\kappa_{\rm B} + L)}{\vartheta}(1-2\mu) \left(\sum_{j=0}^{k} \|g_j\|^2\right)^{1-2\mu}.$$

Solving this inequality for $\sum_{j=0}^{k} \|g_j\|^2$ gives that

$$\sum_{j=0}^{k} \|g_j\|^2 \leq \left[\frac{4\, n\, \kappa_{\rm BBL}}{(1-2\mu)\tau \varsigma^\mu \vartheta^{\frac{3}{2}}}\right]^{\frac{1}{\mu}}$$

and therefore

$$\underset{j\in\{0,\ldots,k\}}{\text{average}} \|g_j\|^2 \leq \left[\frac{4\,n\,\kappa_{\rm BBL}}{(1-2\mu)\tau\varsigma^\mu\vartheta^{\frac{3}{2}}}\right]^{\frac{1}{\mu}} \cdot \frac{1}{k+1}. \tag{3.18}$$

Alternatively, if (3.17) fails, then

$$\underset{j\in\{0,\ldots,k\}}{\text{average}} \|g_j\|^2 < \max\left\{\varsigma, \left[\frac{2^{2\mu}\vartheta(1-2\mu)\Gamma_0}{n(\kappa_{\rm B}+L)}\right]^{\frac{1}{1-2\mu}}\right\} \cdot \frac{1}{k+1}. \tag{3.19}$$

We may also use AS.4 after substituting (3.14) in (3.13) and using (3.12) to deduce that

$$\sum_{j=0}^{k} \frac{\tau\varsigma^\mu\sqrt{\vartheta}\|g_j\|^2}{2\kappa_g^{2\mu}\kappa_{\rm B}(k+1)^\mu} \leq \Gamma_0 + \frac{n(\kappa_{\rm B}+L)}{2\vartheta(1-2\mu)}\left[\left(\varsigma+\kappa_g^2(k+1)\right)^{1-2\mu} - \varsigma^{1-2\mu}\right]$$

and therefore that

$$\begin{aligned}&\underset{j\in\{0,\ldots,k\}}{\text{average}} \|g_j\|^2 \\ &\quad\leq \frac{2\kappa_g^{2\mu}\kappa_{\rm B}}{\tau\varsigma^\mu\sqrt{\vartheta}(k+1)^{1-\mu}}\left[\Gamma_0 + \frac{n(\kappa_{\rm B}+L)}{2\vartheta(1-2\mu)}(k+1)^{1-2\mu}\left((\varsigma+\kappa_g^2)^{1-2\mu}\right) - \varsigma^{1-2\mu}\right].\end{aligned} \tag{3.20}$$

Combining (3.18), (3.19) and (3.20) gives (3.6).

Let us now consider the case where $\mu = \frac{1}{2}$. For each $i \in \{1,\ldots,n\}$, we apply Lemma 3.1 with $a_k = g_{i,k}^2$, $\xi = \varsigma$ and $\alpha = 2\mu = 1$ and obtain that,

$$\sum_{i=1}^{n}\sum_{j=0}^{k}\Delta_{i,j}^2 \leq \frac{1}{\vartheta}\sum_{i=1}^{n}\log\left(\frac{1}{\varsigma}\left(\varsigma + \sum_{\ell=0}^{k}\sum_{i=1}^{n} g_{i,\ell}^2\right)\right) \leq \frac{n}{\vartheta}\log\left(\varsigma + \sum_{\ell=0}^{k}\|g_\ell\|^2\right).$$

and substituting this bound in (3.13) then gives that

$$\sum_{j=0}^{k}\frac{\tau\sqrt{\varsigma\vartheta}\|g_j\|^2}{2\kappa_{\rm B}\max\limits_{i\in\{1,\ldots,n\}} w_{i,k}} \leq \Gamma_0 + \frac{n(\kappa_{\rm B}+L)}{2\vartheta}\log\left(\varsigma + \sum_{j=0}^{k}\|g_j\|^2\right).$$

Suppose now that

$$\sum_{j=0}^{k}\|g_j\|^2 \geq \max\left[\varsigma, \frac{1}{2}e^{\frac{2\vartheta\Gamma_0}{n(\kappa_B+L)}}\right], \tag{3.21}$$

implying that

$$\varsigma + \sum_{j=0}^{k}\|g_j\|^2 \leq 2\sum_{j=0}^{k}\|g_j\|^2 \;\text{ and }\; \Gamma_0 \leq \frac{n(\kappa_{\rm B}+L)}{2\vartheta}\log\left(2\sum_{j=0}^{k}\|g_j\|^2\right).$$

Using (3.12) for $\mu = \frac{1}{2}$, we obtain then that

$$\frac{\tau\sqrt{\varsigma\vartheta}}{2\sqrt{2}\,\kappa_{\rm B}\sqrt{\sum\limits_{\ell=0}^{k}\|g_\ell\|^2}}\sum_{j=0}^{k}\|g_j\|^2 \leq \frac{n(\kappa_{\rm B}+L)}{\vartheta}\log\left(2\sum_{j=0}^{k}\|g_j\|^2\right),$$

that is

$$\frac{\tau\sqrt{2\varsigma}\vartheta^{\frac{3}{2}}}{4\kappa_{\rm B}}\sqrt{\sum_{j=0}^{k}\|g_j\|^2} \leq 2n(\kappa_{\rm B}+L)\log\left(\sqrt{2\sum_{j=0}^{k}\|g_j\|^2}\right). \tag{3.22}$$

Now define

$$\gamma_1 \stackrel{\rm def}{=} \frac{\tau\sqrt{\varsigma}\vartheta^{\frac{3}{2}}}{4\kappa_{\rm B}}, \quad \gamma_2 \stackrel{\rm def}{=} 2n(\kappa_{\rm B}+L) \quad \text{and} \quad u \stackrel{\rm def}{=} \sqrt{2\sum_{j=0}^{k}\|g_j\|^2} \tag{3.23}$$

and observe that that $\gamma_2 > 3\gamma_1$ because $\tau\sqrt{\varsigma}\vartheta^{\frac{3}{2}} \leq 1$ and $\kappa_{\rm B} \geq 1$. The inequality (3.22) can then be rewritten as

$$\gamma_1 u \leq \gamma_2 \log(u). \tag{3.24}$$

Let us denote by $\psi(u) \stackrel{\rm def}{=} \gamma_1 u - \gamma_2\log(u)$. Since $\gamma_2 > 3\gamma_1$, the equation $\psi(u) = 0$ admits two roots $u_1 \leq u_2$ and (3.24) holds for $u \in [u_1, u_2]$. The definition of u_2 then gives that

$$\log(u_2) - \frac{\gamma_1}{\gamma_2}u_2 = 0$$

which is

$$u_2 e^{-\frac{\gamma_1}{\gamma_2}u_2} = 1.$$

Setting $z = -\frac{\gamma_1}{\gamma_2}u_2$, we obtain that

$$ze^z = -\frac{\gamma_1}{\gamma_2}$$

Thus $z = W_{-1}(-\frac{\gamma_1}{\gamma_2}) < 0$, where W_{-1} is the second branch of the Lambert function defined over $[-\frac{1}{e}, 0)$. As $-\frac{\gamma_1}{\gamma_2} \geq -\frac{1}{3}$, z is well defined and thus

$$u_2 = -\frac{\gamma_2}{\gamma_1}z = -\frac{\gamma_2}{\gamma_1}W_{-1}\left(-\frac{\gamma_1}{\gamma_2}\right) > 0.$$

As a consequence, we deduce from (3.24) and (3.23) that

$$\sum_{j=0}^{k}\|g_j\|^2 \leq \frac{1}{2}u_2^2 = \frac{1}{2}\left(\frac{8n\kappa_{\rm B}(\kappa_{\rm B}+L)}{\tau\sqrt{\varsigma}\vartheta^{\frac{3}{2}}}\right)^2\left|W_{-1}\left(-\frac{\tau\sqrt{\varsigma}\vartheta^{\frac{3}{2}}}{8n\kappa_{\rm B}(\kappa_{\rm B}+L)}\right)\right|^2.$$

and

$$\underset{j\in\{0,\ldots,k\}}{\text{average}}\ \|g_j\|^2 \leq \frac{1}{2}\left(\frac{8n\kappa_{\rm B}(\kappa_{\rm B}+L)}{\tau\sqrt{\varsigma}\vartheta^{\frac{3}{2}}}\right)^2\left|W_{-1}\left(-\frac{\tau\sqrt{\varsigma}\vartheta^{\frac{3}{2}}}{8n\kappa_{\rm B}(\kappa_{\rm B}+L)}\right)\right|^2\cdot\frac{1}{k+1}. \tag{3.25}$$

