Response to Reviewer #1 for Manuscript "High-Fidelity Modeling of Turbulent Mixing and Basal Melting in Seawater Intrusion Under Grounded Ice" by Mamer, Robel, Lai, Wilson, and Washam

General comments:
The authors aim to tackle a fascinating problem of practical importance. Melting near grounding lines is thought to have much more impact on glacier dynamics than melting anywhere else, yet observations are limited, and high-fidelity simulations are lacking. Thus, the work is highly novel and has significant potential for improving our understanding of ice-ocean interactions that have a high impact on climate dynamics. However, I am not sure that the ANSYS Fluent RANS solver used is appropriate for this problem. In fact, I have never seen the ANSYS Fluent RANS solver applied to environmental flows as complex as in this work. This does not mean that it cannot work, but that significant effort should be devoted to validating the code. As the governing equations solved by the codes are not clearly presented or discussed (in particular, the boundary conditions, which are yet key to evaluating the melting dynamics), I have found it difficult to assess the appropriateness of the mathematical/numerical model. I am particularly concerned with the modelling of the ice-ocean boundary: there are no salt constraints and melt is said to be activated in a way that I did not find correctly physically motivated. In the unfortunate case that the code cannot in fact solve the exact problem at hand (with temperature and salt stratification coupled through a phase change boundary) I would suggest that the authors reformulate the problem as a list of hypotheses--inspired from the full problem--and testable with simulations of a modified/simpler model (but mathematically transparant), which could be the reduced thermal driving model discussed below (point 2).

We thank the reviewer for their thorough comments on this manuscript. We have identified this reviewer's primary concerns to be:

1. The lack of model description and presentation of main equations used by ANSYS Fluent. Along with this, the reviewer also identified the lack of citations supporting the choice of ANSYS Fluent (e.g. validation studies).
2. The lack of a salinity and pressure-dependent thermal boundary condition for the ice boundary.
3. Overall issues with clarity in describing model equations and model initialization.

To address these concerns, we have revised our appendix to present the full model framework, including governing equations, domain and meshing, and validation studies. In addition, we will re-run each of our simulations with a salinity and pressure-dependent ice thermal boundary condition.

We address each comment below. The reviewer's comments are written in black, our responses are given in red, and the original manuscript text is in grey.

First Comment:

The governing equations that the code solves should be clearly presented, as I do not think that TC readers are familiar with RANS models (and because I have never seen RANS models used in this context). This requires presenting the Reynolds decomposition of the flow (which would help you justify the fact that 2D dynamics is expected since turbulence is not resolved) and the governing equations for the ensemble-average variables. The closure for the Reynolds stresses and turbulent fluxes should then be discussed in greater details than in the current manuscript. The kappa-eps scheme is mentioned (Eq. (2)) but important details are lacking: for instance, how are kappa and epsilon related to the resolved variables?

We have added a subsection to the appendix describing the governing equations of the model and further discussed the turbulence closure scheme. We have revised appendix section A4 to have these subsections:

A 4.1 Reynolds Averaged Navier Stokes Equations

"ANSYS Fluent solves the Reynolds Averaged Navier Stokes (RANS) equations. This formulation of the Navier Stokes equations decomposes the flow into a mean state and fluctuation about the mean:

$$u_i = \overline{u_i} + u_i' \qquad \text{(R1)}$$

where $\overline{u_i}$ is the time-avearged velocity and u_i' is perturbations about the mean velocity. Scalar quantities are also decomposed into their mean and fluctuations about the mean. Substituting these decomposed values into the momentum equations yields:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i}(\rho u_i) = 0 \qquad \text{(R2)}$$

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}(\rho u_i u_j) = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right] + \frac{\partial}{\partial x_j}\left(-\rho\overline{u'_i u'_j}\right) - \frac{\Delta \rho g}{\rho \epsilon^3} \qquad \text{(R3)}$$

Equations R2 and R3 are the RANS equations and have the same general form as the Navier-Stokes equations, however, the velocities and solution variables now represent the time-averaged values. In addition, new terms have been added to incorporate the effect of turbulence. These terms are the Reynolds stresses $(\rho\overline{u'_i u'_j})$. To close this system of equations, the Reynolds stresses must be solved via a turbulence closure. More details on turbulence modeling and our closure choice are given in section A 4.2."

A 4.2 Turbulence Modeling

Edits are given in minor comment 6.

A 4.3 Species Transport Modeling

Edits are given in major comment 2 and minor comment 17

A 4.4 Post-Processing

Edits are given in major comment 6

I would like also to see more details on the so-called damping functions of the low-Re formulation that supposedly enable accurate diffusive boundary layer representation. Is it like in a wall-resolved large-eddy simulation model?

The standard k-ϵ turbulence model relates eddy viscosity to turbulent kinetic energy and turbulent dissipation to close the system of equations. This requires solving the transport equations for turbulent kinetic energy and dissipation. Turbulent kinetic energy vanishes near the wall, producing a singularity in these transport equations. To fix this, a modification must be made to the timescale set by the ratio of turbulent kinetic energy and dissipation. This is where low-Reynolds formulations of the k-ϵ model are helpful, which resolves the transport of turbulent kinetic energy and turbulent dissipation in the low-Reynolds number regions of flow (i.e near no-slip boundaries).

