

NumbaCS: A fast Python package for coherent structure analysis

³ Albert Jarvis ¹ and Shane D. Ross ²

⁴ 1 Department of Mechanical Engineering, Virginia Tech, Blacksburg, VA, USA 2 Department of Ocean

and Aerospace Engineering, Virginia Tech, Blacksburg, VA, USA

• Summary

NumbaCS (Numba Coherent Structures) is a Python package that efficiently implements a 7 variety of methods for studying material transport in time-dependent fluid flows. It leverages Numba – a high performance Python compiler for generating optimized machine code from Python functions – along with other Numba-compatible packages behind the scenes, producing 10 fast and user-friendly implementations of coherent structure methods. "Coherent structure 11 methods" refer to any method which can be used to infer or extract Lagrangian and objective 12 Eulerian coherent structures. The theory behind these methods have been developed over 13 the last few decades with the aim of extending many of the important invariant objects from 14 time-independent dynamical systems theory to the more general setting where a system may 15 have arbitrary time dependence and may only be known or defined for some finite time. These 16 time-dependent systems are ubiquitous in the context of geophysical and engineering flows 17 where the evolution of the velocity field depends on time and velocity data representing these flows is not available for all time. By extending the ideas from the time-independent setting to the more general time-dependent setting, important transient objects (coherent structures) can be identified which govern how material is transported within a flow. Understanding material transport in flows is of great importance for applications ranging from monitoring the transport of a contaminant in the ocean or atmosphere to informing search and rescue strategies for 23 persons lost at sea. 24

Statement of need

As theory and implementations of coherent structures have been developed (Farazmand & 26 Haller, 2012; Haller, 2011; Haller et al., 2016; Haller & Beron-Vera, 2013; Haller & Poje, 27 1998; Mathur et al., 2007; Nolan, Serra, et al., 2020; Schindler et al., 2012; Serra & Haller, 28 2016; Shadden et al., 2005) and the utility of these tools has been demonstrated over the last 29 two decades (Curbelo & Rypina, 2023; Du Toit & Marsden, 2010; Günther et al., 2021; Liu 30 et al., 2018; Nolan, Foroutan, et al., 2020; Peacock & Haller, 2013; Pretorius et al., 2023; 31 Rutherford et al., 2012; Serra et al., 2017), there has been a steadily growing interest in using 32 these methods for real-world applications. Early on, software implementations were largely 33 contained to in-house packages developed by applied mathematicians and engineers advancing 34 the theory. Over the years, there have been a number of software packages developed in an 35 attempt to provide implementations of some of these methods for practitioners outside of the 36 field. Some provide a friendly interface for users (Dynlab – Nolan (2024); LCS MATLAB Kit – 37 Dabiri (2009)), others aim to provide efficient implementations of specific methods (sometimes 38 in specific circumstances) (Lagrangian – Briol & d'Ovidio (2011); Newman – Du Toit (2010); 39 Aquila-LCS – Lagares & Araya (2023)), and a few implement a variety of methods (Tbarrier 40 - Bartos et al. (2022); LCS Tool - Onu et al. (2015)). NumbaCS intends to unite these 41

⁴² aims by providing efficient and user-friendly implementations of a variety of coherent structure

Jarvis, & Ross. (2025). NumbaCS: A fast Python package for coherent structure analysis. *Journal of Open Source Software*, ¿VOL?(¿ISSUE?), 1 7948. https://doi.org/10.xxxxx/draft.

DOI: 10.xxxxx/draft

Software

- Review ¹
- Repository I^A
- Archive 🗗

Editor: Vincent Knight 🖒 💿 Reviewers:

- @janfb
- Øslayoo

Submitted: 12 October 2024 Published: unpublished

License

Authors of papers retain copyright₉ and release the work under a Creative Commons Attribution 4.0 International License (CC BY 4.0)



- methods. By doing this, the hope is to provide a powerful tool for experienced practitioners 43
- and a low barrier of entry for newcomers. In addition, as new methods/implementations arise, 44
- the framework laid out in NumbaCS provides a straightforward environment for contributions 45
- and maintenance. Also of note is another package called CoherentStructures.jl (Junge et 46
- al., 2020), which is fast, user-friendly, and implements a variety of methods. This package 47
- has some overlap with NumbaCS but they both implement methods which the other does not. 48
- CoherentStructures.jl is a powerful tool that should be considered by users who perhaps 49
- prefer Julia to Python or are interested in computing some of the methods not implemented in 50 NumbaCS. For a more detailed breakdown of how all of the mentioned packages compare with
- 51
- NumbaCS, see the documentation. 52

