The Journal of Open Source Software

MultiGridBarrier.jl: quasi-optimal solvers for convex variational problems

Sébastien Loisel [1]

1 Heriot-Watt University, Edinburgh, United Kingdom

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

`MultiGridBarrier.jl` is a Julia (Bezanson et al., 2017) package for solving convex variational problems in function spaces. These are the nonlinear partial differential equations (PDEs) and boundary-value problems that arise from minimizing a convex functional. Representative examples include the p-Laplacian for any $p \in [1, \infty]$, total-variation problems, and obstacle problems. The most useful of these are *nonsmooth*: the energy is convex but not differentiable (for example $p = 1$, or total variation), a regime in which Newton-type solvers applied naively either fail or require a number of iterations that grows rapidly with the mesh resolution.

The package implements the **multigrid barrier method** (Loisel, 2020, 2026b, 2026e), which couples an interior-point (barrier) method with a multigrid hierarchy. For the problem classes covered by the supporting theory the method is *quasi-optimal*: the number of interior-point/Newton iterations grows only mildly with the number of degrees of freedom n. For instance, this count is $O(\sqrt{n} \log n)$ for the p-Laplacian (Loisel, 2020), and polylogarithmic in the analytic, spectral setting (Loisel, 2026e).

`MultiGridBarrier.jl` provides finite-element discretizations in one, two, and three dimensions (simplicial P_1/P_2 elements and tensor-product Q_k elements), as well as Chebyshev spectral discretizations, all with isoparametric element maps. It builds an algebraic-multigrid hierarchy automatically, supports user-specified mesh connectivity (enabling slit domains, branch cuts, and glued manifolds), solves time-dependent problems, and offers optional GPU acceleration through CUDA. A typical solve is three lines:

```julia
using MultiGridBarrier
geom = fem2d_P2()
sol  = mgb_solve(assemble(amg(geom); p = 1.0))   # a nonsmooth p = 1 problem
```


Statement of need

Convex variational problems are ubiquitous in computational science: nonlinear elasticity and plasticity, image denoising and segmentation (total variation), contact and obstacle problems, and non-Newtonian flow (the p-Laplacian). The difficulty is that the most interesting cases are nonsmooth (the energy is convex but not differentiable), so Newton-type methods applied naively either stagnate or require an iteration count that grows rapidly as the mesh is refined.

Interior-point (barrier) methods handle nonsmoothness robustly by following a smooth central path, but a single barrier solve still requires solving a sequence of large, increasingly ill-conditioned linear systems. The multigrid barrier method addresses both issues at once: a multigrid hierarchy preconditions the central-path subproblems so that the *total* cost stays close to linear in the number of unknowns, with rigorous bounds for the covered problem classes (Loisel, 2020, 2026b, 2026e).

The Journal of Open Source Software

State of the field

General-purpose finite-element libraries in Julia, such as `Gridap.jl` (Badia & Verdugo, 2020) and `Ferrite.jl` (Carlsson et al., 2024), and in the wider ecosystem, such as the FEniCS project (Alnæs et al., 2015), provide flexible tools for discretizing PDEs; algebraic-multigrid libraries such as `AlgebraicMultigrid.jl` (JuliaLinearAlgebra contributors, 2024) and `PyAMG` (Bell et al., 2023) solve the resulting linear systems. These are general building blocks, but none provides an out-of-the-box, theoretically grounded solver for *nonsmooth convex variational* problems. `MultiGridBarrier.jl` fills this gap. It packages a discretization, a multigrid hierarchy, and a barrier solver behind a small high-level interface, and builds on (rather than reinvents) the Julia ecosystem: it uses `AlgebraicMultigrid.jl`, or optionally `PyAMG`, to coarsen its auxiliary problems, and runs on both CPU and GPU.

Software design

A problem is solved with four composable steps. A *mesh constructor* (`fem1d`, `fem2d`, `fem2d_P1`, `fem2d_P2`, `fem3d`, `spectral1d`, `spectral2d`) returns a `Geometry` describing the discretization. `amg(geom)` attaches an algebraic-multigrid hierarchy, `assemble(mg; ...)` builds the convex problem (the functional, its barrier, and any constraints), and `mgb_solve(prob)` runs the barrier method.

Internally, the multigrid barrier method tracks the central path of an interior-point method while using a multigrid hierarchy to solve the Newton systems along that path (Loisel, 2026b, 2026e). The finite-element discretizations are isoparametric simplicial P_k and tensor-product Q_k elements. For these, the multigrid hierarchy is built algebraically: the package coarsens an auxiliary P_1/Q_1 problem on the element corners and lifts the resulting transfer operators to the full high-order basis, so the same machinery serves every finite-element family. Mesh topology is represented by an explicit node connectivity array, which decouples geometry from topology and lets geometrically coincident nodes remain distinct, supporting slit domains, branch cuts, and glued manifolds. The package also provides a structured, batched-GEMM assembly of the Newton Hessians that maps efficiently onto GPUs (Loisel, 2026a), a variant for nonuniform grids (Loisel, 2026d), and the algebraic formulation used here (Loisel, 2026c).

The Chebyshev spectral discretizations (`spectral1d`, `spectral2d`) use an intrinsically spectral hierarchy instead. The multigrid levels are a sequence of polynomial approximation spaces of increasing degree, with exact polynomial interpolation as the inter-level transfer and zero-trace boundary conditions imposed by basis construction rather than node masking. This is the setting of the spectral barrier method (Loisel, 2026e): when the solution is analytic, the spectral discretization converges geometrically and the overall method is quasi-optimal, with the iteration count growing only polylogarithmically in the number of degrees of freedom.