In large eddy simulations (LES), only large eddies of a system are resolved, with small-scale (typically sub-grid) eddies being filtered out. For wall-resolved large-eddy simulations, the boundary layer is directly resolved by having a very fine mesh to resolve the large gradients near the wall. The damping functions and adjustments made to the k-ϵ model in order to resolve the boundary layer are similar to wall-resolved LES since they both require a finer mesh and do not rely on wall functions to parameterize the near-boundary gradients. However, because of the averaging done to obtain the RANS equations, a singularity is introduced in the turbulent dissipation equation via the turbulence timescale. The low-Reynolds formulation of the k-ϵ model fixes this (as explained and edited in minor comment 6) and uses the damping function to ensure a smooth transition between the freestream flow and the boundary flow.

We have added a description of one version of a low-Reynolds formulation to the appendix (minor comment 6). Other formulations of low-Reynolds models are also available within ANSYS Fluent, and a sensitivity analysis will be done to determine the most appropriate version. Within the appendix, we will include an explanation of the exact version we use, alongside a description of how it works.

Second Comment:

The decoupling of the salt dynamics from the ice-ocean boundary dynamics is not justified and a priori seems wrong. I assume that this decoupling is due to code limitations. However, if you cannot justify the decoupling, I am afraid this just means that the code is not suited for environmental flows with temperature and salt stratification and melting. In the worst case scenario you might consider reformulating the problem in terms of a single scalar variable, namely thermal driving (ref: Adrian Jenkins' papers and other people's related works). The collapse of the full dynamics onto a reduced thermal driving model is thought to be accurate in highly-turbulent environments (for which kappa-epsilon applies anyway) and small salinity variations. This latter condition is obviously problematic (which should be discussed) with regards to your problem of interest. However, I would rather see thermal driving model simulations transparently solved by ANSYS than results from a non-transparant full temperature-salinity model.

In the original manuscript, the only way salt is not considered is within the melting dynamics. We did not model phase change (regardless of whether this is melting or dissolution) in the simulations. Dissolution-driven melting (where salt would get into the crystal matrix, suppress the freezing point to below the ice temperature, and therefore cause it to melt) would only dominate ice loss if the seawater is subfreezing (below the local pressure and salinity dependent freezing point) and therefore mass transfer into the ice matrix would exceed heat transfer.

In the original simulations, we set the boundary temperature to $0^{\circ}C$. However, based on suggestions from both reviewers, we will re-configure the simulations to have a pressure and salinity dependent thermal boundary condition for the ice boundaries using equation R5. In Figure R1, we demonstrate what this thermal boundary condition would look like for a free-slip case in a simplified pipe flow example.

We have added to line 573:

> "Salt is transported as an active tracer within the fluid, employing an advection-diffusion-reaction equation:
>
> $$\frac{\partial}{\partial t}(\rho S) + \nabla \cdot (\rho \vec{u} S) = -\nabla \cdot \vec{J} + R_i \qquad \text{(R4)}$$
>
> Where $\vec{u}$ is the velocity vector, $\vec{J}$ is diffusion flux, and R_i is the production rate from reactions. The production rate for salt in these simulations is zero."

Further edits regarding salinity sinks and dissolution-driven melting are given in major comment 3.

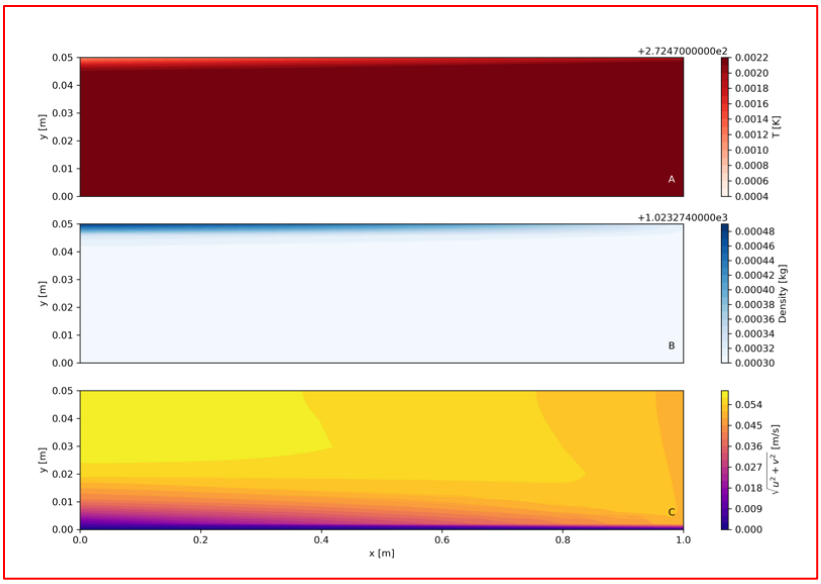

Figure R1

Example run of a 2D pipe-like geometry with ice at y = 0.05 m and a velocity inlet prescribed at x = 1m. This case has a free-slip condition at the ice base (along y =0.05) as well as a salinity and pressure dependent thermal boundary condition (here, we set the ice thickness to be 1000 m). Panel A is temperature, panel B is density, and panel C is velocity magnitude. In panel C, a classical boundary layer forms along the flow direction at the bottom boundary where there is a no-slip condition. In panel A and B, a thin thermal boundary layer forms due to the salinity and pressure dependent thermal boundary condition cooling the near-ice water.

