Disciplined Geodesically Convex Programming

Andrew N. Cheng* ANDREWCHENG@G.HARVARD.EDU
Harvard University
Cambridge, MA 02138, USA

Vaibhav Dixit* VKDIXIT@MIT.EDU
CSAIL, MIT
Cambridge, MA 02139, USA

Melanie Weber MWEBER@SEAS.HARVARD.EDU
Harvard University
Cambridge, MA 02138, USA

Abstract

Convex programming plays a fundamental role in machine learning, data science, and engineering. Testing convexity structure in nonlinear programs relies on verifying the convexity of objectives and constraints. Grant et al. (2006) introduced a framework, *Disciplined Convex Programming* (DCP), for automating this verification task for a wide range of convex functions that can be decomposed into basic convex functions (atoms) using convexity-preserving compositions and transformations (rules). However, the restriction to Euclidean convexity concepts can limit the applicability of the framework. For instance, many notable instances of statistical estimators and matrix-valued (sub)routines in machine learning applications are Euclidean non-convex, but exhibit *geodesic* convexity through a more general Riemannian lens. In this work, we extend disciplined programming to this setting by introducing *Disciplined Geodesically Convex Programming* (DGCP). We determine convexity-preserving compositions and transformations for geodesically convex functions on general Cartan-Hadamard manifolds, as well as for the special case of symmetric positive definite matrices, a common setting in matrix-valued optimization. For the latter, we also define a basic set of atoms. Our paper is accompanied by a Julia package *SymbolicAnalysis.jl*, which provides functionality for testing and certifying DGCP-compliant expressions. Our library interfaces with manifold optimization software, which allows for directly solving verified geodesically convex programs.

Keywords: Riemannian Optimization, Disciplined Convex Programming, Geodesic Convexity

*. Equal contribution. Co-first authors listed alphabetical.

1

1 Introduction

Nonlinear programming, which involves optimization tasks with nonlinear objectives and/or nonlinear constraints, plays a fundamental role in data science, machine learning, engineering, operations research, and economics. Classically, nonlinear programs are solved with Euclidean optimization methods, whose design and mathematical analysis has been the subject of decades of research. Structured nonlinear programs can often be solved more efficiently with specialized methods. This has given rise to a wide range of algorithms for solving special classes of nonlinear programs that leverage special structure in the programs's objective and constraints. *Convex programming* involves nonlinear programs with Euclidean convex objectives and constraints, which gives rise to efficient algorithms with global optimality certificates. While convex programming has a wide range of applications, there are many notable instances in data science and machine learning that do not fit into this restrictive setting. This includes the computation of several important statistical estimators, such as Tyler's and related M-estimators (Tyler, 1987; Wiesel, 2012; Ollila and Tyler, 2014), optimistic likelihood estimation (Nguyen et al., 2019), and certain Wasserstein bounds on entropy (Courtade et al., 2017). Furthermore, a number of matrix-valued (sub-) routines that arise in machine learning approaches fall into this setting, including robust subspace recovery (Zhang, 2016), matrix barycenter problems (Bhatia, 1997), and learning Determinantal Point Processes (DPPs) (Mariet and Sra, 2015). However, a closer analysis of the properties of these nonlinear programs can reveal "hidden" convexity structure, when viewed through a geometric lens: While their objectives and/or constraints may be Euclidean non-convex, they are convex with respect to a different Riemannian metric.

A notable setting where such convexity structure arises are optimization tasks on symmetric positive definite matrices. We can endow this space either with a Euclidean metric or with the affine-invariant Riemannian metric, in which case they form a Cartan-Hadamard manifold, i.e., a manifold of non-positive sectional curvature. The sample applications listed above exhibit convexity in this geometric setting. In practice, if we can reliably identify under which metric a given program exhibits such *geodesic convexity*, we can leverage efficient convex optimization tools with global optimality guarantees. This observation motivates the need for tools that can effectively test and verify the convexity of the objective and constraints of nonlinear programs under generalized metrics. While this can be done "by hand" via mathematical analysis, the development of computational tools that automate this procedure and that can be integrated into numerical software would ensure broad applicability. In the Euclidean setting, *Disciplined Convex Programming* (Grant et al., 2006) (short: *DCP*) has been introduced as a framework for automating the verification of convexity. It decomposes the objective function or a functional description of the constraints into basic functions that are known to be convex (so-called *atoms*) using convexity-preserving compositions and transformations (known as *rules*). The CVX library (Diamond and Boyd, 2016) implements this framework and provides an interface with numerical convex optimization tools. More recently, the DCP framework has been extended to log-log convex (Agrawal et al., 2019) and quasi-convex (Agrawal and Boyd, 2020) programs. However, to the best of our knowledge, no extensions of this framework to the geodesically convex setting have been considered.

In this work, we introduce a generalization of the DCP framework that leverages the intrinsic geometry of the manifold to test convexity. The extension to the *geodesically convex* setting encompasses Euclidean convex programming, as well as programs with objectives and constraints that are convex with respect to more general Riemannian metrics (*Disciplined Geodesically Convex Programming*, short: *DGCP*). We provide a structured overview of geodesic convexity-preserving compositions and transformations of functions defined on Cartan-Hadamard manifolds, which serve as a foundational set of rules in our DGCP framework. Focusing on optimization tasks defined on symmetric positive definite matrices, we define additional rules, as well as a basic set of geodesically convex atoms that allow for testing and certifying the convexity of many classical matrix-valued optimization tasks. This includes in particular statistical estimators and many of the aforementioned subroutines in machine learning and data analysis methods. We further present an accompanying open-source package, *SymbolicAnalysis.jl* *, which implements DGCP, and illustrate its usage on several classical examples.

Related Work. Convex programming has been a major area of applied mathematics research for many decades (Boyd and Vandenberghe, 2004). Extensions of classical convex optimization algorithms to manifold-valued tasks have been studied extensively, resulting in generalized algorithms for convex (Udriste, 1994; Bacák, 2014; Zhang and Sra, 2016), nonconvex (Boumal et al., 2019), stochastic (Bonnabel, 2013; Zhang et al., 2016; Weber and Sra, 2021), constrained (Weber and Sra, 2022b, 2021; Bergmann and Herzog, 2019; Bergmann et al., 2022), and min-max optimization problems (Martínez-Rubio et al., 2023; Jordan et al., 2022), among others. Numerical software for solving geometric optimization problems has been developed in several languages (Boumal et al., 2014; Townsend et al., 2016; Bergmann, 2022; Huang et al., 2016). Disciplined Convex Programming for testing and certifying the Euclidean convexity of nonlinear programs has been developed by Grant et al. (2006) and made available in the CVX library (Diamond and Boyd, 2016). More recently, extensions to quasi-convex programs (*Disciplined Quasi-Convex Programming* (Agrawal and Boyd, 2020)) and log-log convex programs (*Disciplined Geometric Programming* (Agrawal et al., 2019)) have been integrated into CVX. We note that, in the latter, the term "geometric" is used in a different context than in our work: Log-log convexity is a Euclidean concept that evaluates convexity under a specific transformation. In contrast, the notion of geodesic convexity considers the geometry of the domain explicitly. To the best of our knowledge, no extensions of disciplined programming to the geodesically convex setting have been introduced in the prior literature.

*. `https://github.com/Vaibhavdixit02/SymbolicAnalysis.jl`

Summary of contributions. The main contributions of this work are as follows:

1. We introduce *Disciplined Geodesically Convex Programming*, a generalization of the Disciplined Convex Programming framework, which allows for testing and certifying the geodesic convexity of nonlinear programs on geometric domains.
2. Following an analysis of the algebraic structure of geodesically convex funcations, we define convexity-preserving compositions and transformations for geodesically convex functions on Cartan-Hadamard manifolds, as well as for the special case of symmetric positive definite matrices, for which we also define a foundational set of atoms.
3. For the special case of symmetric positive definite matrices, we present an implementation of this framework in the Julia language (Bezanson et al., 2017). Our open-source package, *SymbolicAnalysis.jl* allows for verifying DGCP-compliant convexity structure and interfaces with manifold optimization software, which allows for directly solving verified programs.

2 Background and Notation

In this section, we introduce notation and review standard notions of Riemannian geometry and optimization. For a comprehensive overview see (Boumal, 2023; Bacák, 2014).

2.1 Riemannian Geometry

A *manifold* $\mathcal{M}$ is a topological space that has a local Euclidean structure. Every $x \in \mathcal{M}$ has an associated *tangent space* $\mathcal{T}_x\mathcal{M}$, which consists of the tangent vectors of $\mathcal{M}$ at x. We restrict our attention to *Riemannian manifolds*, which are endowed with a smoothly varying inner product $\langle u, v\rangle_x$ defined on $\mathcal{T}_x\mathcal{M}$ for each $x \in \mathcal{M}$.

A special instance considered in this paper is the manifold of symmetric positive definite matrices, denoted as $\mathbb{P}_d$, which we encounter frequently in matrix-valued optimization. Formally, it is given by the set of $d \times d$ real symmetric square matrices with strictly positive eigenvalues, i.e.,

$$\mathbb{P}_d := \{X \in \mathbb{R}^{d\times d} : X^T = X,\ X \succ 0\}$$

Endowing $\mathbb{P}_d$ with different inner product structures gives rise to different Riemannian lenses on $\mathbb{P}_d$. We recover a Euclidean structure if we endow $\mathbb{P}_d$ with the trace metric given by

$$\langle A, B\rangle = \operatorname{tr}(A^\top B) \qquad \forall A, B \in \mathbb{P}_d.$$

We can induce a *non-flat* Riemannian structure of $\mathbb{P}_d$ by endowing $\mathbb{P}_d$ with the canonical *affine invariant* inner product,

$$\langle A, B\rangle_X = \operatorname{tr}\left(X^{-1}AX^{-1}B\right) \quad X \in \mathbb{P}_d, A, B \in T_X\left(\mathbb{P}_d\right) = \mathbb{H}_d,$$

where the tangent space $\mathbb{H}_d$ is the space of $d \times d$ real symmetric matrices. The resulting Riemannian manifold $\mathbb{P}_d$ is a *Cartan-Hadamard manifold*, i.e., a manifold of non-positive sectional curvature. Importantly, the class of Cartan-Hadamard manifolds is particularly appealing for optimization due to properties such as *unique geodesics* and amenablility to *geodesic convexity analysis* (Bacák, 2014). Recall that *geodesics* are parametric curves that define shortest-paths along the Cartan-Hadamard manifold. Returning to $\mathbb{P}_d$, given any matrices $A, B \in \mathbb{P}_d$, the unique geodesic connecting A to B has the explicit parametrization

$$\gamma(t) = A^{1/2}\left(A^{-1/2}BA^{-1/2}\right)^t A^{1/2}, \quad 0 \leq t \leq 1 \ . \tag{1}$$

The affine-invariant structure on $\mathbb{P}_d$ gives rise to the following *Riemannian distance* on $\mathbb{P}_d$,

$$\delta_2(A, B) = \left\|\log A^{-1/2}BA^{-1/2}\right\|_2 \ ,$$

which corresponds to the length of the geodesic connecting A and B. It is geodesically convex, since $\mathbb{P}_d$ is Cartan-Hadamard (Bacák, 2014; Bhatia, 2007).

2.2 Geodesic Convexity of Functions and Sets

Cartan-Hadamard manifolds allows us to import, to a large extent, classical results from (Euclidean) convex analysis, as well as convex optimization tools to the Riemannian (i.e., geodesically convex) setting. We introduce the analogous notions of convexity of sets and functions in the Riemannian setting. In this paper, we only consider functions that are continuous.

Suppose $\mathcal{M}$ is a Riemannian manifold.

Definition 1 (Geodesic convexity of Sets) *A set $S \subseteq \mathcal{M}$ is* geodesically convex *(short: g-convex) if for any two points $x, y \in \mathcal{M}$, there exists a geodesic $\gamma : [0,1] \to \mathcal{M}$ such that $\gamma(0) = x$ and $\gamma(1) = y$ and the image satisfies $\gamma([0,1]) \subseteq S$.*[*]

Definition 2 (Geodesic convexity of Functions) *We say that $\phi : S \to \mathbb{R}$ is a* geodesically convex function *(short: g-convex) if $S \subseteq \mathcal{M}$ is geodesically convex and $f \circ \gamma : [0,1] \to \mathbb{R}$ is (Euclidean) convex for each geodesic segment $\gamma : [0,1] \to \mathbb{P}_d$ whose image is in S with $\gamma(0) \neq \gamma(1)$.*

As we will see in Section 3.2.2, many of the operations that preserve Euclidean convexity extend to the geodesically convex setting. In Appendix B, we illustrate how the convexity of functions depends naturally on the geometry of the Riemannian manifold.

