Responses to reviewer 3#'s comments point by point

MS No.: essd-2025-384

Title: A Black Hole Eddy Dataset of North Pacific Ocean Based on Satellite Altimetry

Author(s): Fenglin Tian et al.

Dear Reviewer,

We highly appreciate the detailed and valuable comments of the referee on our manuscript entitled "A Black Hole Eddy Dataset of North Pacific Ocean Based on Satellite Altimetry (ID: essd-2025-384)". These comments are all valuable and helpful for revising and improving our paper, as well as providing important guidance for our research. In the past few days, we have referred to the comments and improved the paper.

As follows, we would like to clarify some of the points raised by the Reviewer. The original comments begin with "**Comment**" and are quoted in italicized font, the responses begin with "**Response**" in normal font, the original sentences and phrases are in blue letters, the revised sentences and phrases are in red letters, and the line number in the revised manuscript is highlighted in yellow. We appreciate the Reviewer's warm work and taking the time to review the manuscript, and we hope that the corrections will meet with approval.

Yours Sincerely,

Fenglin Tian, Yingying Zhao, Lan Qin, Shuang Long, Ge Chen

2025-9-30

Comment:

The paper is describing operational methods for the identification of Lagrangian structures (Black Hole Eddies) and transport analysis. Authors are underlining the computational aspect of their methodology.

Vortices are analyzed by using Cauchy - Green strain tensor in two dimensions, calculating distances from grid points in four directions and introducing auxiliary points.

Lorentz transformation, extremal curves with the shortest possible path length or as the curves that depart as little as possible from straightness (they are "locally straight") - i.e. geodesic, are important elements for the analysis. It is assumed that the absolute velocity vanishes.

I must confess that I find it difficult to follow the logical development of the procedures used for the detection and spatial evolution of BHEs.

My first doubt lies in the authors' development of the strain tensor in two dimensions, when they later admit that the flow field is three-dimensional. The transition to polar coordinates is unclear to me. I understand that this step could make the algorithm more efficient, but no explanation is provided.

The impression that emerges from reading the text is that the authors have taken some interesting developments from the literature, but how they are implemented in the context of the publication is somewhat cumbersome.

Response:

Thank you for your detailed review and for providing such valuable and constructive feedback on our manuscript. Your professional insights are incredibly helpful, and we believe that addressing them will significantly improve the clarity and rigor of our paper. We have carefully considered all your comments and plan to revise the manuscript accordingly.

Below are our point-by-point responses and our plan for revision:

Comment 1: "*I must confess that I find it difficult to follow the logical development of the procedures used for the detection and spatial evolution of BHEs.*"

Response: Thank you for your valuable feedback. We completely agree that the clarity of the Methods section is of utmost importance for readers to understand our work.

We recognize that while we provided a flowchart (Figure 1) in our original manuscript to outline the overall framework, it may have only presented the steps at a high level. We understand that it may not have sufficiently revealed the complex computational logic within each step, nor the explicit link between the theoretical formulas and the practical implementation. This likely contributed to the difficulty in following the logical development of our methodology.

To thoroughly address this, and in response to your comment about the implementation being

“somewhat cumbersome”, we have taken a key improvement measure in the revised manuscript: we have added a detailed algorithm pseudocode (Algorithm 1) to the Appendix A section.

We believe this new pseudocode perfectly complements the existing flowchart and will greatly enhance the clarity of our methodology.

The flowchart shows what we do, while the pseudocode clearly explains how we do it. It breaks down each high-level step into a concrete, logical sequence of computational operations. The pseudocode serves as a bridge between abstract mathematical formulas and their concrete implementation. We have included comments alongside key steps in the pseudocode that directly reference the corresponding equation numbers in the text, , which outlines the numerical procedure used to extract BHEs via null geodesic calculation. The pseudocode explicitly includes the main steps performed by the PYTHON code with GPU parallelization. This addition aims to make the implementation transparent and reproducible for users. we have added the pseudocode as follows: (Please see Line 868-870)

Appendix A: GPU-Accelerated Pseudo-Code for BHE Detection Using Null Geodesics

Algorithm 1 provides a brief summary of the main steps performed by the PYTHON code for BHE detection based the GPU-accelerated null geodesics algorithm.

