
trackeddy Documentation

Release Beta 0.1a

Josue Martinez-Moreno

Aug 31, 2018

1	Getting Started	1
1.1	To get the code:	1
1.2	Package structure	1
1.3	Test the code	3
2	Working with trackeddy	5
2.1	Code Structure	5
2.2	Workflow	6
2.3	Diagnostics	6
2.4	Test Suite	6
3	Build Process	9
4	Introduction	11
5	Background	13
5.1	Eddy tracking	13
5.2	Kinetic Energy	15
6	Methods	17
6.1	Eddy Identification Criteria	17
6.2	Eddy Contour Replacement	21
6.3	Eddy Time Tracking	21
6.4	Future Methods	22
7	Applications	23
7.1	Kinetic Energy decomposition	23
7.2	Heatwaves	24
7.3	Oceanic Tracers	24
7.4	Regional Studies	24
8	Results	25
9	About this documentation	27
10	Related Projects	29
10.1	Underlying Python technologies	29
10.2	Packages and libraries related with this code:	29

11 References	31
12 Indices and tables	33

CHAPTER 1

Getting Started

This source code will let you identify all the eddies in the ocean, but also It can be adapted for any other normal function in a 2D surface.

1.1 To get the code:

1.- Make a new directory where you want the repository.

2.- Clone the TrackEddy repository from Github. In the command prompt, type:

```
git clone https://github.com/Josue-Martinez-Moreno/trackeddy.git
cd trackeddy
```

or set up SSH keys for github:

```
git clone git@github.com:Josue-Martinez-Moreno/trackeddy.git
```

3.- Install the package globally:

```
pip install -e .
```

or

```
pip install --force-reinstall -e.
```

or for a local installation:

```
pip install --user -e .
```

1.2 Package structure

Suggested structure for the use of this package after cloning it from github.:

```

trackeddy
├── LICENSE
├── README.md
├── data.input    --> Symbolic link to inputs.
├── data.output   --> Symbolic link to output.
├── docs          --> Documentation.
│   ├── README.md
│   ├── about.rst
│   ├── conf.py
│   ├── getting_started.rst
│   ├── images
│   ├── index.rst
│   ├── pages
│   │   ├── Diagnostics.rst
│   │   └── Tests.rst
│   ├── references.rst
│   ├── related_projects.rst
│   └── using_trackeddy.rst
├── examples      --> Notebooks implementing some of the functions in
│                   the package.
│   ├── Eddies_geostrophic_velocity_field.ipynb
│   ├── Eddies_ssh_satellite.ipynb
│   ├── Eddies_velocity_field.ipynb
│   ├── Eddies_vertical_profiles.ipynb
│   ├── MULTIPLE_STEPS_eddies_southern_ocean.ipynb
│   ├── ONE_STEP_eddies_southern_ocean.ipynb
│   ├── Potential_Vorticity.ipynb
│   ├── eddy_V1_bk.ipynb
│   ├── eddyidentification_fitness.ipynb
│   ├── eddyidentification_fitness_south_africa.ipynb
│   ├── eddyidentification_fitness_specific_eddy.ipynb
│   ├── multiprocessing_bk.ipynb
│   ├── potential_vorticity_bk.ipynb
│   ├── save_eddy_data.ipynb
│   ├── ssh_mean.ipynb
│   ├── ssh_mean_global_bk.ipynb
│   ├── test_geodesics_bk.ipynb
│   ├── track_eddy_bk.ipynb
│   ├── track_eddy_southern_ocean_bk.ipynb
│   └── vorticity_tracking_eddy.ipynb
├── output        --> Figure output or small files
├── setup.py
├── src           --> Work in progress: The core of the trackeddy algorithm
│                   will be coded in Fortran or C.
├── tests         --> Folder full of tests used to check the proper
│                   extraction and analysis of eddies.
│   ├── Centroid_eddy.ipynb
│   ├── Synthetic_fields.ipynb
│   ├── gaussian_fitting_multiple_eta_level.ipynb
│   ├── gaussian_fitting_one_eta_level.ipynb
│   ├── improving_time.ipynb
│   ├── join_files_func.ipynb
│   ├── time_datastruct.ipynb
│   ├── trackeddy_okubo.ipynb
│   ├── trackeddy_ssh.ipynb
│   └── trackeddy_ssh_bk.ipynb
└── trackeddy     --> Functions included in the package.

```

(continues on next page)

(continued from previous page)

```
— __init__.py
— datastruct.py
— geometryfunc.py
— init.py
— physics.py
— plotfunc.py
— printfunc.py
— savedata.py
— tracking.py
```

1.3 Test the code

The source code have been compiled and tested into the [Travis CI](#) environment (Check the build status on the [trackeddy GitHub](#)).

