ACCELERATED OPTIMIZATION WITH ORTHOGONALITY CONSTRAINTS

JONATHAN W. SIEGEL

Abstract. We develop a generalization of Nesterov’s accelerated gradient descent method which is designed to deal with orthogonality constraints. To demonstrate the effectiveness of our method, we perform numerical experiments which demonstrate that the number of iterations scales with the square root of the condition number, and also compare with existing state-of-the-art Riemannian optimization methods. Our experiments show that our method outperforms existing state-of-the-art methods on some large, ill-conditioned problems.

Key words. Riemannian optimization, Stiefel manifold, accelerated gradient descent, eigenvector problems, electronic structure calculations

AMS subject classifications. 65K05, 65N25, 90C30, 90C48

1. Introduction. Optimization problems over the set of orthonormal matrices arise naturally in many scientific and engineering problems. Most notably, eigenfunction and electronic structure calculations involve minimizing functions over the set of orthonormal matrices [1, 3, 8, 17]. In these applications, the objective functions are smooth but often ill-conditioned. There are also more recent applications which involve non-smooth objectives, most notably the calculation of compressed modes [12], which involve an $L^1$ penalization of variational problems arising in physics.

In this paper, we consider optimization problems with orthogonality constraints, i.e. problems of the form

$$\arg\min_{X^T X = I_k} f(X)$$

where $X$ is an $n \times k$ matrix, $I_k$ is the identity matrix, and $f$ is a smooth function. The manifold of orthonormal matrices over which we are optimizing is referred to as the Stiefel manifold in the literature. Many methods have been proposed for solving 1.1, including variants of gradient descent, Newton’s method, quasi-Newton methods, and non-linear conjugate gradient methods [1, 3, 15, 18, 4]. However, existing methods can suffer from slow convergence when the problem is ill-conditioned, by which we mean that the Hessian of $f$ at (or near) the minimizer is ill-conditioned [3]. Such problems are of particular interest, since they arise when doing electronic structure calculations, or when solving non-smooth problems by smoothing the objective, for instance. Moreover, preconditioning such problems can be very difficult due to the manifold constraint.

In an attempt to solve ill-conditioned problems more efficiently, we develop an extension of the well-known Nesterov’s accelerated gradient descent algorithm [9] designed for optimizing functions on the Stiefel manifold. For the class of smooth, strongly convex functions on $\mathbb{R}^n$, accelerated gradient descent obtains an asymptotically optimal iteration complexity of $O(\sqrt{\kappa})$, compared to $O(\kappa)$ for gradient descent with optimal step size selection [2] (here $\kappa$ is the condition number of the problem). Our method extends this convergence behavior from $\mathbb{R}^n$ to the Stiefel manifold, thus
providing an efficient method for solving ill-conditioned optimization problems with orthogonality constraints.

Other work on accelerated gradient methods on manifolds includes [7] and [16]. In [7] an accelerated gradient method on general manifolds is presented. However, their algorithm involves solving a non-linear equation involving both the metric on the manifold and the objective function \( f \). Unfortunately, solving this equation is only feasible for the special type of model problem which they consider and cannot be generally applied to arbitrary optimization problems on the Stiefel manifold. In [16], a theory is developed which shows that a certain type of accelerated method can achieve accelerated convergence locally. However, their method involves calculating a geodesic logarithm in every iteration and hasn’t yet been implemented. In constrast, our method only involves very simple linear algebra calculations in each iteration and can be run efficiently on large problems.

The paper is organized as follows. In section 2, we briefly introduce the necessary notation and ideas from differential geometry. In section 3, we discuss accelerated gradient descent on \( \mathbb{R}^n \). We recall results which are relevant to our work. In section 4, we detail the design of our method. One of the key ingredients is an efficient procedure for performing approximate extrapolation and interpolation on the manifold, which we believe could be useful in developing other optimization methods. In section 5, we show numerical results which provide evidence that our method achieves the desired iteration complexity. Finally, in section 6, we present comparisons with other optimization methods on the Stiefel manifold. We show that our method outperforms existing state of the art methods on some large, ill-conditioned problems.

2. Riemannian Manifolds. Let \( M \) be a smooth manifold and \( x \in M \). We denote the tangent space of \( M \) at \( x \) by \( T_x M \) and the dual tangent space by \( (T_x M)^* \). We denote the tangent bundle of \( M \), i.e. the space of pairs \((x,v)\) with \( x \in M \) and \( v \in T_x M \), by \( TM \), and likewise the dual tangent bundle by \( (TM)^* \).

Suppose \( f \) is a \( C^1 \) function on \( M \). Then the derivative of \( f \) at \( x \in M \), which we denote by \( \nabla f(x) \) is naturally an element of \( (T_x M)^* \). In particular, it is the linear functional which maps a tangent vector \( v \in T_x M \) to the directional derivative of \( f \) in the direction \( v \).

If \( M \) is a Riemannian manifold, then each tangent space \( T_x M \) is equipped with a positive definite inner product \( g : T_x M \times T_x M \to \mathbb{R} \). Because it is positive definite, \( g \) induces a norm on the tangent space

\[ \|v\|_g^2 = g(v,v) = g_{ij}v^i v^j \]

For the last expression above we have fixed a coordinate system and \( g_{ij} \) are the (covariant) components of \( g \) in this coordinate system (we are using the Einstein summation notation).

We also have a dual norm on the dual space

\[ \|w\|_{g^*} = \sup_{\|v\|_g = 1} \langle w,v \rangle = g^{ij}w_i w_j \]

Here \( g^{ij} \) satisfying \( g^{ij}g_{jk} = \delta^i_k \) are the (contravariant) components of \( g \).

