A single potential governing convergence of conjugate gradient, accelerated gradient and geometric descent*

Sahar Karimi[†] Stephen Vavasis[‡]

December 18, 2017

Abstract

Nesterov's accelerated gradient (AG) method for minimizing a smooth strongly convex function f is known to reduce $f(\mathbf{x}_k) - f(\mathbf{x}^*)$ by a factor of $\epsilon \in (0, 1)$ after $k = O(\sqrt{L/\ell}\log(1/\epsilon))$ iterations, where ℓ, L are the two parameters of smooth strong convexity. Furthermore, it is known that this is the best possible complexity in the function-gradient oracle model of computation. Modulo a line search, the geometric descent (GD) method of Bubeck, Lee and Singh has the same bound for this class of functions. The method of linear conjugate gradients (CG) also satisfies the same complexity bound in the special case of strongly convex quadratic functions, but in this special case it can be faster than the AG and GD methods.

Despite similarities in the algorithms and their asymptotic convergence rates, the conventional analysis of the running time of CG is mostly disjoint from that of AG and GD. The analyses of the AG and GD methods are also rather distinct.

Our main result is analyses of the three methods that share several common threads: all three analyses show a relationship to a certain "idealized algorithm", all three establish the convergence rate through the use of the Bubeck-Lee-Singh geometric lemma, and all three have the same potential that is computable at run-time and exhibits decrease by a factor of $1 - \sqrt{\ell/L}$ or better per iteration.

One application of these analyses is that they open the possibility of hybrid or intermediate algorithms. One such algorithm is proposed herein and is shown to perform well in computational tests.

1 First-order methods for strongly convex functions

Three methods for minimizing smooth, strongly convex functions are considered in this work, conjugate gradient, accelerated gradient, and geometric descent. CG is the oldest

*Supported in part by a grant from the U. S. Air Force Office of Scientific Research and in part by a Discovery Grant from the Natural Sciences and Engineering Research Council (NSERC) of Canada.

[†]Department of Combinatorics & Optimization, University of Waterloo, 200 University Ave. W., Waterloo, ON, N2L 3G1, Canada, sahar.karimi@gmail.com.

[‡]Department of Combinatorics & Optimization, University of Waterloo, 200 University Ave. W., Waterloo, ON, N2L 3G1, Canada, vavasis@uwaterloo.ca.

and perhaps best known of the methods. It was introduced by Hestenes and Stiefel [6] for minimizing strongly convex quadratic functions of the form $f(\mathbf{x}) = \mathbf{x}^T A\mathbf{x}/2 - \mathbf{b}^T\mathbf{x}$, where A is a symmetric positive definite matrix. In what follows, we refer to this algorithm as either CG (conjugate gradient) or LCG (linear conjugate gradient); the latter usage is to distinguish the Hestenes-Stiefel method from nonlinear conjugate gradient algorithms that followed and that are discussed further below.

There is a significant body of work on gradient methods for more general smooth, strongly convex functions. We say that a differentiable convex function $f : \mathbb{R}^n \to \mathbb{R}$ is *smooth, strongly convex* [7] if there exist two scalars $L \geq \ell > 0$ such that for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$,

$$\ell\|\mathbf{x} - \mathbf{y}\|^2/2 \leq f(\mathbf{y}) - f(\mathbf{x}) - \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x}) \leq L\|\mathbf{x} - \mathbf{y}\|^2/2. \tag{1}$$

This is equivalent to assuming convexity and lower and upper Lipschitz constants on the gradient:

$$\ell\|\mathbf{x} - \mathbf{y}\| \leq \|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|.$$

Nemirovsky and Yudin [11] proposed a method for minimizing smooth strongly convex functions requiring $k = O(\sqrt{L/\ell}\log(1/\epsilon))$ iterations to produce an iterate $\mathbf{x}_k$ such that $f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \epsilon(f(\mathbf{x}_0) - f(\mathbf{x}^*))$, where $\mathbf{x}^*$ is the optimizer (necessarily unique under the assumptions made). A drawback of their method is that it requires a two-dimensional optimization on each iteration that can be cumbersome to implement (to the best of our knowledge, the algorithm was not ever widely adopted). Nesterov [12] proposed another method, nowadays known as the "accelerated gradient" (AG) method, which achieves the same optimal complexity that requires a single function and gradient evaluation on each iteration.

In the special case of strongly convex quadratic functions, the parameters ℓ and L appearing in (1) correspond to $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$, the extremal eigenvalues of A. The conjugate gradient method has already been known to satisfy the asymptotic iteration bound $k = O(\sqrt{L/\ell}\log(1/\epsilon))$ since the work of Daniel (1967) described below.

Although the two methods satisfy the same asymptotic bound, the analyses of the two methods are completely different. In the case of AG, there are two analyses by Nesterov [12, 13]. In our own previous work [9], we provided a third analysis based on another potential.

In the case of linear conjugate gradient, we are aware of no direct analysis of the algorithm prior to our own previous work [9]. By "direct," we mean an analysis of $f(\mathbf{x}_k) - f(\mathbf{x}^*)$ using the recurrence inherent in CG. Instead, the standard analysis introduced by Daniel, whose theorem is stated precisely below, proves that another iterative method, for example Chebyshev iteration [4] or the heavy-ball iteration [15, 1] achieves reduction of $\left(1 - O(\sqrt{\ell/L})\right)$ per iteration. Then one appeals to the optimality of the CG iterate in the Krylov space generated by all of these methods to claim that the CG iterate must be at least as good as the others.

Recently, Bubeck, Lee and Singh [2] proposed the geometric descent (GD) algorithm for analysis of a variant of accelerated gradient [2], called "geometric descent" (GD). As presented by the authors, the algorithm requires an exact line-search on each iteration, although it is possible that similar theoretical guarantees could be established for an approximate line search. Under the assumption that the line-search requires a constant number of function and gradient evaluations, then GD also requires $k = O(\sqrt{L/\ell}\log(1/\epsilon))$ iterations.

We propose analyses of these three algorithms that share several common features. First, all three algorithms can be analyzed using the geometric lemma of Bubeck, Lee and Singh, which is presented in Section 3. Second, all three are related to an "idealized" algorithm which is described and analyzed in Section 4. Finally, the convergence behavior for all three of them is governed by a potential $\tilde{\sigma}_k$, which has the following three properties:

1. There exists an auxiliary sequence of vectors $\mathbf{y}_0, \mathbf{y}_1, \ldots$ such that
$$\tilde{\sigma}_k^2 \geq \|\mathbf{y}_k - \mathbf{x}^*\|^2 + \frac{2(f(\mathbf{x}_k) - f(\mathbf{x}^*)}{\ell},$$
for $k = 0, 1, 2 \ldots,$
2. $\tilde{\sigma}_{k+1}^2 \leq \left(1 - \sqrt{\frac{\ell}{L}}\right)\tilde{\sigma}_k^2$, and
3. $\tilde{\sigma}_k$ is computable on each iteration (assuming prior knowledge of ℓ, L) in $O(n)$ operations.

These results are established for the GD algorithm in Section 5. Section 6 establishes the relationship between the idealized algorithm and linear CG, and Section 7 shows how to define $\tilde{\sigma}_k$ for linear CG to establish the above three properties. These three results are established for the AG algorithm in Section 8.

Because the three algorithms each compute a scalar $\tilde{\sigma}_k$ satisfying the above properties, it becomes straightforward to create hybrids. In other words, the above analysis treats all three algorithms as essentially 1-step processes as opposed to long inductive chains. In Section 9 we propose a hybrid CG algorithm that performs well in computational tests, which are described in Section 10. The reason from making a hybrid CG algorithm is that the performance of linear conjugate gradient on specific instances can be much better than the worst-case bound given by Daniel's theorem; the performance on specific instances is highly governed by the eigenvalues of A. Therefore, using a conjugate-gradient-like algorithm for a nonlinear problem may also perform better than the $(1 - \sqrt{\ell/L})^k$ worst-case convergence bound. This is also the motivation for traditional nonlinear conjugate gradient, as we discuss below.

We conclude this introductory section with a few remarks about our previous related manuscript [9]. In that work, we established that a potential defined by
$$\Psi_k = \|\mathbf{y}_k - \mathbf{x}^*\|^2 + \frac{2(f(\mathbf{x}_k) - f(\mathbf{x}^*)}{\ell}$$
decreases by a factor $(1 - \sqrt{\ell/L})$ per iteration for both LCG and AG. This Ψ_k is not computable since $\mathbf{x}^*$ is not known *a priori*, and therefore our previous result does not have any immediate practical application. The potential $\tilde{\sigma}_k$ developed herein is computable on every step and therefore may be used to guide a hybrid algorithm. In addition, the current work also applies to the GD method, which was not addressed in our previous manuscript.

2 Notation

Define $B(\mathbf{x}, r) = \{\mathbf{y} \in \mathbb{R}^n : \|\mathbf{x} - \mathbf{y}\| \le r\}$, i.e., the closed ball centered at $\mathbf{x} \in \mathbb{R}^n$ of radius r.

An *affine set* is a set of the form $\mathcal{M} = \{\mathbf{c} + \mathbf{w} : \mathbf{w} \in \mathcal{W}\}$ where $\mathbf{c} \in \mathbb{R}^n$ is fixed and $\mathcal{W} \subset \mathbb{R}^n$ is a linear subspace. We write this as $\mathcal{M} = \mathbf{c} + \mathcal{W}$, a special case of a Minkowski sum. Another notation for an affine set is $\mathrm{aff}\{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$, which is defined as $\{\alpha_1 \mathbf{x}_1 + \cdots + \alpha_k \mathbf{x}_k : \alpha_1 + \cdots + \alpha_k = 1\}$. If $\mathbf{w}_1, \ldots, \mathbf{w}_k$ span $\mathcal{W}$, then it is clear that $\mathbf{c} + \mathcal{W} = \mathrm{aff}\{\mathbf{c}, \mathbf{c} + \mathbf{w}_1, \ldots, \mathbf{c} + \mathbf{w}_k\}$.

Suppose $\mathcal{U}$ is an affine subset of $\mathbb{R}^n$. The set $\mathbf{T}\mathcal{U} = \{\mathbf{x} - \mathbf{y} : \mathbf{x}, \mathbf{y} \in \mathcal{U}\}$ is called the *tangent space* of $\mathcal{U}$ and is a linear subspace. If $\mathcal{U}$ is presented as $\mathcal{U} = \mathbf{c} + \mathcal{W}$, where $\mathcal{W}$ is a linear subspace, then it follows that $\mathbf{T}\mathcal{U} = \mathcal{W}$.

3 Preliminary lemmas

We start with a special case of a lemma from Drusvyatskiy et al. [3], which is an extension of work by Bubeck et al. [2]:

Lemma 1 *Suppose* $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$. *Let* δ, ρ, σ *be three nonnegative scalars such that* $\delta \le \|\mathbf{x} - \mathbf{y}\|$. *Suppose* $\lambda \in [0, 1]$ *and*

$$\mathbf{z} = (1 - \lambda)\mathbf{x} + \lambda \mathbf{y}. \tag{2}$$

Then

$$B(\mathbf{x}, \rho) \cap B(\mathbf{y}, \sigma) \subset B(\mathbf{z}, \xi),$$

where

$$\xi = \sqrt{(1 - \lambda)\rho^2 + \lambda \sigma^2 - \lambda(1 - \lambda)\delta^2}, \tag{3}$$

The argument of the square-root in (3) *is guaranteed to be nonnegative whenever* $B(\mathbf{x}, \rho) \cap B(\mathbf{y}, \sigma) \ne \emptyset$, *or equivalently, whenever* $\rho + \sigma \ge \|\mathbf{x} - \mathbf{y}\|$.