If (3.21) does not hold, we have that

$$\underset{j\in\{0,\ldots,k\}}{\text{average}}\ \|g_j\|^2 < \max\left\{\varsigma, \frac{1}{2}e^{\frac{2\Gamma_0\vartheta}{n(\kappa_B+L)}}\right\}\cdot\frac{1}{k+1}. \tag{3.26}$$

In a manner similar to that used in the case where $\mu < \frac{1}{2}$, we may also substitute the bound

$$\sum_{j=0}^{k}\Delta_{i,j}^2 \leq \frac{1}{\vartheta}\log\left(\frac{1}{\varsigma}\left(\varsigma + \sum_{\ell=0}^{k}g_{i,\ell}^2\right)\right) \leq \frac{1}{\vartheta}\log\left(1+(k+1)\frac{\kappa_g^2}{\varsigma}\right)$$

in (3.13), use AS.4 and (3.12) and obtain that

$$\sum_{j=0}^{k} \frac{\tau\sqrt{\varsigma\vartheta}\|g_j\|^2}{2\kappa_g\kappa_{\rm B}\sqrt{(k+1)}} \leq \Gamma_0 + \frac{n(\kappa_{\rm B}+L)}{2\vartheta}\log\left(1+(k+1)\frac{\kappa_g^2}{\varsigma}\right)$$

and therefore that

$$(k+1)\operatorname*{average}_{j\in\{0,\ldots,k\}} \|g_j\|^2 \leq \frac{2\kappa_g\kappa_{\rm B}}{\tau\sqrt{\varsigma\vartheta}}\sqrt{k+1}\left[\Gamma_0 + \frac{n(\kappa_{\rm B}+L)}{2\vartheta}\log\left(1+(k+1)\frac{\kappa_g^2}{\varsigma}\right)\right]. \quad (3.27)$$

Combining (3.25), (3.26) and (3.27) gives (3.8).

Finally, suppose that $\frac{1}{2} < \mu < 1$. Once more, we apply Lemma 3.1 for each $i \in \{1,\ldots,n\}$ with $a_\ell = g_{i,\ell}^2$, $\xi = \varsigma$ and $\alpha = 2\mu > 1$ and obtain that

$$\sum_{j=1}^{k} \Delta_{i,j}^2 \leq \frac{1}{\vartheta(1-2\mu)}\left(\left(\varsigma + \sum_{\ell=0}^{k} g_{i,\ell}^2\right)^{1-2\mu} - \varsigma^{1-2\mu}\right) \leq \frac{\varsigma^{1-2\mu}}{\vartheta(2\mu-1)}. \quad (3.28)$$

Substituting the bound (3.28) in (3.13) and using (3.12) and AS.4 gives that

$$\sum_{j=0}^{k} \frac{1}{(\varsigma+\sum_{j=0}^{k}\|g_j\|^2)^{\mu}} \frac{\tau\varsigma^{\mu}\sqrt{\vartheta}\|g_j\|^2}{2\kappa_{\rm B}} \leq \Gamma_0 + \frac{n(\kappa_{\rm B}+L)\varsigma^{1-2\mu}}{2\vartheta(2\mu-1)}.$$

If we now suppose that

$$\sum_{j=0}^{k} \|g_j\|^2 \geq \varsigma, \quad (3.29)$$

then

$$\operatorname*{average}_{j\in\{0,\ldots,k\}} \|g_j\|^2 \leq \left[\frac{2^{1+\mu}\kappa_{\rm B}}{\tau\varsigma^{\mu}\sqrt{\vartheta}}\left(\Gamma_0 + \frac{n(\kappa_{\rm B}+L)\varsigma^{1-2\mu}}{2\vartheta(2\mu-1)}\right)\right]^{\frac{1}{1-\mu}} \cdot \frac{1}{k+1}. \quad (3.30)$$

If (3.29) does not hold, we derive that

$$\operatorname*{average}_{j\in\{0,\ldots,k\}} \|g_j\|^2 \leq \frac{\varsigma}{(k+1)}. \quad (3.31)$$

Alternatively again, we may substituting the bound (3.28) in (3.13) and use AS.4 and (3.12) to deduce that

$$\sum_{j=0}^{k} \frac{\tau\varsigma^{\mu}\sqrt{\vartheta}\|g_j\|^2}{2\kappa_g^{2\mu}\kappa_{\rm B}(k+1)^{\mu}} \leq \Gamma_0 + \frac{n(\kappa_{\rm B}+L)\varsigma^{1-2\mu}}{2\vartheta(2\mu-1)}$$

$$(k+1)\operatorname*{average}_{j\in\{0,\ldots,k\}} \|g_j\|^2 \leq \frac{2\kappa_g^{2\mu}\kappa_{\rm B}}{\tau\varsigma^{\mu}\sqrt{\vartheta}}(k+1)^{\mu}\left[\Gamma_0 + \frac{n(\kappa_{\rm B}+L)\varsigma^{1-2\mu}}{2\vartheta(2\mu-1)}\right], \quad (3.32)$$

Thus, (3.30), (3.31) and (3.32) finally imply (3.10). □

The attentive reader has noticed that the constant κ_g does not occur in the expressions for κ_3, κ_6 and κ_7, and, indeed, AS.3 is not needed to derive the upper complexity bounds given by the first terms in the minimum of the right-hand sides of (3.6), (3.8) and (3.10). This allows us to state the following immediate corollary.

Corollary 3.3 Suppose that AS.1, AS.2 and AS.4 hold, and that the ASTR1 algorithm is applied to problem (2.1) with its scaling given by (3.1) and $\mu \in (0,1)$. Then

$$\min_{j\in\{0,\ldots,k\}} \|g_j\| \leq \frac{\kappa_\circ}{\sqrt{k+1}} \tag{3.33}$$

where $\kappa_\circ$ is κ_3 (defined in (3.7)) when $\mu \in (0, \frac{1}{2})$, κ_6 (defined in (3.9)) when $\mu = \frac{1}{2}$ or κ_7 (defined in (3.11)) when $\mu \in (\frac{1}{2}, 1)$.

Proof. This "k-order" bound is obtained by considering only the first terms in the minimum appearing in the right-hand sides of the bounds (3.6), (3.8) and (3.10). □

These result suggest additional remarks.

1. Theorem 3.2 is stronger than Corollary 3.3 in that the second terms in the minima of the right-hand sides of (3.6), (3.8) and (3.10) may provide better bounds that the first for small k, depending on the value of the problem-dependent constants involved[(4)]. The inequality (3.33) however always gives the best bound for large k.

2. That the bounds given by the first terms in the minima are not continuous as stated at $\mu = \frac{1}{2}$ is a result of our bounding process within the proof of Theorem 3.2 (for instance in the last inequality of (3.14)) and, given that the bounds of Lemma 3.1 are continuous, may therefore not be significant. (Note that the bounds given by the second terms are continuous at $\mu = \frac{1}{2}$.)

3. If the algorithm is terminated as soon as $\|g_k\| \leq \epsilon$ (which is customary for deterministic algorithms searching for first-order points), it must stop at the latest at iteration

$$k = \kappa_\star^2 \epsilon^{-2}, \tag{3.34}$$

   where $\kappa_\star = \kappa_3$ for $\mu \in (0, \frac{1}{2})$, $\kappa_\star = \kappa_6$ for $\mu = \frac{1}{2}$ and $\kappa_\star = \kappa_7$ for $\mu \in (\frac{1}{2}, 1)$. It is truly remarkable that there exist first-order OFFO methods whose global complexity order is identical to that of standard first-order methods using function evaluations (see [24, 18, 3] or [6, Chapter 2]), despite the fact that the latter exploit significantly more information.

4. The bounds obtained by considering only the second term in the minima of the right-hand sides of (3.6), (3.8) and (3.10) correspond to deterministic versions of those presented in [17, Theorem 3.5] for the case where $B_k = 0$ for all k and the gradient evaluations are contaminated by random noise.

[(4)] This is in particular the case for $\mu = \frac{1}{2}$, small k and n sufficiently large, because the dependence on n of κ_6 is worse than that of κ_2

5. In particular, if $B_k = 0$ for all k and $\mu = \frac{1}{2}$, (3.8) gives an upper complexity bound for the deterministic momentum-less Adagrad algorithm which is significantly better than that proposed by [10, 17] and more recently by [30] for a very specific choice of the stepsize (learning rate).