Third Comment

The boundary conditions, especially at the ice-ocean interface, should be clearly presented and discussed.

We have elaborated on the boundary conditions (thermal and kinematic) in both the main text and the appendix where the model formulation is discussed.

We specifically edited lines 87 - 88:

> "The ice wall boundaries have a pressure and salinity-dependent thermal boundary condition of:
>
> $$T_b = S_b \lambda_1 + \lambda_2 + z \lambda_3 \quad \text{(R5)}$$
>
> Here, λ_1, λ_2, and λ_3 are constants, and the boundary salinity is S_b. The depth of the ice is equal to z_b, in these simulations we set this to be 1000 m. Both the vertical

and horizontal ice boundaries have a no-slip kinematic condition in the non-melting cases, forcing the freestream fluid velocity to be zero at the ice wall."

In addition, we have added to line 147:

"Therefore, the melting cases have a free slip kinematic boundary condition. The temperature of the inflowing meltwater is set by equation R5."

Where equation R5 is the pressure and salinity-dependent freezing point equation.

We have also edited lines 519 - 522 in the appendix:

"For both melting and non-melting simulations, the ice walls have a thermal boundary condition dependent on near-wall salinity and pressure (eq. R5). Neither ice wall boundary allows for salinity diffusion, which would be another mechanism of melt to account for. The non-melting simulations employ a no-slip kinematic condition, which forces the fluid velocity to be zero at the wall. For melt-enabled cases, the top boundary of the subglacial space is turned into a velocity inlet to simulate melt. In designating this boundary as a velocity inlet, a free-slip kinematic condition is required. The downward vertical velocity set at the melting ice boundary follows equation 4. We turn off melting for the vertical ice boundary to isolate the intrusion-induced melt from the vertical plume dynamics that would arise from the vertical ice boundary."

The physical motivation for the so-called melt-activated formulation is lacking. Melting produces buoyant flows even when the boundary is no slip, simply because melting acts like a sink for salinity (though this is lacking in your model).

Since salt transport is included in the model (i.e. salt is transported as an active tracer) and we are not considering dissolution-driven melting, there does not need to be a sink for salinity in the melting framework. The model solves for the displacement and movement of 'saltier' waters via the salt transport equation (equation R4). Mass is conserved via a pressure outlet (zero gradient flux boundary condition) at the top of the ocean domain.

We have edited lines 104 - 105 to improve clarity:

"Salt is therefore transported as an active tracer within the fluid domain. To ensure mass transport within the computational domain is realistic and physical, there are two velocity (hence mass) inlets (i.e the subglacial discharge and ocean inflow) in the non-melting case and three velocity inlets (subglacial discharge, the melting horizontal ice face, and the ocean inflow) in the melting case. A pressure outlet is defined at the upper portion of the ocean domain, meaning the mass outflow rate along this boundary is not specified and is determined as part of the numerical

solution based on the requirement that all flow variables have zero gradients in the direction normal to the boundary. This kind of arrangement is typically used to emulate fluid flows in an infinite domain as in our case where subglacial channel discharge is released at the ground line into the ocean with infinite extent. ”

Thus, should we envision your melt-activated formulation like a velocity compensation for the lack of salt sink in the model? If so, it was not clear to me whether the velocity is prescribed vertically or horizontally, and I am not sure it is the correct way to compensate the salt sink.

Since we are neglecting melting from dissolution, due to above-freezing fluid conditions, a salt-sink at this melting boundary is not needed. These simulations strictly focus on the heat-driven melting and associated buoyancy effects from added fresh water to the model domain.

The meltwater velocity is the rate at which buoyant freshwater is input into the domain to mimic heat-driven melting. The velocity is prescribed normal to the boundary where it is sourced, in this case vertical.

We have edited lines 143 - 144:

> “In some simulations, we also simulate the added buoyancy flux resulting from the heat-limited melting scenario. Here, we neglect melting driven by dissolution, instead focusing on melting driven by thermal equilibrium at the ice boundary. Since the thermohaline conditions of the fluid domain are non-sub-freezing, the neglect of dissolution-induced melting is justified.”

We have also edited line 150:

> “The downward fluid velocity prescribed at the horizontal ice face is set by the melt rate…”

Estimating the velocity to enforce at the boundary to mimick the salinity sink from Eq. (4) is also not justified, i.e. why should the movement of the interface (assuming there is no immediate hydrostatic equilibrium of the ice shelf at such small scales) be directly used as a buoyancy-driven velocity input?