Proof. We prove the second claim first. The quantity appearing in the square root of (3) is nonnegative as the following inequalities show:

$$\begin{aligned}
(1 - \lambda)\rho^2 + \lambda\sigma^2 - (1 - \lambda)\lambda\delta^2 &= (1 - \lambda)\lambda(\rho^2 + \sigma^2 - \delta^2) + (1 - \lambda)^2\rho^2 + \lambda^2\sigma^2 \\
&\ge (1 - \lambda)\lambda(\rho^2 + \sigma^2 - \delta^2) + 2(1 - \lambda)\lambda\rho\sigma \\
&= (1 - \lambda)\lambda((\rho + \sigma)^2 - \delta^2) \\
&\ge 0,
\end{aligned}$$

where the last line uses the assumptions $\rho + \sigma \ge \|\mathbf{x} - \mathbf{y}\| \ge \delta$.

Now for the first part of the lemma, the proof of (3) follows from more general analysis in Drusvyatskiy et al. [3]. Assume that $\mathbf{p} \in B(\mathbf{x}, \rho) \cap B(\mathbf{y}, \sigma)$ so

$$(\mathbf{p} - \mathbf{x})^T(\mathbf{p} - \mathbf{x}) - \rho^2 \le 0, \tag{4}$$

$$(\mathbf{p} - \mathbf{y})^T(\mathbf{p} - \mathbf{y}) - \sigma^2 \le 0. \tag{5}$$

For $\lambda \in [0,1]$, add $(1-\lambda)$ times (4) to λ times (5) and rearrange to obtain a new inequality satisfied by $\mathbf{p}$:

$$(\mathbf{p}-\mathbf{z})^T(\mathbf{p}-\mathbf{z}) + (1-\lambda)\lambda\|\mathbf{x}-\mathbf{y}\|^2 - (1-\lambda)\rho^2 - \lambda\sigma^2 \le 0,$$

i.e.

$$\|\mathbf{p}-\mathbf{z}\| \le \left((1-\lambda)\rho^2 + \lambda\sigma^2 - (1-\lambda)\lambda\|\mathbf{x}-\mathbf{y}\|^2\right)^{1/2},$$

where $\mathbf{z}$ is defined by (2). By substituting the definition $\delta \le \|\mathbf{x}-\mathbf{y}\|$, we observe that $\mathbf{p} \in B(\mathbf{z}, \xi)$, where ξ is defined by (3). □

This leads to the following, which is a more precise statement of the geometric lemma from Bubeck et al. [2]:

Lemma 2 *Let $\mathbf{x}, \mathbf{y}, \rho, \sigma, \delta$ be as in the preceding lemma. Under the assumption $\rho + \sigma \ge \delta$ and the additional assumption $\delta \ge \sqrt{|\rho^2 - \sigma^2|}$,* (3) *is minimized over possible choices of $\lambda \in [0,1]$ by:*

$$\lambda^* = \frac{\delta^2 + \rho^2 - \sigma^2}{2\delta^2}, \tag{6}$$

yielding

$$\mathbf{z}^* = (1-\lambda^*)\mathbf{x} + \lambda^*\mathbf{y}, \tag{7}$$

in which case the minimum value of (3) *is,*

$$\xi^* = \frac{1}{2}\sqrt{2\rho^2 + 2\sigma^2 - \delta^2 - \frac{(\rho^2-\sigma^2)^2}{\delta^2}}. \tag{8}$$

Proof. First, note that $|\rho^2 - \sigma^2|/\delta^2 \le 1$ by the assumption made, thus ensuring that $\lambda^* \in [0,1]$. Therefore, it follows from the preceding lemma that the quantity appearing in the square root of (3) is nonnegative. The previous lemma establishes that for any $\mathbf{p} \in B(\mathbf{x}, \rho) \cap B(\mathbf{y}, \sigma)$, and for an arbitrary $\lambda \in [0,1]$,

$$\|\mathbf{p}-\mathbf{z}\|^2 \le (1-\lambda)\rho^2 + \lambda\sigma^2 - \lambda(1-\lambda)\delta^2.$$

We observe that the right-hand side is a convex quadratic in λ and hence is minimized when the derivative with respect to λ is zero, and one checks that this value is precisely (6). Substituting $\lambda = \lambda^*$ into (3) yields (8). □

4 Idealized algorithm

We consider the following idealized algorithm for minimizing $f(\mathbf{x})$, where $f : \mathbb{R}^n \to \mathbb{R}$ is smooth, strongly convex. As in the introduction, let ℓ, L denote the two parameters of strong convexity.

Idealized Algorithm (IA)

```
$\mathbf{x}_0 :=$ arbitrary
$\mathcal{M}_1 := \mathbf{x}_0 + \mathrm{span}\{\nabla f(\mathbf{x}_0)\}$
for $k := 1, 2, \ldots$
    $\mathbf{x}_k := \mathrm{argmin}\{f(\mathbf{x}) : \mathbf{x} \in \mathcal{M}_k\}$    (9)
    $\mathbf{y}_k := \mathrm{argmin}\{\|\mathbf{y} - \mathbf{x}^*\| : \mathbf{y} \in \mathcal{M}_k\}$    (10)
    $\mathcal{M}_{k+1} := \mathbf{x}_k + \mathrm{span}\{\mathbf{y}_k - \mathbf{x}_k, \nabla f(\mathbf{x}_k)\}$    (11)
end
```

This algorithm is called "idealized" because it is not implementable in the general case; it requires prior knowledge of $\mathbf{x}^*$ in (10). Nonetheless, we will argue that LCG, accelerated gradient, and geometric gradient are related to the idealized algorithm in different ways.

Notice that $\mathcal{M}_k$ is an affine set that is two-dimensional on most iterations. Alternate notation for this set, also used herein, is $\mathcal{M}_k = \mathrm{aff}\{\mathbf{x}_{k-1}, \mathbf{y}_{k-1}, \mathbf{x}_{k-1} - \nabla f(\mathbf{x}_{k-1})\}$. Note also that by (9) and (10), $\mathbf{x}_k, \mathbf{y}_k \in \mathcal{M}_k$, and by (11), $\mathbf{x}_k, \mathbf{y}_k \in \mathcal{M}_{k+1}$, and therefore $\mathcal{M}_k$, $\mathcal{M}_{k+1}$ have a common 1-dimensional affine subspace.

We start with the main theorem about IA. For iteration k, define a potential Ψ_k as follows:

$$\Psi_k = \|\mathbf{y}_k - \mathbf{x}^*\|^2 + \frac{2(f(\mathbf{x}_k) - f(\mathbf{x}^*))}{\ell}. \tag{12}$$

Theorem 1 *For Algorithm IA, for each $k = 1, 2, \ldots$,*

$$\Psi_{k+1} \le \left(1 - \sqrt{\frac{\ell}{L}}\right)\Psi_k.$$

Proof. The proof follows closely from the analysis in [2]. Define

$$\begin{aligned} \bar{\mathbf{x}}_k &= \mathbf{x}_k - \nabla f(\mathbf{x}_k)/L, \\ \bar{\bar{\mathbf{x}}}_k &= \mathbf{x}_k - \nabla f(\mathbf{x}_k)/\ell. \end{aligned}$$

The point $\bar{\mathbf{x}}_k$ satisfies $f(\bar{\mathbf{x}}_k) \le f(\mathbf{x}_k) - \|\nabla f(\mathbf{x}_k)\|^2/(2L)$. Observe that $\bar{\mathbf{x}}_k \in \mathcal{M}_{k+1}$, so $\bar{\mathbf{x}}_k$ is a candidate for the optimizer in (9) on iteration $k+1$, and hence

$$f(\mathbf{x}_{k+1}) \le f(\mathbf{x}_k) - \|\nabla f(\mathbf{x}_k)\|^2/(2L), \tag{13}$$

which is equivalent to

$$\frac{2(f(\mathbf{x}_{k+1}) - f(\mathbf{x}^*))}{\ell} \le \frac{2(f(\mathbf{x}_k) - f(\mathbf{x}^*))}{\ell} - \frac{\|\nabla f(\mathbf{x}_k)\|^2}{L\ell}. \tag{14}$$

Next, observe that a rearrangement of the definition of strong convexity yields:

$$\frac{-2\nabla f(\mathbf{x}_k)^T(\mathbf{x}_k-\mathbf{x}^*)}{\ell}+\|\mathbf{x}_k-\mathbf{x}^*\|^2\le\frac{-2(f(\mathbf{x}_k)-f(\mathbf{x}^*))}{\ell}. \tag{15}$$

We use this result in the following:

$$\begin{aligned}
\|\bar{\bar{\mathbf{x}}}_k-\mathbf{x}^*\|^2 &= \|\bar{\bar{\mathbf{x}}}_k-\mathbf{x}_k+\mathbf{x}_k-\mathbf{x}^*\|^2 \\
&= \|\bar{\bar{\mathbf{x}}}_k-\mathbf{x}_k\|^2+2(\bar{\bar{\mathbf{x}}}_k-\mathbf{x}_k)^T(\mathbf{x}_k-\mathbf{x}^*)+\|\mathbf{x}_k-\mathbf{x}^*\|^2 \\
&= \frac{\|\nabla f(\mathbf{x}_k)\|^2}{\ell^2}-\frac{2\nabla f(\mathbf{x}_k)^T(\mathbf{x}_k-\mathbf{x}^*)}{\ell}+\|\mathbf{x}_k-\mathbf{x}^*\|^2 \\
&\le \frac{\|\nabla f(\mathbf{x}_k)\|^2}{\ell^2}-\frac{2(f(\mathbf{x}_k)-f(\mathbf{x}^*))}{\ell}\ \text{(by (15))} && (16)\\
&\equiv \rho_k^2, && (17)
\end{aligned}$$

where we introduced ρ_k for the square root of the quantity in (16). Thus, $\mathbf{x}^*\in B(\bar{\bar{\mathbf{x}}}_k,\rho_k)$.

Next, define

$$\sigma_k=\|\mathbf{y}_k-\mathbf{x}^*\|, \tag{18}$$

so that $\mathbf{x}^*\in B(\mathbf{y}_k,\sigma_k)$.

