

# OceananigansLagrangianFilter.jl:

10 February 2026

## Summary

Numerical fluid dynamical simulations of geophysical fluids, such as the ocean and the atmosphere, typically exhibit complex dynamics including vortices and waves. The analysis of these simulations often requires the high frequency component (usually the waves) to be removed from the output to allow analysis of the lower frequency component, which we refer to as the ‘mean flow’. Alternatively, the faster component may be of interest, and must be separated from the mean flow for analysis purposes. Eulerian temporal filtering (or equivalently ‘time averaging’) is most commonly used for this purpose – at a given spatial location, a time average (perhaps with some weight function) can be found either by post-processing simulation output ‘offline’, or more commonly at simulation time by constructing the average cumulatively ‘online’. However, it has long been recognised that a Lagrangian time average, in which the property of interest is averaged in the frame of a (moving) fluid parcel, defines a more physically relevant mean flow (Andrews and McIntyre 1978). Furthermore, Lagrangian filtering, whereby waves are defined by an intrinsic (frame of flow) frequency criterion can most effectively decompose waves and mean flows (Baker et al. 2025). Lagrangian filtering is usually performed using particle tracking methods, but its use has so far been limited by the extensive computational and memory requirements of the calculation.

In this package, we implement methods for Lagrangian filtering based on the solution of partial differential equations (PDEs), building on recent theoretical developments (Kafiabad and Vanneste 2023; Baker et al. 2025; Minz et al. 2025). Although these methods are entirely generalisable to different numerical fluid dynamical solvers, here we implement them for use with the `Oceananigans.jl` ocean modelling framework. `Oceananigans.jl` is a popular and flexible Julia-based open source software package for finite volume simulations of the nonhydrostatic and hydrostatic Boussinesq equations on CPUs and GPUs (Ramadhan et al. 2020). `OceananigansLagrangianFilter.jl` provides a user-friendly way for `Oceananigans.jl` users to analyse their simulations and gain new insights using Lagrangian filtering, either by post-processing their existing simulation output (‘offline’) or integrating the functionality into their simulations at run-

time ('online').

## Statement of Need

Lagrangian filtering consists of finding, for some simulated scalar field  $f(x, t)$ , the generalised Lagrangian mean  $\bar{f}^L(\mathbf{x}, t)$  defined by

$$\bar{f}^L(\bar{\varphi}(\mathbf{a}, t), t) = \int_{-\infty}^{\infty} G(t-s)f(\varphi(\mathbf{a}, s), s) ds,$$

where  $\varphi(\mathbf{a}, t)$  is the flow map, defining the position of a fluid particle with label  $\mathbf{a}$  at time  $t$ ,  $G(t)$  is some weight function (or equivalently the impulse response of the filter), and  $\bar{\varphi}(\mathbf{a}, t)$  is the mean flow map, defined by

$$\bar{\varphi}(\mathbf{a}, t) = \int_{-\infty}^{\infty} G(t-s)\varphi(\mathbf{a}, s) ds.$$

While  $\bar{f}^L$  is the true generalised Lagrangian mean as it is defined at the mean position of a trajectory, we also define a spatial rearrangement  $f^*$  that is instead defined on the trajectory itself

$$f^*(\varphi(\mathbf{a}, t), t) = \int_{-\infty}^{\infty} G(t-s)f(\varphi(\mathbf{a}, s), s) ds,$$

and a map

$$\Xi(\varphi(\mathbf{a}, t), t) = \bar{\varphi}(\mathbf{a}, t),$$

that maps between  $f^*$  and  $\bar{f}^L$ , such that  $\bar{f}^L(\Xi(\mathbf{x}, t), t) = f^*(\mathbf{x}, t)$ .

`Oceananigans.LagrangianFilter.jl` solves directly for  $f^*$  for any specified variables  $f$  (including velocity components), which may be sufficient if  $\Xi$  is close to the identity, or for certain wave decompositions (Baker et al. 2025). Optionally, it also solves for  $\Xi$  and afterward performs a spatial interpolation to recover  $\bar{f}^L$ . The weight function  $G(t)$  is constructed from sums of (complex) exponentials, which allows closed PDE systems for  $f^*$  and  $\Xi$  (Minz et al. 2025) (see figure Figure 3).

Lagrangian filtering is superior to Eulerian filtering in two primary situations. The first is when waves are Doppler-shifted by the mean flow, such that they can be low frequency in the rest frame of the simulation. The canonical example of this is internal lee waves, which are generated by steady stratified flow over topography. Figure Figure 2a shows the horizontal velocity field  $u$  generated by a steady flow  $U = 0.1 \text{ ms}^{-1}$  with stratification  $N = 0.001 \text{ s}^{-1}$  over a Gaussian bump, which was run in 2D using the `HydrostaticFreeSurfaceModel` of `Oceananigans.jl`.