*. For geodesically convex sets on Cartan-Hadamard manifolds, any such geodesic segment is unique.

2.3 Riemannian optimization software

A widely used library for manifold optimization is the *Manopt* toolbox (Boumal et al., 2014), a MATLAB-based software designed to facilitate the experimentation with and application of Riemannian optimization algorithms. *Manopt* simplifies handling complex optimization tasks by providing user-friendly and well-documented implementations of various state-of-the-art algorithms. It separates the manifolds, solvers, and problem descriptions, allowing easy experimentation with different combinations. In addition to the MATLAB version, a Python implementation has been made available (*PyManopt* (Townsend et al., 2016)).

In the Julia programming language, *Manopt.jl* (Bergmann, 2022) offers a comprehensive framework for optimization on Riemannian manifolds. It utilizes *Manifolds.jl* (Axen et al., 2023) for efficient implementations of manifolds like the Euclidean, hyperbolic, and spherical spaces, the Stiefel manifold, the Grassmannian, and the positive definite matrices, among others. It further includes an efficient implementation of important primitives on these manifolds like geodesics, exponential and logarithmic maps, parallel transport, etc. *Manopt.jl* supports a wider range of algorithms than *Manopt* and *PyManopt*, including classical gradient-based methods, quasi-Newton methods like Riemannian L-BFGS, and several nonsmooth optimization techniques. Additionally, there are other software packages such as *ROPTLIB* for C++ (Huang et al., 2016), which provide similar functionalities for manifold optimization.

3 Disciplined Geodesically Convex Programming

In this section we introduce the *Disciplined Geodesically Convex Programming* framework (short: *DGCP*). We discuss the relationship to other classes of convex programming, as well as the essential building blocks of the framework.

3.1 Taxonomy of Convex Programming

We consider *nonlinear programs* (NLP) of the form

$$\begin{aligned} \min_x \quad & f(x) \\ \text{subject to} \quad & g_i(x) \leq 0,\ i = 1, \ldots, m \\ & h_j(x) = 0,\ j = 1, \ldots, n\ , \end{aligned} \tag{2}$$

which are defined by an objective function $f : \mathbb{R}^{n\times n} \to \mathbb{R}$ and a set of inequality $\{g_i\}_{i\in[m]}$ and equality constraints $\{h_j\}_{j\in[n]}$.

Convex Programming. *Convex programs* (CP) are a class of NLPs, in which both the objective and the constraints are convex. Classically, "convexity" refers to Euclidean convexity. Here, we consider the more general class of *geodesically convex programs* (GCP),

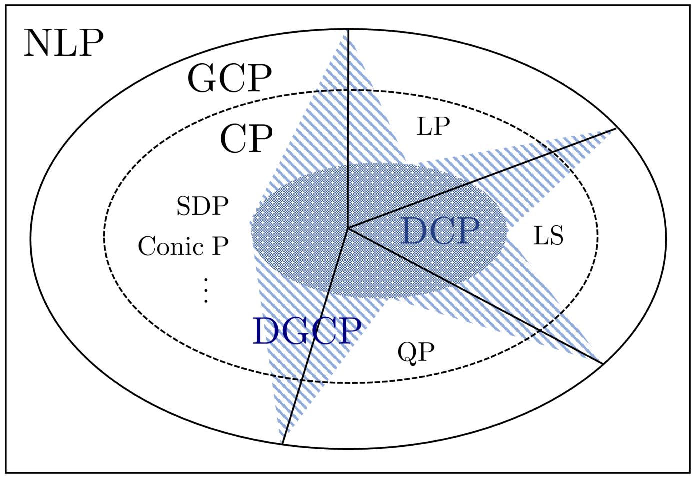

Figure 1: **Taxonomy of Convex Programming.** The diagram shows the relationship of GCP, CP and their subclasses (e.g., SDP, LP, QS etc.). DGCP (blue shaded) has non-empty intersections with GCP, CP and their subclasses and contains DCP (gray shaded) as a special case.

which require that the objective and constraints are geodesically convex under some Riemannian metric, but not necessary the Euclidean metric. Naturally, CP $\subset$ GCP. CP encompasses Linear Programming (LP), Quadratic Programming (QP), Least Squares (LS) problems, as well as a number of optimization problems with special structure, such as semidefinite programs (SDP) and conic programs (Conic P).

From an algorithmic perspective, (geodesic) convexity enables certificates of *global optimality*, in that local optima are guaranteed to be global optima. Since local optimality can be verified, e.g., via KKT conditions, this allows for global convergence guarantees from any initialization in practice – a highly desirable property. Hence, CP and its subclasses have been extensively studied in the Euclidean optimization literature. More recently, GCP (Udriste, 1994; Bacák, 2014; Zhang et al., 2016; Weber and Sra, 2022b, 2023), as well as generalizations of the CP subclasses to the geodesic setting have been studied (Sra and Hosseini, 2015).

Disciplined Programming. Due to the algorithmic benefits discussed above, identifying and verifying CP is of great interest in practise. Aside from formally proving convexity certificates (i.e., verifying Def. 1 for objective functions and Def. 2 for the feasible region), one can also leverage the algebraic structure of convex functions to discover convexity in objectives and constraints. Specifically, many transformations or compositions of convex functions yield convex functions. The idea of *Disciplined Convex Programming* (DCP) (Grant et al., 2006) is to define a set of *atoms* and *rules* to verify convexity properties. Atoms are functions and sets whose properties in terms of convexity and monotonicity are known. Rules encode fundamental principles from convex analysis on transformations and compositions that preserve or induce convexity in functions or sets. Together, they form a modular framework for verifying convexity in functions and sets that can be decomposed into atoms using any combination of rules. In principle, any function that is not verifiable using existing atoms and rules could be added as a new atom, which would allow for creating a library of rules and atoms that could verify the convexity of any CP. However, in practise, DCP libraries are limited to a set of core atoms and rules that allow for verifying commonly encountered mathematical programs. Hence, generally DCP $\subset$ CP.

In this work, we extend the idea of disciplined programming to the geodesically convex setting. We design a library of geodesically convex atoms (sec. 3.3) and rules for preserving or inducing geodesic convexity in functions and sets (sec. 3.2.2). The resulting library, termed *Disciplined Geodesically Convex Programming* (DGCP) allows for verifying a larger subset of CP, as well as a subset of programs that are in GCP, but not in CP. Thus DGCP $\subset$ GCP. A schematic overview of the taxonomy of the different classes of convex programs can be found in Figure 1.

3.2 Rules

In this section, we present operations and transformations that are *DGCP-compliant*, i.e., that preserve geodesic convexity when applied to atoms. We first introduce a set of DGCP-compliant rules for functions on general Cartan-Hadamard manifolds, before focusing on the special case of the manifold of symmetric positive definite manifold, for which we provide an additional set of DGCP-compliant rules that are specific to this geometry.

3.2.1 Cartan-Hadamard manifolds

Recall that Cartan-Hadamard manifolds are manifolds of non-positive sectional curvature with the property that every pair of points can be connected by a unique geodesic that is distance-minimizing with respect to its Riemannian metric. This is a key property in generalizing tools from Euclidean convex analysis to the Riemannian setting (e.g., geodesic convexity) in a global sense. In contrast, such tools cannot be as readily imported to manifolds with positive sectional curvatures. For example, spheres do not admit globally geodesically convex functions beyond the constant function and key operations such as intersections of sets fail to preserve geodesic convexity on spheres. For these reasons and due to their wide range of applications, we focus on developing a disciplined programming framework for Cartan-Hadamard manifolds.

We begin with several elementary DGCP-compliant rules. We defer all proofs to Appendix A.1.

Proposition 1 *Let $(\mathcal{M}, d)$ be a Cartan-Hadamard manifold. Suppose $S \subseteq \mathcal{M}$ is a g-convex subset. Furthermore, suppose $f_i : S \to \mathbb{R}$ are g-convex for $i = 1, \ldots, n$. Then the following functions are also g-convex.*

1. $X \mapsto \max_{i \in \{1,\ldots,n\}} f_i(X)$
2. $X \mapsto \sum_{i=1}^{n} \alpha_i f_i(X)$ *for* $\alpha_1, \ldots, \alpha_n \geq 0$.

Remark 3 *In the setting of Cartan-Hadamard manifolds, property 1 of Proposition 1 can be generalized to an arbitrary collection of g-convex sets. That is, for an arbitrary collection of g-convex functions $\{f_i\}_{i \in \mathcal{I}}$, indexed by $\mathcal{I}$, the map $X \mapsto \sup_{i \in \mathcal{I}} f_i(X)$ is g-convex. This follows from the fact that a function f is g-convex if and only if its epigraph is g-convex (Bacák, 2014) and the fact that the epigraph of the supremum of a collection of functions is the intersection of the epigraphs of each function in such a collection. Finally, the intersection of g-convex sets is g-convex for Cartan-Hadamard manifolds (see, e.g., (Boumal, 2023)). Moreover, property 2 of Proposition 1 can easily be generalized to a countable conic sum of g-convex functions.*

The following rule gives a convexity guarantee for compositions of Euclidean and g-convex functions.

Proposition 2 *Let $(\mathcal{M}, d)$ be a Cartan-Hadamard manifold and $S \subset \mathcal{M}$ g-convex. Suppose $f : S \to \mathbb{R}$ is g-convex. If $h : \mathbb{R} \to \mathbb{R}$ is non-decreasing and Euclidean convex then $h \circ f : S \to \mathbb{R}$ is g-convex.*

We further introduce a set of of scalar composition rules for g-convex functions. Analogous results can be shown for concave and/or non-increasing functions.

Corollary 4 (Scalar Composition Rules)

1. *Let $f : S \to \mathbb{R}$ be geodesically concave. If $h : \mathbb{R} \to \mathbb{R}$ is non-increasing and convex, then $h \circ f$ is geodesically convex on S.*
2. *Let $f : S \to \mathbb{R}$ be geodesically concave. If $h : \mathbb{R} \to \mathbb{R}$ is non-decreasing and concave, then $h \circ f$ is geodesically concave on S.*
3. *Let $f : S \to \mathbb{R}$ be geodesically convex. If $h : \mathbb{R} \to \mathbb{R}$ is non-increasing and concave, then $h \circ f$ is geodesically convex on S.*

Example 1 *If $f : S \to \mathbb{R}$ is g-convex with respect to the canonical Riemannian metric then $\exp f(x)$ is g-convex and $-\log(-f(x))$ is g-convex on $\{x : f(x) < 0\}$. If f is non-negative and $p \geq 1$ then $f(x)^p$ is g-convex.*

3.2.2 Manifold of symmetric positive definite matrices

Our DGCP framework can be defined on any Cartan-Hadamard manifold. In addition to the general rules introduced in the previous section, additional sets of g-convexity preserving rules may be defined that arise from a manifold's specific geometry. We illustrate this for the special case of symmetric positive definite matrices, i.e., by setting $\mathcal{M} = \mathbb{P}_d$, endowed with the Euclidean and Riemannian metrics defined above.

Below we introduce a set of g-convexity preserving rules inherent to this particular geometry and provide several illustrative examples.

The Löwner order introduces a partial order relation on the symmetric positive definite matrices.

Definition 5 (Löwner Order) *For $A, B \in \mathbb{P}_d$ we write $A \succ B$ when $A - B \in \mathbb{P}_d$. Similarly, we write $A \succeq B$ whenever $A - B$ is symmetric positive semi-definite.*

We say a function f is *increasing* if $f(A) \succeq f(B)$ whenever $A \succeq B$.

Definition 6 (Positive Linear Map) *A linear map $\Phi : \mathbb{P}_d \to \mathbb{P}_m$ is positive when $A \succeq 0$ implies $\Phi(A) \succeq 0$ for all $A \in \mathbb{P}_m$. We say that Φ is strictly positive when $A \succ 0$ implies that $\Phi(A) \succ 0$.*

The following proposition gives a g-convexity guarantee for compositions of strictly positive linear maps.