1.- Move to the test directory:

```
cd /path2trackeddy/test/
```

2.- Run any of the scripts located in that folder:

```
# Example:
python test_2d_gaussian_one_level.py
```

Note: If you want to display the diagnostics for each test, just replace: “diagnostics=False” by “diagnostics=True” at the beginning of the test file.

Warning: The testing code it’s in a early version, so please submit all the Issues to [trackeddy GitHub](#).

The following sections show how to break the code into pieces:

2.1 Code Structure

The code was divided in 4 sections to make simpler its use.

2.1.1 Python Source Code

The python source code is Located in the folder `trackeddy`. This folder contains all the utilities used to analyze and decompose an eddy field or a 2D gaussian field.

- `datastruct.py`
- `geometryfunc.py`
- `physics.py`
- `printfunc.py`
- `savedata.py`
- `tracking.py`

2.1.2 Tests

2.1.3 Examples

2.1.4 Fortran Source Code

Warning: The Fortran Source Code is under development.

2.2 Workflow

Trackeddy	
A	B
False	False
True	False
False	True
True	True

2.3 Diagnostics

2.3.1 Online diagnostics

The main function in TrackEddy has multiple options to output diagnostics (Default: False):

- “all” or True : Outputs all diagnostics.
- “ellipse” : Outputs just the ellipse fitting diagnostics.
- “contours” : Outputs the masking of the dataset diagnostic.
- “2dfit” : Outputs the 2D gaussian fitting diagnostics.
- “gauss” : Outputs the minor and major axis gaussian fitting diagnostics.
- “time” : Outputs the time tracking diagnostics.
- False : Doesn’t output any diagnostic.

2.3.2 Offline diagnostics

2.4 Test Suite

Some tests are provided to check the stability of the code and it’s reproducibility.

Inside the folder `tests` are two available tests:

- `*.py` which correspond to the tests used in Travis CI and you can run them using:

```
pytest -m testme
```

- `*.ipynb` which correspond to the same tests used in Travis CI, but displaying more information and diagnostics. Inside the trackeddy directory run:

`jupyter notebook`

then, move into the `tests` folder and select the test you want to check.

2.4.1 Tests description

Warning: This section continues under development!

Now, a detailed description of each test is provided.

`test_2d_gaussian_one_level`

`test_2d_gaussian_multiple_level`

`test_ellipsoid_fitting`

`test_timestep_tracking`

`test_deta_tracking`

CHAPTER 3

Build Process

The ocean regulates Earth's climate through different spatial-temporal scales varying from a few millimetres or seconds to thousands of kilometres or years. Each scale contributes to the control of the carbon cycle, heat balance, global circulation, oceanic biology and energy balance. Ocean processes with spatial scales of 10 to 100 km and temporal scales of days to years are known as mesoscale processes, and include eddies, jets and waves. Mesoscale processes play important roles in energy transfer, biologic dynamics and heat, momentum and tracers transport (Lèvy et al., 2012; Thomas et al., 2008). However, even though mesoscale circulation has been studied since the 1970s (Zhang et al., 2014; Wyrski et al., 1976; Gill et al., 1974), the global scale interaction and impact of each mesoscale process remains unknown. Mesoscale processes constitute a significant component of the oceanic kinetic and available potential energy. In the case of kinetic energy, a Reynolds decomposition separates the flow into a perturbation, or transient, state and a temporal mean state. These two components are commonly defined as the mean kinetic energy (MKE) and eddy kinetic energy (EKE)

(Kang and Curchitser, 2017; Wyrski et al., 1976). Analogous to the case of the KE, the available potential energy (APE) can be decomposed into the time mean available potential energy (MAPE) and the eddy available potential energy (EAPE) (Oort et al., 1989). These decompositions provide the mean background and the time-variation energy budget, but do not differentiate between mesoscale and non-mesoscale processes, which are present in both mean and transient fields.

In many parts of the ocean, mesoscale processes contain more EKE and EAPE than the temporal mean state (Barthel et al., 2017; Chen et al., 2014). Despite the name EKE and EAPE, both decompositions contain all transient processes, not just eddies. This lack of a process-based definition of EKE and EAPE makes it difficult to understand each process in the oceanic transient adjustment to climate change. Thus, the contribution of eddies and jets in the oceanic energy budget remains unknown.

Additionally, Hogg et al. (2015) found a decadal increase in EKE in the Southern Ocean. The present study will explore the consequences of the increasing trend in EKE in more detail. As the mesoscale variability forms a crucial component of the ocean circulation dynamics, including the large-scale and time-mean circulation, this study will be focused on each mesoscale process to understand it on a global scale.