The waves are phase-locked to the topography such that their frequency in the rest frame is zero. When an Eulerian average is taken, the waves therefore remain in the mean component as shown in figure Figure 2b. However, internal gravity waves are subject to a criterion  $\omega > f$  on their intrinsic frequency  $\omega$ , where  $f$  is the Coriolis parameter, which means that they can be filtered from the mean flow using a Lagrangian filter that removes high frequencies. Figure Figure 2c shows  $u^*$ , demonstrating that the wave perturbations have been removed - note the order of magnitude difference in the colorbar range between the top and bottom rows. The effect of the wave displacement remains in  $u^*$ , as visualised by contours of the Lagrangian filtered buoyancy  $b^*$  in figure Figure 2c. However, when  $u^*$  and  $b^*$  are remapped to  $\bar{u}^L$  and  $\bar{b}^L$  respectively, the waves are entirely removed (figure 2d).

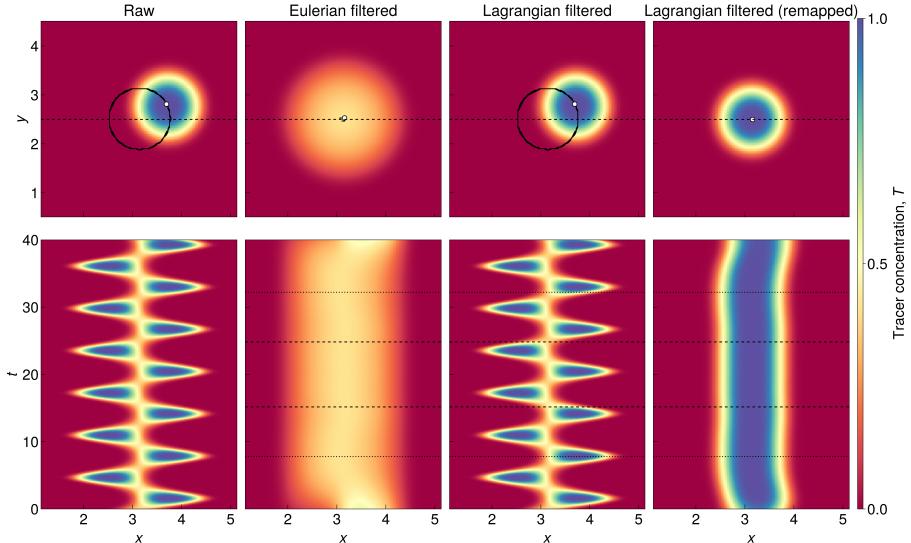


Figure 1: Here's a caption.

The second use-case is when the wave (or other high frequency oscillation) is large compared to the scales of the mean flow – for example, the displacement of a submesoscale eddy field by an inertial oscillation, or tidal displacements of coastal waters. Figures Figure 1a,e show an idealised and non-dimensional `Oceananigans.jl` simulation using the `ShallowWaterModel`, in which a Gaussian blob of tracer  $T$  is advected in a circular path by a spatially uniform inertial oscillation. The Eulerian average (figures Figure 1b,f) leads to blurring of the tracer gradients due to its high-frequency displacement. The Lagrangian average  $T^*$  (figures Figure 1c,g) is in this case identical to the raw field  $T$  (figures Figure 1a,e) since the tracer is conserved along a trajectory, but when remapped to  $\bar{T}^L$  (figures Figure 1d,h) the effect of the displacement is removed and the

tracer maintains its original shape, as in the absence of the inertial oscillation.

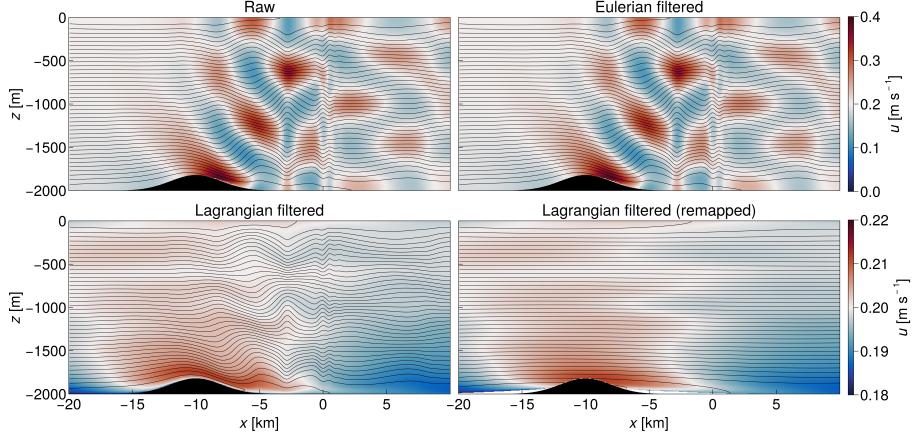


Figure 2: Here's a caption.