Proposition 3 (Proposition 5.8 (Vishnoi, 2018)) *Let $\Phi(X)$ be a strictly positive linear operator from $\mathbb{P}_d$ to $\mathbb{P}_m$. Then $\Phi(X)$ is g-convex with respect to the Löwner order on $\mathbb{P}_m$ over $\mathbb{P}_d$ with respect to the canonical Riemannian inner product $g_X(U, V) := \operatorname{tr}\left[X^{-1}UX^{-1}V\right]$. In other words, for any geodesic $\gamma : [0, 1] \to \mathbb{P}_d$ we have that*

$$\Phi(\gamma(t)) \preceq (1 - t)\Phi(\gamma(0)) + t\Phi(\gamma(1)) \quad \forall t \in [0, 1] \ .$$

Consequently, the following maps are g-convex in this setting:

Example 2 (Strictly Positive Linear Operators) *Let $Y \in \mathbb{P}_d$ fixed. Applying Proposition 3 the following maps are g-convex w.r.t the canonical Riemannian metric on $\mathbb{P}_d$:*

1. $X \mapsto tr(X)$
2. $X \mapsto Y^\top XY$ *for* $Y \in \mathbb{R}^{d\times k}$

3. $X \mapsto \mathrm{Diag}(X) := \sum_j X_{jj} E_{jj}$, *where* E_{jj} *is the* $d \times d$ *matrix with 1 in the* (j,j)*-th element and 0 everywhere else.*

4. *Let* $M \succeq 0$ *and* M *has no zero rows. The function* $\Phi(X) = M \odot X$ *where* $\odot$ *denotes the Hadamard product is a strictly positive linear operator and hence g-convex.*

Moreover, the following proposition guarantees that the composition of positive linear maps with $\log\det(\cdot)$ is g-convex.

Proposition 4 (Proposition 5.9 (Vishnoi, 2018)) *Let* $\Phi(X) : \mathbb{P}_d \to \mathbb{P}_m$ *be a strictly positive linear operator. Then,* $\log\det(\Phi(X))$ *is g-convex on* $\mathbb{P}_d$ *with respect to the metric* $g_X(U,V) := \mathrm{tr}\left[X^{-1}UX^{-1}V\right]$.

Proposition 5 *Let* $f : \mathbb{P}_d \to \mathbb{R}$ *be g-convex. Then* $g(X) = f(X^{-1})$ *is also g-convex.*

Example 3 *Applying Proposition 4 and Lemma 5 the following maps are g-convex with respect to the canonical Riemannian metric.*

1. $X \mapsto \log\det\left(\frac{X+Y}{2}\right)$ *for fixed* $Y \in \mathbb{P}_d$
2. $X \mapsto \log\det\left(X^r Y\right)$ *for fixed* $Y \in \mathbb{P}_d$ *and* $r \in \{-1, 1\}$
3. $X \mapsto \log\det\left(\sum_{i=1}^n Y_i X^r Y_i^\top\right)$ *for* $\{Y_1, \ldots, Y_n\} \subseteq \mathbb{P}_d$ *and* $r \in \{-1, 1\}$.

Moreover, the following map can be seen as a special case of (3).

4. *Let* $y_i \in \mathbb{R}^d \setminus \{0\}$ *for* $i = 1, \ldots, m$. *The function*

$$X \mapsto \log\left(\sum_{i=1}^m y_i^\top X y_i\right)$$

is g-convex with respect to the canonical Riemannian metric.

We provide an additional proof that this function is g-convex in Appendix B.

Example 4 *The following maps are g-convex.*

1. $g(X) = \sum_{i=1}^k \lambda_i^\downarrow(X^{-1})$ *for* $k = 1, \ldots, d$.
2. $g(X) = \sum_{i=1}^k \log\left(\lambda_i^\downarrow(X^{-1})\right)$ *for* $k = 1, \ldots, d$.

3. $g(X) = \log\det\left(\frac{X^{-1}+Y}{2}\right)$ *for fixed* $Y \in \mathbb{P}_d$.

The following result generalizes Proposition 4 beyond the $\log\det(\cdot)$ function and also relaxes the strict positivity to positivity.

Proposition 6 (Theorem 15 (Sra and Hosseini, 2015)) *Let* $h : \mathbb{P}_d \to \mathbb{R}$ *be non-decreasing and g-convex. Let* $r \in \{-1, 1\}$ *and let* Φ *be a positive linear map. Then* $\phi(X) = h\left(\Phi(X^r)\right)$ *is g-convex with respect to the canonical Riemannian metric.*

Example 5 (Examples of Proposition 6) *Fix some* $Y \in \mathbb{P}_d$. *Then the following results following directly from Proposition 6.*

1. *Let* $h(X) = tr(X^\alpha)$ *for* $\alpha \geq 1$ *and* $\Phi(X) = \sum_i Y_i^\top X Y_i$ *then* $X \mapsto tr\left(\sum_i Y_i^\top X^r Y_i\right)^\alpha$ *is g-convex.*
2. *Let* $h(X) = \log\det(X)$ *and* $\Phi(X) = \sum_i Y_i^\top X Y_i$ *then* $X \to \log\det\left(\sum_i Y_i^\top X Y_i\right)$ *is g-convex.*
3. *Let* $M \succeq 0$. *Let* $h(X) = \log\det(X)$ *and* $\Phi(X) = X \odot M$ *then* $X \mapsto \log\det\left(X \odot M\right)$ *is g-convex.*

We can extend the previous proposition to *positive affine operators* which we now define.

Definition 7 (Positive Affine Operator) *Let* $B \succeq 0$ *be a fixed symmetric positive semidefinite matrix and* $\Phi : \mathbb{P}_d \to \mathbb{P}_d$ *be a positive linear operator. Then the function* $\phi : \mathbb{P}_d \to \mathbb{P}_d$ *defined by*

$$\phi(X) \stackrel{def}{=} \Phi(X) + B$$

is an positive affine operator.

Proposition 7 (Geodesic Convexity of Positive Affine Maps) *Let* $\phi(X) \stackrel{def}{=} \Phi(X) + B$ *where* $\Phi(X)$ *is a positive linear map and* $B \succeq 0$. *Let* $f : \mathbb{P}_d \to \mathbb{P}_m$ *be g-convex and monotonically increasing, i.e.,* $f(X) \preceq f(Y)$ *whenever* $X \preceq Y$. *Then the function* $g(X) \stackrel{def}{=} f\left(\phi(X)\right)$ *is g-convex.*

Example 6 *Let* $B \succeq 0$ *and* $Y_i \in \mathbb{P}_d$ *for* $i = 1, \ldots, n$ *be fixed matrices.*

1. $X \mapsto tr\left(B + \sum_i Y_i^\top X^r Y_i\right)^\alpha$ *is g-convex.*
2. $X \mapsto \log\det\left(B + \sum_i Y_i^\top X Y_i\right)$ *is g-convex.*

3. *Let $M \succeq 0$. The map $X \mapsto \log\det(B + X \odot M)$ is g-convex.*

The following result provides a means for constructing geodesically convex *logarithmic tracial* functions.

Theorem 8 (Theorem 17 (Sra and Hosseini, 2015)) *If $f : \mathbb{R} \to \mathbb{R}$ is Euclidean convex, then the function $\phi(X) = \sum_{i=1}^{k} f\left(\log \lambda_i^{\downarrow}(X)\right)$ is g-convex for each $1 \le k \le d$ where $\lambda_i^{\downarrow}(X)$ denotes the ordered spectrum of X, i.e., $\lambda_1^{\downarrow}(X) \ge \lambda_2^{\downarrow}(X) \cdots \ge \lambda_d^{\downarrow}(X)$. Moreover, if $h : \mathbb{R} \to \mathbb{R}$ is non-decreasing and Euclidean convex, then $\phi(X) = \sum_{i=1}^{k} h(|\log \lambda_i^{\downarrow}(X)|)$ is g-convex for each $1 \le k \le n$.*

3.3 Atoms

Geodesically convex functions in DGCP are constructed via compositions and transformations of basic geodesically convex functions, so-called *atoms*. In this section, we provide a foundational set of geodesically convex functions defined on the manifold of symmetric positive definite matrices. Analogous sets of basic geodesically convex functions could be defined on other Cartan-Hadamard manifolds to extend the proposed framework to other settings.

In DGCP, the atoms are either g-convex or g-concave in their argument. Moreover, each atom has a designated curvature, either `GIncreasing` or `GDecreasing`. This monotonicity property relies on a partial order relation on the symmetric positive definite matrices, induced by the *Löwner order* (See Definition 5).

This motivates the following definition:

Definition 9 *A function $f : \mathbb{P}_d \to \mathbb{P}_d$ is `GIncreasing` if it satisfies $f(A) \succeq f(B)$ whenever $A \succeq B$.*

In the following, we list our basic set of DGCP atoms. We defer all proofs of g-convexity to Appendix A.2. We emphasize that our framework has a *modular* design, which allows for implementing additional atoms as needed.

3.3.1 Scalar-valued atoms

We begin with a set of *scalar-valued* DGCP atoms.

Log Determinant. `LinearAlgebra.logdet(X)` represents the log-determinant function $\log\det : \mathbb{P}_d \to \mathbb{R}_{++}$. This is an example of an atom that is `GLinear` (i.e. both g-convex and g-concave) and `GIncreasing`. It is concave in the Euclidean setting.

Trace. `LinearAlgebra.tr(X)` sums the diagonal entries of a matrix. It has `GConvex` curvature and is `GIncreasing`. It is affine in the Euclidean setting.

Sum of Entries. `sum(X)` will sum the entries of X, i.e., returns $\sum_{i,j=1}^{d} X_{ij}$. It has `GConvex` curvature and is `GIncreasing`. It is affine in the Euclidean setting.

S-Divergence. `sdivergence(X,Y)` is defined as

$$\texttt{sdivergence(X,Y)} := \log\det\left(\frac{X+Y}{2}\right) - \frac{1}{2}\log\det(XY).$$

This function is jointly geodesically convex, i.e., it is has `GConvex` curvature in both X and Y and is `GIncreasing`. It is non-convex in the Euclidean setting.

Riemannian Metric. `Manifolds.distance(X,Y)` returns the distance with respect to the *affine-invariant* metric.

$$\texttt{Manifolds.distance(X,Y)} := \left\|\log\left(Y^{-1/2}XY^{-1/2}\right)\right\|_F.$$

It is `GConvex` and is neither `GIncreasing` nor `GDecreasing` hence its monotonocity is unknown i.e. `GAnyMono`.

Quadratic Form. Fix $h \in \mathbb{R}^d$. The following function is g-convex `quad_form(h, X)` $= h^\top X h$ and `GIncreasing`. It is also convex in the Euclidean setting.

Spectral Radius. We define

$$\texttt{LinearAlgebra.eigmax(X)} := \sup_{\|y\|_2=1} y^\top X y\ ,$$

as the function that takes in $X \in \mathbb{P}_d$ and returns the maximum eigenvalue of X. This is a g-convex function and `GIncreasing`. It is also convex in the Euclidean setting.

Log Quadratic Form Let $h_i \in \mathbb{R}^d$ be nonzero vectors for $i = 1, \ldots, n$. Then

$$\texttt{log\_quad\_form(\{h\_1 ..., h\_n\}, X)} = \log\left(\sum_{i=1}^{n} h_i^\top X^r h_i\right) \qquad r \in \{-1, 1\}.$$

This is a g-convex function and `GIncreasing`. See Lemma 1.20 in (Wiesel and Zhang, 2015). It is non-convex in the Euclidean setting.