By the minimality property of $\mathbf{x}_k$, we know that $\nabla f(\mathbf{x}_k)$ is orthogonal to $\mathbf{T}\mathcal{M}_k$, which contains $\mathbf{x}_k-\mathbf{y}_k$, i.e.,

$$\nabla f(\mathbf{x}_k)^T(\mathbf{y}_k-\mathbf{x}_k)=0. \tag{19}$$

Thus,

$$\begin{aligned}
\|\mathbf{y}_k-\bar{\bar{\mathbf{x}}}_k\| &= \|(\mathbf{y}_k-\mathbf{x}_k)+(\mathbf{x}_k-\bar{\bar{\mathbf{x}}}_k)\| \\
&= \|(\mathbf{y}_k-\mathbf{x}_k)+\nabla f(\mathbf{x}_k)/\ell\| \\
&= \sqrt{\|\mathbf{y}_k-\mathbf{x}_k\|^2+\|\nabla f(\mathbf{x}_k)/\ell\|^2}\ \text{(by Pythagoras's theorem)} \\
&\ge \|\nabla f(\mathbf{x}_k)\|/\ell, && (20)
\end{aligned}$$

so define

$$\delta_k=\|\nabla f(\mathbf{x}_k)\|/\ell, \tag{21}$$

to conclude that $\|\mathbf{y}_k-\bar{\bar{\mathbf{x}}}_k\|\ge\delta_k$. We have defined δ_k,ρ_k,σ_k as in Lemma 2. We need to confirm the inequality $\rho_k+\sigma_k\ge\delta_k$:

$$\begin{aligned}
\rho_k+\sigma_k &\ge \|\bar{\bar{\mathbf{x}}}_k-\mathbf{x}^*\|+\|\mathbf{y}_k-\mathbf{x}^*\| \\
&\ge \|\bar{\bar{\mathbf{x}}}_k-\mathbf{y}_k\| \\
&\ge \delta_k.
\end{aligned}$$

The other inequality is derived as follows. First, $\rho_k\le\delta_k$ since δ_k^2 is the first term in (16). Also, $\sigma_k\le\delta_k$ since

$$\begin{aligned}
\sigma_k^2 &= \|\mathbf{y}_k-\mathbf{x}^*\|^2 \\
&\le \|\mathbf{x}_k-\mathbf{x}^*\|^2 && \text{(by the optimality of } \mathbf{y}_k\text{)} \\
&\le \frac{2(f(\mathbf{x}_k)-f(\mathbf{x}^*))}{\ell} && \text{(by strong convexity)} \\
&\le \delta_k^2 && \text{(since (16) is nonnegative).}
\end{aligned}$$

Thus, $\delta_k \geq \max(\rho_k, \sigma_k)$ so $\delta_k^2 \geq |\rho_k^2 - \sigma_k^2|$.

Therefore, we can conclude from Lemma 2 that there exists a $\mathbf{z}_k^* \in \operatorname{aff}\{\bar{\bar{\mathbf{x}}}_k, \mathbf{y}_k\}$ (and hence in $\mathcal{M}_{k+1}$) such that

$$\|\mathbf{z}_k^* - \mathbf{x}^*\| \leq \xi_k^*, \tag{22}$$

where ξ_k^* is defined by (8) for $\rho_k, \sigma_k, \delta_k$ given by (17), (18) and (21) respectively. After some simplification and cancellation of (8), one arrives at:

$$(\xi_k^*)^2 = \|\mathbf{y}_k - \mathbf{x}^*\|^2 - \left(\frac{f(\mathbf{x}_k) - f(\mathbf{x}^*) + \|\mathbf{y}_k - \mathbf{x}^*\| \cdot \ell/2}{\|\nabla f(\mathbf{x}_k)\|}\right)^2. \tag{23}$$

Since $\mathbf{y}_{k+1}$ is the optimizer of (10), $\mathbf{y}_{k+1}$ is at least as close to $\mathbf{x}^*$ as $\mathbf{z}_k^*$, and hence,

$$\|\mathbf{y}_{k+1} - \mathbf{x}^*\|^2 \leq \|\mathbf{y}_k - \mathbf{x}^*\|^2 - \left(\frac{f(\mathbf{x}_k) - f(\mathbf{x}^*) + \|\mathbf{y}_k - \mathbf{x}^*\| \cdot \ell/2}{\|\nabla f(\mathbf{x}_k)\|}\right)^2.$$

Adding this inequality to (14) yields:

$$\begin{aligned}
\Psi_{k+1} &= \|\mathbf{y}_{k+1} - \mathbf{x}^*\|^2 + \frac{2(f(\mathbf{x}_{k+1}) - f(\mathbf{x}^*))}{\ell} \\
&\leq \|\mathbf{y}_k - \mathbf{x}^*\|^2 - \left(\frac{f(\mathbf{x}_k) - f(\mathbf{x}^*) + \|\mathbf{y}_k - \mathbf{x}^*\| \cdot \ell/2}{\|\nabla f(\mathbf{x}_k)\|}\right)^2 + \frac{2(f(\mathbf{x}_k) - f(\mathbf{x}^*))}{\ell} - \frac{\|\nabla f(\mathbf{x}_k)\|^2}{L\ell} \\
&\leq \|\mathbf{y}_k - \mathbf{x}^*\|^2 - \frac{2\left[f(\mathbf{x}_k) - f(\mathbf{x}^*) + \|\mathbf{y}_k - \mathbf{x}^*\| \cdot \ell/2\right]}{\sqrt{L\ell}} + \frac{2(f(\mathbf{x}_k) - f(\mathbf{x}^*))}{\ell} \\
&= \left[\|\mathbf{y}_k - \mathbf{x}^*\|^2 + \frac{2(f(\mathbf{x}_k) - f(\mathbf{x}^*))}{\ell}\right] \cdot \left(1 - \sqrt{\frac{\ell}{L}}\right) \\
&= \Psi_k \cdot \left(1 - \sqrt{\frac{\ell}{L}}\right).
\end{aligned}$$

The third line was obtained by applying the inequality $a^2 + b^2 \geq 2ab$ to the second and fourth terms of the second line. □

5 Analysis of the geometric descent algorithm

In this section we present the geometric descent (GD) algorithm due to [2] and an analysis of it. Our analysis varies slightly from the proof due to [2]; in their proof, the potential involves the term $2(f(\bar{\mathbf{x}}_k) - f(\mathbf{x}^*))/\ell$ rather than $2(f(\mathbf{x}_k) - f(\mathbf{x}^*))/\ell$. The reason for the change is to unify the analysis with the other algorithms considered in order for the NCG construction in Section 9 to be applicable.

```
Geometric Descent
x_0 := arbitrary
$\mathbf{y}_0 := \mathbf{x}_0$
$\sigma_0 := \sqrt{2}\|\nabla f(\mathbf{x}_0)\|/\ell$                                    (24)
for $k := 1, 2, \ldots$
    $\bar{\mathbf{x}}_{k-1} := \mathbf{x}_{k-1} - \nabla f(\mathbf{x}_{k-1})/L$
    $\bar{\bar{\mathbf{x}}}_{k-1} := \mathbf{x}_{k-1} - \nabla f(\mathbf{x}_{k-1})/\ell$
    Determine $\lambda_k$ according to (36) or (40) below.
    $\mathbf{y}_k := (1 - \lambda_k)\bar{\bar{\mathbf{x}}}_{k-1} + \lambda_k \mathbf{y}_{k-1}$                 (25)
    $\mathbf{x}_k := \mathrm{argmin}\{f(\mathbf{x}) : \mathbf{x} \in \mathrm{aff}\{\bar{\mathbf{x}}_{k-1}, \mathbf{y}_k\}\}$        (26)
end
```

Note: The operation in (26) is a line search and requires an inner iteration to find the optimal $\mathbf{x}$ in the specified line.

This algorithm (which is one of several variants of GD presented by the authors) is derived in [2]. It can be regarded as extracting the essential properties of $\mathbf{x}_k$ and $\mathbf{y}_k$ used the proof of Theorem 1 to obtain an implementable algorithm. Indeed, the authors use that proof that we presented in Section 4 to analyze GD rather IA.

Intuitively, the proof in the previous section shows that $\mathbf{x}_k$ need not be the minimizer in (9); it suffices for $\mathbf{x}_k$ to satisfy the two properties (13) and (19). The Geometric Descent algorithm satisfies these two properties with a "dogleg" step in (26) that combines a gradient step with a step toward $\mathbf{y}_k$. Property (13) is satisfied because $f(\mathbf{x}_k) \le f(\bar{\mathbf{x}}_{k-1})$, and property (19) is satisfied because of the minimality of $\mathbf{x}_k$ with respect to $\mathrm{aff}\{\bar{\mathbf{x}}_{k-1}, \mathbf{y}_k\}$.

The proof also shows that it suffices to take a $\mathbf{y}_{k+1}$ that satisfies (22) rather than solving (10).

We now turn to the computation of λ_k and the associated issues with the radii ρ_k, σ_k. Recall that ρ_k^{IA} from (16) and σ_k^{IA} from (18) both involve $\mathbf{x}^*$ and hence are unimplementable. The difficulty with σ_k is straightforward to resolve: define σ_k to be an upper on $\|\mathbf{y}_k - \mathbf{x}^*\|$ rather than its exact value, and ensure inductively that σ_{k+1} is an upper bound on $\|\mathbf{y}_{k+1} - \mathbf{x}^*\|$.

The difficulty with (16) is resolved using offsets denoted by γ_k, a clever device from [2]. As in (16) and (17),

$$\begin{aligned}
\|\bar{\bar{\mathbf{x}}}_k - \mathbf{x}^*\|^2 &\le \frac{\|\nabla f(\mathbf{x}_k)\|^2}{\ell^2} - \frac{2(f(\mathbf{x}_k) - f(\mathbf{x}^*))}{\ell} \text{ (by (15))} && (27)\\
&\equiv \rho_k^2, && (28)\\
&\equiv \tilde{\rho}_k^2 - \gamma_k && (29)
\end{aligned}$$

where

$$\tilde{\rho}_k = \frac{\|\nabla f(\mathbf{x}_k)\|}{\ell}, \tag{30}$$

$$\gamma_k = \frac{2(f(\mathbf{x}_k) - f(\mathbf{x}^*))}{\ell}. \tag{31}$$

Let $\sigma_0, \sigma_1, \ldots,$ be a sequence of positive scalars such that $\sigma_k \geq \|\mathbf{y}_k - \mathbf{x}^*\|$ for all $k = 0, 1, \ldots,$ and suppose that

$$\tilde{\sigma}_k = (\sigma_k^2 + \gamma_k)^{1/2}. \tag{32}$$

Thus, we have the relationships:

$$\begin{aligned} \tilde{\sigma}_k^2 &= \sigma_k^2 + \gamma_k, \\ \tilde{\rho}_k^2 &= \rho_k^2 + \gamma_k. \end{aligned}$$

Note that $\tilde{\rho}_k$ is easily computable on the kth iteration, while $\tilde{\sigma}_k$ can be updated recursively. The rationale of these definitions is as follows. From (6), one sees that if σ^2 and ρ^2 are both incremented by the same constant additive term γ_k, then λ^* is unaffected. Also, it follows from this observation and from (8) that if σ^2 and ρ^2 are both incremented by γ_k, then $(\xi^*)^2$ is also incremented by γ_k. Thus, the GD algorithm works throughout with radii whose squares are incremented by γ_k. This increment γ_k changes from one iteration to the next and hence must be adjusted at the start of each iteration (see (38) below).

In more detail, the sequence of computations is as follows. Assuming inductively that $\tilde{\sigma}_{k-1}$ is already known, compute as follows:

$$\tilde{\rho}_{k-1} := \frac{\|\nabla f(\mathbf{x}_{k-1})\|}{\ell} \tag{33}$$

$$\text{if } \tilde{\sigma}_{k-1}^2 \leq 2\tilde{\rho}_{k-1}^2 \tag{34}$$

$$\delta_{k-1} := \|\mathbf{y}_{k-1} - \bar{\bar{\mathbf{x}}}_{k-1}\|, \tag{35}$$

$$\lambda_k := \frac{\delta_{k-1}^2 + \tilde{\rho}_{k-1}^2 - \tilde{\sigma}_{k-1}^2}{2\delta_{k-1}^2}, \qquad \text{(as in (6))} \tag{36}$$

$$\tilde{\xi}_k^* := \frac{1}{2}\sqrt{2\tilde{\rho}_{k-1}^2 + 2\tilde{\sigma}_{k-1}^2 - \delta_{k-1}^2 - \frac{(\tilde{\rho}_{k-1}^2 - \tilde{\sigma}_{k-1}^2)^2}{\delta_{k-1}^2}}, \qquad \text{(as in (8))} \tag{37}$$

$$\tilde{\sigma}_k := \sqrt{(\tilde{\xi}_k^*)^2 - \gamma_{k-1} + \gamma_k} \tag{38}$$

$$\text{else} \tag{39}$$

$$\lambda_k := 0 \tag{40}$$

$$\tilde{\sigma}_k := \sqrt{\tilde{\rho}_{k-1}^2 - \gamma_{k-1} + \gamma_k}. \tag{41}$$

Although computation of γ_k alone requires prior knowledge of $\mathbf{x}^*$, the difference $\gamma_k - \gamma_{k-1}$ appearing in (38) and (41) does not, as is evident from (31). In the theorems below it is confirmed that the square roots in (38) and (41) take nonnegative arguments.

