The Journal of Open Source Software

GeometricIntegrators.jl: Geometric Numerical Integration in Julia

Michael Kraus[1, 2]

1 Max-Planck-Institut für Plasmaphysik, Boltzmannstraße 2, 85748 Garching, Germany **2** Technische Universität München, Zentrum Mathematik, Boltzmannstraße 3, 85748 Garching, Germany

DOI: 10.21105/joss.02954

Software

- Review
- Repository
- Archive

Editor: Viviane Pons

Reviewers:

- @ChrisRackauckas
- @ranocha

Submitted: 21 December 2020
Published: 14 January 2021

Summary

GeometricIntegrators.jl is a library of geometric numerical algorithms for the solution of differential equations in Julia. It provides a unified interface for many different algorithms and various types of equations, such as ordinary differential equations, stochastic differential equations and differential algebraic equations, in particular in Lagrangian (variational) and Hamiltonian (symplectic) form. It aims at providing a comprehensive collection of geometric or structure-preserving algorithms, which can be used either interactively or as computational core in other codes. The library provides both, a high-level interface that requires only very few lines of code to solve an actual problem, and a lean low-level interface that allows for straightforward integration into application codes via the exchange of very small data structures. In both cases, the library leaves maximum control to the user, e.g., with respect to the choice of numerical methods and the setup of linear and nonlinear solvers.

Statement of need

Differential equations are ubiquitous in science and engineering. Many equations possess geometric features or abstract mathematical structures that need to be preserved in the discretisation in order to obtain reliable simulation results, especially for nonlinear problems and long-time simulations. The preservation of such properties improves stability, bounds global error growth and reduces numerical artefacts (Blanes & Casas, 2016; Hairer et al., 2006; Leimkuhler & Reich, 2004; Sanz-Serna & Calvo, 1994). Robust, performant and structure-preserving solvers for different types of differential equations are thus needed across many disciplines. *GeometricIntegrators.jl* provides such solvers and makes them available for both direct use as well as integration into other codes. Furthermore, the implemented algorithms can also be used within the *DifferentialEquations.jl* ecosystem (Rackauckas & Nie, 2017), which is the defacto standard differential equation solver for the Julia programming language (Bezanson et al., 2017).

GeometricIntegrators.jl provides a comprehensive library of existing geometric integration as well as non-geometric algorithms, such as explicit, implicit, partitioned and stochastic Runge-Kutta methods, SPARK methods, splitting methods, symplectic methods and variational integrators. Most integrators are implemented in an abstract way that allows for the flexible choice of tableaus, approximation spaces, basis functions, quadrature rules, and thus order of convergence. *GeometricIntegrators.jl* also serves as a testbed for the development and analysis of novel algorithms (Kraus, 2017, 2020; Kraus & Tyranowski, 2020). Due to the modular structure and the use of the multiple dispatch paradigm, the library can easily be extended, e.g., towards new algorithms or new types of equations. The library is designed to minimize

overhead and maximize performance in order to be able to perform simulations with millions or even billions of time steps, facilitating the study of the long-time behaviour of both numerical algorithms and dynamical systems.

Other Software

A Julia package closely related to *GeometricIntegrators.jl* is *DifferentialEquations.jl* (Rackauckas & Nie, 2017). Althouh the two libraries provide similar functionality, their scope is rather different. While *DifferentialEquations.jl* provides a feature-rich ecosystem for the solution of differential equations, the focus of *GeometricIntegrators.jl* is on algorithms. In fact, *GeometricIntegrators.jl* can be used as backend for *DifferentialEquations.jl* via GeometricIntegratorsDiffEq.jl.

Quality control and contributions

GeometricIntegrators.jl uses continuous integration testing with all supported versions of Julia on macOS, Linux and Windows via GitHub CI. The tests cover most of the library, checking for the correct functioning of all submodules and convergence of the implemented algorithms. The tests can also be run locally in the Julia's REPL by the command `]test GeometricIntegrators`.

Support and submission of contributions to the library are handled through the GitHub repository via issues or by pull requests.

Acknowledgements

This work has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 708124. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

Tomasz M. Tyranowski provided the initial implementation of stochastic integrators.

References

Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2017). Julia: A Fresh Approach to Numerical Computing. *Siam Review*, *59*(1), 65–98. https://doi.org/10.1137/141000671

Blanes, S., & Casas, F. (2016). *A concise introduction to geometric numerical integration*. CRC Press. https://doi.org/10.1201/b21563

Hairer, E., Lubich, C., & Wanner, G. (2006). *Geometric numerical integration*. Springer. https://doi.org/10.1007/3-540-30666-8

Kraus, M. (2017). *Projected variational integrators for degenerate Lagrangian systems*. http://arxiv.org/abs/1708.07356

Kraus, M. (2020). *SPARK methods for degenerate Lagrangian systems*.

Kraus, M., & Tyranowski, T. M. (2020). Variational integrators for stochastic dissipative Hamiltonian systems. *IMA Journal of Numerical Analysis*. https://doi.org/10.1093/imanum/draa022

Leimkuhler, B., & Reich, S. (2004). *Simulating Hamiltonian dynamics*. Cambridge University Press. https://doi.org/10.1017/CBO9780511614118

Rackauckas, C., & Nie, Q. (2017). DifferentialEquations.jl - a performant and feature-rich ecosystem for solving differential equations in Julia. *Journal of Open Research Software*, *5*, 15. https://doi.org/10.5334/jors.151

Sanz-Serna, J. M., & Calvo, M. P. (1994). *Numerical Hamiltonian problems*. Chapman; Hall.