Algorithm 1. BHE detection based the GPU-accelerated null geodesics algorithm.

Input:

- u(x, y, t), v(x, y, t): 2D velocity field in time window $[t_0, t_1]$

- Range of stretch rate $\lambda \in [0.9, 1.1]$ with step size $\Delta\lambda$

- Grid of initial polar angle $\theta \in [0, 2\pi]$

- dt: integration time step

- grid resolution: Resolution of the grid

- domain bounds: Domain boundaries for the particle seeding

#Step 1: Load preprocessed velocity field (u, v)

VelocityField,velocity_field = load_velocity_data("u_field.nc", "v_field.nc");

#Step 2: Initialize particle grid and allocate on GPU(see equations (5) and (6))

Grid grid = initialize_uniform_grid(resolution=1/32.0);

Particle* d_particles = allocate_particles_on_GPU(grid);

#Step 3: Integrate particle trajectories using RK4 (GPU parallelism)

RK4_integrate<<<gridDim, blockDim>>>(velocity_field, d_particles, time_step=0.1, total_steps);

#Step 4: Compute Cauchy-Green strain tensor on GPU (see equation (3))

compute_CG_tensor<<<gridDim, blockDim>>>(d_particles, d_C_tensor, grid);

#Step 5: Perform eigendecomposition for each C tensor (on GPU) (see equation (4))

eigendecompose_C_tensor<<<gridDim, blockDim>>>(d_C_tensor, d_lambda1, d_lambda2, d_xi1, d_xi2);

#Step 6: Extract null geodesics using ZeroSet condition (each thread for φ at fixed λ) (see equation (12))

for λ in linspace (0.9, 1.1, 9): #λ from 0.9 to 1.1 with step of 0.025

extract_null_geodesics<<<gridDim_lambda,blockDim_phi>>>(d_C_tensor,λ,d_phi_array,d_null_flags;

#Step 7: Check which geodesics are closed based on φ = 2π

check_geodesic_closure<<<gridDim_geo,blockDim>>>(d_null_flags,d_phi_diff_array,d_closed_flags);

#Step 8: Integrate valid seed points inside tensor field to generate nested LCS(see equations (13) and (14))

integrate_seeds_in_tensor<<<gridDim_seed,blockDim>>>(d_C_tensor,d_seeds,d_nested_curves,target_phi=2π);

#Step 9: Transfer nested curves to CPU for clustering and BHE boundary extraction

download_curves_to_host(d_nested_curves, nested_curves);

clusters = cluster_curves_by_centroids(nested_curves);

for cluster in clusters:

    areas = compute_area_of_each_curve(cluster);

    max_curve = find_curve_with_max_area(areas);

    save_BHE_boundary(max_curve);

Output:

- A closed contour representing the boundary of Black Hole Eddy

---

Comment 2: "*My first doubt lies in the authors' development of the strain tensor in two dimensions, when they later admit that the flow field is three-dimensional.*"

Response: Thank you for highlighting this crucial point of ambiguity. You are correct to question the use of "three-dimensional", and we appreciate the opportunity to clarify our terminology and its context.

Our analysis is performed in two spatial dimensions (longitude and latitude) because our input data consists of satellite-derived surface geostrophic velocities. This dataset is inherently two-dimensional. The confusion arises from two specific sentences in our manuscript where our terminology was imprecise. We will correct both instances to ensure clarity and consistency. Below, we explain the issue with each sentence and provide a comparison of the original and revised text.

1. The first instance appears in our description of the algorithm's mathematical formulation. The term "three-dimensional" here refers to the mathematical state space of the solver (two spatial coordinates, x and y, and an angle variable, ϕ), not the physical dimensions of the ocean flow. While this is a mathematically accurate description of the method from Serra and Haller (2017), we now recognize it is highly ambiguous for the reader. We will revise the sentence to clarify this context.

- Original sentence:

  "...the proposed approach formulates a unified three-dimensional initial value problem...",

- Revised sentence:

  "...formulates a unified initial value problem in a three-dimensional state space (two spatial coordinates and an angle variable), enabling fully automated and robust vortex boundary detection (Serra and Haller, 2017)" (Please see Line 257)

2. The second instance is the primary source of the confusion and is incorrectly phrased in a fluid dynamics context. Our intention was to describe a flow field that is two-dimensional in space but evolves over time. We will completely rephrase this to accurately describe the system.

