Lagrangian Filtering: A novel method for separating internal waves from non-wave flows in high-resolution simulations

Callum J. Shakespeare[1], Angus H. Gibson[2], Andrew McC. Hogg[2], Shane Richard Keating[3], Scott Daniel Bachman[4], and Nick Velzeboer[2]

[1]Autralian National University
[2]Australian National University
[3]University of New South Wales
[4]NCAR

November 23, 2022

Abstract

Identifying internal waves in complex flow fields is a long-standing problem in fluid dynamics, oceanography and atmospheric science, owing to the overlap of internal waves temporal and spatial scales with other flow regimes. Lagrangian filtering — that is, temporal filtering in a frame of reference moving with the flow — is one proposed methodology for performing this separation. Here we (i) describe a new implementation of the Lagrangian filtering methodology and (ii) introduce a freely available, parallelised Python package that applies the method. We show that the package can be used to directly filter output from a variety of common ocean models including MITgcm, ROMS and MOM5 for both regional and global domains at high resolution. The Lagrangian filtering is shown to be a useful tool to both identify (and thereby quantify) internal waves, and to remove internal waves to isolate the 'balanced' flow field.

Lagrangian Filtering: A novel method for separating internal waves from non-wave flows in high-resolution simulations

Callum J. Shakespeare[1,2], Angus H. Gibson[1], Andrew McC. Hogg[1,2], Scott D. Bachman[3], Shane R. Keating[2,5], and Nick Velzeboer[1,2]

[1]Research School of Earth Sciences, Australian National University, Canberra, Australia.
[2]ARC Centre of Excellence in Climate Extremes, Australian National University, Canberra, Australia.
[3]National Center for Atmospheric Research, Boulder, CO, USA.
[4]School of Mathematics and Statistics, University of New South Wales, Sydney, Australia.
[5]ARC Centre of Excellence in Climate Extremes, University of New South Wales, Sydney, Australia.

Key Points:

- An improved method is proposed for filtering internal waves from high-resolution numerical model output.
- An open-source parallelised Python package is available that implements the method.
- The package may be applied to directly filter output from a number of common ocean models including MITgcm, ROMS and MOM5.

Corresponding author: Callum J. Shakespeare, `callum.shakespeare@anu.edu.au`

Abstract

Identifying internal waves in complex flow fields is a long-standing problem in fluid dynamics, oceanography and atmospheric science, owing to the overlap of internal waves temporal and spatial scales with other flow regimes. Lagrangian filtering — that is, temporal filtering in a frame of reference moving with the flow — is one proposed methodology for performing this separation. Here we (i) describe a new implementation of the Lagrangian filtering methodology and (ii) introduce a freely available, parallelised Python package that applies the method. We show that the package can be used to directly filter output from a variety of common ocean models including MITgcm, ROMS and MOM5 for both regional and global domains at high resolution. The Lagrangian filtering is shown to be a useful tool to both identify (and thereby quantify) internal waves, and to remove internal waves to isolate the 'balanced' flow field.

Plain Language Summary

Ocean flows are a superposition of many different flow phenomena including eddies, jets, currents, and waves. As computing power increases, high-resolution numerical ocean models are simultaneously resolving more of these phenomena. Quantifying a particular phenomenon therefore requires a method to identify and separate that phenomenon from others in the model output, in order to be able to assess the associated energy and energetic exchanges. Here we propose a method to identify a particular phenomenon known as 'internal waves' — hourly to daily oscillatory motions which propagate three-dimensionally throughout the ocean, driving mixing and thereby supporting the global ocean circulation. Our method involves a combination of a coordinate transformation and a high-pass temporal filter. The method has been implemented in a freely available, efficient and user-friendly open-source Python package.

1 Introduction

Internal waves are the natural response of a stratified fluid to thermal or mechanical perturbations, whether periodic (e.g. tides) or localised in time (e.g. sudden wind gusts or thermal forcing), and are therefore ubiquitous in the ocean. Internal waves are associated with significant fluxes of energy (e.g. Waterhouse et al., 2014) and momentum (e.g. Naveira Garabato et al., 2013; Shakespeare & Hogg, 2019) that act to mix and force the ocean. The quantification of internal waves — and their attendant fluxes — is therefore of significant interest to the oceanographic community and has been the focus of many numerical modelling campaigns in recent years. Quantifying internal wave fluxes in the output of such models requires first identifying and separating the wave component of the flow from other signals.

In lower resolution models, internal waves are readily identified as high-frequency (sub-daily) motion, as compared to the much slower (monthly to yearly) balanced flow consisting of currents, jets and mesoscale eddies (e.g. the 0.25° model of Simmons & Alford, 2012). In such models, which do not resolve the high frequency ocean submesoscale (usually identified as sub-10km horizontal scales and daily timescales; e.g. Thomas et al., 2008; Shcherbina et al., 2013), internal waves are the only high-frequency signal, making their identification straightforward via a direct temporal filter at fixed points in space (an Eulerian filter). However, as computer power increases, models are simultaneously resolving both submesoscales and smaller scale internal waves. Indeed, sub-1km horizontal resolution regional simulations have become commonplace in recent years (e.g. Capet, McWilliams, & Shchepetkin, 2008; Capet, McWilliams, Molemaker, & Shchepetkin, 2008a,b; Nikurashin et al., 2013; Nagai et al., 2015; Mashayek et al., 2017; Shakespeare & Hogg, 2017; Bachman et al., 2020, among others). The identification of internal waves in such high-resolution models is challenging because (i) the internal wave timescales overlap with those of the submesoscale, and (ii), the propagation of internal waves at these scales is strongly modified by the flow in which they propagate. In particular, Doppler shifting of the wave frequency (e.g. Bretherton & Garrett, 1969) can mean that the Eulerian frequency (the frequency observed at a fixed point) is very low, or even zero, comparable to that of the balanced flow (e.g. jets, eddies). Thus, an Eulerian high-pass filter is insufficient to identify small-scale waves in high-resolution simulations.

Instead, it is desirable to use the more sophisticated definition of an internal wave as 'a high-frequency motion as measured moving with the flow' (e.g. Polzin & Lvov, 2011). Shakespeare & Hogg (2017) — and separately Nagai et al. (2015) — introduced an internal wave filtering methodology using this definition which they called 'Lagrangian filtering'. This method involves transforming model output to a flow-following (i.e. Lagrangian) reference frame and temporally filtering fields in this frame, before transforming the data back to fixed points. In both Nagai et al. (2015) and Shakespeare & Hogg (2017), this method was introduced to address the challenge of quantifying the 'spontaneous' generation of internal waves — a process by which internal waves emerge from a flow field without external forcing. Usually such waves emerge from high Rossby number flows (i.e. a strong submesoscale is present) and with small scales (i.e. large Doppler shifts), making their identification impossible using an Eulerian filter. The Lagrangian filtering methodology addresses both of these challenges since the frequencies are measured in a frame moving with the total flow, including the submesoscale flow, allowing unique identification of the wave field as the high-frequency signal in this flow-following frame.

More recently, the Lagrangian filtering methodology has been applied to quantify internal tide energy and momentum fluxes in tidally-forced, submesoscale-resolving simulations — in both idealised (Shakespeare & Hogg, 2019) and realistic regional (Bachman et al., 2020) configurations. The latter study investigated internal tides in a highly dynamic region (the Indonesian Seas), characterised by fast and strongly divergent horizontal flows. Such divergence introduces additional complications for the Lagrangian filtering methodology developed by Shakespeare & Hogg (2017) since it leads to a degradation of parcel concentrations in divergent regions, and an accumulation in convergent regions, effectively degrading the resolution of the filtered data (see Bachman et al., 2020, for details). Here we present a modification of the method that solves this problem and maintains the input resolution of the model data at all locations.

The paper is laid out as follows. In Section 2 we describe the new Lagrangian filtering methodology and identify the differences from previous formulations. We also provide a technical description of the Python package implementation of the method, and demonstrate its operation via a simple synthetic flow example. Section 3 then presents multiple case studies using the Python package to directly filter output from three widely used ocean models (MITgcm, ROMS, and MOM5) and demonstrates the utility of the method for both the problem of identifying wave fields and the converse problem of eliminating waves in order to identify the 'balanced' flow. We then conclude in Section 4.