6. It is possible to give a more explicit bound on κ_6 by finding an upper bound on the value of the involved Lambert function. This can be obtained by using [7, Theorem 1] which states that, for $x > 0$,

$$\left|W_{-1}(-e^{-x-1})\right| \leq 1 + \sqrt{2x} + x. \tag{3.35}$$

   Remembering that, for γ_1 and γ_2 given by (3.23), $\log\left(\frac{\gamma_2}{\gamma_1}\right) \geq \log(3) > 1$ and taking $x = \log\left(\frac{\gamma_2}{\gamma_1}\right) - 1 > 0$ in (3.35) then gives that

$$\left|W_{-1}\left(-\frac{\gamma_1}{\gamma_2}\right)\right| \leq \log\left(\frac{\gamma_2}{\gamma_1}\right) + \sqrt{2\left(\log\left(\frac{\gamma_2}{\gamma_1}\right) - 1\right)}.$$

7. If the approximations B_k are chosen to be positive-semidefinite and $\mu = \frac{1}{2}$, then the fourth term on the right-hand side of (2.14) can be ignored and second term in the minimum of the right-hand side of (3.8) simplifies to

$$\frac{2\kappa_g\kappa_{\rm B}\Gamma_0}{\sqrt{k+1}} + \left(\frac{n\kappa_g\kappa_{\rm B}L}{\tau\sqrt{\varsigma}}\right)\frac{\log\left(1+(k+1)\kappa_g^2\right)}{\sqrt{k+1}}.$$

8. If the choices $B_k = 0$ and $\mu = \frac{1}{2}$ are made at every iteration (yielding a deterministic momentum-less Adagrad with unit stepsize), one verifies that the second term in the minimum of the right-hand side of (3.8) then reduces to

$$\frac{2\kappa_g\Gamma_0}{\sqrt{k+1}} + \left(\frac{n\kappa_g L}{\tau\sqrt{\varsigma}}\right)\frac{\log\left(1+(k+1)\kappa_g^2\right)}{\sqrt{k+1}},$$

   which gives a deterministic variant of the bound given in [10, Theorem 1].

9. It is also possible to extend the definition of s_k^L in (2.6) by premultiplying it by a stepsize $\alpha_k \in [\alpha_{\min}, 1]$ for some $\alpha_{\min} \in (0, 1]$. Our results again remain valid (with modified constants). Covering a deterministic momentum-less Adam would require extending the results to allow for (3.1) to be replaced by

$$w_{i,k} = \varsigma + \sum_{j=0}^{k} \beta_2^{k-j} g_{i,j}^2 \quad (i \in \{1, \ldots, n\}) \tag{3.36}$$

   for some $\beta_2 < 1$. This can be done by following the argument of Theorem 2 in [10]. However, as in this reference, the final bound on the squared gradient norms does not tend to zero when k grows[(5)], illustrating the (known) lack of convergence of Adam. We therefore do not investigate this option in detail.

[(5)] A constant term in $-\log(\beta_2)$ refuses to vanish.

10. The ϵ-order (3.34) also improves slightly on the $\mathcal{O}(\epsilon^{-2} + |\log(\epsilon)|)$ ϵ-order obtained for the DFO-based approach mentioned in the introduction (which, we recall, requires very strong assumptions on global minimization).

Is the bound (good) upper bound (3.33) sharp?

Theorem 3.4 The bound (3.33) is essentially sharp in that, for each $\mu \in (0,1)$ and each $\eta \in (0,1]$, there exists a univariate function $f_{\mu,\eta}$ satisfying AS.1-AS.4 such that, when applied to minimize $f_{\mu,\eta}$ from the origin, the ASTR1 algorithm with (3.1) and $\vartheta = 1$ produces a sequence of gradient norms given by $\|g_k\| = \frac{1}{k^{\frac{1}{2}+\eta}}$.

Proof. Following ideas of [6, Theorem 2.2.3], we first construct a sequence of iterates $\{x_k\}$ for which $f_{\mu,\eta}(x_k) = f_k$ and $\nabla^1_x f_{\mu,\eta}(x_k) = g_k$ for associated sequences of function and gradient values $\{f_k\}$ and $\{g_k\}$, and then apply Hermite interpolation to exhibit the function $f_{\mu,\eta}$ itself. We start by defining

$$g_0 \stackrel{\rm def}{=} -2, \quad g_k \stackrel{\rm def}{=} -\frac{1}{k^{\frac{1}{2}+\eta}} \quad (k>0), \tag{3.37}$$

$$s_0 \stackrel{\rm def}{=} \frac{2}{(\varsigma+4)^{\mu}}, \quad s_k \stackrel{\rm def}{=} \frac{1}{k^{\frac{1}{2}+\eta}(\varsigma+\sum_{j=0}^{k} g_j^2)^{\mu}} \quad (k>0) \tag{3.38}$$

yielding that

$$|g_0 s_0| = \frac{4}{(\varsigma+4)^{\mu}}, \quad |g_k s_k| = \frac{1}{k^{1+2\eta}(\varsigma+\sum_{j=0}^{k} g_j^2)^{\mu}} \leq \frac{1}{k^{1+2\eta}} \quad (k>0) \tag{3.39}$$

(remember that $g_0^2 = 4$). We then define $B_k \stackrel{\rm def}{=} 0$ for all $k \geq 0$,

$$x_0 = 0, \quad x_{k+1} = x_k + s_k \quad (k>0) \tag{3.40}$$

and

$$f_0 = \frac{4}{(\varsigma+4)^{\mu}} + \zeta(1+2\eta) \quad \text{and} \quad f_{k+1} = f_k + g_k s_k \quad (k \geq 0), \tag{3.41}$$

where $\zeta(\cdot)$ is the Riemann zeta function. Observe that the sequence $\{f_k\}$ is decreasing and that, for all $k \geq 0$,

$$f_{k+1} = f_0 - \sum_{k=0}^{k} |g_k s_k| \geq f_0 - \frac{4}{(\varsigma+4)^{\mu}} - \sum_{k=1}^{k} \frac{1}{k^{1+2\eta}} \geq f_0 - \frac{4}{(\varsigma+4)^{\mu}} - \zeta(1+2\eta)$$

where we used (3.41) and (3.39). Hence (3.41) implies that

$$f_k \in [0, f_0] \ \text{ for all } \ k \geq 0. \tag{3.42}$$

Also note that, using (3.41),

$$|f_{k+1} - f_k - g_k s_k| = 0, \tag{3.43}$$

while, using (3.38),

$$|g_0 - g_1| = 1 \leq \frac{1}{2}(\varsigma + 4)^\mu \, s_0.$$

Moreover, using the fact that $1/x^{\frac{1}{2}+\eta}$ is a convex function of x over $[1, +\infty)$, and that from (3.38) $s_k \geq \frac{1}{k^{\frac{1}{2}+\eta}(\varsigma+4+k)^\mu}$, we derive that, for $k > 0$,

$$\begin{aligned}
|g_{k+1} - g_k| &= \left| \frac{1}{(k+1)^{\frac{1}{2}+\eta}} - \frac{1}{k^{\frac{1}{2}+\eta}} \right| \\
&\leq \left(\frac{1}{2} + \eta \right) \frac{1}{k^{\frac{3}{2}+\eta}} \\
&\leq \frac{3}{2} \, \frac{(\varsigma + 4 + k)^\mu}{k k^{\frac{1}{2}+\eta} (\varsigma + 4 + k)^\mu} \\
&\leq \frac{3}{2} \, \frac{(\varsigma + 4 + k)^\mu}{k} s_k \\
&\leq \frac{3}{2} \, (\varsigma + 5)^\mu \, s_k.
\end{aligned}$$

These last bounds and (3.42) allow us to use standard Hermite interpolation on the data given by $\{f_k\}$ and $\{g_k\}$: see, for instance, Theorem A.9.1 in [6] with $p = 1$ and

$$\kappa_f = \max \left[\frac{3}{2}(\varsigma + 5)^\mu, f_0, 2 \right] \tag{3.44}$$

(the second term in the max bounding $|f_k|$ because of (3.42) and the third bounding $|g_k|$ because of (3.37)). We then deduce that there exists a continuously differentiable function $f_{\mu,\eta}$ from $\mathbb{R}$ to $\mathbb{R}$ with Lipschitz continuous gradient (i.e. satisfying AS.1 and AS.2) such that, for $k \geq 0$,

$$f_{\mu,\eta}(x_k) = f_k \quad \text{and} \quad \nabla^1_x f_{\mu,\eta}(x_k) = g_k.$$

Moreover, the range of $f_{\mu,\eta}$ and $\nabla^1_x f_{\mu,\eta}$ are constant independent of η, hence guaranteeing AS.3 and AS.4. The definitions (3.37), (3.38), (3.40) and (3.41) imply that the sequences $\{x_k\}$, $\{f_k\}$ and $\{g_k\}$ can be seen as generated by the ASTR1 algorithm (with $B_k = 0$) applied to $f_{\mu,\eta}$, starting from $x_0 = 0$ and the desired conclusion follows. □

The bound (3.33) is therefore *essentially sharp* (in the sense of [5]) for the ASTR1 algorithm with (3.1) and $\vartheta = 1$, which is to say that the lower complexity bound for the algorithm is arbitrarily close to its upper bound (3.33). Interestingly, the argument in the proof of the above theorem fails for $\eta = 0$, as this choice yields that

$$\sum_{j=0}^{k} g_j^T s_j \geq \sum_{j=0}^{k} \frac{1}{k(\varsigma + \log(k+1))^\mu}.$$

Since

$$\int_1^k \frac{dt}{t(\log(t+1))^\mu} > \int_1^k \frac{dt}{(t+1)(\log(t+1))^\mu} = \frac{(\log(k+1))^{1-\mu}}{1-\mu} - \frac{\log(2)^{1-\mu}}{1-\mu}$$

tends to infinity as k grows, this indicates (in view (3.41)) that AS.4 cannot hold.

