The Journal of Open Source Software

Optim: A mathematical optimization package for Julia

Patrick K Mogensen[1] **and Asbjørn N Riseth**[2]

1 University of Copenhagen **2** University of Oxford

DOI: 10.21105/joss.00615

Software

- Review
- Repository
- Archive

Submitted: 09 March 2018
Published: 04 April 2018

Summary

Optim provides a range of optimization capabilities written in the Julia programming language (Bezanson et al. 2017). Our aim is to enable researchers, users, and other Julia packages to solve optimization problems without writing such algorithms themselves. The package supports optimization on manifolds, functions of complex numbers, and input types such as arbitrary precision vectors and matrices. We have implemented routines for derivative free, first-order, and second-order optimization methods. The user can provide derivatives themselves, or request that they are calculated using automatic differentiation or finite difference methods. The main focus of the package has currently been on unconstrained optimization, however, box-constrained optimization is supported, and a more comprehensive support for constraints is underway.

Similar to Optim, the C library NLopt (Johnson 2008) contains a collection of nonlinear optimization routines. In Python, scipy.optimize supports many of the same algorithms as Optim does, and Pymanopt (Townsend, Niklas, and Weichwald 2016) is a toolbox for manifold optimization. Within the Julia community, the packages BlackBoxOptim.jl and Optimize.jl provide optimization capabilities focusing on derivative-free and large-scale smooth problems respectively. The packages Convex.jl and JuMP.jl (Dunning, Huchette, and Lubin 2017) define modelling languages for which users can formulate optimization problems. In contrast to the previously mentioned optimization codes, Convex and JuMP work as abstraction layers between the user and solvers from a other packages.

Optimization routines

As of version 0.14, the following optimization routines are available.

- Second-order methods
  - Newton
  - Newton with trust region
  - Hessian-vector with trust region
- First-order methods
  - BFGS
  - L-BFGS (with linear preconditioning)
  - Conjugate gradient (with linear preconditioning)
  - Gradient descent (with linear preconditioning)
- Acceleration methods
  - Nonlinear GMRES
  - Objective acceleration
- Derivative-free methods
  - Nelder–Mead
  - Simulated annealing
  - Particle swarm
- Interval bound univariate methods

- Brent's method
- Golden-section search

The derivative based methods use line searches to assist convergence. Multiple line search algorithms are available, including interpolating backtracking and methods that aim to satisfy the Wolfe conditions.

Usage in research and industry

The optimization routines in this package have been used in both industrial and academic contexts. For example, parts of the internal work in the company Ternary Intelligence Inc. (Paramonov 2017) rely on the package. Notably, an upcoming book on optimization (Kochenderfer and Wheeler Forthcoming, 2018) uses Optim for its examples. Optim has been used for a wide range of applications in academic research, including optimal control (Riseth, Dewynne, and Farmer 2017; Riseth 2017a), parameter estimation (Riseth and Taylor-King 2017; Rackauckas and Nie 2017; and Dony, He, and Stumpf 2018), quantum physics (Damle, Levitt, and Lin 2018), crystalline modelling (Chen and Ortner 2017; Braun, Buze, and Ortner 2017), and the large-scale astronomical cataloguing project Celeste (Regier et al. 2015; Regier et al. 2016). A new acceleration scheme for optimization (Riseth 2017b), and a preconditioning scheme for geometry optimisation (Packwood et al. 2016) have also been tested within the Optim framework.

Acknowledgements

John Myles White initiated the development of the Optim code base in 2012. We owe much to him and Timothy Holy for creating a solid package for optimization that the rest of the Julia community could further improve upon. We would also like to thank everyone who has contributed with code and discussions to help improve the package. In particular, Antoine Levitt, Christoph Ortner, and Chris Rackauckas have been helpful in providing suggestions and code contributions towards more modularity and greater support for non-trivial inputs and decision spaces.

Funding

Asbjørn Riseth is partially supported by the EPSRC research grant EP/L015803/1.

