juliacon

Groups and smooth geometry using LieGroups.jl

Ronny Bergmann[1]

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

ABSTRACT

LieGroups.jl is a Julia package that provides on the one hand an interface to define Lie groups as well as the corresponding Lie algebra and Lie group actions. On the other hand, it offers a well-documented, performant, and well-tested library of existing Lie groups, their Algebra and corresponding group actions.
This paper presents the main features of the interfaces and how that is integrated within the `JuliaManifolds` ecosystem. We further present an overview on existing Lie groups implemented in `LieGroups.jl` as well as how to get started to use the package in practice.

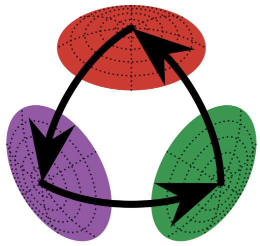
Keywords

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

1. Introduction

In many situations, one encounters data that does not reside in a vector space. We can hence not used standard linear algebra tools to work with such data. For example in robotics, the configuration space of a rigid body in three-dimensional space is given by the special Euclidean group $\mathrm{SE}(3)$, consisting of all translations and rotations. A subset of these is the space of rotations, given by the special orthogonal group $\mathrm{SO}(3)$, or more generally $\mathrm{SO}(n)$ in n-dimensional space.
These spaces are examples of Lie groups, formally defined as a smooth manifold equipped with a group structure. They have applications e.g. in physics, robotics, stochastic processes, information geometry, see e.g. [5, 6], but are also interesting from their mathematical viewpoint [8] and their numerical aspects, e.g. when solving differential equations on Lie groups [9].
The package LieGroups.jl[1] provides an easy access to both defining and using Lie groups within the Julia programming language [4] by defining an interface of Lie groups, as well as implementing a library of Lie groups, that can directly be used.
This paper provides an overview of the main features of `LieGroups.jl`. After introducing the necessary mathematical background and notation in Section 2, we present the interface in Section 3, split into the Lie group, the Lie algebra, and group actions. In Section 4, we present an overview of currently implemented Lie groups. Finally, in Section 5, we demonstrate how to get started and use `LieGroups.jl`.

[1]Available at juliamanifolds.github.io/LieGroups.jl/stable/, see also the zenodo archive [1].



Logo of `LieGroups.jl`.

2. Mathematical Background

The following notation and definitions follow the text book [8], for more details on Riemannian manifolds, see also [7].
We denote a Lie group by $\mathcal{G} = (\mathcal{M}, \cdot)$ where $\mathcal{M}$ is a smooth manifold and $\cdot$ is the group operation. A smooth manifold $\mathcal{M}$ is a topological space that is locally isomorphic to an Euclidean space $\mathbb{R}^n$ for some $n \in \mathbb{N}$, but globally may have a different topology. We call n the dimension of the manifold $\mathcal{M}$, denoted by $\dim(\mathcal{M}) = n$. As an example, take the 2-dimensional sphere $\mathbb{S}^2 = \{x \in \mathbb{R}^3 \mid \|x\| = 1\}$, which locally looks like $\mathbb{R}^2$, think of charts in an atlas, but globally it is not. Finally we denote the tangent space at a point $p \in \mathcal{M}$ by $T_p\mathcal{M}$. This can be thought of as all "velocities" (direction and speed) in which a curve can "pass through" a point. Formally it is set the equivalence classes of derivatives of smooth curves. Each tangent space $T_p\mathcal{M}$ is a n-dimensional vector space and we call the disjoint union of all tangent spaces $T\mathcal{M} = \dot{\bigsqcup}_{p\in\mathcal{M}} T_p\mathcal{M}$ the *tangent bundle* of $\mathcal{M}$.
As a group operation $\cdot\colon \mathcal{G} \times \mathcal{G} \to \mathcal{G}$ has to satisfy the group axioms associativity, existence of an identity element $e \in \mathcal{G}$, and existence of inverses $x^{-1} \in \mathcal{G}$ for all $x \in \mathcal{G}$. Furthermore the group operation $\cdot$ (on $\mathcal{G} \times \mathcal{G}$) and the inversion map $\iota\colon \mathcal{G} \to \mathcal{G}, x \mapsto x^{-1}$ have to be smooth maps. As an example, consider the special orthogonal group $\mathrm{SO}(n)$, consisting of all $n \times n$ orthogonal matrices, with determinant 1, i.e. for $p \in \mathrm{SO}(n)$, we have $p^\mathrm{T} p = I$ and $\det(p) = 1$ together with the group operation $\cdot$ given by matrix multiplication. For $n = 2$ these are rotations in the plane, hence each operation can be identified with an angle $\alpha \in [-\pi, \pi)$, or in other words the circle. The identity element is given by the identity matrix I (or the angle $\alpha = 0$) and the inverse of a rotation matrix is given by its transpose (or an angle $-\alpha$).
The tangent space at the identity element $e \in \mathcal{G}$, denoted by $\mathfrak{g} = T_e\mathcal{G}$, plays a special role and is called the *Lie algebra* of the Lie group $\mathcal{G}$. The reason is that to represent arbitrary tangent vectors $X \in T_g\mathcal{G}$ at a point $g \in \mathcal{G}$, since we can use the group operation: we denote by $\lambda_g(h) = g \cdot h$ the left multiplication with $g, h \in \mathcal{G}$. Then, using the differential (or pushforward)