We acknowledge that adding a vertical velocity to mimic the vertical movement of a solid is a bit crude, however, because this vertical velocity (the melt rate) is small relative to the main flow, its direct contribution to the momentum flux is small or negligible. The buoyancy it brings to the fluid domain is significant, however, and is therefore an integral part of the heat-driven melting process. In addition, the vertical velocity that injects meltwater into the domain is not constant across the horizontal ice face but rather varies according to the underlying thermal equilibrium prescribed by equation 4.

We have added to line 160:

> "In setting a vertical velocity to mimic a moving interface, we introduce additional sources of momentum to the fluid. The vertical velocity arising from the meltwater inlet is small relative to the main flow and is therefore negligible, but the buoyancy that the meltwater brings into the fluid domain is an integral part of the heat-driven melting process."

We have also edited lines 154 - 156:

> "This framework represents the conservation of heat at the ice-ocean interface, which varies along the horizontal ice face as the near-wall thermal gradient changes due to seawater intrusion and vertical mixing."

Fourth Comment

In would like to see a validation of the code, which includes the choice of turbulence closure. A simple benchmark case should be set-up, for which turbulence-resolving simulation data exist (either from direct numerical simulation (DNS) or large-eddy simulation (LES)). Several groups (with Catherine Vreugdenhil, John Taylor, Ken Zhao etc) have published such DNS and/or LES data over the past 5 years or so, making it practical. Typical configurations are channel flow configurations, which should be accessible to ANSYS. Validation could be based on quantitative comparisons of mean variable profiles (e.g. temperature, TKE) normal to the ice-ocean interface.

ANSYS Fluent was chosen to address this research question because of its extensive validation and history of practical use within the engineering fluid mechanics community.

We have added to line 63:

> "ANSYS Fluent has been extensively validated across diverse flow geometries and conditions. Zangiabadi et al. (2015) validated ANSYS Fluent's suitability in simulating flow structures around realistic bathymetric highs in a study evaluating coastal tidal turbine deployment. Here, they found the RANS k-ϵ method to be more precise than the large eddy simulation (Zangiabadi et al., 2015). Chan et al. (2020) demonstrated good agreement between ANSYS Fluent's simulation of multi-phase sediment-laden plumes and prior experiments by Hall et al. (2010) and Virdung and Ramuson (2007), using the k-ϵ turbulence closure. Similarly, Sultan et al. (2018) achieved results consistent with experimental data for both multi-phase and single-phase flow in pipe geometries. Al-Zubaidy and Hilo (2022) found strong agreement between their model of lateral intakes for engineered flow channels with field measurements."

We believe these existing studies validate a wide range of cases similar to our setup, and any additional validation would be outside the scope of this paper.

Fifth Comment

Successful applications of ANSYS Fluent RANS solver to environmental flow configurations with temperature and salinity stratification should be cited and discussed (in particular how they validated the code).

We appreciate the reviewer highlighting the lack of justification for choosing ANSYS Fluent within the paper. We have incorporated examples of validation and applications of ANSYS Fluent to stratified environmental flow scenarios.

We have added to lines 63 - 64:

> "Most relevant to this study, Chala et al. (2024) validated Ansys Fluent's capability to model seawater intrusion in porous aquifers, achieving close alignment between experimental data and model predictions for intrusion length and shape. Furthermore, ANSYS Fluent has recently been used to model freeze desalination processes, where the volume of fluid method is applied to a species mixture of salt and water with a cooling base. Jayakody et al. (2017) demonstrated ANSYS Fluent's applicability to conducting parametric studies for freeze desalination processes by validating it with experimental studies."

Sixth Comment

The discussion of the steady state and transient runs is really confusing. You should distinguish the existence (or non-existence) of a steady state from your strategy of successive runs to achieve it. The key point that should be in the main text is that the problem has a natural steady state (for the ensemble-average variables) as there is no external variability and the ensemble-average variables do not exhibit temporal fluctuations once equilibrated. The strategy to reach it should then be discussed in an appendix.

The purpose of the steady-state solver is to generate an initial condition that is reminiscent of our expected solution since Fluent's steady-state solver is sensitive to initial conditions. We then ran the transient solver because the initial steady-state solver did not reach the seawater's full distance of intrusion. At the end of the transient run, a final steady-state solver run was conducted with the final transient run as the initial condition. This final step was to ensure the transient intrusion distances were not a product of numerical noise. Figure A1 in the appendix compares the time-averaged intrusion distances from the transient runs to the final steady-state run.

We have edited lines 166-169 in the main body:

> "Each simulation is initialized with the steady-state solver employed by ANSYS Fluent. This step to initialize the domain is necessary because the model is sensitive to initial conditions and it cuts down on the total run time needed to achieve a quasi-steady-state. The transient solver is then run for 12 hrs at 5 s time steps. A quasi-steady-state is reached at 6000 s. All results presented are from the transient simulations and a comparison to the steady-state solver solutions is shown in section A4 (Figure A1). Note that the steady-state solver is different than the quasi-steady-state achieved in the transient runs. The steady-state solver employed by ANSYS Fluent drops the time derivative, while the quasi-steady-state achieved during the transient run is a stable intrusion state. A list of all simulations is presented in section A4 (Table A2)."

