Accelerated gradient descent: A guaranteed bound for a heuristic restart strategy

Walaa M. Moursi[*] **Viktor Pavlovic**[†] **Stephen A. Vavasis**[‡]

Abstract

The $\mathcal{O}(1/k^2)$ convergence rate in function value of accelerated gradient descent is optimal, but there are many modifications that have been used to speed up convergence in practice. Among these modifications are restarts, that is, starting the algorithm with the current iteration being considered as the initial point. We focus on the adaptive restart techniques introduced by O'Donoghue and Candès, specifically their gradient restart strategy. While the gradient restart strategy is a heuristic in general, we prove that applying gradient restarts preserves and in fact improves the $\mathcal{O}(1/k^2)$ bound, hence establishing function value convergence, for one-dimensional functions. Applications of our results to separable and nearly separable functions are presented.

1. Introduction

In 1983, Nesterov introduced acceleration to the classical gradient descent method [10]. This acceleration method involves the addition of an extrapolation based on previous iterates, the new iterate is then computed as a classical gradient step from this extrapolation. Accelerated gradient descent (AGD) is also known as gradient descent with momentum. This is because the extrapolation step is computed using an increasing sequence of scalars. Due to this added momentum AGD is a non-monotonic method as extrapolated points can overshoot the minimizer and cause an increase in function value.

Several restarting algorithms have been proposed recently as discussed in Section 2.1. Restarting means that the momentum term is set to its initial value of zero, and the current iteration is used as the new starting point. In effect, this deletes the memory used previously to compute the extrapolation steps.

O'Donoghue and Candès introduced an adaptive restart strategy that does not depend on any properties of the objective function [11]. The authors offer two heuristic schemes to restart AGD. The first is a function value scheme that restarts whenever non-monotonicity is detected, and the second restarts when the gradient makes an obtuse angle with the direction of the iterates. The latter scheme uses only already computed information. The authors suggest that the gradient based scheme is favorable as it is more numerically stable. While both schemes perform well in practice and tend

[*] Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada. E-mail: walaa.moursi@uwaterloo.ca

[†] Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada. E-mail: v2pavlov@uwaterloo.ca

[‡] Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada. E-mail: vavasis@uwaterloo.ca.

© .

to drastically improve the performance of AGD, unlike AGD, there is no proof of function value convergence for the restarted scheme. The authors have provided an analysis when the objective is quadratic and suggest that this analysis carries over to a quadratic region around the minimizer. There have however been examples, such as in the work by Hinder and Lubin [8] who give a simple function for which the restarts drastically underperform until the iterates get close to the minimizer [8]. Recall that a function $f : \mathbb{R}^n \to \mathbb{R}$ is *L-smooth* if it has Lipschitz continuous gradient, i.e., for all x and y in $\mathbb{R}^n$ we have,

$$||\nabla f(x) - \nabla f(y)|| \leq L||x - y||. \tag{1}$$

Throughout the remainder of this paper, we assume that

$$f : \mathbb{R}^n \to \mathbb{R} \text{ is convex and } L\text{-smooth, where } L > 0, \text{ and } S := \operatorname{Argmin} f \neq \varnothing \tag{2}$$

and we set

$$f^* := \min f(\mathbb{R}^n). \tag{3}$$

In this paper we provide analysis for the case $n = 1$. Our results, primarily Theorem 8 and Theorem 9 below, reveal that applying the gradient restart improves the classical right-hand side bound of AGD and hence preserves the function value convergence of the iterates of the restarted scheme. Unlike the O'Donoghue and Candès restart scheme, our analysis varies in that if the restart condition (see (16) below) is satisfied then we restart at x_{k+1} rather than x_k. Computational experiments (see Appendix D and Appendix E) suggest that in practice restarting from x_{k+1} offers slightly better practical performance for $n \geq 1$.

2. Framework and Related Works

Recalling (2), our framework for AGD is that of an unconstrained convex optimization problem

$$\min_{x \in \mathbb{R}^n} f(x). \tag{4}$$

Let $(t_k)_{k \in \mathbb{N}}$ be a sequence of scalars such that $t_0 = 1$ and $(\forall k \geq 1)$ (see, e.g., [5])

$$t_k \geq \tfrac{k+2}{2} \geq 1 = t_0 \text{ and } t_k^2 \geq t_{k+1}^2 - t_{k+1}. \tag{5}$$

Two popular choices for the sequence of t_k are given by,

$$t_{k+1} = \tfrac{1+\sqrt{1+4t_k^2}}{2}, \quad t_k = \tfrac{k+2}{2}.t_{k+1} = \tfrac{1+\sqrt{1+4t_k^2}}{2}, \quad t_k = \tfrac{k+2}{2}. \tag{6}$$

Let $x_0 \in \mathbb{R}^n$, and set $y_0 = x_0$. Throughout this paper we set

$$T = \mathrm{Id} - \tfrac{1}{L}\nabla f. \tag{7}$$

Classical AGD updates x_0 $(\forall k \in \mathbb{N})$ as follows:

$$x_{k+1} := T(y_k) \tag{8a}$$

$$y_{k+1} := x_{k+1} + \tfrac{t_k - 1}{t_{k+1}}\left(x_{k+1} - x_k\right). \tag{8b}$$

We set

$$x^* := P_S(x_0) \in S, \tag{9}$$

where P_S denotes the orthogonal projection (this is also known as the closest point mapping) onto the set S which is convex, closed, and by assumption (see (2)) nonempty.

Note that the classical upper bound for AGD is ($\forall k \in \mathbb{N}$)

$$f(x_k) - f^* \leq \frac{2L||x_0 - x^*||^2}{(k+1)^2}. \tag{10}$$

We now recall the following useful fact which will be used later.

Fact 1 (see [4, Theorem 10.16]) *Let $x \in \mathbb{R}^n$ and let $y \in \mathbb{R}^n$. Then*

$$f(x) - f(T(y)) \geq \tfrac{L}{2}||x - T(y)||^2 - \tfrac{L}{2}||x - y||^2. \tag{11}$$

2.1. Restarts

We will focus on adaptive restart strategies. These strategies restart the algorithm according to whether some condition is satisfied at the current iteration. In [11] O'Donoghue and Candès proposed two conditions for restarts: the *function value* and *gradient based* conditions. The function value condition imposes restarts whenever

$$f(x_{k+1}) > f(x_k). \tag{12}$$

On the other hand, the gradient based condition imposes restarts whenever

$$\langle \nabla f(y_k), x_{k+1} - x_k \rangle > 0. \tag{13}$$

This indicates that there is a disagreement in the direction the iterates should go in, and thus it would be good to restart. The function scheme requires evaluating the objective at x_k which may be expensive, while the gradient based scheme requires no new computations whatsoever.

