We appreciate the reviewer's careful and thoughtful comments and have made extensive edits to address these, which are tagged and marked in blue within the revised text, the attached gives detail responses

Review #1

Review of "OceanTracker 0.5: Fast Adaptable Lagrangian Particle Tracking in Structured and Unstructured Grids" by Vennell et al

In this manuscript, the authors showcase the OceanTracker software and describe its features and design decisions. They compare its performance in some realistic cases to the OpenDrift and Parcels frameworks.

My main comments are

1. I miss a discussion of input reading as a performance bottleneck for Lagrangian simulations. Depending on the number of nodes in the grid, this is in e.g. Parcels the major control on execution time in Lagrangian simulations.
    - Good idea, added new subsection to speed section 4.1, see (Aa line 538)
2. In general, I miss some critical reflection on the downsides of some of the choices made in the OceanTracker design (e.g. line 35). How are the downsides of the design decisions weighted?
    - Have taken parts of existing text and expanded on design compromises in new subsection to discussion 5.1, see (Ab line 538)
3. The graphs in Figure 5 are very difficult to parse, because colors are hard to distinguish and because different hydrodynamic models are combined into single panels. I would strongly recommend making this key figure easier to understand.
    - Have changed the colours to make them more distinct and adjusted the legends
4. The writing is rather sloppy, with many type-os (e.g. line 19, 160, 174, 175, 178, 246, 413, 454, 463, 503 and probably other locations) and mistakes in Figure references (line 183).
    - Had a professional proofread and fixed Fig ref.

Minor comments

1. Line 1: The authors could be clearer about their versioning. The version number (0.5) only appears in the title, and is not mentioned anywhere else (e.g. line 83). Furthermore, it would be good to briefly mention what extra steps need to be taken to get to v1.0.
    - Tidied up intro to make clear what aspects are new/added in version 0.5 over ver. 0.1 (see multiple blue text segments labelled (A1) in introduction, reviewer B had a similar comment)
    - Added improvements required to become version 1 in summary at (A1 line 586)

2. Line 9: mention that these text files are yaml or json
    - Done see, (A2) in abstract
3. Line 13 and at other locations: Parcels has had a recent rebranding, and has dropped the "Ocean" from its name. Apply that here too?
    - Done see multiple (A3)'s
4. Line 13: why is this "despite"? briefly explain?
    - Added explanation: " which might be expected to be computationally slower than structured grids due to the time required to locate each particle within a hydrodynamic grid cell..", (A4 line 13)
5. Line 26: this can also be reversed: users run as many particles as they can within a user-acceptable time frame
    - Add sentence "This first challenge may result in users choosing to simulating smaller numbers of particles, in order to get results within project deadlines on accessible hardware" . (A5 line 27)
6. Line 48: after a certain PLD?
    - Added note that settlement "during the time settlment period, i.e. their pelagic larval duration, \cite{demmer2022larval}}", (A6 line 47)
7. Line 63: The new Parcels v4 (now in alpha release) is also compatible with structured and unstructured grids
    - Added Parcels 4 alpha to list , (A7 line 65)
8. Line 64: what does it mean to "identify useful optional variables"? I don't understand this sentence

- Rewritten, to be about useful hindcast fields, “if useful fields are in the hindcast, such as bottom stress used in particle re-suspension “ (A8 line 65

9. Lines 70-83: In each of these examples, also highlight the grid information (number of nodes) that the simulations were run on?

    - Added node numbers (A9 line 80ff)

10. The caption of Figure 1 misses some information. For example, what are the crosses on the top-left of panel a? Which region of the ocean are we looking at? And where is the scale-bar for panels c and d?

    - Now noted in caption, crosses in upper left are locations of a grid release, while those mid-plots are for two point releases, location noted as central NZ within a wider Cook Strait see (A10s line 110ff)

    - Scale bar now on all figures

11. The heatmap in Figure 1d is on a different grid as the hydrodynamic model. How is it specified? Can the user do that?

