GeophysicalFlows.jl: Solvers for geophysical fluid dynamics problems in periodic domains on CPUs & GPUs

Navid C. Constantinou[1, 2], **Gregory LeClaire Wagner**[3], **Lia Siegelman**[4], **Brodie C. Pearson**[5], **and André Palóczy**[4]

1 Australian National University **2** ARC Centre of Excellence for Climate Extremes **3** Massachussetts Institute of Technology **4** University of California San Diego **5** Oregon State University

DOI: 10.21105/joss.02996

Software

- Review
- Repository
- Archive

Editor: Pending Editor

Submitted: 29 January 2021
Published: 30 January 2021

Summary

`GeophysicalFlows.jl` is a Julia package that contains partial differential equations solvers for a collection of geophysical fluid systems in periodic domains. All modules use Fourier-based pseudospectral numerical methods and leverage the framework provided by the `FourierFlows.jl` (Constantinou et al., 2021) Julia package for time-stepping, diagnostics, and saving output.

`GeophysicalFlows.jl` utilizes Julia's functionality and abstraction to enable all modules to run on CPUs or GPUs, and to provide a high level of customizability within modules. This allows simulations to be tailored for specific research questions, via the choice of parameters, domain properties, and schemes for damping, forcing, time-stepping etc. Simulations can easily be carried out on different computing architectures, selection of the architecture on which equations are solved is done by providing the argument `CPU()` or `GPU()` during the construction of a particular problem.

Documented examples for each geophysical system (module) appear in the package's documentation, providing a stepping stone for new users and for the development of new or customized modules. Current modules include two-dimensional flow and a variety of quasi-geostrophic (QG) dynamical systems, which provide analogues to the large-scale dynamics of atmospheres and oceans. The QG systems currently in `GeophysicalFlows.jl` extend two-dimensional dynamics to include the leading order effects of a third dimension through planetary rotation, topography, surface boundary conditions, stratification and quasi-two-dimensional layering. A community-based collection of diagnostics throughout the modules are used to compute quantities like energy, enstrophy, dissipation, etc.

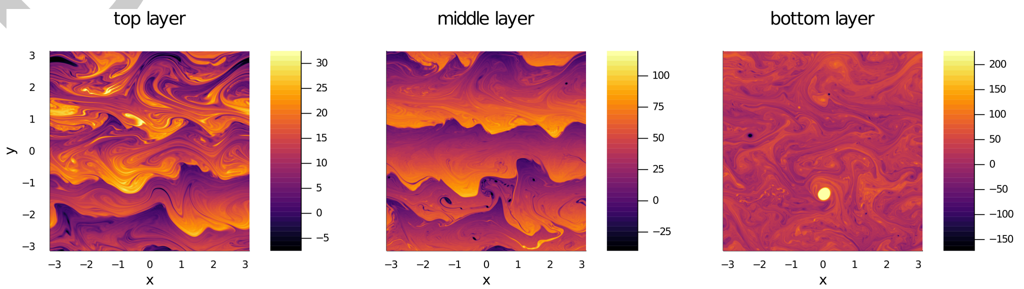

Figure 1: Snapshots from a nonlinearly equilibrated simulation of the Eady instability over a meridional ridge. Simulation used `MultiLayerQG` module of `GeophysicalFlows.jl`. The Eady problem was approximated here using 5 layers stacked up in the vertical. Each layer was simulated with 512^2 grid-points. Plots were made with the `Plots.jl` Julia package, which utilizes the `cmocean` colormaps collection (Thyng et al., 2016). Scripts to reproduce the simulation reside in the repository `github.com/FourierFlows/MultilayerQG-example`.

