The Journal of Open Source Software

LightKrylov: Lightweight implementation of Krylov subspace techniques in modern Fortran

J. Simon Kern [1*], **Ricardo S. Frantz** [1*], **and Jean-Christophe Loiseau** [1*¶]

1 Arts et Métiers Institute of Technology ¶ Corresponding author * These authors contributed equally.

DOI: 10.21105/joss.09623

Software

- Review
- Repository
- Archive

Editor: Daniel S. Katz

Reviewers:

- @joewallwork
- @ewu63
- @alexfikl

Submitted: 20 October 2025
Published: 29 January 2026

Summary

Direct solvers for linear algebraic systems scale cubically in the problem's dimension, rapidly becoming intractable for large-scale problems, while sparse factorization may still require quadratic storage due to fill-in. Krylov techniques (Krylov, 1931) avoid these costs by needing only a routine that computes a matrix-vector product, iteratively building a subspace from which the solution is obtained, see Ipsen & Meyer (1998), Saad (2003), and Frantz et al. (2023). `LightKrylov` is a Fortran package providing a suite of such Krylov methods along with an easy-to-use high level API based on `abstract` types. It is primarily intended for applications where the linear operator of interest is only available implicitly via a matrix-vector subroutine and enables users to maximally re-use existing components of their code base (including parallelisation), thus requiring a minimal set of changes without sacrificing computational performance.

Statement of need

A collection of Krylov-based algorithms in pure modern Fortran

`LightKrylov` provides Fortran users with SciPy-inspired interfaces to widely used Krylov techniques, including:

- **Linear systems** - Conjugate Gradient (CG), Generalized Minimal Residual method (GMRES), and Flexible GMRES (Saad, 1993).
- **Spectral decomposition** - Arnoldi method (with Krylov-Schur restart) for non-Hermitian operators, Lanczos tridiagonalisation for Hermitian ones.
- **SVD** - Golub-Kahan bidiagonalisation.

It is a pure Fortran package, compliant with the 2018 standard, and requiring only the community-led Fortran standard library `stdlib` (Kedward et al., 2022; Perini et al., 2026) as dependency. Moreover, its build process relies on the Fortran package manager `fpm`, facilitating its integration with the modern Fortran ecosystem.

A focus on abstract linear operators and abstract vectors

Krylov methods can be implemented without explicit reference to the data structure used to represent a vector or linear operator, nor to how the matrix-vector product is implemented. To do so, `LightKrylov` uses modern Fortran `abstract` types. A stripped-down version of the abstract vector type is shown below.

```
type, abstract :: abstract_vector_rdp
  contains
    ! Abstract procedure to compute the scalar-vector product.
    procedure(abstract_scal_rdp), pass(self), deferred :: scal
```

```
    ! Abstract procedure to compute y = alpha*x + beta*y.
    procedure(abstract_axpby_rdp), pass(self), deferred :: axpby
    ! Abstract procedure to compute the vector dot product.
    procedure(abstract_dot_rdp), pass(self), deferred :: dot
end type
```

The type-bound procedures cover the basic operations on vectors: scalar-vector product, linear combination of two vectors, and the dot product. These operations are the essential building blocks required by Krylov algorithms. Their signatures follow, to the extent possible, the BLAS standard. For instance, the `abstract_axpby_rdp` interface reads

```
abstract interface
  subroutine abstract_axpby_rdp(alpha, vec, beta, self)
    double precision, intent(in)    :: alpha, beta
    class(abstract_vector_rdp), intent(in)    :: vec
    class(abstract_vector_rdp), intent(inout) :: self
  end subroutine
end interface
```

mimicking the signature of the (extended) BLAS-1 subroutine `axpby`. Abstract linear operators are defined similarly, with two type-bound procedures required to implement the matrix-vector and transpose (or Hermitian) matrix-vector product. Using `abstract` types enables us to focus on the high-level implementation of the different algorithms while leaving the performance-critical details to the users. In addition, `LightKrylov` exposes abstract types for preconditioners, as well as a Newton-GMRES solver for nonlinear systems. After extending these `abstract` types for their application, one can solve linear systems or compute eigenvalues as easily as `call gmres(A, b, x, info)` or `call eigs(A, V, lambda, residuals, info)`.