The first complication over the understanding of eddy's energy is the lack of strict definition of an eddy. The second problem is the complexity of all the processes who interact with these transient features. Finally, the definition of EKE does not differentiate between eddies, jets and background, where each of this process corresponds to a different reservoir of energy. In order to understand the ubiquity of eddies, their complexity and the mixture of energy reservoirs, the present section explore the background of eddy tracking algorithms and the mathematical definition of KE.

5.1 Eddy tracking

Mesoscale turbulence now is possible using Earth-orbiting satellite data and numerical simulations. Due to the necessity to understand the impact of the eddies in the ocean and Earth's climate, autonomous methods were needed to extract mesoscale features. Since the 70's the study of ocean eddies mainly used satellite data of ocean colour and sea surface temperature. However, recently the sea level anomaly provided a more related field with the ocean eddy field than previous datasets. Eddies are classified based on their rotational direction as cyclonic if they rotate counter-clockwise (in the Northern Hemisphere) or anti-cyclonic otherwise [Figure 1]. Cyclonic eddies produce a negative perturbation in SLA and elevations in subsurface density surfaces. Anti-cyclonic eddies, cause a positive perturbation in SLA and depressions in the subsurface density surfaces. These characteristics have been used in the identification of ocean eddies in SLA, where closed-contours represent the definition of an eddy.

One of the first automated eddy detection algorithms used physical criteria and relied on a measure of rotation and deformation known as Okubo-Weiss (W) parameter [2]:

$$W = \underbrace{\left(\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y}\right)^2}_{\text{Normal component of strain}} + \underbrace{\left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right)^2}_{\text{Shear component of strain}} - \underbrace{\left(\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}\right)^2}_{\text{Relative Vorticity}}$$

The eddies were defined as features where the W -parameter was below an expert-specified negative threshold. This parameter was typically applied in a region, where a specific threshold was defined. Chelton *et. al.*, 2007 presented the first global observational eddy tracking study based on the W -parameter, using a global uniform threshold [3]. Some years later, Petersen *et. al.*, 2012 analysed a 3D regional model using the W -parameter in addition of a constraint on the shape of the eddies [4].