Explain here briefly the benefit of being able to make this separation (cite other filtering papers) - identification of internal waves in simulations.

## State of the field

- Generally particle tracking, both online and offline. Online methods often suffer from clustering and interpolation errors, offline from memory requirements. In particular, it is generally necessary to track particles along an entire trajectory to find the Lagrangian mean at one timestep.
- Also explain the online/offline nature and why the flexibility is good.
- This method allows calculation of the Lagrangian mean and or high frequency component at every timestep. Talk about relation to Shakespeare 2021 particle tracking.

## Software design

- Several options for implementation of Lagrangian mean via PDEs - why I chose this one and offline and online options.
- Offline and online (include figure of filter shapes) different uses in different contexts. Keep as much of Oceananigans infrastructure as possible, building on the existing constructors.
- Refer to equations? To demonstrate what is actually found by different methods, cite Baker 25 for f. *It is a choice to find f first* (no interpolation)

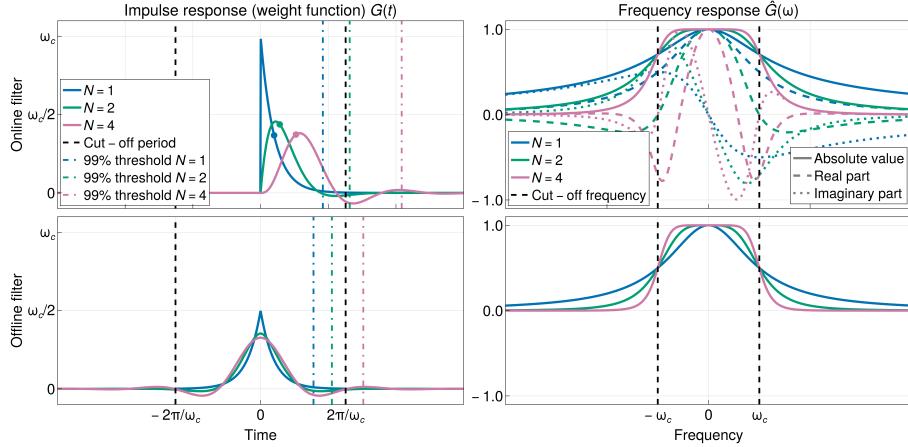


Figure 3: Caption for filter shape.

## Research impact statement

- Makes available for implementation in other fluid dynamical solvers
- Analytical methods have been published, but this implementation allows the methods to be easily accessible to the community.
- Useful for calculating Lagrangian means for saving as transport velocities

## AI usage disclosure

## Acknowledgements

Thank NFFDy. We want to thank the Climate Modeling Alliance team and `Oceananigans.jl` contributors for their fantastic project. We also thank `JuliaGPU` contributors for `KernelAbstractions.jl` and `CUDA.jl` which make this code possible.

## References

- Andrews, D. G., and M. E. McIntyre. 1978. “An Exact Theory of Nonlinear Waves on a Lagrangian-mean Flow.” *Journal of Fluid Mechanics* 89 (4): 609–46.
- Baker, Lois E, Hossein A Kafiabad, Cai Maitland-Davies, and Jacques Vanneste. 2025. “Lagrangian Filtering for Wave–Mean Flow Decomposition.” *Journal of Fluid Mechanics* 1009: A40. <https://doi.org/10.1017/jfm.2025.42>.
- Kafiabad, Hossein A., and Jacques Vanneste. 2023. “Computing Lagrangian Means.” *Journal of Fluid Mechanics* 960 (April): A36. <https://doi.org/10.1017/jfm.2023.228>.

- Minz, Abhijeet, Lois E. Baker, Hossein A. Kafiabad, and Jacques Vanneste. 2025. “Efficient Lagrangian Averaging with Exponential Filters.” *Phys. Rev. Fluids* 10 (7): 074902. <https://doi.org/10.1103/4d4f-38wx>.
- Ramadhan, Ali, Gregory LeClaire Wagner, Chris Hill, Campin Jean-Michel, Valentin Churavy, Andre Souza, Adam Edelman, Raffaele Ferrari, and John Marshall. 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>.