

SOFTWARE METAPAPER

Windspharm: A High-Level Library for Global Wind Field Computations Using Spherical Harmonics

Andrew Dawson

Atmospheric, Oceanic & Planetary Physics, Department of Physics, University of Oxford, Oxford, UK
andrew.dawson@physics.ox.ac.uk

The *windspharm* library is a Python package for performing computations on global wind fields in spherical geometry. It provides a high-level interface for computing derivatives and integrals of vector wind fields over a sphere using spherical harmonics. The software allows for computations with plain arrays, or with structures that include metadata, integrating with several popular data analysis libraries from the atmospheric and climate science community. The software is available on Github.

Keywords: Spherical harmonics; Meteorology; Wind vectors; Python

(1) Overview

Introduction

Analysis of global wind fields is common in meteorology and climate science. Often we want to examine not just the vector wind field, but more complex derived quantities that allow deeper insight into the properties of the flow such as rotation and convergence/divergence. These properties of the flow are often expressed as derivatives or integrals of the vector wind. For example vorticity, a measure of rotation in a fluid, is computed as the curl of the wind field, and the corresponding streamfunction is the inverse Laplacian of vorticity. The derivative quantities could be computed via finite difference approximations, although care needs to be taken to correctly account for spherical geometry. However, the integral quantities can be much more difficult to compute.

One way to approach the problem of computing complex derivatives and integrals over the sphere is to use a spectral representation, where a field over the surface of a sphere is represented as the summation of a finite series of basis functions. The appropriate basis functions on the sphere are the spherical harmonics [1]. Spherical harmonics have some nice properties that make derivatives and integrals of fields expressed in terms of them straightforward to compute. For example, the inverse Laplacian operator required to compute streamfunction from vorticity is simply a scalar multiplication in the spectral representation, since the Laplacian of a spherical harmonic is proportional to the spherical harmonic itself [2]. A particular caveat of using the spectral approach is that the field to be analysed must have global coverage, it cannot work on data that cover only a limited area of the globe as the expansion in terms of basis spherical harmonics would not be defined. However, there are still many use cases

in meteorology and climate science where this restriction does not present a problem.

Computations using the spectral method require a spherical harmonic transform, a procedure that takes a field defined on a regular global grid and transforms it into its spectral representation in terms of spherical harmonic coefficients. An inverse spherical harmonic transform procedure is also required in order to return a field represented in terms of spherical harmonic coefficients back to a regular grid. Finally, procedures to compute derivatives and integrals of the fields in spectral form are required. These are not trivial to implement, and doing so is out of scope for most researchers simply wishing to analyse global atmospheric circulation. However, these transforms and calculus operations are commonly used in atmospheric modelling, and well tested and optimized implementations of the required algorithms already exist. The *windspharm* library is designed to bridge the gap between the data analyst and the numerical computation, by providing a simple high-level interface to an existing numerical code that performs the complex calculations for the spectral method.

Of the existing software packages for computing with spherical harmonics, many are intended as low-level building blocks for implementing computational models [3,4,5,6]. These packages expose the spectral transform algorithms at a low-level, and are not easy to use or appropriate for most data analyst users. Some packages offer a higher-level interface to spherical harmonic computations aimed at data analysis, for example NCL [7] has a good suite of spherical harmonic operators that are tailored to atmospheric science use cases, although the capability to retain metadata through the computations is only partial due to language-level restrictions. *windspharm* was designed to allow users to take advantage of the powerful

spectral computations without having to handle spherical harmonic coefficients themselves. The *windspharm* library brings a high-level interface, designed to be easy to use for atmospheric science use cases, along with good metadata support to atmospheric scientists working with Python.

Implementation and architecture

The *windspharm* library is implemented in a hierarchical structure. The core of the library consists of a solver object, which is constructed by passing input vector wind components on a regular grid represented as NumPy arrays [8]. Method calls to the solver object are made to construct the required output, again in the form of gridded arrays. Given vector wind components the solver can compute vorticity (relative and absolute), divergence, streamfunction, velocity potential, and rotational and divergent components of the wind field. **Figure 1** shows example outputs from computing vorticity and streamfunction from a wind field. The solver can also apply spectral truncation to scalar fields, and compute the gradient of an arbitrary scalar field. These supplementary operations are useful for processing quantities output from other methods of the solver.

The solver is built on top of the *pyspharm* library, which is a Python interface to the *SPHEREPACK* software [3,9]. All of the numerical computation is handled by *SPHEREPACK*, a well-tested and highly optimized computational core, via the *pyspharm* Python interface. The primary role of *windspharm* is to provide an interface to these computations that does not expose the details of the methods. The user inputs data on a regular grid and always gets back results on the same regular grid, with *windspharm* managing calls to *pyspharm* to perform the operations required to compute the result.