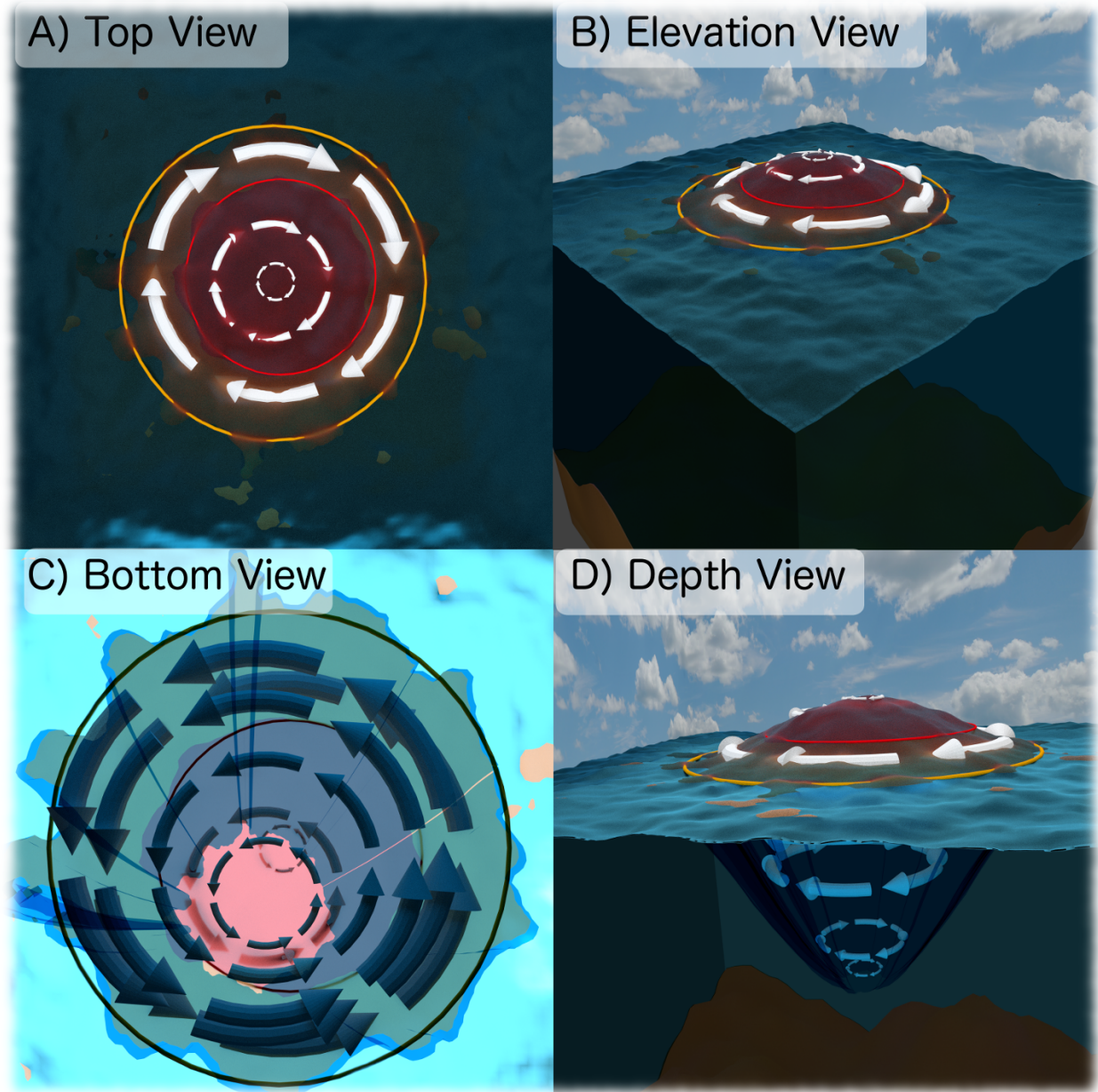


Fig. 1: Figure 1. Representation of an anti-cyclonic eddy (Northern hemisphere) and its velocities over the SLA field.

However, the W -parameter approach has been criticized for its dependence on thresholds and sensitivity to noise. As a result, later research efforts developed new methods as wavelet analysis, winding angle, reversal of the flow field, perturbation of the sea surface temperature (SST), the outermost closed SLA anomaly contours, or a combination of physical and geometric methods. All these methods try to alleviate the threshold dependent, however, they depend on expert-defined parameters to determine what constitutes an eddy.

While the algorithm may be different, SLA methods follow a similar workflow, which could include pre- and post-processing steps (temporal and spatial filters). Everything starts with a single snapshot of SLA (satellite or model), where these methods isolate the potential eddies. Notice that a three-dimensional study using surface information is not appropriated cause that data is noisy and contains multiple processes other than eddies. Is important to point out, that the criteria that define an eddy changes from study to study but generally involve and strict expert-defined criteria such as minimal size or shape. This process is repeated each timestep of the dataset, then the eddy structure and trajectories are constructed by associating features in the time $n - 1$ to the nearest and most similar feature in n . Finally, some studies filter their results to only include eddies that persist a specific window of time (often several weeks or months).

This track eddy algorithm has three main steps. First, identify eddy like features using the closed-contours in SLA or other physic parameters that could have multiple extremums. Second, adjust an two dimension gaussian to remove the background and any other noise contained by the signal. Third, we track each feature in time by associating a feature in on time-step to the nearest feature in the subsequent one.

5.2 Kinetic Energy

Energetics analysis has been used to understand the oceanic variability [REFERENCES]. The total kinetic energy (KE) has been decomposed into the mean flow and a time-varying flow. This approximation follows the perturbation theory, where the velocity field is split into a mean and the perturbation, or difference between the original field and the mean. This decomposition contains crossed terms, which contain information of both flows. However, the residual term vanishes when it is averaged over the same time-mean period used in the velocity decomposition. This KE decomposition is referred as an orthogonal decomposition, resulting on two KE components commonly known as the Mean Kinetic Energy (MKE) and the Eddy Kinetic Energy (EKE).

Structure

1. Read in ssh

- remove anomaly
- detrend
- Filter

One point for each flow chart

This algorithm was developed with the main idea of decomposing oceanic Kinetic Energy into processes. Even when its implementation can be useful in multiple applications, the current version of the algorithm is primarily focused on methods to extract energy from SSH fields. However, according to few tests, it could be implemented to track eddies in pressure levels, isopycnals and even vorticity.

The criteria ranges were defined using a repetitive Southern Ocean simulation, particularly this section of the documentation was done analysing a section of the Agulhas current. For a detailed - code description, refer to [how_it_works.ipynb](#)

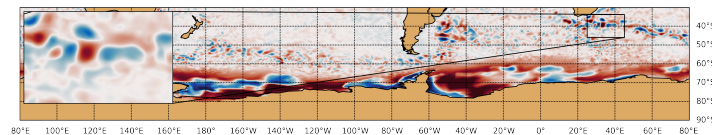


Fig. 1: Figure 1. Section of the Agulhas current used to explain how the algorithm works.

