GRADIENT BASED ADAPTIVE RESTART IS LINEARLY CONVERGENT*

CAN KIZILKALE†, SHIVKUMAR CHANDRASEKARAN‡, MUSTAFA Ç. PINAR§, AND MING GU‡

Abstract.
Moment based gradient methods such as Polyak's Heavy-ball method, Nesterov's Accelerated gradient method are known to have non-monotonic periodic convergence behavior in the high momentum regime. If important function parameters like the condition number are known, the momentum can be adjusted to get linear convergence. Unfortunately these parameters are usually not accessible, and adaptive restarting is a technique used to tackle this problem. One of the most intuitive and well known heuristics is to look at the inner product of the momentum and gradient vector, and restart when this inner product is positive. In this paper we first prove that the convergence rate of this adaptive restarting heuristic is linear for functions fulfilling criteria which are also satisfied for strongly convex functions. Hence, for our analysis strong convexity is not a necessary condition and, therefore can be relaxed. Then we introduce a new restarting criterion that we call "cone based restart", and prove linear convergence under identical conditions. We also apply the restart heuristic to (approximations of) non-smooth convex functions.

Key words. Heavy-ball method, Accelerated gradient, restart, convex optimization, strong convexity.

AMS subject classifications. 68Q25, 68R10, 68U05

1. Introduction.

Speeding up gradient based algorithms via momentum is a well-known technique. Polyak's heavy-ball method [9] and Nesterov's accelerated gradient algorithm [6] are probably the two most famous algorithms using momentum.

Nesterov's method is well-known for achieving fast convergence despite not being more complex than the classical gradient descent algorithm. Although the algorithm was introduced more than three decades ago, it became very popular in the last decade due to its benefits in solving large scale problems in sparse signal recovery, machine learning, composite function optimization, etc., where higher order methods become infeasible.

The idea behind accelerated gradient scheme is building up momentum to increase the convergence rate. At each step instead of just taking into account the gradient we also take into account the momentum vector which is essentially a weighted sum of all the previous steps. The momentum vector contains some second order information about the objective function which leads to accelerated convergence when used correctly.

An important problem with the accelerated gradient algorithm (and momentum based methods in general) is that it exhibits non-monotonic convergence behavior. This behavior seems to be periodic and lowers the convergence rate. An intuitive explanation of this behavior is that, as the momentum increases, the algorithm takes much larger steps towards the optimum point, leading to faster decrease in the function

*Submitted to the editors DATE.

†Department of Computer Science, University of California Santa Barbara, Santa Barbara, CA (kizilkalecan@gmail.com)

‡ECE Department, University of California Santa Barbara, Santa Barbara, CA (shivchandrasekaran@ucsb.edu)

§Department of Industrial Engineering, Bilkent, Ankara,Turkey (mustafap@bilkent.edu.tr). For Ming Gu: Department of Mathematics, University of California Berkeley, Berkeley, CA (mgu@berkeley.edu)

value, until the point where the momentum vector makes the iterates move away from the optimum(overshoots) causing the function value to increase until the gradient of the objective function nullifies and corrects the direction.

One noteworthy observation is that when step sizes are chosen small enough the algorithm exhibits monotonic convergence until the first point of overshoot. The original algorithm lets the gradient slow down the algorithm once it overshoots. Yet we can obviously do better if we slow it down or stop it "artificially" when overshoot happens. Instead of slowing the algorithm using the gradient we restart it, which erases the history and starts the algorithm afresh using the current iterate as the initial point. If we know the condition number then one can exploit the periodic behavior of the non-monotonicity and employ periodic restarts at those points to achieve linear convergence [10]. When we do not have that information though, it is difficult to decide on the right periodicity.

Some of the tests for detecting overshoot are the exact non-monotonicity test [3], and the gradient-mapping test [8], both of which seem to work well in practice.

We shall first focus on the gradient-mapping test based restart and prove that it exhibits linear convergence under conditions which are more relaxed than strong convexity. To the best of our knowledge our paper is among the the first to establish a convergence rate result for gradient-based method with restarts. Prior analysis was restricted to quadratic functions only. Some other recent analyses of accelerated gradient methods have been based on ODEs [10, 11]. The main idea is to analyze the continuous case, where step size is arbitrarily small, and then expand the analysis by quantizing the continuous path. However, we use here the classical approach in proving the convergence rate. We show that with the gradient based restart condition the algorithm becomes monotonic. The momentum vector in the worst case grows like $O(\sqrt{k})$ (c.f. Corollary 2), and even in this case we have shown linear convergence rate (for additional experimental results on the effectiveness of this restart rule the reader can also refer to [3, 8]).

