Answer to reviewer 1

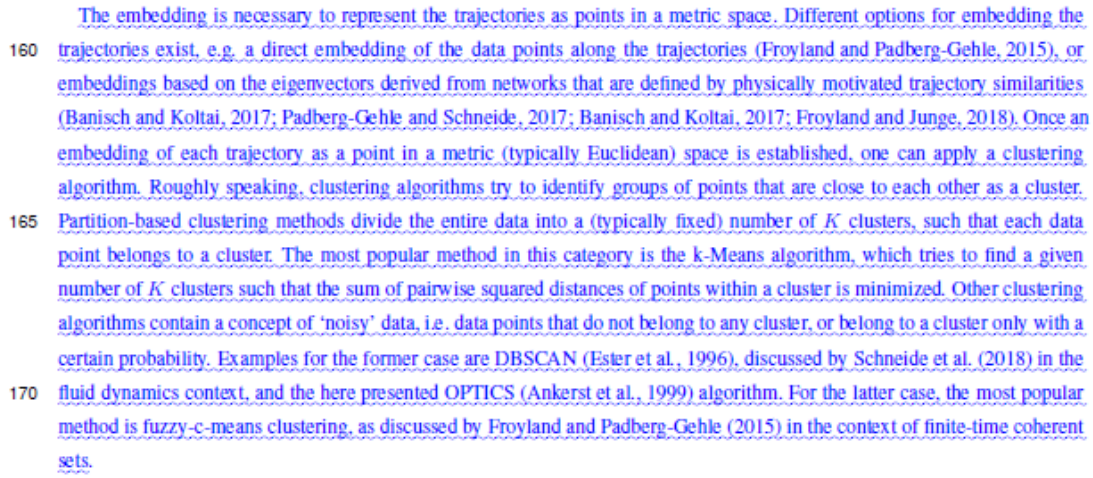
General answer:
We thank the reviewer for the detailed comments on the paper. They have helped us to significantly improve on the readability and clarity in the revised version. We have implemented changes for every comment raised by the reviewer.

Please note:
The images in this file are excerpts of the revised version in latexdiff. Please apologize the formatting problems of latexdiff that cuts off references at line breaks. This is not the case in the revised version.

Comment 1
I find there to be room for improvement on a few presentational issues. (i). There seems to be an assumption of familiarity with other clustering methods. The paper would be more accessible, and therefore useful, if the authors took just slightly more time in defining new terms and in providing the intuitive content of mathematical concepts.

Answer to comment 1
Thank you for your comment. We have made the following changes in the revised version:

1. Additional paragraph in the methods section that briefly describes why embedding / clustering is necessary, and also explains in one sentence what k-Means does


The embedding is necessary to represent the trajectories as points in a metric space. Different options for embedding the trajectories exist, e.g. a direct embedding of the data points along the trajectories (Froyland and Padberg-Gehle, 2015), or embeddings based on the eigenvectors derived from networks that are defined by physically motivated trajectory similarities (Banisch and Koltai, 2017; Padberg-Gehle and Schneide, 2017; Banisch and Koltai, 2017; Froyland and Junge, 2018). Once an embedding of each trajectory as a point in a metric (typically Euclidean) space is established, one can apply a clustering algorithm. Roughly speaking, clustering algorithms try to identify groups of points that are close to each other as a cluster. Partition-based clustering methods divide the entire data into a (typically fixed) number of K clusters, such that each data point belongs to a cluster. The most popular method in this category is the k-Means algorithm, which tries to find a given number of K clusters such that the sum of pairwise squared distances of points within a cluster is minimized. Other clustering algorithms contain a concept of 'noisy' data, i.e. data points that do not belong to any cluster, or belong to a cluster only with a certain probability. Examples for the former case are DBSCAN (Ester et al., 1996), discussed by Schneide et al. (2018) in the fluid dynamics context, and the here presented OPTICS (Ankerst et al., 1999) algorithm. For the latter case, the most popular method is fuzzy-c-means clustering, as discussed by Froyland and Padberg-Gehle (2015) in the context of finite-time coherent sets.

2. Additional explanation in the methods section that describes why the embedding we choose is expected to create a detectable signal for OPTICS.

3.2 Trajectory embedding

3.2.1 Direct embedding

The direct embedding of each trajectory in $\mathbb{R}^{DT}$ is the most ~~straight forward~~ straightforward embedding as it requires no further pre-processing of the trajectory data. For simplicity, assume we are given a set of N trajectories in a 3-dimensional space, i.e. $(x_i(t), y_i(t), z_i(t))$ where $i = 1, \ldots, N$ and $t = t_1, \ldots, t_T$. We then simply define the embedding of trajectory i in the abstract $3T$-dimensional space as

$$u_i = (x_i(t_0), x_i(t_1), \ldots, x_i(t_T), y_i(t_0), y_i(t_1), \ldots, y_i(t_T), z_i(t_0), z_i(t_1), \ldots, z_i(t_T)) \in \mathbb{R}^{3T}, \quad (3)$$

and impose an Euclidean metric in $\mathbb{R}^{3T}$ to measure distances between different embedded trajectories. The resulting embedded data matrix $\bar{X}$ is then simply given by the vertical concatenation of the different embedding vectors. This kind of embedding was also explored by Froyland and Padberg-Gehle (2015), together with a fuzzy-c-means clustering. Intuitively, if two trajectories i and j belong to the same finite-time coherent set, the corresponding particles follow very similar pathways, i.e. the Euclidean distance of the embedding vectors $d_{ij} = ||u_i - u_j||$ is expected to be small. On the other hand, a particle i that belongs to a coherent set is expected to have a larger distance to a particle j that is not part of the set. In other words, groups of particles that form a finite-time coherent set are *dense* in the embedding space. This motivates to use a density-based clustering algorithm to detect finite-time coherent sets.