Figure 1 shows the behaviour of $f_{\mu,\eta}(x)$ for $\mu = \frac{1}{2}$ and $\eta = \varsigma = \frac{1}{100}$, its gradient and Hessian. The top three panels show the interpolated function resulting from the first 100 iterations of the ASTR1 algorithm with (3.1), while the bottom three panels report using 10^4 iterations. (We have chosen to shift f_0 to 100 in order to avoid large numbers on the vertical axis of the left panels.) One verifies that the gradient is continous and converges to zero. Since the Hessian remains bounded where defined, this indicates that the gradient is Lipschitz continuous. Due to the slow convergence of the series $\sum_j 1/j^{\frac{1}{1+2/100}}$, illustrating the boundeness of $f_0 - f_{k+1}$ would require many more iterations. One also notes that the gradient is not monotonically increasing, which implies that $f_{\mu,\eta}(x)$ is nonconvex, although this is barely noticeable in the left panels of the figure. Note that the fact that the example is unidimensional is not restrictive, since it is always possible to make the value of its objective function and gradient independent of all dimensions but one.

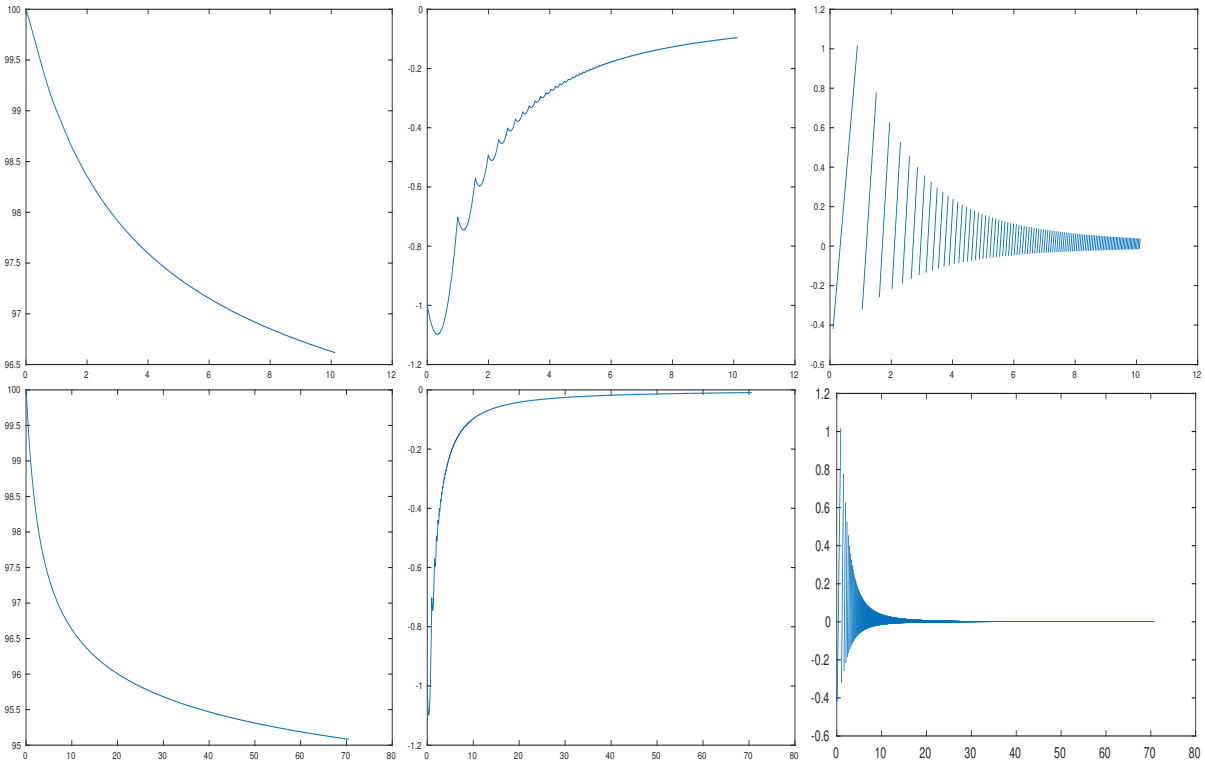

Figure 1: The function $f_{\mu,\eta}(x)$ (left), its gradient $\nabla_x^1 f_{\mu,\eta}(x)$ (middle) and its Hessian $\nabla_x^2 f_{\mu,\eta}(x)$ (right) plotted as a function of x, for the first 100 (top) and 10^4 (bottom) iterations of the ASTR1 algorithm with (3.1) ($\mu = \frac{1}{2}$, $\eta = \varsigma = \frac{1}{100}$)

4 A further "divergent stepsizes" variation on this theme

We now use a different proof technique to design new variants of ASTR1 with a fast global k-order. This is achieved by modifying the definition of the scaling factors $w_{i,k}$, requiring them to satisfy a fairly general growth condition explicitly depending on k, the iteration index. The motivation for introducing these new variants is their remarkable numerical performance when applied to noisy examples [17].

Theorem 4.1 Suppose that AS.1–AS.4 hold and that the ASTR1 algorithm is applied to problem (2.1), where, the scaling factors $w_{i,k}$ are chosen such that, for some power parameter $0 < \nu \leq \mu < 1$, all $i \in \{1, \ldots, n\}$ and some constants $\kappa_w \geq \varsigma_i \in (0, 1]$,

$$\varsigma_i \, (k+1)^\nu \leq w_{i,k} \leq \kappa_w \, (k+1)^\mu \quad (k \geq 0). \tag{4.1}$$

Then, for any $\theta \in (0, \tau\varsigma_{\min})$,

$$\underset{j \in \{j_\theta+1, \ldots, k\}}{\text{average}} \|g_j\|^2 \leq \kappa_\diamond \frac{(k+1)^\mu}{k - j_\theta} \leq \frac{2\kappa_\diamond (j_\theta + 1)}{k^{1-\mu}}, \tag{4.2}$$

where

$$j_\theta \stackrel{\text{def}}{=} \left(\frac{\kappa_{\rm B}(\kappa_{\rm B} + L)}{\varsigma_{\min}(\tau\varsigma_{\min} - \theta)} \right)^{\frac{1}{\nu}} \tag{4.3}$$

$\Gamma_0 \stackrel{\text{def}}{=} f(x_0) - f_{\rm low}$ and

$$\kappa_\diamond \stackrel{\text{def}}{=} \frac{2\kappa_w \kappa_{\rm B}}{\theta} \left(\Gamma_0 + n(j_\theta + 1) \frac{\kappa_g^2 (\kappa_{\rm B} + L)}{2\varsigma_{\min}^2} \right).$$

Proof. From (2.11) and AS.4, using $w_{\min,j} \stackrel{\text{def}}{=} \min_{i \in \{1,\ldots,n\}} w_{i,k}$ ensures that

$$\Gamma_0 \geq f(x_0) - f(x_{k+1}) \geq \sum_{j=0}^{k} \sum_{i=1}^{n} \frac{g_{i,j}^2}{2\kappa_{\rm B} w_{i,j}} \left[\tau\varsigma_{\min} - \frac{\kappa_{\rm BBL}}{w_{\min,j}} \right]. \tag{4.4}$$

Consider now an arbitrary $\theta \in (0, \tau\varsigma_{\min})$ and suppose first that, for some j,

$$\left[\tau\varsigma_{\min} - \frac{\kappa_{\rm BBL}}{w_{\min,j}} \right] \leq \theta, \tag{4.5}$$

i.e., using (4.1),

$$\varsigma_i \, j^\nu \leq w_{\min,j} \leq \frac{\kappa_{\rm BBL}}{\tau\varsigma_{\min} - \theta}.$$