2 Method

The separation of wave and non-wave flows in high-resolution models is a challenging problem since both flows often occupy the same spatial and temporal scales. This is increasingly a major issue as models begin to be routinely run in submesoscale-permitting regimes (i.e. horizontal resolution of hundreds of metres to a few kilometres). In such regimes, neither a purely temporal — nor a purely spatial — filtering method is sufficient. Instead, here we describe a spatio-temporal filtering method called 'Lagrangian filtering' (Nagai et al., 2015; Shakespeare & Hogg, 2017, 2018, 2019; Bachman et al., 2020) since it involves temporal filtering in a Lagrangian (flow-following) reference frame. In such a frame, motions such as eddies appear as very low frequencies because the parcels are following their flow. By contrast, internal waves have a fixed minimum frequency (the Coriolis frequency, f) in a flow-following frame as imposed by the internal gravity wave dispersion relation (e.g. Bühler, 2014)

$$f \leq \omega = \sqrt{\frac{N^2(k^2+l^2)+f^2m^2}{k^2+l^2+m^2}} \leq N, \tag{1}$$

where ω is the Lagrangian frequency, N the buoyancy frequency, and (k, l, m) the wavevector in the Cartesian (x, y, z) directions. In practice, the models run in these regimes are usually hydrostatic ($k^2 + l^2 \ll m^2$), and the appropriate dispersion relation is thus

$$f \leq \omega = \sqrt{\frac{N^2(k^2+l^2)+f^2m^2}{m^2}} < \infty, \tag{2}$$

and the upper limit on wave frequencies (N) is not present. In either case, in a flow-following frame, internal waves exist only at frequencies exceeding the local Coriolis frequency, and non-wave flows exist at very low (near-zero) frequencies. Thus, a scale separation exists and the high-frequency wave component may be obtained via a high-pass filter.

We use standard Lagrangian tracking algorithms to follow fluid parcels that are advected by flow fields from ocean model output. The Lagrangian filtering method is implemented as follows:

1. Initialise parcels on the model grid at the time of interest.
2. Advect parcels forward and backwards using model output of horizontal velocity (u, v) for a given time window (user-specified; often 2 days).
3. Interpolate fields of interest (e.g. u, v, ρ, p, etc.) to the parcel paths.
4. Apply high-pass filter to data along parcel paths (user-specified cut-off frequency; usually close to f).
5. Save the filtered values at the time of interest (at the initial grid locations, which is the time-centred parcel position). Discard all other values.
6. Repeat for next time of interest.

This procedure can be repeated for all required time outputs to obtain a full timeseries of filtered fields. As with previous implementations, parcels are only advected horizontally on the basis that the vertical advection (and thus wave Doppler shift) is small; that is, $mW \ll kU+lV$ for a mean flow (U, V, W). With this assumption, the filter for each vertical level of the model is independent and the filtering calculation can be trivially parallelised over times and model levels. This approach works well for models that use depth coordinates (z, z^*) and likely also (though this remains to be tested) for models that use isopycnal coordinates, since ocean flows are along isopycnals to first order. However, this approach will fail for models that use terrain-following (σ) or similar coordinates, since the cross-coordinate flow can be large. For such models, output must first be interpolated to depth coordinates before applying the filtering (e.g. as done by Bachman et al., 2020).

A key improvement to our implementation of the Lagrangian filtering method compared with previous implementations (Shakespeare & Hogg, 2017, 2018, 2019; Bachman et al., 2020) is the forward-backward advection from the model gridpoints (step 2) which allows the time-centred data to be naturally defined on the model grid (step 5), without interpolation. Previous implementations kept all data along parcel tracks and reversed interpolated from scattered parcel positions to the model grid. The interpolation operation is computationally expensive, and more importantly, leads to the degradation of the resolution of the filtered fields in regions of strongly divergent flow (e.g. as noted by Bachman et al., 2020). Using the new method, the resolution of the filtered fields remains identical to the input (raw model) fields.

2.1 The Lagrangian filtering Python package

The Lagrangian filtering algorithm is encapsulated in the `lagrangian-filtering` library (`https://github.com/angus-g/lagrangian-filtering`), written in Python and available within the Conda package manager. This library makes use of the Parcels framework (Delandmeter & van Sebille, 2019), which handles some of the computational challenges involved. In particular, Parcels provides a unified interface to different model outputs, support for sampling on both rectilinear and curvilinear meshes, and dynamic just-in-time sampling kernels. On top of Parcels, the Lagrangian filtering library provides another level of abstraction. The library exposes a simple interface to automatically define the relevant Lagrangian sampling kernels, as well as the full filtering workflow. For a given time slice, the library can seed the initial particles, perform the forward and backward advection steps, then perform the final filtering reduction on the Lagrangian data. In order to implement the filtering workflow efficiently, the Lagrangian filtering library has a strong focus on performance, in both the advection and filtering stages.

The Lagrangian transformation using forward and backward advection is a large component of the computational effort of the filtering algorithm, and thus its performance is paramount. To improve raw advection performance, in parallel with the Lagrangian filtering development, Parcels was translated to a structure-of-arrays representation (Kehl et al., 2021). This change also permitted Parcels to process large numbers (tens of millions) of particles, which was previously untenable. To make use of modern multicore hardware, the main advection loop in Parcels was also converted to OpenMP shared-memory parallelism.

Excluding the advection portion of the filtering algorithm, there are other performance considerations. As the algorithm needs to store the full trajectory for every particle, it may have large memory requirements for certain datasets. To prevent excessive memory use, the filtering library is able to cache the advection trajectories on-disk in HDF5 files. The approach of reducing the full trajectories back to a single, filtered time-slice of data is able to take maximal advantage of parallel processing, and we use Dask (Dask Development Team, 2016) to distribute the filtering across multiple workers. An additional library (sosfilt; `https://github.com/angus-g/sosfilt`) has been written to allow the efficient application of multiple second-order section (SOS) filters at different cutoff frequencies (see Section 3.3 for an example application), which is otherwise a very inefficient operation in the base scipy (Virtanen et al., 2020) implementation.

2.2 Synthetic flow example

To build understanding of the utility of the method, here we consider a simple synthetic flow example. The Python code to generate this example is provided via github (see Acknowledgments). Consider a doubly periodic $x - y$ Cartesian plane, where

$$0 \le x \le L, \quad 0 \le y \le L, \tag{3}$$

with constant Corolis parameter f. Here we construct a flow field that is a combination of a uniform mean flow, $\mathbf{u}_0 = (U_0, 0, 0)$, a stationary trapped wave, $\mathbf{u}_w = (v_w, u_w, w_w)$, and an eddy being advected by the mean flow, $\mathbf{u}_e = \nabla \times \psi \hat{\mathbf{z}}$, such that

$$u = U_0 + u_w \sin kx + \frac{\partial \psi}{\partial y}, \tag{4}$$

$$v = v_w - \frac{\partial \psi}{\partial x}. \tag{5}$$

The wave field is defined such that it satisfies the steady momentum equations linearised about the uniform mean flow (as per the typical lee wave problem), with no variation in the y direction:

$$U_0 \frac{\partial u_w}{\partial x} - f v_w = -\frac{\partial p_w}{\partial x}, \tag{6}$$

$$U_0 \frac{\partial v_w}{\partial x} + f u_w = -\frac{\partial p_w}{\partial y}, \tag{7}$$

$$\frac{\partial u_w}{\partial x} + \frac{\partial v_w}{\partial y} + \frac{\partial w_w}{\partial z} = 0. \tag{8}$$

We choose $u_w = U_w \sin kx$ and therefore $v_w = \frac{fU_w}{kU_0} \cos kx$. This steady flow is a wave signal for wavenumbers k where the Doppler shift exceeds the inertial frequency, or $kU_0 > f$. The pressure and vertical velocity are unimportant to the current example. The eddy is constructed as a barotropic flow with streamfunction

$$\psi = U_e L_e \exp\left(-((x - U_0 t)^2 + (y - L/2)^2)/L_e^2\right), \tag{9}$$

where U_e is the maximum eddy flow speed and L_e is a measure of the radius. All that remains is to select appropriate values for the various constants. Here we choose a mean flow of $U_0 = 0.2$m/s, wave and eddy flows of $U_e = U_w = 0.01$m/s, and a Coriolis parameter of $f = 5 \times 10^{-5}$/s. The wavenumber is chosen to be $k = 10^{-3}$/m and the domain equal to 8 wavelengths, $L = 8(2\pi/k) = 50.265$km. Lastly, the eddy radius is taken as $L_e = 10$km. The effective frequency of the eddy is set by the rate of advection past a fixed point, or $U_e/L_e = 2 \times 10^{-5}$/s.