As well as lines 574 to 582 in the appendix:

> "The results presented in the main body of this work are the time-averaged results from the transient run of 12 hrs. A series of tests were conducted to evaluate ANSYS Fluent's steady-state solver to the transient solver and their corresponding solutions. ANSYS Fluent's steady state solver drops the time derivative from the RANS equations and is sensitive to initial conditions, which is different than achieving a quasi-steady-state. The domain is initialized with a saline, warm ocean tank, and a fresh, cold subglacial environment with no intrusion. We run the steady-state solver with this initialization (named steady state pre), and in most cases, a small intrusion develops and occasionally no intrusion. We then run the transient solver for 12 hrs at 5s time steps where a quasi-steady-state intrusion develops. A secondary run is conducted with the steady-state solver, using the quasi-steady-state transient solution as the initial condition (named steady-state post). In the post-steady-state run, the intrusion developed during the transient simulation persists. The comparison of results between the pre and post steady-state solver runs demonstrates the steady-state solver's sensitivity to initial conditions. We thus disregard the steady state pre-results. To compare the steady state post and transient solutions, we plot the intrusion distances against each other, in which they nearly collapse on a 1 to 1 line (Figure A1). This comparison reinforces the development of a quasi-steady-state in the 12 hr transient runs."

Seventh Comment

I have found many typos in the appendices ("Need to list reference values, materials info, the methods and controls" found line 566), suggesting that these were not carefully reviewed by the authors. At the moment the appendices seem primarily like a draft list of comments and figures that did not make it into the main text (some in fact repeating what is already in the main text).

A careful review of the appendix will be conducted. In addition, we have added further descriptions of the governing equations (major comment 1, minor comments 4, 5, 6 and 7),

model boundary conditions (major comment 3), the use of transient and steady-state solver (major comment 6), and analysis of turbulence parameters. We have moved sensitivity tests and extra simulations to the supplementary material to make the appendix more concise.

Eighth Comment

The figures are lisible but could be improved(e.g. the x-axis label of Fig. 6 is quite small).

We will standardize the figures' text sizes and shapes for all figure axes and titles.

Ninth Comment

Because of the many questions I had/have with respect to the simulation code, I was not able to appreciate the comparison of the simulation results with the parameterized predictions of melt rates. If the authors can validate their code, I agree that this comparison would be an important addition to the paper, but it would have to be a fair comparison. That is, if the authors end up solving a model that is distinct from the model that the parameterizations (necessarily approximate) aim to mimic (arguably the real exact model), they should discuss result differences in light of model differences.

We understand the concerns of the simulation code to be: salinity sinks, ice-ocean boundary temperature independence of salinity, and vertical velocity as a proxy for a moving ice interface.

We reiterate that salt transport is included in these simulations and the only way it is not represented in the melting framework is by having a suppressed boundary temperature.We can safely neglect dissolution-driven melting due to the above-freezing fluid conditions. Including the depressed freezing point would only increase our simulated melt rates, by increasing the thermal driving (i.e., the current simulated melt rates are a lower bound). Thus, because the parameterized melt rates are already lower than the simulated melt rates, including a boundary temperature dependent on near-ice salinity would further increase the disagreement between simulated melt rates and parameterized melt rates. Our discussion of the simulation-parameterization disagreement attempts to highlight the differences in light of the assumptions of the fluid-structure that are inherent in the parameterization. i.e. well-mixed, higher Reynolds number, etc.

However, considering comments from both reviewers, we will re-run each simulation with a thermal boundary condition that is dependent on near-wall salinity and a reference glaciostatic pressure (i.e. at 1000 m). A proof-of-concept model run is demonstrated in Figure R1 for a simplified pipe flow geometry.

Since salt transport is represented in the model by its advection-diffusion equation, a sink does not need to be provided where meltwater enters the domain. If we consider melting

from dissolution processes, the ice boundary would need to remove salt. However, because the fluid domain simulated here is non-sub-freezing, heat-driven melting will dominate and therefore the neglect of diffusion in the model presented here is justified.

We acknowledge inputting a vertical velocity as a proxy for a moving ice interface is crude, however, because this vertical velocity is O(-6) and the freestream flow is O(-1) – O(-3) we believe this choice is justified.

Tenth Comment

Simulation snapshots and movies would really help visualize the flow.

In the supplementary material, we will include simulation snapshots for the initial steady-state runs (i.e. the initial condition) and movies of the transient runs with updated simulations. These will ideally aid Figure 1 from the original manuscript in describing what the flow regime looks like.

Additional comments, including technical corrections: typing errors, etc.

1. line 29: could you clarify the idea of "tidally asymmetric"?
    - Added to line 29:

        "Such asymmetry results in stronger melting during the ascent of high tide and weaker melting during the tidal ebb."

2. line 87: writing "The ice wall boundaries have a temperature boundary condition of 0◦C" really felt like a bomb! And the lack of justification or description of the full mathematical model did not help disarm it. It is really not expected that prescribing 0◦C at the ice-water interface is reasonable, especially when the salinity goes from 0 to 30 psu.
    - Based on suggestions from both reviewers, we will re-run all simulations with the pressure and salinity dependent freezing point boundary condition. Figure 1 above demonstrates how this will be done.
    - We have edited the manuscript at lines 87-88, 147, and 519-522 as written in major comment 3.