Our results are obtained under conditions more general than strong convexity. To prove our results we focus on the heavy ball method with constant inertial parameter. In the "restart strategy" considered in the literature, the inertial coefficients are the ones used for general convex functions and are restarted whereas our method is a classical gradient step without the inertial term. Hence, our usage of the term "restart" is different from its occurrences in the literature. We introduce three conditions which are sufficient for linear convergence. There has been quite an extensive research on achieving linear convergence without strong convexity (such as [1, 2, 4, 12]). The main ingredient used to achieve linear convergence is quadratic growth condition which is a quadratic lower bound on the objective $f(x)$ with respect to the shortest distance from x to the optimal set. Our first condition originates from KL condition, and in this sense it is similar to quadratic growth condition. Our second condition assumes a bound on the function where we check what happens in between subsequent iterates. We then show that these conditions can be satisfied without the need of strong convexity.

Finally we will introduce a new restarting condition, *cone based restart* which also achieves linear convergence under our conditions. This scheme gives similar performance to adaptive gradient scheme yet it uses the gradient in the beginning of each restart as pivot which shows that the information on the direction of acceleration is available to us in the beginning of each restart.

As regards recent work related to the present, we note that by analyzing accelerated proximal gradient methods under a local quadratic growth condition, Fercoq

and Qu [2] showed that restarting these algorithms at any frequency gives a globally linearly convergent algorithm. This result was previously known only for long enough frequencies. Then as the rate of convergence depends on the match between the frequency and the quadratic error bound, they design a scheme to automatically adapt the frequency of restart from the observed decrease of the norm of the gradient mapping. Another recent work related to the present is by Iutzeler and Malick [5] where the authors investigate the attractive properties of the proximal gradient algorithm with inertia. They show that using alternated inertia yields monotonically decreasing functional values, which contrasts with usual accelerated proximal gradient methods, and provide convergence rates for the algorithm with alternated inertia based on local geometric properties of the objective function.

2. Preliminaries. We are interested in solving the general unconstrained convex optimization problem,

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

where $f : R^n \to R$ is a convex function with L-Lipschitz continuous gradient where gradient being L-Lipschitz continuous is defined as follows (for the rest of the paper we will be denoting euclidean norm with $\|.\|$ unless otherwise is stated).

DEFINITION 2.1. *The gradient of f is L-Lipschitz continuous if there exists a constant $L > 0$ such that $\forall\ x, y \in dom(f)$*

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

Strong convexity is defined next. There are several equivalent definitions of strong convexity. We will use the following one.

DEFINITION 2.2. *A function $f : R^n \to R$ is strongly convex if $\forall\ x, y \in dom(f)$*

$$f(y) \geq f(x) + \nabla f(x)^T(y - x) + (\mu/2)\|y - x\|^2, \tag{2}$$

for some constant $\mu > 0$.

Nesterov's accelerated gradient algorithm is an instance of the general momentum based algorithms. This algorithm produces a sequence of iterates $x_k, y_k \in R^n$ by the following update rule.

DEFINITION 2.3. *Generalized accelerated gradient update rule:*

$$y_k = x_k + \beta_k(x_k - x_{k-1}) \tag{3}$$

$$x_{k+1} = y_k - \alpha_k \nabla f(y_k), \tag{4}$$

where the term $\beta_k(x_k - x_{k-1})$ is the momentum *term at each step.*

It is well-known that accelerated gradient (we will refer to Nesterov's algorithm as "accelerated gradient" for the rest of the paper) has a guaranteed convergence rate of $O(k^{-2})$. However, for strongly convex functions, if the condition number μ and Lipschitz constant L are known, it can be improved to linear convergence, $O(c^{-k})$ (c is some constant that is greater than 1) [7]. Unfortunately both are unknown in many problems. Moreover, it is frequently impractical to estimate μ.

To make the analysis shorter, we are going to investigate Polyak's Heavy-ball method which has the following update rule.

DEFINITION 2.4. *Heavy-ball update rule:*

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

From now on we will refer to the difference between two back to back iterates, $x_{k+1} - x_k$, as the **momentum** at step $k+1$.