6.1 Eddy Identification Criteria

TrackEddy identifies eddies when their outer most contours can be fitted by an ellipse (A. Fernandes and S. Nascimento, (2006)), the area of the eddy contour is smaller than $2\pi L_r$ (Klocker, A., & Abernathey, R. (2014)), the eccentricity of the fitted ellipse is smaller than $\frac{b}{2a}$ and the field profile along the minor and major axis of the fitted ellipse

must adjust to a Gaussian. Optionally, an additional criterion is implemented when the 2D Gaussian fitting is allowed, this criterion identifies eddies only if the fitted 2D Gaussian correlates over 90%. For additional information, please refer to the following subsections.

These criteria allows eddies to contain multiple local extreme values, which is particularly beneficial when they merge, interact or form from jets or other processes.

6.1.1 Ellipse Fitting

According with Fernandes (2006) an eddy can be represent as an ellipse. Therefore, in this algorithm the optimal ellipse is fitted to any close contour and in order to determine if it corresponds to an eddy, the correlation between the fitted ellipse and the close contour should be within the interval (e) :

$$0.85 < e \leq 1$$

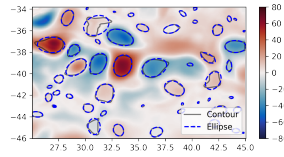


Fig. 2: Figure 2. Identified contours only within the ellipse fitting interval (Ignoring any additional criteria).

Values around 1 represent an exact fitness and the minimum value accepted should be higher than 0.85.

6.1.2 Area Check

The eddy area (A_{eddy}) was defined as a box with sides of two semi-minor axis and two semi-major axis of the fitted ellipse. Klocker, A. (2014) proposed that the eddy length scale (L_{eddy}) is always smaller than two π Rossby Radius.

Therefore, the area of any identified eddy should be less or equal to a square with sides two times the Rossby Radius.

$$A_{eddy} \leq \left(2\pi \frac{(g'D)^{\frac{1}{2}}}{f} \right)^2 = (2\pi Lr)^2$$

Note: The Rossby Radius was obtained from the Global Atlas of the First-Baroclinic Rossby Radius of Deformation ([Click here](#)). Where values were inexistent, they were replaced by the closest known value (Fig. 3).

Attention: The decision to calculate areas using boxes instead of polygons reduced the computational time significantly.

6.1.3 Eccentricity

In order to remove elongated features, potentially jets and because eddies are stable coherent structures a condition of eccentricity was imposed. The ellipse eccentricity (ϵ) range goes from 0 (perfect circle) to 1 (line). Thus the selected

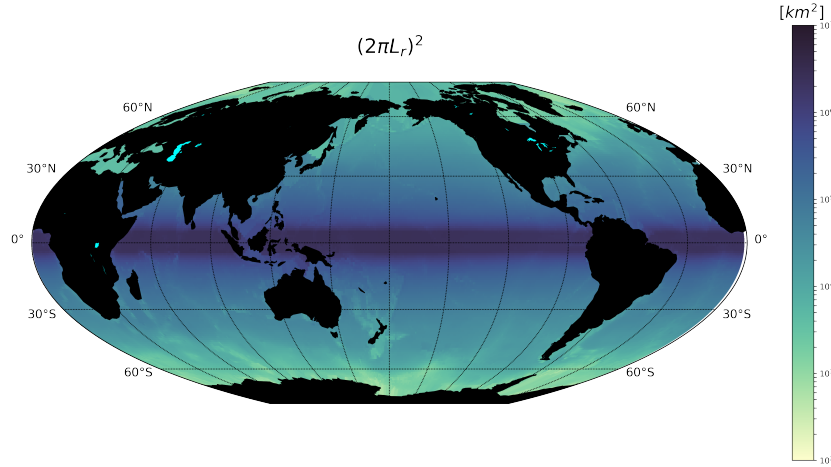


Fig. 3: Figure 3. Global eddy area based on the First-Baroclinic Rossby Radius of Deformation.

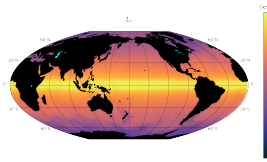


Fig. 4: Figure 4. Global First-Baroclinic Rossby Radius of Deformation.

value to constrain it represents a semi-minor axis two times smaller than the semi-major axis ($a = 2b$) or:

$$\epsilon = \left(1 - \frac{b^2}{a^2}\right)^{\frac{1}{2}} \text{ if } a \leq 2b \rightarrow \text{Eddy is identified}$$

or

$$0 \leq \epsilon \leq 0.85$$

6.1.4 Gaussian Axis Check

According with 500 detected eddies (Fig. 5), their mean profile can be fitted by a Gaussians and/or paraboloids, however the best fit was found on the Gaussian fit. Additionally, according with diffusion and advection we will expect a decay (Gaussian) instead of an abrupt change (Parabolic). Therefore, to identify an eddy, the data profile of the minor and major axis should have a high coefficient of determination (ψ) with its optimal fitted gaussian. The interval was define as:

$$0.80 < \psi \leq 1$$

Values around 1 represent a exact fitness and the minimum value accepted should be higher than 0.8.