3. line 108: this equation of state may be suited for small salinity variations, but with a range of 0 to 30 psu, it may not be appropriate, unless the grounding line is beneath a lot of ice. I would recommend that you use a higher-order approximation of the true equation of state, or at least acknowledge that you know of the anomaly of the equation of state at low salinity and pressure but that you discard its effects to simplify the problem. Ideally you could discuss/speculate on how the results might change should you consider an accurate equation of state.
    - We appreciate you bringing this concern to our attention. We will run a sensitivity test comparing our linear E.O.S. to the higher order approximation in Roquet et al. (2015):

$$\rho' = -\frac{C_b}{2}(\Theta - \Theta_0)^2 - T_h Z\Theta + b_0 S_A \qquad \text{(R6)}$$

Where:

$$C_b = 0.011\ kg/m^3K^2$$

$$T_h = 2.5\ x\ 10^{-5} kg/m^4K$$

$$b_0 = 0.77\ kg/m^3(kg/g)$$

$$\Theta_0 = -4.5^{\circ}C$$

The results of this sensitivity test will be discussed in the appendix.

4. line 113: conduction, diffusion, and molecular dissipation all sound like the same thing to me. Could you explain how they differ? (reading your appendices it looks like conduction might be turbulent conduction)
    - Conduction represents the transfer of heat due to the thermal gradient. Within ANSYS Fluent, the conductive value used is the 'effective conductivity' which is the summation of the thermal conductivity of the fluid and turbulent conductivity. The diffusive term represents heat transfer due to species diffusion – in this case, salt. In equation A7, this term is the sum of every species' diffusive flux multiplied by their enthalpy. Molecular dissipation is another way of phrasing viscous dissipation, which is the heat transfer due to viscous forces, represented by the effective shear stress and kinematic viscosity.
    - Edited line 113 :

        "Energy, and therefore fluid temperature, is evolved via an energy conservation equation employed by the CFD solver resolving advection, conduction, salt diffusion, and viscous dissipation ~~molecular dissipation~~. (ANSYS Inc., 2024). Conduction represents heat transfer due to thermal gradients, and viscous dissipation is the transformation of kinetic energy into thermal energy due to shear forces. As salt diffuses in the medium, it also transfers heat due to its unique thermal properties, and therefore must also be included."

5. line 124: the quadratic quantity indicated is one among many, such that the sentence does not read well. Please reformulate.
    - To clarify, we rewrote the quadratic quantity to have i,j notation.
    - Edited lines 123 - 127 are shown below.
6. line 126: could you recall what the Boussinesq hypothesis is?
    - Edited lines 123-127:

        "A closure scheme is necessary because averaging the RANS equations introduces Reynolds stresses due to turbulent motion within the fluid. Reynold's stresses take the form $\overline{u_i' u_j'}$, the averaged product of turbulent velocity fluctuations. One class of closure models employs the turbulent-viscosity hypothesis, which relates the deviatoric Reynolds Stresses to the mean strain rate via a positive scalar eddy viscosity (Pope, 2000). Here, we utilize the two-equation k-ϵ closure scheme, which solves for the eddy viscosity by:

$$\mu_T = \rho_w C_\mu \, k^2/\epsilon \qquad \text{(R7)}$$ ”

- Added to Section A4, starting at line 551:

  “The turbulent-viscosity hypothesis (also known as the Boussinesq Hypothesis) is used in turbulence modeling to solve for the Reynolds stresses. This hypothesis states the deviatoric Reynolds stresses (those deviating from the mean) are proportional to the mean strain rate tensor by a positive scalar. This scalar represents the eddy viscosity (also referred to as turbulent viscosity). This relationship is:

$$-\rho\overline{u_i' u_j'} + \frac{2}{3}\rho k \, \delta_{ij} = \rho\mu_T \left(\frac{\partial \overline{U_i}}{\partial x_j} + \frac{\partial \overline{U_j}}{\partial x_i}\right) \qquad \text{(R8)}$$

  The only unknown left in the system of equations is the eddy viscosity, which can be solved for by a variety of different turbulence closure schemes. Here, we employ the two-equation k-ϵ closure scheme which solves for eddy viscosity by relating it to the square of turbulent kinetic energy and inverse of turbulent dissipation by a positive scalar C_μ. This closure scheme requires two additional equations to solve for turbulent kinetic energy and turbulent dissipation. These equations are:

$$\frac{\partial}{\partial t}(\rho k) + \frac{\partial}{\partial x_i}(\rho k \, u_i) = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right] + G_k + G_b - \rho\epsilon - Y_M + S_k \qquad \text{(R9)}$$

  For turbulent kinetic energy and:

$$\frac{\partial}{\partial t}(\rho\epsilon) + \frac{\partial}{\partial x_i}(\rho\epsilon u_i) = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\frac{\partial \epsilon}{\partial x_j}\right] + C_{1\epsilon}\frac{\epsilon}{k}(G_k + C_{3\epsilon}G_b) - C_{2\epsilon}\rho\frac{\epsilon^2}{k} + S_\epsilon \qquad \text{(R10)}$$

For turbulent dissipation. Note, in the equation for turbulent dissipation, turbulent kinetic energy is in the denominator which results in issues when k approaches zero near wall boundaries. To resolve boundary layer dynamics with the k-ϵ closure, we use the low-Reynolds formulation which employs damping functions and fixes the singularity that arises with low values of k.