The gradient-mapping restart test was proposed in [8]. The restart is initiated if an ascent direction has a positive projection on the gradient.

DEFINITION 2.5. *Gradient-mapping restart condition:*

$$\nabla f(x_k)^T (x_k - x_{k-1}) > 0.$$

3. General restarted momentum based gradient descent and its convergence rate. We start by analyzing the general version of a restarted moment based gradient descent algorithm (we will call it GRMD) given in Algorithm (1). This algorithm uses heavy-ball update rule instead of Nesterov's accelerated gradient method. After the analysis below we shall explain how the analysis can be extended to Nesterov's case.

Algorithm 1 Generalized Restarted Momentum Method

```
Choose x_{-1} ∈ R^n
x_0 = x_{-1}
for k ≥ 0 do
  z_{k+1} = β_k(x_k − x_{k−1}) − α_k ∇f(x_k)
  if restart condition is satisfied then
    x_{k+1} = x_k − α_k ∇f(x_k)
  else
    x_{k+1} = x_k + z_{k+1}
  end if
end for
```

Let x^* represent a minimizer. Assume that the objective function $f(x)$ is convex, smooth, and moreover the function and the algorithm satisfy following criteria:

$$\|\nabla f(x)\|^2 \geq (f(x) - f(x^*))/M, \tag{5}$$

for some $M > 0$,

$$f(x_k) - f(x_{k+1}) \geq m\|x_k - x_{k+1}\|^2, \tag{6}$$

for some $m > 0$ where k is the number of steps taken since the latest restart, and

$$\text{Restart is initiated if } \nabla f(x_k + z_{k+1})^T z_{k+1} > 0. \tag{7}$$

We should mention that although condition (6) depends on the algorithm iterates rather than arbitrary x, y as we will see in the examples later that it is possible to check this condition and see that it is weaker than strong convexity. We are going to show that under conditions (5),(6),(7) Algorithm (1) has linear convergence rate.

Observe that when there is no restart (from (7))

$$(x_k - x_{k-1})^T \nabla f(x_k) \leq 0,$$

then

$$\|\beta_k(x_k - x_{k-1}) - \alpha_k \nabla f(x_k)\|^2 \geq \|\beta_k(x_k - x_{k-1})\|^2 + \alpha_k^2 \|\nabla f(x_k)\|^2, \tag{8}$$

where the left hand side is the momentum at the *next* step $k+1$: $\|x_{k+1}-x_k\|^2$, if there is no restart in that step either.

Now let k_s denote the **first** iteration where we restart:

$$(x_{k_s}-x_{k_s-1})^T\nabla f(x_{k_s})\le 0. \tag{9}$$

Assume that

$$c(f(x_0)-f(x^*))=f(x_{k_s})-f(x^*).$$

To show linear convergence it is sufficient to establish that c has an upper bound strictly smaller than 1.

In the rest of the analysis, for the sake of simplicity, we will fix $\alpha_k=\alpha$ and $\beta_k=\beta$.

LEMMA 3.1. *For fixed α and β and $k\le k_s$,*

$$\|x_k-x_{k-1}\|\ge\alpha\sqrt{\frac{1}{M}\left(f(x_{k_s})-f(x^*)\right)\sum_{i=0}^{k-1}\beta^{2i}}.$$

Proof. When there is no restart we have

$$\gamma_k\equiv\|x_k-x_{k-1}\|=\|\beta_{k-1}(x_{k-1}-x_{k-2})-\alpha_{k-1}\nabla f(x_{k-1})\|.$$

From (5) we know that at each step $k\le k_s$,

$$\|\nabla f(x_k)\|\ge\sqrt{(f(x_{k_s})-f(x^*))/M}.$$

Combining this with (8), we get

$$\gamma_k^2\ge\beta^2\gamma_{k-1}^2+\frac{1}{M}\alpha^2(f(x_{k_s})-f(x^*)),$$

which yields the desired bound when combined with the fact that

$$\gamma_1=\alpha\|\nabla f(x_0)\|. \tag{10}$$

∎

COROLLARY 3.2. *The momentum grows like $O(\sqrt{k})$.*

Proof. We can pick β as close to 1 as possible and for such β, from Lemma 3.1 the result follows. ∎