Giselsson and Boyd introduced different adaptive restart schemes [7]. Their framework considers the composite model where the objective function is of the form $F = f + g$ where g is convex, lower semicontinuous, and proper. This is the setup for FISTA [5]. Note that AGD can be seen as a special case of FISTA when we set $g \equiv 0$. They suggest that non-monotonicity of function values is a good indicator of when to restart, and provide a new condition which implies non-monotonicity, namely

$$\left\langle y_{k-1} - x_k, x_{k+1} - \tfrac{1}{2}(x_k + y_{k-1}) \right\rangle > 0. \tag{14}$$

In addition to this new restart test, they provide a convergence rate to a modified version of the restarted AGD algorithm. A notable difference between their restart scheme and the ones introduced in [11] is that they do not reset the parameter sequence of t_k's. This allowed the authors to deduce an $O(1/k^2)$ convergence rate similar to classical AGD.

Another common assumption is that of quadratic growth [9], [1], [6] and [2]. We say that f satisfies a quadratic growth condition if there exists $\mu > 0$ such that

$$f(x) - f^* \geq \frac{\mu}{2}||x - P_S(x)||^2. \tag{15}$$

Necoara, Nesterov, and Glineur provide a fixed restart strategy for such functions [9]. They derive the optimal restart frequency based on the knowledge of the parameter μ.

The growth parameter μ is rarely known exactly, and thus many restart strategies for this class of functions work by estimating it. In Feroq and Qu [6], the authors show that their adaptive restart strategy which requires an estimation of the growth parameter, satisfies a linear convergence rate. In Aujol et al. [2] the authors use restarting and adaptive backtracking to estimate both L and μ and prove a linear convergence rate without any prior knowledge of the actual values of the parameters, thus they are able to run FISTA without any parameters.

Hinder and Lubin provide another adaptive restart strategy based on a potential function that is always computable [8, Theorem 3]. Their analysis requires the assumption that f is strongly convex, and this assumption cannot be reduced to quadratic growth. According to their computational experiments, their restart method is comparable to the heuristic methods in [11] in general. However, they provided an example of a strongly convex and $L-$smooth separable function on $\mathbb{R}^n$ where the gradient based restarts (13) performed worse than classical AGD while their proposed restart scheme performed better. In passing, we point out that our analysis identifies the drawback of applying restart condition (13) in this case. As a consequence of our analysis, we show that for separable functions the condition (13) should be applied in a separable way by checking (13) for each coordinate.

3. Contributions

Recalling (2), we provide analysis for the case $n = 1$. We focus only on the gradient restart condition (13). In the one-dimensional case this condition boils down to

$$f'(y_k)(x_{k+1} - x_k) > 0. \tag{16}$$

Observe that if for some $k \in \{0, 1, 2 \dots\}$ we had $f'(x_k) = 0$, then x_k is a minimizer and we stop the algorithm.

Remark 2 *In passing we point out that in our scheme, we employ the restart condition (16) with a slight modification to that proposed in [11]. Indeed, if (16) is satisfied our algorithm keeps x_{k+1} the same, it resets $y_{k+1} := x_{k+1}$ and $t_{k+1} := 1$. Note that this is different from the scheme proposed in [11] which sets $x_{k+1} := x_k, y_{k+1} := x_k, t_{k+1} := 1$ whenever (16) is satisfied. We have observed that our modification offers slightly better performance with the restarts (see Fig. 1 and Fig. 2 below).*

Our proposed restart algorithm in the one-dimensional case is given in Algorithm 1 below.

Algorithm 1: AGD with Restarts
Input: $x_0 \in \mathbb{R}$, $(\widetilde{t}_k)_{k\in\mathbb{N}}$ is a sequence of scalars that satisfies (5).
Initialization: $y_0 = x_0$, $(t_k)_{k\in\mathbb{N}} := (\widetilde{t}_k)_{k\in\mathbb{N}}$.
General step: For $k \geq 0$ update x_k via

$$x_{k+1} := T(y_k)$$

If

$$f'(y_k)(x_{k+1} - x_k) \leq 0$$

update y_k via

$$y_{k+1} := x_{k+1} + \tfrac{t_k - 1}{t_{k+1}}\left(x_{k+1} - x_k\right).$$

Else restart as follows: $y_{k+1} := x_{k+1}$, $(t_{k+m+1})_{m\in\mathbb{N}} := (\widetilde{t}_m)_{m\in\mathbb{N}}$.

Proposition 3 *Let $x_0 \in \mathbb{R}$ and let $(x_k)_{k\in\mathbb{N}}$ and $(y_k)_{k\in\mathbb{N}}$ be the sequences obtained by Algorithm 1. Let $k \in \mathbb{N}$ be such that for all $\overline{k} \leq k$ the restart condition (16) has not been satisfied. Then the following hold.*

(i) If $x_k < x^*$ then the restart condition in (16) is satisfied at iteration k if and only if $x_k < x^* < x_{k+1}$.

(ii) If $x_k > x^*$ then the restart condition in (16) is satisfied at iteration k if and only if $x_k > x^* > x_{k+1}$.

Proof See Appendix A. ■

Remark 4 *A direct consequence of Proposition 3 is that we get an ordering of the iterates x_k between restarts. Indeed, suppose we have two restarts, r_1 and r_2 with $r_2 > r_1$. If $y_{r_1} < x^*$ then for all $k \in \{r_1, \dots r_2 - 1\}$ we have $f'(y_k) < 0$. Since we do not restart at any of these iterations we must have that $x_{k+1} - x_k > 0$. Similarly, if $y_{r_1} > x^*$ we deduce that $x_{k+1} - x_k < 0$.*

We will also state the following proposition:

Proposition 5 *Let $x_0 \in \mathbb{R}$ and let $(x_k)_{k\in\mathbb{N}}$ and $(y_k)_{k\in\mathbb{N}}$ be the sequences obtained by Algorithm 1. Let $k \in \mathbb{N}$ be such that for all $\overline{k} \leq k - 1$ the restart condition (16) has not been satisfied and that k is the first iteration where the restart condition is satisfied, i.e.,*

$$f'(y_k)(x_{k+1} - x_k) > 0. \tag{17}$$

Define $z_{k+1} := (1 - t_k)x_k + t_k x_{k+1}$. Then,

$$|z_{k+1} - x^*| \geq t_k |x_{k+1} - x^*|.$$

Proof Suppose that $x_0 < x^*$. By Proposition 3 we have $x_k < x^* < x_{k+1}$. It follows from this, the definition of z_{k+1} and (5) that

$$\begin{aligned} |z_{k+1} - x^*| &= |(1-t_k)x_k + t_k x_{k+1} - x^*| \\ &= |(1-t_k)(x_k - x^*) + t_k(x_{k+1} - x^*)| \\ &= (t_k - 1)|x_k - x^*| + t_k|x_{k+1} - x^*| \\ &\geq t_k|x_{k+1} - x^*|. \end{aligned} \tag{18}$$

The case where $x_0 > x^*$ is shown similarly. ■

Remark 6 *A careful look at the proof of Proposition 5 in view of Remark 4 reveals that the conclusion of Proposition 5 generalizes to any number of restarts.*

We now recall the following useful fact.

