

RegularizedOptimization.jl: A Julia framework for regularized and nonsmooth optimization

Y. Diouane, M. Gollier, M. L. Habiboullah, D. Orban

G-2025-75

November 2025

La collection *Les Cahiers du GERAD* est constituée des travaux de recherche menés par nos membres. La plupart de ces documents de travail a été soumis à des revues avec comité de révision. Lorsqu'un document est accepté et publié, le pdf original est retiré si c'est nécessaire et un lien vers l'article publié est ajouté.

Citation suggérée : Y. Diouane, M. Gollier, M. L. Habiboullah, D. Orban (Novembre 2025). *RegularizedOptimization.jl: A Julia framework for regularized and nonsmooth optimization*, Rapport technique, Les Cahiers du GERAD G-2025-75, GERAD, HEC Montréal, Canada.

Avant de citer ce rapport technique, veuillez visiter notre site Web (<https://www.gerad.ca/fr/papers/G-2025-75>) afin de mettre à jour vos données de référence, s'il a été publié dans une revue scientifique.

The series *Les Cahiers du GERAD* consists of working papers carried out by our members. Most of these pre-prints have been submitted to peer-reviewed journals. When accepted and published, if necessary, the original pdf is removed and a link to the published article is added.

Suggested citation: Y. Diouane, M. Gollier, M. L. Habiboullah, D. Orban (November 2025). *RegularizedOptimization.jl: A Julia framework for regularized and nonsmooth optimization*, Technical report, Les Cahiers du GERAD G-2025-75, GERAD, HEC Montréal, Canada.

Before citing this technical report, please visit our website (<https://www.gerad.ca/en/papers/G-2025-75>) to update your reference data, if it has been published in a scientific journal.

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2025
– Bibliothèque et Archives Canada, 2025

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Legal deposit – Bibliothèque et Archives nationales du Québec, 2025
– Library and Archives Canada, 2025

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Télec. : 514 340-5665
info@gerad.ca
www.gerad.ca

RegularizedOptimization.jl: A Julia framework for regularized and nonsmooth optimization

Youssef Diouane

Maxence Gollier

Mohamed Laghdaf Habiboullah

Dominique Orban

GERAD and Department of Mathematics and Industrial Engineering, Polytechnique Montréal, Montréal (Qc), Canada, H3T 2A7

youssef.diouane@polymtl.ca

maxence-2.gollier@polymtl.ca

mohamed-laghdaf-2.habiboullah@polymtl.ca

dominique.orban@polymtl.ca

November 2025

Les Cahiers du GERAD

G–2025–75

Copyright © 2025 Diouane, Gollier, Habiboullah, Orban

Les textes publiés dans la série des rapports de recherche *Les Cahiers du GERAD* n'engagent que la responsabilité de leurs auteurs. Les auteurs conservent leur droit d'auteur et leurs droits moraux sur leurs publications et les utilisateurs s'engagent à reconnaître et respecter les exigences légales associées à ces droits. Ainsi, les utilisateurs:

- Peuvent télécharger et imprimer une copie de toute publication du portail public aux fins d'étude ou de recherche privée;
- Ne peuvent pas distribuer le matériel ou l'utiliser pour une activité à but lucratif ou pour un gain commercial;
- Peuvent distribuer gratuitement l'URL identifiant la publication.

Si vous pensez que ce document enfreint le droit d'auteur, contactez- nous en fournissant des détails. Nous supprimerons immédiatement l'accès au travail et enquêterons sur votre demande.

The authors are exclusively responsible for the content of their research papers published in the series *Les Cahiers du GERAD*. Copyright and moral rights for the publications are retained by the authors and the users must commit themselves to recognize and abide the legal requirements associated with these rights. Thus, users:

- May download and print one copy of any publication from the public portal for the purpose of private study or research;
- May not further distribute the material or use it for any profit-making activity or commercial gain;
- May freely distribute the URL identifying the publication.

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Abstract : This paper presents RegularizedOptimization.jl, a Julia package that implements families of nonsmooth optimization methods.

Keywords: Julia; nonsmooth optimization; nonconvex optimization; regularization methods; trust-region methods

Résumé : Cet article présente RegularizedOptimization.jl, un module Julia qui implémente des familles de méthodes d'optimisation non lisse.

Mots clés: Julia; optimisation non lisse; optimisation non convexe; méthodes de régularisation; méthodes de région de confiance

Acknowledgements: The authors would like to thank A. De Marchi for the Augmented Lagrangian solver. M. L. Habiboullah is supported by an excellence FRQNT grant. Y. Diouane, M. Gollier and D. Orban are partially supported by an NSERC Discovery Grant.

