

ManifoldsBase.jl – an interface for manifolds in Julia

Mateusz Baran² and Ronny Bergmann^{1, 2}

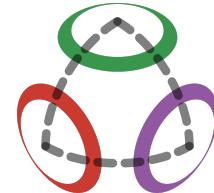
¹Department of Mathematical Sciences, Norwegian University of Science and Technology, Trondheim, Norway

²AGH University of Science and Technology, Kraków, Poland

ABSTRACT

We present an overview about the interface of the Julia package `ManifoldsBase.jl` to define and work with manifolds in Julia. The package is a central component of the `JuliaManifolds` ecosystem and is used by nearly all other packages in the ecosystem and several packages in the Julia community.

We discuss the main design ideas of the interface and present an overview of the features and functions provided in `ManifoldsBase.jl` 2.0.



Logo of `ManifoldsBase.jl`.

Keywords

Julia, Riemannian manifolds, Lie groups, differential geometry, numerical analysis, scientific computing

1. Introduction

In many scenarios one encounters data that does not lie in a Euclidean space. Informally phrased, for certain data, we are not allowed do just add or scale them. Examples are rotations, data on spheres, in hyperbolic spaces, like when working with general relativity, symmetric positive definite matrices, like when working with covariance matrices, or when the data is either bases of subspaces or subspaces themselves, i.e. data on the Stiefel or Grassmann manifold. The data of interest for this paper still bear enough “structure”, i.e. they live on smooth manifolds.

In a prior paper about `Manifolds.jl` [2], a partial presentation of the interface of `ManifoldsBase.jl` was given, when the interface was still in an early stage. A few fundamental design decisions have changed since then.

This paper presents the current state of the interface of `ManifoldsBase.jl` 2.0, since that state can be considered as having reached a very stable state. The paper is organized as follows: in Section ?? we introduce the mathematical background and notation. In Section 3 we present the main design principles and structure of the interface. Section 4 presents the interface components in detail. In Section 6 we present an example of defining a new manifold, and Section 6 presents an overview of packages using `ManifoldsBase.jl`.

2. Mathematical Background

This section introduces the mathematical background and notation used throughout this paper, for more details see, e.g. [5, 1, 4].

A *manifold* \mathcal{M} is a topological space¹ that locally resembles Euclidean space \mathbb{R}^n , $n \in \mathbb{N}$. To make this more precise, for every point $p \in \mathcal{M}$, there exists a neighborhood $\mathcal{U} \subseteq \mathcal{M}$ and a homeomorphism $\varphi: \mathcal{U} \rightarrow \mathbb{R}^n$. The dimension of the manifold is defined as the dimension of the Euclidean space \mathbb{R}^n . This can be interpreted as the number of degrees of freedom. The homeomorphism φ is called a *chart*, $\varphi(p)$ is called the *coordinates* of the point p in the chart φ . A collection of charts $\varphi_i: \mathcal{U}_i \rightarrow \mathbb{R}^n$, $i \in \mathcal{I}$ for some index set \mathcal{I} , that cover the manifold $\mathcal{M} = \bigcup_i \mathcal{U}_i$ is called an *atlas*. If we can “smoothly change charts”, i.e. the transition maps $\varphi_j \circ \varphi_i^{-1}$ are smooth², we say that the atlas \mathcal{A} is *smooth*. A manifold \mathcal{M} together with a smooth atlas is called a *smooth manifold*. Informally, a smooth manifold allows us to compute derivatives on the manifold by “looking at it through a chart”. The smooth atlas then allows to argue, that this derivative is independent of the chosen chart.

¹we assume that the spaces we consider here are second countable and Hausdorff

²while one could consider different kinds of smoothness, C^1, C^2, \dots , for simplicity one usually just assumes C^∞

3. Components and design principles of the interface

3.1 Main types

3.2 General function design and a scheme of 3 layers

3.3 A trait system to avoid code duplication.

4. The interface

4.1 Topological functions

4.2 Metric and connection related functions

4.3 ???

5. An example

5.1 Defining an own manifold

5.2 Adding a second metric to an existin manifold

5.3 Implementing a generic algorithm on a manifold

6. Where the interface is used

—Manifolds.jl
—Manopt.jl [3]
—ManifoldDiff.jl
—ManifoldDiffEq.jl
—GeometricKalman.jl
—ExponentialFamilyManifolds.jl (Mykola) - and a second one he built on that?
—ROMe.jl (?)

7. References

- [1] Pierre-Antoine Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press. doi:[10.1515/9781400830244](https://doi.org/10.1515/9781400830244).
- [2] 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), dec 2023. doi:[10.1145/3618296](https://doi.org/10.1145/3618296).
- [3] Ronny Bergmann. Manopt.jl: Optimization on manifolds in Julia. *Journal of Open Source Software*, 7(70):3866, 2022. doi:[10.21105/joss.03866](https://doi.org/10.21105/joss.03866).
- [4] Manfredo Perdigão do Carmo. *Riemannian Geometry*. Mathematics: Theory & Applications. Birkhäuser Boston, Inc., 1992.
- [5] John M. Lee. *Introduction to Smooth Manifolds*, volume 218 of *Graduate Texts in Mathematics*. Springer, New York. doi:[10.1007/978-0-387-21752-9](https://doi.org/10.1007/978-0-387-21752-9).