To take into account the πr_0-periodicity in x-direction of the Bickley jet flow, we first put the individual 2-dimensional data points on the surface of a cylinder with radius $r_0/2$ in $\mathbb{R}^3$, and interpret the resulting ~~$(D = 3)$~~ trajectories in a 3-dimensional Euclidean space. The resulting data matrix is $\bar{X} \in \mathbb{R}^{N \times 3T}$, with $N = 12,000$ and $T = 41$. For the Agulhas particles, we put the single data points on the earth surface in a 3-dimensional Euclidean embedding space by the standard coordinate transformation of spherical to Euclidean coordinates. The resulting data matrix is thus $\bar{X} \in \mathbb{R}^{N \times 3T}$ with $N = 23,821$ and $T = 21$.

Comment 2

(ii) I find it a little strange that some figures are presented in the appendix, but discussed only in the main text. Some of these make good illustrations of the performance of the method with respect to others, e.g. D1&D2. I feel this tends to negatively impact the narrative. If figures are discussed in the main text, I would present them there also.

Answer to comment 2

Thank you for this comment. We agree with the reviewer and have now included the clustering results of the classical MDS method in the main text. In the revised version, we provide the results of OPTICS together with its comparison to k-Means for both of the model flows. We have decided to leave the discussion of the embedded network of Padberg-Gehle and Schneide (2017) together with the previous figures D1-D3 in the appendix. This is because the major focus of the paper is the OPTICS clustering on the direct embedding of the trajectories, as this removes the need of several parameters compared to Padberg-Gehle and Schneide (2017), such as the cut-off parameter *d*, and the embedding dimensions. A reader that is interested in the application of OPTICS to the spectral embedding of Padberg-Gehle and Schneide (2017) gets a full account on that topic in the appendix. We do not discuss these results in the main text, but only mention them quickly. The actual discussion is contained in appendix C of the revised version.

Comment 3

(iii) The paper has a highly technical focus throughout. More framing of the import of this problem at the start and end would have been appreciated.

Answer to comment 3

Thank you for this suggestion. We have now added more content on the problem itself, i.e. the detection of many small coherent structures in a large, noisy ocean domain.

1. Introduction

should not belong to any cluster. Graph partitioning has been shown to work in situations where the finite-time coherent sets are not too small compared to the fluid domain (Froyland and Padberg-Gehle, 2015; Hadjighasem et al., 2016; Padberg-Gehle and Schneide, 2 . For applications to Lagrangian trajectory datasets on basin-scale ocean domains, where multiple small-scale coherent sets (eddies) coexist with noisy trajectories in the background, graph partitioning is however likely to fail. Similar observations were made ~~for the spectral clustering approaches of particle based networks and~~ by Froyland et al. (2019) for the partition-based clustering approaches based on transfer and dynamic Laplace operators ~~by Froyland et al. (2019)~~(Froyland and Junge, 2018) . Although some attempts have been made to accommodate such concepts in hard partitioning, e.g. by incorporating one additional cluster corresponding to noise (Hadjighasem et al., 2016), this approach is likely to fail for large ocean domains, as

2

discussed by Froyland et al. (2019) and shown in section 4 of this paper. Froyland et al. (2019) have developed ~~an algorithm~~ a special form of trajectory embedding based on sparse eigenbasis decomposition given the eigenvectors of transfer operators and dynamic Laplacians. By superposing different sparse eigenvectors, they successfully separate coherent vortices from unclustered background noise.

2. Conclusion

The abstract embedding of particle trajectories in a metric space with subsequent clustering is a promising field of research for the detection of finite-time coherent sets in oceanography~~, as it can be potentially applied to sparse sets of trajectories e. g. from drifter release experiments.~~ . Yet, most of the existing methods ~~lack the ability to separate finite-time coherent structures from noisy trajectories that do not belong to any such structure, which hampers the application to large ocean domains. This is because the clustering methods proposed so far~~ have been based on graph partitioning, which ~~treats~~ has no concept of noisy, unclustered ~~data points insufficiently. In this article, we presented a simple way to overcome this problem by using~~ trajectories.