1 Summary

[RegularizedOptimization.jl](#) is a Julia package that implements families of quadratic regularization and trust-region methods for solving the nonsmooth optimization problem

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x) + h(x) \quad \text{subject to} \quad c(x) = 0, \quad (1)$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and $c : \mathbb{R}^n \rightarrow \mathbb{R}^m$ are continuously differentiable, and $h : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ is lower semi-continuous. The nonsmooth objective h can be a *regularizer*, such as a sparsity-inducing penalty, model simple constraints, such as x belonging to a simple convex set, or be a combination of both. All f , h and c can be nonconvex. [RegularizedOptimization.jl](#) provides a modular and extensible framework for solving (1), and developing novel solvers. Currently, the following solvers are implemented:

- **Trust-region solvers TR and TRDH** (Aravkin et al., 2022; Leconte and Orban, 2025)
- **Quadratic regularization solvers R2, R2DH and R2N** (Diouane et al., 2024b; Aravkin et al., 2022)
- **Levenberg-Marquardt solvers LM and LMTR** (Aravkin et al., 2024) used when f is a least-squares residual.
- **Augmented Lagrangian solver AL** (De Marchi et al., 2023).

All solvers rely on first derivatives of f and c , and optionally on their second derivatives in the form of Hessian-vector products. If second derivatives are not available, quasi-Newton approximations can be used. In addition, the proximal mapping of the nonsmooth part h , or adequate models thereof, must be evaluated. At each iteration, a step is computed by solving a subproblem of the form (1) inexactly, in which f , h , and c are replaced with appropriate models about the current iterate. The solvers R2, R2DH and TRDH are particularly well suited to solve the subproblems, though they are general enough to solve (1). All solvers are implemented in place, so re-solves incur no allocations. To illustrate our claim of extensibility, a first version of the AL solver was implemented by an external contributor. Furthermore, a nonsmooth penalty approach, described in (Diouane et al., 2024a) is currently being developed, that relies on the library to efficiently solve the subproblems.

2 Statement of need

2.1 Model-based framework for nonsmooth methods

In Julia, (1) can be solved using [ProximalAlgorithms.jl](#), which implements splitting schemes and line-search-based methods (Stella et al., 2017; Themelis et al., 2018). Among others, the **PANOC** (Stella et al., 2017) solver takes a step along a direction d , which depends on the L-BFGS quasi-Newton approximation of f , followed by proximal steps on h .

By contrast, [RegularizedOptimization.jl](#) focuses on model-based trust-region and quadratic regularization methods, which typically require fewer evaluations of f and its gradient than first-order line search methods, at the expense of more evaluations of proximal operators (Aravkin et al., 2022). However, each proximal computation is inexpensive for numerous commonly used choices of h , such as separable penalties and bound constraints, so that the overall approach is efficient for large-scale problems.

[RegularizedOptimization.jl](#) provides an API to formulate optimization problems and apply different solvers. It integrates seamlessly with the [JuliaSmoothOptimizers](#) (Migot et al., 2021) ecosystem.

The smooth objective f can be defined via [NLPModels.jl](#) (Orban et al., 2020), which provides a standardized Julia API for representing nonlinear programming (NLP) problems. The nonsmooth term h can be modeled using [ProximalOperators.jl](#).

Given f and h , the companion package [RegularizedProblems.jl](#) provides a way to pair them into a *Regularized Nonlinear Programming Model*

```
reg_nlp = RegularizedNLPModel(f, h)
```

They can also be paired into a *Regularized Nonlinear Least-Squares Model* if $f(x) = \frac{1}{2}\|F(x)\|^2$ for some residual $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$, in the case of the **LM** and **LMTR** solvers.

```
reg_nls = RegularizedNLSModel(F, h)
```

[RegularizedProblems.jl](#) also provides a set of instances commonly used in data science and in non-smooth optimization, where several choices of f can be paired with various regularizers. This design makes for a convenient source of problem instances for benchmarking the solvers in [RegularizedOptimization.jl](#).

2.2 Support for both exact and approximate Hessian

In contrast with [ProximalAlgorithms.jl](#), [RegularizedOptimization.jl](#), methods such as **R2N** and **TR** methods support exact Hessians as well as several Hessian approximations of f . Hessian-vector products $v \mapsto Hv$ can be obtained via automatic differentiation through [ADNLPModels.jl](#) or implemented manually. Limited-memory and diagonal quasi-Newton approximations can be selected from [LinearOperators.jl](#). This design allows solvers to exploit second-order information without explicitly forming dense or sparse Hessians, which is often expensive in time and memory, particularly at large scale.

3 Example

We illustrate the capabilities of [RegularizedOptimization.jl](#) on a Support Vector Machine (SVM) model with a $\ell_{1/2}^{1/2}$ penalty for image classification (Aravkin et al., 2024).