But this is impossible for $j > j_\theta$ for j_θ given by (4.3), and hence (4.5) fails for all $j > j_\theta$. But we also know from (4.1) that, for $i \in \{1, \ldots, n\}$,

$$\left[\tau\varsigma_{\min} - \frac{\kappa_{\rm BBL}}{w_{\min,j}} \right] \geq -\frac{\kappa_{\rm BBL}}{\varsigma_{\min}}$$

so that, using (4.1) and the definition of $\kappa_{\rm BBL}$,

$$\begin{aligned} \sum_{j=0}^{j_\theta} \sum_{i=1}^{n} \frac{g_{i,j}^2}{2\kappa_{\rm B} w_{i,j}} \left[\tau\varsigma_{\min} - \frac{\kappa_{\rm BBL}}{w_{\min,j}} \right] &\geq -\sum_{j=0}^{j_\theta} \frac{\|g_j\|^2}{2\kappa_{\rm B} w_{i,j}} \frac{\kappa_{\rm BBL}}{\varsigma_{\min}} \\ &\geq -\frac{n\kappa_g^2(\kappa_{\rm B} + L)}{2\varsigma_{\min}^2} \sum_{j=0}^{j_\theta} \frac{1}{(j+1)^\nu} \\ &\geq -(j_\theta + 1) \frac{n\kappa_g^2(\kappa_{\rm B} + L)}{2\varsigma_{\min}^2}. \end{aligned} \tag{4.6}$$

Thus, combining (4.1), (4.4) and (4.6), we deduce that

$$\frac{\theta}{2\kappa_w\kappa_{\rm B}}\sum_{j=j_\theta+1}^{k}\frac{\|g_j\|^2}{(j+1)^\mu} \leq \sum_{j=j_\theta+1}^{k}\frac{\|g_j\|^2}{2\kappa_{\rm B}w_{i,j}}\left[\tau\varsigma_{\min}-\frac{\kappa_{\rm BBL}}{w_{\min,j}}\right]$$
$$\leq \Gamma_0 + n(j_\theta+1)\frac{\kappa_g^2(\kappa_{\rm B}+L)}{2\varsigma_{\min}^2},$$

in turn implying that

$$\frac{k-j_\theta}{(k+1)^\mu}\underset{j\in\{j_\theta+1,\ldots,k\}}{\text{average}}\|g_j\|^2 \leq \frac{2\kappa_w\kappa_{\rm B}}{\theta}\left(\Gamma_0 + n(j_\theta+1)\frac{\kappa_g^2(\kappa_{\rm B}+L)}{2\varsigma_{\min}^2}\right),$$

finally giving (4.2). □

We again provide some comments on this last result.

1. The choice (4.1) is of course reminiscent, in a smooth and nonconvex setting, of the "divergent stepsize" subgradient method for non-smooth convex optimization (see [2] and the many references therein), for which a $\mathcal{O}(1/\sqrt{k})$ global rate of convergence is known (Theorems 8.13 and 8.30 in this last reference).

2. Theorem 4.1 gives a complexity bound for iterations that are beyond an *a priori* computable iteration index. Indeed that j_θ only depends on ν and problem's constants and, in particular does not depend on k. As a consequence, it can be used to determine a complexity bound to reach an iteration satisfying the accuracy requirement $\|g_k\| \leq \epsilon$, which is then $\mathcal{O}\left(\epsilon^{-\frac{2}{1-\mu}}\right) + j_\theta(\nu)$.

3. As the chosen values of μ and ν approach zero, then the k-order of convergence beyond j_θ tends to $\mathcal{O}(1/\sqrt{k})$, which the order derived for the methods of the previous section and is the standard k-order for first-order methods using evaluations of the objective function, albeit the value of j_θ might increase.

4. Choosing $\nu > 1$ would prevent the ASTR1 algorithm with (4.1) to converge from arbitrary starting points since then (2.2), (2.4), AS.3 and (4.1) imply that

$$\sum_{k=0}^{\infty}\|s_k\|_\infty \leq \sum_{k=0}^{\infty}\max_{i\in\{1,\ldots,n\}}\frac{|g_{i,k}|}{w_{i,k}} \leq \kappa_g\sum_{k=0}^{\infty}k^{-\nu} < +\infty$$

   which would limit the distance between x_0 and any ϵ-approximate first-order point.

5. Note that the requirement (4.1) allows a variety of choices for the scaling factors. Some possible choices will be explored from the numerical point of view in the next section.

We are now again interested to estimate how sharp the k-order bound (4.2) in $\mathcal{O}(\frac{1}{k^{(1-\nu)/2}})$ is.

Theorem 4.2 The bound (4.2) is essentially sharp in that, for any $\omega > \frac{1}{2}(1-\nu)$, there exists a univariate function $f_\omega(x)$ satisfying AS.1–AS.4 such that the ASTR1 algorithm with (4.1) applied to this function produces a sequence of gradient norms given by $\|g_k\| = \frac{1}{k^\omega}$.

Proof. Consider the sequence defined, for some $\omega \in (\frac{1}{2}(1-\nu), 1]$ and all $k \geq 0$, by

$$g_k = -\frac{1}{k^\omega} \quad w_k = k^\nu, \quad s_k = \frac{1}{k^{2\omega-\nu}} < 1 \quad \text{and} \quad f_{k+1} = f_k + g_k s_k, \tag{4.7}$$

where we have chosen $f_0 = \zeta(2\omega+\frac{1}{2})$ where $\zeta(\cdot)$ is the Riemann zeta function. Immediately note that

$$\lim_{k\to\infty} |g_k| = 0,$$

and $|g_k| \leq= 1 = \kappa_g$ for all k. We now verify that, if

$$x_0 = 0 \quad \text{and} \quad x_k = x_{k-1} + s_{k-1} \quad \text{for} \quad k \geq 1,$$

then exists a function $f_\omega(x)$ satisfying AS.1–AS.3 such that, for all $k \geq 0$,

$$f_\omega(x_k) = f_k, \quad \text{and} \quad g_\omega(x_k) = g_k,$$

and such that the sequence defined by (4.7) is generated by applying the ASTR1 algorithm using $B_k = 0$. The function $f_\omega(x)$ is constructed using Hermite interpolation on each interval $[x_k, x_{k+1}]$ (note that the x_k are monotonically increasing), which known (see [3] or [6, Th. A.9.2]) to exist whenever there exists a constant $\kappa_f \geq 0$ such that, for each k,

$$|f_{k+1} - f_k - g_k s_k| \leq \kappa_f |s_k|^2 \quad \text{and} \quad |g_{k+1} - g_k| \leq \kappa_f |s_k|.$$

The first of these conditions holds by construction of the $\{f_k\}_{k\geq 0}$. To verify the second, we first note that, because $1/k^\omega$ is a convex function of k and $|1/k| \leq 1$,

$$\frac{|g_{k+1} - g_k|}{|s_k|} \leq \frac{\omega k^{2\omega-\nu}}{k^{1+\omega}} \leq \frac{\omega}{k^{\nu-\omega+1}} \leq \omega \quad (k \geq 0), \tag{4.8}$$

where $\nu - \omega + 1 \geq \nu > 0$, so that the desired inequality holds with $\kappa_f = \omega$.

Moreover, Hermite interpolation guarantees that $f_\omega(x)$ is bounded below whenever $|f_k|$ and $|s_k|$ remain bounded. We have already verified the second of these conditions in (4.7). We also have from (4.7) that

$$f_0 - f_{k+1} = \sum_{j=0}^{k} \frac{1}{j^{2\omega} j^\nu} \tag{4.9}$$

which converges to the finite limit $\zeta(2\omega+\nu)$ because we have chosen $\omega > \frac{1}{2}(1-\nu)$. Thus $f_k \in (0, \zeta(2\omega+\nu)]$ for all k and the first condition is also satisfied and AS.4 holds. This completes our proof. □

The conclusions which can be drawn from this theorem parallel those drawn after Theorem 3.4. The bound (4.2) is essentially sharp (in the sense of [5][(6)]) for the ASTR1 algorithm with (4.1). Figure 2 shows the behaviour of $f_\omega(x)$ for $\nu = \frac{1}{9}$ and $\omega = \frac{4}{9} + \frac{1}{100}$, its gradient and Hessian. The top three panels show the interpolated function resulting from the first 100 iterations of the ASTR1 algorithm with (4.1), while the bottom three panels report using 5.10^4 iterations. (We have again chosen to shift f_0 to 100 in order to avoid large numbers on the vertical

[(6)] Observe that f_0 now tends to infinity when ω tends to $\frac{1}{2}(\nu-1)$ and hence that AS.4 fails in the limit. As before, the structure of (4.2) implies that the complexity bound deteriorates when the gap $\Gamma_0 = f(x_0) - f_{\rm low}$ grows.

axis of the left panels.) As above, one verifies that the gradient is continous, non-monotone and converges to zero and that the Hessian remains bounded where defined, illustrating the gradient's Lipschitz continuity. Finally, as for Theorem 3.4, the argument in the proof of Theorem 4.2 fails for $\omega = \frac{1}{2}(1-\nu)$ because the sum in (4.9) is divergent in this case, which prevents AS.4 to hold.