Definition 10 (Symmetric Gauge Functions) *A map $\Phi : \mathbb{R}^d \to \mathbb{R}_+$ is called a symmetric gauge function if*

1. Φ *is a norm.*

2. $\Phi(Px) = \Phi(x)$ *for all* $x \in \mathbb{R}^n$ *and all* $n \times n$ *permutation matrices* P*. This is known as the symmetric property.*

3. $\Phi(\alpha_1 x_1, \ldots, \alpha_n x_n) = \Phi(x_1, \ldots, x_n)$ *for all* $x \in \mathbb{R}^n$ *and* $\alpha_k \in \{\pm 1\}$*. This is known as the gauge invariant or absolute property.*

Proposition 8 (Symmetric Gauge Functions are g-convex (Cheng and Weber, 2024)) *Let* $\Phi : \mathbb{R}^d \to \mathbb{R}$ *be a symmetric gauge function. Then the function* $f(A) := \Phi(\lambda(A))$ *is geodesically convex where* $\lambda(A) = \{\lambda_1(A), \ldots, \lambda_d(A)\} \in \mathbb{R}^d$ *is the eigenspectrum of* A*.*

Remark 11 *For a symmetric gauge function* $\Phi : \mathbb{R}^d \to \mathbb{R}$ *and a matrix* $A \in \mathbb{P}_d$ *we use the notation* $\Phi(A)$ *to mean* $\Phi(\lambda(A))$*, i.e.* $\Phi(A)$ *acts on the eigenspectrum of* A*.*

Example 7 (Symmetric Gauge Functions) *The two canonical symmetric gauge functions are the Ky Fan and p-Schatten norm.*

1. *The* k-Ky Fan function *of* X *is the sum of the top* k *eigenvalues, i.e.,*

$$\Phi(X) = \sum_{i=1}^{k} \lambda_i^{\downarrow}(X) \qquad 1 \le k \le d$$

   where $\lambda_i^{\downarrow}(X)$ *is the sorted spectrum of* X*. The atom for* k-Ky Fan function *in our library is available as* `eigsummax(X, k)`*.*

2. *The* p-Schatten norm *for* $p \ge 1$ *is defined as*

$$\Phi(X) = \left(\sum_{i=1}^{d} \lambda_i^{p}(X) \right)^{\frac{1}{p}}$$

   The corresponding atom in our library is provided as `schatten_norm(X, p)`*.*

Example 8 *The following logarithmic symmetric gauge functions are g-convex by applying Theorem 8. They can be used with the* `sum_log_eigmax` *atom in our implementation.*

1. *Let* $f(t) = t$ *be the identity function in Theorem 8. Then*

$$\phi(X) = \sum_{i=1}^{k} \log \lambda_i^{\downarrow}(X) = \Phi(\log(X)) \qquad 1 \le k \le d$$

   is g-convex where $\Phi(\cdot)$ *is the k-Ky fan norm.*

2. Let $f(t) = t^p$ for $p \geq 1$ in Theorem 8. Then the function

$$\phi(X) = \sum_{i=1}^{k} \left(\log \lambda_i^{\downarrow}(X)\right)^p \qquad 1 \leq k \leq d$$

is g-convex.

Positive Affine Maps The results in 7 can be leveraged using the `affine_map` atom in our accompanying package.

3.3.2 Matrix-valued atoms

Our framework further incorporates a set of *matrix-valued DGCP atoms*, which are crucial for verifying the g-convexity of matrix-valued objectives and constraints.

Conjugation. Let $X \in \mathbb{P}_d$ and $A \in \mathbb{R}^{n\times n}$ then $\texttt{conjugation}(X, A) = A^\top X A$. This atom has `GConvex` curvature and is `GIncreasing`. It is Affine in the Euclidean setting.

Adjoint. Let $X \in \mathbb{P}_d$ then `adjoint(X)` $= X^\top$ has `GConvex` curvature and `GIncreasing`. It is Affine in the Euclidean setting.

Inverse. Let $X \in \mathbb{P}_d$ then `inv(X)` $= X^{-1}$ has `GConvex` curvature and `GDecreasing`. It is also Convex in the Euclidean setting.

Hadamard product. Let $X \in \mathbb{P}_d$ then `hadamard_product(X, B)` $= X \odot B$ has `GConvex` curvature and `GIncreasing`. It is affine in the Euclidean setting.

4 Implementation

The implementation of disciplined geodesically convex programming (DGCP) in this work is based on the foundation of symbolic computation and rewriting capability of the *Symbolics.jl* package (Gowda et al., 2022).

Each expression written with *Symbolics* is represented as a tree, where the nodes represent functions (or atoms), and the leaves represent variables or constants (see example in Figure 2). This representation enables the propagation of function properties, such as curvature and monotonicity, through the expression tree.

Previous implementations of disciplined programming, in CVXPY (Diamond and Boyd, 2016) and *Convex.jl* (Udell et al., 2014), define a class in the Object-Oriented Programming sense for each atom. We take a different approach in our DGCP implementation.

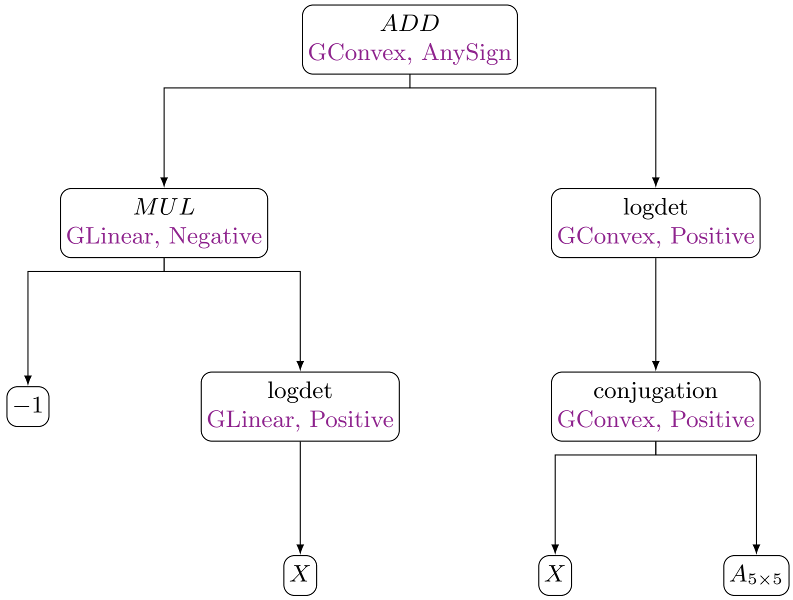

Figure 2: Expression tree for the problem of computing Brascamp-Lieb constants given in Eq. 7. The properties of the components are propagated up through the tree using the known properties of the atoms that make up the expression, giving the final geodesic curvature as `GConvex` and sign of the function as `AnySign`.

The relevant properties, such as domain, sign, curvature and monotonicity, are added as metadata to the leaves, and then propagated by looking up the corresponding property for every atomic function. The DGCP compliant rules are implemented using the rule-based term rewriting provided by *SymbolicUtils.jl* (Gowda et al., 2020). For analyzing arbitrary expressions, the properties are recursively added on by a postorder tree traversal. This approach allows for greater flexibility and modularity in defining new atoms and rules, enabling the incorporation of domain-specific atoms. Since the atoms are directly the Julia functions, the DGCP implementation avoids the need to create and maintain implementations of numerical routines.

4.1 Atom Library

The atoms in DGCP are stored as a key-value pair in a dictionary. Wherein the key is the Julia method corresponding to the atom and the value is a tuple containing the manifold, the sign of the function, and its known geodesic curvature and the monotonicity. For a Julia function to be compliant with the rule propagation discussed in the next sub-section, it needs to be a registered primitive in *Symbolics* through the `@register_symbolic` macro from *Symbolics*. For example, the `logdet` atom representing the log-determinant of a symmetric positive definite matrix, implemented with the function from the *LinearAlgebra* standard library of Julia, is defined as follows:

```julia
@register_symbolic LinearAlgebra.logdet(X::Matrix{Num})
add_gdcprule(LinearAlgebra.logdet, SymmetricPositiveDefinite, Positive,
↪  GLinear, GIncreasing)
```

Listing 1: The `logdet` atom is defined on the `SymmetricPositiveDefinite` manifold, has a positive sign, is geodesically linear, and is geodesically increasing.

Some atoms in DGCP do not have preexisting implementations in Julia, so first a function is defined for it and the same machinery as before is then used to register. For instance, the `conjugation` atom is defined as follows:

```julia
function conjugation(X, B)
    return B' * X * B
end

@register_array_symbolic conjugation(X::Matrix{Num}, B::Matrix) begin
    size = (size(B, 2), size(B, 2))
end

add_gdcprule(conjugation, SymmetricPositiveDefinite, Positive, GConvex,
↪  GIncreasing)
```

Listing 2: The `conjugation` atom is defined on the `SymmetricPositiveDefinite` manifold, has a positive sign, is geodesically convex, and is geodesically increasing.

The extensibility of the atom library is an important feature of this implementation. Users can define atoms and specify their properties using the provided macros and functions, allowing the incorporation of domain-specific atoms and the ability to handle a wide range of optimization problems. The modular design of the atom library enables the addition of new atoms without modifying the core implementation and allows more disciplined programming paradigms to be implemented similarly.

4.2 Rewriting System for Rule Propagation

The DGCP compliant ruleset 3.2.2 lends itself naturally to a rewriting system (Dershowitz and Jouannaud, 1990), as has been shown before for DCP (Agrawal et al., 2018). The *SymbolicUtils.jl* package provides the rewriting infrastructure that enables the application of DGCP rules to symbolic expressions.

In the DGCP implementation, rewriting is employed to propagate the mathematical properties of functions as metadata. The rewriting system applies the rules using a post-order traversal of the expression tree, ensuring that the properties of subexpressions are propagated before determining the properties of parent expressions.

The DGCP ruleset is implemented using the `@rule` macro. For example, the following rule propagates the curvature through addition of subexpressions:

```
@rule +(~~x) => setgcurvature(~MATCH, add_gcurvature(~~x))
```

Listing 3: Using the `@rule` macro for propagating Geodesic Curvature through addition

This rule matches an addition expression `+(~~x)` and sets the curvature of the matched expression (`~MATCH`) to the result of the `add_gcurvature` function applied to the subexpressions (`~~x`).

The rewriting and metadata propagation from *SymbolicUtils* allows for a declarative specification of the rules, reducing the lines of code required to implement the DGCP ruleset.

4.3 Integration with Optimization Frameworks

To leverage the DGCP in applications, we require an integration of our framework with manifold optimization software for solving the verified programs. This has been done with *OptimizationManopt*, which is the interface to *Manopt.jl* with the *Optimization.jl* (Dixit and Rackauckas, 2023) package. This integration allows us to define the optimization problem, either with an algebraic or a functional interface, and perform this analysis to determine whether the objective function and/or constraints are geodesically convex.

During the initialization phase in *Optimization.jl*, the symbolic expressions for the objective function and constraints are generated by tracing through the imperative code with symbolic variables. This automatic generation of symbolic expressions allows for a transition from the optimization problem specification to the symbolic representation required for verification with DGCP. As mentioned above, this can also be done by using the algebraic interface, in which case the analysis still proceeds as before, except that symbolic tracing isn't needed as the user already provides the expression.

The generated symbolic expressions are then leveraged to propagate the sign information and geodesic curvature using the `propagate_sign`, and `propagate_gcurvature` functions, and the user is informed if the problem can be recognized to be disciplined geodesically convex or otherwise. For example:

After the curvature propagation step, the optimization problem can be solved using the selected solver from *Manopt.jl*, thus providing a generic non-linear programming interface for Riemannian solvers. Hence, future work on other manifolds can be integrated trivially and solved using specialized algorithms.

```
julia> A = randn(5, 5) #initialize random matrix
       A = A * A' #make it a SPD matrix

       function matsqrt(X, p = nothing) #setup objective function
           return SymbolicAnalysis.sdivergence(X, A) +
                  SymbolicAnalysis.sdivergence(X,
                  ↪  Matrix{Float64}(LinearAlgebra.I(5)))
       end

       optf = OptimizationFunction(matsqrt, Optimization.AutoZygote())
       ↪  #setup oracles
       prob = OptimizationProblem(optf, A / 2, manifold = M) #setup problem
       ↪  with manifold and initial point

       sol = solve(prob, GradientDescentOptimizer()) #solve the problem
[ Info: Objective Euclidean curvature: UnknownCurvature
[ Info: Objective Geodesic curvature: GConvex
```

Listing 4: Solving the matrix square root problem in geodesically convex formulation from (Sra, 2015) with Geodesic Convexity certificate.

5 Applications

In this section we illustrate the analysis and verification of geodesic convexity with DGCP on four problems.

5.1 Matrix Square Root

Computing the square root $A^{\frac{1}{2}}$ of a symmetric positive definite matrix $A \in \mathbb{P}_d$ is an important subroutine in many statistics and machine learning applications. Among other, several first-order approaches have been introduced (Jain et al., 2017; Sra, 2015). Notably, Sra (2015) gives a geodesically convex formulation of the problem, given by

$$\min_{X \in \mathbb{P}_d} \phi(X) := \delta_S^2(X, A) + \delta_S^2(X, I)\ , \tag{3}$$

where δ_s denotes the s-divergence. Listing 4.3 illustrates the use of DGCP to verify the geodesic convexity of Eq. 3 and leverage the optimization interface to solve the verified problem with a Riemannian solver.