To generate the synthetic data we write a netCDF file consisting of 2 weeks of output on a 200 × 200 grid ($\Delta x = 250$m horizontal resolution) at hourly time intervals. The required frequency of output is set by the need to resolve the timescales of the flow; here the timescales are set by the eddy advection and are of order $L_e/U_e = 14$ hours. To maintain the effective spatial resolution during advection (a CFL criteria for advection; e.g. Keating et al., 2011) we require an advection timestep smaller than $\Delta x/U_{max} =$ 1142 s for the maximum flow speed of $U_{max} = 0.22$m/s. We therefore select an advection timestep of 600 seconds (the Lagrangian filter will interpolate between the hourly data).

The Lagrangian filter is applied to the synthetic data with a cut-off frequency equal to the minimum wave frequency of f and a 7-day (±3.5) time window. The Lagrangian filtered output is shown in Figure 1 for the u velocity at $t = 7$ days. The method cleanly separates the steady wave flow (Figure 1b) from the eddy flow (Figure 1a). For comparison, the results from an Eulerian filter are also shown in Figure 1c,d. The Eulerian filter is identical to the Lagrangian except it is applied for time series at fixed points in space, rather than time series along parcel tracks. Unsurprisingly, the Eulerian filter identifies almost the entire flow as low-frequency 'mean' flow, except for the smaller scale part of the eddy which appears as a relatively low-amplitude high-frequency 'wave' flow. This splitting of a particular dynamical feature of the flow (i.e. the eddy) is a common issue with filtering approaches where there is no physical temporal (or spatial) scale separation between the dynamical regimes, and thus any choice of filter cut-off is a largely arbitrary one. Conversely, the Lagrangian filter succeeds because there is a clear temporal scale separation in the flow-following frame between the low frequency eddy and high-frequency wave.

3 Model case studies

Here we present three examples of the application of the Python package to filter output from three widely used ocean models. The Python code for each example is provided via github (see Acknowledgments).

3.1 Nested Scotia Sea model at 200m resolution (MITgcm)

This Massachusetts Institute of Technology General Circulation Model (MITgcm; Marshall et al. (1997)) configuration covers a 250km by 250km region of the Scotia Sea in the Antarctic Circumpolar Current downstream of Drake Passage at 200m horizon-

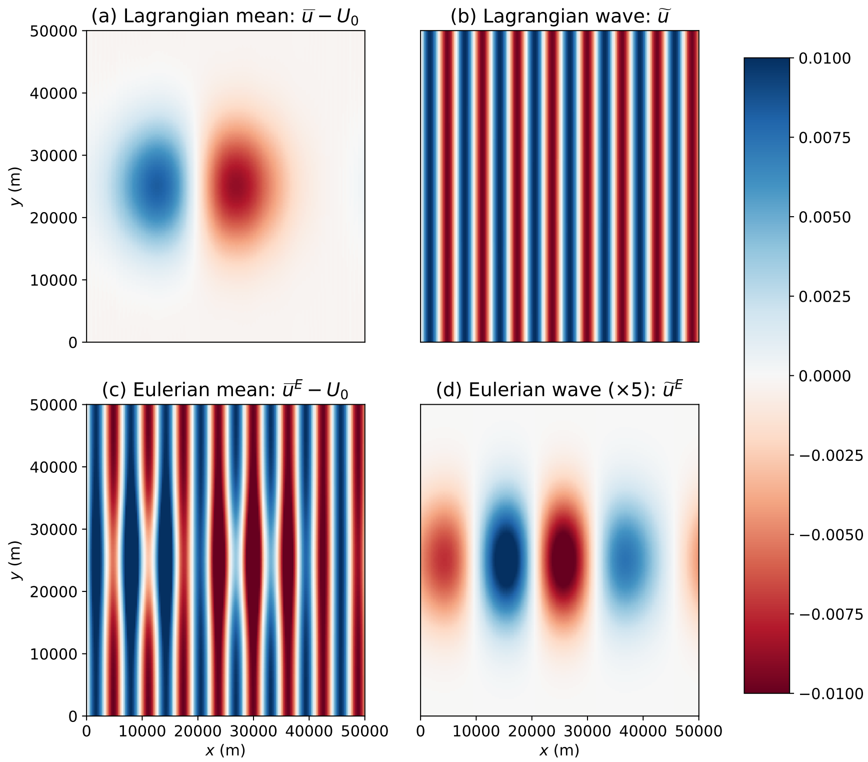

Figure 1. Filtered zonal flow fields for the synthetic data described in the text at time $t = 7$ days. (a) The Lagrangian filtered mean/eddy flow, $\bar{u} - U_0$. (b) The Lagrangian filtered wave flow, $\widetilde{u}$. (c) The Eulerian mean flow, $\bar{u}^E - U_0$. (b) The Eulerian filtered wave flow, $\widetilde{u}^E$.

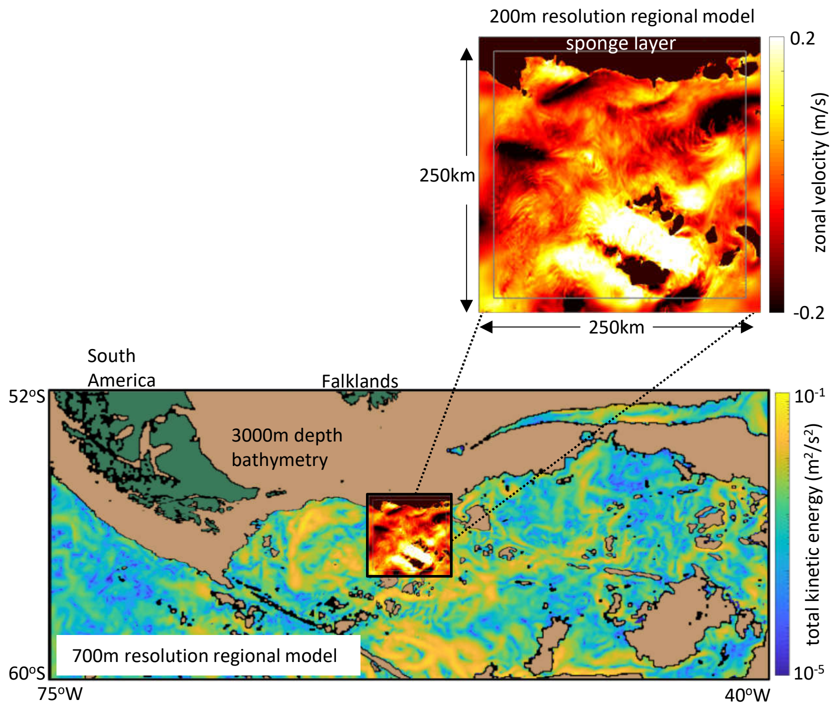

Figure 2. Regional models of the Southern Ocean downstream of Drake Passage. The 200 m resolution, 200 vertical level MITgcm regional configuration used here (top right) is nested within the lower resolution configuration of Velzeboer (2019). The boundaries are forced by 20 km -wide sponges (indicated). Zonal velocity (inset) and kinetic energy (main) are shown at 3000m depth. Surface (green) and 3000 m (brown) bathmetry contours are also shown.

tal resolution, with 200 vertical levels (Figure 2). The model is forced by a tidal potential from TPX08 global tidal model (Egbert & Erofeeva, 2002) including 8 constituents (M2, S2, N2, K2, K1, O1, P1, and Q1) and surface forcing from hourly ERA5. In addition, forcing at the edges of the domain is provided by hourly output from a larger-domain MITgcm regional model at 700m resolution (Velzeboer, 2019), which includes the same tides and ERA5 surface forcing, and is itself forced at the boundary by the Mercator GLORYS12V1 reanalysis product (`http://marine.copernicus.eu/`) and tidal velocities from TPX08. Further details and analysis of the regional model will be reported separately. Here we instead focus on the application and utility of the Lagrangian filtering for a model of this type (i.e. a relatively small-domain regional model with open boundaries). We consider one week of model output from 14-21 December 2010. The frequency cut-off is selected as the Coriolis frequency (constant for the f-plane model) of $f = 1.206 \times 10^{-4}$ s^{-1}.