Fact 7 (see, e.g., [3, Theorem 30.4]) *Let $x_0 \in \mathbb{R}$ and let $(x_k)_{k\in\mathbb{N}}$, and $(t_k)_{k\in\mathbb{N}}$ be given by AGD (8). For $k \in \{0, 1, 2 \dots\}$ we set*

$$z_{k+1} := (1-t_k)x_k + t_k x_{k+1}, \tag{19a}$$

$$\delta_k := f(x_k) - f(x^*). \tag{19b}$$

Then we have the following monotonicity property

$$t_k^2\delta_{k+1} + \tfrac{L}{2}|z_{k+1} - x^*|^2 \leq t_{k-1}^2\delta_k + \tfrac{L}{2}|z_k - x^*|^2. \tag{20}$$

We are now ready for our main result. For restarts using the gradient condition (16), we have the following theorem.

Theorem 8 (a single restart.) *Let $x_0 \in \mathbb{R}$ and let $(x_k)_{k\in\mathbb{N}}, (y_k)_{k\in\mathbb{N}}, (t_k)_{k\in\mathbb{N}}$ be given by Algorithm 1. Suppose that iteration r is the first iteration where we have*

$$f'(y_r)(x_{r+1} - x_r) > 0, \tag{21}$$

and that iteration $\overline{r}$ is the second iteration where we have $f'(y_{\overline{r}})(x_{\overline{r}+1} - x_{\overline{r}}) > 0$. Then the following hold.

(i) $(\forall k \leq r+1)$ *we have* $f(x_k) - f^* \leq \frac{2L(x_0-x^*)^2}{(k+1)^2}$.

(ii) $f(x_{r+2}) \leq f(x_{r+1})$.

(iii) $(\forall k \in \{r+3, \dots, \overline{r}+1\})$ *we have*

$$f(x_k) - f^* \leq \left(\frac{4}{(k-r)(r+2)}\right)^2 \frac{L(x_0-x^*)^2}{2} \leq \frac{2L(x_0-x^*)^2}{(k+1)^2}. \tag{22}$$

Proof See Appendix B. ■

Theorem 9 (multiple restarts.) *Let $x_0 \in \mathbb{R}$ and set $(\forall k \in \mathbb{N})\ t_k = \frac{k+1}{2}$. Let $(x_k)_{k\in\mathbb{N}}$ be given by Algorithm 1. Let $p \in \mathbb{N}$. Suppose that r_p is the pth-iteration such that*

$$f'(y_{r_p})(x_{r_p+1} - x_{r_p}) > 0. \tag{23}$$

Then $(\forall p \geq 2)\ (\forall k \in \{r_p + 2, \ldots, r_{p+1} + 1\})$ we have

$$f(x_k) - f(x^*) \leq \left(\frac{2^{p+1}}{(k - r_p)(r_p - r_{p-1} + 2)...(r_2 - r_1 + 2)(r_1 + 2)} \right)^2 \frac{L}{2}(x_0 - x^*)^2 \tag{24a}$$

$$\leq \frac{2L(x_0 - x^*)^2}{(k+1)^2}. \tag{24b}$$

Proof See Appendix C. ■

Remark 10 *For sufficiently large k and r, the upper bound of $8/((k-r)(r+2))^2$ in (22) can be much smaller than the classical factor $2/(k+1)^2$ for $k > r$. For example, if $r = 10,000$, $k = 11,000$, then the factor in our new bound is $\approx 8 \cdot 10^{-14}$ versus $2/(k+1)^2 \approx 1.7 \cdot 10^{-8}$. The right-hand side of (24a) further improves (22).*

Corollary 11 *Let $x_0 \in \mathbb{R}$ and set $(\forall k \in \mathbb{N})\ t_k = \frac{k+1}{2}$. Let $(x_k)_{k\in\mathbb{N}}$ be given by Algorithm 1 . Then*

$$f(x_k) \to f^*. \tag{25}$$

Proof This is a direct consequence of Theorem 8 and Theorem 9. ■

4. Conclusion

In our main result, we prove that the gradient based restarts of O'Donoghue and Candès [11] employed with a slight modification (see Remark 2) improve the classical right-hand side bound from AGD when $n = 1$. The modification in Remark 2 allowed the use of Proposition 5, and in computational experiments it performed slightly better, even in cases where $n \geq 1$.
One remark is that our proof translates to higher dimensions if one has a restart condition that implies $t_k||x_{k+1} - x^*|| \leq ||z_{k+1} - x^*||$. While this cannot be useful in practice because it requires prior knowledge of the minimizer, we have experimentally observed that in cases where we do know the minimizer this restart condition performs well. Giselsson and Boyd also noted that having $||x_k - x^*|| \leq ||z_k - x^*||$ would improve the constant in their convergence analysis. A direction of further research would be to create a restart condition which would imply this condition. Although the analysis of Algorithm 1 does not extend to higher dimensions, we want to remark on the case of nearly separable functions. For such functions, we observed that running Algorithm 1 in parallel along each coordinate can perform better than using the restarting algorithm of O'Donoghue and Candès as illustrated in Appendix F.

Acknowledgements

The research of WMM and SAV is supported by their Natural Sciences and Engineering Research Council of Canada Discovery Grants (NSERC-DG).

References

[1] Teodoro Alamo, Pablo Krupa, and Daniel Limon. Gradient based restart FISTA. In 2019 IEEE 58th Conference on Decision and Control (CDC), pages 3936–3941, 2019. doi: 10.1109/CDC40024.2019.9029983.

[2] Jean-François Aujol, Luca Calatroni, Charles Dossal, Hippolyte Labarrière, and Aude Rondepierre. Parameter-free FISTA by adaptive restart and backtracking. Arxiv preprint 2307.14323, 2023.

[3] Heinz H. Bauschke and Walaa M. Moursi. An Introduction to Convexity, Optimization, and Algorithms. Society for Industrial and Applied Mathematics, 2024.

[4] Amir Beck. First-Order Methods in Optimization. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2017. doi: 10.1137/1.9781611974997. URL https://epubs.siam.org/doi/abs/10.1137/1.9781611974997.

[5] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM Journal on Imaging Sciences, 2(1):183–202, 2009. doi: 10.1137/080716542. URL https://doi.org/10.1137/080716542.