LEMMA 3.3. *Let k_s be the first restarting step. Then,*

$$f(x_0)-f(x^*)\ge\left(f(x_{k_s})-f(x^*)\right)\left(1+\frac{m}{M}\alpha^2\sum_{k=0}^{k_s-1}\sum_{i=0}^{k}\beta^{2i}\right).$$

Proof. From (6), for $k<k_s$, and the fact that

$$f(x_k)-f(x^*)>f(x_{k_s})-f(x^*),$$

we have,

$$f(x_k)-f(x_{k+1})\ge\frac{m}{M}\alpha^2(f(x_{k_s})-f(x^*))\sum_{i=0}^{k}\beta^{2i}.$$

Therefore

$$f(x_0)-f(x_{k_s})\ge\frac{m}{M}\alpha^2(f(x_{k_s})-f(x^*))\sum_{k=0}^{k_s-1}\sum_{i=0}^{k}\beta^{2i},$$

which yields the desired bound. ∎

LEMMA 3.4. *If* $0 < \beta < 1$

$$k_s \leq \left[\frac{1}{2 \ln \beta} \ln \left(1 - \frac{1-\beta^2}{\frac{m}{M}\alpha^2} \right) - 1 \right]_+ .$$

Proof. From inequalities (5), (8), and Lemma 3.1, for fixed α and β we have:

$$\begin{aligned} \|x_{k+1} - x_k\|^2 &= \|\beta(x_k - x_{k-1}) - \alpha \nabla f(x_k)\|_2^2 \\ &\geq \|\beta(x_k - x_{k-1})\|^2 + \frac{1}{M}\alpha^2 (f(x_k) - f(x^*)) \\ &\geq \beta^{2k} \|x_1 - x_0\|^2 \\ &\geq \frac{1}{M}\alpha^2 \beta^{2k} \left(f(x_0) - f(x^*) \right). \end{aligned} \tag{11}$$

Substituting (6) in (11) and summing over k we get,

$$f(x_0) - f(x^*) \geq f(x_0) - f(x_{k_s}) \geq \frac{m}{M}\alpha^2 \sum_{k=0}^{k_s} \beta^{2k} (f(x_0) - f(x^*)).$$

Hence,

$$1 \geq \frac{m}{M}\alpha^2 \frac{1 - \beta^{2(k_s+1)}}{1 - \beta^2},$$

which yields, when $0 < \beta < 1$,

$$k_s \leq \left[\frac{1}{2 \ln \beta} \ln \left(1 - \frac{1-\beta^2}{\frac{m}{M}\alpha^2} \right) - 1 \right]_+ .$$

∎

We should note that this upper bound on k_s is probably not sharp but it will suffice for our purposes. Although the upper bound on k_s seems to be dependent on m/M (as we will see for the adaptive restart this expression is a function of μ) the selection of α can be completely automatized independent of m/M (one obvious way is to update α as $\frac{\alpha}{2}$ when $k_s < 2$).

LEMMA 3.5. *If* $\alpha < 1/L$ *then* $k_s \geq 2$. *Also, for any* $t \geq 2$, *there exists an* $\alpha > 0$, *such that* $k_s \geq t$.

Proof. Since ∇f is Lipschitz continuous

$$\|\nabla f(x) - \nabla f\left(x - \alpha \nabla f(x)\right)\| \leq L\alpha \|\nabla f(x)\|.$$

If $0 < \alpha < L^{-1}$ then

$$\nabla f(x)^T \nabla f(x - \alpha \nabla f(x)) \geq \nabla f(x)^T \left(\nabla f(x) - L\alpha \nabla f(x) \right) \geq 0.$$

Therefore $k_s \geq 2$ since the initial momentum is zero.

A similar, but more tedious, argument, shows that for all $t \geq 2$ there exists a small enough $\alpha > 0$ such that $k_s \geq t$. The basic idea is that for sufficiently small α the initial momentum can be kept as small as desired. Then the Lipschitz continuity is used as above to show that the restart condition will not be satisfied. ∎