3.1.1 Impact of width of filter time window

A key parameter in applying the Lagrangian filter is selecting the length of the data window about the time of interest. In a Fourier sense, the length of the time window T determines the frequency-space resolution (i.e. $\Delta\omega = 2\pi/T$). It is anticipated that the

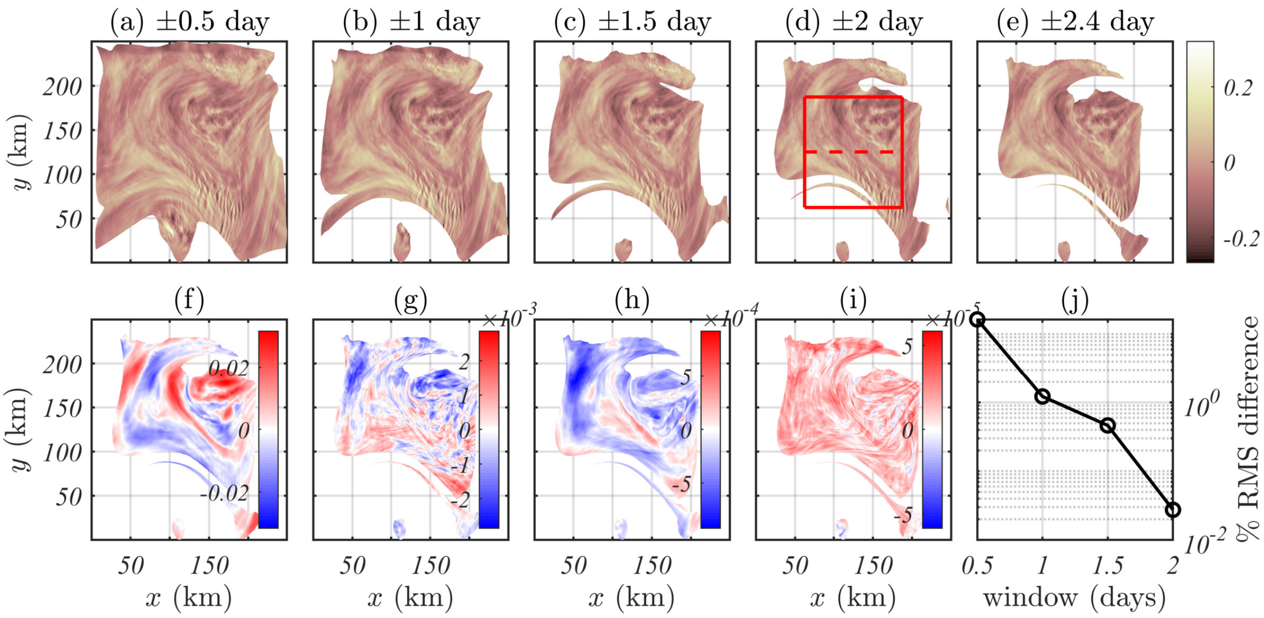

Figure 3. The impact of filtering window width on the calculated wave field. (a-e) Snapshots of filtered wave zonal velocity at 12 noon UTC on 16 Dec 2010 for various window widths. (f-i) Difference in filtered wave velocity with respect to the ±2.4 day window. (j) Percentage root-mean-square (RMS) difference in wave field for each window, compared with the ±2.4 day window. The red square indicates the region shown in Figure 4a-d, and the red dashed line the transect in Figure 4e-f.

filtered fields will converge as the resolution of the filter (window width) increases. Thus, the question we address here is: What window width shows sufficient convergence of the filtered fields for practical applications?

Figure 3a-e displays the filtered zonal velocity fields for 5 different time window widths from ±0.5 to ±2.4 days. By eye, the filtered fields look almost identical in each case with the only obvious difference being that longer filter windows produce a smaller region of usable data, since parcels reach the domain edges within the window period and are omitted from the output. However, differences in the filtered fields do exist: to highlight these, the difference of each filtered field with respect to the longest (±2.4 day) window is also shown. Differences reduce by approximately an order of magnitude for each 1 day increase in filter window width. The RMS difference (Figure 3j) is 10% for a ±0.5 day window, 1% for a ±1 day window and 0.03% for a ±2 day window. Based on these results, a ±2 day window is clearly more-than-satisfactory for the practical application of the Lagrangian filtering method — and we therefore use a ±2 day window in our further analysis below.

The Lagrangian filtering package also has the ability to use a variable or adaptive window length. In this mode, instead of omitting parcels that leave the domain, the package will truncate the time series at the time of exit, and filter based on the reduced time series. This approach will increase the error, but may be the best choice for very small domains where no significant loss of model data around the edges can be afforded.

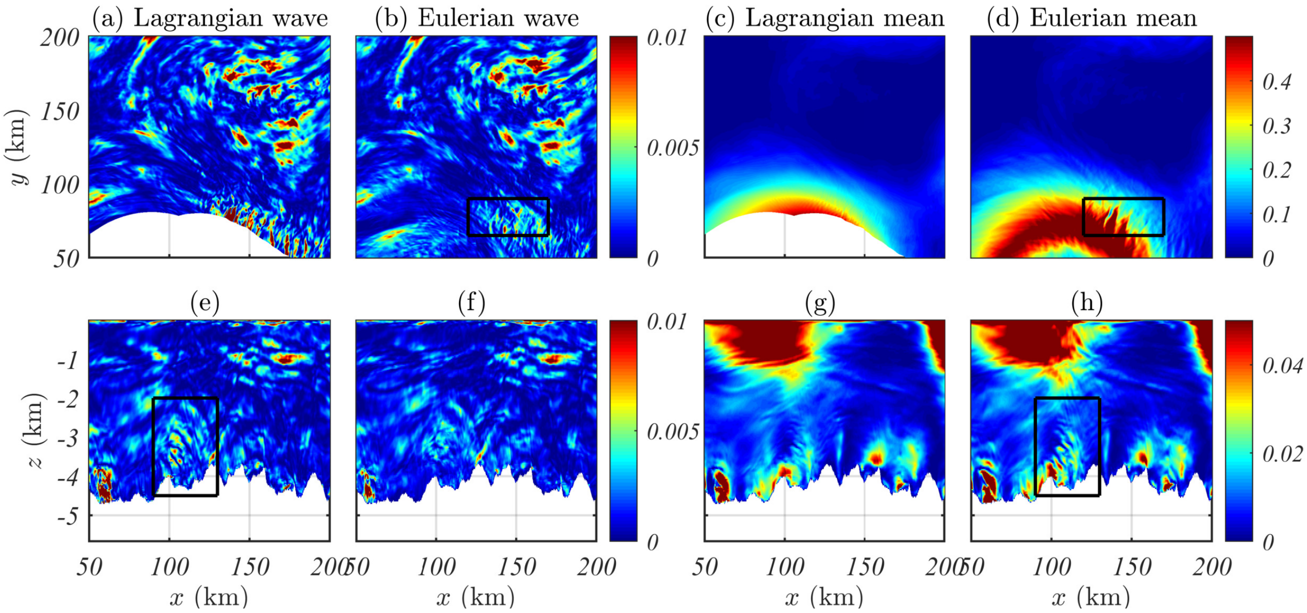

Figure 4. Snapshots of (a,e) Lagrangian and (b,f) Eulerian filtered wave kinetic energy. (c,g) Lagrangian and (d,h) Eulerian filtered mean (non-wave) kinetic energy. (a-d) 1000 m depth. (e-h) Transect at y =125 km. Black boxes indicate regions of interest discussed in the text.