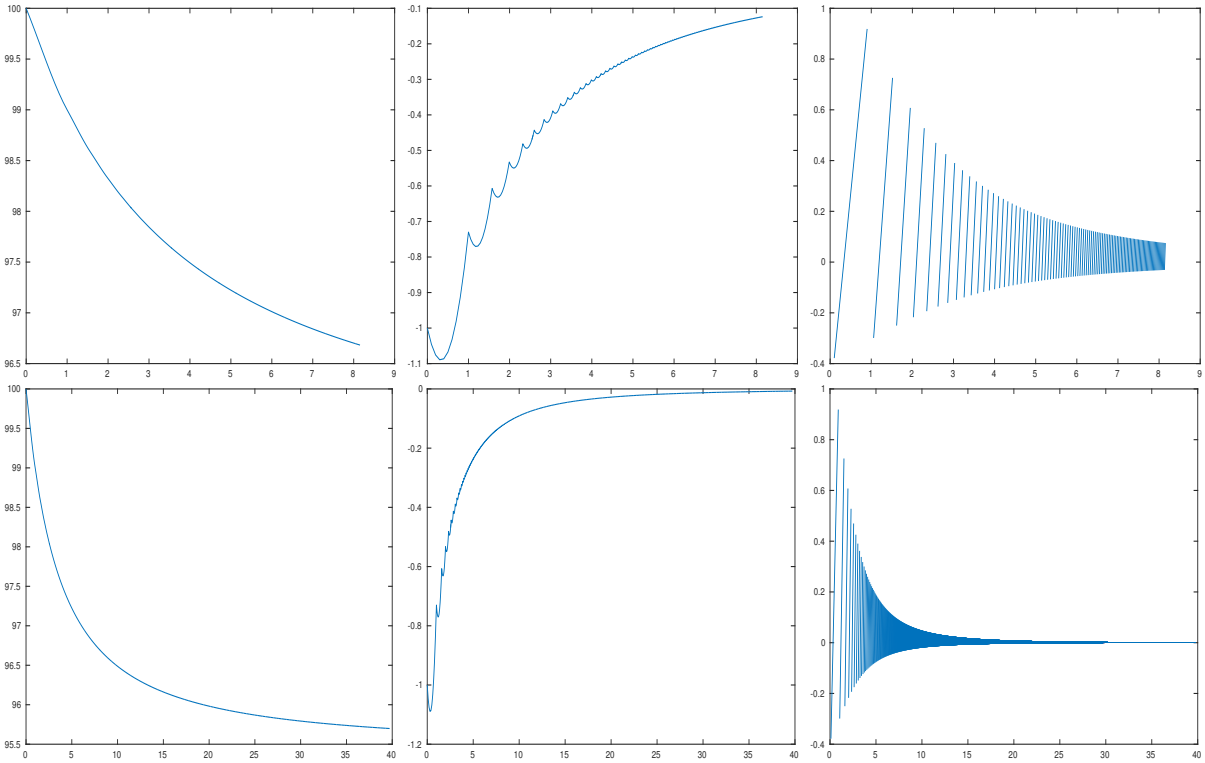

Figure 2: The function $f_\omega(x)$ (left), its gradient $\nabla^1_x f_\omega(x)$ (middle) and its Hessian $\nabla^2_x f_\omega(x)$ (right) plotted as a function of x, for the first 100 (top) and 5.10^4 (bottom) iterations of the ASTR1 algorithm with (4.1) ($\nu = \frac{1}{9}$, $\omega = \frac{4}{9} + \frac{1}{100}$)

5 Numerical illustration

Because the numerical behaviour of the methods discussed above is not well-known[(7)], we now provide some numerical illustration. For the sake of clarity and conciseness, we needed to keep the list of algorithmic variants reported here reasonably limited, and have taken the following considerations into account for our choice.

1. Both scaling techniques (3.1) and (4.1) are illustrated. Moreover, since the Adam algorithm using (3.36) is so commonly used the stochastic context, we also included it in the comparison.

2. Despite Theorems 3.2 and 4.1 covering a wide choice of the parameters μ and ν, we have chosen to focus here on the most common choice for (3.1) and (3.36) (i.e. $\mu = \frac{1}{2}$

[(7)] Except for stochastic finite sum minimization.

and $\beta_2 = \frac{9}{10}$, corresponding to Adagrad and Adam). When using (4.1), we have also restricted our compararison to the single choice of μ and ν used (with reasonable success) in [17], namely $\mu = \nu = \frac{1}{10}$.

3. In order to be able to test enough algorithmic variants on enough problems in reasonable computing time, we have decided to limit our experiments to low-dimensional problems. For the same reason, we have focused our experiments on the case where $\vartheta = 1$.

4. We have chosen to define the step s_k in Step 3 of the ASTR1 algorithm by approximately minimizing the quadratic model (2.10) within the ℓ_∞ trust-region using a projected truncated conjugate-gradient approach [22, 23] which is terminated as soon as

$$\|g_k + B_k s_k\|_2 \leq \max\left[10^{-12}, 10^{-5}\|g_k\|_2\right].$$

   We also considered an alternative, namely that of minimizing the quadratic model in an Euclidean ℓ_2 trust region (with the same accuracy requirement) using a Generalized Lanczos Trust Region (GLTR) technique [14].

5. We thought it would be interesting to compare "purely first-order" variants (that is variants for which $B_k = 0$ for all k) with methods using some kind of Hessian approximation. Among many possibilities, we selected three types of approximations of interest. The first is the diagonal Barzilai-Borwein approximation [1]

$$B_{k+1} = \frac{\|s_k\|_2^2}{y_k^T s_k} I_n \tag{5.1}$$

   where I_n is the identity matrix of dimension n, $y_k = g_{k+1} - g_k$ and $y_k^T s_k >= 10^{-15}\|s_k\|_2^2$. The second is limited-memory BFGS approximations [21], where a small number of BFGS updates are added to the matrix (5.1), each update corresponding to a secant pair (y_k, s_k). The third is not to approximate the Hessian at all, but to use its exact value, that is $B_k = \nabla_x^2 f(x_k)$ for all k.

Given these considerations, we have selected the following algorithmic variants:

`adag1` : the `ASTR1` algorithm using the Euclidean $\|\cdot\|_2$ norm and

$$w_{i,k} = \left[\varsigma + \sum_{j=0}^{k} \|g_j\|_2^2\right]^{\frac{1}{2}} \qquad \text{for } i \in \{1,\ldots,n\},$$

`adagi1` : the `ASTR1` algorithm using the $\|\cdot\|_\infty$ norm and (3.1) with $\mu = \frac{1}{2}$,

`adag2` : the `ASTR1` algorithm using the Euclidean $\|\cdot\|_2$ norm and

$$w_{i,k} = \left[\varsigma + \sum_{j=0}^{k} \beta_2^{k-j}\|g_j\|_2^2\right]^{\frac{1}{2}} \qquad \text{for } i \in \{1,\ldots,n\},$$

with $\beta_2 = \frac{9}{10}$,

`adagi2` : the `ASTR1` algorithm using the $\|\cdot\|_\infty$ norm and (3.36) with $\mu = \frac{1}{2}$ and $\beta_2 = \frac{9}{10}$,

`maxg01` : the `ASTR1` algorithm using (4.1) with

$$w_{i,k} = (k+1)^{\frac{1}{10}} \max\left[\varsigma, \max_{j\in\{0,\ldots,k\}} \|g_j\|_2\right] \qquad \text{for } i \in \{1,\ldots,n\},$$

`maxgi01` : the `ASTR1` algorithm using (4.1) with

$$w_{i,k} = (k+1)^{\frac{1}{10}} \max\left[\varsigma, \max_{j\in\{0,\ldots,k\}} |g_{i,j}|\right] \qquad \text{for } i \in \{1,\ldots,n\},$$

`sdba` : the standard steepest-descent algorithm using Armijo backtracking (see [6, Algorithm 2.2.1], for instance),

`b1adagi1` : `adagi1` where B_k is the Barzilai-Borwein Hessian approximation (5.1),

`lmadagi3b` : `adagi1` where B_k is a limited-memory BFGS Hessian approximation with 3 secant pairs,

`Eadagi1` : `adagi1` using the exact Hessian, i.e. $B_k = \nabla^2 f(x_k)$ for all k.

When relevant, all variants use $\varsigma = 0.01$. The first seven algorithms are "purely first-order" in the sense discussed above. Note that, under AS.3, `maxg01` and `maxgi01` satisfy (4.1) with $\mu = \nu = \frac{1}{10}$, $\varsigma_i = \varsigma$ and $\kappa_w = \kappa_g$. Also note that `adagi1` and `adagi2` are nothing but the deterministic versions of Adagrad and Adam, respectively. All on the algorithms were run[(8)] on the low dimensional instances of the problems[(9)] of the OPM collection (January 2022) [19] listed with their dimension in Table 1, until either $\|\nabla_x^1 f(x_k)\|_2 \leq 10^{-6}$, or a maximum of 100000 iterations was reached, or evaluation of the gradient returned an error.