Additionally, the inner product \( g \) provides an isomorphism \( \phi_g : (T_x M)^* \to T_x M \) with the property that

\[ \|w\|_{g^*}^2 = \langle w, \phi_g(w) \rangle = \|\phi_g(w)\|_g^2 \]
In terms of the metric, the map $\phi_g$ is given by raising the indices of $w$, i.e.

$$ \phi_g^i(w)^j = g^{ij}w_j $$

and its inverse $\phi_g^{-1} : T_xM \rightarrow (T_xM)^*$ is given by lowering the indices of $v$, i.e.

$$ \phi_g^{-1}(v)_i = g_{ij}v^j $$

Given a smooth curve $c : [0, 1] \rightarrow M$, the length of the curve is defined by

$$ l(c) = \int_0^1 \|c'(t)\|_g dt $$

which allows us to define the distance between points $x, y \in M$ as follows

$$ d(x, y) = \inf_{c : [0, 1] \rightarrow M} \int_0^1 \|c'(t)\|_g^2 dt $$

Note that the minimizer in the above expression is not unique, even if $d(x, y)$ is very small. This is due to the fact that the length of a curve is invariant under reparametrizations. The unit speed geodesic between $x$ and $y$ is given by

$$ \arg \min_{c : [0, d(x, y)] \rightarrow M} \int_0^d(x, y) \|c'(t)\|_g^2 dt $$

which exists and is unique as long as $x$ and $y$ are sufficiently close. The minimizing curve $c$ will satisfy the geodesic equations

$$ \frac{d^2c^i}{dt^2} + \Gamma^i_{kl} \frac{dc^k}{dt} \frac{dc^l}{dt} = 0 $$

where $\Gamma^i_{kl}$ are the Christoffel symbols.

Assuming that the geodesic equations can be solved globally in time (which is true for the Stiefel manifold that we are interested in) we can define the exponential map $\exp_x : T_xM \rightarrow M$ as follows

$$ \exp_x(v) = c_v(1) $$

where $c_v$ is the (unique) unit speed geodesic satisfying $c'_v(0) = v$.

Let $x \in M$ and $f \in C^2(M)$. We define the following quadratic form on $T_xM$, called the Hessian of $f$ (which generalizes the Hessian in Euclidean space), as follows

$$ Hf(x)(v) = \frac{d^2f(\exp_x(tv))}{dt^2} \bigg|_{t=0} $$

The Hessian can be given in matrix form as follows

$$ Hf(x)_{ij} = \frac{\partial^2f}{\partial_i \partial_j} - \Gamma^k_{ij} \frac{\partial f}{\partial_k} $$

The condition number of $Hf(x)$ is the condition number of the matrix representation with respect to an orthonormal basis of $T_xM$. Alternatively, it is the ratio

$$ \kappa(Hf(x)) = \frac{\sup_{\|v\|_g = 1} Hf(x)(v)}{\inf_{\|v\|_g = 1} Hf(x)(v)} $$

In Riemannian optimization, geodesics are often expensive to compute exactly, which leads to the concept of a retraction [1].
Definition 2.1. Let $M$ be a (smooth) manifold. A retraction on $M$ is a (smooth) map $R : TM \to M$ (here $TM$ denotes the tangent bundle of $M$) satisfying for all $x \in M$ and $v \in T_x M$.

\[(2.14)\quad R(x, 0) = x\]

\[(2.15)\quad \left. \frac{d}{dt} \right|_{t=0} R(x, tv) = v\]

(Here I write $R(x, v)$ for the image of the point $(x, v) \in TM$ under $R$.)

We proceed to specialize each of the above concepts to the Stiefel manifold, on which we will ultimately do our calculations.

2.1. The Stiefel Manifold. The Stiefel manifold $S_{n,k}$ is the set of $n \times k$ orthonormal matrices, i.e.

\[S_{n,k} = \{ X \in \mathbb{R}^{n \times k} : X^T X = I_k \}\]

We begin by describing the Riemannian metric which we put on the Stiefel manifold and giving formulas for calculating inner products, raising and lowering indices and geodesics. Of course, the metric which we consider is not unique as any diffeomorphism of $M$ onto itself provides a new metric (which is the same only if the diffeomorphism is an isometry).

In fact, there are two metrics commonly put on the Stiefel manifold in the literature. One is obtained by viewing $S_{n,k} \subset \mathbb{R}^{nk}$ and considering the metric induced by the ambient space $\mathbb{R}^{nk}$. The other, called the canonical metric and which we will be considering for the remainder of this chapter, is obtained by viewing $S_{n,k} = O(n)/O(n-k)$ as the quotient of the orthogonal group $O(n)$ by the right action of $O(n-k)$. Specifically, the action is given by right multiplication by

\[(2.16)\quad \begin{bmatrix} I_{k \times k} & 0_{k \times n} \\ 0_{n \times k} & O_{(n-k) \times (n-k)} \end{bmatrix}\]

where $O_{(n-k) \times (n-k)} \in O(n-k)$. This induces a quotient metric on $S_{n,k}$. For more details on the former metric and the differences between these two viewpoints, see [3].

Before we describe the metric in more detail, we must fix a representation of the elements of $S_{n,k}$ and its tangent and dual tangent space. Throughout, the elements of $S_{n,k}$ will be represented by $n \times k$ orthonormal matrices (even though our metric is induced by viewing $S_{n,k}$ as a quotient $O(n)/O(n-k)$). The tangent space at a point $X \in S_{n,k}$ is then naturally identified with the set $T_X = \{ V \in \mathbb{R}^{n \times k} : VT X + X^T V = 0 \}$. We choose to represent the dual space by elements of the same set, i.e. $(T_X)^* = \{ W \in \mathbb{R}^{n \times k} : WT X + X^T W = 0 \}$, with the pairing between $T_X$ and $(T_X)^*$ given by $\langle V, W \rangle = \text{Tr}(V^T W)$ (i.e. the usual inner product on $\mathbb{R}^{nk}$).