Warning: This criterion potentially will be removed in further versions of the algorithm due to it's minimal impact over the detected eddies.

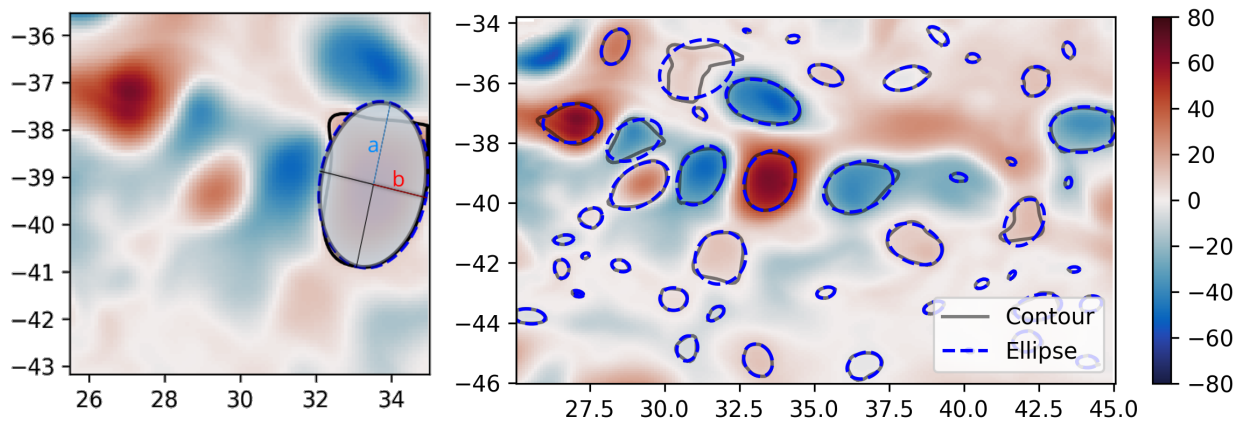


Fig. 5: Figure 5. Eddy characterisation based on the eccentricity of the fitted ellipse (blue line).

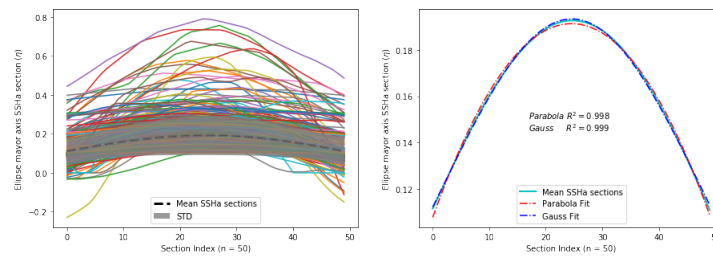


Fig. 6: Figure 6. Gaussian and parabolic fit over the average of 500 eddies.

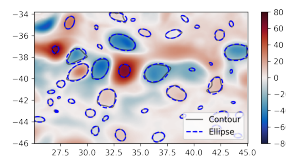


Fig. 7: Figure 7. Identified contours using all criteria.

Note: After all the previous described criteria the Figure 6 show all identified eddies and their correspondent contour.

6.1.5 2D Gaussian (Optional)

The fitness of a 2D Gaussian is constrain by the coefficient of determination between the integral of the original field and the fitted field. Additionally, the 2D gaussian fitted must satisfy the same criteria as the eddy identification, otherwise the eddy is discarded.

- Fitted contour area should be within:

$$\frac{A_{contour}}{1.05} \leq A_{2D\ Gaussian} \leq 1.05 A_{contour}$$

- 2D gaussian eccentricity should be on range:

$$0 \leq e < 0.95$$

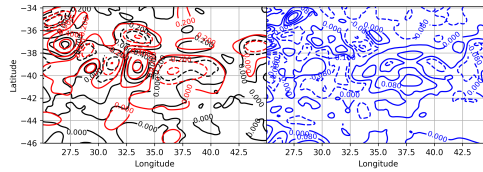


Fig. 8: Figure 8. Gaussian fitting. Left panel shows the original field (black line) underlying the reconstructed field (red line). Right panel shows the difference between fields.

Important: If this option is allowed, the condition to identify eddies depends on the fitness of the fitted Gaussian. Which should be within the interval $0.90 < e_G \leq 1$. Otherwise, the eddy is discarded.