19

This is a problem for applications in the ocean, where many eddies are transported in a noisy background flow on large domains. This study is motivated by the success of Froyland et al. (2019) in overcoming the problem of graph partitioning by a sophisticated form of trajectory embedding. Here, we show how the density-based clustering algorithm OPTICS (Ankerst et al., 1999) can be used instead of graph partitioning, in order to detect small-scale eddies in large ocean domains. Different from ~~partition-based~~ partition-based clustering methods such as k-Means, OPTICS ~~detects the clustering structure of the embedded trajectories by looking for~~ does not require to fix the number of clusters beforehand. Clusters are detected by identifying dense accumulations of points, i.e. groups of trajectories that are close to each other in embedding space. Coherent groups of particle trajectories can be identified as valleys in the reachability plot computed by the OPTICS algorithm. This plot also has a natural interpretation in terms of cluster hierarchies, i.e. finite-time coherent sets that are by themselves part of a larger scale finite-time coherent set. Such hierarchies are present in the surface ocean flow, where the subtropical basins are approximately coherent and at the same time ~~comprise~~ contain other finite-time coherent structures such as eddies and jets. ~~This hierarchical property is a clear advantage compared to DBSCAN, which has been used before to detect coherent sets (Schneide et al., 2018). One run of OPTICS can in principle produce all possible results of DBSCAN clustering for the same parameter ϵ_{min}. In addition, different from DBSCAN, OPTICS can detect clusters of varying density, and detect them by locating the valleys with the steepest boundaries in the reachability plot with the ζ-clustering method of Ankerst et al. (1999).~~

3. We have now also discussed the relation of our method to existing methods. In particular, we stress that we focus on a new clustering algorithm instead of a new form of embedding, as e.g. done by Froyland et al. (2018).

3.4 Comparison to related methods

Our method is closely related to existing methods to detect finite-time coherent sets with clustering techniques. Most notably, Froyland and Padberg-Gehle (2015) also use a direct embedding of individual trajectories similar to eq. (3), together with fuzzy-c-means clustering. Hadjighasem et al. (2016), Banisch and Koltai (2017), Padberg-Gehle and Schneide (2017) and Froyland and J use spectral embeddings of graphs that are defined on some form of physical intuition or of dynamical operators, together with k-Means clustering. These studies show applications of their methods to example flows where the size of almost-coherent sets is not too small compared to the fluid domain. Such examples are the Bickley jet flow, which we also study in section 4.1, the five major ocean basins (Froyland and Padberg-Gehle, 2015; Banisch and Koltai, 2017), or few individual eddies in an ocean or atmospheric flow (Hadjighasem et al., 2016; Padberg-Gehle and Schneide, 2017; Froyland and Junge, 2018). In such situations, noisy background trajectories can be detected as individual clusters by the partitioning method, as discussed by Hadjighasem et al. (2016). For applications in large ocean domains, where the number of eddies is not known beforehand and where there are many more noisy trajectories than coherent trajectories, such an approach is likely to fail, see also the discussion by Froyland et al. (2019). OPTICS does not require to fix the number of clusters beforehand, and also contains an intrinsic concept of noisy trajectories that do not belong to any cluster, making OPTICS suitable for challenging flows in large domains.

As mentioned, OPTICS also contains an intrinsic notion of cluster hierarchy, i.e. coherent sets that are themselves part of coherent sets at larger scales. Ma and Bollt (2013) studied hierarchical coherent sets in the transfer operator framework of Froyland et al. (2010), in the spirit of the hierarchical clustering method proposed by Shi and Malik (2000). Their approach is also partition-based, i.e. there is no concept of noisy trajectories. In addition, at each stage of the hierarchy, a fixed cut-off has to be chosen based on minimizing an objective function (Ma and Bollt, 2013). Different from that approach, the main result of OPTICS, the reachability plot, contains such hierarchical information in a smooth and intrinsic manner.

As described in section 3.3, clustering results of the DBSCAN algorithm (Ester et al., 1996) can be derived from the reachability

plot of OPTICS. DBSCAN has been used in the context of coherent sets before by Schneide et al. (2018), although not to identify specific clusters, but to distinguish noisy from clustered trajectories. The potential of density-based clustering for applications in the ocean and its comparison to other existing clustering methods for flow examples such as the Bickley jet (cf. section 2.1) has not been explored so far. Different from OPTICS, DBSCAN detects clusters with a certain fixed minimum density, although clusters with varying densities might be present in a dataset (Ankerst et al., 1999). More specifically, the value for the cut-off parameter ϵ, cf. section 3.3, has to be set beforehand. Choosing a good value for the density parameter in DBSCAN is challenging if there is no underlying physical intuition for the density structure. As described in section 3.3, OPTICS allows one to derive any DBSCAN clustering result, with the same value for the parameter s_{min}, after computing the reachability plot, i.e. after one can get first insights into the clustering structure of the dataset to make an appropriate choice for ϵ. Furthermore, it also allows one to use the ξ-clustering method instead of DBSCAN (cf. section 3.3).