A version of the low-Reynolds k-ϵ closure we employ here is the Yang-Shih version (Yang and Shih 1993). In this formulation, the authors set the near-wall turbulence timescale to be the Kolmogorov timescale ($T_k \propto (\nu/\epsilon)^{1/2}$). In doing so, the equation for eddy viscosity near the wall becomes:

$$\mu_T = C_\mu f_\mu k \left(T_k + \frac{k}{\epsilon}\right) \quad \text{(R11)}$$

Where f_μ is the "damping function" and equal to:

$$f_\mu = \left[1 - exp\left(-a_1 R_y - a_3 R_y^3 - a_5 R_y^5\right)\right]^{1/2} \quad \text{(R12)}$$

and $R_y = \frac{k^{\frac{1}{2}} y}{\nu}$. The constants a_1, a_3, and a_5 are constrained from DNS experiments for turbulent channel flow.

The final adjustment to the standard k-ϵ formulation for near-wall flows is to add an additional source of dissipation, which results from inhomogeneity in the mean flow field. This takes the form:

$$E = \nu \mu_T \frac{\partial U_i}{\partial x_j \partial x_k} \frac{\partial U_i}{\partial x_j \partial x_k} \quad \text{(R13)}$$

This formulation of the low-Reynolds k-ϵ turbulence closure allows for solving the free-stream portion of the flow regime as well as the near-wall region where viscous effects dominate, since the added terms tend to zero when turbulence is high."

7. Eq. (2) P2: what are the equations for kappa and epsilon? I expect that they involve the resolved ensemble-average variables.
    - We have added these equations to the appendix in the model description as written above in comment 6.
8. Section 2.4: this section is really confusing as I already mentioned earlier, because buoyancy isn't due to some interfacial velocities but to changes in salinity and temperature at the phase-change boundary. Please reformulate.
    - We have emphasized that any buoyancy effects are exclusively due to the difference in density between the input meltwater and the ambient water, and not due to some prescribed velocity.
    - We have added to lines 160-161:

        "In setting a vertical velocity to mimic a moving interface, we introduce additional sources of momentum to the fluid. However, the vertical velocity arising from the meltwater inlet is small relative to the main flow and is therefore negligible. The input of fresh, cold water due to melting introduces a buoyancy flux to the domain, due to density differences between the meltwater and intrusion."
9. line 150: I don't see the physical justification for prescribing a "horizontal ice (?!) velocity" at the horizontal ice-water interface.
    - A fluid velocity-inlet boundary condition is applied to the horizontal ice base when melting is enabled. The fluid velocity prescribed here is normal to the boundary and therefore is downward, not horizontal.
    - Edited line 150:

        "The downward fluid velocity prescribed at the horizontal ice face is set by the melt rate, $\dot{m}$, and is a function of the difference between the near-wall cel''s centroid temperature T_w and the ice-ocean interfacial temperature T_b, thermal conductivity k_T, and density of ice ρ_i:..."
10. line 179: can you provide a reference for "realistic estuarine-like mixing rates"?
    - We appreciate the reviewer bringing the lack of citations to our attention. We have amended line 179 to:

        "Turbulent mixing, as modulated by C_μ, affects intrusion distance to a lesser degree than freshwater discharge velocity when varied over a wide range encompassing likely values on the lower-end for realistic estuarine-like mixing rates (Geyer et al. 2000, Geyer et al. 2008)."
11. line 193: rewrite "retrograde bed slope".
    - We have done this.
12. You refer to figure 5 before figure 4 so the two should be swapped.

- We have done this.

13. line 253: Could we say "vertical baroclinic convective motion"?
    - We appreciate this suggestion and have made this edit.
14. Eq. (9) line 325: this equation doesn't look like a parameterization but rather like the exact expression for the melt rate as a function of the conductive heat flux at the interface.
    - Based on both reviewers' suggestions, we have removed this section from the results.
15. Fig. 5-7: it was not clear to me whether the results you plotted were for the melt-enabled model or not.
    - In Figure 5, the dashed lines represent melt-enabled scenarios as denoted in the caption. To address confusion, we have edited line 248:

      > "When melt is enabled the horizontal extent of stratification in the subglacial environment is reduced, but where stratification occurs, it is stronger (e.g. dashed lines in Figure 4)."

    - For Figure 6, all data comes from melt-enabled cases, as denoted in the caption. We have added to line 341 to improve clarity:

      > "Here, we tested the sensitivity of equations 6-9 to various choices of ice distance to obtain Tw, Sw, ρw, and u for the melt-enabled cases."

    - For Figure 7, both melt-enabled and no-melting cases are represented as denoted in the legend. We have edited the figure caption to improve clarity:

      > "Wilson et al. (2020) experimental data (gray markers) and intrusion characteristics found in this study (red and blue markers). The red markers represent simulations with melting enabled, and blue markers represent non-melting simulations. The black dashed line is the numerical solution to Robel et al. (2022) with γ= 2."