`GeophysicalFlows.jl` is a unique Julia package and shares similarities in functionality to the Python's `pyqg` (Abernathey et al., 2019). Beyond their base language, the major differences between these packages are that `GeophysicalFlows.jl` can be run on GPUs or CPUs and leverages a separate package (`FourierFlows.jl`; which is continuously developed) to solve differential equations and compute diagnostics, while `pyqg` can only be run on CPUs and uses a self-contained kernel. Dedalus (Burns et al., 2020) is Python software with an intuitive script-based interface that uses spectral methods to solve general partial differential equations, such as the ones within `GeophysicalFlows.jl`. There are also some other isolated codes/scripts in personal websites and in open-source public repositories that have similar functionality as some `GeophysicalFlows.jl` modules.

`GeophysicalFlows.jl` can be used to investigate a variety of scientific research questions thanks to its various modules and high customizability, and its ease-of-use makes it an ideal teaching tool for fluids courses (Constantinou, 2020; Constantinou & Wagner, 2020). `GeophysicalFlows.jl` has been used in developing Lagrangian vortices identification algorithms (Karrasch & Schilling, 2020). Currently, `GeophysicalFlows.jl` is being used, e.g., (i) to test new theories for diagnosing turbulent energy transfers in geophysical flows (e.g. Pearson et al., Under Review), (ii) to compare different observational sampling techniques in these flows, (iii) to study the bifurcation properties Kolmogorov flows (Constantinou & Drivas, 2020), (iv) to study the genesis and persistence of the polygons of vortices present at Jovian high latitudes (Siegelman, Young, and Ingersoll; in prep)."

Acknowledgements

We acknowledge discussions with Keaton Burns, Valentin Churavy, Cesar Rocha, and William Young. B. C. P. was supported by the National Science Foundation under Grant No. 2023721. We would also like to take a moment to remember Sean R. Haney (February 1987-January 2021) who left us a bit too early.

References

Abernathey, R., Rocha, C. B., Jansen, M., Poulin, F. J., Constantinou, N. C., Balwada, D., Sinha, A., Bueti, M., Penn, J., Wolfe, C. L. P., & Boas, B. V. (2019). *Pyqg/pyqg: v0.3.0* (Version v0.3.0) [Computer software]. Zenodo. https://doi.org/10.5281/zenodo.3551326

Burns, K. J., Vasil, G. M., Oishi, J. S., Lecoanet, D., & Brown, B. P. (2020). Dedalus: A flexible framework for numerical simulations with spectral methods. *Phys. Rev. Research*, *2*, 023068. https://doi.org/10.1103/PhysRevResearch.2.023068

Constantinou, N. C. (2020). CLExWinterSchool2020. In *GitHub repository*. GitHub. https://github.com/navidcy/CLExWinterSchool2020

Constantinou, N. C., & Drivas, T. D. (2020). KolmogorovFlow. In *GitHub repository*. GitHub. https://github.com/navidcy/KolmogorovFlow

Constantinou, N. C., & Wagner, G. L. (2020). GeophysicalFlows-examples. In *GitHub repository*. GitHub. https://github.com/FourierFlows/GeophysicalFlows-Examples

Constantinou, N. C., Wagner, L. C., & Palóczy, A. (2021). *FourierFlows/FourierFlows.jl: v0.6.10* (Version v0.6.10) [Computer software]. Zenodo. https://doi.org/10.5281/zenodo.4460734

Karrasch, D., & Schilling, N. (2020). Fast and robust computation of coherent Lagrangian vortices on very large two-dimensional domains. *SMAI Journal of Computational Mathematics*, *6*, 101–124. https://doi.org/10.5802/smai-jcm.63

Pearson, B. C., Pearson, J. L., & Fox-Kemper, B. (Under Review). Advective structure functions in anisotropic two-dimensional turbulence. *J. Fluid Mech.*

Thyng, K. M., Greene, C. A., Hetland, R. D., Zimmerle, H. M., & DiMarco, S. F. (2016). True colors of oceanography: Guidelines for effective and accurate colormap selection. *Oceanography*, *29*(3), 9–13. https://doi.org/10.5670/oceanog.2016.66