[6] Olivier Fercoq and Zheng Qu. Adaptive restart of accelerated gradient methods under local quadratic growth condition. IMA Journal of Numerical Analysis, 39(4):2069–2095, 03 2019. ISSN 0272-4979. doi: 10.1093/imanum/drz007. URL https://doi.org/10.1093/imanum/drz007.

[7] Pontus Giselsson and Stephen Boyd. Monotonicity and restart in fast gradient methods. In 53rd IEEE Conference on Decision and Control, pages 5058–5063. IEEE, 2014.

[8] Oliver Hinder and Miles Lubin. A generic adaptive restart scheme with applications to saddle point algorithms. Arxiv preprint 2006.08484, 2020.

[9] I. Necoara, Yu. Nesterov, and F. Glineur. Linear convergence of first order methods for non-strongly convex optimization. Math. Program., 175(1):69–107, May 2019. ISSN 1436-4646. doi: 10.1007/s10107-018-1232-1.

[10] Yurii Nesterov. A method for solving the convex programming problem with convergence rate $O(1/k^2)$. Proceedings of the USSR Academy of Sciences, 269:543–547, 1983.

[11] Brendan O'Donoghue and Emmanuel Candès. Adaptive Restart for Accelerated Gradient Schemes. Found. Comput. Math., 15(3):715–732, June 2015. ISSN 1615-3383. doi: 10.1007/s10208-013-9150-3.

Appendix A.

Proof of Proposition 3. Let $j \geq 1$ and suppose that x_j and y_j are given by the updates in (8). Using the fact that $f'(x^*) = 0$ yields

$$(x_j - x^*)(y_{j-1} - x^*) = (y_{j-1} - x^*)^2 - \tfrac{1}{L} f'(y_{j-1})(y_{j-1} - x^*) \geq 0. \tag{26}$$

Moreover, because x_j is a gradient step from y_{j-1} of size $1/L$ we learn that

$$|x_j - x^*| \leq |y_{j-1} - x^*|. \tag{27}$$

(i): "$\Rightarrow$": Suppose that the gradient condition is satisfied at iteration k, i.e., $f'(y_k)(x_{k+1} - x_k) > 0$. Then for $j \leq k-1$ we have $f'(y_j)(x_{j+1} - x_j) \leq 0$. In particular, for $j = k-1$ we learn that

$$f'(y_{k-1})(x_k - x_{k-1}) \leq 0. \tag{28}$$

Because $x_k < x^*$, using (27) and (26) applied with j replaced by k we conclude that $y_{k-1} \leq x_k < x^*$. Therefore $f'(y_{k-1}) < 0$ and consequently by (28) we have

$$x_k \geq x_{k-1}. \tag{29}$$

We claim that $y_k > x^*$. Indeed, Algorithm 1 gives us,

$$y_k = x_k + \tfrac{t_k - 1}{t_{k+1}}(x_k - x_{k-1}). \tag{30}$$

Since t_k is an increasing sequence and $t_k \geq 1$ for all $k \geq 1$, alongside the fact that $x_k - x_{k-1} \geq 0$, we obtain that $y_k \geq x_k$. Suppose for eventual contradiction that $y_k < x^*$. Then the above arguments yields $f'(y_k) < 0$ and $x_{k+1} = y_k - \frac{1}{L} f'(y_k) > y_k \geq x_k$. This implies that

$$f'(y_k)(x_{k+1} - x_k) < 0$$

which is absurd in view of (17). Therefore we have that $y_k > x^*$ which, in turn, implies that $f'(y_k) > 0$ and that $x^* < x_{k+1} \leq y_k$. Hence we conclude $x_k < x^* < x_{k+1}$. "$\Leftarrow$": Suppose that $x_k < x^* < x_{k+1}$. Then (26) and (27) applied with j replaced by k and with j replaced by $k+1$ implies

$$y_{k-1} < x_k < x^* < x_{k+1} < y_k, \tag{31}$$

which immediately implies that

$$f'(y_k)(x_{k+1} - x_k) > 0. \tag{32}$$

Therefore the gradient restart condition is satisfied at iteration k. (ii): Proceed similarly to the proof of (i). ■

Appendix B.

Proof of Theorem 8. Observe that $r \geq 2$. In the following we let $(\overline{x}_k)_{k\in\mathbb{N}}$, $(\overline{y}_k)_{k\in\mathbb{N}}$, and $(\overline{t}_k)_{k\in\mathbb{N}}$ be given by the classical AGD update (8) with the starting point $\overline{x}_0 := x_0$. Clearly, $\overline{t}_0 = t_0 = 1$. Moreover, for all $k \leq r$ we have,

$$\overline{x}_k = x_k, \overline{t}_k = t_k. \tag{33}$$

Furthermore, since we keep the point after the restart iteration we have $\overline{x}_{r+1} = x_{r+1}$. Now since at iteration $r+1$ we reset the parameter sequence we have that now $t_{r+1} = 1$, which means that for $k \geq r+1$ we have the relation that $t_k = \overline{t}_{k-(r+1)}$. For $k \in \{0, 1, 2 \dots\}$ we set

$$\overline{z}_{k+1} := (1 - t_k)\overline{x}_k + \overline{t}_k \overline{x}_{k+1}, \tag{34a}$$

$$\overline{\delta}_k := f(\overline{x}_k) - f(x^*) \tag{34b}$$

and we set

$$z_{k+1} := (1 - t_k)x_k + t_k x_{k+1}, \tag{35a}$$

$$\delta_k := f(x_k) - f(x^*). \tag{35b}$$

Due to the restart,

$$z_{r+2} = (1 - t_{r+1})x_{r+1} + t_{r+1}x_{r+2} = x_{r+2}. \tag{36}$$

Let $k \geq r+2$. Using Fact 7 we have

$$\delta_k = f(x_k) - f(x^*) \leq \tfrac{1}{t_{k-1}^2}\left(\tfrac{L}{2}(z_k - x^*)^2 + t_{k-1}^2 \delta_k\right) \tag{37a}$$

$$\leq \tfrac{1}{t_{k-1}^2}\left(\tfrac{L}{2}(z_{k-1} - x^*)^2 + t_{k-2}^2 \delta_{k-1}\right) \tag{37b}$$

$$\leq \tag{37c}$$

$$\vdots \tag{37d}$$

$$\leq \tfrac{1}{t_{k-1}^2}\left(\tfrac{L}{2}(z_{r+2} - x^*)^2 + t_{r+1}^2 \delta_{r+2}\right) \tag{37e}$$

$$= \tfrac{1}{t_{k-1}^2}\left(\tfrac{L}{2}(x_{r+2} - x^*)^2 + \delta_{r+2}\right) \tag{37f}$$

where we now used in the equality that $t_{r+1} = 1$, and that $z_{r+2} = x_{r+2}$. Recall from Algorithm 1 that we update x_k by taking a gradient step from y_{k-1}.