5.2 Karcher Mean

Given a set of symmetric positive definite matrices $\{A_j\} \subseteq \mathbb{P}_d$, the Karcher mean is defined as the solution to the problem

$$X^* \stackrel{\text{def}}{=} \operatorname*{argmin}_{X \succ 0} \left[\phi(X) = \sum_{i=1}^{m} w_i \delta_2^2 \left(X, A_i\right) \right] , \tag{4}$$

where $w_i \geq 0$ are the weights, and

$$d_2^2(X, A) = \left\| \log \left(A^{-\frac{1}{2}} X A^{-\frac{1}{2}} \right) \right\|_F^2 , \qquad X, Y \in \mathbb{P}_d \tag{5}$$

is the *Riemannian distance* of the $\mathbb{P}_d$ manifold. The Karcher mean has found applications in medical imaging (Carmichael et al., 2013), kernel methods (Jayasumana et al., 2013), and interpolation (Absil et al., 2016). Since (4) is a conic sum of g-convex functions the problem itself is g-convex. However, the problem is not Euclidean convex. Notably, Problem 4 does not admit a closed form solution for $m > 2$. Hence, in contrast to other notions of matrix averages (e.g. arithmetic and geometric mean), the computation of the Karcher mean requires Riemannian solvers.

Using DGCP, we can test and verify these convexity properties as follows:

```
julia> M = SymmetricPositiveDefinite(5)
       objective_expr = sum(Manifolds.distance(M, As[i], X)^2 for i in 1:5)
       analyze_res = analyze(objective_expr, M)
       println(analyze_res.gcurvature)
GConvex
```

5.3 Computation of Brascamp-Lieb Constants

The Brascamp-Lieb (short: BL) inequalities (Brascamp and Lieb, 1976; Brascamp et al., 1974) form an important class of inequalities that encompass many well-known inequalities (e.g. Hölder's inequality, Loomis–Whitney inequality, etc.) in functional analysis and probability theory. Beyond its applications in various mathematical disciplines, the BL inequalities have applications in machine learning and information theory (Dvir et al., 2018; Hardt and Moitra, 2013; Carlen and Cordero-Erausquin, 2009; Liu et al., 2016).

Crucial properties of BL inequalities are characterized by so-called *BL-datum* $(\mathcal{A}, w)$, where $\mathcal{A} = \left(A_1, \ldots, A_m\right)$ is a tuple of surjective, linear transformations and $\boldsymbol{w} = (w_1, \ldots, w_m)$ is a vector with real, non-negative entries. The BL datum defines a corresponding BL inequality

$$\int_{x \in \mathbb{R}^d} \Big(\prod_{j \in [m]} f_j(A_j x)^{w_j} \Big) dx \leq C(\mathcal{A}, \boldsymbol{w}) \prod_{j \in [m]} \Big(\int_{x \in \mathbb{R}^{d'}} f_j(x) dx \Big)^{w_j} , \tag{6}$$

where $f_j : \mathbb{R}^{d'} \to \mathbb{R}$ denote real-valued, non-negative, Lebesgue-measurable functions. The properties of this inequality a characterized by the *BL-constant*, which corresponds to the smallest constant $C(\mathcal{A}, \boldsymbol{w})$ for which the above inequality holds. The value of $C(\mathcal{A}, \boldsymbol{w})$ (and whether it is finite or infinite) is of crucial importance in practise.

The computation of BL constants can be formulated as an optimization task on the positive definite matrices (Brascamp and Lieb, 1976; Brascamp et al., 1974); one formulation of which is given by (Sra et al.)

$$\min_{X \in \mathbb{P}_d} \left[F(X) = -\log\det(X) + \sum_i w_i \log\det\left(A_i^\top X A_i\right)\right]. \tag{7}$$

This problem is g-convex, but not Euclidean convex, which has motivated the analysis of this problem with g-convex optimization tools (Gurvits, 2004; Garg et al., 2018; Bürgisser et al., 2018; Weber and Sra, 2022a). We can test and verify the convexity properties of problem 7 as follows:

```
julia> M = SymmetricPositiveDefinite(5)
       objective_expr = logdet(SymbolicAnalysis.conjugation(X, A)) -
       ↪  logdet(X)
       analyze_res = analyze(objective_expr, M)
       println(analyze_res.gcurvature)
GConvex
```

5.4 Robust Subspace Recovery

Robust subspace recovery seeks to find a low-dimensional subspace in which a (potentially noisy) data set concentrates. Standard dimensionality reduction approaches, such as Principal Component Analysis, can perform poorly in this setting, which motivates the use of other, more robust statistical estimators. One popular choice is Tyler's M-estimator (Tyler, 1987). It can be interpreted as the maximum likelihood estimator for the multivariate student distribution with degrees of freedom parameter $\nu \to 0$ (Maronna et al., 2006). Since the multivariate Student distribution is heavy-tailed, Tyler's M-estimator is more robust to outliers.

Suppose our given data set consists of observations $\{x_i\}_{i=1}^N \subseteq \mathbb{R}_d$. Then Tyler's M-estimator is given by the solution to the following geometric optimization problem, defined on the positive definite matrices:

$$\Sigma = \operatorname*{arg\,min}_{\Sigma \in \mathbb{P}_d} \frac{1}{n} \sum_{i=1}^n \log\left(x_i^\top \Sigma^{-1} x_i\right) + \frac{1}{d} \log\det(\Sigma). \tag{8}$$

Notably, this problem is g-convex. To see this, note that the function

$$f_i(\Sigma) = \log\left(x_i^\top \Sigma^{-1} x_i\right) \qquad i = 1, \dots n$$

is g-convex, which follows from the g-convexity of the function $g_i(\Sigma) = \log\left(x_i^\top \Sigma x_i\right)$ (see Proposition 12) and Lemma 5. Moreover, the function $f(\Sigma) = \log\det\Sigma$ is g-convex. Thus, problem 8 is g-convex following Proposition 1.

We note that to ensure an unique solution to Problem (8) one typically enforces the condition $\mathrm{tr}(\Sigma) = c$ for some constant $c > 0$. However, for the purposes of this paper, we restrict our focus on verifying the geodesic convexity of the standard formulation using DGCP; the corresponding expression is shown below.

```
julia> @variables Sigma[1:5, 1:5]
       xs = [rand(5) for i in 1:2]
       ex = sum(SymbolicAnalysis.log_quad_form(x, inv(Sigma)) for x in xs)
        ↪  + 1/5*logdet(Sigma)
       analyze_res = SymbolicAnalysis.analyze(ex, M)
       println(analyze_res.gcurvature)
GConvex
```

6 Conclusions

In this paper we introduced the *Disciplined Geodesically Convex Programming* (*DGCP*) framework, which allows for testing and certifying the geodesic convexity of objective functions and constraints in geometric optimization problems. The paper is accompanied by the package *SymbolicAnalysis.jl*, which implements the foundational atoms and rules of our framework, as well as an interface with *Manopt.jl* and *Optimization.jl* that provides access to standard solvers for the verified programs.

The initial implementation of DGCP is limited to basic atoms and rules, which allow for verifying the geodesic convexity of several classical tasks. However, the implementation of additional atoms and rules could significantly widen the range of applications. In particular, future work could focus on implementing additional functionality for verifying program structures that frequently occur in machine learning and statistical data analysis, which we envision as major application areas of our framework. Furthermore, our current framework focuses solely on optimization tasks on symmetric positive definite matrices. While this setting is often considered in the geodesically convex optimization literature, we note that geodesically convex problems arise on more general classes of manifolds, specifically, Cartan-Hadamard manifolds. While we present a general set of rules for geodesic convexity preserving operations on such manifolds, specialized sets of atoms need to be defined for individual manifolds. An extension of the DGCP framework and *SymbolicAnalysis.jl* package beyond the manifold of symmetric positive definite matrices is an important avenue for future work. Even in the special case of symmetric positive definite matrices, other (Riemannian) metrics could be considered. For instance, recent literature has analyzed optimization tasks on positive definite matrices through the lens of Bures-Wasserstein (Chewi et al., 2020) and Thompson (Weber and Sra, 2022a) geometries.

The *Optimization.jl* interface for *Manopt* is under active development to achieve feature parity. Enhancing this interface will be crucial in enabling the community to more effectively leverage the contributions from this work. Other directions for future work include the improvement and extension of the *SymbolicAnalysis.jl* package. Currently, we only provide an implementation of DGCP in Julia; however, other languages, in particular Python and Matlab, are popular in the Riemannian optimization community. Hence, providing an implementation in these languages could make our framework more widely applicable.

Acknowledgments and Disclosure of Funding

We thank Shashi Gowda, Christopher Rackauckas, Theo Diamandis, Flemming Holtorf and Alan Edelman from the Julia Lab, as well as Ronny Bergmann, for helpful discussions and comments.

AC and MW were partially supported by the Harvard Dean's Competitive Fund for Promising Scholarship and NSF award 2112085. AC was partially supported by an NSERC Postgraduate Fellowship.

VD is a member of the Julia Lab, which acknowledges the following support: This material is based upon work supported by the National Science Foundation under grant no. OAC-1835443, grant no. SII-2029670, grant no. ECCS-2029670, grant no. OAC-2103804, grant no. DMS-2325184, and grant no. PHY-2021825. The information, data, or work presented herein was funded in part by the Advanced Research Projects Agency-Energy (ARPA-E), U.S. Department of Energy, under Award Number DE-AR0001211 and DE-AR0001222. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof. This material was supported by The Research Council of Norway and Equinor ASA through Research Council project "308817 - Digital wells for optimal production and drainage". Research was sponsored by the United States Air Force Research Laboratory and the United States Air Force Artificial Intelligence Accelerator and was accomplished under Cooperative Agreement Number FA8750-19-2-1000. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the United States Air Force or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

P.-A. Absil, Pierre-Yves Gousenbourger, Paul Striewski, and Benedikt Wirth. Differentiable Piecewise-Bézier Surfaces on Riemannian Manifolds. *SIAM Journal on Imaging Sciences*, 9(4):1788–1828, 2016.

Akshay Agrawal and Stephen Boyd. Disciplined quasiconvex programming. *Optimization Letters*, 14(7):1643–1657, 2020.

Akshay Agrawal, Robin Verschueren, Steven Diamond, and Stephen Boyd. A rewriting system for convex optimization problems. *Journal of Control and Decision*, 5(1):42–60, 2018.

Akshay Agrawal, Steven Diamond, and Stephen Boyd. Disciplined geometric programming. *Optimization Letters*, 13:961–976, 2019.

Seth D Axen, Mateusz Baran, Ronny Bergmann, and Krzysztof Rzecki. Manifolds. jl: an extensible julia framework for data analysis on manifolds. *ACM Transactions on Mathematical Software*, 49(4):1–23, 2023.

Miroslav Bacák. Convex analysis and optimization in Hadamard spaces. In *Convex Analysis and Optimization in Hadamard Spaces*. de Gruyter, 2014.

Ronny Bergmann. Manopt.jl: Optimization on manifolds in Julia. *Journal of Open Source Software*, 7(70):3866, 2022. doi: 10.21105/joss.03866.

Ronny Bergmann and Roland Herzog. Intrinsic formulation of kkt conditions and constraint qualifications on smooth manifolds. *SIAM Journal on Optimization*, 29(4):2423–2444, 2019.

Ronny Bergmann, Roland Herzog, Julián Ortiz López, and Anton Schiela. First-and second-order analysis for optimization problems with manifold-valued constraints. *Journal of Optimization Theory and Applications*, 195(2):596–623, 2022.

Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. Julia: A fresh approach to numerical computing. *SIAM review*, 59(1):65–98, 2017.

Rajendra Bhatia. *Matrix Analysis*. Springer New York, 1997.

Rajendra Bhatia. *Positive Definite Matrices*. Princeton Series in Applied Mathematics. Princeton University Press, Princeton, NJ, USA, 2007.

Rajendra Bhatia. *Positive Definite Matrices*. Princeton University Press, USA, 2015. ISBN 0691168253.

Jacob Bien. Sparse Estimation of a Gaussian's Covariance Matrix. Talk at Statistical and Applied Mathematical Sciences Institute, NC State University, 2018.

Silvere Bonnabel. Stochastic gradient descent on Riemannian manifolds. *IEEE Transactions on Automatic Control*, 58(9):2217–2229, 2013.