Before considering the results, we make two additional comments. The first is that very few of the test functions satisfy AS.3 on the whole of $\mathbb{R}^n$. While this is usually not a problem when testing standard first-order descent methods (because AS.3 may then be true in the level set determined by the starting point), this is no longer the case for significantly non-monotone methods like the ones tested here. As a consequence, it may (and does) happen that the gradient evaluation is attempted at a point where its value exceeds the Matlab overflow limit, causing the algorithm to fail on the problem. The second comment is that the

[(8)] In Matlab® on a Lenovo ThinkPad X1 Carbon with four cores and 8 GB of memory.

[(9)] From their standard starting point.

Problem	n	Problem	n	Problem	n	Problem	n	Problem	n	Problem	n
argauss	3	chebyqad	10	dixmaanl	12	heart8ls	8	msqrtals	16	scosine	10
arglina	10	cliff	2	dixon	10	helix	3	msqrtbls	16	sisser	2
arglinb	10	clplatea	16	dqartic	10	hilbert	10	morebv	12	spmsqrt	10
arglinc	10	clplateb	16	edensch	10	himln3	2	nlminsurf	16	tcontact	49
argtrig	10	clustr	2	eg2	10	himm25	2	nondquar	10	trigger	7
arwhead	10	cosine	10	eg2s	10	himm27	2	nzf1	13	tridia	10
bard	3	crglvy	4	eigfenals	12	himm28	2	osbornea	5	tlminsurfx	16
bdarwhd	10	cube	2	eigenbls	12	himm29	2	osborneb	11	tnlminsurfx	16
beale	2	curly10	10	eigencls	12	himm30	3	penalty1	10	vardim	10
biggs5	5	dixmaana	12	engval1	10	himm32	4	penalty2	10	vibrbeam	8
biggs6	6	dixmaanb	12	engval2	3	himm33	2	penalty3	10	watson	12
brownden	4	dixmaanc	12	expfit	2	hypcir	2	powellbs	2	wmsqrtals	16
booth	2	dixmaand	12	extrosnb	10	indef	10	powellsg	12	wmsqrtbls	16
box3	3	dixmaane	12	fminsurf	16	integreq	10	powellsq	2	woods	12
brkmcc	2	dixmaanf	12	freuroth	4	jensmp	2	powr	10	yfitu	3
brownal	10	dixmaang	12	genhumps	5	kowosb	4	recipe	2	zangwill2	2
brownbs	2	dixmaanh	12	gottfr	2	lminsurg	16	rosenbr	10	zangwill3	3
broyden3d	10	dixmaani	12	gulf	4	macino	10	sensors	10		
broydenbd	10	dixmaanj	12	hairy	2	mexhat	2	schmvett	3		
chandheu	10	dixmaank	12	heart6ls	6	meyer3	3	scurly10	10		

Table 1: The OPM test problems and their dimension

(sometimes quite wild) non-monotonicity of the methods considered here has another practical consequence: it happens on several nonconvex problems[(10)] that convergence of different algorithmic variants occurs to points with gradient norm within termination tolerance (the methods are thus achieving their objective), but these points can be quite far apart and have very different function values. It is therefore impossible to meaningfully compare the convergence performance to such points across algorithmic variants. This does reduce the set of problems where several variants can be compared.

We discuss the results of our tests from the efficiency and reliability points of view. Efficiency is measured in number of derivatives' evaluations (or, equivalently, iterations)[(11)]: the fewer evaluations the more efficient the algorithm. In addition to presenting the now standard performance profile [11] for our selection of algorithms in Figure 3, we follow [25] and consider the derived "global" measure $\pi_{\texttt{algo}}$ to be $\frac{1}{50}$ of the area below the curve corresponding to `algo` in the performance profile, for abscissas in the interval $[1, 50]$. The larger this area and closer $\pi_{\texttt{algo}}$ to one, the closer the curve to the right and top borders of the plot and the better the global performance. When reporting reliability, we say that the run of an algorithmic variant on a specific test problem is successful if the gradient norm tolerance has been achieved, or if the final relative error on the objective-function value is below 10^{-7} or, should the optimal value be below 10^{-7}, if the final absolute error is below 10^{-7}. These last two criteria were applied to instances of a total of 21 problems[(12)]. In what follows, $\rho_{\texttt{algo}}$ denotes the percentage of successful runs taken on all problems were comparison is meaningful. Table 2 presents the values of these statistics in two columns: for easier reading, the variants are sorted by

[(10)] broyden3d, broydenbd, curly10, gottfr, hairy, indef, jensmp, osborneb, sensors, wmsqrtals, wmsqrtbls, woods.

[(11)] For `sdba`, gradient and objective-function evaluations.

[(12)] biggs6, brownden, box3, chebyqad, crglvy, cube, dixmaanb, dixmaanh, dixmaani, dixmaanj, dixmaanl, edensch, engval2, freuroth, indef, msqrtbls, osborneb, powellsq, rosenbr, vardim, zangwil3.

algo	$\pi_{\texttt{algo}}$	$\rho_{\texttt{algo}}$	algo	$\pi_{\texttt{algo}}$	$\rho_{\texttt{algo}}$
lmadagi3b	0.72	74.79	adagi1	0.70	75.63
adagi1	0.70	75.63	b1adagi1	0.68	74.79
maxgi01	0.70	71.43	lmadagi3b	0.72	74.79
sdba	0.69	73.95	sdba	0.69	73.95
b1adagi1	0.68	74.79	maxgi01	0.70	71.43
Eadagi1	0.60	67.23	adag1	0.54	68.91
adag1	0.54	68.91	Eadagi1	0.60	67.23
adag2	0.52	36.13	maxg01	0.52	63.03
maxg01	0.52	63.03	adag2	0.52	36.13
adagi2	0.51	31.09	adagi2	0.51	31.09

Table 2: Performance statistics for deterministic OFFO algorithms on OPM problems

decreasing global performance ($\pi_{\texttt{algo}}$) in the first, and by decreasing reliability ($\rho_{\texttt{algo}}$) in the second.

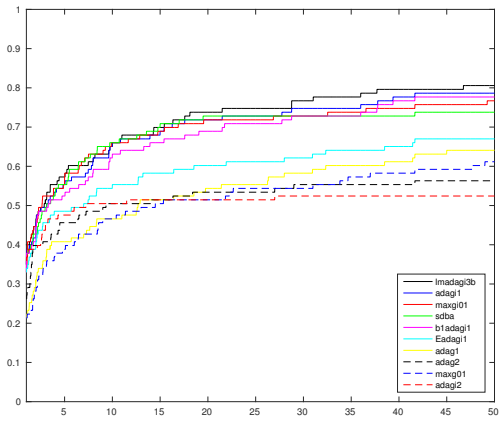

Figure 3: Performance profile for deterministic OFFO algorithms on OPM problems

A total of 22 problems[(13)] could not be successfully solved by any of the above algorithms, we believe mostly because of ill-conditioning.

The authors are of course aware that the very limited experiments presented here do not replace extended numerical practice and could be completed in various ways. They nevertheless suggest the following (very tentative) comments.

1. There seems to be a definite advantage in using the $\|\cdot\|_\infty$ norm over $\|\cdot\|_2$, as can be seen by comparing `adag1` with `adagi1`, `adag2` with `adagi2`, and `maxg01` with `maxgi01`. While this may be due in part to the fact that the trust region in ℓ_∞ norm is larger than that in ℓ_2 norm (and thus allows larger steps), it is also the case that the disaggregate

[(13)] biggs5, brownal, brownbs, cliff, eg2s, genhumps, gulf, heart8ls, himm29, mexhat, meyer3, nondquar, osbornea, penalty2, powellbs, powellsg, scurly10, scosine, trigger, vibrbeam, watson, yfitu.

definition of the scaling factors $w_{i,k}$ ((3.1), (3.36) or (4.1)) used in conjunction with the ℓ_∞ norm may allow a better exploitation of differences of scale between coordinates.

2. Among the "purely first-order" methods, `sdba`, `maxgi01` and `adagi1` are almost undistinguishable form the performance point of view, with a small reliability advantage for `adagi1` (Adagrad). This means that, at least in those experiments, the suggestion resulting from the theory that OFFO methods may perform comparably to standard first-order methods seems vindicated.

3. The Adam variants (`adag2` and `adagi2`) are clearly outperformed in our tests by the Adagrad ones (`adag1` and `adagi1`). We recall that analytical examples where Adam fails do exist, while the convergence of Adagrad is covered by our theory.

4. The theoretical difference in global rate of convergence between `adagi1` and `maxgi01` does not seem to have much impact on the relative performance of these two methods.