On top of the core solver are several interfaces that can work with data structures that contain structured metadata as well as data values. Currently there are metadata interfaces that understand the data structures from iris [10], xarray [11], and cdms2 (part of UV-CDAT) [12]. The purpose of these interfaces is to allow the application of the computational routines to metadata-containing data structures without losing metadata. The consistency of metadata and data provenance in general are extremely important for correct and reproducible science, so this functionality holds significant value. The metadata interfaces are able to use the provided metadata to automatically reconfigure input data structures to the form required by the computational routines, and reform the output to match the input shape and dimensionality in a way that is transparent to the user. The presence of metadata also allows *windspharm* to perform more comprehensive error checking (e.g., identification of invalid input grids). The metadata interfaces are therefore not only convenient for the user, but also provide a benefit for program correctness. Extending the library to support other metadata-containing data structures is straightforward since the hierarchical design allows the implementation of additional metadata interfaces without having to implement any of the numerical components.

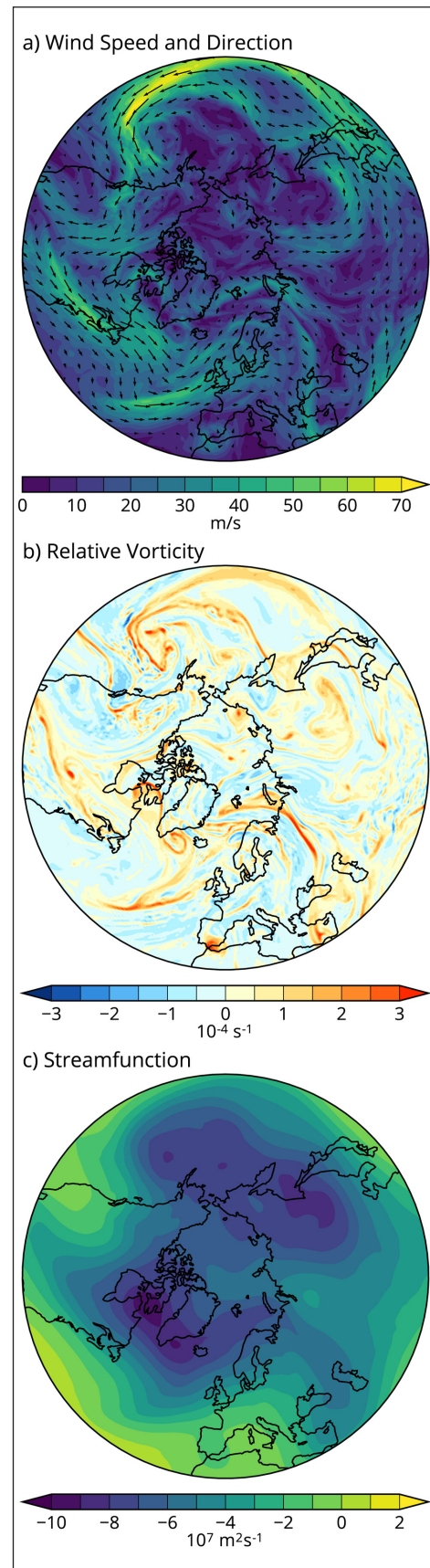


Figure 1: Examples of quantities computed with *windspharm*. a) Wind speed (colours) and direction (arrows) on the 500 hPa pressure surface from the ECMWF analysis valid at 00:00 UTC on the 27th January 2016. b) Relative vorticity, and c) streamfunction, both computed from the wind field using *windspharm*.

Quality control

The *windspharm* library features a suite of unit and integration tests that test the core functionality of the software. The end user can run these tests against their installed version of *windspharm* to verify it is functioning correctly. The test suite is updated as development on the library progresses. A continuous integration service Travis CI [13] is used to automatically run the full test suite on a variety of Python versions each time a pull request to the *windspharm* repository is made, which helps to ensure that changes and new features do not break the existing functionality.

The *windspharm* library also comes with some example programs and data, to demonstrate examples of how the library may be used. These example programs may require additional dependencies for plotting output, which are noted within the documentation for each program.

(2) Availability

Operating system

Linux, OSX

Programming language

Python 2.7 or Python \geq 3.3

Dependencies

windspharm has a set of core dependencies that must be installed to use the software:

- `setuptools \geq 0.7.2`
- `numpy \geq 1.6`
- `pyspharm \geq 1.8`

The metadata interfaces are optional, and must be supported by additional optional dependencies. It is not expected that any one user would require all the available metadata interfaces, only needing the dependency for their interface(s) of choice from the list below:

- `iris \geq 1.2` (for the iris metadata interface)
- `cdms2` (for the cdms2 metadata interface)
- `xarray` (for the xarray metadata interface)

Running the provided test suite also requires additional optional dependencies that are not required for normal use of *windspharm*:

- `nose`
- `pep8`

Software location

Archive

Name: Zenodo

Persistent identifier: <http://dx.doi.org/10.5281/zenodo.50571>

Licence: MIT License

Publisher: Andrew Dawson

Version published: 1.5.0

Date published: 27/04/2016

Code repository

Name: Github

Identifier: <https://github.com/ajdawson/windspharm>

Licence: MIT License

Date published: 27/04/2016

Language

English

(3) Reuse potential

windspharm has been well used in the meteorology and climate science community. The library is distributed as part of the Ultrascale Visualization Climate Data Analysis Tools (UV-CDAT) project [12], and has been used in a significant number of publications that the author is aware of [14,15,16,17,18,19,20,21,22,23,24]. The library is also provided as part of the JASMIN Analysis Platform [25], indicating it has a significant user base. The potential for reuse is high, since *windspharm* implements a simple interface to complex vector computations that are extremely common in meteorology. Multiple metadata interfaces add to this potential, by allowing easy access to metadata retaining computations without forcing a particular environment on the user. Some users may be concerned about errors arising from the representation of the Earth as a sphere. On the typical spatial scales analysed with *windspharm*, errors due to shape of the Earth approximations are likely to be negligible. In addition to this, most applications of *windspharm* are to model derived data sets where the model will also have used a spherical Earth approximation, and therefore this potential source of error is not a concern. There also is potential for reuse in the wider field of planetary sciences. The package was designed for studies of terrestrial winds, but supports setting the radius of the sphere used in the spherical harmonic computations to an arbitrary distance, allowing its use on data and model output relevant to non-Earth planets with different planetary scales.

The software is documented on-line at <http://ajdawson.github.io/windspharm>. The software is supported on a voluntary basis through the repository's issue tracker. Contributions to the project are welcomed, and can be submitted by making a pull request to the *windspharm* Github repository.

Acknowledgements

The author thanks Jeff Whitaker for providing *pyspharm*, the Python interface to *SPHEREPACK*, as free software. The plots in this paper were produced with *cartopy* [26], *iris* [10], and *matplotlib* [27].

Competing Interests

The author declares that they have no competing interests.

References

1. **Riley, K F, Hobson, M P and Bence, S J** 2006 *Mathematical Methods for Physics and Engineering*. Cambridge University Press, third edition. DOI: <http://dx.doi.org/10.1017/CBO9780511810763>