Let k_j denote the number of iterations between the jth and $(j-1)$th restarts. Based on Lemmas 3.4 and 3.5, once $0 < \alpha < L^{-1}$ is fixed, we can choose $0 < \beta < 1$, such that there exist constants p and q which guarantee that

$$2 \leq p \leq k_j \leq q < \infty.$$

LEMMA 3.6. *Let r be the total number of iterations. Then*

$$f(x_r) - f(x^*) \leq (f(x_0) - f(x^*)) \left[\frac{1}{1 + \alpha^2 \frac{m}{M} \sum_{k=0}^{p-1} \sum_{i=0}^{k} \beta^{2i}} \right]^{\frac{r}{q}}.$$

Proof. Let $\hat{x}_j$ denote the point right at the beginning of the jth restart where $\hat{x}_0 = x_0$. From Lemma (3.3) and $k_j \geq p$, right at the beginning of the jth restart we have,

$$f(\hat{x}_{j-1}) - f(x^*) \geq (f(\hat{x}_j) - f(x^*)) \left(1 + \frac{m}{M} \alpha^2 \sum_{k=0}^{p-1} \sum_{i=0}^{k} \beta^{2i} \right).$$

If there are a total of N restarts until iteration r this inequality leads to,

$$(f(x_r) - f(x^*)) \leq (f(\hat{x}_0) - f(x^*)) \left[\frac{1}{1 + \alpha^2 \frac{m}{M} \sum_{k=0}^{p-1} \sum_{i=0}^{k} \beta^{2i}} \right]^{N}.$$

From $k_j \leq q$ we have $N \geq \frac{r}{q}$ combining with $\hat{x}_0 = x_0$ the result follows. ∎

Now we have all the ingredients we need to state the main result of this paper.

THEOREM 3.7. *Convergence rate of Algorithm (1) is linear.*

Proof. The lower and upper bounds on p and q from Lemmas 3.5 and 3.4 combined with the result in Lemma 3.6 yields

$$(f(x_k) - f(x^*)) \leq (f(x_0) - f(x^*)) \left[\frac{1}{1 + \alpha^2 \frac{m}{M} (\beta^2 + 1)} \right]^{\frac{k}{\left[\frac{1}{2 \ln \beta} \ln\left(1 - \frac{1-\beta^2}{\frac{m}{M}\alpha^2}\right) - 1 \right]_+}}$$

Let

$$0 < \tau = \left[\frac{1}{1 + \alpha^2 \frac{m}{M} (\beta^2 + 1)} \right]^{\frac{1}{\left[\frac{1}{2 \ln \beta} \ln\left(1 - \frac{1-\beta^2}{\frac{m}{M}\alpha^2}\right) - 1 \right]_+}} < 1.$$

Then we see that Algorithm 1 converges like $O(\tau^k)$ which is linear as claimed. ∎

Algorithm 2 Momentum accelerated gradient algorithm with gradient-mapping restart

Choose $x_{-1} \in R^n$
$x_0 = x_{-1}$
for $k \geq 0$ **do**
  $z_{k+1} = \beta_k (x_k - x_{k-1}) - \alpha_k \nabla f(x_k)$
  if $\nabla f(x_k + z_{k+1})^T z_{k+1} > 0$ **then**
    $x_{k+1} = x_k - \alpha_k \nabla f(x_k)$
  else
    $x_{k+1} = x_k + z_{k+1}$
  end if
end for

3.1. Convergence Rate of Adaptive Restart under Strong Convexity. Now, we show that under strong convexity the adaptive restart rule achieves linear convergence (this version of the algorithm is given in Algorithm (2)). We shall henceforth refer to Algorithm 2 alternatively as MAGR. It is enough to establish that the criteria (5), (6) and (7) are satisfied. Criterion (7) is satisfied by definition of the adaptive restart rule.

Assuming that no restart was initiated,

$$(x_{k+1} - x_k)^T \nabla f(x_{k+1}) \leq 0.$$

Then from equation (2) we have that,

$$f(x_k) \geq f(x_{k+1}) + \nabla f(x_{k+1})^T (x_k - x_{k+1}) + (\mu/2)\|x_{k+1} - x_k\|^2,$$

which implies that,

$$f(x_k) - f(x_{k+1}) \geq (\mu/2)\|x_{k+1} - x_k\|^2. \tag{12}$$

So criterion (6) is satisfied for $m = \frac{\mu}{2}$. Strong convexity can be also used to bound the gradients at each step.