A more recent and powerful technique to detect finite-time coherent sets in sparse trajectory data was presented by Froyland et al. (2019), based on dynamic Laplacian and transfer operators (Froyland and Junge, 2018). Froyland et al. (2019) apply their method to a trajectory dataset in the Western Boundary Current region in the North Atlantic Ocean, and successfully detect many eddies by superposing individual eigenvectors. The methods presented there are based on a form of spectral embedding, derived from discretized dynamical operators. Based on this embedding, clustering results have also been derived with k-Means by Froyland and Junge (2018) and with individual thresholding by Froyland et al. (2019). Froyland et al. (2019) also show how the low-order eigenvectors correspond to large-scale coherent features, while the individual eddies are derived by a sparse eigenbasis approximation of a number of eigenvectors. The latter approach is essentially a transformation of the embedding to represent the most reliable features, such that a superposition of the eigenvectors alone yields the information about the location and size of finite-time coherent sets (without a clustering step). This is essentially an optimized form of embedding, i.e. the second step in fig. 1. Our aim here is to focus on the third step in fig. 1, i.e. to demonstrate the potential of the density-based clustering algorithm OPTICS, together with a very simple embedding of eq. (3).

A downside of our method compared to other approaches is the rather ad-hoc choice of embedding, cf. eq. (3). Different from many other methods, most notably the ones of Banisch and Koltai (2017), Froyland and Junge (2018) and Froyland et al. (2019), this type of embedding is not derived from a meaningful dynamical operator. It could be fruitful to explore a combination of these more meaningful embeddings together with OPTICS as a clustering algorithm in future research.

Comment 4

(iv) For a short paper, the abstract is perhaps disproportionately long.

Answer to comment 4

Thanks for noting. We have shortened the abstract a bit in the new version.

Abstract. The detection of finite-time coherent particle sets in Lagrangian trajectory data using data clustering techniques is an active research field at the moment. Yet, the clustering methods mostly employed so far have been based on graph partitioning, which assigns each trajectory to a cluster, i.e. there is no concept of noisy, incoherent trajectories. This is problematic for applications ~~to~~ in the ocean, where many small coherent eddies are present in a large~~fluid domain. In addition, to our knowledge none of the existing methods to detect finite-time coherent sets has an intrinsic notion of coherence hierarchy, i.e. the detection of finite-time coherent sets at different spatial scales. Such coherence hierarchies are present in the ocean, where basin-scale coherence coexists with smaller coherent structures such as jets and mesoscale eddies.~~ , mostly noisy fluid flow. Here, for the first time in this context, we use the density-based clustering algorithm OPTICS (Ankerst et al., 1999) to detect finite-time coherent particle sets in Lagrangian trajectory data. Different from ~~partition based~~ partition-based clustering methods, ~~OPTICS does not require to fix the number of clusters beforehand. Derived~~ derived clustering results contain a concept of noise, such that not every trajectory needs to be part of a cluster. OPTICS also has a major advantage compared to the previously used DBSCAN method, as it can detect clusters of varying density. ~~Further, clusters can also be detected based on density changes instead of absolute density. Finally, OPTICS based clusters~~ The resulting clusters have an intrinsically hierarchical structure, which allows one to detect coherent trajectory sets at different spatial scales at once. We apply OPTICS directly to Lagrangian trajectory data in the Bickley jet model flow and successfully detect the expected vortices and the jet. The resulting clustering separates the vortices and the jet from background noise, with an imprint of the hierarchical clustering structure of coherent, ~~small-scale~~ small-scale vortices in a coherent, large-scale, background flow. We then apply our method to a set of virtual trajectories released in the eastern South Atlantic Ocean in an eddying ocean model and successfully detect Agulhas rings. ~~At larger scale, our method also separates the eastward and westward moving parts of the subtropical gyre.~~ We illustrate the difference between our approach and ~~partition-based~~ partition-based k-Means clustering using a 2-dimensional embedding of the trajectories derived from classical multidimensional scaling. We also show how OPTICS can be applied to the spectral embedding of a ~~trajectory-based~~ trajectory-based network to overcome the problems of k-Means spectral clustering in detecting Agulhas rings.

Comment 5

There is one point raised in the paper that I felt required more elaboration. A selling point the authors bring up for this method is that it can in principle be applied to real-world trajectory data, see line 86 and also line 306. This is true but incomplete. Real-world Lagrangian instruments are sufficiently sparse that it is rare to find more than one in the same eddy at the same time. Thus, the application presented herein—finding eddies is idealized configurations—is not really relevant for how one would apply this method to real-world trajectories. The data density used here is orders of magnitude greater than for real-world instruments. Since the authors bring this up as an advantage of the method, a more fair and nuanced discussion of its potential and limitations with respect to real-world data is called for. I would say, rather, that the method seems more suitable in application to model data or virtual trajectories from altimetry, where it benefits from a simplicity with respect to some other proposed methods.

Answer to comment 5

The reviewer is correct that an application of our method to real drifters to detect eddies is not possible due to the limited coverage of drifter data. Note that two studies applied their methods to real drifters, as we mentioned in the introduction (Froyland and Padberg-Gehle (2015) and Banisch and Koltai (2017)), however to detect the five major ocean basins and not eddies. In the new version, we omit the reference to real ocean drifters at other places but the introduction, where we now explicitly mention the application to ocean basins (and not eddies).

1. Changes in introduction to clarify that trajectory-based clustering has been applied to real drifter data only in the context of detecting the ocean basins, not individual eddies.