References

Bezanson, Jeff, Alan Edelman, Stefan Karpinski, and Viral B Shah. 2017. "Julia: A Fresh Approach to Numerical Computing." *SIAM Review* 59 (1). SIAM:65–98. https://doi.org/10.1137/141000671.

Braun, Julian, Maciej Buze, and Christoph Ortner. 2017. "The Effect of Crystal Symmetries on the Locality of Screw Dislocation Cores." *arXiv Preprint arXiv:1710.07708*.

Chen, Huajie, and Christoph Ortner. 2017. "QM/Mm Methods for Crystalline Defects. Part 2: Consistent Energy and Force-Mixing." *Multiscale Modeling & Simulation* 15 (1). SIAM:184–214. https://doi.org/10.1137/15M1041250.

Damle, A., A. Levitt, and L. Lin. 2018. "Variational formulation for Wannier functions with entangled band structure." *ArXiv E-Prints*, January.

Dony, Leander, Fei He, and Michael Stumpf. 2018. "Parametric and Non-Parametric Gradient Matching for Network Inference." *bioRxiv*. Cold Spring Harbor Laboratory. https://doi.org/10.1101/254003.

Dunning, Iain, Joey Huchette, and Miles Lubin. 2017. "JuMP: A Modeling Language for Mathematical Optimization." *SIAM Review* 59 (2). SIAM:295–320. https://doi.org/10.1137/15M1020575.

Johnson, Steven G. 2008. "The Nlopt Nonlinear-Optimization Package." http://ab-initio.mit.edu/nlopt.

Kochenderfer, Mykel J., and Tim A. Wheeler. Forthcoming, 2018. *Algorithms for Optimization*. MIT Press.

Packwood, David, James Kermode, Letif Mones, Noam Bernstein, John Woolley, Nicholas Gould, Christoph Ortner, and Gábor Csányi. 2016. "A Universal Preconditioner for Simulating Condensed Phase Materials." *The Journal of Chemical Physics* 144 (16). AIP Publishing:164109. https://doi.org/10.1063/1.4947024.

Paramonov, Pavel. 2017. Private communication, Chief Science and Technology Officer; Ternary Intelligence Inc.

Rackauckas, Christopher, and Qing Nie. 2017. "Differentialequations.jl – a Performant and Feature-Rich Ecosystem for Solving Differential Equations in Julia." *Journal of Open Research Software* 5 (1). Ubiquity Press. https://doi.org/10.5334/jors.151.

Regier, J., K. Pamnany, R. Giordano, R. Thomas, D. Schlegel, J. McAuliffe, and Prabhat. 2016. "Learning an Astronomical Catalog of the Visible Universe through Scalable Bayesian Inference." *ArXiv E-Prints*, November.

Regier, Jeffrey, Andrew Miller, Jon McAuliffe, Ryan Adams, Matt Hoffman, Dustin Lang, David Schlegel, and Mr Prabhat. 2015. "Celeste: Variational Inference for a Generative Model of Astronomical Images." In *International Conference on Machine Learning*, 2095–2103.

Riseth, A. N. 2017a. "Dynamic pricing in retail with diffusion process demand." *ArXiv E-Prints*, September.

———. 2017b. "Objective acceleration for unconstrained optimization." *ArXiv E-Prints*, October.

Riseth, A. N., and J. P. Taylor-King. 2017. "Operator Fitting for Parameter Estimation of Stochastic Differential Equations." *ArXiv E-Prints*, September.

Riseth, A. N., J. N. Dewynne, and C. L. Farmer. 2017. "A comparison of control strategies applied to a pricing problem in retail." *ArXiv E-Prints*, October.

Townsend, James, Koep Niklas, and Sebastian Weichwald. 2016. "Pymanopt: A Python Toolbox for Optimization on Manifolds Using Automatic Differentiation." *Journal of Machine Learning Research* 17 (137):1–5. http://jmlr.org/papers/v17/16-177.html.