Below is a condensed example showing how to define and solve the problem, and perform a solve followed by a re-solve:

```
using LinearAlgebra, Random, ProximalOperators
using NLPModels, RegularizedProblems, RegularizedOptimization
using MLDatasets

Random.seed!(1234)
model, nls_model, _ = RegularizedProblems.svm_train_model() # Build SVM model
f = LSR1Model(model) # L-SR1 Hessian approximation
λ = 1.0 # Regularization parameter
h = RootNormLhalf(λ) # Nonsmooth term
reg_nlp = RegularizedNLPModel(f, h) # Regularized problem
solver = R2NSolver(reg_nlp) # Choose solver
stats = RegularizedExecutionStats(reg_nlp)
solve!(solver, reg_nlp, stats; atol=1e-4, rtol=1e-4, verbose=1, sub_kwargs=(max_iter=200,))
solve!(solver, reg_nlp, stats; atol=1e-5, rtol=1e-5, verbose=1, sub_kwargs=(max_iter=200,))
```

3.1 Numerical results

We compare **TR**, **R2N**, **LM** and **LMTR** from our library on the SVM problem.

The table reports the convergence status of each solver, the number of evaluations of f , the number of evaluations of ∇f , the number of proximal operator evaluations, the elapsed time and the final ob-

jective value. For TR and R2N, we use limited-memory SR1 Hessian approximations. The subproblem solver is **R2**.

Method	Status	$t(s)$	$\#f$	$\#\nabla f$	$\#prox$	Objective
TR	first_order	3.9349	347	291	4037	179.837
R2N	first_order	1.9511	185	101	27932	192.493
LM	first_order	19.7826	6	2876	1001	201.186
LMTR	first_order	12.4967	11	1614	432	188.274

For the **LM** and **LMTR** solvers, $\#\nabla f$ counts the number of Jacobian–vector and adjoint–Jacobian–vector products.

All methods successfully reduced the optimality measure below the specified tolerance of 10^{-4} , and thus converged to an approximate first-order stationary point. Note that the final objective values differ due to the nonconvexity of the problem.

R2N is the fastest in terms of time and number of gradient evaluations. However, it requires more proximal evaluations, but these are inexpensive. **LMTR** and **LM** require the fewest function evaluations, but incur many Jacobian–vector products, and are the slowest in terms of time.

Ongoing research aims to reduce the number of proximal evaluations.

References

- A. Y. Aravkin, R. Baraldi, and D. Orban. A proximal quasi-Newton trust-region method for nonsmooth regularized optimization. *SIAM J. Optim.*, 32(2):900–929, 2022. DOI: [10.1137/21M1409536](https://doi.org/10.1137/21M1409536).
- A. Y. Aravkin, R. Baraldi, and D. Orban. A Levenberg–Marquardt method for nonsmooth regularized least squares. *SIAM J. Sci. Comput.*, 46(4):A2557–A2581, 2024. DOI: [10.1137/22M1538971](https://doi.org/10.1137/22M1538971).
- A. De Marchi, X. Jia, C. Kanzow, and P. Mehlitz. Constrained composite optimization and augmented Lagrangian methods. *Math. Program.*, 201(1):863–896, 9 2023. DOI: [10.1007/s10107-022-01922-4](https://doi.org/10.1007/s10107-022-01922-4).
- Y. Diouane, M. Gollier, and D. Orban. A nonsmooth exact penalty method for equality-constrained optimization: complexity and implementation. *Cahier G-2024-65*, GERAD, Montréal, Canada, 2024a.
- Y. Diouane, M. L. Habiboullah, and D. Orban. A proximal modified quasi-Newton method for nonsmooth regularized optimization. *Cahier G-2024-64*, GERAD, Montréal, Canada, 2024b. URL <https://www.gerad.ca/fr/papers/G-2024-64>.
- G. Leconte and D. Orban. The indefinite proximal gradient method. *Comput. Optim. Appl.*, 91(2):861–903, 2025. DOI: [10.1007/s10589-024-00604-5](https://doi.org/10.1007/s10589-024-00604-5).
- T. Migot, D. Orban, and A. S. Siqueira. The JuliaSmoothOptimizers ecosystem for linear and nonlinear optimization, 2021. URL <https://juliasmoothoptimizers.github.io/>.
- D. Orban, A. S. Siqueira, and contributors. *NLPModels.jl: Data structures for optimization models*, July 2020. URL <https://github.com/JuliaSmoothOptimizers/NLPModels.jl>.
- L. Stella, A. Themelis, P. Sopasakis, and P. Patrinos. A simple and efficient algorithm for nonlinear model predictive control. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 1939–1944, 2017. DOI: [10.1109/CDC.2017.8263933](https://doi.org/10.1109/CDC.2017.8263933).
- A. Themelis, L. Stella, and P. Patrinos. Forward-backward envelope for the sum of two nonconvex functions: Further properties and nonmonotone line search algorithms. *SIAM J. Optim.*, 28(3):2274–2303, 2018. DOI: [10.1137/16M1080240](https://doi.org/10.1137/16M1080240).