High-level comparison with other libraries

PETSc (Balay et al., 2025) is a widely used library for large-scale linear algebra problems, especially those resulting from discretizations of partial differential equations. It can be installed using various package managers (conda, apt, Homebrew, spack, etc.) and bindings for several different languages exist (including C, Fortran, Python, Rust, and Julia). Yet, while it offers more than Krylov methods, its many data structures can make integration difficult into an already existing large code base when only linear solvers are needed.

`LightKrylov` is thus closer to `Krylov.jl` (Montoison & Orban, 2023) in Julia: a minimal package with a high level of abstraction specialised for Krylov methods only. While the latter offers a broader collection of methods, we are actively working to bridge the gap. Additionally, calling Julia code from Fortran remains a delicate process, burdened by the *two-language problem*. In that regard, the fact that `LightKrylov` is written in pure, standard-compliant Fortran makes it an ideal candidate for integration into existing Fortran codebases. Moreover, its high level of abstraction enables users to re-use existing components of their codebase (including parallelisation), requiring only minimal changes while preserving computational performance.

Performance in a production-ready open-source codebase

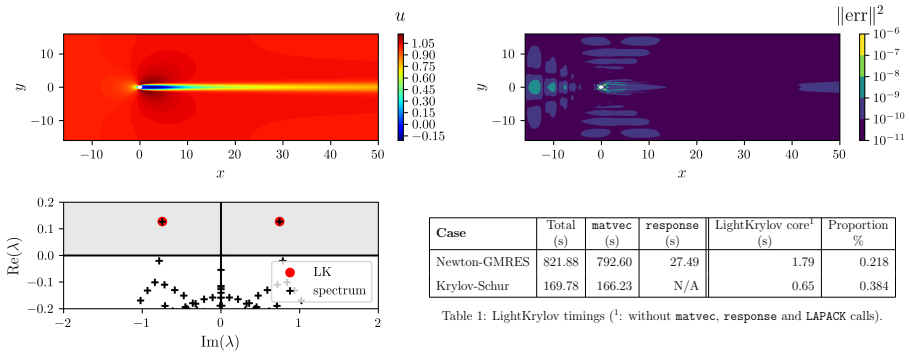
`LightKrylov` has been integrated into `neklab`, a toolbox for stability and bifurcation analysis for the spectral element solver `Nek5000`. The abstract vector interface allows direct use of `Nek5000`'s distributed data structures, and the pure-Fortran nature facilitated integration with its existing build system, demonstrating the library's suitability for large-scale HPC applications.

Hydrodynamic stability of an unstable fixed point of the nonlinear Navier-Stokes equations

Using the two-dimensional flow past a circular cylinder at Reynolds number of 100, we showcase the efficient integration of `LightKrylov` and `Nek5000` and validate the results with algorithms provided by the `KTH Framework` toolbox (Massaro et al., 2024) based on the same solver. Discretisation of the governing equations leads to systems with approximately 175,000 degrees of freedom. All computations were run in parallel on 12 Intel Core Ultra 7 processors and the numerical settings are identical for both libraries.

The unstable fixed point of the nonlinear Navier-Stokes equations is computed using both `LightKrylov`'s *time-stepper*-based Newton-GMRES solver and the selective frequency damping implementation from `KTH Framework` . Likewise, the leading eigenpair of the corresponding linearised Navier-Stokes operator is computed using `LightKrylov`'s implementation of the Krylov-Schur algorithm and the `KTH Framework`'s wrapper for `ARPACK` (R. B. Lehoucq et al., 1998).

A visual comparison is provided in Figure 1 showing excellent agreement. The table in the lower-right panel of Figure 1 summarizes the wall-clock times of the `neklab` computations. Isolating the intrinsic cost of the algorithms in `LightKrylov` from the cost of the calls to LAPACK and the linear and nonlinear Navier-Stokes solvers (`matvec` and `response`, respectively) shows that extended `abstract` types and object-oriented programming in Fortran incurs a negligible computational overhead for such large-scale applications.



Case	Total (s)	`matvec` (s)	`response` (s)	LightKrylov core[1] (s)	Proportion %
Newton-GMRES	821.88	792.60	27.49	1.79	0.218
Krylov-Schur	169.78	166.23	N/A	0.65	0.384

Table 1: LightKrylov timings ([1]: without `matvec`, `response` and `LAPACK` calls).

Figure 1: Validation of `LightKrylov`: Newton-GMRES and `eigs`. The top row depicts the streamwise velocity of the unstable solution computed using `LightKrylov` and the pointwise difference with the reference one computed with `KTH Framework`.