$$x_{r+2} = y_{r+1} - \frac{1}{L} f'(y_{r+1})$$

but since we restarted we have that $y_{r+1} = x_{r+1}$, so

$$x_{r+2} = x_{r+1} - \frac{1}{L} f'(x_{r+1}).$$

This observation allows us to make further progress. On the one hand, recalling $y_{r+1} = x_{r+1}$, it follows from (27) applied with j replaced by $r+2$ that

$$(x_{r+2} - x^*)^2 \leq (x_{r+1} - x^*)^2. \tag{38}$$

On the other hand, because x_{r+2} is a gradient step from x_{r+1} with a proper step-length we have that $f(\overline{x}_{r+1}) = f(x_{r+1}) \geq f(x_{r+2}) > f(x^*)$ which, in view of (35b), implies that $\delta_{r+2} \leq \delta_{r+1} = \overline{\delta}_{r+1}$. Combining this and (37f) yields

$$\delta_k \leq \tfrac{1}{\overline{t}_{k-r-2}^2}\left(\tfrac{L}{2}(x_{r+1} - x^*)^2 + \overline{\delta}_{r+1}\right) \tag{39}$$

where we used that $x_{r+1} = \overline{x}_{r+1}$, and that $t_{k-1} = \overline{t}_{k-(r+1)-1} = \overline{t}_{k-r-2}$. Proposition 5 yields

$$\delta_k \leq \frac{1}{\overline{t}_{k-r-2}^2}\left(\frac{L}{2}\frac{1}{\overline{t}_r^2}(\overline{z}_{r+1} - x^*)^2 + \overline{\delta}_{r+1}\right) \tag{40a}$$

$$= \frac{1}{\overline{t}_{k-r-2}^2\overline{t}_r^2}\left(\frac{L}{2}(\overline{z}_{r+1} - x^*)^2 + \overline{t}_r^2\overline{\delta}_{r+1}\right) \tag{40b}$$

$$\leq \tag{40c}$$

$$\vdots \tag{40d}$$

$$\leq \frac{1}{\overline{t}_{k-r-2}^2\overline{t}_r^2}\left(\tfrac{L}{2}(\overline{z}_1 - x^*)^2 + \overline{\delta}_1\right) \tag{40e}$$

$$= \frac{1}{\overline{t}_{k-r-2}^2\overline{t}_r^2}\left(\tfrac{L}{2}(\overline{x}_1 - x^*)^2 + \overline{\delta}_1\right) \tag{40f}$$

$$\leq \frac{L}{2(\overline{t}_{k-r-2}\overline{t}_r)^2}\left(x_0 - x^*\right)^2 \tag{40g}$$

$$\leq \frac{8L}{((k-r)(r+2))^2}\left(x_0 - x^*\right)^2, \tag{40h}$$

where (40g) follows from Fact 1 applied with x replaced by x^* and y replaced by x_0 and (40g) follows from (5). To finish the proof we claim that for $k \geq r+3$ we have

$$\frac{4}{((k-r)(r+2))^2} \leq \frac{1}{(k+1)^2}. \tag{41}$$

Indeed, observe that

$$(41) \Leftrightarrow 2(k+1) \leq (k-r)(r+2) \tag{42a}$$

$$\Leftrightarrow r(k-(r+2)) \geq 2, \tag{42b}$$

which is always true for $k \geq r+3$ by recalling that $r \geq 2$. By definition, $x_{r+2} = T(x_{r+1})$ and hence $|x_{r+2} - x^*| \leq |x_{r+1} - x^*|$ and $f(x_{r+2}) \leq f(x_{r+1})$. ■

Appendix C.

Proof of Theorem 9. We introduce the following notation to identify the multiple restarts. We denote by $(x_j^{(r_i)})_{j\in\mathbb{N}}$ the sequence of iterates given by starting AGD with $x_0^{(r_i)} = x_{r_i+1}$ and $t_0^{(r_i)} = t_{r_i+1} = 1$. We can now identify that between two consecutive restarts say r_i and r_{i+1} we have that $(x_0^{(r_i)}, x_1^{(r_i)}, x_2^{(r_i)}, \ldots x_{r_{i+1}-r_i}^{(r_i)})$ coincide with $(x_{r_i+1}, x_{r_i+2}, x_{r_i+3}, \ldots, x_{r_{i+1}})$. For iterations $k > r_p$ we have that, $(x_0^{(r_p)}, x_1^{(r_p)}, \ldots, x_{k-r_p-1}^{(r_p)})$ coincide with with $(x_{r_p+1}, x_{r_p+2}, \ldots, x_k)$. Upholding the notation of (19) we similarly denote the corresponding values of $z^{(r_i)}$, $\delta^{(r_i)}$ and $t^{(r_i)}$. We proceed

similarly to the proof of Theorem 8. Let $k \geq r_p + 2$. Then

$$
\begin{aligned}
f(x_k) - f(x^*) &= f(x_{k-r_p-1}^{(r_p)}) - f(x^*) && \text{(43a)}\\
&\leq \frac{1}{(t_{k-r_p-2}^{(r_p)})^2}\left(\frac{L}{2}(z_{k-r_p-1}^{(r_p)} - x^*)^2 + (t_{k-r_p-2}^{(r_p)})^2 \delta_{k-r_p-1}^{(r_p)}\right) && \text{(43b)}\\
&\leq && \text{(43c)}\\
&\vdots \\
&\leq \frac{1}{(t_{k-r_p-2}^{(r_p)})^2}\left(\frac{L}{2}(z_1^{(r_p)} - x^*)^2 + (t_0^{(r_p)})^2 \delta_1^{(r_p)}\right) && \text{(43d)}\\
&= \frac{1}{(t_{k-r_p-2}^{(r_p)})^2}\left(\frac{L}{2}(x_1^{(r_p)} - x^*)^2 + \delta_1^{(r_p)}\right). && \text{(43e)}
\end{aligned}
$$

We now use that $x_1^{(r_p)}$ is a gradient step from $x_0^{(r_p)}$ to conclude,

$$
\begin{aligned}
f(x_k) - f(x^*) &\leq \frac{1}{(t_{k-r_p-2}^{(r_p)})^2}\left(\frac{L}{2}(x_0^{(r_p)} - x^*)^2 + \delta_0^{(r_p)}\right) && \text{(44a)}\\
&= \frac{1}{(t_{k-r_p-2}^{(r_p)})^2}\left(\frac{L}{2}(x_{r_p-r_{p-1}+1}^{(r_p-1)} - x^*)^2 + \delta_{r_p-r_{p-1}+1}^{(r_p-1)}\right) && \text{(44b)}
\end{aligned}
$$

where we get equality because when the r_p restart gets triggered we keep $x_{r_p-r_{p-1}+1}^{(r_p-1)}$ unchanged and set $x_0^{(r_p)} = x_{r_p-r_{p-1}+1}^{(r_p-1)}$. We can now use Proposition 5 to obtain

$$
|x_{r_p-r_{p-1}+1}^{(r_p-1)} - x^*| \leq \frac{1}{t_{r_p-r_{p-1}}^{(r_p-1)}} |z_{r_p-r_{p-1}+1}^{(r_p-1)} - x^*|. \tag{45}
$$