$$f(x^*) - f(x) - \nabla f(x)^T (x^* - x) \geq (\mu/2)\|x - x^*\|^2,$$

where x^* denotes the minimum of f, leading to,

$$\begin{aligned} f(x) - f(x^*) &\leq \nabla f(x)^T (x - x^*) - (\mu/2)\|x - x^*\|^2 \\ &\leq \|\nabla f(x)\|\|x - x^*\| - (\mu/2)\|x - x^*\|^2 \\ &= \frac{\|\nabla f(x)\|^2}{2\mu} - \left(\|x - x^*\|\sqrt{\frac{\mu}{2}} - \frac{\|\nabla f(x)\|}{\sqrt{2\mu}}\right)^2 \\ &\leq \frac{\|\nabla f(x)\|^2}{2\mu}. \end{aligned} \tag{13}$$

hence criterion (5) is satisfied for $M = \frac{1}{2\mu}$. As we can see all three criteria are satisfied for this case hence we conclude that "Adaptive Restart" scheme has linear convergence.

3.2. Extension to Accelerated Gradient update rule. At this point we sketch how this analysis can be extended for the generalized accelerated update rule in Definition (2.3). In accelerated gradient method the intermediate gradient $\nabla f(y_k)$ is accumulated instead of $\nabla f(x_k)$ hence instead of $(x_k - x_{k-1})^T \nabla f(x_k)$ algorithm checks $(y_k - x_{k-1})^T \nabla f(y_k)$ and restarts if this inner product is positive. In this case instead of (8) we will get $\|\beta_k(x_k - x_{k-1}) - \alpha_k \nabla f(y_k)\|_2^2 \geq \|\beta_k(x_k - x_{k-1})\|^2 + \alpha_k^2 \|\nabla f(y_k)\|^2$ and the rest of the analysis will follow very similarly. Since the gradient is smooth it is reasonable to expect that $(x_{k+1} - x_k)^T \nabla f(x_{k+1})$ and $(y_k - x_{k-1})^T \nabla f(y_k)$ to have the same sign most of the time (it also gives almost identical results in the experiments) however using $(y_k - x_{k-1})^T \nabla f(y_k)$ as the restarting criteria prevents any complications. This extension shows that even if we pick a constant β instead of the one in the Accelerated gradient with adaptive restart, it will have linear convergence. When we pick β_k as in Accelerated gradient, the convergence rate will even be faster in between restarts. For further clarification, observe that since $\beta_k \to 1$ monotonically and since no matter how close to 1 β is, the analysis continues to hold (when we pick β greater than β_2 in accelerated gradient, we guarantee that the summation $\sum_{k=0}^{2} \beta_k^{2k}$ stays small enough so that the lower bound for k_s still holds; notice also that since β_k will eventually exceed any constant β, k_s is still bounded from above).

3.3. A Non-Strongly Convex example. A simple example of a non-strongly convex function that satisfies the two conditions is $f(x) = x^T Ax/2$ where A is a symmetric positive semi-definite matrix with at least one zero eigenvalue. Since A is not full rank it is obvious that this objective function is not strongly convex.

However at every x that is not a minimum we have

$$\nabla f(x) = Ax = \sum_i c_i v_i \neq 0,$$

where v_i are eigenvectors corresponding to eigenvalues $\lambda_i > 0$ (Since A is positive semi definite, the eigenvectors corresponding to 0 eigenvalue span the null-space of A hence $\nabla f(x)$ is a linear combination of eigenvectors corresponding to non-zero eigenvalues). Then for $\hat{\lambda} = \min_{\lambda_i > 0} \lambda_i$ we have

$$f(x) - f(x^*) = \frac{x^T Ax}{2} \leq \frac{x^T A^2 x}{2\hat{\lambda}} = \frac{\|\nabla f(x)\|^2}{2\hat{\lambda}}.$$

So, criterion (5) is satisfied. Since $x_k - x_{k+1}$ is a linear combination of the gradients until step k then it is a linear combination of eigenvectors corresponding to non-zero eigenvalues. We get

$$(x_k - x_{k+1})^T A(x_k - x_{k+1}) \geq \hat{\lambda}\|x_k - x_{k+1}\|^2.,$$

and

$$x_k^T Ax_k - x_{k+1}^T Ax_{k+1} = (x_k - x_{k+1})^T A(x_k - x_{k+1}) + 2(x_k - x_{k+1})^T Ax_{k+1}.$$

From the restart condition $\nabla f(x_{k+1})^T (x_k - x_{k+1}) \geq 0$, we conclude that,

$$x_k^T Ax_k - x_{k+1}^T Ax_{k+1} \geq (x_k - x_{k+1})^T A(x_k - x_{k+1}) \geq \hat{\lambda}\|x_k - x_{k+1}\|^2,$$

which satisfies criterion (6). Therefore, this example is not strongly convex yet it satisfies both criteria and adaptive restart scheme will achieve linear convergence. While this example is admittedly simple the goal is just to show that the conditions we have introduced are weaker than strong convexity.