Using these representations, the metric on $S_{n,k}$ is given by (see [3])

\[(2.17)\quad g(Y, Z) = \text{Tr} \left( Y^T \left( I - \frac{1}{2} XX^T \right) Z \right)\]

where $Y, Z \in T_X S_{n,k}$. The formula for the inner product on the dual space is

\[(2.18)\quad g^*(Y, Z) = \text{Tr} \left( Y^T \left( I - \frac{1}{2} XX^T \right)^{-1} Z \right)\]
for $Y, Z \in (T_{X}S_{n,k})^{*}$. Since $X$ is orthonormal, it follows that $XX^T$ is a projection, and we thus have

$$(2.19) \quad \left( I - \frac{1}{2}XX^T \right)^{-1} = (I + XX^T)$$

So we can rewrite the dual space inner product as

$$(2.20) \quad g^*(Y, Z) = Tr(Y^T (I + XX^T) Z)$$

Finally, the maps corresponding to raising and lowering the indices are

$$(2.21) \quad \phi_g(W) = \left( I - \frac{1}{2}XX^T \right)^{-1} W = (I + XX^T) W$$

and

$$(2.22) \quad \phi^g(V) = \left( I - \frac{1}{2}XX^T \right)^{T} V$$

respectively.

The advantage of using canonical metric, i.e. the metric induced by the quotient structure of $S_{n,k}$, is that geodesics can be computed using the matrix exponential. In fact, the constant-speed geodesic starting at $X \in S_{n,k}$ and moving initially in the direction $V \in T_{X}S_{n,k}$ is given by (see [3] for details)

$$(2.23) \quad X(t) = \exp(t(VX - XV) X)$$

If we give our direction via a dual vector $W$ and raise indices first, we obtain the simpler expression

$$(2.24) \quad X(t) = \exp(t(WX - XW)) X$$

Since the matrix $WX - XW$ has rank $2k$, this exponential can be calculated by diagonalizing a $2k \times 2k$ antisymmetric matrix, as shown in [3].

In [15], a more efficient retraction is introduced, which can be viewed as a Padé approximation of the above exponential. Their retraction, called the Cayley retraction, is defined on a dual vector by the formula

$$(2.25) \quad R_C(X, \phi_g(W)) = \left( I - \frac{1}{2}(WX^T - X^T W) \right)^{-1} \left( I + \frac{1}{2}(WX^T - X^T W) \right) X$$

which can be calculated using the Sherman-Morrison-Woodbury formula [13] as (with $U = [W, X]$ and $Z = [X, -W]$)

$$(2.26) \quad R_1(X, \phi_g(W)) = X + 2U(I - Z^T U)^{-1}Z^T X$$

3. Accelerated Gradient Descent. Let $f : \mathbb{R}^n \to \mathbb{R}$ be a differentiable convex function. We say that $f$ is $\mu$-strongly convex if

$$(3.1) \quad f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu}{2} \|x - y\|_2^2$$

We also say that $f$ is $L$-smooth if

$$(3.2) \quad f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2} \|x - y\|_2^2$$
One way of thinking about these definitions is that \( \mu \)-strong convexity implies that the eigenvalues of the Hessian of \( f \) at every point are \( \geq \mu \) and \( L \)-smoothness implies that the eigenvalues are \( \leq L \).

In his seminal paper [9], Nesterov introduced first-order methods which achieves the asymptotically optimal objective error for the class of \( L \)-smooth convex functions and for the class of \( L \)-smooth and \( \mu \)-strongly convex functions. These methods take the form

\[
(3.3) \quad x_0 = y_0, \; x_{n+1} = y_n - \gamma_n \nabla f(y_n), \; y_{n+1} = x_{n+1} + \alpha_n(x_{n+1} - x_n)
\]

The choice of \( \gamma_n \) and \( \alpha_n \) depend on whether the function \( f \) is strongly convex (as opposed to only convex and \( L \)-smooth), and also on the precise parameters \( \mu \) and \( L \).

If \( f \) is \( \mu \)-strongly convex and \( L \)-smooth, then setting \( \alpha_n = \frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}} \) and \( \gamma_n = 1/L \) produces the asymptotically optimal objective error of \( O((1 - \sqrt{\frac{\mu}{L}})^n) \) (compared with \( O((1 - \frac{\mu}{L})^{-n}) \) for gradient descent), as the following theorem shows.

**Theorem 3.1.** Assume that \( f \) is \( \mu \)-strongly convex and \( L \)-smooth. Let \( x^* \) be the minimizer of \( f \). If we let \( \alpha_n = \frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}} \) and \( \gamma_n = 1/L \) in (3.3), then we have that

\[
(3.4) \quad f(x_n) - f(x^*) \leq 2 \left(1 - \frac{\sqrt{\mu}}{\sqrt{L}}\right)^n (f(x_0) - f(x^*))
\]

*Proof.* See, for instance, [2]. \( \Box \)

One disadvantage of the method analyzed in Theorem 3.1 is that setting the proper step size and momentum parameter requires knowing the smoothness parameter \( L \) and the strong convexity parameter \( \alpha \).

The optimal method for \( L \)-smooth functions is more flexible. In particular, no knowledge about the smoothness parameter is needed. One can use a line search to determine the correct step size and still obtain the optimal objective error of \( O(n^{-2}) \) (compared with \( O(n^{-1}) \) for gradient descent). In particular, we have the following result (which slightly generalizes the results in [14] to obtain a larger family of accelerated schemes). For the proof, see [14].

**Theorem 3.2.** Assume that \( f \) is convex and differentiable with minimizer \( x^* \). Let \( q_n \) be any sequence of non-negative real numbers satisfying \( q_0 = 0 \) and \( (q_{n+1} + 1)^2 \leq (q_n + 2)^2 + 1 \) (in particular \( q_{n+1} \leq q_n + 1 \) works).

Then, if in iteration 3.3, \( \gamma_n \) is chosen so that \( \gamma_n \leq \gamma_{n-1} \) and \( f(x_{n+1}) \leq f(y_n) - (\gamma_n/2) \| \nabla f(y_n) \|^2 \), and \( \alpha_n = \frac{q_n}{2 + q_{n+1}} \), we have

\[
(3.5) \quad f(x_n) - f(x^*) \leq 2(\gamma_n q_n (q_n + 2))^{-1} \| x_0 - x^* \|^2
\]

Note that in the above theorem we made no assumption that \( f \) was \( L \)-smooth. This emphasizes that the method is independent of the particular value of \( L \). We choose the step size \( \gamma_n \) to provide a sufficient decrease in the objective. Such a \( \gamma_n \) can be found using a line search and will be about \( 1/L \) in the worst case (within a constant depending on the precise line search scheme).