Functionality 53

55

56

57

58

59

60

61

62

64

65

67

- NumbaCS implements the following features for both analytical and numerical flows: 54
 - Standard flow map computation
 - Flow map composition method (Brunton & Rowley, 2010)
 - Finite time Lyapunov exponent (FTLE) (Shadden et al., 2005)
 - instantaneous Lyapunov exponent (iLE) (Nolan, Serra, et al., 2020)
 - Lagrangian averaged vorticity deviation (LAVD) (Haller et al., 2016)
 - Instantaneous vorticity deviation (IVD) (Haller et al., 2016)
 - FTLE ridge extraction (Schindler et al., 2012; Steger, 1998)
 - Variational hyperbolic LCS (Farazmand & Haller, 2012; Haller, 2011)
- Variational hyperbolic OECS (Serra & Haller, 2016) 63 •
 - LAVD-based elliptic LCS (Haller et al., 2016)
 - IVD-based elliptic OECS (Haller et al., 2016)

For flows defined by numerical velocity data: 66

Simple creation of JIT compiled linear and cubic interpolants

All of these implementations are relatively straightforward to use and quite efficient. This is due 68 to three key dependencies NumbaCS utilizes to speed up computations. The first is Numba (Lam 69 et al., 2015), a JIT compiler for Python which can drastically speed up numerical operations 70 and provides a simple framework for parallelizing tasks. Next, numbalsoda (Wogan, 2021) is 71 a Python wrapper to ODE solvers in both C++ (LSODA) and FORTRAN (DOP853) that 72 bypasses the Python interpreter and can be used within Numba functions (common Python 73 ODE solvers, such as those provided by the SciPy package, cannot be executed within Numba 74 functions). This package is crucial to the efficiency of NumbaCS as particle integration is often 75 the most costly part of finite-time coherent structure methods. Finally, the interpolation 76 package (Winant et al., 2017) provides optimized interpolation in Python and is utilized in 77 NumbaCS to create JIT compiled interpolant functions, producing efficient implementations of 78 methods even for flows defined by numerical data. By taking advantage of these packages 79 behind the scenes, NumbaCS is able to maintain the simplicity and readability of an interpreted language while achieving runtimes closer to that of a compiled language. 81

Examples 82

A User Guide is provided which details the workflow in NumbaCS and a number of examples 83 demonstrating the functionality are covered in the Example Gallery. Here we show the output 84 of a few examples, provide the runtime of each, and breakdown the runtime based on the parts 85 of each method. "Flowmap" refers to the particle integration step, "C eig" and "S eig" refer 86 to the eigenvalue/vector step for Lagrangian and Eulerian methods respectively (this time will 87 be roughly equal to the FTLE and iLE times), and the last is the extraction time for a given 88

method. For examples that require particle integration, the default solver (DOP853) was used 89



- with the default error tolerances (relative tolerance = 1e-6, absolute tolerance = 1e-8). All
- ⁹¹ runs were performed on an Intel^(R) CoreTM i7-3770K CPU @ 3.50GHz (which has 4 cores and
- $_{\rm 92}~$ 8 total threads). Warm-up time 1 is not included in the timings.
 - 1.0 1.0 0.8 0.8 0.6 0.6 0.4 0.4 0.2 0.2 0.0 0.0 0.00 0.25 0.50 0.75 1.00 1.25 1.50 1.75 2.00 0.25 0.50 0.75 1.00 1.25 1.50 1.75 2.00
- 93 Analytical Flow (Double Gyre)

Left: DG FTLE ridges at $t_0 = 0$, integration time T = -10. Total runtime per iterate: ~0.424s (flowmap: ~0.390s; C eig: ~0.025s; FTLE ridge extraction: ~0.009s). Right: DG hyperbolic LCS at $t_0 = 0$, integration time T = -10. Total runtime per iterate: ~5.219s (flowmap (aux grid): ~1.83s; C eig (aux grid): ~0.039s; hyperbolic LCS extraction: ~3.350s). Both are computed over a 401x201 grid.