The convergence of the GD algorithm is proved via two theorems, which are both variants of theorems due to [2].

Before stating and proving the two theorems, we establish two inequalities. As noted earlier, (19) holds for GD, and hence so does (20). Then it follows from (20) combined with (33), (35) that

$$\tilde{\rho}_{k-1} = \frac{\|\nabla f(\mathbf{x}_{k-1})\|}{\ell} \le \delta_{k-1}. \tag{42}$$

Next, regarding $\gamma_k - \gamma_{k-1}$ appearing in (38) and (41), observe

$$\begin{aligned}\gamma_k - \gamma_{k-1} &= \frac{2(f(\mathbf{x}_k) - f(\mathbf{x}_{k-1}))}{\ell} \\ &\le \frac{2(f(\bar{\mathbf{x}}_{k-1}) - f(\mathbf{x}_{k-1}))}{\ell} \\ &\le -\frac{\|\nabla f(\mathbf{x}_{k-1})\|^2}{L\ell} \end{aligned} \tag{43}$$

where the second line follows because $f(\mathbf{x}_k) \le f(\bar{\mathbf{x}}_{k-1})$ by (26) while the third follows by (13).

Theorem 2 *For all $k = 1, \ldots,$*

$$\tilde{\sigma}_k^2 \ge \|\mathbf{y}_k - \mathbf{x}^*\|^2 + \frac{2(f(\mathbf{x}_k) - f(\mathbf{x}^*))}{\ell}. \tag{44}$$

Proof. Note that the statement of the theorem may be equivalently written,

$$\tilde{\sigma}_k^2 \ge \|\mathbf{y}_k - \mathbf{x}^*\|^2 + \gamma_k.$$

The proof is by induction. The base case is that $\tilde{\sigma}_0^2 \ge \|\mathbf{y}_0 - \mathbf{x}^*\|^2 + \gamma_0$, i.e., Both terms may be bounded by noting that strong convexity applied to the two points $\mathbf{x}_0, \mathbf{x}^*$ and rearranged may be written:

$$-\frac{2}{\ell}\nabla f(\mathbf{x}_0)^T(\mathbf{x}^* - \mathbf{x}_0) \ge \|\mathbf{x}_0 - \mathbf{x}^*\|^2 + \gamma_0.$$

Again by strong convexity, $\|\mathbf{x}_0 - \mathbf{x}^*\| \le \|\nabla f(\mathbf{x}_0)\|/\ell$, so we can apply this inequality and the Cauchy-Schwarz inequality on the left-hand side to obtain

$$\frac{2\|\nabla f(\mathbf{x}_0)\|^2}{\ell^2} \ge \|\mathbf{x}_0 - \mathbf{x}^*\|^2 + \gamma_0.$$

Thus, the bound $\tilde{\sigma}_0^2 \ge \|\mathbf{y}_0 - \mathbf{x}^*\|^2 + \gamma_0$ is assured by (24).

For the induction case, assume $k \ge 1$ and the induction hypothesis

$$\tilde{\sigma}_{k-1}^2 \ge \|\mathbf{y}_{k-1} - \mathbf{x}^*\|^2 + \gamma_{k-1}.$$

There are two possibilities depending on the "if"-statement (34). First, suppose the condition of (34) holds, which may be rewritten as $\tilde{\sigma}_{k-1}^2 - \tilde{\rho}_{k-1}^2 \le \tilde{\rho}_{k-1}^2$. By (42), this implies $\delta_{k-1}^2 \geq \tilde{\sigma}_{k-1}^2 - \tilde{\rho}_{k-1}^2$, and we already know from (42) that $\delta_{k-1}^2 \geq \tilde{\rho}_{k-1}^2 \geq \tilde{\rho}_{k-1}^2 - \tilde{\sigma}_{k-1}^2$. The conclusion from all these inequalities is

$$\delta_{k-1}^2 \geq \tilde{\rho}_{k-1}^2 \geq |\tilde{\rho}_{k-1}^2 - \tilde{\sigma}_{k-1}^2|. \tag{45}$$

Referring now to Lemma 2, make the following identifications:

$$\begin{aligned}
\mathbf{x} &= \bar{\bar{\mathbf{x}}}_{k-1}, \\
\mathbf{y} &= \mathbf{y}_{k-1}, \\
\rho^2 &= \tilde{\rho}_{k-1}^2 - \gamma_{k-1}, \\
\sigma^2 &= \tilde{\sigma}_{k-1}^2 - \gamma_{k-1}, \\
\delta &= \delta_{k-1},
\end{aligned}$$

in order to apply the lemma. The condition $\rho+\sigma \geq \delta$ follows immediately since we already have assumed by induction that $\mathbf{x}^* \in B(\mathbf{y}, \sigma)$ and it follows from (28) that $\mathbf{x}^* \in B(\mathbf{x}, \rho)$, thus implying that $B(\mathbf{x}, \rho) \cap B(\mathbf{y}, \sigma) \neq \emptyset$. The condition $\delta^2 \geq \rho^2 - \sigma^2$ follows because $\delta_{k-1} \geq \tilde{\rho}_{k-1}$ as in (45). The condition $\delta^2 \geq \sigma^2 - \rho^2$ follows because $\delta_k^2 \geq \tilde{\sigma}_{k-1}^2 - \tilde{\rho}_{k-1}^2$ (as established in (45)). Therefore, by the lemma, if we define λ^*

$$\lambda^* = \frac{\delta^2 + \rho^2 - \sigma^2}{2\delta^2} = \frac{\delta_{k-1}^2 + \tilde{\rho}_{k-1}^2 - \tilde{\sigma}_{k-1}^2}{2\delta_{k-1}^2},$$

i.e., the formula for λ_k in (36) (notice that the two terms γ_{k-1} cancel), and we define $\mathbf{y}_k$ as in (25), then

$$\mathbf{x}^* \in B(\mathbf{y}_k, \xi) \tag{46}$$

where

$$\begin{aligned}
\xi^2 &= \frac{1}{4}\left(2\rho^2 + 2\sigma^2 - \delta^2 - \frac{1}{\delta^2}(\rho^2 - \sigma^2)^2\right) \\
&= \frac{1}{4}\left(2\tilde{\rho}_{k-1}^2 + 2\tilde{\sigma}_{k-1}^2 - 4\gamma_{k-1} - \delta_{k-1}^2 - \frac{1}{\delta_{k-1}^2}\left(\tilde{\rho}_{k-1}^2 - \tilde{\sigma}_{k-1}^2\right)^2\right) \\
&= (\tilde{\xi}_k^*)^2 - \gamma_{k-1} && \text{(by (37))} \\
&= \tilde{\sigma}_k^2 - \gamma_k, && \text{(by (38))}.
\end{aligned}$$

This establishes the theorem in the first case.

If the condition in (34) fails, then $\lambda_k = 0$ as in (40), implying from (25) that $\mathbf{y}_k = \bar{\bar{\mathbf{x}}}_{k-1}$. Then $\|\mathbf{y}_k - \mathbf{x}^*\|^2 \leq \tilde{\rho}_{k-1}^2 - \gamma_{k-1}$ by (29), implying by (41) that $\|\mathbf{y}_k - \mathbf{x}^*\|^2 \leq \tilde{\sigma}_k^2 - \gamma_k$, thus establishing the theorem in the second case. □

Theorem 3 *For each $k = 1, 2, \ldots,$*

$$\tilde{\sigma}_k^2 \leq \left(1 - \sqrt{\frac{\ell}{L}}\right)\tilde{\sigma}_{k-1}^2. \tag{47}$$

Proof. We again take two cases depending on whether the condition in (34) holds. If it holds, then (45) in the preceding proof holds. Observe that the function $x \mapsto x + C/x$ for $C > 0$ is unimodal on $(0, \infty)$ with a minimizer at $\sqrt{C}$, which means that if all the other parameters are fixed, the maximizing choice for δ_{k-1}^2 in (37) is $\delta_*^2 = |\tilde{\rho}_{k-1}^2 - \tilde{\sigma}_{k-1}^2|$. Therefore, by unimodality combined with the ordering $\delta_{k-1} \geq \tilde{\rho}_{k-1} \geq \delta_*$ (which is (45)), the right-hand side of (37) can only increase if we replace δ_{k-1} by $\tilde{\rho}_{k-1}$, thus obtaining,

$$\begin{aligned}
(\tilde{\xi}_k^*)^2 &= \frac{1}{4}\left(2\tilde{\rho}_{k-1}^2 + 2\tilde{\sigma}_{k-1}^2 - \delta_{k-1}^2 - \frac{(\tilde{\rho}_{k-1}^2 - \tilde{\sigma}_{k-1}^2)^2}{\delta_{k-1}^2}\right) \\
&\leq \frac{1}{4}\left(2\tilde{\rho}_{k-1}^2 + 2\tilde{\sigma}_{k-1}^2 - \tilde{\rho}_{k-1}^2 - \frac{(\tilde{\rho}_{k-1}^2 - \tilde{\sigma}_{k-1}^2)^2}{\tilde{\rho}_{k-1}^2}\right) \\
&= \tilde{\sigma}_{k-1}^2 - \frac{\tilde{\sigma}_{k-1}^4}{4\tilde{\rho}_{k-1}^2}.
\end{aligned}$$

Therefore,

$$\begin{aligned}
\tilde{\sigma}_k^2 &= (\tilde{\xi}_k^*)^2 - \gamma_{k-1} + \gamma_k \\
&\leq \tilde{\sigma}_{k-1}^2 - \frac{\tilde{\sigma}_{k-1}^4}{4\tilde{\rho}_{k-1}^2} - \gamma_{k-1} + \gamma_k \\
&= \tilde{\sigma}_{k-1}^2 - \frac{\tilde{\sigma}_{k-1}^4}{4\|\nabla f(\mathbf{x}_{k-1})\|^2/\ell^2} - \gamma_{k-1} + \gamma_k \\
&\leq \tilde{\sigma}_{k-1}^2 - \frac{\tilde{\sigma}_{k-1}^4}{4\|\nabla f(\mathbf{x}_{k-1})\|^2/\ell^2} - \frac{\|\nabla f(\mathbf{x}_{k-1})\|^2}{L\ell} && \text{(by (43))} \\
&\leq \tilde{\sigma}_{k-1}^2 - 2 \cdot \frac{\tilde{\sigma}_{k-1}^2}{2\|\nabla f(\mathbf{x}_{k-1})\|/\ell} \cdot \frac{\|\nabla f(\mathbf{x}_{k-1})\|}{\sqrt{L\ell}} && \text{(since } a^2 + b^2 \geq 2ab\text{)} \\
&= \tilde{\sigma}_{k-1}^2 \left(1 - \sqrt{\frac{\ell}{L}}\right).
\end{aligned}$$

In the other case, $\tilde{\sigma}_{k-1}^2/2 \geq \tilde{\rho}_{k-1}^2$ so we obtain

$$\begin{aligned}
\tilde{\sigma}_k^2 &= \tilde{\rho}_{k-1}^2 - \gamma_{k-1} + \gamma_k && \text{(by (41))} \\
&= \frac{\|\nabla f(\mathbf{x}_{k-1})\|^2}{\ell^2} - \gamma_{k-1} + \gamma_k \\
&\leq \frac{\|\nabla f(\mathbf{x}_{k-1})\|^2}{\ell^2} - \frac{\|\nabla f(\mathbf{x}_{k-1})\|^2}{L\ell} && \text{(by (43))} \\
&= \frac{\|\nabla f(\mathbf{x}_{k-1})\|^2}{\ell^2}(1 - \ell/L) \\
&= \tilde{\rho}_{k-1}^2(1 - \ell/L) \\
&\leq \tilde{\sigma}_{k-1}^2(1 - \ell/L)/2 && \text{(by the hypothesis of the case).}
\end{aligned}$$

It is a simple matter to confirm that $(1-\epsilon)/2 \leq (1-\sqrt{\epsilon})$ for any $\epsilon \in [0,1]$, thus establishing the theorem in this case. □

A point to make about the GD algorithm and its analysis is that the potential $\tilde{\sigma}_k^2$ is computable on every iteration of the algorithm, i.e., it does not require prior knowledge of $\mathbf{x}^*$. (It does, however, require prior knowledge of ℓ.) The implementability of the potential can also be deduced from the original presentation of [2], although the computability is not further used therein.