Also, setting \( q_n = \alpha n \) and \( \gamma_n = 1/L \) for \( \alpha \leq 1 \) recovers the result from [14] (with \( r = 1 + 2/\alpha \)). In particular, the special case \( \alpha = 1 \) gives \( f(x_n) - f(x^*) \leq 2Ln^{-2} \| x_0 - x^* \|^2 \).
4. Acceleration on the Stiefel Manifold. We now come to the heart of the paper. In this section we develop a version of Nesterov’s accelerated gradient descent which is designed for efficiently optimizing functions on the Stiefel manifold. There are three main difficulties which we must overcome in developing such a method.

First, the (local, i.e. near the minimizer) strong convexity parameter $\mu$ and smoothness parameter $L$ are not known. This problem occurs when applying accelerated methods to convex functions in $\mathbb{R}^n$ as well. Theorem 3.2 shows that estimating the smoothness parameter $L$ is not an issue as we can use a line search to find a point satisfying a sufficient decrease condition. Getting around knowledge of the strong convexity parameter is a much more difficult problem.

Second, the functions which we will be minimizing are non-convex. This is due to the fact that all globally convex functions on the Stiefel manifold are constant (since the manifold is compact). Because of this, we cannot hope to obtain a global convergence rate. However, we want a method which is guaranteed to converge and which will achieve an accelerated rate once it is close enough to the (local) minimizer.

Finally, we must find an efficient way of generalizing the momentum step $y_{n+1} = x_{n+1} + \alpha_n (x_{n+1} - x_n)$ of 3.3 to the Stiefel manifold. We will develop a very efficient method for averaging and extrapolating on the Stiefel manifold, which can potentially be used to design a variety of other optimization methods as well.

In the following subsections, we will describe how each of these difficulties is overcome.

4.1. Adaptive Restart. We first address the lack of knowledge of the smoothness and strong convexity parameters. This issue arises even when considering convex optimization in $\mathbb{R}^n$. Recall that setting the proper the momentum and step size parameters for smooth strongly convex functions in iteration 3.3 requires knowing $\mu$ and $L$. In general, $\mu$ and $L$ are not known and many researchers have considered the problem of estimating them adaptively (see [10], [6] and [11], for instance).

In developing our method, we build upon the work presented in [11]. The methods introduced there are based on the following observation.

Suppose we are given a $\mu$-strongly convex, $L$-smooth function $f$. Then since $f$ is convex and $L$-smooth, we can run iteration 3.3 with the parameters given in Theorem 3.2 (setting $q_n = n$) and obtain the following objective error

\[ f(x_n) - f(x^*) \leq 2Ln^{-2}\|x_0 - x^*\|_2^2 \]

The strong convexity of $f$ now allows us to bound the iterate error by the objective error, since strong convexity implies that $(\mu/2)\|x_n - x^*\|_2^2 \leq f(x_n) - f(x^*)$. Combining this with equation 4.1 we see that

\[ \|x_n - x^*\|_2^2 \leq 4(L/\mu)n^{-2}\|x_0 - x^*\|_2^2 = 4\kappa n^{-2}\|x_0 - x^*\|_2^2 \]

where $\kappa = (L/\mu)$ is the condition number of $f$. This implies that after $n = \sqrt{8\kappa}$ iterations, we will have

\[ \|x_n - x^*\|_2^2 \leq \frac{\|x_0 - x^*\|_2^2}{2} \]

So by restarting the method (i.e. setting $x_0 = x_n$ and resetting the momentum parameter) every $\sqrt{8\kappa}$ iterations, we halve the iterate error every time we restart.
This means that it takes $O(\sqrt{\kappa \log(\epsilon)})$ iterations to attain an $\epsilon$-accurate solution and thus restarting the method at this frequency recovers the asymptotically optimal convergence rate (for $\mu$-strongly convex $L$-smooth functions).

Of course, in order to apply this scheme, we must know the condition number $\kappa$ in order to determine the correct restart frequency. To get around this, the method proposed in [11] adaptively chooses when to restart based on an observable condition on the iterates. Specifically, they consider two restart conditions

- Function Restart Scheme: Restart when $f(x_k) > f(x_{k-1})$
- Gradient Restart Scheme: Restart when $\nabla f(y_{k-1}) \cdot (x_k - x_{k-1}) > 0$

Both of these restart conditions are based upon the analysis of a quadratic objective and it is an open problem to fully analyze their behavior when applied to an arbitrary strongly convex, smooth function. However, experimental results in [11] show empirically that the adaptively restarted methods perform well in practice.

We generalize these adaptively restarted methods to solve optimization problems on the Stiefel manifold. In the next subsection, we will modify the function restart scheme to additionally address the problem of non-convexity of functions on the Stiefel manifold. Later on, we will also show how to adapt the gradient restart scheme to the manifold setting.

4.2. Restart for Non-convex Functions. When adapting accelerated gradient methods to the Stiefel manifold, we are faced with the issue that the manifold is compact and so the only convex functions are constant. Consequently, the functions which we are optimizing are necessarily non-convex. In this case the convergence results of Theorems 3.1 and 3.2 don’t apply and in fact we cannot hope for a ‘global’ convergence rate.

Instead, what we note is that in a small neighborhood of a local optimum $X^*$ the function will be strongly convex and smooth, provided that the Hessian at $X^*$ is positive definite. Moreover, the ratio of the strong convexity and smoothness parameters in this neighborhood will be close to the condition number of $\nabla^2 f(X^*)$, which we denote by $\kappa(X^*)$.

