The Journal of Open Source Software

SpeedyWeather.jl: Reinventing atmospheric general circulation models towards interactivity, extensibility and composability

Milan Klöwer[1,2¶], **Maximilian Gelbrecht**[3,4], **Daisuke Hotta**[5,6], **Justin Willmert**[7], **Simone Silvestri**[1], **Gregory L Wagner**[1], **Alistair White**[3,4], **Sam Hatfield**[8], **David Meyer**[8], **Tom Kimpson**[2,9], **Navid C Constantinou**[10], and **Chris Hill**[1]

1 Massachusetts Institute of Technology, Cambridge, MA, USA **2** University of Oxford, UK **3** Technical University Munich, Germany **4** Potsdam Institute for Climate Impact Research, Germany **5** Japan Meteorological Agency, Tsukuba, Japan **6** University of Reading, UK **7** University of Minnesota, Minneapolis, MN, USA **8** European Centre for Medium-Range Weather Forecasts, Reading, UK **9** University of Melbourne, Melbourne, Australia **10** Australian National University, Canberra, Australia ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

SpeedyWeather.jl is a library to simulate and analyse the global atmospheric circulation on the sphere. It implements several 2D and 3D models to solve the primitive equations with and without humidity, the shallow water equations (Figure 1), or the barotropic vorticity equations with spherical harmonics.

The user interface of SpeedyWeather.jl is heavily influenced by Oceananigans.jl (Ramadhan et al., 2020). A monolithic interface is deliberately avoided

To be extendible and composable, SpeedyWeather.jl relies on multiple dispatch from the Julia programming language (Bezanson et al., 2017). Every model component is defined as a new type `SomeComponent <: AbstractComponent`, i.e. subtype of an abstract super type `AbstractComponent`. Extending SpeedyWeather.jl can therefore easily achieved by defining a new `OtherComponent <: AbstractComponent`

The dynamical core of SpeedyWeather.jl uses established numerics (Bourke, 1972; Hoskins & Simmons, 1975; Simmons et al., 1978; Simmons & Burridge, 1981), widely adapted in numerical weather prediction. It is based on the spherical harmonic transform with a leapfrog-based semi-implicit time integration (Hoskins & Simmons, 1975) and a Robert-Asselin-Williams filter (Amezcua et al., 2011; Williams, 2011). The spherical harmonic transform is grid-flexible. Any iso-latitude ring-based grid can be used and new grids can be externally defined and passed on as argument. Many grids are already implemented: The conventional Gaussian grid, a regular longitude-latitude grid, the octahedral Gaussian grid (Malardel et al., 2016), the octahedral Clenshaw-Curtis grid (Hotta & Ujiie, 2018), and the HEALPix grid (Górski et al., 2005). Both SpeedyWeather.jl and its spherical harmonic transform are also number format-flexible. 32-bit single-precision floating-point numbers (Float32) are the default as adapted by other modelling efforts (Váňa et al., 2017), but Float64 and other custom number formats can be used with a single code basis. Julia will compile to the choice of the number format and grid (and other model components) just-in-time.

Statement of need

Most weather and climate models are written in Fortran and have been developed over decades. From this tradition follows a specific programming style and associated user interface. SpeedyWeather.jl is a fresh approach to atmospheric models that have been very influential in many areas of scientific and high-performance computing as well as climate change mitigation and adaptation.