6.2 Eddy Contour Replacement

The algorithm correlates vertical contours whenever the level $l(n-1)$ share their local maxima value and the local coordinates of the maxima with the current analysed level $l(n)$. This process is only allowed when the contour with level $l(n)$ passes all the Eddy Identification Criteria

According with the criteria described before, the current algorithm is capable of extracting the eddy signal from Aviso's dataset.

6.3 Eddy Time Tracking

All the transient features are identified in each SLA snapshot, following the eddy identification algorithm, a time tracking is applied: For each eddy feature identified at time t , the features at time $t+1$ are searched to find an eddy feature inside the close contour or the closest feature within the distance an eddy can displace between two successive time frames. This constrain uses the phase speed of a baroclinic Rossby wave, calculated from the Rossby radius of deformation as presented in Celton *et. al.* [4] and a 180 degree window search using the last preferential direction where the eddy was propagating.

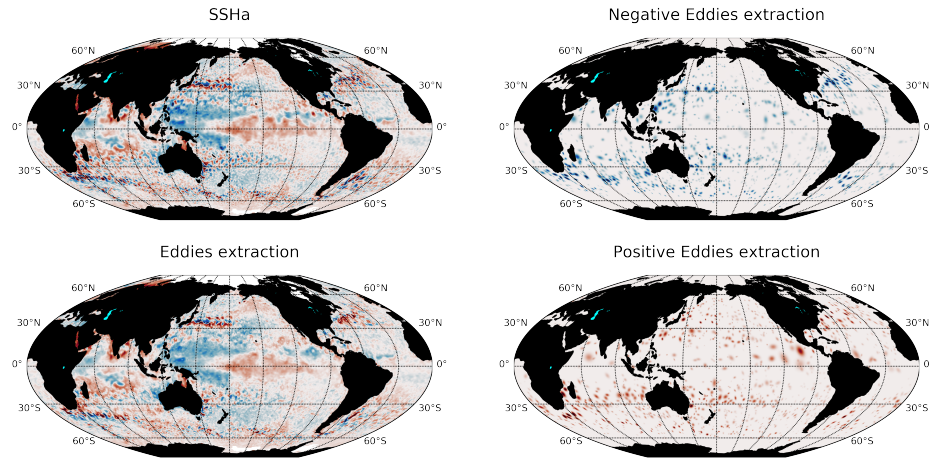


Fig. 9: Figure 5. Gaussian fitting in two dimensions to recreate the eddy field. (A) Anti-cyclonic eddy. (B) Cyclonic eddy. (C) Synthetic eddy field. (D) Difference between the original field and the synthetic field [cm].

Once a feature at time t is associated with another feature at time $t + 1$ their amplitude and area is compared. However, this comparison doesn't avoid the association of eddies cause the nature and purpose of this tracking algorithm.

When global model data is used, the eddies continuity on time is not significative affected, therefore the eddies do not disappear as often as in satellite data (AVISO products). Nonetheless, this tracking algorithm contain an automatic procedure, which allows feature to be reassociated using an user-defined number of time-steps as threshold before terminating the track (This is also related with the traveled distance by the eddy).

6.4 Future Methods

6.4.1 Identification

Note:

- The phase angle will be implemented in the Beta 0.2 release [5].
 - The eddy's 3D structure will be implemented in the V.1 release.
-

6.4.2 Time

Note: The 180 degree window and closest feature within the baroclinic Rossby wave speed will be implemented for the next release.

This code was developed with the goal of analyse and extract the energy contained by Eddies and jets in the ocean. However it can be used for a variety of processes. To know more about these processes, please se below:

7.1 Kinetic Energy decomposition

In many parts of the ocean, transient processes contain more kinetic energy (commonly known as eddy kinetic energy, or EKE) than the men kinetic energy (MKE). Variations in the oceanic energy balance can be diagnosed through EKE , which allows the analysis of positive or negative feedbacks on climate change. However, one difficulty in determining the role of eddies in the ocean transient adjustment to climate change is the lack of a process-based definition of EKE .