Thus the accelerated gradient method analyzed in Theorem 3.1 suggests that we should be able to find a method which achieves a convergence rate of $O((1 - \kappa(X^*)^{-1/2})^n)$ once it is close enough to the local minimum $X^*$. But since we have to deal with functions which are not globally convex, we hope to design a method which is guaranteed to converge to a local minimum even for non-convex functions, and which achieves the optimal convergence rate once it is close enough to the local minimum.

Our approach is to modify the function restart scheme considered in [11] and described in the previous section. We introduce the following restart condition, which forces a sufficient decrease in the objective.

- Modified Function Restart Scheme: Restart when

$$ f(x_{n+1}) > f(x_n) - c_R \gamma_n \|\nabla f(y_n)\|^2 $$

where $c_R$ is a parameter we take to be a small constant (recall that $\gamma_n$ is the step size at step $n$).

We now prove that with this restart condition, the algorithm converges in an appropriate sense.

**Theorem 4.1.** Let $f$ be a differentiable, $L$-smooth function, i.e. $\nabla f$ is Lipschitz with constant $L$. Assume also that $f$ is bounded below.
Consider the iteration 3.3 with step size \( \gamma_n \) chosen to satisfy \( c/L \leq \gamma_n \leq \gamma_{n-1} \) for some \( c \leq 1 \) and \( f(x_{n+1}) \leq f(y_n) - (\gamma_n/2)\|\nabla f(y_n)\|^2 \).

If this iteration is restarted whenever 4.4 holds (with the new \( \gamma_0 \) chosen to be \( \leq \gamma_n \)), then we have

\[
\lim_{n \to \infty} \|\nabla f(x_n)\|_2 \to 0
\]

Proof. Note first that our condition on the step size \( \gamma_n \) can always be satisfied, since by the \( L \)-smoothness of \( f \) we have that \( \gamma_n = c/L \) will always work.

Also note that since \( x_0 = y_0 \), the condition on the step size always guarantees that \( f(x_1) \leq f(x_0) - c_R\gamma_0\|\nabla F(y_0)\|^2 \).

So we can always run the algorithm 3.3 in a way which satisfies the conditions of the theorem.

To complete the proof, we note that the restart condition combined with the observation that we always take at least one step implies that

\[
f(x_{n+1}) \leq f(x_n) - c_R\gamma_n\|\nabla f(y_n)\|^2
\]

Summing this, we obtain

\[
c_R \sum_{n=0}^{\infty} \gamma_n\|\nabla f(y_n)\|^2_2 \leq f(x_0) - f(x_n)
\]

Since \( f \) is bounded below, say by \( M \) and \( \gamma_n \geq c/L \) we see that

\[
\sum_{n=0}^{\infty} \|\nabla f(y_n)\|^2_2 \leq \frac{L(f(x_0) - M)}{c_Rc} < \infty
\]

This implies that \( \|\nabla f(y_n)\| \to 0 \). Now we simply note that since \( f \) is \( L \)-smooth and \( x_n = y_{n-1} - \gamma_n \nabla f(y_{n-1}) \), we have that

\[
\|\nabla f(x_n)\|_2 \leq (1 + L\gamma_n)\|\nabla f(y_{n-1})\|_2 \leq (1 + L\gamma_0)\|\nabla f(y_{n-1})\|_2
\]

where the last inequality is because \( \gamma_n \leq \gamma_0 \) by assumption. Thus, \( \|\nabla f(x_n)\| \to 0 \) as desired.

4.3. Extrapolation and Interpolation on the Stiefel Manifold. We have now seen how to get around knowing the strong convexity and smoothness parameters and how to deal with non-convex functions in the process. We proceed to address the third difficulty mentioned at the beginning of the section. Namely, we consider the problem of generalizing the momentum step of 3.3

\[
Y_{n+1} = X_{n+1} + \alpha_n(X_{n+1} - X_n)
\]

to the manifold setting.

More generally, we will consider the problem of efficiently extrapolating and interpolating on the Stiefel manifold, i.e. given two points \( X, Y \in S_{n,k} \) and \( \alpha \in \mathbb{R} \), we want to calculate points \( (1 - \alpha)X + \alpha Y \) on a curve through \( X \) and \( Y \). By setting \( \alpha \in (0,1) \) this gives a way of averaging points on the manifold and by setting \( \alpha < 1 \) we can extrapolate as in 4.10.

One very simple approach would be to perform the extrapolation or interpolation in Euclidean space and then project back onto the Stiefel manifold. However,
this projection step is quite expensive. One could also replace the projection by a
reorthogonalization procedure such as Gram-Schmidt (or a QR factorization). How-
ever, this is quite inaccurate if $k$ (the number of vectors) is large and is also relatively
expensive.

The approach we take is both simpler and more efficient. What we propose for
generalizing

$$(1 - \alpha)X + \alpha Y$$

is to solve for a $V \in (T_X S_{n,k})^*$ which satisfies (here $R$ is a retraction which we have
fixed in the course of designing our method)

$$Y = R(X, \phi_g(V))$$

and to then extrapolate or average by setting

$$(1 - \alpha)X + \alpha Y = R(X, \phi_g((1 + \alpha)V))$$

Note that the use of $\phi_g$ simply allows us to work in the dual tangent space.

The obvious difficulty with this is solving equation 4.12 for $V$, i.e. finding a $V$ such
that $R(X, \phi_g(V)) = Y$ for some given $X$ and $Y$. However, if we take our retraction
to be $R_1$ from the previous section (this is the Cayley retraction introduced in [15]),
then this boils down to solving

$$(I + \frac{1}{2}(VX^T - XV^T))X = \left(I - \frac{1}{2}(VX^T - XV^T)\right)Y$$

for $V$. Since $X^T X = Y^T Y = I$, one can now easily check that $V = 2Y(I + X^T Y)^{-1}$
solves this equation (of course $V$ is not unique, one can add $XS$ to it where $S$ is
an arbitrary symmetric matrix). Thus, for this particular choice of retraction, this
problem is computationally very easy to solve (it only requires solving a $k \times k$ linear
system).