Nicolas Boumal. *An introduction to optimization on smooth manifolds*. Cambridge University Press, 2023.

Nicolas Boumal, Bamdev Mishra, P.-A. Absil, and Rodolphe Sepulchre. Manopt, a Matlab toolbox for optimization on manifolds. *Journal of Machine Learning Research*, 15(42): 1455–1459, 2014. URL `https://www.manopt.org`.

Nicolas Boumal, Pierre-Antoine Absil, and Coralia Cartis. Global rates of convergence for nonconvex optimization on manifolds. *IMA Journal of Numerical Analysis*, 39(1):1–33, 2019.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

Herm Jan Brascamp and Elliott H. Lieb. Best constants in Young's inequality, its converse, and its generalization to more than three functions. *Advances in Mathematics*, 20(2):151 – 173, 1976.

Herm Jan Brascamp, Elliott H. Lieb, and Joaquin Mazdak Luttinger. A general rearrangement inequality for multiple integrals. *Journal of Functional Analysis*, 17(2):227 – 237, 1974.

Peter Bürgisser, Cole Franks, Ankit Garg, Rafael Oliveira, Michael Walter, and Avi Wigderson. Efficient algorithms for tensor scaling, quantum marginals, and moment polytopes. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 883–897. IEEE, 2018.

Eric A Carlen and Dario Cordero-Erausquin. Subadditivity of the entropy and its relation to Brascamp–Lieb type inequalities. *Geometric and Functional Analysis*, 19(2):373–405, 2009.

Owen Carmichael, Jun Chen, Debashis Paul, and Jie Peng. Diffusion tensor smoothing through weighted karcher means. *Electron. J. Stat.*, 7(none):1913–1956, 2013.

Andrew Cheng and Melanie Weber. Structured Regularization for Constrained Optimization on the SPD Manifold. *In preparation*, 2024.

Sinho Chewi, Tyler Maunu, Philippe Rigollet, and Austin J Stromme. Gradient descent algorithms for Bures-Wasserstein barycenters. In *Conference on Learning Theory*, pages 1276–1304. PMLR, 2020.

Thomas A Courtade, Max Fathi, and Ashwin Pananjady. Wasserstein stability of the entropy power inequality for log-concave random vectors. In *2017 IEEE International Symposium on Information Theory (ISIT)*, pages 659–663. IEEE, 2017.

Nachum Dershowitz and Jean-Pierre Jouannaud. Rewrite systems. In *Formal models and semantics*, pages 243–320. Elsevier, 1990.

Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

Vaibhav Kumar Dixit and Christopher Rackauckas. Optimization.jl: A unified optimization package, March 2023. URL `https://doi.org/10.5281/zenodo.7738525`.

Zeev Dvir, Ankit Garg, Rafael Oliveira, and József Solymosi. Rank bounds for design matrices with block entries and geometric applications. *Discrete Analysis*, 5:24, 2018.

Ankit Garg, Leonid Gurvits, Rafael Oliveira, and Avi Wigderson. Algorithmic and optimization aspects of Brascamp-Lieb inequalities, via operator scaling. *Geometric and Functional Analysis*, 28(1):100–145, 2018.

Shashi Gowda, Yingbo Ma, and Mason Protter. SymbolicUtils.jl: Symbolic expressions, rewriting and simplification. `https://github.com/JuliaSymbolics/SymbolicUtils.jl`, 2020.

Shashi Gowda, Yingbo Ma, Alessandro Cheli, Maja Gwóźzdź, Viral B. Shah, Alan Edelman, and Christopher Rackauckas. High-performance symbolic-numerics via multiple dispatch. *ACM Commun. Comput. Algebra*, 55(3):92–96, 2022.

Michael Grant, Stephen Boyd, and Yinyu Ye. *Disciplined Convex Programming*. Springer, 2006.

Leonid Gurvits. Classical complexity and quantum entanglement. *Journal of Computer and System Sciences*, 69(3):448 – 484, 2004. ISSN 0022-0000. Special Issue on STOC 2003.

Moritz Hardt and Ankur Moitra. Algorithms and hardness for robust subspace recovery. In *Proceedings of the 26th Annual Conference on Learning Theory*, volume 30, pages 354–375, Princeton, NJ, USA, 12–14 Jun 2013. PMLR.

Roger A. Horn and Charles R. Johnson. *Matrix analysis*. Cambridge University Press, 2013.

Wen Huang, P.-A. Absil, K. A. Gallivan, and Paul Hand. ROPTLIB: an object-oriented C++ library for optimization on Riemannian manifolds. Technical Report FSU16-14.v2, Florida State University, 2016.

Prateek Jain, Chi Jin, Sham Kakade, and Praneeth Netrapalli. Global convergence of non-convex gradient descent for computing matrix squareroot. In *Artificial Intelligence and Statistics*, pages 479–488. PMLR, 2017.

Sadeep Jayasumana, Richard Hartley, Mathieu Salzmann, Hongdong li, and Mehrtash Harandi. Kernel Methods on the Riemannian Manifold of Symmetric Positive Definite Matrices. 06 2013.

Michael Jordan, Tianyi Lin, and Emmanouil-Vasileios Vlatakis-Gkaragkounis. First-order algorithms for min-max optimization in geodesic metric spaces. *Advances in Neural Information Processing Systems*, 35:6557–6574, 2022.

Jingbo Liu, Thomas A Courtade, Paul Cuff, and Sergio Verdú. Smoothing Brascamp-Lieb inequalities and strong converses for common randomness generation. In *2016 IEEE International Symposium on Information Theory (ISIT)*, pages 1043–1047. IEEE, 2016.

Zelda Mariet and Suvrit Sra. Fixed-point algorithms for learning determinantal point processes. In *International Conference on Machine Learning*, pages 2389–2397. PMLR, 2015.

Ricardo A. Maronna, R. Douglas Martin, and Víctor J. Yohai. *Robust Statistics: Theory and Methods*. Wiley, March 2006.

David Martínez-Rubio, Christophe Roux, Christopher Criscitiello, and Sebastian Pokutta. Accelerated Methods for Riemannian Min-Max Optimization Ensuring Bounded Geometric Penalties. *arXiv preprint arXiv:2305.16186*, 2023.

Viet Anh Nguyen, Soroosh Shafieezadeh Abadeh, Man-Chung Yue, Daniel Kuhn, and Wolfram Wiesemann. Calculating optimistic likelihoods using (geodesically) convex optimization. *Advances in Neural Information Processing Systems*, 32, 2019.

Esa Ollila and David E Tyler. Regularized M-estimators of scatter matrix. *IEEE Transactions on Signal Processing*, 62(22):6059–6070, 2014.

Suvrit Sra. On the matrix square root via geometric optimization. *arXiv preprint arXiv:1507.08366*, 2015.

Suvrit Sra and Reshad Hosseini. Conic geometric optimization on the manifold of positive definite matrices. *SIAM Journal on Optimization*, 25(1):713–739, 2015.

Suvrit Sra, Nisheeth K. Vishnoi, and Ozan Yildiz. On Geodesically Convex Formulations for the Brascamp-Lieb Constant. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2018)*, pages 25:1–25:15.

James Townsend, Niklas Koep, and Sebastian Weichwald. Pymanopt: A python toolbox for optimization on manifolds using automatic differentiation. *Journal of Machine Learning Research*, 17(137):1–5, 2016.

David E. Tyler. A distribution-free m-estimator of multivariate scatter. *The Annals of Statistics*, 15(1), March 1987.

Madeleine Udell, Karanveer Mohan, David Zeng, Jenny Hong, Steven Diamond, and Stephen Boyd. Convex optimization in Julia. *SC14 Workshop on High Performance Technical Computing in Dynamic Languages*, 2014.

Constantin Udriste. *Convex Functions and Optimization Methods on Riemannian Manifolds*, volume 297. Springer Science & Business Media, 1994.

Nisheeth K. Vishnoi. Geodesic convex optimization: Differentiation on manifolds, geodesics, and convexity. *ArXiv*, 2018.

Melanie Weber and Suvrit Sra. Projection-free nonconvex stochastic optimization on Riemannian manifolds. *IMA Journal of Numerical Analysis*, 42(4):3241–3271, 2021.

Melanie Weber and Suvrit Sra. Computing Brascamp-Lieb constants through the lens of Thompson geometry. *arXiv preprint arXiv:2208.05013*, 2022a.

Melanie Weber and Suvrit Sra. Riemannian Optimization via Frank-Wolfe Methods. *Mathematical Programming*, 2022b.

Melanie Weber and Suvrit Sra. Global optimality for Euclidean CCCP under Riemannian convexity. In *International Conference on Machine Learning*, 2023.

Ami Wiesel. Geodesic convexity and covariance estimation. *IEEE Transactions on Signal Processing*, 60(12):6182–6189, 2012.

Ami Wiesel and Teng Zhang. Structured robust covariance estimation. *Foundations and Trends® in Signal Processing*, 8(3):127–216, 2015.

Hongyi Zhang and Suvrit Sra. First-order methods for geodesically convex optimization. In *COLT*, 2016.

Hongyi Zhang, Sashank J Reddi, and Suvrit Sra. Riemannian SVRG: Fast stochastic optimization on Riemannian manifolds. *Advances in Neural Information Processing Systems*, 29, 2016.

Teng Zhang. Robust subspace recovery by Tyler's M-estimator. *Information and Inference: A Journal of the IMA*, 5(1):1–21, 2016.

Appendix A. Deferred Proofs

Notation. For any two symmetric positive definite matrices $A, B \in \mathbb{P}_d$ we use the notation $A \sharp B$ to denote the geometric mean between A and B

$$A \sharp B \stackrel{\text{def}}{=} A^{\frac{1}{2}} \left(A^{-1/2} B A^{-1/2} \right)^{\frac{1}{2}} A^{\frac{1}{2}}.$$

Moreover, we use the $A \sharp_t B$ to denote the geodesic connecting A to B

$$A \sharp_t B \stackrel{\text{def}}{=} A^{\frac{1}{2}} \left(A^{-1/2} B A^{-1/2} \right)^{t} A^{\frac{1}{2}} \qquad \forall t \in [0,1].$$

We will use the following lemma in the proofs to come.

Lemma 12 *For any $A, B \in \mathbb{P}_d$ it holds that*

$$(A \sharp_t B)^{-1} = A^{-1} \sharp_t B^{-1} .$$

Proof This follows from the basic computation

$$\begin{aligned}
(A \sharp_t B)^{-1} &= \left(A^{\frac{1}{2}} \left(A^{-1/2} B A^{-1/2} \right)^{t} A^{\frac{1}{2}} \right)^{-1} \\
&= A^{-\frac{1}{2}} \left(A^{1/2} B^{-1} A^{1/2} \right)^{t} A^{-\frac{1}{2}} \\
&= A^{-1} \sharp_t B^{-1} .
\end{aligned}$$

■

Lemma 13 (Midpoint convexity) *A continuous function f on a g-convex set $\mathcal{S} \subseteq \mathcal{M}$ is g-convex if $f(X \sharp Y) \leq \frac{1}{2} f(X) + \frac{1}{2} f(Y)$ for any $X, Y \in \mathcal{S}$.*

Proof The proof is analogous to showing the Euclidean midpoint convex condition. Namely, instead of recursively applying the hypothesis to line segments of length 2^{-k} for $k \in \mathbb{N}$, we apply it to the midpoints of geodesic segments.

Let $X_0, Y_0 \in \mathcal{S}$. Let $\gamma : [0,1] \to \mathcal{M}$ be a geodesic segment such that $\gamma(0) = X_0 \neq Y_0 = \gamma(1)$ and $\gamma(t) \in S$ for all $t \in [0,1]$.

We need to verify f is geodesically convex, i.e. show that

$$f(\gamma(t)) \leq (1-t) f(\gamma(0)) + t f(\gamma(1)) \tag{9}$$

holds for all $t \in [0,1]$. The hypothesis implies (9) holds for $t = \frac{1}{2}$. Since $\gamma(\frac{1}{2}) \in \mathcal{S}$, we can now recursively apply the hypothesis to the sub-geodesic segments defined by the images $\gamma\left([0, \frac{1}{2}]\right)$ and $\gamma\left([\frac{1}{2}, 1]\right)$. In turn, (9) holds for $t \in \{0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}$. Applying this argument k times shows that (9) holds for $t \in \mathcal{I}_K \overset{\text{def}}{=} \{\frac{\ell}{2^k} : 0 \leq \ell \leq 2^k\}$. The set $\mathcal{I}_\infty$ is dense in $[0,1]$ the argument follows by the continuity of f. ■

A.1 Rules

Proof [Proposition 1]

We prove the proposition for the case $n = 2$ and note that the arguments can be easily generalized for arbitrary $n \in \mathbb{N}$.