      In addition, we have added to lines 387-389:

      > "Our simulated intrusions for both non-melting and melt-enabled scenarios follow the general trend and scale sensitivity to those identified in previous laboratory experiments (Figure. 7) (Wilson et al. 2020) which are within a factor of 10 to the theoretical prediction (dashed line) from Robel et al. (2022)."

16. Eq. (A3): should it be Re_L? Or change L into x?

- We appreciate you pointing out this inconsistency. We have amended the equation to be:

$$Re_L = \frac{uL\rho_w}{\mu} \quad \text{(R14)}$$

17. Appendix A4: I have found many typos, you need to use capital letters for Reynolds number, kappa has become k etc Eq (A7) has signs/symbols displaced. Please discuss Eq (A7) more carefully. What is tau_eff? What is species diffusion in your case?
    - In combination with previous comments (e.g. 3, 4,5, 6, 7, and major comments 1 and 5) we have revised the appendix to encapsulate every part of the model and more thoroughly justify model choices. In this rewrite, we standardized the symbols used and refined appendix equations.
    - T_{eff} is the effective shear stress, which includes viscous shearing effects as well as shear from the no-slip boundary conditions.
    - Species Diffusion in our case represents salt diffusion. Since we are transporting salt as an active tracer throughout the fluid, we must include its heat transport in the energy equation.
    - We have edited lines 569-571:

> "The first three terms on the right-hand side represent energy transfer due to the conduction of heat ($\nabla\ (k_{eff}\ \nabla T)$), species diffusion ($\nabla\ (-\sum_j h_j\ J_j)$), and molecular dissipation ($\nabla\ (\tau_{eff}\ v)$), respectively. Conduction represents heat transfer due to thermal gradients, and viscous dissipation is the transformation of kinetic energy into thermal energy due to shear forces from viscous effects and wall boundary effects (τ_{eff}). As salt diffuses in the medium, it also transfers heat due to its unique thermal properties, and therefore must also be included."

Works Cited

Pope, S.B. (2000). Turbulent Flows. *Cambridge University Press*, Cambridge, 305-308.

Geyer, W.R., Trowbridge, J. H., & Bowen, M.M. (2000) The Dynamics of a Partially Mixed Estuary. *J. Phys. Oceanogr.* **30**, Iss 8, 2035-2048. https://doi.org/10.1175/1520-0485(2000)030<2035:TDOAPM>2.0.CO;2

Geyer, W.R., Scully, M.E. & Ralston, D.K. (2008). Quantifying vertical mixing in estuaries. *Environ Fluid Mech* **8**, 495–509. https://doi.org/10.1007/s10652-008-9107-2

Roquet, F., Madex, G., Brodeau, L., Nycander, J. (2015). Defining a Simplified Yet "Realistic" Equation of State for Seawater. *J. Phys. Oceanogr.* **45**, Iss 10, 2564-2579. https://doi.org/10.1175/JPO-D-15-0080.1

Chalá, D.C.; Castro-Faccetti, C.; Quiñones-Bolaños, E.; Mehrvar, M. Salinity Intrusion Modeling Using Boundary Conditions on a Laboratory Setup: Experimental Analysis and CFD Simulations. *Water* **2024**, *16*, 1970. https://doi.org/10.3390/w16141970

Zangiabadi, E.; Edmunds, M.; Fairley, I.A.; Togneri, M.; Williams, A.J.; Masters, I.; Croft, N. Computational Fluid Dynamics and Visualisation of Coastal Flows in Tidal Channels Supporting Ocean Energy Development. *Energies* **2015**, *8*, 5997-6012. https://doi.org/10.3390/en8065997

Chan, S.N., Lai, A.C.H., Law, A.W.K., Eric Adams, E. (2020). Two-Phase CFD Modeling of Sediment Plumes for Dredge Disposal in Stagnant Water. In: Nguyen, K., Guillou, S., Gourbesville, P., Thiébot, J. (eds) Estuaries and Coastal Zones in Times of Global Change. Springer Water. Springer, Singapore. https://doi.org/10.1007/978-981-15-2081-5_24

Rana A. Al-Zubaidy, Ali N. Hilo, (2022). Numerical investigation of flow behavior at the lateral intake using Computational Fluid Dynamics (CFD). Materials Today: Proceedings. Volume 56, Part 4. Pages 1914-1926. ISSN 2214-7853. https://doi.org/10.1016/j.matpr.2021.11.172.

Sultan, R. A., Rahman, M. A., Rushd, S., Zendehboudi, S., & Kelessidis, V. C. (2018). Validation of CFD model of multiphase flow through pipeline and annular geometries. *Particulate Science and Technology*, *37*(6), 685–697. https://doi.org/10.1080/02726351.2018.1435594

Jayakdoy, H., Al-Dadah, R., Mahmoud, S. (2017). Computational fluid dynamics investigation on indirect contact freeze desalination. Desalination. Volume 420, Pages 21-33. https://doi.org/10.1016/j.desal.2017.06.023