2. **Holton, J R** 2004 An introduction to dynamic meteorology. Academic Press, fourth edition.
3. **Adams, J C** and **Swarztrauber, P N** 2011 SPHEREPACK 3.2: A Package for Modeling Geophysical Processes. Available at <https://www2.cisl.ucar.edu/resources/legacy/spherepack>.
4. **Schaeffer, N** 2013 Efficient spherical harmonic transforms aimed at pseudospectral numerical simulations. *Geochemistry, Geophysics, Geosystems*, 14(3): 751–758. DOI: <http://dx.doi.org/10.1002/ggge.20071>
5. **Wieczorek, M A**, **Meschede, M** and **Oshchepkov, I** 2015 SHTOOLS – Tools for working with spherical harmonics (v3.1). DOI: <http://dx.doi.org/10.5281/zenodo.20920>
6. **Whitaker, J S** 2016 pyspharm: An object-oriented python interface to the NCAR SPHEREPACK library. Available at <https://github.com/jswhit/pyspharm>.
7. **UCAR/NCAR/CISL/TDD** 2016 The NCAR Command Language (Version 6.3.0). DOI: <http://dx.doi.org/10.5065/D6WD3XH5>
8. **van der Walt, S**, **Colbert, S C** and **Varoquaux, G** 2011 The NumPy Array: A Structure for Efficient Numerical Computation. *Computing in Science and Engineering*, 13(2): 2230, DOI: <http://dx.doi.org/10.1109/MCSE.2011.37>
9. **Adams, J C** and **Swarztrauber, P N** 1999 SPHEREPACK 3.0: A model development facility. *Monthly Weather Review*, 127: 1872–1878. DOI: [http://dx.doi.org/10.1175/1520-0493\(1999\)127%3C1872:SAMDF%3E2.0.CO;2](http://dx.doi.org/10.1175/1520-0493(1999)127%3C1872:SAMDF%3E2.0.CO;2)
10. **Met Office** 2016 Iris: a Python library for meteorology and climatology. Available at <http://scitools.org.uk/iris>.
11. **Hoyer, S**, **Kleeman, A** and **Brevdo, E** 2016 Xarray: n-d labeled arrays and datasets in Python. Available at <http://xarray.pydata.org>.
12. **Williams, D N**, **Doutriaux, C**, **Chaudhary, A**, **Fries, S**, **Lipsa, D**, **Jhaveri, S**, **Durack, P J**, **Painter, J**, **Nadeau, D** and **Maxwell, T** 2016 uvcdat v2.4.0. DOI: <http://dx.doi.org/10.5281/zenodo.45136>
13. **Travis CI**: <https://travis-ci.org>.
14. **Dawson, A**, **Matthews, A J** and **Stevens, DP** 2011 Rossby wave dynamics of the extra-tropical response to El Nino: Importance of the basic state in coupled GCMs. *Climate Dynamics*, 37(1): 391–405. DOI: <http://dx.doi.org/10.1007/s00382-012-1518-6>
15. **Matthews, A J** 2012 A multiscale framework for the origin and variability of the South Pacific Convergence Zone. *Quarterly Journal of the Royal Meteorological Society*, 138(666): 1165–1178. DOI: <http://dx.doi.org/10.1002/qj.1870>
16. **Dawson, A**, **Stevens, DP**, **Matthews, A J**, **Roberts, M J** and **Vidale, P L** 2013 Importance of oceanic resolution and mean state on the extra-tropical response to El Nino in a matrix of coupled models. *Climate Dynamics*, 41(5): 1439–1452. DOI: <http://dx.doi.org/10.1007/s00382-012-1518-6>
17. **Peatman, S C**, **Matthews, A J** and **Stevens, D P** 2014 Propagation of the Madden-Julian Oscillation through the Maritime Continent and scale interaction with the diurnal cycle of precipitation. *Quarterly Journal of the Royal Meteorological Society*, 140(680): 814–825. DOI: <http://dx.doi.org/10.1002/qj.2161>
18. **Irving, D** and **Simmonds, I** 2015 A novel approach to diagnosing Southern Hemisphere planetary wave activity and its influence on regional climate variability. *Journal of Climate*, 28: 9041–9057. DOI: <http://dx.doi.org/10.1175/JCLI-D-15-0287.1>
19. **van der Wiel, K**, **Matthews, A J**, **Stevens, D P** and **Joshi, M M** 2015 A dynamical framework for the origin of the diagonal South Pacific and South Atlantic convergence zones. *Quarterly Journal of the Royal Meteorological Society*, 141(691): 1997–2010. DOI: <http://dx.doi.org/10.1002/qj.2508>
20. **Xu, G**, **Osborn, T J**, **Matthews, A J** and **Joshi, M M** 2015 Different atmospheric moisture divergence responses to extreme and moderate El Ninos. *Climate Dynamics*. DOI: <http://dx.doi.org/10.1007/s00382-015-2844-2>
21. **Irving, D** and **Simmonds, I** 2016 A new method for identifying the Pacific-South American Pattern and its influence on regional climate variability. *Journal of Climate*, in press. DOI: <http://dx.doi.org/10.1175/JCLI-D-15-0843.1>
22. **van der Wiel, K**, **Matthews, A J**, **Joshi, M M** and **Stevens, D P** 2016 The influence of diabatic heating in the South Pacific Convergence Zone on Rossby wave propagation and the mean flow. *Quarterly Journal of the Royal Meteorological Society*, 142(695): 901–910. DOI: <http://dx.doi.org/10.1002/qj.2692>
23. **van der Wiel, K**, **Matthews, A J**, **Joshi, M M** and **Stevens, D P** 2016 Why the South Pacific Convergence Zone is diagonal. *Climate Dynamics*, 46(5): 1683–1698. DOI: <http://dx.doi.org/10.1007/s00382-015-2668-0>
24. **Watson, P A G**, **Weisheimer, A**, **Knight, J R** and **Palmer, T N** 2016 The role of the tropical West Pacific in the extreme Northern Hemisphere winter of 2013/2014. *Journal of Geophysical Research*, 121(4): 1698–1714. DOI: <http://dx.doi.org/10.1002/2015JD024048>
25. **Science and Technology Facilities Council/Natural Environment Research Council**: <http://www.jasmin.ac.uk>.
26. **Met Office** 2016 Cartopy: a cartographic python library with a matplotlib interface. Available at <http://scitools.org.uk/cartopy>.
27. **Hunter, J D** 2007 Matplotlib: A 2D graphics environment. *Computing in Science and Engineering*, 9(3): 90–95. DOI: <http://dx.doi.org/10.1109/MCSE.2007.55>

How to cite this article: Dawson, A 2016 Windspharm: A High-Level Library for Global Wind Field Computations Using Spherical Harmonics. *Journal of Open Research Software*, 4: e31, DOI: <http://dx.doi.org/10.5334/jors.129>

Submitted: 09 May 2016 **Accepted:** 13 July 2016 **Published:** 02 August 2016

Copyright: © 2016 The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC-BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. See <http://creativecommons.org/licenses/by/4.0/>.



Journal of Open Research Software is a peer-reviewed open access journal published by Ubiquity Press

OPEN ACCESS The Open Access icon, which is a stylized 'a' inside a circle.