4. Cone based restart. We now introduce a new gradient based restart criteria which we call "cone based restart". As we will see in the experiments the corresponding Algorithm (3) has very similar convergence behavior and speed to Algorithm (2), but it has some nice properties that make it easier to analyze. Moreover the coefficient c in Algorithm (3) makes it possible to tune the algorithm.

The restart condition

$$\nabla f(x_k + z_{k+1})^T g_r < c\,\|\nabla f(x_k + z_{k+1})\|\|g_r\|,$$

guarantees that all of the gradients until the next restart lie in the cone centered around a pivot vector (gradient right after the restart) g_r. Here the parameter c is used to adjust the width of the cone, $c = 0$ will make it a half-space and maximize the growth of momentum but will cause overshooting while $c = 1$ will make this cone just a ray hence the algorithm will practically behave like the classical gradient descent algorithm. For strongly convex objective functions, when we pick $1 > c > \frac{1}{\sqrt{2}}$, for all k we will have $(x_k - x_{k+1})^T \nabla f(x_{k+1}) > 0$ when there is no restart and this leads to linear convergence.

Algorithm 3 Momentum accelerated gradient algorithm with cone based restart

Choose $x_{-1} \in R^n$
Choose $c > 1/\sqrt{2}$
$x_0 = x_{-1}$
$g_r = \nabla f(x_0)$
for $k \geq 0$ **do**
  $z_{k+1} = \beta_k(x_k - x_{k-1}) - \alpha_k \nabla f(x_k)$
  if $\nabla f(x_k + z_{k+1})^T g_r < c\|\nabla f(x_k + z_{k+1})\|\|g_r\|$ **then**
    $x_{k+1} = x_k - \alpha_k \nabla f(x_k)$
    $g_r = \nabla f(x_{k+1})$
  else
    $x_{k+1} = x_k + z_{k+1}$
  end if
end for

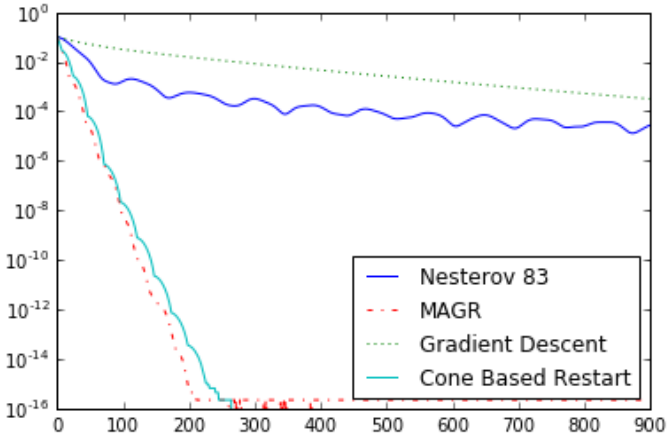

FIG. 1. *Optimizing the smooth version for $\rho = 1$. The vertical axis depicts $\frac{(f(x_n)-f^*)}{f^*}$, and the horizontal axis depicts the iteration number n.*

4.1. Another Example. We will look into the following problem.

$$f(x) = \rho \log\left(\sum_{i=1}^{m} \exp((a_i^T x - b_i)/\rho)\right). \tag{14}$$

This problem is a smoothed version of the more general problem of

$$f(x) = \max_{i=1,\ldots,m} (a_i^T x - b_i). \tag{15}$$

In our numerical experiments we took $\alpha_k = 0.99$, $\beta_k = (r+1)/(r-1)$(for constant β 0.99 is selected) where r is the number of steps taken after the latest restart and c is taken slightly larger than $\frac{1}{\sqrt{2}}$.