3.1.2 Comparison with Eulerian filter

As in the synthetic example in the previous subsection, it is useful to compare the operation of a Lagrangian and Eulerian filter in separating waves and non-wave flow. Figure 4 shows a comparison of the kinetic energy in the wave and non-wave (mean) flows as defined by each method using a ± 2 day window, for the same time as shown in the previous figure (12 noon UTC on 16 Dec 2010), and for the region indicated by the red square in Figure 3d. This subdomain can be specified directly in the Lagrangian filter which will restrict the computation to only this region and produce NaNs elsewhere in the output files. The Eulerian filtered fields are computed in the same way over the same time window, by simply turning off the advection kernel (see example script).

The Eulerian and Lagrangian filtered fields exhibit a number of significant differences. In particular, the Eulerian mean flow (Figure 4d) is much less smooth than the Lagrangian mean (Figure 4c), which is indicative of incomplete (or incorrect) filtering of waves. For example, the black boxes in Figure 4b,d highlight a region where the Eulerian filter misidentifies (i) fast moving mean flow as waves, and (ii), waves that are trapped in a jet (and thus Doppler shifted) as mean flow. This example is taken at 1000m depth, showing that difference between the Eulerian and Lagrangian filters can be significant, even in the ocean interior. Another important difference is apparent in the vertical transects in Figure 4e-h. A lee wave generation event is highlighted in the Lagrangian-filtered wave field (the black box in Figure 4e). However, since these waves are trapped steady flows, the Eulerian filter also incorrectly identifies them as mean flow, as seen in Figure 4h. Overall, the Eulerian filter tends to underestimate wave kinetic energy by 20-50% in the MITgcm regional model (Figure 5) with the biggest differences in the upper 1800 m.

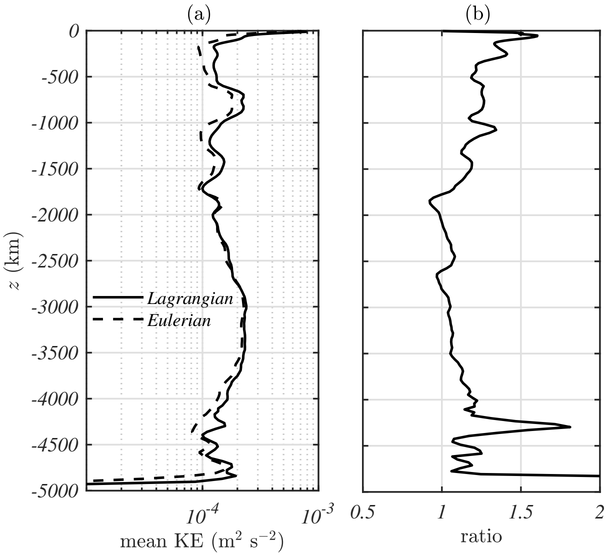

Figure 5. (a) Horizontally-averaged Lagrangian (solid) and Eulerian (dashed) filtered wave kinetic energy. (b) Ratio of horizontally-averaged Lagrangian to Eulerian filtered wave kinetic energy.

3.2 Coral Triangle regional model at 1.5km resolution (ROMS)

This Regional Ocean Modelling System (ROMS; Shchepetkin & McWilliams (2005)) simulation spans a cross-equatorial region around the Indonesian islands known as the 'Coral Triangle' at 1.5 km resolution, and is described in detail in Bachman et al. (2020). The simulation is forced by surface fluxes and tidal forcing in the interior, and via output fields from a lower resolution model run at the open boundaries. The Coral Triangle is known for extreme tidal and wave dynamics, including large amplitude surface and non-linear solitary internal waves (solitons). These signals are especially strong in the free surface height, thus providing a challenging test case for the effectiveness of Lagrangian filtering in wave-balanced flow separation of ocean surface elevations. We anticipate that our filtering methodology may be particularly useful for the high resolution modelling efforts being undertaken to prepare for the SWOT satellite mission (Morrow et al., 2019), many of which are aimed at filtering model output to obtain a 'balanced' flow field. We observe that when applied to filter surface fields (as here) the 'semi-Lagrangian' method used by the filter is fully Lagrangian, since the neglected vertical advection of parcels is identically zero.

As noted in Bachman et al. (2020), applying the Lagrangian filtering method in a near-equatorial region presents the additional challenge of identifying an appropriate cut-off frequency that separates 'wave' and 'non-wave' flow. Clearly, it is not possible to simply select the inertial frequency as the cut-off, since this vanishes at the equator. Thus, we use the method adopted in Bachman et al. (2020) of identifying the minimum in the frequency spectra separating slow non-wave motion from fast wave motion. It is anticipated that computing such spectra will be a common requirement when setting up an analysis workflow for any new regional simulation, and this functionality has therefore been built into the Lagrangian filtering package (see example script). Figure 6 shows both the Lagrangian and Eulerian frequency spectra for both the kinetic energy and free surface height, averaged over the domain for a ± 2 day time window. A minimum is observed in the kinetic energy at $\omega = 5 \times 10^{-5}$/s, which we therefore select as the cut-off frequency.

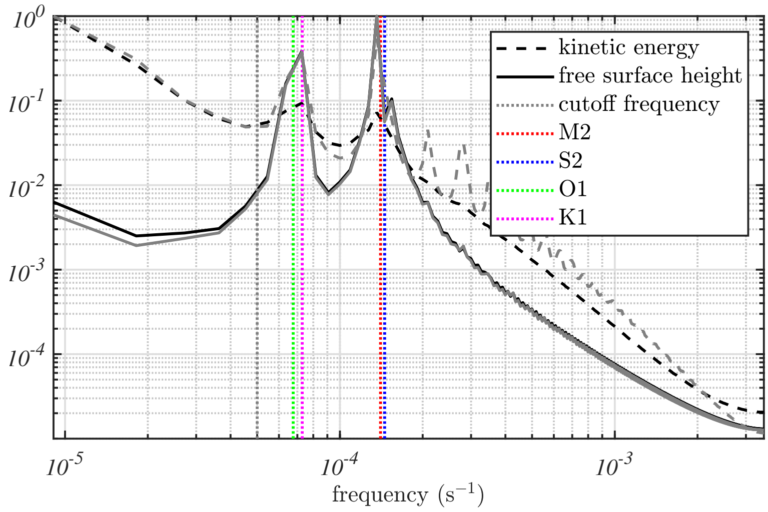

Figure 6. Particle track-averaged Lagrangian frequency (black) and domain averaged Eulerian (grey) spectra of free surface height (solid) and surface kinetic energy (dashed) from the model of Bachman et al. (2020). Frequencies of the major tidal constituents are indicated as coloured lines. A cut-off frequency of $\omega_c = 5 \times 10^{-5}$ s^{-1} (grey line) is selected to coincide with the minimum in the spectra.