- Original sentence:

  "Given the three-dimensional nature of the flow field-encompassing longitude, latitude, and time..."

- Revised sentence:

  "As particles move through the unsteady, two-dimensional flow field (defined in longitude, latitude, and time), interpolation is required to estimate their trajectories." (Please see Line 323-324)

Reference:

Serra, M. and Haller, G.: Efficient computation of null geodesics with applications to coherent vortex detection, Proceedings of the Royal Society a-Mathematical Physical and Engineering Sciences, 473, https://doi.org/10.1098/rspa.2016.0807, 2017.

Comment 3: *The transition to polar coordinates is unclear to me. I understand that this step could make the algorithm more efficient, but no explanation is provided.*

Response: Thank you for this insightful feedback. We have expanded our explanation to clarify its fundamental importance to our methodology. The approach from Serra et al. (2017), which we refer to

as the geodesic-based BHE identification algorithm, simplifies the calculation of closed null geodesics. Its key innovation is the introduction of polar coordinates, which transforms the challenge from solving a variational problem to solving a more straightforward functional problem. This simplification is what enables the efficient identification of vortex boundaries. In the revised text, we now explicitly state that this is the reason for the transition.

- Original sentence:

"... Each discrete λ yields a corresponding geodesic calculation. In the process of calculating geodesics and extracting null geodesic contours, the original algorithm is inefficient, as it requires significant time to compute and traverse each contour individually. However, since these contours are independent of one another, we can enhance the computation process by employing a GPU parallel acceleration algorithm. In this approach, each thread corresponds to a single geodesic line, allowing for simultaneous calculations and significantly reducing processing time. Utilizing a polar coordinate system, the partial derivatives of the tensor field calculated in equation are used as inputs, the computation steps for the null geodesic field adhere to the following equation: ..."

- Revised sentence:

"...with each discrete λ yielding a corresponding geodesic calculation. The traditional approach of calculating these geodesics and extracting null geodesic contours individually is computationally inefficient. To overcome this problem, we adopt the key innovation from Serra et al. (2017) by introducing a polar coordinate system. This transforms the challenge from solving a complex variational problem into a more straightforward functional problem. The primary advantage is that a vortex boundary-a closed loop by definition-can be rigorously identified by testing if the rotation angle,ϕ, completes a full 2π rotation. This mathematical simplification is particularly well-suited for parallelization, as it reframes the problem as a large set of independent calculations. We leverage this structure by employing a GPU-based parallel algorithm. In this approach, each thread is assigned the task of computing a single geodesic line, allowing for thousands of simultaneous calculations and thus reducing the overall processing time. Following this parallelized approach within the polar framework, the partial derivatives of the tensor field are used as inputs, and the computation steps for the null geodesic field adhere to the following equation: ..." (Please see Line 365-377)

Comment 4: *The impression that emerges from reading the text is that the authors have taken some interesting developments from the literature, but how they are implemented in the context of the publication is somewhat cumbersome.*

Response:

Thank you for this insightful feedback. We agree that the connection between the theoretical concepts and our specific computational implementation needs to be strengthened. To address these points together and resolve the overall impression of a "cumbersome" implementation, we have significantly revised our Methods section to provide a clearer and more detailed account of our computational approach.

Our revisions include two new schematic diagrams (Fig. 2 and Fig. 3) and a detailed algorithm pseudocode (now in Appendix A), which work together to bridge the gap between theory and practice.

First, to demystify the specific computational details, we have added two new figures.