Convex constraints (p-norm / Euclidian-power, linear, and piecewise) are built in and composable for any discretization, and `parabolic_solve` extends the method to time-dependent problems.

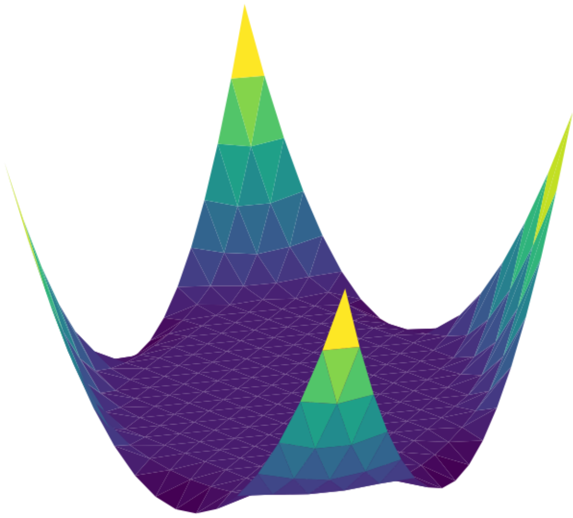

Figure 1: A nonsmooth ($p = 1$) solution of a two-dimensional p-Laplace problem, computed with `MultiGridBarrier.jl` on a refined Q_2 mesh and rendered with the package's plotting front-end.

Research impact

The methods implemented in `MultiGridBarrier.jl` underpin the numerical experiments in the papers that introduce them (Loisel, 2020, 2026b, 2026e), and the solver and its earlier implementations have been used by other researchers. For example, Zhang and Jiang use this algorithm within a convolutional-neural-network reduced-order modeling method for multiscale problems (Zhang & Jiang, 2025). More broadly, the underlying p-Laplacian algorithm (Loisel, 2020) has been cited around 26 times (Google Scholar, as of June 2026) across the numerical-PDE and optimization literature, in areas such as computational p-Laplacian numerics (Balci et al., 2023) and p-harmonic shape optimization (Müller et al., 2021).

AI usage disclosure

Generative AI assistance (Anthropic Claude, via the Claude Code command-line tool) was used in preparing this submission: drafting and editing this paper and parts of the documentation and README, generating the illustrative figure, and assisting with some software changes (refactoring, a correctness fix, and a mesh-connectivity feature). All AI-assisted output was reviewed, validated, and edited by the author, who made all core design and research decisions and takes full responsibility for the software and the manuscript.

Acknowledgements

This work received no specific external funding. The author declares no conflicts of interest.

References

Alnæs, M. S., Blechta, J., Hake, J., Johansson, A., Kehlet, B., Logg, A., Richardson, C., Ring, J., Rognes, M. E., & Wells, G. N. (2015). The FEniCS project version 1.5. *Archive of Numerical Software*, *3*(100), 9–23. https://doi.org/10.11588/ans.2015.100.20553

Badia, S., & Verdugo, F. (2020). Gridap: An extensible finite element toolbox in Julia. *Journal of Open Source Software*, *5*(52), 2520. https://doi.org/10.21105/joss.02520

Balci, A. Kh., Diening, L., & Storn, J. (2023). Relaxed Kačanov scheme for the p-laplacian with large exponent. *SIAM Journal on Numerical Analysis*, *61*(6), 2775–2794. https://doi.org/10.1137/22M1528550

Bell, N., Olson, L. N., Schroder, J., & Southworth, B. (2023). *PyAMG: Algebraic multigrid solvers in Python*. https://github.com/pyamg/pyamg.

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

Carlsson, K., Ekre, F., & Ferrite.jl contributors. (2024). *Ferrite.jl: A finite element toolbox for Julia*. https://doi.org/10.5281/zenodo.18196628

JuliaLinearAlgebra contributors. (2024). *AlgebraicMultigrid.jl: Algebraic multigrid (AMG) solvers in Julia*. https://github.com/JuliaLinearAlgebra/AlgebraicMultigrid.jl.

Loisel, S. (2020). Efficient algorithms for solving the p-laplacian in polynomial time. *Numerische Mathematik*, *146*(2), 369–400. https://doi.org/10.1007/s00211-020-01141-z

Loisel, S. (2026a). *A high-performance solver for convex optimization problems in function spaces*.

Loisel, S. (2026b). *Algorithm MGB to solve highly nonlinear elliptic PDEs in $\tilde{O}(n)$ FLOPS*.

Loisel, S. (2026c). The algebraic multi-grid-barrier method for solving p-laplace and other convex optimization problems. *Proceedings of the 29th International Conference on Domain Decomposition Methods, Milan, Italy*.

Loisel, S. (2026d). *The multi-grid-barrier method on nonuniform grids*.

Loisel, S. (2026e). The spectral barrier method to solve analytic convex optimization problems in function spaces. *Numerische Mathematik*, *158*(1), 281–302. https://doi.org/10.1007/s00211-025-01508-0

Müller, P. M., Kühl, N., Siebenborn, M., Deckelnick, K., Hinze, M., & Rung, T. (2021). A novel p-harmonic descent approach applied to fluid dynamic shape optimization. *Structural and Multidisciplinary Optimization*, *64*(6), 3489–3503. https://doi.org/10.1007/s00158-021-03030-x

Zhang, X., & Jiang, L. (2025). Convolutional neural network based reduced order modeling for multiscale problems. *Journal of Computational Physics*, *524*, 113710. https://doi.org/10.1016/j.jcp.2024.113710