Figure 7 shows how the sequential application of Lagrangian and Eulerian filters may be used to separate a total free surface height (top) into different dynamical constituents (bottom). The figure shows the gradient of free surface height (rather than free surface height itself) to highlight the different dynamical regimes on the same colour scale. Of particular interest here is the low-frequency non-wave component (bottom left) resulting from the sequential application of a low-pass Lagrangian and low-pass Eulerian filter. Recall that the Eulerian filter imposes the constraint $\omega \ll \omega_c$ assuming all amplitude is well below the cut-off frequency; that is, the timescale of the flow is long. Secondly, the Lagrangian filter imposes the constraint that $\omega - Uk \ll \omega_c$, for flow speed U and wavenumber k (again assuming all amplitude is well below the cut-off frequency). We can simplify this constraint for $\omega \ll \omega_c$ to be simply $Uk \ll \omega_c$ or $Ro = kU/\omega_c \ll 1$; that is, the flow has a small Rossby number. Thus, the doubly low pass filtered field is both slow and has a small Rossby number; we thus argue that this field best represents the balanced flow (Ford et al., 2000; Vanneste, 2013). If we were to only apply one of these constraints, Figure 7 displays the 'errors' that would occur in this definition of balance. For example, using only a Lagrangian filter would include high-frequency non-wave processes such as submesoscale eddies ($\omega > \omega_c$, $Ro \geq 1$). Similarly, using only an Eulerian filter would include low-frequency waves such as lee waves ($\omega < \omega_c$, $Ro \geq 1$). In either case, the error introduced could be significant (about 10% of the free surface height in this example, and a first order contribution to the height gradient).

3.3 Global ocean model at 0.1° resolution (MOM5)

As a final example, we apply the Lagrangian filtering to identify internal waves in the global ocean sea-ice model ACCESS-OM2 (Kiss et al., 2020), the ocean component of which is the Modular Ocean Model (MOM) version 5.1 from the Princeton Geophysical Fluid Dynamics Laboratory (`https://mom-ocean.github.io`; known as 'MOM5'). Application of the technique to a global ocean model requires a spatially variable cutoff filter such that an appropriate wave frequency limit (e.g. the local inertial frequency $f = 2\Omega \sin\theta$) may be imposed at each latitude. This spatially-variable cutoff is imple-

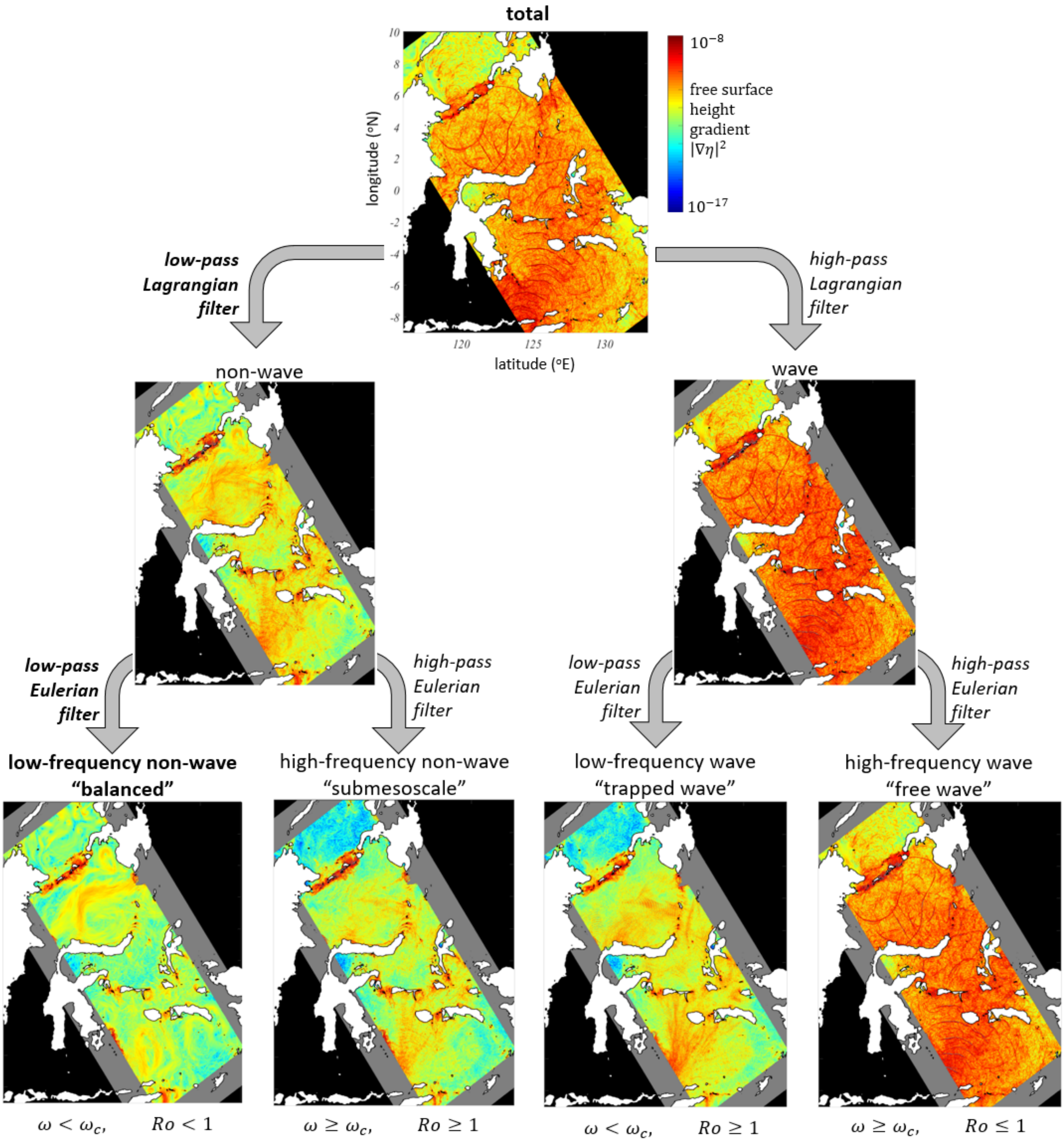

Figure 7. Methodology to identify a 'balanced' free surface height field, using output from the model of Bachman et al. (2020) as an example. Plots show the gradient-squared of free surface height $|\nabla\eta|^2$. (top) The total height gradient at 20:15 on 9 October 2016. (middle) Lagrangian filtered (left) non-wave and (right) wave. (bottom) Lagrangian and Eulerian dual-filtered fields. The approximate parameter regime for each field is shown at the bottom where $Ro = kU/\omega_c$. The low-frequency non-wave component (bottom left; bold) provides a best-estimate of the balanced flow field. All filters in this example use a cut-off frequency of $\omega_c = 5 \times 10^{-5}$ s^{-1}. Greyed areas show the part of the model domain omitted in the filtering. Black areas are ocean outside the model domain.

mented efficiently via the `sosfilt` library as discussed in Section 2.1. Here we impose the local inertial frequency as the cut-off, except equatorward of 10° latitude where we set the cut-off as the inertial frequency at 10°. The ACCESS-OM2 model uses a tripolar grid in the Arctic for which the Parcels advection kernel fails. While it is anticipated that further developments will address this issue[1], for the present example we simply mask all areas north of 60°N, and only filter the region to the south where the grid is rectilinear.

Figure 8 shows the results of the Eulerian and Lagrangian filtering methods applied to the surface velocity fields from the global model, using hourly model output and a ±2 day filtering window. Low-mode internal waves are visible in the Southern Ocean (consistent with wind-generated near-inertial waves; e.g. Simmons & Alford (2012)) in both the (a) Lagrangian and (b) Eulerian filtered zonal velocity. Indeed, by eye, these two filtered fields look identical. However, taking the difference (Figure 8c) reveals that the fields differ by ∼10% at smaller scales in strongly eddying regions of the ocean (e.g. western boundary currents, Antarctic Circumpolar Current, etc.), indicating the presence of trapped waves and/or submesoscales (as per Figure 7). This behaviour is consistent with the expected wave dynamics: large-scale, low-mode waves are unaffected by the Doppler shift and hence appear equally in both filtering methods, whereas smaller scale waves in regions of strong flow experience a large Doppler shift and can become sub-inertial in an Eulerian frame, meaning they are captured by the Lagrangian filter but not the Eulerian. Conversely, submesoscales (if they exist with any significant amplitude, which is unlikely at 0.1° resolution) would be captured by the Eulerian filter but not the Lagrangian. The kinetic energy in the Lagrangian high-pass filtered field is significantly higher than in the Eulerian high-pass filtered field in almost all regions (Figure 8d,e), indicating that it is trapped waves (and not submesoscales) that dominate the difference between the filtered fields. This difference can comprise a large fraction of the total energy in eddying regions (Figure 8f).