The detection of coherent Lagrangian vortices using abstract embeddings of Lagrangian trajectories together with data clustering techniques has received significant attention in the recent literature ~~(Froyland and Padberg-Gehle, 2015; Hadjighasem et al., 2016; Padl . Examples include the direct embedding of trajectories in a high dimensional Euclidean space (Froyland and Padberg-Gehle, 2015) , or more abstract embeddings based on related networks constructed from particle trajectories (Hadjighasem et al., 2016; Padberg-Gehl~~ (Froyland and Padberg-Gehle, 2015; Hadjighasem et al., 2016; Padberg-Gehle and Schneide, 2017; Banisch and Koltai, 2017; Schne . Using embedded trajectories for the detection of finite-time coherent sets is interesting as it allows ~~to use scarce~~ one to use sparse trajectory data, and it can in principle be applied to ocean drifter trajectories, as ~~done~~ demonstrated by Froyland and Padberg-Gehle (2015) and Banisch and Koltai (2017) for the detection of the five ocean basins. Yet, ~~the methods proposed so~~

2. End of the introduction

In section 4, we first show how OPTICS detects finite-time coherent sets at different scales for the Bickley jet model flow (also discussed e.g. by Hadjighasem et al. (2017)), successfully detecting the six coherent vortices and the jet as the steepest valleys in the reachability plot. The general structure of the reachability plot also reveals the large-scale finite-time coherent sets, i.e. the northern and southern parts of the model flow, separated by the jet. We then apply our method to Lagrangian particle trajectories released in the eastern South Atlantic Ocean, where large rings detach from the Agulhas Current (e.g. Schouten et al. (2000)). We detect several Agulhas rings, and on the larger scale also separate the eastward and westward moving branches of the South Atlantic Subtropical Gyre. While the traditional approach to study Agulhas rings is based on sea surface height analysis (see e.g. Dencausse et al. (2010)), several methods based on virtual Lagrangian trajectories have been applied to Agulhas ring detection before (Haller and Beron-Vera, 2013; Beron-Vera et al., 2013; Froyland et al., 2015; Hadjighasem et al., 2016; Tarshish et al., 2018). Our method is different from these approaches in that it is directly applicable to a trajectory ~~data set~~dataset, i.e. without much pre-processing of the data. As the OPTICS algorithm is readily available in the sklearn package of SciPy, the detection of finite-time coherent sets can be done without much effort and with only a few lines of code. A further difference is the mentioned intrinsic notion of coherence hierarchy, which allows for simultaneous analysis of trajectory data at different scales. ~~Finally, trajectory based approaches can in principle be applied to scarce trajectory data, i.e. to any Lagrangian particle simulation result without much care for the spatial coverage of the initial conditions.~~ While we mainly focus on the direct embedding of trajectories in an abstract high-dimensional Euclidean space, we also show in ~~section C in the appendix~~ appendix C that OPTICS can be used to overcome the limits of k-Means clustering in the context of spectral clustering of ~~physically motivated trajectory based networks, such as the works presented by Hadjighasem et al. (2016), Padberg-Gehle and Schneide (2017) or Banisch and Koltai (2017)~~the trajectory-based network of Padberg-Gehle and Schneide (2017).

2. First sentence in conclusion

The abstract embedding of particle trajectories in a metric space with subsequent clustering is a promising field of research for the detection of finite-time coherent sets in oceanography~~, as it can be potentially applied to sparse sets of trajectories e. g. from drifter release experiments.~~ Yet, most of the existing methods ~~lack the ability to separate finite-time coherent structures~~

Comment 6

Line 99. Do you not want to cite Bickley? My understanding is that the term “Bickley jet” itself is used to refer to a steady solution with a sechˆ2 u-velocity, see e.g. Swaters (1999). The authors’ Eq. (2) is an added perturbation. As read, it sounds like the whole thing is the Bickley jet.

Answer to comment 6

Thank you for this comment and the careful check of our references of the flow. You are indeed right that the Bickley Jet is a steady, sechˆ2 velocity profile. We have added the reference to Bickley now,

together with a reference to the paper of del Castillo-Negrete and Morrison (1993), where the perturbed form of the jet is motivated.

Comment 7

Section 3.2.2. I didn't really understand this section, or what B is encoding in Eq. (4). A more intuitive description would be helpful. When you say, "pairwise distances are approximately preserved", this is with respect to what? Also, why are two dimensions chosen?

Answer to comment 7

Thank you for this comment. In the new version, we elaborate more on the intuitive goal of classical MDS in this section. We choose two dimensions because we wish to visualize the data in the plane. We have made this more clear in the new version.

3.2.2 ~~Classical~~ Dimensionality reduction with classical multidimensional scaling

To get an intuition for what the OPTICS algorithm does, and the differences to k-Means, we wish to visualize the data structure in the ~~clustering results of the OPTICS algorithm , we visualize the density structure of the trajectories in the 2-dimensional plane~~plane. For this, it is necessary to reduce the embedding dimension of each trajectory from $3T$ to two in a way that the density structure, and hence the individual Euclidean distances between embedded trajectories $d_{ij} = ||u_i - u_j||$, cf. eq. (3), are preserved. We do so by a common method of nonlinear dimensionality reduction, called classical ~~Multidimensional~~ multidimensional scaling (MDS), see e.g. chapter 10.3 of Fouss et al. (2016). Classical MDS tries to find an embedding of the high-dimensional data points in a low dimensional space such that the pairwise distances are approximately preserved.