5. The use of limited memory Hessian approximation (`lmadagi1`) appears to enhance the performance of `adagi1`, but this is not the base of the Barzilai-Borwein approximation (`b1adagi1`) or, remarkably, for the use of the exact Hessian (`Eadagi1`). When these methods fail, this is often because the steplength is too small to allow the truncated conjugate-gradient solver to pick up second-order information in other directions than the negative gradient. What favours the limited memory approach remains unclear at this stage.

Finally, and although this is a slight digression from the paper's main topic, we report in Table 3 how reliability of our selection of OFFO variants is impacted by noise. To obtain these results, we ran the considered methods on all test problems where the evaluations (function[(14)] and derivatives) are contaminated by 5, 15, 25 or 50 % of relative Gaussian noise with unit variance. The reliability percentages in the table result from averaging ten sets of independent runs.

	$\rho_{\texttt{algo}}$/relative noise level				
algo	0%	5%	15%	25%	50%
adagi1	75.63	70.42	70.34	72.02	72.77
b1adagi1	74.79	75.38	75.13	75.38	75.88
lmadagi3b	74.79	74.79	70.67	71.34	71.09
sdba	73.95	34.29	35.04	36.30	36.89
maxgi01	71.43	70.59	70.42	73.61	74.45
adag1	68.91	64.12	68.91	71.09	71.01
Eadagi1	67.23	67.98	68.74	70.17	70.08
maxg01	63.03	63.28	59.50	61.60	63.70
adag2	36.13	28.74	30.76	39.16	42.35
adagi2	31.09	24.12	25.88	28.66	31.93

Table 3: Reliability of OFFO algorithms as a function of the relative Gaussian noise level

[(14)] For `sdba`.

As can be seen in the table, the reliability of the `sdba` methods dramatically drops as soon as noise is present, while that of the other OFFO methods is barely affected and remains globally unchanged for increasing noise levels. This is consistent with widespread experience in the deep learning context, where noise is caused by sampling among the very large number of terms defining the objective function. This observation vindicates the popularity of methods such as Adagrad in the noisy context and suggests that the new OFFO algorithms may have some practical potential.

6 Conclusions

We have presented a parametric class of deterministic "trust-region minded" extensions of the Adagrad method, allowing the use of second-order information, should it be available. We then prove that, for OFFO algorithms in this class, $\min_{j\in\{0,\ldots,k\}} \|g_j\| = \mathcal{O}(1/\sqrt{k+1})$. This bound, which we have shown to be essentially sharp, is *identical to the global rate of convergence of standard first-order methods using both objective-function and gradient evaluations*, despite the fact that the latter exploit significantly more information. Thus, *if one considers the order of global convergence only, evaluating the objective-function values is an unnecessary effort.* We have also considered another class of OFFO algorithms inspired by the "divergent stepsize" paradigm in non-smooth convex optimization and have provided an essentially sharp (but slighlty worse) global rate of convergence for this latter class. Limited numerical experiments suggest that the above theoretical conclusions may translate to practice and remain, for OFFO methods, relatively independent of noise.

Although discussed here in the context of unconstrained optimization, adaptation of the above OFFO algorithms to problems involving convex constraints (such as bounds on the variables) is relatively straightforward and practical: one then needs to intersect the trust-region with the feasible set and minimize the quadratic model in this intersection (see [8, Chapter 12]). It will be also of interest to further analyze the possible links between our proposals and those of [16], both from the theoretical and practical perspectives.

References

[1] J. Barzilai and J. Borwein. Two-point step size gradient method. *IMA Journal of Numerical Analysis*, 8:141–148, 1988.

[2] A. Beck. *First-order Methods in Optimization*. Number 25 in MOS-SIAM Optimization Series. SIAM, Philadelphia, USA, 2017.

[3] C. Cartis, N. I. M. Gould, and Ph. L. Toint. On the complexity of steepest descent, Newton's and regularized Newton's methods for nonconvex unconstrained optimization. *SIAM Journal on Optimization*, 20(6):2833–2852, 2010.

[4] C. Cartis, N. I. M. Gould, and Ph. L. Toint. On the oracle complexity of first-order and derivative-free algorithms for smooth nonconvex minimization. *SIAM Journal on Optimization*, 22(1):66–86, 2012.

[5] C. Cartis, N. I. M. Gould, and Ph. L. Toint. Worst-case evaluation complexity and optimality of second-order methods for nonconvex smooth optimization. In B. Sirakov, P. de Souza, and M. Viana, editors, *Invited Lectures, Proceedings of the 2018 International Conference of Mathematicians (ICM 2018), vol. 4, Rio de Janeiro*, pages 3729–3768. World Scientific Publishing Co Pte Ltd, 2018.

[6] C. Cartis, N. I. M. Gould, and Ph. L. Toint. *Evaluation complexity of algorithms for nonconvex optimization*. Number 30 in MOS-SIAM Series on Optimization. SIAM, Philadelphia, USA, June 2022.

[7] I. Chatzigeorgiou. Bounds on the Lambert function and their application to the outage analysis of user cooperation. *IEEE Communications Letters*, 17(8):1505–1508, 2013.

[8] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. *Trust-Region Methods*. Number 1 in MOS-SIAM Optimization Series. SIAM, Philadelphia, USA, 2000.

[9] R. M. Corless, G. H. Gonnet, D. E. Hare, D. J. Jeffrey, and D. E. Knuth. On the Lambert W function. *Advances in Computational Mathematics*, 5:329–359, 1996.

[10] A. Défossez, L. Bottou, F. Bach, and N. Usunier. A simple convergence proof for Adam and Adagrad. arXiv:2003.02395v2, 2020.

[11] E. D. Dolan, J. J. Moré, and T. S. Munson. Optimality measures for performance profiles. *SIAM Journal on Optimization*, 16(3):891–909, 2006.

[12] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12, July 2011.

[13] J. Fan and Y. Yuan. A new trust region algorithm with trust region radius converging to zero. In D. Li, editor, *Proceedings of the 5th International Conference on Optimization: Techniques and Applications (ICOTA 2001, Hong Kong)*, pages 786–794, 2001.

[14] N. I. M. Gould, S. Lucidi, M. Roma, and Ph. L. Toint. Solving the trust-region subproblem using the Lanczos method. *SIAM Journal on Optimization*, 9(2):504–525, 1999.

[15] G. N. Grapiglia. Quadratic regularization methods with finite-difference gradient approximations. Optimization Online, November 2021.

[16] G. N. Grapiglia and G. F. D. Stella. An adaptive trust-region method without function evaluation. Optimization Online, February 2022.

[17] S. Gratton, S. Jerad, and Ph. L. Toint. Parametric complexity analysis for a class of first-order Adagrad-like algorithms. arXiv:2203.01647, 2022.

[18] S. Gratton, A. Sartenaer, and Ph. L. Toint. Recursive trust-region methods for multiscale nonlinear optimization. *SIAM Journal on Optimization*, 19(1):414–444, 2008.

[19] S. Gratton and Ph. L. Toint. OPM, a collection of optimization problems in Matlab. arXiv:2112.05636, 2021.

[20] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proceedings in the International Conference on Learning Representations (ICLR)*, 2015.

[21] D. C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming, Series B*, 45(1):503–528, 1989.

[22] J. J. Moré and G. Toraldo. Algorithms for bound constrained quadratic programming problems. *Numerische Mathematik*, 14:14–21, 1989.

[23] J. J. Moré and G. Toraldo. On the solution of large quadratic programming problems with bound constraints. *SIAM Journal on Optimization*, 1(1):93–113, 1991.

[24] Yu. Nesterov. *Introductory Lectures on Convex Optimization*. Applied Optimization. Kluwer Academic Publishers, Dordrecht, The Netherlands, 2004.

[25] M. Porcelli and Ph. L. Toint. A note on using performance and data profiles for training algorithms. *ACM Transactions on Mathematical Software*, 45(2):1–25, 2019.

[26] S. Reddi, S. Kale, and S. Kumar. On the convergence of Adam and beyond. In *Proceedings in the International Conference on Learning Representations (ICLR)*, 2018.

[27] T. Tieleman and G. Hinton. Lecture 6.5-RMSPROP. COURSERA: Neural Networks for Machine Learning, 2012.

[28] R. Ward, X. Wu, and L. Bottou. Adagrad stepsizes: sharp convergence over nonconvex landscapes. In *Proceedings in the International Conference on Machine Learning (ICML2019)*, 2019.

[29] Y. Yuan. Recent advances in trust region algorithms. *Mathematical Programming, Series A*, 151(1):249–281, 2015.

[30] D. Zhou, Y. Tang, Z. Yang, Y. Cao, and Q. Gu. On the convergence of adaptive gradient methods for nonconvex optimization. In *Proceedings of OPT2020: 12th Annual Workshop on Optimization for Machine Learning*, 2020.