Perspectives

Despite being in its early development stage, `LightKrylov` has already been used in production runs on dozens of processors. It has also been interfaced with `neko`, a modernized implementation of `Nek5000` running on GPUs, and is currently being interfaced with `dNami`, a high-performance source code generator for hyperbolic partial differential equations. Current development efforts include a `Cmake` build system for easier integration into non-`fpm` codes and the implementation of the Saunders-Simon-Yip Krylov process (Saunders et al., 1988) for solving saddle point problems ubiquitous in convex optimization.

Acknowledgements

We acknowledge the financial support of the French National Agency for Research (ANR) through the ANR-33-CE46-0008-CONMAN grant agreement. We also thank the fortran-lang community for the development of `stdlib`, and in particular Frederico Perini, Jeremie

Vandenplas, and José Alvez for their work on the `stdlib_linalg` module.

References

Balay, S., Abhyankar, S., Adams, M. F., Benson, S., Brown, J., Brune, P., Buschelman, K., Constantinescu, E. M., Dalcin, L., Dener, A., Eijkhout, V., Faibussowitsch, J., Gropp, W. D., Hapla, V. clav, Isaac, T., Jolivet, P., Karpeev, D., Kaushik, D., Knepley, M. G., … Zhang, J. (2025). *PETSc web page*. https://petsc.org/. https://petsc.org/

Frantz, R. A. S., Loiseau, J.-Ch., & Robinet, J.-Ch. (2023). Krylov methods for large-scale dynamical systems: Application in fluid dynamics. *Applied Mechanics Reviews*, *75*(3). https://doi.org/10.1115/1.4056808

Ipsen, I. C. F., & Meyer, C. D. (1998). The idea behind Krylov methods. *The American Mathematical Monthly*, *105*(10), 889–899. https://doi.org/10.1080/00029890.1998.12004985

Kedward, L. J., Aradi, B., Čertík, O., Curcic, M., Ehlert, S., Engel, P., Goswami, R., Hirsch, M., Lozada-Blanco, A., Magnin, V., Markus, A., Pagone, E., Pribec, I., Richardson, B., Snyder, H., Urban, J., & Vandenplas, J. (2022). The state of Fortran. *Computing in Science & Engineering*, *24*(2), 63–72. https://doi.org/10.1109/MCSE.2022.3159862

Krylov, A. N. (1931). On the numerical solution of the equation by which, in technical matters, frequencies of small oscillations of material systems are determined. *Izvestija AN SSSR (News of Academy of Sciences of the USSR), Otdel. Mat. I Estest. Nauk*, *7*(4), 491–539.

Massaro, D., Peplinski, A., Stanly, R., Mirzareza, S., Lupi, V., Mukha, T., & Schlatter, P. (2024). A comprehensive framework to enhance numerical simulations in the spectral-element code Nek5000. *Computer Physics Communications*, *302*, 109249. https://doi.org/10.1016/j.cpc.2024.109249

Montoison, A., & Orban, D. (2023). Krylov.jl: A Julia basket of hand-picked Krylov methods. *Journal of Open Source Software*, *8*(89), 5187. https://doi.org/10.21105/joss.05187

Perini, F., Vandenplas, J., Alves, J., Clodius, W. B., Jing, Curcic, M., Skocic, M., Ehlert, S., Godara, A., Čertík, O., Loiseau, J.-C., Jahagirdar, S. S., chuckyvt, Voyles, E., Zhihua, Z., gareth-nx, Beekman, I. "Zaak", Brown, G., Degawa, T., … McMillan, L. (2026). *Fortran-lang/stdlib: v0.8.1* (Version v0.8.1). Zenodo. https://doi.org/10.5281/zenodo.18375788

R. B. Lehoucq, D. C. Sorensen, & Yang, C. (1998). Introduction to ARPACK. In *ARPACK users' guide* (pp. 1–7). https://doi.org/10.1137/1.9780898719628.ch1

Saad, Y. (1993). A flexible inner-outer preconditioned GMRES algorithm. *SIAM Journal on Scientific Computing*, *14*(2), 461–469. https://doi.org/10.1137/0914028

Saad, Y. (2003). *Iterative methods for sparse linear systems*. Society for Industrial and Applied Mathematics. https://doi.org/10.1137/1.9780898718003

Saunders, M. A., Simon, H. D., & Yip, E. L. (1988). Two conjugate-gradient-type methods for unsymmetric linear equations. *SIAM Journal on Numerical Analysis*, *25*(4), 927–940. https://doi.org/10.1137/0725052