~~Classical~~ Similar to a principal component analysis, classical MDS makes use of the eigenvectors corresponding to the largest eigenvalues of ~~the kernel matrix~~ a kernel matrix, which is in this case defined by

$$B = -\frac{1}{2} H \Delta^2 H, \tag{4}$$

where $\Delta^2 \in \mathbb{R}^{N \times N}$ is a matrix containing all squared distances between the points, $\Delta^2_{ij} = ||u_i - u_j||^2$, and H is the centring matrix with $H_{ij} = \delta_{ij} - 1/N$, where δ_{ij} denotes the Kronecker delta. The matrix B in eq. (4) is called the centred inner product matrix. If $\bar{B}$ is the matrix of inner products of the embedded data points, i.e. $\bar{B}_{ij} = u_i \cdot u_j$ with Euclidean scalar product, then B can be obtained by removing the mean of all rows and columns of $\bar{B}$, cf. chapter 10.3 of Fouss et al. (2016). An embedding of the data points using the eigenvectors corresponding to the leading non-negative eigenvalues of B in eq. (4) ensures to capture the main variance of the (squared) distance structure, similar to a principal component analysis.

We compute Δ^2 with the Euclidean ~~embeddings~~ embedding described in section 3.2.1 and restrict ourselves to the first two dimensions to visualize the data structure in the plane, i.e. the embedding is defined by

$$u_i = (w_{0,i}, w_{1,i}),\ i = 1, \ldots, N, \tag{5}$$

where $K w_j = \lambda_j w_j$, and $\lambda_0 \geq \lambda_1 \geq \lambda_k$ for all $k = 2, \ldots N-1$. This choice of embedding ensures to capture the main variance of the data points, and we therefore also expect to capture the main structure in terms of data density. For large particle sets however, computing the spectrum of H in eq. (4) is computationally not feasible, as the matrix B is ~~in general~~ dense and computing the spectrum scales with $O(N^3)$. We apply classical MDS to the 12,000 particles of the Bickley jet model flow, and a random selection of the equal number of particles for the Agulhas flow. In our context, the method is most useful for visualization purposes, as it provides a good 2-dimensional approximation of the point distances, i.e. also the density structure of the embedded trajectories.

Comment 8

Line 193. The intuitive meaning of the ‘generating distances’ that are not being used here should be mentioned

Answer to comment 8

Than you for the comment. In the new version, we briefly mention what a finite generating distance would mean.

For $\delta \in \mathbb{R}$, the δ-neighbourhood of a point $p \in \mathbb{R}^M$ is defined as the M-dimensional ball of radius δ around p. Define ~~$M_\varepsilon(p)$~~

$M_\delta(p)$ as the number of points that is in the ~~ε~~δ-neighbourhood of p, including p itself. OPTICS requires one parameter, an integer s_{min} (called MinPts by ~~Ester et al. (1996)~~Ankerst et al. (1999)), that defines the *core-distance* of a point p as

$$c(p) = \{\min(\varepsilon\delta) \mid M_{\varepsilon\delta}(p) \geq s_{min}\}. \quad (6)$$

The core distance is simply the minimum radius of a ball around p, such that the ball contains s_{min} points. Note that the generating distance that we set to infinity is a maximum cut off distance for the computation of the core distance in eq. (6), beyond which the core distance is not defined. As we do not have an intuition for a good value of such a cut off, we remove it by setting it to infinity.

Comment 9

Line 196. The definition of the epsilon neighborhood appears incomplete. Is it not the M-dimensional sphere of radius epsilon? Otherwise, what is the epsilon?

Answer to comment 9

Indeed the epsilon-neighborhood of p is just the M-dimensional ball around the point p, and the previous version was incomplete. We have changed this in the new version, together with renaming epsilon to delta, see our answer to comment 8.

Comment 10

Line 200. It would be very helpful to write out in words the meaning of Eq. (6). My understanding is that c(p) is minimum distance epsilon such that the number of points in an epsilon neighborhood is greater than a specified number.

Answer to comment 10

Thank you for your comment. Your interpretation was correct. We have made it more clear in the new version, see the answer to comment 8.

Comment 11

Line 213. I did not immediately understand how it arises that there are valleys in the reachability if you have sorted iteratively on the reachability. You might explain that this happens as you encounter

groups of points that are all near to each other, thus replacing earlier high values of reachability with lower values.

Answer to comment 11
Thank you for the comment. Indeed, it is the sorting that is the most important step in the algorithm. We added some more explanation in the new version.

Note that the ordering of points is achieved by constantly updating the ordered seed list, cf. step 3. In this way, the algorithm
275 iterates through groups of dense points one after the other, and only continues with other points once a dense region has been fully explored. Note also that the entire algorithm depends on the choice of the parameter s_{min}. The value of s_{min} should be chosen roughly as a minimum value of the expected cluster size. In the examples presented in this paper, we take values for s_{min} that correspond to the estimated minimum size of the coherent sets.