This gives us a computationally efficient procedure for averaging and extrapo-
lating on the Stiefel manifold. We have already described how this can be used to
generalize accelerated gradient methods to the Stiefel manifold. We also propose that
this averaging and extrapolation procedure could potentially be a building block in
other novel optimization algorithms on the manifold.

4.4. Gradient Restart Scheme. We can use the idea of the previous subsection
to generalize the gradient restart scheme to the Stiefel manifold. Recall that the
gradient restart scheme restarts iteration 3.3 whenever

$$\nabla f(y_{k-1}) \cdot (x_k - x_{k-1}) > 0$$

We begin by noting that $x_k = y_{k-1} - \gamma_{k-1} \nabla f(y_k)$ and so we can rewrite this condition as

$$-\gamma_{k-1} \| \nabla f(y_{k-1}) \|^2 + \nabla f(y_{k-1}) \cdot (y_{k-1} - x_{k-1}) > 0$$

Now it is clear that on the manifold $\| \nabla f(y_{k-1}) \|^2$ should become $\| \nabla f(y_{k-1}) \|^2_{g^*}$. The
tricky part is generalizing $\nabla f(y_{k-1}) \cdot (y_{k-1} - x_{k-1})$. What we propose is to solve for
a $V \in (S_{y_{k-1}})^*$ such that

$$x_{k-1} = R(y_{k-1}, \phi_g(V))$$
**Algorithm 4.1** Accelerated Gradient Descent with Function Restart Scheme

**Data:** \( f \) a smooth function, \( \epsilon \) a tolerance, \( c_R \) a small restart parameter

**Result:** A point \( X_n \) such that \( \|\nabla f(X_n)\|_g^* < \epsilon \)

\[
\begin{align*}
X_0 &\leftarrow \text{initial point} \\
Y_0 &\leftarrow X_0 \\
n &\leftarrow 0 \\
k &\leftarrow 0
\end{align*}
\]

**while** \( \|\nabla f(X_n)\|_g^* \geq \epsilon \) **do**

\[
\begin{align*}
X_{n+1} &\leftarrow R_1(Y_n, \phi_g(-\gamma_n \nabla f(Y_n))) \text{ evaluated using equation 2.26 with } \gamma_n \text{ chosen so that } f(X_{n+1}) \leq f(Y_n) - \frac{1}{2} \gamma_n \|\nabla f(Y_n)\|_g^2 \text{ (Armijo condition) and } \gamma_n \leq \gamma_{n-1} \\
\text{if } f(X_{n+1}) > f(X_n) - c_R \gamma_n \|\nabla f(Y_n)\|_g^2 \text{ (Restart Condition) then} \\
X_{n+1} &\leftarrow X_n \\
Y_n &\leftarrow X_{n+1} \\
k &\leftarrow k + 1
\end{align*}
\]

**else**

\[
\begin{align*}
V_n &\leftarrow 2X_{n+1}(I + X_{n+1}^T X_n)^{-1} \\
Y_{n+1} &\leftarrow R_1(X_n, (1 + \frac{k}{k+3}) \phi_g(V_n)) \text{ (apply momentum)} \\
k &\leftarrow k + 1
\end{align*}
\]

**end**

\[
n \leftarrow n + 1
\]

**end**

---

**Table 4.1**

*Accelerated Gradient Descent with Function Restart Scheme*

This element \( V \) then serves as \( x_{k-1} - y_{k-1} \) and the analogue of the gradient restart condition becomes

\[
(4.17) \quad - \gamma_{k-1} \|\nabla f(y_{k-1})\|_g^2 - \langle \nabla f(y_{k-1}), V\rangle_g > 0
\]

As in the previous subsection, we see that equation 4.16 can be efficiently solved for \( V \) if the retraction we are using is \( R_1 \) (the Cayley retraction introduced in [15]).

**4.5. Accelerated Gradient Descent on the Stiefel Manifold.** We now put together all of the ideas presented in this section to obtain two versions of accelerated gradient descent on the Stiefel manifold; the function restart variant, presented in table 4.1, and the gradient restart scheme, presented in table 4.2. In the next section we will present numerical results which demonstrate empirically that our methods achieve the desired iteration complexity, i.e. that the number of iterations scales as \( O(\sqrt{\kappa}) \), where \( \kappa \) is the condition number of the objective function.
Algorithm 4.2 Accelerated Gradient Descent with Gradient Restart Scheme

**Data:** \( f \) a smooth function, \( \epsilon \) a tolerance

**Result:** A point \( X_n \) such that \( \| \nabla f(X_n) \|_g^* < \epsilon \)

\[
\begin{align*}
X_0 &\leftarrow \text{initial point} \\
Y_0 &\leftarrow X_0 \\
n &\leftarrow 0 \\
k &\leftarrow 0 \\
\text{while } &\| \nabla f(X_n) \|_g^* \geq \epsilon \text{ do} \\
&X_{n+1} \leftarrow R_1(Y_n, \phi_g(-\gamma_n \nabla f(Y_n))) \text{ evaluated using equation 2.26 with } \gamma_n \text{ chosen} \\
&\text{so that } f(X_{n+1}) \leq f(Y_n) - \frac{1}{2} \gamma_n \| \nabla f(Y_n) \|_g^*^2 \text{ (Armijo condition) and } \gamma_n \leq \gamma_{n-1} \\
&W_n \leftarrow 2X_n(I + X_n^TY_n)^{-1} \\
&\text{if } \langle \nabla f(Y_n), W_n \rangle_g < -\gamma_n \| \nabla f(Y_n) \|_g^*^2 \text{ (Restart Condition) then} \\
&X_{n+1} \leftarrow X_n \\
&Y_n \leftarrow X_{n+1} \\
&k \leftarrow 0 \\
&\text{else} \\
&V_n \leftarrow 2X_{n+1}(I + X_{n+1}^TX_{n+1})^{-1} \\
&Y_{n+1} \leftarrow R_1(X_n, (1 + \frac{k}{k+3})\phi_g(V_n)) \text{ (apply momentum)} \\
&k \leftarrow k + 1 \\
&\text{end} \\
n &\leftarrow n + 1 \\
\end{align*}
\]

5. Numerical Results. In this section, we provide the results of numerical experiments which test the convergence properties and robustness of the two algorithms described in the previous section. We test the algorithms on a sequence of eigenvector calculations with increasing condition numbers. This allows us to investigate how the iteration count scales with the condition number of the problem. The reason we do eigenvector calculations is that the condition number of the corresponding objectives can be evaluated with relative ease.