Combining (44) and (45) and factoring out the $\frac{1}{t_{r_p-r_{p-1}}^{(r_p-1)}}$ term we obtain

$$
f(x_k) - f(x^*) \leq \frac{1}{(t_{k-r_p-2}^{(r_p)})^2 (t_{r_p-r_{p-1}}^{(r_p-1)})^2}\left(\frac{L}{2}(z_{r_p-r_{p-1}+1}^{(r_p-1)} - x^*)^2 + (t_{r_p-r_{p-1}}^{(r_p-1)})^2 \delta_{r_p-r_{p-1}+1}^{(r_p-1)}\right). \tag{46}
$$

We now repeat the same procedure. Since the terms in the parentheses on the right-hand side of (46) have the monotonicity property from Fact 7, we use it repeatedly until we obtain an expression of the form of the right-hand side of in (20) that features $z_1^{(r_i)}$. We then apply the fact that $z_1^{(r_i)} = x_1^{(r_i)}$, which is a gradient step from $x_0^{(r_i)}$, and finally use Proposition 5. Observe that every time we apply Proposition 5 an additional factor of $1/(t_{r_{i+1}-r_1}^{(r_i)})^2$ for some $i \in \{1, \ldots, p\}$ appears in the denominator of the right-hand side of the inequality (46). This is done p times until the term in the parentheses reduces to $(L/2)(x_0 - x^*)^2$. We therefore obtain

$$
f(x_k) - f(x^*) \leq \frac{1}{(t_{k-r_p-2}^{(r_p)})^2 \left(\prod_{i=1}^{p} (t_{r_i-r_{i-1}}^{(r_i)})^2\right)}\left(\frac{L}{2}(x_0 - x^*)^2\right). \tag{47}
$$

Recall that for each $i \in \{1, \dots, p\}$ $(\forall k \in \{0, \dots, r_i\})$ we have $t_k^{(r_i)} = (k+2)/2$, hence:

$$\frac{1}{t_{k-r_p-2}^{(r_p)}} = \frac{2}{k - r_p} \tag{48}$$

and

$$\frac{1}{t_{r_i - r_{i-1}}^{(r_i)}} = \frac{2}{r_i - r_{i-1} + 2}. \tag{49}$$

Now let $k \in \mathbb{N}$. Observe that if $k \leq r_2 + 1$ we obtain the $\mathcal{O}(1/k^2)$ function value convergence rate from Theorem 8. We now show that $(\forall p \geq 2)$ $(\forall k \geq r_p + 2)$ we have

$$\frac{2^{p+1}}{(k - r_p)(r_p - r_{p-1} + 2) \dots (r_2 - r_1 + 2)(r_1 + 2)} \leq \dots \leq \frac{4}{(k - r_1)(r_1 + 2)} \leq \frac{2}{(k+1)}. \tag{50}$$

We proceed by induction on p. We first verify the base case at $p = 2$. Applying (48) and (49) to (47) we get

$$f(x_k) - f(x^*) \leq \left(\frac{8}{(k - r_2)(r_2 - r_1 + 2)(r_1 + 2)} \right)^2 \frac{L}{2} (x_0 - x^*)^2. \tag{51}$$

For $k = r_2 + 2$ we have

$$\frac{8}{(k - r_2)(r_2 - r_1 + 2)(r_1 + 2)} = \frac{8}{(r_2 + 2 - r_2)(r_2 - r_1 + 2)(r_1 + 2)} \tag{52a}$$

$$= \frac{8}{(2)((r_2 + 2) - r_1)(r_1 + 2))} \tag{52b}$$

$$= \frac{4}{(k - r_1)(r_1 + 2)} \leq \frac{2}{k+1}, \tag{52c}$$

where the inequality follows from the fact that $r_1 \geq 2$ and $k = r_2 + 2 \geq r_1 + 3$. Now let $k \geq r_2 + 3$. Then

$$\frac{8}{(k - r_2)(r_2 - r_1 + 2)(r_1 + 2)} - \frac{4}{(k - r_1)(r_1 + 2)}$$

$$= \frac{4}{(r_1 + 2)} \left(\frac{2}{(k - r_2)(r_2 - r_1 + 2)} - \frac{1}{k - r_1} \right) \tag{53a}$$

$$= \frac{4}{(r_1 + 2)} \left(\frac{2(k - r_1)}{(k - r_2)(r_2 - r_1 + 2)(k - r_1)} - \frac{(k - r_2)(r_2 - r_1 + 2)}{(k - r_2)(r_2 - r_1 + 2)(k - r_1)} \right) \tag{53b}$$

$$= \frac{4}{(r_1 + 2)} \left(\frac{2(k - r_1) - (k - r_2)(r_2 - r_1 + 2)}{(k - r_2)(r_2 - r_1 + 2)(k - r_1)} \right). \tag{53c}$$

Write $k = r_2 + a$ where $a \geq 3$. It is sufficient to show that the numerator of the right-hand side of (53c) is nonpositive. To this end we have

$$2(k - r_1) - (k - r_2)(r_2 - r_1 + 2) = 2(r_2 + a - r_1) - (r_2 + a - r_2)(r_2 - r_1 + 2) \tag{54a}$$

$$= 2a(r_2 - r_1) - 2a - a(r_2 - r_1) \tag{54b}$$

$$= (2 - a)(r_2 - r_1) \leq 0. \tag{54c}$$

Combining (52), (53), and (54) we conclude that for $k \geq r_2 + 2$ we have

$$\frac{8}{(k-r_2)(r_2-r_1+2)(r_1+2)} \leq \frac{4}{(k-r_1)(r_1+2)} \leq \frac{2}{k+1}.$$

This proves the base case. Now suppose that we have restarted p times, $p \geq 2$, and for $k \geq r_p + 2$ the following inequalities hold

$$\frac{2^{p+1}}{(k-r_p)(r_p-r_{p-1}+2)\dots(r_2-r_1+2)(r_1+2)} \leq \dots \leq \frac{4}{(k-r_1)(r_1+2)} \leq \frac{2}{(k+1)}. \quad (55)$$