    - Clarified heat maps are for a user specified regular grid overlaid on hydrodynamic model, as shown in code appendix, “The heat maps in the lower plots are based on a user specified regular grid with a 100m resolution. “ (A11 line 110ff)

12. What does “read” in table 1 mean?

    - Clarified as read the hindcast in caption, “Read times are the time spent reading the hindcast from the local disk, with the balance of time spent computing particle tracks. “ (A12 line 90ff)

13. Line 128: another advantage of offline simulation is the ability to run backwards in time

    - Have added backtracking to features (A13 line 168) “{\bf Backtracking:} \OT supports reverse-time simulations, which can be useful for identifying potential sources of particles arriving at a given location \citep{thygesen2011reverse}. Note that dispersion is not time-reversible, and this operates the same in the forward-direction time, producing outputs, like heat maps, that offer a probabilistic view of sources \citep{thygesen2011reverse}”.

14. Line 133: How can diffusivity be both vertical and 3D? should it not be either of these?

- Dropped unnecessary 3D reference and rewrote paragraph, inline with similar comment by reviewer B (B6 line 151)

15. Line 137: so diffusivity is modelled as a Markov(1)-process? Perhaps good to mention that explicitly

    - Noted as a first order Markov process (A15 line 151)

16. Line 159: I don't think that these interpolated velocities are "Lagrangian" in the kinematic sense of the word; I would advice to remove the term

    - Oops!, they are not Lagrangian, but Eulerian, edited (A16 line 182)
      "At a high level, particle tracking code takes the Eulerian water velocity field from a hydrodynamic model and interpolates these to provide velocities at particle locations. These velocities are then numerically integrated to give particle trajectories"

17. Line 248: what does "bookkeeping information" mean? Do the authors mean "Boolean"?

    - Drop first reference to bookkeeping , and expanded explanation of bookkeeping part. props, at (A17 line 278).
      "bookkeeping particle properties, such as particleID, release groupID, or particle status, e.g. whether computationally alive, on the sea bed, stranded by the tide etc."

18. Line 296: specify how the quad cells are divided into triangles. And discuss what the impact is on memory/IO/performance?

    - Explained quad cell (A18 line 328) "The unstructured SCHISM and DEFLT3D-FM grids contain a mixture of triangular and quad cells. For these formats and regular grid formats, quad cells are divided into triangles. This division simply takes the first three nodes of a quad cell as one triangle, then creates a new triangle from the third, fourth and first nodes. Modifying the ``triangulation'' of the hydrodynamic model grid by splitting quad cells is only done once at the start of the run and it has no impact on the time spent reading the hindcast. It also adds little to the memory used, as the underlying field values are still stored as nodal values. By increasing the number of cells to search, quad cell splitting may sightly increase the time taken to locate the cell containing each particle. However, it simplifies the code by only requiring one type of cell to search."

19. Line 370-378: I wonder whether the authors have also considered the precision of particle properties. Are these stored as float64 or float32? Does that matter for performance?

    - Yes, all hydrodynamic model files are stores as float32 , speed advantage noted at (A19 line 439).

20. Line 414: As far as I know, GLORYS is not the hydrodynamics (that is NEMO), but the data assimilation.

    - Yes, we found the distinction confusing and got backwards , many people get GLORYS data, while fewer know it derives from NEMO hydrodynamics. Have elected to just use both each time and not explicitly distinguish which is which. (A20 line 472)

Review #2

We appreciate the reviewer's careful and thoughtful comments and have made extensive edits to address these, which are tagged and marked in blue within the revised text, the attached gives detail responses

The manuscript presents the new capabilities of the particle-tracking model OceanTracker and compares its performance with two other models, OpenDrift and Parcels. It also explains the design choices behind the model's structure and features and illustrates these through examples.

General comments:

a. The introduction should more clearly and concisely introduce the previous version of OceanTracker (including the hydrodynamic models it supported, as stated in the abstract) and explicitly state that this manuscript presents and evaluates the new version of the model.
    - Added text to intro which specially notes the aspects new in this version of the code as also suggested by reviewer A, see multiple A1 in the introduction.

b. The manuscript would benefit from additional references throughout. A few specific places where references are needed are noted below, but this is a broader recommendation for the entire text.
    - Have added references (A15 line 151, B6 line 151, B7 line 160, A10 line 355, A15 line 15) plus others
c. A thorough proofread is recommended, as there are numerous typos. I have listed the ones I noticed below, but I might have missed some.
    - Have taken the suggestion of a profession proofread.

Specific comments:

1. *line 62: Although fewer particle tracking tools exist for structured grids, a few are available. For example PyLag (https://pylag.readthedocs.io/en/latest/)*
    - Added reference to Pylag (B1 line 70)
2. Figure 1: Please add color bars for panels 1b, 1c, and 1d. Also specify the time elapsed before the snapshots were taken. Clarify the meaning of the crosses shown on the maps.