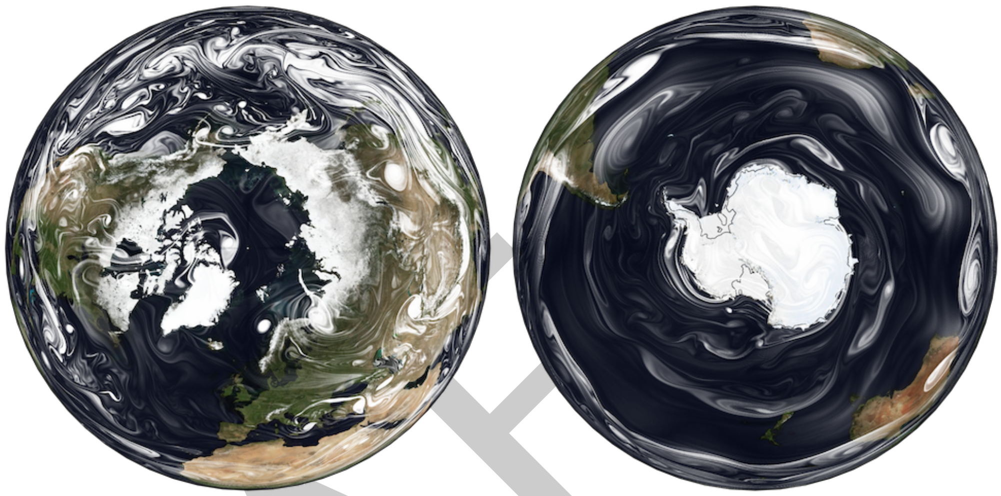

Figure 1: Shallow water simulation with SpeedyWeather.jl. Relative vorticity with a spectral resolution of T1023 (about 20km) simulated in Float32 on an octahedral Clenshaw-Curtis grid (Hotta & Ujiie, 2018). Relative vorticity is visualised with Matplotlib (Hunter, 2007) and Cartopy (Met Office, 2010 - 2015) using a transparent-to-white colormap to mimic the appearance of clouds. Underlain is NASA's blue marble from June 2004.

Acknowledgements

We acknowledge contributions from Mosè Giordano, Valentin Churavy and Pietro Monticone who have also committed to the SpeedyWeather.jl repository, and the wider Julia community help and support. We gratefully acknowledge funding from the National Science Foundation (…) and the European Research Council under the European Union's Horizon 2020 research and innovation programme for the ITHACA grant (no. 741112).

References

Amezcua, J., Kalnay, E., & Williams, P. D. (2011). The Effects of the RAW Filter on the Climatology and Forecast Skill of the SPEEDY Model. *Monthly Weather Review*, *139*(2), 608–619. https://doi.org/10.1175/2010MWR3530.1

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

Bourke, W. (1972). An Efficient, One-Level, Primitive-Equation Spectral Model. *Monthly Weather Review*, *100*(9), 683–689. https://doi.org/10.1175/1520-0493(1972)100%3C0683:AEOPSM%3E2.3.CO;2

Górski, K. M., Hivon, E., Banday, A. J., Wandelt, B. D., Hansen, F. K., Reinecke, M., & Bartelmann, M. (2005). HEALPix: A Framework for High-Resolution Discretization and

The Journal of Open Source Software

Fast Analysis of Data Distributed on the Sphere. *The Astrophysical Journal*, *622*(2), 759. https://doi.org/10.1086/427976

Hoskins, B. J., & Simmons, A. J. (1975). A multi-layer spectral model and the semi-implicit method. *Quarterly Journal of the Royal Meteorological Society*, *101*(429), 637–655. https://doi.org/10.1002/qj.49710142918

Hotta, D., & Ujiie, M. (2018). A nestable, multigrid-friendly grid on a sphere for global spectral models based on Clenshaw–Curtis quadrature. *Quarterly Journal of the Royal Meteorological Society*, *144*(714), 1382–1397. https://doi.org/10.1002/qj.3282

Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Malardel, S., Wedi, N., Deconinck, N., Diamantakis, M., Kuehnlein, C., Mozdzynski, G., Hamrud, M., & Smolarkiewicz, P. (2016). A new grid for the IFS. In *ECMWF Newsletter*. https://www.ecmwf.int/node/15041.

Met Office. (2010 - 2015). *Cartopy: A cartographic python library with a matplotlib interface*. https://scitools.org.uk/cartopy

Ramadhan, A., Wagner, G. L., Hill, C., Campin, J.-M., Churavy, V., Besard, T., Souza, A., Edelman, A., Ferrari, R., & Marshall, J. (2020). Oceananigans.jl: Fast and friendly geophysical fluid dynamics on GPUs. *Journal of Open Source Software*, *5*(53), 2018. https://doi.org/10.21105/joss.02018

Simmons, A. J., & Burridge, D. M. (1981). An Energy and Angular-Momentum Conserving Vertical Finite-Difference Scheme and Hybrid Vertical Coordinates. *Monthly Weather Review*, *109*(4), 758–766. https://doi.org/10.1175/1520-0493(1981)109%3C0758:AEAAMC%3E2.0.CO;2

Simmons, A. J., Hoskins, B. J., & Burridge, D. M. (1978). Stability of the Semi-Implicit Method of Time Integration. *Monthly Weather Review*, *106*(3), 405–412. https://doi.org/10.1175/1520-0493(1978)106%3C0405:SOTSIM%3E2.0.CO;2

Váňa, F., Düben, P., Lang, S., Palmer, T., Leutbecher, M., Salmond, D., & Carver, G. (2017). Single Precision in Weather Forecasting Models: An Evaluation with the IFS. *Monthly Weather Review*, *145*(2), 495–502. https://doi.org/10.1175/MWR-D-16-0228.1

Williams, P. D. (2011). The RAW Filter: An Improvement to the Robert–Asselin Filter in Semi-Implicit Integrations. *Monthly Weather Review*, *139*(6), 1996–2007. https://doi.org/10.1175/2010MWR3601.1