The experiments show linear convergence of both Cone Based Restart and Algorithm (2), and how close their convergence behaviors are. In these experiments we did not add the simulations for the accelerated gradient version since it is guaranteed

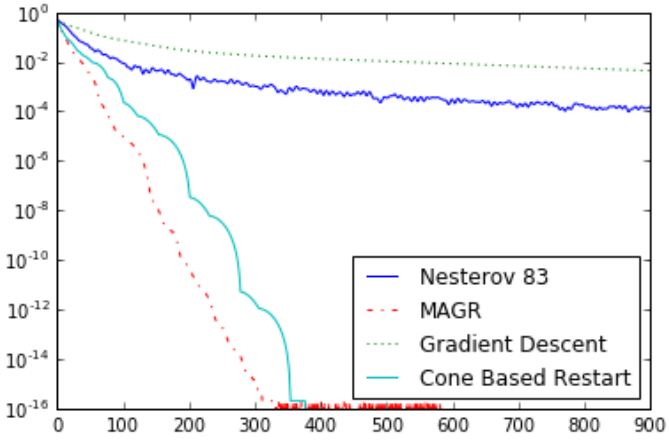

FIG. 2. *Optimizing the smooth version for* $\rho = 0.1$. *The vertical axis depicts* $\frac{(f(x_n)-f^*)}{f^*}$, *and the horizontal axis depicts the iteration number* n.

to be at least as fast. Moreover that comparison is already given in papers such as ([8]).

5. Conclusions. This paper has shown that the gradient-mapping based restart scheme will improve the convergence rate of momentum based algorithms to linear. Although this was suspected to be the case in practice we have now proved it to be true under the assumptions (5), (6) and (7), which are less restrictive than the assumption of strong convexity. The proposed cone based restarting condition has very similar convergence behavior, while using the initial gradient as pivot. This shows that the acceleration happens when the steps do not deviate too much along a reference vector and this vector be chosen in the beginning of each restart. Since it also has the flexibility of the tuning parameter c we believe it may serve well where MAGR cannot. Looking forward, it might be possible to give a more general analysis that covers an even larger class of restarting schemes. We would also like to study an extension to problems of non-smooth nature.

REFERENCES

[1] J. BOLTE, S. SABACH, AND M. TEBOULLE, *Proximal alternating linearized minimization for nonconvex and nonsmooth problems*, Mathematical Programming, 146 (2014), pp. 459–494.
[2] O. FERCOQ AND Z. QU, *Adaptive restart of accelerated gradient methods under local quadratic growth condition*, IMA Journal of Numerical Analysis, 39 (2019), pp. 2069–2095.
[3] P. GISELSSON AND S. BOYD, *Monotonicity and restart in fast gradient methods*, in Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on, IEEE, 2014, pp. 5058–5063.
[4] K. HOU, Z. ZHOU, A. M.-C. SO, AND Z.-Q. LUO, *On the linear convergence of the proximal gradient method for trace norm regularization*, in Advances in Neural Information Processing Systems, 2013, pp. 710–718.
[5] F. IUTZELER AND J. MALICK, *On the proximal gradient algorithm with alternated inertia*, Journal of Optimization Theory and Applications, 176 (2018), pp. 688–710.
[6] Y. NESTEROV, *A method of solving a convex programming problem with convergence rate o (1/k2)*, in Soviet Mathematics Doklady, vol. 27, 1983, pp. 372–376.
[7] Y. NESTEROV, *Smooth minimization of non-smooth functions*, Mathematical Programming, 103 (2005), pp. 127–152.
[8] B. O'DONOGHUE AND E. CANDES, *Adaptive restart for accelerated gradient schemes*, Founda-

tions of Computational Mathematics, 15 (2015), pp. 715–732.
[9] B. T. Polyak, *Some methods of speeding up the convergence of iteration methods.*, USSR Computational Mathematics and Mathematical Physics, 5 (1964), pp. 1–17.
[10] W. Su, S. Boyd, and E. J. Candes, *A differential equation for modeling nesterov's accelerated gradient method: theory and insights*, Journal of Machine Learning Research, 17 (2016), pp. 1–43.
[11] A. C. Wilson, B. Recht, and M. I. Jordan, *A lyapunov analysis of momentum methods in optimization*, arXiv preprint arXiv:1611.02635, (2016).
[12] Z. Zhou, Q. Zhang, and A. M.-C. So, *1,p-norm regularization: Error bounds and convergence rate analysis of first-order methods.*, in ICML, 2015, pp. 1501–1510.