Left: QGE FTLE ridges at $t_0 = 0$, integration time T = 0.1. Total runtime per iterate: ~2.461s (flowmap: ~2.400s; C eig: ~0.038s; FTLE ridge extraction: ~0.023s). Middle: QGE hyperbolic OECS at $t_0 = 0.15$. Total runtime per iterate: ~2.238s (S eig: ~0.038s; hyperbolic OECS extraction: ~2.200s). Right: QGE elliptic OECS at $t_0 = 0.5$. Total runtime per iterate: ~0.0452s (IVD: ~0.0002s; elliptic OECS extraction: ~0.045s). All are computed over a 257x513 grid.

Jarvis, & Ross. (2025). NumbaCS: A fast Python package for coherent structure analysis. *Journal of Open Source Software*, ¿VOL?(¿ISSUE?), 3 7948. https://doi.org/10.xxxxx/draft.

¹Since many functions in NumbaCS are JIT compiled, these functions are optimized and compiled into machine code on the first function call. This initial delay is often referred to as "warm-up time". After the first call, subsequent function calls are much faster.



¹⁰⁸ Analytical Flow (Bickley jet)



Bickley jet elliptic LCS at $t_0 = 0$, integration time T = 40 days. Total runtime per iterate: $\sim 5.065s$ (flowmap: $\sim 4.490s$; LAVD: $\sim 0.565s$; elliptic LCS extraction: $\sim 0.010s$). Computed over 482x121 grid.



Numerical Flow (MERRA-2)

¹¹⁵ MERRA-2 FTLE ridges at $t_0 = 06/16/2020-00:00$, integration time T = -72hrs. Total runtime ¹¹⁶ per iterate: ~7.835s (flowmap: ~7.480s; C eig: ~0.085s; FTLE ridge extraction: ~0.27s). ¹¹⁷ Computed over 902x335 grid.

118 Datasets

Two datasets are provided with NumbaCS to test the functionality for flows defined by numerical 119 velocity data. One is a numerical simulation of the quasi-geostrophic equations (QGE). We 120 thank the authors of Mou et al. (2021) for providing us with this dataset, which was used 121 extensively during development, and allowing a piece of the dataset to be included in the 122 package. The full dataset was over the time span [10,81] with dt = 0.01. We provide the 123 velocity fields over the much shorter time span of [10,11] with the same dt. For details on 124 parameters used in the simulation, refer to the cited paper. The other dataset is a MERRA-2 125 vertically averaged reanalysis dataset (Gelaro et al., 2017; GMAO, 2015), which was used as 126 part of a paper (Jarvis et al., 2024) coauthored by the authors of this paper. Wind velocity 127 fields were vertically averaged over pressure surfaces ranging from 500 hPa to 800 hPa. The 128 corresponding latitude, longitude, and date arrays are also provided. All data can be downloaded 129 from the data folder on the GitHub page. 130

¹³¹ Usage in ongoing research

As of the writing of this paper, NumbaCS has not been public for long but has been utilized in one publication (Jarvis et al., 2024) where it was the computational tool for all coherent



structure methods. In addition, it is currently being used in an ongoing project focused on

airborne invasive species traveling from Australia to New Zealand titled "Protecting Aotearoa

- ¹³⁶ from wind-dispersed pests". This is a five year (October 2023 October 2028) Scion-led and
- ¹³⁷ Ministry of Business, Innovation and Employment (MBIE)-supported program.

Acknowledgments

This work was partially supported by the National Science Foundation (NSF) under grant
 number 1821145 and the National Aeronautics and Space Administration (NASA) under grant
 number 80NSSC20K1532 issued through the Interdisciplinary Research in Earth Science (IDS)

¹⁴² and Biological Diversity & Ecological Conservation programs.