Table 1. : Implemented Lie groups in `LieGroups.jl` as of version 0.1.6

Group	**Symbol**	comment/code
`CircleGroup()`	$\mathbb{S}^1$	3 representations
`GeneralLinearGroup(n, F)`	$\mathrm{GL}(n,\mathbb{F})$	$\mathbb{F} \in \{\mathbb{R},\mathbb{C}\}$
`HeisenbergGroup(n)`	$\mathrm{H}(n)$	
`OrthogonalGroup(n)`	$\mathrm{O}(n)$	
`PowerLieGroup(G, n)`	$\mathcal{G}^n$	`G^n`
`ProductLieGroup(G1, G2,...)`	$\mathcal{G}_1 \times \mathcal{G}_2 \times \ldots$	`G1` $\times$ `G2` $\times \ldots$
Semidirect product group	$\mathcal{G}_1 \ltimes \mathcal{G}_2$	`G1` $\ltimes$ `G2`
	$\mathcal{G}_1 \rtimes \mathcal{G}_2$	`G1` $\rtimes$ `G2`
`SpecialEuclideanGroup(n)`	$\mathrm{SE}(n)$	
`SpecialGalileanGroup(n)`	$\mathrm{SGal}(n)$	
`SpecialLinearGroup(n, F)`	$\mathrm{SL}(n,\mathbb{F})$	$\mathbb{F} \in \{\mathbb{R},\mathbb{C}\}$
`SpecialOrthogonalGroup(n)`	$\mathrm{SO}(n)$	
`SpecialUnitaryGroup(n)`	$\mathrm{SU}(n)$	
`SymplecticGroup(n)`	$\mathrm{Sp}(2n)$	
`TranslationGroup(n; field=`$\mathbb{F}$`)`	$\mathbb{F}^n$	$\mathbb{F} \in \{\mathbb{R},\mathbb{C},\mathbb{H}\}$
`UnitaryGroup(n)`	$\mathrm{U}(n)$	

$D\lambda_g(h)\colon T_g\mathcal{G} \to T_h\mathcal{G}$, we can generate a so-called *left-invariant vector field* $\mathcal{X}(g) := D\lambda_g(e)[X]$ which is uniquely determined by the choice of $X \in \mathfrak{g}$. Hence we can identify tangent vectors $\mathcal{X}(g) \in T_g\mathcal{G}$ at arbitrary points $g \in \mathcal{G}$ with X from the Lie algebra $\mathfrak{g}$.
Finally, a *group action* of a Lie group $\mathcal{G}$ on a smooth manifold $\mathcal{M}$ is a smooth map $\rho\colon \mathcal{G} \times \mathcal{M} \to \mathcal{M}$ such that for all $g, h \in \mathcal{G}$ and $p \in \mathcal{M}$ it holds that $\rho(e,p) = p$ and $\rho(g, \rho(h,p)) = \rho(g \cdot h, p)$. Informally a group action describes how elements of the Lie group $\mathcal{G}$ "act on" points on the manifold $\mathcal{M}$. As an example, think of the special orthogonal group $\mathrm{SO}(3)$ acting on points on Euclidean space $\mathbb{R}^3$ "moving" them somewhere by rotating them around the origin. The same action can also be applied to points from the sphere $\mathbb{S}^2$.

3. The interface

Since a Lie group $\mathcal{G}$ consists of two main components, the smooth manifold $\mathcal{M}$ and the group operation $\cdot$, we can reuse existing functionality from the existing interface for manifolds provided by `ManifoldsBase.jl`, and later concrete manifolds provided by `Manifolds.jl` [2]. This is done in a transparent way, i.e. the `AbstractLieGroup` itself is a subtype of `AbstractManifold` from `ManifoldsBase.jl` and can hence also be used in all existing places, as for example optimization on manifolds provided by `Manopt.jl` [3].

3.1 Lie groups

TODO 1-1.5 pages on the interfaces and main functions provided for Lie groups and group operations, Lie algebra, and group actions. Maybe take over some story parts from the talk.
[8]

3.2 Lie algebras

3.3 Group actions

4. Implemented Lie groups

5. An example how to use `LieGroups.jl`

6. References

[1] Seth Axen, Mateusz Baran, Ronny Bergmann, Yueh-Hua Tu, and Olivier Verdier. LieGroups.jl, 2025. doi:10.5281/zenodo.15343362.

[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.

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

[4] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. Julia: A fresh approach to numerical computing. *SIAM review*, 59(1):65–98, 2017. doi:10.1137/141000671.

[5] Gregory S. Chirikjian. *Stochastic Models, Information Theory, and Lie Groups, Volume 1: Classical Results and Geometric Methods*. 2009. doi:10.1007/978-0-8176-4803-9.

[6] Gregory S. Chirikjian. *Stochastic Models, Information Theory, and Lie Groups, Volume 2: Analytic Methods and Modern Applications*. 2012. doi:10.1007/978-0-8176-4944-9.

[7] Manfredo Perdigão do Carmo. *Riemannian Geometry*. Mathematics: Theory & Applications. Birkhäuser Boston, Inc., 1992.

[8] Joachim Hilgert and Karl-Hermann Neeb. *Structure and Geometry of Lie Groups*. Springer Monographs in Mathematics, 2012. doi:10.1007/978-0-387-84794-8.

[9] Hans Munthe-Kaas. Runge-kutta methods on lie groups. *BIT Numerical Mathematics*, 38(1):92–111, 1998. doi:10.1007/bf02510919.