6 Analysis of linear conjugate gradient

The linear conjugate gradient (LCG) algorithm for minimizing $f(\mathbf{x}) = \mathbf{x}^T A\mathbf{x}/2 - \mathbf{b}^T\mathbf{x}$, where A is a symmetric positive definite matrix, is due to Hestenes and Stiefel [6] and is as follows.

```
Linear Conjugate Gradient
$\mathbf{x}_0 :=$ arbitrary
$\mathbf{r}_0 := \mathbf{b} - A\mathbf{x}_0$
$\mathbf{p}_1 := \mathbf{r}_0$
for $k := 1, 2, \ldots,$
    $\alpha_k := \frac{\mathbf{r}_{k-1}^T\mathbf{r}_{k-1}}{\mathbf{p}_k^T A\mathbf{p}_k}$      (48)
    $\mathbf{x}_k := \mathbf{x}_{k-1} + \alpha_k\mathbf{p}_k$      (49)
    $\mathbf{r}_k := \mathbf{r}_{k-1} - \alpha_k A\mathbf{p}_k$      (50)
    $\beta_{k+1} := \frac{\mathbf{r}_k^T\mathbf{r}_k}{\mathbf{r}_{k-1}^T\mathbf{r}_{k-1}}$      (51)
    $\mathbf{p}_{k+1} := \beta_{k+1}\mathbf{p}_k + \mathbf{r}_k$      (52)
end
```

We now show that linear conjugate gradient (LCG) exactly implements Algorithm IA, and therefore also satisfies the bound of Theorem 1. This is perhaps surprising because LCG does not have prior information about $\mathbf{x}^*$. The following key results about LCG are from the original paper:

Theorem 4 *([6]) Let $\mathcal{V}_k = \mathbf{x}_0 + \mathrm{span}\{\mathbf{r}_0, \ldots, \mathbf{r}_{k-1}\}$ in LCG. Then*

(a) An equivalent formula is $\mathcal{V}_k = \mathbf{x}_0 + \mathrm{span}\{\mathbf{p}_1, \ldots, \mathbf{p}_k\}$,

(b) $\mathbf{x}_k$ is the minimizer of $f(\mathbf{x})$ over $\mathcal{V}_k$,

(c) $\mathbf{r}_k = -\nabla f(\mathbf{x}_k)$, and

(d) $\mathbf{x}_k + \tau_k\mathbf{p}_k$ is the minimizer of $\|\mathbf{x} - \mathbf{x}^*\|$ over $\mathcal{V}_k$, where

$$\tau_k = \frac{2(f(\mathbf{x}_k) - f(\mathbf{x}^*))}{\|\mathbf{r}_{k-1}\|^2}. \tag{53}$$

Part (b) appears as Theorem 4:3 of [6], while parts (a) and (c) are not stated explicitly. All of (a)–(c) are covered by most textbook treatments of LCG. Part (d) appears as Theorem 6:5 and is less well known.

The following theorem establishes the claim that Algorithm LCG implements IA. The principal result is part (b). The remaining parts are necessary to support the induction proof.

Theorem 5 *Suppose IA and LCG are applied to the same quadratic function with the same* $\mathbf{x}_0$*. Let the sequences of iterates be denoted* $\mathbf{x}_k^{\mathrm{IA}}$ *and* $\mathbf{x}_k^{\mathrm{LCG}}$ *respectively. Then for each* $k = 1, 2, \ldots,$

(a) $\mathcal{M}_k \subset \mathcal{V}_k$,

(b) $\mathbf{x}_k^{\mathrm{LCG}} = \mathbf{x}_k^{\mathrm{IA}}$*, and*

(c) $\mathbf{y}_k = \mathbf{x}_k^{\mathrm{LCG}} + \tau_k \mathbf{p}_k$.

Proof. For the $k = 1$ case, observe that $\mathbf{p}_1 = -\nabla f(\mathbf{x}_0)$ so $\mathcal{M}_1 = \mathcal{V}_1$. Since $\mathbf{x}_1^{\mathrm{LCG}}$ minimizes $f(\mathbf{x})$ over $\mathcal{V}_1$ while $\mathbf{x}_1^{\mathrm{IA}}$ minimizes $f(\mathbf{x})$ over $\mathcal{M}_1$, we conclude $\mathbf{x}_1^{IA} = \mathbf{x}_1^{\mathrm{LCG}}$. For (c), observe that $\mathbf{y}_1$ minimizes $\|\mathbf{y} - \mathbf{x}^*\|$ over $\mathcal{M}_1$ by (10), while $\mathbf{x}_1^{\mathrm{LCG}} + \tau_1 \mathbf{p}_1$ minimizes the same function over the same affine space, so (c) is established.

Now assuming (a)–(c) hold for some $k \geq 1$, we establish them for $k + 1$. We will write $\mathbf{x}_k$ for both $\mathbf{x}_k^{\mathrm{LCG}}$ and $\mathbf{x}_k^{\mathrm{IA}}$ since these are equal by induction. For (a), we start with $\mathcal{M}_{k+1} = \mathbf{x}_k + \mathrm{span}\{\mathbf{x}_k - \mathbf{y}_k, \nabla f(\mathbf{x}_k)\}$. We already know from (c) that $\mathbf{y}_k - \mathbf{x}_k = \tau_k \mathbf{p}_k \in \mathbf{T}\mathcal{V}_k \subset \mathbf{T}\mathcal{V}_{k+1}$. Also, $\nabla f(\mathbf{x}_k) = -\mathbf{r}_k = \beta_{k+1}\mathbf{p}_k - \mathbf{p}_{k+1}$ (by (52)), hence $\nabla f(\mathbf{x}_k) \in \mathbf{T}\mathcal{V}_{k+1}$ Thus, $\mathbf{T}\mathcal{M}_{k+1} \subset \mathbf{T}\mathcal{V}_{k+1}$, so showing $\mathcal{M}_{k+1} \subset \mathcal{V}_{k+1}$ is reduced to finding a single common point, and we may take $\mathbf{x}_k$ to be this point.

For (b), $\mathbf{x}_{k+1}^{\mathrm{LCG}}$ minimizes $f(\mathbf{x})$ over $\mathcal{V}_{k+1}$ by Theorem 4(a). We also know that $\mathbf{x}_{k+1}^{\mathrm{LCG}} \in \mathcal{M}_{k+1}$ because

$$\begin{aligned}
\mathbf{x}_{k+1}^{\mathrm{LCG}} &= \mathbf{x}_k + \alpha_{k+1}\mathbf{p}_{k+1} && \text{(by (49))}\\
&= \mathbf{x}_k + \alpha_{k+1}(\mathbf{r}_k + \beta_{k+1}\mathbf{p}_k) && \text{(by (52))}\\
&= \mathbf{x}_k + \alpha_{k+1}(-\nabla f(\mathbf{x}_k) + \beta_{k+1}\mathbf{p}_k) && \text{(by Theorem 4(b))}\\
&= \mathbf{x}_k + \alpha_{k+1}(-\nabla f(\mathbf{x}_k) + \beta_{k+1}(\mathbf{y}_k - \mathbf{x}_k)/\tau_k) && \text{(by induction, part (c))}\\
&\in \mathbf{x}_k + \mathrm{span}\{\nabla f(\mathbf{x}_k), \mathbf{y}_k - \mathbf{x}_k\}.
\end{aligned}$$

Since $\mathcal{M}_{k+1} \subset \mathcal{V}_{k+1}$ according to part (a), the optimality of $\mathbf{x}_{k+1}^{\mathrm{LCG}}$ with respect to $\mathcal{V}_{k+1}$ implies that it is also optimal for $f(\mathbf{x})$ with respect to $\mathcal{M}_{k+1}$, hence $\mathbf{x}_{k+1}^{\mathrm{IA}} = \mathbf{x}_{k+1}^{\mathrm{LCG}}$. Thus, write $\mathbf{x}_{k+1}$ for both vectors for the remainder of the argument.

A similar argument establishes (c). First, we observe that $\mathbf{x}_{k+1} + \tau_{k+1}\mathbf{p}_{k+1}$ lies in $\mathcal{M}_{k+1}$ because $\mathbf{x}_{k+1} + \tau_{k+1}\mathbf{p}_{k+1} = \mathbf{x}_k + (\alpha_{k+1} + \tau_{k+1})\mathbf{p}_{k+1}$, and the we can proceed as in the last paragraph except with $(\alpha_{k+1} + \tau_{k+1})$ playing the role of α_{k+1}. Next, $\mathbf{x}_{k+1} + \tau_{k+1}\mathbf{p}_{k+1}$ minimizes $\|\mathbf{x} - \mathbf{x}^*\|$ over $\mathcal{V}_{k+1}$ by Theorem 4(d). Thus, $\mathbf{x}_{k+1} + \tau_{k+1}\mathbf{p}_{k+1}$ must be the same as $\mathbf{y}_{k+1}$. This concludes the induction. □

The surprising aspect of this analysis is that LCG exactly identifies $\mathcal{M}_k$ that appears in Algorithm IA despite not ever computing $\mathbf{y}_k^{\mathrm{IA}}$. The reason is that the line $\mathrm{aff}\{\mathbf{x}_k, \mathbf{y}_k^{\mathrm{IA}}\}$ agrees with the line $\mathbf{x}_k + \mathrm{span}\{\mathbf{p}_k\}$, which is computed by LCG.

7 A computable potential for linear conjugate gradient

The analysis in the preceding section shows that LCG implements the idealized algorithm. However, the decrease in the potential cannot be measured during the algorithm because (53) requires prior knowledge of the optimizer. In this section, we observe that the GD potential can also be used for LCG, yielding a computable potential. An application of this potential will be presented in Section 9.

We define an auxiliary sequence of vectors $\mathbf{y}_k$ for minimizing $\|\mathbf{y} - \mathbf{x}^*\|$ that are not optimal (we already know from Theorem 4 that $\mathbf{x}_k + \tau_k \mathbf{p}_k$ is optimal) but nonetheless converge to $\mathbf{x}^*$ sufficiently fast to establish the necessary decrease in the potential.