5.1. Single Eigenvector Calculations. We begin by testing our algorithms on the sphere (which is a special case of the Stiefel manifold \( S_{n,k} \) with \( k = 1 \)). The problem we solve is the eigenvector calculation

\[
(5.1) \quad \arg\min_{X \in S^n} \frac{1}{2} X^TAX
\]

where \( A \) is a symmetric matrix. The solution to this problem is the eigenvector corresponding to the smallest eigenvalue of \( A \).

In order to evaluate the performance of our algorithm, we must investigate how the number of iterations scales with the condition number of 5.1 (not to be confused with the condition number of \( A \)). A trivial calculation shows that this condition number is given by
\begin{equation}
\kappa(H f(v^*)) = \frac{\lambda_n - \lambda_1}{\lambda_2 - \lambda_1}
\end{equation}

where \(\lambda_1, \ldots, \lambda_n\) are the eigenvalues of \(A\).

In Figure 5.1 we present the results of applying our algorithms to the above problem with \(A_n = \text{diag}(0, 1, \ldots, n)\) on \(S^{n+1}\). By the above calculation, the condition number of this problem is \(n\). We initialize \(X_0\) at a uniformly random point on the sphere and plot the number of iterations (with the tolerance \(\epsilon = 1e^{-3}\)) vs the condition number, for \(n = 100, (1.5 \cdot 100), (1.5^2 \cdot 100), \ldots, (1.5^{20}) \cdot 100\). To reduce the random fluctuations, we solve each problem 10 times (with different random starting points) and plot the average number of iterations against the condition number.

We see that our method empirically achieves the desired convergence behavior. Indeed, we plot a log linear fit whose coefficient is slightly larger than .5 in both cases. This indicates that the method achieves the desired scaling with respect to condition number.

### 5.2. Multiple Eigenvector Calculations

We now test our algorithms on the Stiefel manifold \(S_{n,k}\) with \(k > 1\). The problem we consider is that of calculating the smallest \(k\) eigenvectors of a symmetric linear operator \(A\) by minimizing the Brockett cost

\begin{equation}
\arg\min_{X \in S_{n,k}} \frac{1}{2} \sum_{i=1}^{k} \alpha_k (X_k, AX_k)
\end{equation}

where \(X_k\) denotes the \(k\)-th column of \(X\) and \(0 < \alpha_1 < \alpha_2 < \ldots < \alpha_k\) are coefficients which force the minimizer to consist of eigenvectors of \(A\) rather than eigenvectors up to an orthogonal transformation.

As before, we want to investigate how the number of iterations depends upon the condition number of 5.3. A simple calculation shows that the condition number of the Brockett cost 5.3 is

\begin{equation}
\kappa(H f(X^*)) = \frac{\alpha_k (\lambda_n - \lambda_1)}{\min\{\alpha_1 (\lambda_{k+1} - \lambda_k), \min_{i<k} (\lambda_{i+1} - \lambda_i)(\alpha_{i+1} - \alpha_i)\}}
\end{equation}

We briefly note that if one knew the eigenvalues \(\lambda_1, \ldots, \lambda_n\), then an optimal choice of the coefficients \(\alpha_1, \ldots, \alpha_n\) produces a condition number of

\begin{equation}
\kappa(H f(X^*))_{opt} = (\lambda_n - \lambda_1) \left( \sum_{i=1}^{k} \frac{1}{\lambda_{i+1} - \lambda_i} \right)
\end{equation}

We now present our numerical results. As in the single eigenvector calculations, we let \(A_n = \text{diag}(0, \ldots, n)\) and calculate the first \(k\) eigenvectors by optimizing over \(S_{n+1,k}\). Setting the weights \(\alpha_i = i\) produces the optimal condition number of \(kn\) in 5.5. We set \(k = 10\), initialize \(X_0\) at a uniformly random point on \(S_{n+1,10}\) and plot the number of iterations (with the tolerance \(\epsilon = 1e^{-3}\)) vs the condition number, for \(n = 100, (1.5 \cdot 100), (1.5^2 \cdot 100), \ldots, (1.5^{20}) \cdot 100\). To reduce the random fluctuations, we solve each problem 10 times (with different random starting points) and plot the average number of iterations against the condition number in Figure 5.2.
We again see that our method empirically achieves the desired convergence behavior. Indeed, we plot a log linear fit whose coefficient is slightly larger than 0.5 in both cases, similar to what was observed on the sphere. This indicates that the method also works well when optimizing over a larger set of orthonormal vectors.

6. Comparison with Existing State-of-the-Art Methods. In this section, we compare our algorithm with existing state of the art methods. We only consider our function restart scheme since it performed better than the gradient restart scheme in our previous tests. We compare with the state of the art quasi-newton method implemented in the ROTLIB library [5] (see also [4]).

We run both methods against each other on an ill-conditioned Brockett cost function, given by

\[ \arg \min_{X \in S_{n,k}} \sum_{i=1}^{k} \alpha_{k} \langle X_{k}, AX_{k} \rangle \]

where the spectrum of the \( n \times n \) matrix \( A \) is given by \( \{ \frac{k^{2}}{n} \} \) and the weights are \( \alpha_{k} = k \). In order to obtain reliable results, we test both methods with the same objective and the same random initial points. We repeat each test 10 times and record the average number of iterations, number of gradient evaluations, number of function evaluations, and total computational time. The same stopping condition is used for both methods, namely we stop when the gradient norm is less than 0.001.