Consider the iterations $k \geq r_{p+1} + 2$ where r_{p+1} is the iteration of the $p+1$ restart. If we apply (48) and (49) to (47) we obtain,

$$f(x_k) - f(x^*) \leq \left(\frac{2^{p+2}}{(k-r_{p+1}) \prod_{i=1}^{p+1}(r_i - r_{i-1} + 2))} \right)^2 \frac{L}{2}(x_0 - x^*)^2.$$

We are concerned with iterations $k \geq r_{p+1}+2$ since those would be generated after the $p+1$ restart. At $k = r_{p+1} + 2$ we have

$$\frac{2^{p+2}}{(k-r_{p+1}) \prod_{i=1}^{p+1}(r_i - r_{i-1} + 2)} = \frac{2^{p+2}}{(r_{p+1}+2-r_{p+1}) \prod_{i=1}^{p+1}(r_i - r_{i-1} + 2)} \quad (56a)$$

$$= \frac{2^{p+2}}{2(r_{p+1}-r_p+2) \prod_{i=1}^{p}(r_i - r_{i-1} + 2)} \quad (56b)$$

$$= \frac{2^{p+1}}{(k-r_p) \prod_{i=1}^{p}(r_i - r_{i-1} + 2)}. \quad (56c)$$

Observe that this is exactly what we obtain when we have p restarts. Therefore, by the inductive hypothesis, we know that for $k = r_{p+1}+2$ we satisfy the upper bound at $x_{r_{p+1}+2}$. For $k \geq r_{p+1}+3$ we examine

$$\frac{2^{p+2}}{(k-r_{p+1}) \prod_{i=1}^{p+1}(r_i - r_{i-1} + 2))} - \frac{2^{p+1}}{(k-r_p) \prod_{i=1}^{p}(r_i - r_{i-1} + 2))}. \quad (57)$$

Proceeding similarly to the arguments in the base case (see (53) and (54)), we conclude that it is sufficient to examine the sign of $2(k-r_p) - (k-r_{p+1})(r_{p+1}-r_p+2)$. Write $k = r_{p+1} + a$ where $a \geq 3$, we now examine

$$2(k-r_p) - (k-r_{p+1})(r_{p+1}-r_p+2)$$

$$= 2(r_{p+1}+a-r_p) - (r_{p+1}+a-r_{p+1})(r_{p+1}-r_p+2) \quad (58a)$$

$$= 2(r_{p+1}+a-r_p) - (a)(r_{p+1}-r_p+2) \quad (58b)$$

$$= 2a + 2(r_{p+1}-r_p) - 2a - a(r_{p+1}-r_p) \quad (58c)$$

$$= (2-a)(r_{p+1}-r_p) \leq 0. \quad (58d)$$

Hence we conclude for $k \geq r_{p+1} + 2$

$$\frac{2^{p+2}}{(k-r_{p+1}) \prod_{i=1}^{p+1}(r_i - r_{i-1} + 2))} \leq \frac{2^{p+1}}{(k-r_p) \prod_{i=1}^{p}(r_i - r_{i-1} + 2))} \leq \frac{2}{k+1},$$

where the second inequality follows from the inductive hypothesis. Altogether we have shown that $(\forall p \geq 2)$, $(\forall k \in \{r_p + 2, \ldots, r_{p+1} + 1\})$ we have

$$f(x_k) - f(x^*) \leq \left(\frac{2^{p+1}}{(k - r_p)(r_p - r_{p-1} + 2) \ldots (r_2 - r_1 + 2)(r_1 + 2)} \right)^2 \frac{L}{2}(x_0 - x^*)^2 \tag{59a}$$

$$\leq \frac{2L}{(k+1)^2}(x_0 - x^*)^2. \tag{59b}$$

The proof is complete. ■

Appendix D. Numerical example: Simple Quadratic

In this and the upcoming appendices, we provide some numerical examples using the gradient-based restart strategy. The examples indicate that choosing x_{k+1} as the new initial point after the restart does not hamper the algorithm.

The first example is a simple quadratic $f(x) = \frac{1}{2}x^T Q x - q^T x$, where Q is an $n \times n$ positive definite matrix obtained by $Q = Q_0 + Q_0^T + 50\,\mathrm{Id}$ where Q_0 is generated using a $U[0, 1]$ distribution, and $q \in \mathbb{R}^n$ is randomly generated from a standard normal distribution. The Lipschitz constant L is given by the maximum eigenvalue of Q. For this experiment we set $n = 500$. We can see in Fig. 1 that keeping x_{k+1} at the restart yields a modest improvement over keeping x_k. Our theory for the $n = 1$ case required keeping x_{k+1} rather than x_k.

Appendix E. Huber Regression

Another numerical example is the following problem of Huber Regression. Define,

$$\psi_\tau(x) = \begin{cases} x^2, & |x| \leq \tau \\ 2\tau x - \tau^2 & x \geq \tau \\ -2\tau x - \tau^2 & x \leq -\tau. \end{cases} \tag{60}$$

Given $A \in \mathbb{R}^{m \times n}$ and $y \in \mathbb{R}^m$ we consider the optimization problem,

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} \sum_{i=1}^{m} \psi_\tau(a_i x - y_i) \tag{61}$$

where a_i is a row of A. For this experiment we set $\tau = 0.5$, $m = 300, n = 50$, and A and y were randomly generated using a standard normal distribution. In Fig. 2 it can be observed that restarting using x_{k+1} offers slightly better performance.

Appendix F. The Hinder-Lubin Example Expanded

Hinder and Lubin constructed a function [8, Appendix D.4] for which the restarts of O'Donoghue and Candés performed poorly until the iterates of $(x_k)_{k \in \mathbb{N}}$ were close to the minimizer. Their objective function is defined as