4 Discussion

Here we have introduced a new Python package that implements an updated version of the Lagrangian filtering methodology first proposed by Nagai et al. (2015) and Shakespeare & Hogg (2017). The method separates the high frequencies as measured in a frame moving with the flow (i.e. internal waves) from the remainder of the flow, with the parcel-tracking kernel using the existing OceanParcels framework (Delandmeter & van Sebille, 2019). The filtering package can be applied directly to netCDF format ocean model output from both high-resolution global and regional simulations. Successful use of the method requires model output that is high resolution in time (typically hourly, to resolve internal waves) and has a ∼ ±2 day window of output data around the time of interest. The package acts independently on particular vertical levels and times, making the filtering an embarrassingly parallel operation. In addition, openMP parallelisation is implemented to speed up computations of the filtered fields for individual levels/times.

We have presented a number of examples to demonstrate the usefulness of the Lagrangian filtering method and how it differs from the more common direct temporal filtering at fixed points in space (Eulerian filtering). Lagrangian filtering permits the correct identification of Doppler shifted internal waves — for example, lee waves (see Figure 4) — and by extension, removes such waves from the non-wave or 'mean' flow field. This dual role is vital in facilitating calculation of wave energy fluxes and wave-mean energy exchanges and is the primary reason for the original development of Lagrangian filtering (Nagai et al., 2015; Shakespeare & Hogg, 2017, 2018, 2019). In particular, the use

[1] The current solution is to interpolate the tripolar grid model output onto a rectilinear grid, but this adds an undesirable intermediate processing step.

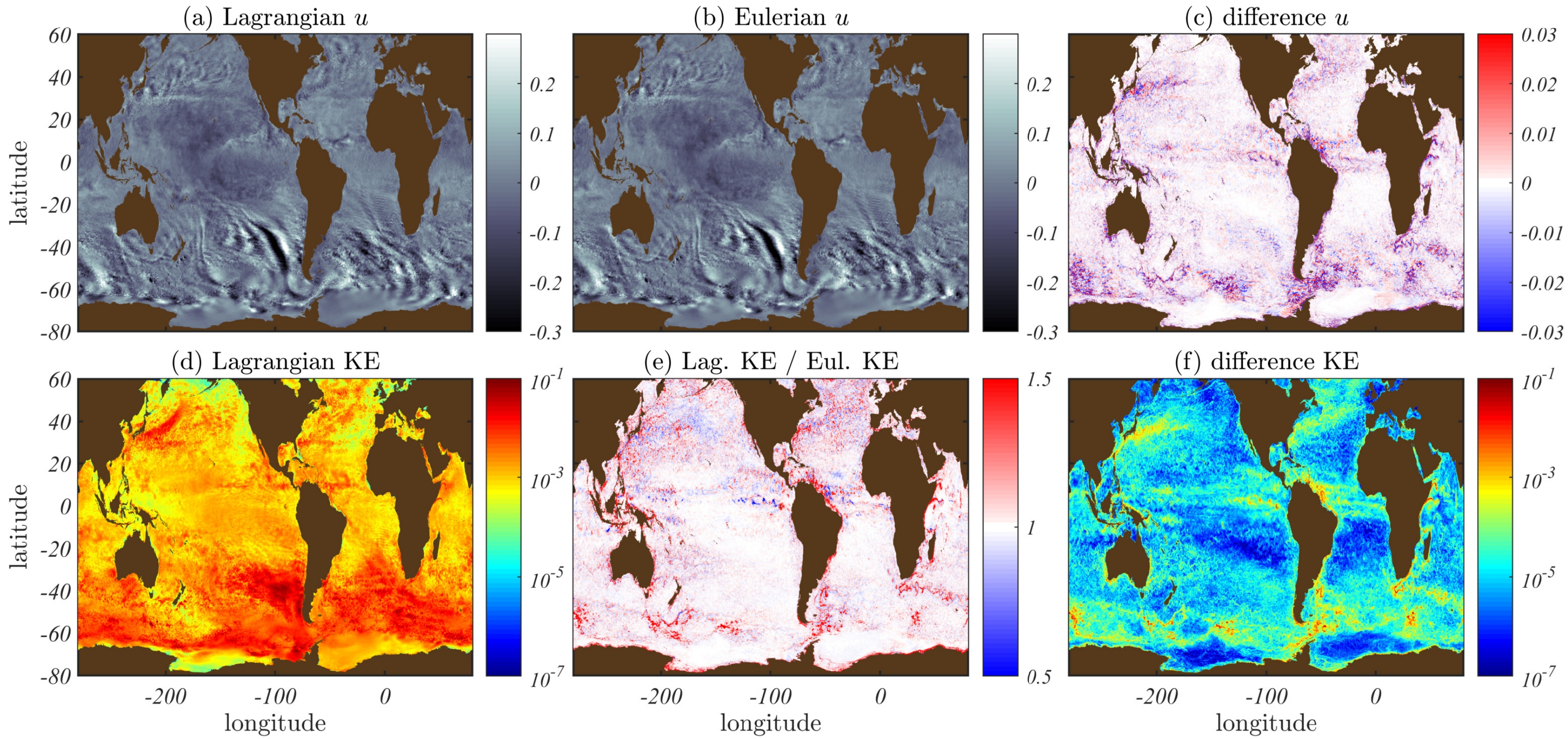

Figure 8. Comparison of Lagrangian and Eulerian filtering in the ACCESS-OM2 global ocean model (Kiss et al., 2020). Snapshot of the (a) Lagrangian high-pass filtered and (b) Eulerian high-pass filtered zonal velocity at the ocean surface. (c) The difference between (a) and (b). (d) The Lagrangian filtered wave kinetic energy, averaged over 4 days. (e) The ratio of the 4-day averaged Lagrangian filtered kinetic energy to the Eulerian filtered kinetic energy. (f) The 4-day averaged kinetic energy in the difference between Lagrangian and Eulerian filtered fields.

of Lagrangian filtering enabled Shakespeare & Hogg (2017, 2018) to formulate the first-ever closed internal wave energy budget in high-resolution regional simulations.

Here we have also shown the potential utility of Lagrangian filtering in the related problem of identifying the 'balanced' flow field. While this is a long-standing problem in the field (e.g. see the review of Vanneste, 2013), it has attracted additional attention in recent years as preparatory work is undertaken for the Surface Water and Ocean Topography (SWOT) satellite mission to launch in late 2022 (Morrow et al., 2019). This satellite will collect ocean topography data at high resolution which will include 'unbalanced' internal waves and submesoscales. However, given the repeat orbit period of 21 days, the corresponding hourly and daily timescales of these dynamics will not be resolved. Thus, dynamical models are required that can bridge this gap and assist in disentangling satellite-observed balanced and unbalanced motions. Here we have shown how a combination of Lagrangian and Eulerian filtering could play a role in this disentanglement by separating not only wave versus non-wave free surface height fields, but also balanced (low Rossby number) and unbalanced (high Rossby number) free surface height (see Figure 7).

The Lagrangian filtering methodology and package presented here is likely to be of broad use to the oceanographic community (and possibly beyond; e.g. atmospheric science), having already played a significant role in a number of studies of internal wave processes (Nagai et al., 2015; Shakespeare & Hogg, 2017, 2018, 2019; Bachman et al., 2020). Until now, the method has been largely inaccessible to the general community owing to the significant development overhead required to implement it in a sufficiently efficient manner to cater for the very large datasets emerging from high resolution models. Ongoing development and user involvement will likely lead to further performance improvements and the implementation of additional features.

Acknowledgments

The Lagrangian filtering Python package is freely available at `github.com/angus-g/lagrangian-filtering` with extensive documentation located at `lagrangian-filtering.readthedocs.io/en/latest/`. Version 0.90beta1 used for the examples in this manuscript is permanently archived at `https://doi.org/10.5281/zenodo.4722046`, but users are advised to refer to the github repository for the latest version.