In our tests we consider two different values of \( n \) and \( k \). First, we test with \( n = 1000, \ k = 10 \), whose results are shown in table 6.1. Then we test with the larger values \( n = 2000, \ k = 20 \). The results of this test are shown in table 6.2. We see that our method outperforms the method in [4], using about half as many gradient evaluations, about the same number of functions evaluations, and about half the time as the quasi-newton method.

| Method    | Iterations | Gradient Evals | Function Evals | Time (s) |
|-----------|------------|----------------|----------------|----------|
| Quasi-Newton | 18,023     | 18,024         | 18,884         | 15       |
| Our Method   | 12,045     | 12,046         | 24,150         | 8        |

*Table 6.1*  
Comparison of the methods on a problem of size \( n = 1000, \ k = 10 \).

| Method    | Iterations | Gradient Evals | Function Evals | Time (s) |
|-----------|------------|----------------|----------------|----------|
| Quasi-Newton | 47,057     | 47,058         | 49,372         | 91       |
| Our Method   | 28,023     | 28,024         | 56,102         | 46       |

*Table 6.2*  
Comparison of the methods on a problem of size \( n = 2000, \ k = 20 \).

7. Conclusion. In this paper, we developed novel accelerated first-order optimization methods designed to handle orthogonality constraints. The algorithms developed are a generalization of Nesterov’s gradient descent to the Stiefel manifold. In the process, we constructed an efficient way of averaging and extrapolating points on the manifold, which we believe can be useful in developing other novel optimization algorithms. Numerical experiments indicate that our methods not only achieve the
desired scaling with the condition number of the problem, but also outperform state of the art methods on some large, ill-conditioned problems.

We would also like to note that if the objective has a group of symmetries, then our algorithm behaves as if it were running on the quotient of $S_{n,k}$ by this symmetry group. Thus we recover linear convergence (even though the objective is not strongly convex if the symmetry group is continuous) and the important quantity is the condition number of the objective as a function on this quotient manifold. This means, for instance, that our algorithm can also be used to optimize over the Grassmann manifold.

8. Acknowledgements. We are grateful to Russel Caflisch, Stanley Osher, and Vidvuds Ozolins for suggesting this project and providing continued guidance. We would also like to thank Jinchao Xu for helpful guidance. This work was supported by AFOSR grant FA9550-15-1-0073.

References.
[1] P-A Absil, Robert Mahony, and Rodolphe Sepulchre. Optimization algorithms on matrix manifolds. Princeton University Press, 2009.
[2] Sébastien Bubeck et al. “Convex optimization: Algorithms and complexity”. In: Foundations and Trends® in Machine Learning 8.3-4 (2015), pp. 231–357.
[3] Alan Edelman, Tomás A Arias, and Steven T Smith. “The geometry of algorithms with orthogonality constraints”. In: SIAM journal on Matrix Analysis and Applications 20.2 (1998), pp. 303–353.
[4] Wen Huang, Kyle A Gallivan, and P-A Absil. “A Broyden class of quasi-Newton methods for Riemannian optimization”. In: SIAM Journal on Optimization 25.3 (2015), pp. 1660–1685.
[5] Wen Huang et al. “ROPTLIB: an object-oriented C++ library for optimization on Riemannian manifolds”. In: ACM Transactions on Mathematical Software (TOMS) 44.4 (2018), p. 43.
[6] Qihang Lin and Lin Xiao. “An adaptive accelerated proximal gradient method and its homotopy continuation for sparse optimization”. In: International Conference on Machine Learning. 2014, pp. 73–81.
[7] Yuanyuan Liu et al. “Accelerated first-order methods for geodesically convex optimization on riemannian manifolds”. In: Advances in Neural Information Processing Systems. 2017, pp. 4868–4877.
[8] Richard M Martin. Electronic structure: basic theory and practical methods. Cambridge university press, 2004.
[9] Yurii Nesterov. “A method of solving a convex programming problem with convergence rate $O(1/k^2)$”. In: Soviet Mathematics Doklady. Vol. 27. 2. 1983, pp. 372–376.
[10] Yurii Nesterov et al. Gradient methods for minimizing composite objective function. 2007.
[11] Brendan O’donoghue and Emmanuel Candes. “Adaptive restart for accelerated gradient schemes”. In: Foundations of computational mathematics 15.3 (2015), pp. 715–732.
[12] Vidvuds Ozoliņš et al. “Compressed modes for variational problems in mathematics and physics”. In: Proceedings of the National Academy of Sciences 110.46 (2013), pp. 18368–18373.
[13] Jack Sherman and Winifred J Morrison. “Adjustment of an inverse matrix corresponding to a change in one element of a given matrix”. In: The Annals of Mathematical Statistics 21.1 (1950), pp. 124–127.
[14] Weijie Su, Stephen Boyd, and Emmanuel J Candes. “A differential equation for modeling Nesterov’s accelerated gradient method: theory and insights”. In: *Journal of Machine Learning Research* 17.153 (2016), pp. 1–43.

[15] Zaiwen Wen and Wotao Yin. “A feasible method for optimization with orthogonality constraints”. In: *Mathematical Programming* 142.1-2 (2013), pp. 397–434.

[16] Hongyi Zhang and Suvrit Sra. “Towards Riemannian Accelerated Gradient Methods”. In: *arXiv preprint arXiv:1806.02812* (2018).

[17] Xin Zhang et al. “Gradient type optimization methods for electronic structure calculations”. In: *SIAM Journal on Scientific Computing* 36.3 (2014), pp. C265–C289.

[18] Xiaoqing Zhu. “A Riemannian conjugate gradient method for optimization on the Stiefel manifold”. In: *Computational Optimization and Applications* 67.1 (2017), pp. 73–110.
Fig. 5.1. Iteration Count vs Condition Number (Sphere, larger range of condition numbers), Function Restart (top), Gradient Restart (bottom)
Fig. 5.2. Iteration Count vs Condition Number (k = 10, larger range of condition numbers),
Function Restart (top), Gradient Restart (bottom)