The aim of this algorithm is to decompose and analyse EKE according to different ocean processes. Specifically, the separation of kinetic energy will be recustructed using a 2D gaussian fitting for each closedd eddy detected (EKE_{eddy}) from the eddy kinetic energy due to meandering jets (EKE_{jets}) and the background eddy kinetic energy ($EKE_{background}$):

$$EKE = EKE_{eddy} + \underbrace{EKE_{jets} + EKE_{background}}_{EKE_{residual}}$$

However, this decomposition represents several challenges like:

- Second order terms which maybe important in the energy balance.

$$KE = MKE + EKE$$

Expanding this equation we obtain:

$$KE = MKE + EKE_{eddy} + \underbrace{EKE_{jets} + EKE_{background}}_{EKE_{residual}} + EKE'_{O^1}$$

Replacing the kinetic energy definition:

$$\frac{1}{2}\rho_0(u^2 + v^2) = \frac{1}{2}\rho_0(\bar{u}^2 + \bar{v}^2) + \frac{1}{2}\rho_0(u_{eddy}^2 + v_{eddy}^2) +$$

$$\begin{aligned}
 & \frac{1}{2}\rho_0(u_{jets}^2 + v_{jets}^2) + \frac{1}{2}\rho_0(u_{background}^2 + v_{background}^2) + \\
 & \rho_0(\bar{u}u_{eddy} + \bar{v}v_{eddy}) + \rho_0(\bar{u}u_{jets} + \bar{v}v_{jets}) + \\
 & \rho_0(\bar{u}u_{background} + \bar{v}v_{background}) + \rho_0(u_{eddy}u_{jets} + v_{eddy}v_{jets}) + \\
 & \rho_0(u_{eddy}u_{background} + v_{eddy}v_{background}) + \\
 & \rho_0(u_{jets}u_{background} + v_{jets}v_{background})
 \end{aligned}$$

where $u = \bar{u} + u_{eddy} + u_{jet} + u_{background}$. Assuming $\iff EK\bar{E}'_{O1} \rightarrow 0 \implies$ we can ignore those terms. However, this implications is really hard to prove unless we define an exact way to extract the velocity field for each eddy.

Because of this, the first approach to this problem will be the decomposition of the components.

7.2 Heatwaves

7.3 Oceanic Tracers

7.4 Regional Studies

CHAPTER 8

Results

Warning: Work in progress!

About this documentation

This readthedocs site hosts the documentation of trackeddy.

Here is where to find documentation:

Download, install and run: Installation documentation is in the form of a README.md located in the trackeddy GitHub repository. Go to <https://github.com/Josue-Martinez-Moreno/trackeddy> or have a look at <http://trackeddy.readthedocs.io/en/latest/index.html>.

User guide: [This site](#) provides a descriptive overview of the algorithm as well as the documentation of the source code.

This user guide is written in reStructuredText (.rst files) that reside in `docs/` of the [trackeddy source code](#). The rst files are processed by sphinx and hosted on [readthedocs](#).

CHAPTER 10

Related Projects

Trackeddy is an algorithm for eddy tracking and the extraction of eddy properties using high resolution input like models (MOM5, MOM6, ACCESS-OM2) and satellite data (AVISO+). This algorithm is running on NCI (Australia) resources such as raijin.

Active development is currently underway as part of the PhD Josue Martinez-Moreno.

10.1 Underlying Python technologies

Jupyter Python

10.2 Packages and libraries related with this code:

- [eddyTracking](#)
- [Cosima-Cookbook](#)

References

- 1.- Faghmous, J. H., Frenger, I., Yao, Y., Warmka, R., Lindell, A., & Kumar, V. (2015). A daily global mesoscale ocean eddy dataset from satellite altimetry. *Scientific data*, 2.
- 2.- Chang, Y. L., & Oey, L. Y. (2014). Analysis of STCC eddies using the Okubo–Weiss parameter on model and satellite data. *Ocean Dynamics*, 64(2), 259-271.
- 3.- Chelton, D. B., Schlax, M. G., Samelson, R. M., & de Szoeke, R. A. (2007). Global observations of large oceanic eddies. *Geophysical Research Letters*, 34(15).
- 4.- Faghmous, J., Frenger, I., Yao, Y., Warmka, R., Lindell, A., & Kumar, V. (2015). A daily global mesoscale ocean eddy dataset from satellite altimetry. *Scientific Data*, 2, sdata201528. doi:10.1038/sdata.2015.28
- 5.- Ashkezari, M., Hill, C., Follett, C., Forget, G., & Follows, M. (2016). Oceanic eddy detection and lifetime forecast using machine learning methods. *Geophysical Research Letters*, 43(23). doi:10.1002/2016GL071269
- 6.- Klocker, A., & Abernathey, R. (2014). Global Patterns of Mesoscale Eddy Properties and Diffusivities. *Journal of Physical Oceanography*, 44(3), 1030–1046. doi:10.1175/JPO-D-13-0159.1
- 7.- A. Fernandes and S. Nascimento, “Automatic water eddy detection in SST maps using random ellipse fitting and vectorial fields for image segmentation,” in *Discovery Science*, vol. 4265. Berlin, Germany: Springer-Verlag, 2006, pp. 77–88.

CHAPTER 12

Indices and tables

- `genindex`
- `search`