Comment 12
Line 216. The phrasing here made me wonder if this was a second, different epsilon. It would be clearer to say that you choose a value for the parameter epsilon. Also, it appears this is conditional on a choice of s_min which should then be emphasized.

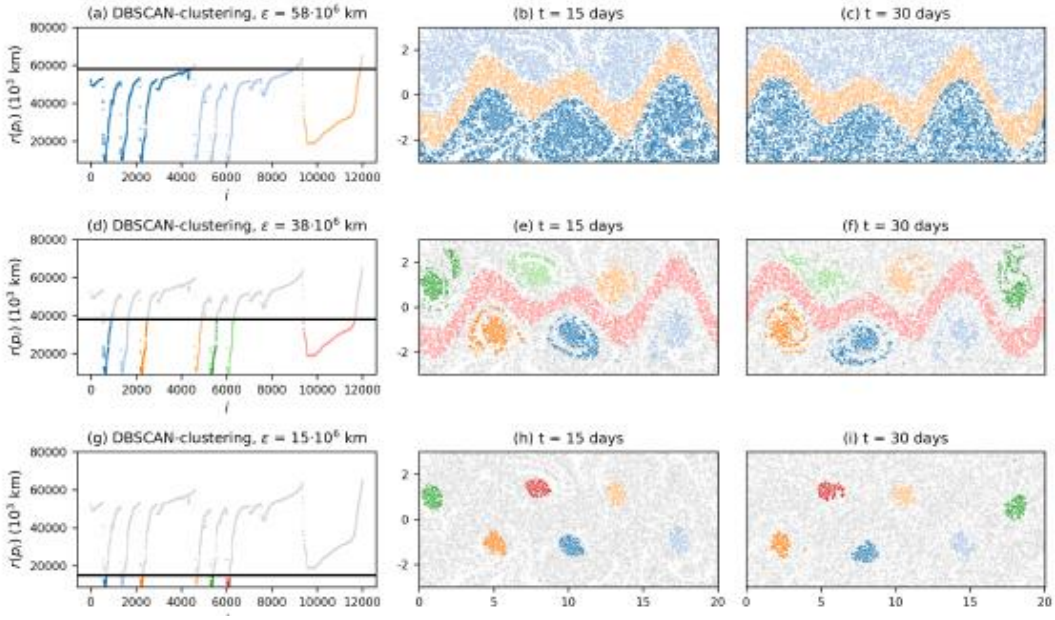
Answer to comment 12
Thank you for very much for pointing this out. Indeed, this was a second epsilon, and the presentation in the first version was confusing. We have made the appropriate changes in the new version by re-naming one of the epsilons into delta. See our answer to comment 8.

Comment 13
Line 228. What are the permissible values of k in condition (a)?

Answer to comment 13
We have made this more precise in the new version. It can be any integer larger than zero and smaller than N - l.

Comment 14
Figure 2, what are the units of the y-axis in the left column of plots?

Answer to comment 14
Thank you for this comment. Indeed, we missed to specify the units of all reachability values. We do so in all figures in the new version (apart from the network embedding case in the appendix, where quantities are dimensionless), see the example below.



Comment 15

Figs 2 and 3, some of the colored dots lie above the epsilon threshold.

Answer to comment 15

This is correct, DBSCAN classifies the points below the line only up to boundary points, i.e. there can be points at the cluster boundary that belong to the cluster. We have made this more clear in the new version.

4 Results

4.1 Bickley jet flow

375 We start with the direct embedding of the ~~trajectories. As explained in section 2, the~~ Bickley jet flow trajectories, cf. section 2. The data matrix has dimension ~~$X \in \mathbb{R}^{12,000\times 143}$~~ $X \in \mathbb{R}^{12,000\times 123}$. We apply the OPTICS algorithm to the resulting points ~~,~~ together with DBSCAN clustering, choosing $s_{min} = 80$ as a minimum size of the finite-time coherent sets. In the following, all axis units are in multiples of 1000 km. Figure 2 shows the reachability plot, together with the DBSCAN clustering result of three different choices of ϵ. The six vortices and the jet are clearly visible as the major valleys in the reachability plot. The hierachical

---

380 structure of the DBSCAN clustering with decreasing ϵ is visible in the figures from top (~~larg scale~~ large-scale coherence) to bottom (~~small scale~~ small-scale coherence). ~~Being able to study this hierarchical structure with one run of OPTICS is a major advantage compared to DBSCAN and other methods to detect finite-time coherent sets. Note again that one run of OPTICS provides~~ Note that for the DBSCAN clustering ~~result of any parameter ϵ (with the same s_{min}).~~ results, boundary points of the clusters can be above the hozitonal line at $y = \epsilon$. This is because of the definition of the DBSCAN clustering in section 3.3.

Comment 16

Figure 4. I really don't understand the two dimensions of these plots, nor the star-shaped patterns, could you explain these more?

Answer to comment 16

We have now made the presentation of the methods regarding classical MDS more clear, also relating it to principal component analysis, see the answer to your comment 7. In addition, we have provided more explanation on the star-shaped structure in the results section.