The Jupyter notebook for the analytic example in Section 2.2 and the Python scripts for all examples in Section 3 are available at `https://doi.org/10.5281/zenodo.4748246`.

The individual datasets for the MITgcm, ROMS, and MOM5 examples are not provided owing to their large size which makes archiving impractical. However, model configuration files for the MOM5 model run by the Consortium for Ocean-Sea Ice Modelling in Australia (COSIMA; `cosima.org.au`) are publicly available at `github.com/COSIMA/access-om2`. Significant output from the ACCESS-OM2 models is also available at `http://dx.doi.org/10.4225/41/5a2dc8543105a`.

Analysis of model output was undertaken at the National Computational Infrastructure (NCI), Canberra, Australia.

Results shown in Section 3.2 are based upon work supported by the National Center for Atmospheric Research (NCAR), which is a major facility sponsored by the National Science Foundation under Cooperative Agreement 1852977.

We acknowledge high-performance computing support from Cheyenne (doi:10.5065/D6RX99HX) provided by NCAR's Computational and Information Systems Laboratory, sponsored by the National Science Foundation.

CJS acknowledges support from an ARC Discovery Early Career Researcher Award DE180100087 and an Australian National University Futures Scheme award.

SRK acknowledges support from ARC Discovery Project DP210102745 and a UNSW Faculty Research Grant.

References

Bachman, S. D., Shakespeare, C. J., Kleypas, J., Castruccio, F. S., & Curchitser, E. (2020). Particle-based lagrangian filtering for locating wave-generated thermal refugia for coral reefs. *Journal of Geophysical Research: Oceans*, *125*(7), e2020JC016106.

Bretherton, F. P., & Garrett, C. J. R. (1969). Wave trains in inhomogeneous moving media. *Proc. Roy. Soc. A*, *302*, 529–554.

Bühler, O. (2014). *Waves and mean flows.* Cambridge University Press.

Capet, X., McWilliams, J. C., Molemaker, M. J., & Shchepetkin, A. F. (2008a). Mesoscale to submesoscale transition in the California Current system. Part II: Frontal processes. *J. Phys. Oceanogr.*, *38*, 44–64.

Capet, X., McWilliams, J. C., Molemaker, M. J., & Shchepetkin, A. F. (2008b). Mesoscale to submesoscale transition in the California Current system. Part III: Energy balance and flux. *J. Phys. Oceanogr.*, *38*, 2256–2269.

Capet, X., McWilliams, J. C., & Shchepetkin, A. F. (2008). Mesoscale to submesoscale transition in the California Current system. Part I: Flow structure, eddy flux, and observational tests. *J. Phys. Oceanogr.*, *38*, 29–43. doi: 10.1175/2007jpo3671.1

Dask Development Team. (2016). Dask: Library for dynamic task scheduling [Computer software manual]. Retrieved from `https://dask.org`

Delandmeter, P., & van Sebille, E. (2019). The parcels v2. 0 lagrangian framework: new field interpolation schemes. *Geoscientific Model Development*, *12*(8), 3571–3584.

Egbert, G. D., & Erofeeva, S. Y. (2002). Efficient inverse modeling of barotropic ocean tides. *Journal of Atmospheric and Oceanic Technology*, *19*(2), 183–204.

Ford, R., McIntyre, M. E., & Norton, W. A. (2000). Balance and the slow quasimanifold: some explicit results. *J. Atmos. Sci.*, *57*, 1236–1254.

Keating, S. R., Smith, K. S., & Kramer, P. R. (2011). Diagnosing lateral mixing in the upper ocean with virtual tracers: Spatial and temporal resolution dependence. *Journal of physical oceanography*, *41*(8), 1512–1534.

Kehl, C., van Sebille, E., & Gibson, A. (2021). *Speeding up python-based lagrangian fluid-flow particle simulations via dynamic collection data structures.*

Kiss, A. E., Hogg, A. M., Hannah, N., Boeira Dias, F., Brassington, G. B., Chamberlain, M. A., … others (2020). ACCESS-OM2 v1. 0: A global ocean–sea ice model at three resolutions. *Geoscientific Model Development*, *13*(2), 401–442.

Marshall, J., Adcroft, A., Hill, C., Perelman, L., & Heisey, C. (1997). A finite-volume, incompressible Navier Stokes model for studies of the ocean on parallel computers. *J. Geophys. Res.*, *102*(C3), 5753–5766.

Mashayek, A., Ferrari, R., Merrifield, S., Ledwell, J. R., St Laurent, L., & Garabato, A. N. (2017). Topographic enhancement of vertical turbulent mixing in the southern ocean. *Nature communications*, *8*, 14197.

Morrow, R., Fu, L.-L., Ardhuin, F., Benkiran, M., Chapron, B., Cosme, E., … others (2019). Global observations of fine-scale ocean surface topography with the Surface Water and Ocean Topography (SWOT) mission. *Frontiers in Marine Science*, *6*, 232.

Nagai, T., Tandon, A., Kunze, E., & Mahadevan, A. (2015). Spontaneous generation of near-inertial waves by the Kuroshio Front. *J. Phys. Oceanogr.*, *45*(9), 2381–2406.

Naveira Garabato, A. C., Nurser, A. G., Scott, R. B., & Goff, J. A. (2013). The impact of small-scale topography on the dynamical balance of the ocean. *J. Phys. Oceanogr.*, *43*(3), 647–668.

Nikurashin, M., Vallis, G. K., & Adcroft, A. (2013). Routes to energy dissipation for geostrophic flows in the Southern Ocean. *Nature Geosci.*, *6*(1), 48–51.

Polzin, K. L., & Lvov, Y. V. (2011). Toward regional characterizations of the oceanic internal wavefield. *Rev. Geophys.*, *49*(4).

Shakespeare, C. J., & Hogg, A. M. (2017). Spontaneous surface generation and interior amplification of internal waves in a regional-scale ocean model. *J. Phys. Oceanogr.*. doi: 10.1175/JPO-D-16-0188.1

Shakespeare, C. J., & Hogg, A. M. (2018). The life cycle of spontaneously generated internal waves. *J. Phys. Oceanogr.*, *48*(2), 343-359. doi: 10.1175/JPO-D-17-0153.1

Shakespeare, C. J., & Hogg, A. M. (2019). On the momentum flux of internal tides. *J. Phys. Oceanogr.*. doi: JPO-D-18-0165.1

Shchepetkin, A. F., & McWilliams, J. C. (2005). The regional oceanic modeling system (ROMS): a split-explicit, free-surface, topography-following-coordinate oceanic model. *Ocean modelling*, *9*(4), 347–404.

Shcherbina, A. Y., D'Asaro, E. A., Lee, C. M., Klymak, J. M., Molemaker, M. J., & McWilliams, J. C. (2013). Statistics of vertical vorticity, divergence, and strain in a developed submesoscale turbulence field. *Geophys. Res. Lett.*, *40*. doi: 10.1002/grl.50919

Simmons, H. L., & Alford, M. H. (2012). Simulating the long-range swell of internal waves generated by ocean storms. *Oceanography*, *25*(2), 30–41.

Thomas, L. N., Tandon, A., & Mahadevan, A. (2008). Submesoscale processes and dynamics. In *Geophysical monograph series 177: Ocean modeling in an eddying regime.* American Geophysical Union.

Vanneste, J. (2013). Balance and spontaneous wave generation in geophysical flows. *Annu. Rev. Fluid Mech.*, *45*, 147–172.

Velzeboer, N. (2019). *The spatiotemporal variability of the internal wave field in the Southern Ocean* (Master's thesis, Australian National University). doi: https://doi.org/10.5281/zenodo.4722008

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., ... SciPy 1.0 Contributors (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, *17*, 261–272. doi: 10.1038/s41592-019-0686-2

Waterhouse, A. F., MacKinnon, J. A., Nash, J. D., Alford, M. H., Kunze, E., Simmons, H. L., ... others (2014). Global patterns of diapycnal mixing from measurements of the turbulent dissipation rate. *J. Phys. Oceanogr.*, *44*(7), 1854–1872.