143 **References**

- Bartos, A. P. E., Kaszás, B., & Haller, G. (2022). *Haller-group/TBarrier: TBarrier* (Version v1.0.0). Zenodo. https://doi.org/10.5281/zenodo.6779400
- Briol, F., & d'Ovidio, F. (2011). Lagrangian. In *GitHub repository*. GitHub. https://github.
 com/CNES/aviso-lagrangian
- Brunton, S. L., & Rowley, C. W. (2010). Fast computation of finite-time Lyapunov exponent
 fields for unsteady flows. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 20(1),
 017503. https://doi.org/10.1063/1.3270044
- ¹⁵¹ Curbelo, J., & Rypina, I. I. (2023). A Three Dimensional Lagrangian Analysis of the Smoke
 ¹⁵² Plume From the 2019/2020 Australian Wildfire Event. *Journal of Geophysical Research:* ¹⁵³ Atmospheres, 128(21), e2023JD039773. https://doi.org/10.1029/2023JD039773
- 154 Dabiri, J. (2009). LCS MATLAB Kit. https://dabirilab.com/software
- ¹⁵⁵ Du Toit, P. C. (2010). Transport and Separatrices in Time-Dependent Flows [PhD the ¹⁵⁶ sis, California Institute of Technology]. https://resolver.caltech.edu/CaltechTHESIS:
 ¹⁵⁷ 10072009-165901284
- ¹⁵⁸ Du Toit, P. C., & Marsden, J. E. (2010). Horseshoes in hurricanes. *Journal of Fixed Point* ¹⁵⁹ *Theory and Applications*, 7(2), 351–384. https://doi.org/10.1007/s11784-010-0028-6
- Farazmand, M., & Haller, G. (2012). Computing Lagrangian coherent structures from their
 variational theory. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 22(1), 013128.
 https://doi.org/10.1063/1.3690153
- Gelaro, R., McCarty, W., Suarez, M. J., Todling, R., Molod, A., Takacs, L., Randles, C. A.,
 Darmenov, A., Bosilovich, M. G., Reichle, R., Wargan, K., Coy, L., Cullather, R., Draper,
 C., Akella, S., Buchard, V., Conaty, A., Silva, A. M. da, Gu, W., ... Zhao, B. (2017). The
 Modern-Era Retrospective Analysis for Research and Applications, Version 2 (MERRA-2).
 Journal of Climate, 30(14), 5419–5454. https://doi.org/10.1175/JCLI-D-16-0758.1
- GMAO. (2015). Global Modeling and Assimilation Office, MERRA-2 inst3_3d_asm_Np: 3d,
 3-Hourly, Instantaneous, Pressure-Level, Assimilation, Assimilated Meteorological Fields
 V5.12.4, Greenbelt, MD, USA, Goddard Earth Sciences Data and Information Services
 Center (GES DISC), Accessed on 09-22-2023. https://doi.org/10.5067/QBZ6MG944HW0
- Günther, T., Horváth, Á., Bresky, W., Daniels, J., & Buehler, S. A. (2021). Lagrangian Coherent
 Structures and Vortex Formation in High Spatiotemporal-Resolution Satellite Winds of
 an Atmospheric Kármán Vortex Street. *Journal of Geophysical Research: Atmospheres*,
 126(19), e2021JD035000. https://doi.org/10.1029/2021JD035000
- Haller, G. (2011). A variational theory of hyperbolic Lagrangian Coherent Structures. *Physica D: Nonlinear Phenomena*, 240(7), 574–598. https://doi.org/10.1016/j.physd.2010.11.010