Figure 2 now illustrates our grid configuration, showing how we enhance simulation accuracy by densifying the original 1/4° velocity field grid by a factor of 8 using main and auxiliary nodes for the Cauchy-Green strain tensor calculation. (Please see Line 303-306)

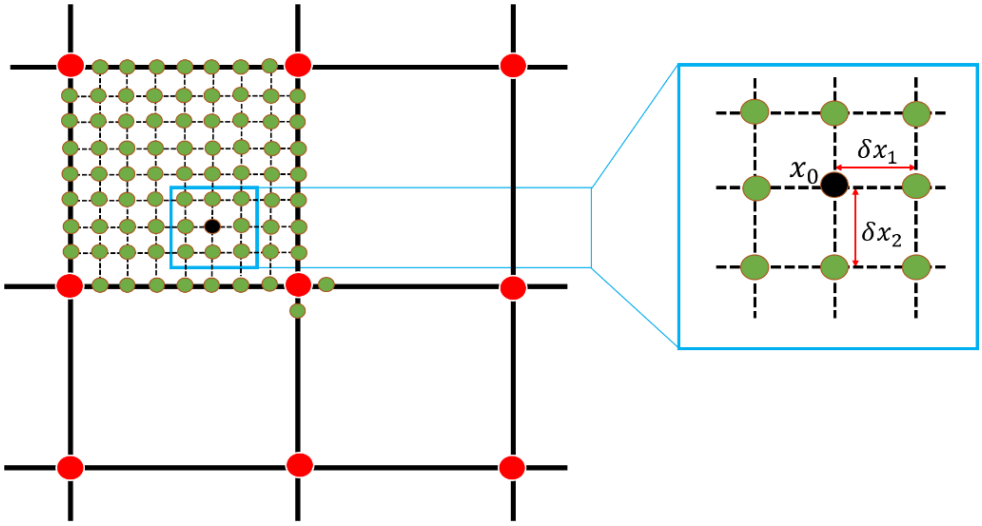

Figure 2. Schematic diagram of the spatial relationship between a main grid point and its surrounding auxiliary points in the Cauchy-Green strain tensor field. The black dot represents an auxiliary grid point x_0, δx_1 and δx_2 denotes the distance between the auxiliary grid point (green) and the main grid point (red).

Figure 3 provides a clear schematic of our CUDA thread allocation logic, explaining how the grid and block sizes are determined based on the input grid and GPU capacity. Once the size of the initial grid and the number of threads per Block in the GPU are inputted, the size of the Block to be allocated for each Grid can be determined (Fig. 3). (Please see Line 351-353)

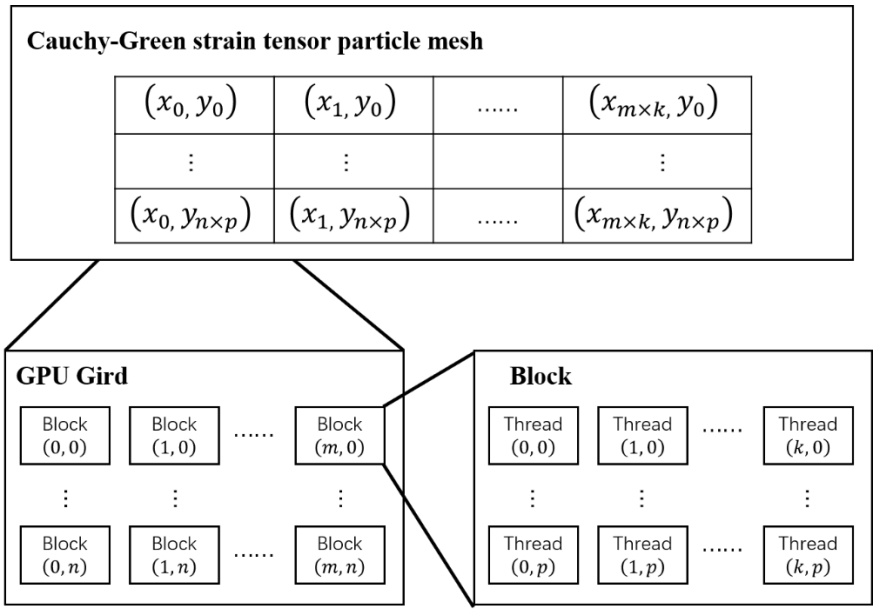

Figure 3. Thread allocation diagram.

Complementing these diagrams, the new pseudocode in Appendix A provides the overarching algorithmic logic (**see response1**). It breaks down abstract concepts (like the Cauchy-Green strain tensor and null geodesics) into a clear sequence of computational steps (e.g., loops, conditional statements, numerical integration). To make the connection explicit, we have included comments within the pseudocode that refer back to specific equation numbers in the text.