$$f(x) = \sum_{i=1}^{n} i h_\delta(x_i) + \frac{\alpha}{2}||x||_2^2, \tag{62}$$

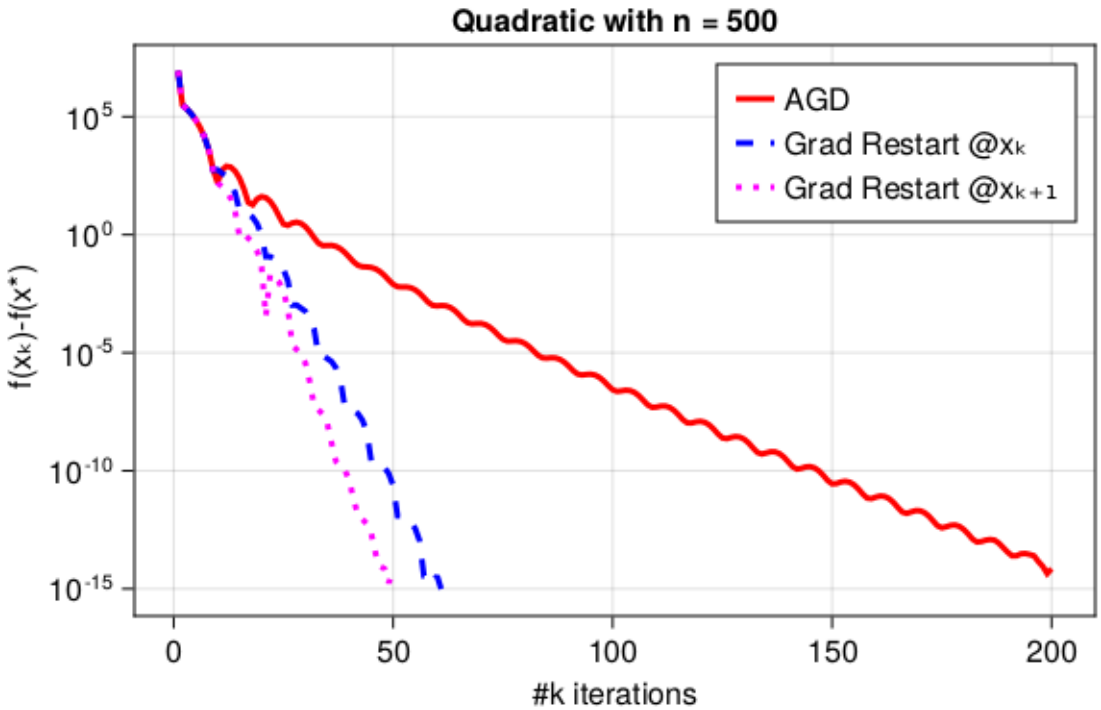

Figure 1: A `Julia` plot. A comparison of the performances of the restart scheme in Algorithm 1 at x_{k+1} (the pink dotted curve), the restart scheme in [11] at x_k (the blue dashed curve) and the classical AGD (the solid red curve) when applied to minimize simple quadratic.

where

$$h_\delta(z) = \begin{cases} z^2/2, & z \geq -\delta \\ -\delta z - \delta^2/2 & z < -\delta. \end{cases} \tag{63}$$

Remark that $f : \mathbb{R}^n \to \mathbb{R}$ is $(n+\alpha)-$smooth and α-strongly convex with a unique minimizer at $x = 0$. The function f is also separable, and therefore one can apply Algorithm 1 along each coordinate. We see that the restart scheme of Algorithm 1 performs well. In most cases, we do not have knowledge of whether the function is separable or not but there are cases where running Algorithm 1 in parallel along each coordinate has merit. We introduce a modified function obtained from (62) that is not separable. Given a matrix $A \in \mathbb{R}^{m\times n}$, define

$$F(x) = f(x) + \gamma \sum_{i=1}^{m} \left(\langle a_i, x\rangle + \sqrt{\langle a_i, x\rangle^2 + 1} \right) \tag{64}$$

where f is the Hinder and Lubin function defined in (62), a_i are the rows of A, and $\gamma \in \mathbb{R}$. For the experiment we chose $m = 110, n = 100$, and set $\delta = \alpha = \gamma = 10^{-4}$. Note that F is now $\left(n + \alpha + \gamma \sum_{i=1}^{m} ||a_i||^2\right)$ $-$smooth. In the experiment we compare AGD, AGD with the gradient

Figure 2: A `Julia` plot. A comparison of the performances of the restart scheme in Algorithm 1 at x_{k+1} (the pink dotted curve), the restart scheme in [11] at x_k (the blue dashed curve) and the classical AGD (the solid red curve) when applied to solve the Huber Regression problem.

restarts, and Algorithm 1 running along each coordinate. In Fig. 3 it can be seen that the gradient based restarts perform poorly until the iterates get close to the minimizer. On the other hand, restarting along each coordinate performs much better even though F is not separable.

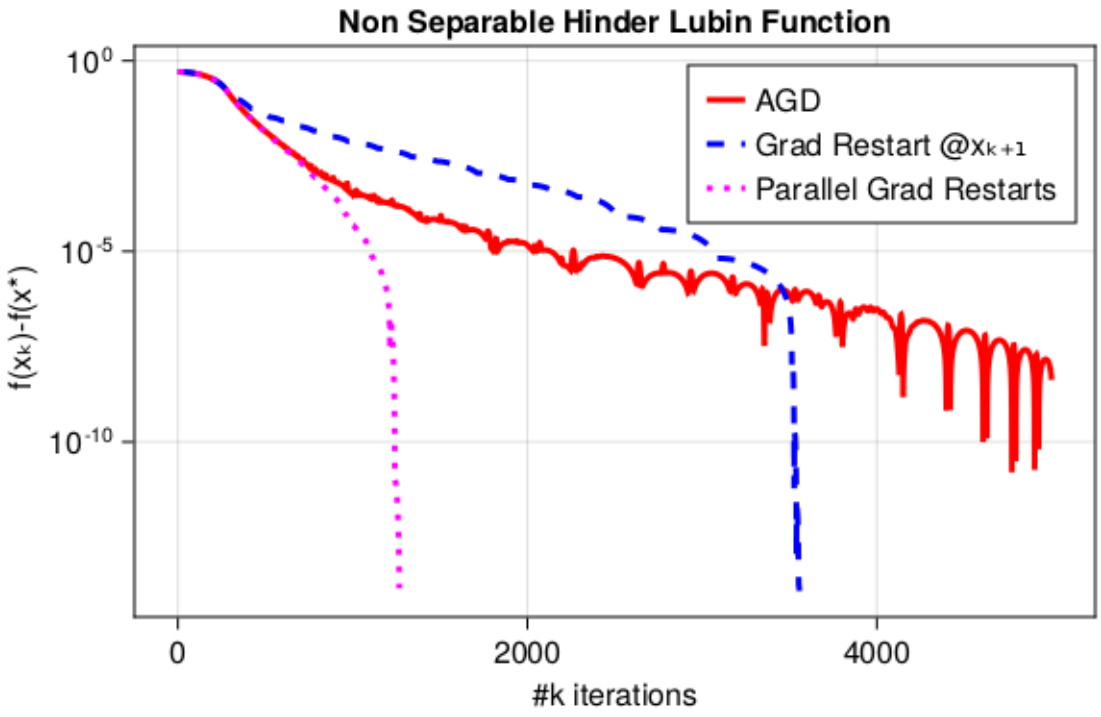

Figure 3: A `Julia` plot. A comparison of the performances of the parallel gradient restart scheme using Algorithm 1 for each coordinate (the pink dotted curve), the restart scheme in [11] at x_{k+1} (the blue dashed curve) and the classical AGD (the red curve) when applied to solve the modified Hinder-Lubin function (64). As the plots reflect, while the objective function is not separable, separate restarts still improve performance.