In particular, we exactly mimic the equations that define the quantities $\tilde{\sigma}_k$, $\tilde{\rho}_k$, λ_k, δ_k, $\mathbf{y}_k$ in GD, and modify only $\mathbf{x}_k$ so that it is computed using the LCG algorithm instead of the GD algorithm. The two same two theorems that held for GD also hold for LCG:

Theorem 6 *For each $k = 0, 1, 2, \ldots$,*

$$\|\mathbf{x}^* - \mathbf{y}_k\|^2 + \frac{2(f(\mathbf{x}_k) - f(\mathbf{x}^*))}{\ell} \le \tilde{\sigma}_k^2.$$

Theorem 7 *For each $k = 1, 2, \ldots$,*

$$\tilde{\sigma}_k^2 \le \left(1 - \sqrt{\frac{\ell}{L}}\right) \tilde{\sigma}_{k-1}^2.$$

The following observations about $\mathbf{x}_k$ computed in LCG show that the same proofs of the previous theorems work for LCG.

First, $\mathbf{y}_k \in \mathcal{V}_k$, whereas $\mathbf{r}_k$ is orthogonal to $\mathbf{T}\mathcal{V}_k$ (a well known property of LCG), and thus $\nabla f(\mathbf{x}_k)^T(\mathbf{y}_k - \mathbf{x}_k) = 0$. This is (19), which was a necessary ingredient in the proof of GD.

Second, (43) still holds because the LCG step from $\mathbf{x}_{k-1}$ to $\mathbf{x}_k$ improves f at least as much as the step from $\mathbf{x}_{k-1}$ to $\bar{\mathbf{x}}_{k-1}$, since $\bar{\mathbf{x}}_k$ lies in the Krylov space $\mathcal{V}_k$ where $\mathbf{x}_k$ is optimal for f over this space.

8 Accelerated gradient

The Accelerated Gradient (AG) algorithm of Nesterov is an even looser approximation to IA than GD in the sense that there is no optimization subproblem per iteration; instead, all step lengths are fixed. For this section, let us define

$$\kappa = \frac{L}{\ell}, \tag{54}$$

because this ratio, sometimes called the *condition number* of f, is used often throughout the algorithm and analysis.

The algorithm is as follows.

```
Accelerated Gradient
$\mathbf{x}_0 :=$ arbitrary
$\mathbf{w}_0 := \mathbf{x}_0$
for $k := 1, 2, \ldots,$
    $\mathbf{x}_k := \mathbf{w}_{k-1} - \nabla f(\mathbf{w}_{k-1})/L$    (55)
    $\mathbf{w}_k := \mathbf{x}_k + \theta(\mathbf{x}_k - \mathbf{x}_{k-1})$ where    (56)
        $\theta = \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}.$    (57)
end
```

Note that some versions of AG in the literature vary the choice of θ (e.g., see [13]).

For the purpose of analysis, let us define the following auxiliary sequences of vectors and scalars:

$$
\begin{aligned}
\overline{\overline{\mathbf{w}}}_k &= \mathbf{w}_k - \nabla f(\mathbf{w}_k)/\ell, && (k = 0, 1, \ldots) && (58)\\
\mathbf{y}_0 &= \mathbf{x}_0, \\
\mathbf{y}_k &= \mathbf{x}_k + \tau(\mathbf{x}_k - \mathbf{x}_{k-1}), && (k = 1, 2, \ldots) && (59)\\
\tilde{\sigma}_0 &= \sqrt{2}\|\nabla f(\mathbf{x}_0)\|/\ell, && && (60)\\
\tilde{\sigma}_{k+1} &= \Bigg[(1-\kappa^{-1/2})\tilde{\sigma}_k^2 + \frac{2(f(\mathbf{x}_{k+1}) - f(\mathbf{w}_k))}{\ell} \\
&\qquad + \frac{\|\nabla f(\mathbf{w}_k)\|^2}{L\ell} - (\kappa^{1/2} - \kappa^{-1/2})\|\mathbf{w}_k - \mathbf{x}_k\|^2\Bigg]^{1/2} && (k = 0, 1, 2, \ldots) && (61)
\end{aligned}
$$

where

$$\tau = \sqrt{\kappa} - 1. \tag{62}$$

We prove two main results about these scalars. The first shows that $\tilde{\sigma}_k$ is decreasing at the appropriate rate, while the second shows that it is an upper bound on the distance to the optimizer.

Theorem 8 *For each $k = 0, 1, 2, \ldots,$*

$$\tilde{\sigma}_{k+1}^2 \le (1 - \kappa^{-1/2})\tilde{\sigma}_k^2. \tag{63}$$

Proof. By squaring both sides of (61), it is apparent that (63) reduces to showing:

$$\frac{2(f(\mathbf{x}_{k+1}) - f(\mathbf{w}_k))}{\ell} + \frac{\|\nabla f(\mathbf{w}_k)\|^2}{L\ell} - (\kappa^{1/2} - \kappa^{-1/2})\|\mathbf{w}_k - \mathbf{x}_k\|^2 \le 0.$$

Clearly it suffices to show

$$\frac{2(f(\mathbf{x}_{k+1}) - f(\mathbf{w}_k))}{\ell} + \frac{\|\nabla f(\mathbf{w}_k)\|^2}{L\ell} \le 0.$$

This follows immediately from (55), which implies that $f(\mathbf{x}_{k+1}) \le f(\mathbf{w}_k) - \|\nabla f(\mathbf{w}_k)\|^2/(2L)$. □

Theorem 9 *For each* $k = 0, 1, 2 \ldots$,

$$\|\mathbf{y}_k - \mathbf{x}^*\|^2 + \frac{2(f(\mathbf{x}_k) - f(\mathbf{x}^*))}{\ell} \le \tilde{\sigma}_k^2. \tag{64}$$

Proof. The proof of (64) is by induction on k. We start by deriving some preliminary relationships. It is clear from (56) and (59) that $\mathbf{w}_k, \mathbf{x}_k, \mathbf{y}_k$ are collinear and the tangent to their common line is $\mathbf{x}_k - \mathbf{x}_{k-1}$, hence we easily obtain from these equations:

$$\begin{aligned}
\mathbf{y}_k &= \mathbf{x}_k + \frac{\tau}{\theta}(\mathbf{w}_k - \mathbf{x}_k) && \text{(by (56) and (59))} \\
&= \mathbf{x}_k + (\sqrt{\kappa} + 1)(\mathbf{w}_k - \mathbf{x}_k) && \text{(by (57) and (62))} \\
&= (\sqrt{\kappa} + 1)\mathbf{w}_k - \sqrt{\kappa}\mathbf{x}_k. && (65)
\end{aligned}$$

For the $k = 0$ case, (64) follows from the initialization in (60) and strong convexity.

We now assume the result (64) holds for k and establish it for $k = 1$. The proof relies on Lemma 1, so first we must argue that $\mathbf{y}_{k+1}$ lies on the line segment between $\mathbf{y}_k$ and $\overline{\overline{\mathbf{w}}}_k$. This is the content of the following derivation:

$$\begin{aligned}
\mathbf{y}_{k+1} &= \mathbf{x}_{k+1} + (\sqrt{\kappa} - 1)(\mathbf{x}_{k+1} - \mathbf{x}_k) && \text{(by (59) and (62))} \\
&= \sqrt{\kappa}\mathbf{x}_{k+1} - (\sqrt{\kappa} - 1)\mathbf{x}_k \\
&= \sqrt{\kappa}\mathbf{w}_k - (\sqrt{\kappa} - 1)\mathbf{x}_k - \sqrt{\kappa}\nabla f(\mathbf{w}_k)/L && \text{(by (55))} \\
&= \sqrt{\kappa}\mathbf{w}_k - (\sqrt{\kappa} - 1)\mathbf{x}_k - \kappa^{-1/2}\nabla f(\mathbf{w}_k)/\ell && \text{(by (54))} \\
&= \kappa^{-1/2}\mathbf{w}_k + (1 - \kappa^{-1/2})((\sqrt{\kappa} + 1)\mathbf{w}_k - \sqrt{\kappa}\mathbf{x}_k) \\
&\qquad - \kappa^{-1/2}\nabla f(\mathbf{w}_k)/\ell \\
&= \kappa^{-1/2}\mathbf{w}_k + (1 - \kappa^{-1/2})\mathbf{y}_k - \kappa^{-1/2}\nabla f(\mathbf{w}_k)/\ell && \text{(by (65))} \\
&= \kappa^{-1/2}\overline{\overline{\mathbf{w}}}_k + (1 - \kappa^{-1/2})\mathbf{y}_k. && \text{(by (58))} \quad (66)
\end{aligned}$$

We take $\mathbf{x}$, $\mathbf{y}$ appearing in Lemma 1 to be $\overline{\overline{\mathbf{w}}}_k$, $\mathbf{y}_k$ respectively. Next we need to define δ, ρ, σ to be used in Lemma. In the case of ρ and we copy the definitions used in the analysis of IA:

$$\begin{aligned}
\|\overline{\overline{\mathbf{w}}}_k - \mathbf{x}^*\|^2 &\le \frac{\|\nabla f(\mathbf{w}_k)\|^2}{\ell^2} - \frac{2(f(\mathbf{w}_k) - f(\mathbf{x}^*))}{\ell} && \text{(by (16))} \\
&\equiv \rho^2, && (67)
\end{aligned}$$

For σ, we use the induction hypothesis:

$$\begin{aligned}
\|\mathbf{y}_k - \mathbf{x}^*\|^2 &\le \tilde{\sigma}_k^2 - \frac{2(f(\mathbf{x}_k) - f(\mathbf{x}^*))}{\ell} \\
&\equiv \sigma^2. && (68)
\end{aligned}$$

In the case of δ, we have:

$$\begin{align*}
\|\overline{\overline{\mathbf{w}}}_k - \mathbf{y}_k\|^2 &= \|\mathbf{w}_k - \nabla f(\mathbf{w}_k)/\ell - (\sqrt{\kappa}+1)\mathbf{w}_k + \sqrt{\kappa}\mathbf{x}_k\|^2 && \text{(by (58) and (65))} \\
&= \|\sqrt{\kappa}(\mathbf{x}_k - \mathbf{w}_k) - \nabla f(\mathbf{w}_k)/\ell\|^2 \\
&= \kappa\|\mathbf{w}_k - \mathbf{x}_k\|^2 + \frac{2\sqrt{\kappa}(\mathbf{w}_k - \mathbf{x}_k)^T \nabla f(\mathbf{w}_k)}{\ell} \\
&\qquad + \frac{\|\nabla f(\mathbf{w}_k)\|^2}{\ell^2} \\
&\geq \kappa\|\mathbf{w}_k - \mathbf{x}_k\|^2 + \frac{2\sqrt{\kappa}(f(\mathbf{w}_k) - f(\mathbf{x}_k))}{\ell} \\
&\qquad + \sqrt{\kappa}\|\mathbf{w}_k - \mathbf{x}_k\|^2 + \frac{\|\nabla f(\mathbf{w}_k)\|^2}{\ell^2} && \text{(by strong convexity)} \\
&= (\kappa + \sqrt{\kappa})\|\mathbf{w}_k - \mathbf{x}_k\|^2 + \frac{2\sqrt{\kappa}(f(\mathbf{w}_k) - f(\mathbf{x}_k))}{\ell} \tag{69} \\
&\qquad + \frac{\|\nabla f(\mathbf{w}_k)\|^2}{\ell^2} \\
&\equiv \delta^2. \tag{70}
\end{align*}$$