- Have added colour bars and explicitly indicated crosses as a grid release in caption (B2 line 51)
  "In the upper figures, the crosses in the upper left are the locations of a grid particle release, while the black polygon is the area covered by a polygon release. The crosses near the middle of the plots are locations of two point releases."

3. *section 1.1: It would help to include a brief description of the particle tracking setup used in Figure 1 (e.g., study area location, simulation duration).*
   - Add release rates and numbers to subsection, see multiple in Fig. caption and text body (B3's line 110ff)
4. *line 96-97: Clarify whether this value refers to the average number of particles per cell over the full simulation period.*
   - Add comment on the average of 300 particles per cell out of 5 million total (B4 line 119)
     "The two points in the lower plots of \fref{fig:examples} each released 5000 particles every 10 min., giving 5.7 million particles by the end of the 4 day run. This resulted in an average of 500 particles being counted within each grid cell to create the heat maps"
5. *line 100: Rewrite as "some of **the new** Ocean Tracker **0.5** features" to explicitly refer to the new version.*
   - Edited as suggested (B5 line 125)
     "This section highlights some \edit{B5}{ of the new features of \OT v0.5 } that minimise user effort and its physics."
6. *line 132-133: specify "**vertical** dispersion" and remove "3D" in "vertical turbulent viscosity". Also clarify whether vertical dispersion is an addition to horizontal dispersion already available in earlier versions of OceanTracker.*
   - have clarified that using vertical turbulence profiles for random walk is new to this version, plus added references (B6 line 151ff)
     "As in the earlier version of \OT, random walk can be incorporated using constant turbulent eddy viscosities in both the horizontal and vertical directions. Optionally in the new version,} if vertical turbulent viscosity profiles are available in a 3D hydrodynamic model, the profiles are interpolated to calculate the size of the vertical random walk. This requires the inclusion of an additional vertical velocity that is equal to the vertical gradient of the turbulent viscosity \edit{B6){ \citep{visser1998using,thorpe2005turbulent}}."

7. line 140-142: Add appropriate references.
   - cited equations in lynch, B7 line 160
     "cite{lynch2014particles} equations 5.5 and 5.6."

8. *line 154: Correct the phrase "adding addition."*
    - Edited "adding additional", (B8 line 177)
9. *line 177: Correct "example"*
    - Edited at B9
10. paragraph lines 180-189: Incorrect figure is cited; please revise.
    - Changed at multiple (B10 line 204ff)
11. line 188: Clarify how users are expected to know which classes must be explicitly provided.
    - Add comment on minimum requirement of a reader and release group, changes at multiple (B11 line 211)
      "To run \OT a user must add at least a reader class and one release group class. In addition, they must set the name of the folder where output will be written to (see \frefs{fig:code-param-example}{a})."

12. Figure 3: Figure 3: The separation of dispersion and RK advection appears inconsistent with lines 136–137, which state that the random walk is applied to each RK sub-step rather than as a separate displacement.
    - Yes this does appear confusing, the size of the random walk as a velocity is only calculated once per full step as in figure, but this additional velocity is applied at each sub-step to move particles, clarified and modified figure to indicate computation of total velocity modifiers. (B12 line 151)
      "This dispersion velocity modifier is calculated once for each full time step and added to the hydrodynamic model's velocity at each Runge-Kutta (RK) sub-step, see \fref{fig:CP} and \sref{sec:vel_mods}."
13. *Caption of Figure 4: YAML configuration files are not mentioned anywhere in the text; a brief explanation should be added.*
    - YAML format now referenced at (B13 line 58) and abstract A2
14. Minor edits fixed
    - Line 207: Correct "their a new location"
    - Line 219: Fix spacing.
    - Line 239: Correct "manages they appropriately."
    - Line 246: Correct "can different types."
    - Line 248: Fix punctuation.
    - Changed order as suggested, - *Line 250-260: Consider reversing the order of the "custom particle properties" and "core particle properties" paragraphs. The current order feels confusing because the text returns to custom properties afterward.*

15. Line 288: Explain why 24 hydrodynamic model time steps were chosen.

- Have added comment noting why 24 (B16 line 321)
  "This default number of steps balances reducing wall time, by reading in multiple time steps at the same time, while only requiring of 100s of megabytes of computer RAM for each buffered hindcast variable. The user can reduce this buffer size if RAM is limited."