385 ~~Next~~To illustrate the difference between OPTICS and k-Means, we use the embedded trajectories and apply classical MDS to obtain a 2-dimensional embedding. As ~~mentioned~~ described in section 3.2.2, this assures to capture the major variance along the embedding axes. The spectrum of B in eq. (4) is shown in fig. A1 in the appendix, with two clearly dominant eigenvalues. ~~Figure 4 shows the result of OPTICS for this case of embedding. Most notably, applying MDS has lead to the vortices and the jet having comparable depth in the reachability plot, such that a single DBSCAN clustering result detects all six vortex~~ 390 ~~centres and the jet in the middle~~The fact that there are two very dominant eigenvalues assures that the illustration of the data in the plane captures the major variance of the data points. Figure 3a shows the corresponding embedding of the trajectories in the 2-dimensional ~~embedding space, and fig. 3b~~ Euclidean space. The star-shaped distribution of data points reflect the strong

symmetries of the underlying idealized Bickley jet flow. Such symmetry is not expected to be present for more realistic flows. Figures 3b and 3c show the cluster labels for OPTICS with DBSCAN clustering at ~~$\epsilon = 1000$, as shown in fig. 4. The jet and the~~ 395 ~~six vortices are clearly recognizable as dense accumulations of points in this 2-dimensional space. Figure 3c shows the result~~ $\epsilon = 10^6$ km, and for a k-Means clustering with $K = 8$ clusters, ~~which~~ respectively. $K = 8$ corresponds to the six vortices, the jet, and one noise cluster as suggested by Hadjighasem et al. (2016).

In addition, we have further discussed the failure of k-Means in relation to the star-shaped structure of the embedding.

The corresponding clustering ~~result is shown in fig. 5 in the appendix, showing~~ results in real space are shown in figs. 4 and 5 for OPTICS and k-Means, respectively. The jet and the six vortices are clearly recognizable as dense accumulations of points in 400 the 2-dimensional space of fig. 3b, see fig. 4 for the corresponding colours. The clustering result with k-Means in fig. 5 shows that the clusters corresponding to the vortices are much less focussed. In addition, each of the eight clusters in fig. 3c contains some of the noisy points of fig. 3b, which shows that using one additional cluster for noise does not ~~really address the issue of not detecting the vortices properly for this case of embedding~~work in this situation. It is interesting to note that capturing the noisy data points of fig. 3b by an additional cluster in k-Means is geometrically impossible, simply because k-Means clusters 405 are circular. Covering all noisy points without including the centre, i.e. the jet in fig. 3b, is not possible for k-Means.

Comment 17

Data locations at Zenodo should be cited, not only the papers referring to them.

Answer to comment 17

The reference is actually a Zenodo link, not a paper. Note that there were two references Wichmann 2020 (Zenodo link) and Wichmann et al. (2020) (previous paper). In the new version, there is now also a Zenodo link to an animation for the Agulhas flow.

Comment 18
Throughout the paper, the authors consistently omit the subject ahead of an infinitive, e.g. “which allows to detect”. I believe this is grammatically incorrect (in US usage anyway). “allows one to detect” or “allowing the detection of” sound better

Answer to comment 18
Thank you, we have made appropriate changes in the new version.

Comment 19
l 42 and 90. “sparse” should probably be used instead of “scarce”. The former means thinly distributed while the latter means hard to come by.

Answer to comment 19
We have made appropriate changes in the new version.

Comment 20
l 128. NumPy and Zenodo are the standard capitalizations

Answer to comment 20
We made the suggested changes in the new version. Thank you for noting.

Comment 21
l 141. “method” should be “methods”

Answer to comment 21
Thank you for noting, we corrected it in the new version.

Comment 22
l 156. Straightforward

Answer to comment 22
Thank you for noting, we corrected it in the new version.

Comment 23
l 191. “and as will become clear”

Answer to comment 23
Thank you for noting, we corrected it in the new version.

Comment 24
l 217. “is equal to” should be “set of points is equivalent to”.

Answer to comment 24
Thank you for noting, we corrected it in the new version.

Comment 25
l 243. “a priory” should be “a priori”

Answer to comment 25
We corrected it in the new version.

Comment 26
l 279. “large- and small-scale”

Answer to comment 26
Thank you for noting, we corrected it in the new version.

Comment 27
l 354. GitHub

Answer to comment 27
Thank you for noting, we corrected it in the new version.

Comment 28

l 359. There is a title of an appendix with no appendix.

Answer to comment 28
The content of appendix C consisted of only two figures, C1 and C2. It appeared as without content due to the page break. In the new version, we have removed one appendix as we include the figures in the main text, such that the formatting looks better.

Comment 29
l 360 & 361. “particle-based”

Answer to comment 29

Thank you for noting, we corrected it in the new version.

Comment 30
l 383. “ot”

Answer to comment 30
Thanks for the careful read, we made the changes in the revised manuscript.

Comment 31
l 389. There should be a period at the end of this sentence

Answer to comment 31
Done. Thanks for noting.

Comment 32
Figure C1, “three” eigenvalues should be “two”, correct?

Answer to comment 32
Yes, indeed. Thanks for reading also the appendix figure captions so carefully! We corrected this in the new version.