From (66), $\lambda = 1 - \kappa^{-1/2}$ (and hence $\lambda(1-\lambda) = \kappa^{-1/2} - \kappa^{-1}$). Finally, by Lemma 1,

$$\begin{aligned}
\|\mathbf{y}_{k+1} - \mathbf{x}^*\|^2 &\leq \kappa^{-1/2}\rho^2 + (1-\kappa^{-1/2})\sigma^2 - (\kappa^{-1/2} - \kappa^{-1})\delta \\
&= \frac{\|\nabla f(\mathbf{w}_k)\|^2}{L\ell} - \frac{2(f(\mathbf{w}_k) - f(\mathbf{x}^*))}{\ell} + (1-\kappa^{-1/2})\tilde{\sigma}_k^2 - (\kappa^{1/2} - \kappa^{-1/2})\|\mathbf{w}_k - \mathbf{x}_k\|^2 \\
&= \tilde{\sigma}_{k+1}^2 - \frac{2(f(\mathbf{x}_{k+1}) - f(\mathbf{x}^*))}{\ell},
\end{aligned}$$

thus completing the induction. The second line was obtained by substituting (67), (68), (70) in the first line followed by cancellation of like terms. The third line was obtained from (61). □

9 A hybrid nonlinear conjugate gradient

In this section, we propose a hybrid nonlinear conjugate gradient algorithm with a convergence guarantee for smooth, strongly convex functions which is related to an algorithm from the PhD thesis of the first author [10]. Classical nonlinear conjugate gradient methods such as the Fletcher-Reeves and Polak-Ribière methods are known to have poor worst-case performance for this class of functions–worse even than steepest descent. See [11] for more information. The method developed in this section guarantees $O(\log(1/\epsilon)\sqrt{L/\ell})$ convergence, the best possible, and reduces to the optimal LCG algorithm in the case of a quadratic function.

The algorithm proposed below uses classical nonlinear conjugate gradient steps mixed with geometric descent steps. The rationale for developing this algorithm is as follows. Classical NCG, although it has no global convergence bound even for strongly convex functions, behaves well on “nearly quadratic” functions. For typical objective functions occurring in practice, nearly quadratic behavior is expected close to the solution. Therefore, a method that can switch between steps with a guaranteed complexity and NCG steps has the possibility of outperforming both methods.

A summary of the algorithm is as follows. At the beginning of iteration k, the algorithm has a quadruple $(\mathbf{x}_{k-1}, \mathbf{y}_{k-1}, \mathbf{p}_{k-1}, \tilde{\sigma}_{k-1})$. From this quadruple, a step of nonlinear conjugate gradient can be applied. For the line search, the line-search function of α, namely, $f(\mathbf{x}_{k-1} + \alpha\mathbf{p}_{k-1})$, is approximated by a univariate quadratic, whose quadratic coefficient is obtained by computing $\mathbf{p}_{k-1}^T \nabla^2 f(\mathbf{x}_{k-1})\mathbf{p}_{k-1}$ using reverse-mode automatic differentiation. This approximation is exact in the case that f itself is a quadratic function, in which case the hybrid algorithm reproduces the steps of linear conjugate gradient.

The hybrid algorithm then computes $\mathbf{x}_k = \mathbf{x}_{k-1} + \alpha\mathbf{p}_{k-1}$ and computes $\mathbf{y}_k$ as in the GD algorithm. It checks whether f has decreased and whether $\tilde{\sigma}_k^2 \le (1 - \sqrt{\ell/L})\tilde{\sigma}_{k-1}^2$. If so, the iteration is over, and the nonlinear CG step is accepted. If not, then a GD step is taken instead. The detailed specification of the algorithm is as follows.

```
**Hybrid Nonlinear Conjugate Gradient**
$\mathbf{x}_0 :=$ arbitrary;   $\mathbf{y}_0 := \mathbf{x}_0$;   $\mathbf{p}_0 := \mathbf{0}$
$\mathbf{g}_{-1} := \mathbf{0}; \tilde{\sigma}_0 := \sqrt{2}\|\nabla f(\mathbf{x}_0)\|/\ell$
for $k = 1, 2, \ldots,$
    $\mathbf{g}_{k-1} := \nabla f(\mathbf{x}_{k-1})$
    $(\mathbf{x}_k^{\mathrm{CG}}, \mathbf{p}_k) := \mathit{CGSTEP}(\mathbf{x}_{k-1}, \mathbf{p}_{k-1}, \mathbf{g}_{k-2}, \mathbf{g}_{k-1}, k)$
    $(\mathbf{y}_k, \tilde{\xi}_k^*) := \mathit{YCOMPUTE}(\mathbf{x}_{k-1}, \mathbf{g}_{k-1}, \mathbf{y}_{k-1}, \tilde{\sigma}_{k-1})$
    $\hat{\gamma}_k^{\mathrm{CG}} := 2(f(\mathbf{x}_k^{\mathrm{CG}}) - f(\mathbf{x}_{k-1}))/\ell$                    (71)
    $\tilde{\sigma}_k^{\mathrm{CG}} := \sqrt{(\tilde{\xi}_k^*)^2 + \hat{\gamma}_k^{\mathrm{CG}}}$
    if $\hat{\gamma}_k^{\mathrm{CG}} < 0$ and $(\tilde{\sigma}_k^{\mathrm{CG}})^2 \le \left(1 - \sqrt{\ell/L}\right)\tilde{\sigma}_{k-1}^2$
        $\mathbf{x}_k := \mathbf{x}_{k-1}^{\mathrm{CG}}$
        $\tilde{\sigma}_k := \tilde{\sigma}_k^{\mathrm{CG}}$
    else
        $\bar{\mathbf{x}}_{k-1} := \mathbf{x}_{k-1} - \mathbf{g}/L$
        $\mathbf{x}_k := \mathrm{argmin}\{f(\mathbf{x}) : \mathbf{x} \in \mathrm{aff}\{\bar{\mathbf{x}}_{k-1}, \mathbf{y}_k\}\}$       (72)
        $\hat{\gamma}_k := 2(f(\mathbf{x}_k) - f(\mathbf{x}_{k-1}))/\ell$                    (73)
        $\tilde{\sigma}_k := \sqrt{(\tilde{\xi}_k^*)^2 + \hat{\gamma}_k}$
    end
end
```

```
  **Function** $CGSTEP(\mathbf{x}_{k-1}, \mathbf{p}_{k-1}, \mathbf{g}_{k-2}, \mathbf{g}_{k-1}, k)$
  if $k == 1$
    $\mathbf{p}_k := -\mathbf{g}_{k-1}$
  else
    $\mathbf{z} := \mathbf{g}_{k-1} - \mathbf{g}_{k-2}$
    $\beta_k := \frac{1}{\mathbf{z}^T \mathbf{p}_{k-1}} \left( \mathbf{z} - \frac{2\mathbf{p}_{k-1} \|\mathbf{z}\|^2}{\mathbf{z}^T \mathbf{p}_{k-1}} \right)^T \mathbf{g}_{k-1}$          (74)
    $\mathbf{p}_k := \beta_k \mathbf{p}_{k-1} - \mathbf{g}_{k-1}$
  end
  $\alpha_k := -\frac{\mathbf{p}_k^T \mathbf{g}_{k-1}}{\mathbf{p}_k^T \nabla^2 f(\mathbf{x}_{k-1}) \mathbf{p}_k}$          (75)
  $\mathbf{x}_k := \mathbf{x}_{k-1} + \alpha_k \mathbf{p}_k$
  return $(\mathbf{x}_k, \mathbf{p}_k)$
```

```
**Function** $YCOMPUTE(\mathbf{x}_{k-1}, \mathbf{g}_{k-1}, \mathbf{y}_{k-1}, \tilde{\sigma}_{k-1})$
$\bar{\bar{\mathbf{x}}}_{k-1} := \mathbf{x}_{k-1} - \mathbf{g}_{k-1}/\ell$
$\tilde{\rho}_{k-1} := \|\mathbf{g}_{k-1}\|/\ell$
if $\tilde{\sigma}_{k-1}^2 \leq 2\tilde{\rho}_{k-1}^2$
  $\delta_{k-1} := \|\mathbf{y}_{k-1} - \bar{\bar{\mathbf{x}}}_{k-1}\|$          (76)
  if $\delta_{k-1} > \tilde{\rho}_{k-1}$ and $\tilde{\rho}_{k-1} > |\tilde{\rho}_{k-1}^2 - \tilde{\sigma}_{k-1}|^{1/2}$          (77)
    $\lambda_k := \frac{\delta_{k-1}^2 + \tilde{\rho}_{k-1}^2 - \tilde{\sigma}_{k-1}^2}{2\delta_{k-1}^2}$
    $\tilde{\xi}_k^* := \frac{1}{2} \sqrt{2\tilde{\rho}_{k-1}^2 + 2\tilde{\sigma}_{k-1}^2 - \delta_{k-1}^2 - \frac{(\tilde{\rho}_{k-1}^2 - \tilde{\sigma}_{k-1}^2)^2}{\delta_{k-1}^2}}$
  else
    $\lambda_k := 1$
    $\tilde{\xi}_k^* := \tilde{\sigma}_{k-1}$
  end
else
  $\lambda_k := 0$
  $\tilde{\xi}_k^* := \tilde{\rho}_{k-1}$
end
$\mathbf{y}_k := (1 - \lambda_k)\bar{\bar{\mathbf{x}}}_{k-1} + \lambda_k \mathbf{y}_{k-1}$
return $(\mathbf{y}_k, \tilde{\xi}_k^*)$
```

Some remarks on this procedure are as follows. The variable $\hat{\gamma}_k$ in (71) and (73) stands for $\gamma_k - \gamma_{k-1}$, where γ_k is defined as in (31). The line-search implicit in (72) is carried out with a univariate Newton method. Because we have not made sufficient assumptions about f to guarantee convergence of Newton's method, the Newton method is safeguarded with a bisection method. The univariate second derivative of f needed for the Newton method can be computed using reverse-mode automatic differentiation in time proportional to evaluate $f(\mathbf{x})$ (refer to [14]). This univariate second derivative is also needed in (75).

The formula for β_k in (74) is from the CG-Descent algorithm of Hager and Zhang [5]. As mentioned earlier, the formula for α_k appearing in (75) is based on a univariate quadratic Taylor expansion of the line-search function at $\mathbf{x}_k$ in the direction $\mathbf{p}_k$. This formula is exact if f itself is quadratic, but in all other cases it is speculative. However, if it yields a poor answer, the overall algorithm is still robust because when the CG step gives a poor answer, the GD algorithm serves as a backup.

The main theorem about this method, which follows from the material presented so far, is as follows.

Theorem 10 *Assuming exact line-search in* (72)*, the Hybrid NCG algorithm produces a sequence of iterates* $(\mathbf{x}_k, \mathbf{y}_k, \sigma_k)$ *satisfying* (44) *and* (47)*. Furthermore, if* $f(\mathbf{x})$ *is a quadratic function, then Hybrid NCG produces the same sequence of iterates as linear conjugate gradient.*

We now turn to three important numerical issues with this method. The first issue to note is that function *YCOMPUTE* has an "if" statement (77) not present in (33)–(41) when the GD algorithm was described. In the case of the "exact" GD algorithm, the condition of the if-statement is guaranteed to hold as established in Section 5. However, because the line-search is only approximate, (19) does not hold exactly, and therefore the condition of the (77) may occasionally fail. In this case, we safeguard its failure by defining $\mathbf{y}_k := \mathbf{y}_{k-1}$ and $\sigma_k := \sigma_{k-1}$ (so that $\tilde{\sigma}_k$ is updated only due to the decrease in the objective), i.e., we keep the same containing sphere for the optimizer as on the previous step.