16. Line 294-295: This information should be introduced earlier in the introduction.
    - Moved to introduction as suggested
17. Line 303-304: The sentence is unclear; please rewrite and add a reference describing the method.
    - Rewritten and expanded at (B18 line 342)
      "Vertical interpolation to the position of each particle is also linear between the layers of the hydrodynamic model and between time steps. The one exception to linear vertical interpolation is the water velocity within the seabed layer. Here, vertical interpolation is based on a logarithmic layer, ensuring that particles near the seabed experience a more realistic velocity profile than the linear variation in higher layers \cite{lynch2014particles"

18. Line 378-385: Is there no downside to this method?
    - Noted that it means OT is not using the native hydro model grid, but an approximation of it (B19 line 422)
      "Though vertical interpolation means particle motions are not strictly a result of velocities of the native hydrodynamic model grid, but rather an approximation of those velocities."
19. Minor edits done to fix
    - Line 402: Fix spacing.
    - Line 403: Fix spacing.
    - Line 454: Fix spacing.
    - Line 497: this is a conclusion of the paper so may not have been noted in a prior publication? Added a reference (B20 line 586)

Reviewer 3#

We appreciate the reviewer's careful and thoughtful comments and have made extensive edits to address these, which are tagged and marked in blue within the revised text, the attached gives detail responses

The manuscript _OceanTracker 0.5: Fast Adaptable Lagrangian Particle Tracking in Structured and Unstructured Grids_ by _Ross Vennell et al._ describes the software OceanTracker 0.5. I don't consider this paper publishable in its current form. Based on the scientific and formal quality of the manuscript alone, I'd tend to recommend a rejection. But the design choices made I, with only cursor n the software are novel and the reported (though partially only claimed) performance and versatility of the tool are impressive. Hence I recommend _major_ revisions.

I include an annotated copy of the manuscript with detailed comments. Please refer to these comments for detailed and granular remarks and suggestions. Throughout the annotation, I tried adhering to the following color scheme: Green is for positive aspects. Blue highlights wording, grammar, or typographic issues. Orange requests clarification of numerical, physical or software engineering details which I feel are necessary to support otherwise poorly substantiated and often unclear claims of greatness of the described tool which I highlighted in red. Many of the highlighted lines have notes attached to them.

a. Its clear we did not manage expectations by making the intended audience the paper clearer, this is done now (Ca line 94) in the introduction. The reviewer seems to have been expecting a paper outlining the technical performance developments in a manner that would enable a particle tracker developer to implement them. The previous paper outlines the main performance techniques. This paper is user focused, showcasing the features that novice though expert users may find useful. There are only high-level comments on how some aspects were implemented within the code and its overall structure which may be helpful for intermediate users. The details of how the structure is implemented would be a subsequent paper.
b. The aim of showcasing features for prospective users also dictates a different style of presentation. Thus, the introduction does have some as yet unsupported claims. Also, some paragraphs may start with an assertion to engage readers/users, followed what it means, rather than the more standard evidence then conclusion style of a more standard scientific paper. Some claims are likely

subjective, e.g. acceptable time frames, adaptability, which aspects OT addresses of this is now explicitly noted (C10.1 line 47), with follow ups at (C10.2 line 53), (C10.3 line 56)

Positive aspects:

- The argument for using uniform sigma grids (around L380) is properly substantiated: It identifies a problem, and then describes the solution implemented in OceanTracker in a way that a reader from a background similar to the authors could directly implement themselves.

- The paragraphs around L470 has an almost perfect structure: Clearly identify a challenge, describe a range of solutions, indicate which solution was picked, and be transparent about tradeoffs and downsides.

Major concerns: *[have split and reordered comments]*

1. Form and language: The manuscript appears to be sloppily prepared. There's misalignment and artefacts in schematics, there's informal and un-precise language, there's typos and typographical issues all over the text. Especially the first part of the manuscript is full of rather un-specific and bold claims of greatness which often remain completely obscure.

    - See above, plus a have done a professional proofread and changes based on marked up text with responses.