Consider $f, g : S \subseteq \mathcal{M} \to \mathbb{R}$ to be two g-convex functions on a g-convex set S. Let $x, y \in S$ and $\gamma : [0,1] \to \mathcal{M}$ be a geodesic that connects $\gamma(0) = x$ to $\gamma(1) = y$ such that $\gamma[0,1] \subseteq S$. Then for all $t \in [0,1]$,

$$\begin{aligned}\alpha f(\gamma(t)) + \beta g(\gamma(t)) &\leq \alpha\Big((1-t)f(\gamma(0)) + tf(\gamma(1))\Big) + \beta\Big((1-t)g(\gamma(0)) + tg(\gamma(1))\Big) \\ &= (1-t)\left(\alpha f(\gamma(0)) + \beta g(\gamma(0))\right) + t\big(\alpha f(\gamma(1)) + \beta g(\gamma(1))\big).\end{aligned}$$

Moreover,

$$\begin{aligned}\max\left\{f(\gamma(t)), g(\gamma(t))\right\} &\leq \max\left\{(1-t)f(\gamma(0)) + tf(\gamma(1)), (1-t)g(\gamma(0)) + tg(\gamma(1))\right\} \\ &\leq (1-t)\max\left\{f(\gamma(0)), g(\gamma(0))\right\} + t\max\left\{f(\gamma(1)), g(\gamma(1))\right\}.\end{aligned}$$

■

Proof [Proposition 2] By applying convexity results and the fact that $h(\cdot)$ is nondecreasing we obtain

$$h(f(\gamma(t)) \leq h\left((1-t)f(\gamma(0)) + tf(\gamma(1))\right) \leq (1-t)h(f(\gamma(0))) + th(f(\gamma(1))).$$

■

Proof [Proposition 5] Suppose $A, B \in \mathbb{P}_d$ and $f(X)$ is g-convex. Then for all $t \in [0,1]$ we have

$$g(A\sharp_t B) = f\left((A\sharp_t B)^{-1}\right) = f(A^{-1}\sharp_t B^{-1}) \leq (1-t)\, f\left(A^{-1}\right) + tf\left(B^{-1}\right) = (1-t)g(A) + tg(B)$$

where in the second equality we applied Lemma 12.

■

In order to prove Proposition 7 we need the following lemmas.

Lemma 14 (Theorem 4.1.3 Bhatia (2007)) *Let $A, B \in \mathbb{P}_d$. Their geometric mean $A\sharp B$ satisfies the following extremal property:*

$$A\sharp B = \max\{X : X = X^\top,\ \begin{bmatrix} A & X \\ X & B \end{bmatrix} \succeq 0\}.$$

In particular, if X is symmetric and satisfies the condition

$$\begin{bmatrix} A & X \\ X & B \end{bmatrix} \succeq 0$$

then $A\sharp B \succeq X$.

Lemma 15 *If $X \succeq 0$ then the matrix $\tilde{X}$ defined as follows satisfies*

$$\tilde{X} = \begin{bmatrix} X & X \\ X & X \end{bmatrix} \succeq 0.$$

Lemma 16 *Let $B \succeq 0$ and $\Phi(X)$ be a positive linear map, that is, $\Phi(X) \succeq 0$ whenever $X \succeq 0$. Then the function $\phi : \mathbb{R}^{d\times d} \to \mathbb{R}^{d\times d}$ defined by $\phi(X) \overset{def}{=} \Phi(X) + B$*

$$\phi\left(X\sharp Y\right) \preceq \phi(X)\sharp\phi(Y) \qquad \forall X, Y \in \mathbb{P}_d.$$

Proof [Lemma 16] Let $X, Y \in \mathbb{P}_d$ and since $X\sharp Y \in \mathbb{P}_d$ we have by Exercise 3.2.2 (ii) Bhatia (2007) that

$$\begin{bmatrix} X & X\sharp Y \\ X\sharp Y & Y \end{bmatrix} \succeq 0 \implies \begin{bmatrix} \Phi(X) & \Phi\left(X\sharp Y\right) \\ \Phi\left(X\sharp Y\right) & \Phi(Y) \end{bmatrix} \succeq 0. \tag{10}$$

By applying Lemma 15 we have

$$\begin{bmatrix} B & B \\ B & B \end{bmatrix} \succeq 0$$

thus we have

$$\begin{bmatrix} \Phi(X) & \Phi\left(X\sharp Y\right) \\ \Phi\left(X\sharp Y\right) & \Phi(Y) \end{bmatrix} + \begin{bmatrix} B & B \\ B & B \end{bmatrix} = \begin{bmatrix} \Phi(X) + B & \Phi\left(X\sharp Y\right) + B \\ \Phi\left(X\sharp Y\right) + B & \Phi(Y) + B \end{bmatrix} = \begin{bmatrix} \phi(X) & \phi\left(X\sharp Y\right) \\ \phi\left(X\sharp Y\right) & \phi(Y) \end{bmatrix} \succeq 0. \tag{11}$$

By applying the extremal characterization of geometric mean we get $\phi(X)\sharp\phi(Y) \succeq \phi(X\sharp Y)$ which is our desired result. ■

Now we can prove Proposition 7.

Proof [Proposition 7] It suffices to check midpoint convexity.

$$\begin{aligned} g(X\sharp Y) &\overset{\text{def}}{=} f\left(\phi(X\sharp Y)\right) \\ &\preceq f\left(\phi(X)\sharp\phi(Y)\right) \qquad \text{(Lemma 16)} \\ &\preceq \frac{f(\phi(X)) + f(\phi(Y))}{2} \qquad (f \text{ is g-convex}) \\ &= \frac{g(X) + g(Y)}{2}. \end{aligned}$$

■

A.2 Atoms

In this section, we prove that the list of atoms in Section 3.3 is g-convex with respect to the canonical Riemannian metric. The proofs demonstrate the application of the propositions found in Section 3.2.2.

Lemma 17 (Epigraphs and g-convexity (Lemma 2.2.1, Bacák (2014))) *Let $f : \mathbb{P}_d \to \mathbb{R}$ be geodesically convex and define its epigraph as*

$$\textbf{\textit{epi}}(f) \overset{def}{=} \{(X, t) : X \in \mathbb{P}_d \text{ and } f(X) \leq t\} \subseteq S \times \mathbb{R}.$$

Then f is geodesically convex if and only if $\textbf{\textit{epi}}(f)$ is a closed geodesically convex subset of $\mathbb{P}_d \times \mathbb{R}$.

Proposition 9 *Let $S \subseteq \mathbb{R}^d$ and $y \in S$. Suppose $f(X, y) : \mathbb{P}_d \to \mathbb{R}$ is g-convex in X, then define the function $g : \mathbb{P}_d \to \mathbb{R}$ by*

$$g(X) = \sup_{y \in S} f(X, y).$$

Then $g(X, y)$ is g-convex on $\mathbb{P}_d$ with respect to the canonical Riemannian metric. The domain of g is

$$\textbf{\textit{dom}}(g) = \{X \in \mathbb{P}_d : (X, y) \in \textbf{\textit{dom}}(f) \text{ for all } y \in S,\ \sup_{y \in S} f(X, y) < \infty\}.$$

Proof We claim that

$$\mathbf{epi}(g) = \bigcap_{y \in S} \mathbf{epi}(f(\cdot, y)) \overset{\text{def}}{=} \bigcap_{y \in S} \{(X,t) : f(X,y) \leq t\}.$$

Let $(X,t) \in \mathbf{epi}(g)$. Then

$$\begin{aligned}
&\sup_{y \in S} f(X,y) \leq t \text{ and } X \in \mathbf{dom}(f) \\
&\iff f(X,y) \leq t \text{ for all } y \in S \text{ and } X \in \mathbf{dom}(f) \\
&\iff (X,t) \in \bigcap_{y \in S} \mathbf{epi}(f)(\cdot, y).
\end{aligned}$$

But $f(\cdot, y)$ is g-convex hence $\mathbf{epi} f(\cdot, y)$ is g-convex for all $y \in S$. Now note that the intersection of g-convex sets on Cartan-Hadamard manifolds (e.g., $\mathbb{P}_d$) is g-convex (see Chapter 11 in Boumal (2023)). By Proposition 17 we obtain our desired result. ■

Proposition 10 *Let $h, h_1 \ldots, h_n \in \mathbb{R}^d$ be fixed. The following functions $f : \mathbb{P}_d \to \mathbb{R}$ are geodesically convex with respect to the canonical Riemannian metric.*

(1) $f(X) = \log\left(\sum_{i=1}^n h_i^\top X h_i\right)$

(2) $f(X) = \log\det(X)$

(3) $f(X) = h^\top X h$

(4) $f(X) = tr(X)$

(5) $f(X) = \delta_S^2(X,Y) := \log\det\left(\frac{X+Y}{2}\right) - \frac{1}{2}\log\det(XY)$ *for fixed* $Y \in \mathbb{P}_d$.

(6) $f(X,Y) = \|\log\left(Y^{-\frac{1}{2}} X Y^{-\frac{1}{2}}\right)\|_F^2$ *for fixed* $Y \in \mathbb{P}_d$.

(7) $f(X) = \sup_{\{y:\mathbb{R}^d:\|y\|_2=1\}} y^\top X y$

(8) $f(X) = X^{-1}$.

Proof We defer the proofs of (1), (2), and (3) to Propositions 12, 13, and 14 respectively.

(4) It is clear that $\mathrm{tr}(X)$ is a strictly positive linear map and thus by Proposition 3 it is g-convex.

(5) For the S-divergence, we apply Proposition 6 with $h_1(X) = \log\det(X)$ and $\Phi(X) = \frac{X+Y}{2}$, i.e., the function

$$h_1(\Phi(X)) = \log\det\left(\frac{X+Y}{2}\right)$$

is g-convex. Moreover, by Proposition 4, we have that

$$X \mapsto -\log\det(X)$$

is g-convex (in fact, g-linear) and so

$$h_2(X) = -\frac{1}{2}\log\det(XY) = -\frac{1}{2}\left(\log\det(X) + \log\det(Y)\right)$$

is g-convex. Since conic combinations of g-convex functions are g-convex (see Proposition 1) we have that

$$\delta_S^2(X,Y) = h_1(X) + h_2(X)$$

is g-convex.

(6) We refer the reader to Corollary 19 ((Sra and Hosseini, 2015)) for a proof involving symmetric gauge functions. For a more general proof we refer the reader to Corollary 6.1.11 ((Bhatia, 2015)).

(7) This is a direct consequence of Proposition 9.

(8) It suffices to establish midpoint convexity. Observe that for any $A, B \in \mathbb{P}_d$

$$(A\sharp B)^{-1} = \left(A^{\frac{1}{2}}\left(A^{-\frac{1}{2}}BA^{-\frac{1}{2}}\right)^t A^{\frac{1}{2}}\right)^{-1} = A^{-\frac{1}{2}}\left(A^{\frac{1}{2}}B^{-1}A^{\frac{1}{2}}\right)^t A^{-\frac{1}{2}} = A^{-1}\sharp B^{-1}.$$

It follows from the AM-GM inequality for positive linear operators that

$$A^{-1}\sharp B^{-1} \preceq \frac{A^{-1}+B^{-1}}{2},$$

thus verifying g-convexity. ■

Appendix B. Additional results and discussion of g-convexity

We show that geodesic convexity, like Euclidean convexity, is generally not preserved under products.