207

- Haller, G., & Beron-Vera, F. J. (2013). Coherent Lagrangian vortices: the black holes of 178 turbulence. Journal of Fluid Mechanics, 731, R4. https://doi.org/10.1017/jfm.2013.391 179
- Haller, G., Hadjighasem, A., Farazmand, M., & Huhn, F. (2016). Defining coherent vortices 180 objectively from the vorticity. Journal of Fluid Mechanics, 795, 136-173. https://doi.org/ 181 10.1017/jfm.2016.151 182
- Haller, G., & Poje, A. C. (1998). Finite time transport in aperiodic flows. Physica D: Nonlinear 183 Phenomena, 119(3), 352-380. https://doi.org/10.1016/S0167-2789(98)00091-8 184
- Jarvis, A., Hossein Mardi, A., Foroutan, H., & Ross, S. D. (2024). Atmospheric transport 185 structures shaping the "Godzilla" dust plume. Atmospheric Environment, 333, 120638. 186 https://doi.org/10.1016/j.atmosenv.2024.120638 187
- Junge, O., Diego, A. de, Karrasch, D., & Schilling, N. (2020). CoherentStructures.jl. In 188 GitHub repository. GitHub. https://github.com/CoherentStructures/CoherentStructures.jl 189
- Lagares, C., & Araya, G. (2023). A GPU-Accelerated Particle Advection Methodology for 190 3D Lagrangian Coherent Structures in High-Speed Turbulent Boundary Layers. Energies, 191 16(12), 4800. https://doi.org/10.3390/en16124800 192
- Lam, S. K., Pitrou, A., & Seibert, S. (2015). Numba: a LLVM-based Python JIT compiler. 193 Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC. 194 https://doi.org/10.1145/2833157.2833162 195
- Liu, Y., Wilson, C., Green, M. A., & Hughes, C. W. (2018). Gulf Stream Transport and Mixing 196 Processes via Coherent Structure Dynamics. Journal of Geophysical Research: Oceans, 197 123(4), 3014-3037. https://doi.org/10.1002/2017JC013390 198
- Mathur, M., Haller, G., Peacock, T., Ruppert-Felsot, J. E., & Swinney, H. L. (2007). Uncovering 199 the Lagrangian Skeleton of Turbulence. Phys. Rev. Lett., 98, 144502. https://doi.org/10. 200 1103/PhysRevLett.98.144502 201
- Mou, C., Wang, Z., Wells, D. R., Xie, X., & Iliescu, T. (2021). Reduced Order Models for 202 the Quasi-Geostrophic Equations: A Brief Survey. Fluids, 6(1). https://doi.org/10.3390/ 203 fluids6010016 204
- Nolan, P. (2024). Dynlab. In GitHub repository. GitHub. https://github.com/hokiepete/ 205 dynlab 206
- Nolan, P., Foroutan, H., & Ross, S. D. (2020). Pollution Transport Patterns Obtained Through Generalized Lagrangian Coherent Structures. Atmosphere, 11(2). https://doi.org/10.3390/ 208 atmos11020168 209
- Nolan, P., Serra, M., & Ross, S. D. (2020). Finite-time Lyapunov exponents in the instantaneous 210 limit and material transport. Nonlinear Dyn, 100, 3825-3852. https://doi.org/10.1007/ 211 s11071-020-05713-4 212
- Onu, K., Huhn, F., & Haller, G. (2015). LCS Tool: A computational platform for Lagrangian 213 coherent structures. Journal of Computational Science, 7, 26–36. https://doi.org/10.1016/ 214 j.jocs.2014.12.002 215
- Peacock, T., & Haller, G. (2013). Lagrangian coherent structures: The hidden skeleton of 216 fluid flows. Physics Today, 66(2), 41-47. https://doi.org/10.1063/PT.3.1886 217
- Pretorius, I., Schou, W. C., Richardson, B., Ross, S. D., Withers, T. M., Schmale III, D. G., & 218 Strand, T. M. (2023). In the wind: Invasive species travel along predictable atmospheric 219 pathways. Ecological Applications, 33(3), e2806. https://doi.org/10.1002/eap.2806 220
- Rutherford, B., Dangelmayr, G., & Montgomery, M. T. (2012). Lagrangian coherent structures 221 in tropical cyclone intensification. Atmospheric Chemistry and Physics, 12(12), 5483-5507. 222 https://doi.org/10.5194/acp-12-5483-2012 223



- Schindler, B., Peikert, R., Fuchs, R., & Theisel, H. (2012). Ridge Concepts for the Visualization
 of Lagrangian Coherent Structures. In R. Peikert, H. Hauser, H. Carr, & R. Fuchs
 (Eds.), Topological methods in data analysis and visualization II: Theory, algorithms,
- and applications (pp. 221–235). Springer Berlin Heidelberg. https://doi.org/10.1007/

- Serra, M., & Haller, G. (2016). Objective Eulerian coherent structures. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 26(5), 053110. https://doi.org/10.1063/1.4951720
- Serra, M., Sathe, P., Beron-Vera, F., & Haller, G. (2017). Uncovering the Edge of the Polar
 Vortex. Journal of the Atmospheric Sciences, 74(11), 3871–3885. https://doi.org/10.
 1175/JAS-D-17-0052.1
- Shadden, S. C., Lekien, F., & Marsden, J. E. (2005). Definition and properties of Lagrangian
 coherent structures from finite-time Lyapunov exponents in two-dimensional aperiodic flows.
 Physica D: Nonlinear Phenomena, 212(3), 271–304. https://doi.org/10.1016/j.physd.2005.
 10.007
- Steger, C. (1998). An unbiased detector of curvilinear structures. *IEEE Transactions on Pattern* Analysis and Machine Intelligence, 20(2), 113–125. https://doi.org/10.1109/34.659930
- Winant, P., Coleman, C., & Lyon, S. (2017). Interpolation.py. In *GitHub repository*. GitHub.
 https://github.com/EconForge/interpolation.py
- Wogan, N. (2021). Numbalsoda. In *GitHub repository*. GitHub. https://github.com/
 Nicholaswogan/numbalsoda

⁹⁷⁸⁻³⁻⁶⁴²⁻²³¹⁷⁵⁻⁹_15