2. The notion of "speed" and "acceptable time" from a user perspective are not defined. And "millions of particles" is used as a placeholder for a large and challenging task without any relation to requirements like statistical significance of derived statistics, variability of the underlying currents, experiment time frames etc.
    - Yes, we were not clear in the use of "speed", have qualified it as computational speed or wall time many places.
    - Acceptable time is entirely subjective depending on a user's goals, resources and project deadlines so difficult to define. "Large numbers" is also subjective for the same reasons, though here we implicitly start on the basis on millions being large. E.g. (A5 line 27) makes this point explicitly.
    - Also, we did conflate these two aspects of speed in places. Eg specific that we are talking about computational speed e.g. (C3.2 line 409), now explicitly wall time.

3. Most of the major technical achievements which would warrant the publication of this manuscript in the first place are only hinted at.

    - Most the techniques giving computational speed are covered by the previous paper, this paper only provides a glimpse of internal structure implementation, the details of how it was implemented would distract the intended audience of users but may warrant another paper.

4. And internal design decisions such as what exactly the data structures holding particle properties look like are described in different and contradicting ways.
    - After looking through the inline comments, it is not clear which are the different and contradictory ways. Have made changes based on marked up version which hopefully address this

5. However, adaptability of software is usually not hampered by lack of modularity (while modularity can be _one_ strategy to achieve better adaptability) but by hidden complexity, leaky abstractions etc. How do the authors ensure their modular structure actually _is_ easy to reconfigure or adapt?

    - Yes, while modularity may make some aspects easier for user to access, it may make others hard or even impossible. This critique is noted within a new section suggested by another reviewer on design compromises (C6 line 559). The ease of adaptation is dealt with below in response #10

6. As a consequence, even a reader familiar with the challenges of developing similar scientific software, is left with hardly an idea about how (and why?) specific design decisions were made, let alone being enabled to validate or falsify specific claims or learn from what's presented for similar software
    - As noted above the how and why is deliberately high level in this paper with a user target audience. The overarching why is hopefully clearer in the revision and response to marked up comments.

7. Before it's even clear what OceanTracker does exactly aim to do, its speed is pointed out.

    - The paper attempts to engage readers/users by describing two obvious challenges in particle tracking and how the new OT addresses them. Though we erred by not describing OT as particle tracking code when it was first referenced in the introduction, this is now done (C8 line 35).

8. Later, it claims to achieve "full configurability and adaptability" by users not knowing how to code only to state in the very next sentence that there's a coding-based mechanism supporting more complex simulations. But how can both be true?
    - This was poorly worded, as noted by other reviewers. Have rewritten at (C24 line 204), (C25 line 205), they are both true only when using built in components.

9. Many of the paragraphs use a cyclic argument. For example, the claim that modularity leads to adaptability is made in L42, followed by details of the modular structure, which in turn are followed by a rephrased repetition of the initial claim. However, adaptability of software is usually not hampered by lack of modularity (while modularity can be _one_ strategy to achieve better adaptability) but by hidden complexity, leaky abstractions etc. How do the authors ensure their modular structure actually _is_ easy to reconfigure or adapt?
    - The introduction section was a bit vague on what adaptability is. Firstly, for most users this is easily incorporating the particle behaviours they need from a suite of supplied options using a configuration file. Secondly for coding users, being able to easily extend and adapt the given options to their specialised needs. Have been explicit on what aspects of adaptability we try to address (C10.1 line 47), (C10.2 line 53), (C10.3 line 56).

10. multi-threading and multi-processing are conflated.
    - Yes, these were conflated in places, have modified text at multiple (C43 line 461ff) to better differential between threading and multiprocessing use on multiple cores.

11. The performance analyses lack specifics which in my experience are crucial for understanding performance of scientific software especially on HPC systems which is where any serious general circulation model output would be available and where substantial Lagrangian simulations would hence be performed.

    - Agreed it is useful to have specifics of HPC systems to assess performance. However, the main target for OT is the large number of users who only have access to more basic laptop/desktop hardware, and don't have the option of accessing hundreds of cores on HPC systems. Thus, the table is intended as a rough guide on relative performance on different hardware. Have made it explicit that desktop/laptop users are a prime focus for OT (C12.1 line 35) . The speed comparison between OT and other

particle tracking codes is only intended to compare relative speeds on the same hardware, to give a sense of where OT's performance lies. Have explicitly noted relative performance may differ on other systems (C12.2 line 487).

Other responses are as reply to comments is the supplied marked up pdf.