Counterexample. For simplicity and without loss of generality we take $\log(\cdot) := \log_2(\cdot)$. We take $A = \mathrm{Diag}(1,1)$ and $B := \mathrm{Diag}(16,16)$ and the two g-convex functions to be $f_1(X) := \mathrm{tr}(X)$ and $f_2(X) = -\log\det(X)$. We show that $(f_1 f_2)(X) := -\mathrm{tr}(X)\log\det(X)$ is not g-convex. To this end, suppose $t = 1/2$. Then

$$\gamma(1/2) := A^{1/2}\left(A^{-1/2}BA^{-1/2}\right)^t A^{1/2} = \mathrm{Diag}(4,4).$$

Thus $f_1(\gamma(1/2))f_2(\gamma(1/2)) = -32$. Moreover, observe that

$$\begin{aligned} f_1(A) &= 2, \qquad & f_2(A) &= 0 \\ f_1(B) &= 32, \qquad & f_2(B) &= -8. \end{aligned}$$

Finally, we obtain

$$\frac{1}{2}\left(f_1(A)f_2(A)\right) + \frac{1}{2}\left(f_1(B)f_2(B)\right) = -128$$

Thus

$$f_1(\gamma(1/2))f_2(\gamma(1/2)) > \frac{1}{2}\left(f_1(A)f_2(A)\right) + \frac{1}{2}\left(f_1(B)f_2(B)\right)$$

thus $(f_1 f_2)(X)$ is not g-convex. $\square$

We show a function that is g-convex with respect to the Euclidean metric but not with respect to the canonical Riemannian metric.

Proposition 11 (Bien (2018)) *The function $f(X) := \|X\|_1 := \sum_{i,j}|X_{ij}|$ is g-convex with respect to the Euclidean metric but not with respect to the canonical Riemannian metric.*

Proof Let $f(X) := \|X\|_1 := \sum_{i,j}|X_{ij}|$ be the element-wise 1-norm. Observe for all $X, Y \in \mathbb{P}_d$

$$f\left(\theta X + (1-\theta)Y\right) = \sum_{ij=1}^{d}|\theta X_{ij} + (1-\theta)Y_{ij}| \le \theta\sum_{ij=1}^{d}|X_{ij}| + (1-\theta)\sum_{ij=1}^{d}|Y_{ij}| = \theta f(X) + (1-\theta)f(Y).$$

This establishes that f is g-convex with respect to the Euclidean metric on $\mathbb{P}_d$. In contrast, take the matrices

$$\Sigma_1 = I_3 \qquad \text{and} \qquad \Sigma_2 = \begin{pmatrix} 1.0 & 0.5 & -0.6 \\ 0.5 & 1.2 & 0.4 \\ -0.6 & 0.4 & 1.0 \end{pmatrix}.$$

Let $\gamma : [0,1] \to \mathbb{P}_d$ be the geodesic induced by the canonical Riemannian. metric. That is,

$$\gamma(t) = \Sigma_1^{1/2}\left(\Sigma_1^{-1/2}\Sigma_2\Sigma_1^{-1/2}\right)^t \Sigma_1^{1/2}.$$

Then observe that

$$f(\gamma(1/2)) = \|\Sigma_2^{1/2}\|_1 = 4.7638... > 4.6 = \frac{1}{2}\|\Sigma_1\|_1 + \frac{1}{2}\|\Sigma_2\|_1 = \frac{1}{2}f(\Sigma_1) + \frac{1}{2}f(\Sigma_2)$$

which violates the definition of g-convex of f. ■

The following two examples are g-convex with respect to the canonical Riemannian metric but not with respect to the Euclidean metric.

Proposition 12 *Let $y_i \in \mathbb{R}^d$ be nonzero vectors for $i = 1, \ldots, n$. The function*

$$f(X) = \log\left(\sum_{i=1}^{n} y_i^\top X y_i\right)$$

is g-convex with respect to the canonical Riemannian metric but is not g-convex with respect to the Euclidean metric.

Proof First we show that $f(X)$ is not g-convex with respect to the Euclidean metric. Observe that for any $y \in \mathbb{R}^d \setminus \{0\}$, $\theta \in (0,1)$ and $X, Y \in \mathbb{P}_d$, we have

$$\begin{aligned}\log\Big(y^\top\left(\theta X + (1-\theta)Y\right)y\Big) &= \log\Big(\theta y^\top X y + (1-\theta) y^\top Y y\Big) \\ &> \theta \log\Big(y^\top X y\Big) + (1-\theta)\log\Big(y^\top Y y\Big)\end{aligned}$$

where the strict inequality follows from the fact that $\log(\cdot)$ is a strict concave function on $(0, \infty)$.

To prove that $f(X)$ is g-convex with respect to the canonical Riemannian metric, we follow the proof from Lemma 1.20 (Wiesel and Zhang, 2015) and Lemma 3.1 (Zhang, 2016). To this end, let $X, Y \in \mathbb{P}_d$ and verify the midpoint convexity condition

$$f(X\sharp Y) \leq \frac{1}{2}f(X) + \frac{1}{2}f(Y)$$

where $\sharp$ denotes the geometric mean of X and Y. By simple algebra one can show that the condition above is equivalent to

$$\left(\sum_{i=1}^{n} y_i^T [X\sharp Y] y_i\right)^2 \leq \left(\sum_{i=1}^{n} y_i^T X y_i\right)\left(\sum_{i=1}^{n} y_i^T Y y_i\right). \tag{12}$$

For simplicity, we define

$$u_i := X^{\frac{1}{2}} y_i \qquad \text{and} \qquad v_i := \left(X^{-\frac{1}{2}} Y X^{-\frac{1}{2}}\right)^{\frac{1}{2}} X^{\frac{1}{2}} y_i.$$

Observe that by applying Cauchy-Scwartz twice we get

$$\begin{aligned}\left(\sum_{i=1}^{n} \mathbf{u}_i^T \mathbf{v}_i\right)^2 &= \left(\sum_{i=1}^{n} \left|\mathbf{u}_i^T \mathbf{v}_i\right|\right)^2 \\ &\leq \left(\sum_{i=1}^{n} \|\mathbf{u}_i\| \, \|\mathbf{v}_i\|\right)^2 \\ &\leq \left(\sum_{i=1}^{n} \|\mathbf{u}_i\|^2\right)\left(\sum_{i=1}^{n} \|\mathbf{v}_i\|^2\right).\end{aligned}$$

It suffices to check that

$$\left(\sum_{i=1}^{n} \mathbf{u}_i^T \mathbf{v}_i\right)^2 \leq \left(\sum_{i=1}^{n} \|\mathbf{u}_i\|^2\right)\left(\sum_{i=1}^{n} \|\mathbf{v}_i\|^2\right)$$

if and only if (12) holds. ■

Proposition 13 *The function $f(X) = \log\det X$ is g-convex (in fact, g-linear) with respect to the canonical metric but is g-concave with respect to the Euclidean metric.*

Proof To show that $f : \mathbb{P}_d \to \mathbb{R}_{++}$ is indeed g-concave with respect to the Euclidean metric we refer the reader to Section 3.1.5 (Boyd and Vandenberghe, 2004). Let $X, Y \in \mathbb{P}_d$ and $\gamma : [0,1] \to \mathbb{P}_d$ be the geodesic segment connecting $\gamma(0) = A$ to $\gamma(1) = B$. For $t \in [0,1]$

$$\begin{aligned}\log\det\left(\gamma(t)\right) &= \log\det\left(X^{1/2}(X^{-1/2}YX^{-1/2})^t X^{1/2}\right) \\ &= \log\left(\det(X)\det(X^{-1})^t\det(Y)^t\right) \\ &= \log\det(X) - t\log\det(X) + t\log\det(Y) \\ &= (1-t)\log\det(X) + t\log\det(Y).\end{aligned}$$

■

Finally, we show an example of a function that is g-convex with respect to both the Euclidean and canonical Riemannian metric. To this end, we need the following lemma.

Lemma 18 *[Theorem 7.6(a) (Horn and Johnson, 2013)] Let $A, B \in \mathbb{P}_d$ be two positive definite matrices. Then A and B are simultaneously diagonalizable by a congruence, i.e., there exists a nonsingular matrix $S \in \mathbb{R}^{n\times n}$ such that*

$$A = SIS^\top \qquad \textit{and} \qquad B = S\Lambda S^\top$$

where the main diagonal entries of Λ are the eigenvalues of the diagonal matrix $A^{-1}B$. In fact, one possible choice of S is $S = A^{\frac{1}{2}}U$ where U is any orthogonal matrix such that $A^{-\frac{1}{2}}BA^{-\frac{1}{2}} = U\Lambda U^\top$ is a spectral decomposition.

Proposition 14 *Fix $y \in \mathbb{R}^d \setminus \{0\}$. The function $f(X) = y^\top X y$ is g-convex with respect to both the Euclidean metric and the canonical Riemannian metric.*

Proof We can apply the *trace trick* to write

$$f(X) = y^\top X y = \operatorname{tr}\left(X y y^\top\right) = \operatorname{tr}(XY)$$

where $Y \overset{\text{def}}{=} yy^\top$. With respect to the Euclidean metric, we observe that $f(X)$ is a composition of g-linear functions and thus it is g-linear with respect to the Euclidean metric. That is, for all $\theta \in [0,1]$ and $X, Z \in \mathbb{P}_d$ we have

$$f(\theta X + (1-\theta) Y) = \operatorname{tr}\left((\theta X + (1-\theta) Z)\, Y\right) = \theta \operatorname{tr}(XY) + (1-\theta)\operatorname{tr}(ZY) = \theta f(X) + (1-\theta) f(Z).$$

Now we show $f(X)$ is g-convex with respect to the canonical Riemannian metric. Apply Lemma 18 to obtain

$$A = SIS^\top \qquad \text{and} \qquad B = S\Lambda S^\top$$

where we choose $S = A^{\frac{1}{2}} U$ where U is any orthogonal matrix such that $A^{-\frac{1}{2}} B A^{-\frac{1}{2}} = U \Lambda U^\top$ is a spectral decomposition. Then the geodesic that connects A to B is reduced as follows:

$$\begin{aligned} \gamma(t) &= A^{\frac{1}{2}} \left(A^{-\frac{1}{2}} B A^{-\frac{1}{2}}\right)^t A^{\frac{1}{2}} \\ &= A^{\frac{1}{2}} \left(U \Lambda U^\top\right)^t A^{\frac{1}{2}} \\ &= A^{\frac{1}{2}} U \Lambda^t U^\top A^{\frac{1}{2}}. \end{aligned}$$

Hence for $t \in [0,1]$ we have

$$\phi(\gamma(t)) = \left(y^\top A^{\frac{1}{2}} U\right) \Lambda^t \left(U^\top A^{\frac{1}{2}} y\right) = \tilde{y}^\top \Lambda^t \tilde{y}$$

where $\tilde{y} = U^\top A^{\frac{1}{2}} y$. Since U orthogonal and $A^{\frac{1}{2}} \in \mathbb{P}_d$ we have that $U^\top A^{\frac{1}{2}}$ is invertible and thus acts as a change-of-basis that diagonalizes the quadratic form $\phi(\gamma(t))$. In fact, the eigenvalues of such a diagonalization are precisely the generalized eigenvalues of the pair matrices (B, A) raised to the t-th power.

Also, we have

$$\begin{aligned}(1-t)\phi(A)+t\phi(B) &= y^\top\left((1-t)A+tB\right)y\\ &= y^\top\left((1-t)SS^\top+tS\Lambda S^\top\right)y\\ &= y^\top S\left((1-t)I+t\Lambda\right)S^\top y\\ &= \left(y^\top A^{\frac{1}{2}}U\right)\left((1-t)I+t\Lambda\right)\left(U^\top A^{\frac{1}{2}}y\right)\\ &= \tilde{y}^\top\left((1-t)I+t\Lambda\right)\tilde{y}.\end{aligned}$$

Finally, ϕ is geodesically convex if and only if

$$\phi(\gamma(t))=\tilde{y}^\top\Lambda^t\tilde{y}\leq\tilde{y}^\top\left((1-t)I+t\Lambda\right)\tilde{y}=(1-t)\phi(A)+t\phi(B)\qquad\forall t\in[0,1].$$

Since Λ^t and $(1-t)I+t\Lambda$ are both diagonal matrices we have the equivalent inequality

$$\Lambda^t\stackrel{\text{def}}{=}\mathbf{diag}(\lambda_1^t,\ldots,\lambda_n^t)\preceq(1-t)I+t\Lambda\qquad\forall t\in[0,1].$$

By the weighted AM-GM inequality, we indeed have

$$\lambda_i^t\leq(1-t)+t\lambda_i\qquad\forall i\in[n]\ \forall t\in[0,1].$$

Since $y\in\mathbb{R}^n$ was arbitrarily selected and we proved

$$\phi(\gamma(t))\leq(1-t)\phi(A)+t\phi(B)\qquad\forall t\in[0,1]$$

our desired result is proved. ■