The second numerical issue concerns the computation of δ_{k-1} in (76). This formula is prone to roundoff error as the algorithm converges because $\mathbf{y}_{k-1}$ and $\bar{\bar{\mathbf{x}}}_{k-1}$ both tend to $\mathbf{x}^*$ in the limit. In our implementation, we addressed this issue by maintaining a separate program variable storing the vector $\mathbf{y}_k - \mathbf{x}_k$. This vector is updated using a recurrent formula that is straightforward to derive; the recurrence updates $\mathbf{y}_k - \mathbf{x}_k$ using vectors that also tend to $\mathbf{0}$. Given an accurate representation of $\mathbf{y}_k - \mathbf{x}_k$ it is clear that (76) can be computed without significant roundoff issues.

The third numerical issue concerns the subtractions in (71) and (73), which are also prone to roundoff error as $\mathbf{x}_k$ converges. These errors could upset the computation of $\tilde{\sigma}_k$. Our implementation addressed this using "computational divided differences"; see, e.g., [16].

10 Computational Results

We implemented four methods: Geometric Descent (GD), Accelerated Gradient (AG), Hybrid Nonlinear Conjugate Gradient (HyNCG, described in the preceding section), and NCG. NCG stands for nonlinear conjugate gradient using the Hager-Zhang formula for β_k given by (74). (The entirety of their NCG is called "CG-Descent"; however, we did not implement other aspects of CG-Descent such as the line search.) The line search used by GD, HyNCG and NCG is based on Newton's method and is safeguarded with a bisection. The techniques to address numerical issues described in the preceding section were applied in GD, HyNCG and NCG. (The line search and the numerical techniques are not needed for AG).

We applied these four methods to two problem classes: approximate BPDN and hinge-loss halfspace classification.

BPDN (basis pursuit denoising) refers to the unconstrained convex optimization problem:

$$\min \|A\mathbf{x} - \mathbf{b}\|^2 + \lambda \|\mathbf{x}\|_1$$

in which $\lambda > 0$ and $A \in \mathbb{R}^{m \times n}$ has fewer rows than columns, so that the problem is neither strongly convex nor smooth. However, the following approximation (called APBDN) is both smooth and strongly convex on any bounded domain:

$$\min \|A\mathbf{x} - \mathbf{b}\|^2 + \lambda \sum_{i=1}^{n} \sqrt{x_i^2 + \delta}$$

where $\delta > 0$ is a fixed scalar. It is easy to see that as $\delta \to 0$, the original problem is recovered. As $\delta \to 0$, $\ell \to 0$ and $L \to \infty$, where ℓ, L are the moduli of strong, smooth convexity.

In our tests of ABPDN we took A to be a subset of $\sqrt{n}$ rows of the discrete-cosine transform matrix of size $n \times n$, where n is an even power of 2. (This matrix and its transpose, although dense, can be applied in $O(n \log n)$ operations.) The subset of rows was selected to be those numbered by the first $m = \sqrt{n}$ prime integers in order to get reproducible pseudorandomness in the choices. Similarly, in order to obtain a pseudorandom $\mathbf{b}$, we selected $\mathbf{b} \in \mathbb{R}^m$ according to the formula $b_i = \sin(i^2)$. The value of λ was fixed at 10^{-3} in all tests; the convergence criterion was $\|\nabla f(\mathbf{x}_k)\| \leq 10^{-8}$. Finally, we varied $\delta = 10^{-2}, 10^{-3}, 10^{-4}$ and we tried both $n = 65536$ and $n = 262144$.

The second test-case is the hinge-loss (HL) function for half-space identification taken from [2], which is as follows: $f(\mathbf{x}) = H(\mathbf{b} \circ (A\mathbf{x})) + \lambda \|\mathbf{x}\|^2/2$, where A is a given $m \times n$ matrix, $\mathbf{b}$ is a given m-vector of ± 1, '$\circ$' denotes Hadamard product (i.e., the entrywise product of two vectors), $\lambda > 0$ is a regularization parameter, and $H(\mathbf{v}) = \sum_{i=1}^{m} h(v_i)$ where

$$h(v) = \begin{cases} 0.5 - v, & v \leq 0, \\ (1-v)^2/2, & v \in [0,1], \\ 0, & v \geq 1. \end{cases}$$

Minimizing $f(\cdot)$ corresponds to finding a hyperplane determined by $\mathbf{x}$ of the form $U = \{\mathbf{z} \in \mathbb{R}^n : \mathbf{x}^T\mathbf{z} = 0\}$ such that rows i of A, $i = 1, \ldots, m$, for which $b_i = 1$ lie on one side

	GD	AG	NCG	HyNCG
ABPDN, $n = 65536$, $\delta = 10^{-2}$	58060	DNC	12345	6757
ABPDN, $n = 65536$, $\delta = 10^{-3}$	315020	DNC	DNC	109703
ABPDN, $n = 65536$, $\delta = 10^{-4}$	584976	DNC	DNC	292474
ABPDN, $n = 262144$, $\delta = 10^{-2}$	7565	34758	488	186
ABPDN, $n = 262144$, $\delta = 10^{-3}$	783022	DNC	DNC	15751
ABPDN, $n = 262144$, $\delta = 10^{-4}$	DNC	DNC	DNC	DNC
HL, $m = 200000$, $\lambda = 0.3$	154	13170	112	44
HL, $m = 200000$, $\lambda = 0.03$	151	29218	110	43
HL, $m = 200000$, $\lambda = 0.003$	152	58793	113	47

Table 1: Number of iterations (see text for details) of four algorithms on nine synthetic test cases.

of U (i.e., $A(i,:)\mathbf{x} > 0$) while rows i of A for which $b_i = -1$ lie on the opposite side (i.e., $A(i,:)\mathbf{x} < 0$). The objective function penalizes misclassified points as well as penalizing a large value of $\mathbf{x}$.

This function is smooth and strongly convex. As $\lambda \to 0$, the strong convexity parameter ℓ vanishes.

Unlike [2], who test GD applied to this function on data sets available on the web, we have tested the four algorithms on synthetic data for the purpose of better control over experimental conditions. In our tests, $m = 200000$, $n = 447$ (so that $n \approx \sqrt{m}$), $\lambda = 3 \cdot 10^{-1}, 3 \cdot 10^{-2}, 3 \cdot 10^{-3}$. For each i, $i = 1, \ldots, m$, $b(i) = \pm 1$ chosen at random with probability 0.5. If $b(i) = 1$, then $A(i,:) = [1, \ldots, 1]/\sqrt{n} + \mathbf{w}_i^T$, where $\mathbf{w}_i$ is a noise vector chosen as a spherical Gaussian with covariance matrix $\mathrm{diag}(\sigma^2, \ldots, \sigma^2)$, where $\sigma = 0.4$. If $b(i) = -1$, then $A(i,:) = -[1, \ldots, 1]/\sqrt{n} + \mathbf{w}_i^T$. For these tests, the convergence test was $\|\nabla f(\mathbf{x}_k)\| \le 10^{-6}$.

The results of all tests are shown in Table 1. For NCG and GD, the numbers in this table are the number of inner iterations (line search steps), which is the dominant cost in these algorithms. In the case of HyNCG, we have reported the sum of the number of CG steps (which do not require a line-search) plus the number of inner line-search iterations. For AG we have reported the number of outer iterations. The notation DNC indicates that the algorithm did achieve the requisite tolerance after 10^5 outer iterations.

One sees from the table that HyNCG was superior in every test case, sometimes by a wide margin. An unexpected feature of the table, for which we currently do not have an explanation, is that in the case of the HL suite of problems, the number of iterations was nearly invariant with respect to variation in λ, except for AG, whose running time grows steadily with decreasing λ.

11 Conclusions

We have demonstrated that a single computable potential bounds the convergence of three algorithms, conjugate gradient, accelerated gradient and geometric descent. We have also pointed out other connections between the algorithms, namely, their relationship to an idealized algorithm and their relationship to the Bubeck-Lee-Singh lemma. The existence of this potential enables the formulation of a hybrid method for convex optimization that duplicates the steps of conjugate gradient in the case of conjugate gradient but nonetheless achieves the optimal complexity for general smooth, strongly convex problems. Directions for future work include the following.

- The hybrid algorithm requires prior knowledge of ℓ, L; it would be interesting to develop an algorithm with the same guarantees that does not need prior knowledge of them. Note that linear conjugate gradient does not need any such prior knowledge of the coefficient matrix A.
- It would be interesting to establish a theoretical result about the improved performance of the hybrid algorithm in the case of "nearly quadratic" functions.
- Although accelerated gradient has been extended well beyond the realm of unconstrained smooth, strongly convex functions, none of the other algorithms has been. It would be interesting to extend the conjugate gradient ideas outside this space. See, for example, [8].

References

[1] D. P. Bertsekas. *Nonlinear programming (2nd edition).* Athena Scientific, 1999.

[2] S. Bubeck, Y. T. Lee, and Mohit Singh. A geometric alternative to Nesterov's accelerated gradient descent. `http://arxiv.org/abs/1506.08187`, 2015.

[3] D. Drusvyatskiy, M. Fazel, and S. Roy. An optimal first order method based on optimal quadratic averaging. `http://arxiv.org/abs/1604.06543`, 2016.

[4] G. H. Golub and C. F. Van Loan. *Matrix Computations, 2nd Edition.* Johns Hopkins University Press, Baltimore, 1989.

[5] W. Hager and H. Zhang. A new conjugate gradient method with guaranteed descent and an efficient line search. *SIAM J. Optimization*, 16:170–192, 2005.

[6] Magnus Rudolph Hestenes and Eduard Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49:409–436, 1952.

[7] Jean-Baptiste Hiriart-Urruty and Claude Lemaréchal. *Fundamentals of convex analysis.* Springer, 2012.

[8] S. Karimi and S. Vavasis. IMRO: A proximal quasi-Newton method for solving l1-regularized least squares problem. `http://arxiv.org/abs/1401.4220`, 2014.

[9] S. Karimi and S. Vavasis. A unified convergence bound for conjugate gradient and accelerated gradient. `http://arxiv.org/abs/1605.00320`, 2016.

[10] Sahar Karimi. *On the relationship between conjugate gradient and optimal first-order methods for convex optimization.* PhD thesis, University of Waterloo, 2014.

[11] A. S. Nemirovsky and D. B. Yudin. *Problem Complexity and Method Efficiency in Optimization.* John Wiley and Sons, Chichester, 1983. Translated by E. R. Dawson from *Slozhnost' Zadach i Effektivnost' Metodov Optimizatsii*, 1979, Glavnaya redaktsiya fiziko-matematicheskoi literatury, Izdatelstva "Nauka".

[12] Y. Nesterov. A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$. *Doklady AN SSSR (translated as Soviet Math. Dokl.)*, 269(3):543–547, 1983.

[13] Y. Nesterov. *Introductory Lectures on Convex Optimization.* Kluwer, 2003.

[14] J. Nocedal and S. Wright. *Numerical Optimization, 2nd Edition.* Springer, New York, 2006.

[15] R. Polyak. Modified barrier functions (theory and methods). *Mathematical Programming*, 54:177–222, 1992.

[16] S. Vavasis. Some notes on applying computational divided differencing in optimization. `http://arxiv.org/abs/1307.4097`, 2013.