Responses to Referee 2 for the manuscript:
“Modelling the small-scale deposition of snow onto structured Arctic sea ice during a MOSAiC storm using snowBedFoam 1.0.”

March 11, 2022

1 General answer

The present document provides an answer to the second referee, who posted his/her comments on January 3, 2022 in the open discussion of our GMD manuscript. We thank the reviewer for his/her thoughtful comments and efforts towards improving our manuscript as well as for the expertise provided.

2 Major comments

For the sake of clarity, we chose to divide the major comments in sub-sections and named them after their content.

2.1 Wind velocity modelling at the boundary

“The finest vertical grid is 0.1m, while the saltation layer of blowing snow is almost the same order of magnitude. It means almost all the interactions between snow particles and airflow are in the first of the vertical grid, which indicates that the wind velocity estimation on particle location is important. Could the authors add more detail about the wind velocity approach where the snow particles located.”

This is, indeed, an important point. The wind velocity in the near-wall region is modelled through a wall function from the standard OpenFOAM library. Wall functions satisfy the physics near the wall and aim to bridge the inner region between the wall and the fully developed turbulent region. We made use of the so-called *epsilonLowReWallFunction* and *kLowReWallFunction* wall functions in OpenFOAM, which are adapted to both low and high Reynolds numbers and operate in two modes based on the cell size, viscosity and friction velocity ratio (y+). With this approach, the boundary layer does not need to be resolved, which significantly reduces the computational domain and mesh size. Wall functions have been commonly used in other snow transport models (e.g. Kang et al. 2018; V. Sharma, Comola, and M. Lehning 2018). More details regarding the OpenFOAM wall functions are available in Fangqing 2016.

In our modelling framework, any Eulerian quantity needed for the Lagrangian calculations (e.g. flow velocity) is interpolated based on the particle location (Smith and Ebert 2010; Tofighian, Amani, and Saffar-Avval 2019). We used the so-called "*cellPoint*" interpolation method in OpenFOAM, which works by dividing the cells into tetrahedrons connecting the cell center and the faces, then determining which tetrahedron encloses the parcel location, and finally performing linear interpolation using inverse distance weights (Leonard, Qiao, and Nabi 2021). We added more information about the interpolation method to the parcel location in Section 2.3.3. Numerics of the manuscript.

2.2 Eulerian and Lagrangian timesteps

"What are the time steps for airflow and particle, respectively? Since this model considers the splash process, the time step for the particle should be very limited to a small value."

For the flow, we make use of an automatic time step control called "*adjustableRunTime*" available in OpenFOAM, which adapts the time step based on a maximum Courant number value defined by the user (in our case, it was set to 1). The Courant number as the stability criterion is defined as

$$Co = \frac{\mathbf{U}^f \Delta t_f}{L} \tag{1}$$

where $\mathbf{U}^f$ is the fluid velocity, L is the numerical cell length scale defined as the ratio between the cell volume and the cell surface area, and Δt_f is the fluid-phase time step. This definition implies that the Eulerian time step changes with the grid size and local velocity. More information regarding the adjustable time step method for the flow is available in Moukalled, Mangani, and Darwish 2015 and Jafari, Varun Sharma, and Michael Lehning 2022.

There are various factors impacting the parcel (Lagrangian) time step. Within one Eulerian time step (Δt_f), there are several smaller Lagrangian time steps (Δt_p), which allow to adequately capture the parcel motion. In our Eulerian-Lagrangian model, the so-called "face-to-face tracking algorithm" (Peng 2008; Macpherson, Nordin, and Weller 2009) ensures that a parcel cannot cross a cell boundary without updating its properties accordingly. In other words, as soon as a parcel encounters a face including cell boundary faces, functions related to the face type are called within the solver. If a parcel reaches the surface (sea ice) boundary, the rebound-splash function is automatically called (Equations 12 to 14 in the manuscript) and the splashing process is fully resolved. Thus, the face-to-face algorithm adapts the Lagrangian time step depending on the crossed boundaries and automatically limits its value to appropriately capture micro-scale processes such as the rebound-splash of snow grains at the surface. We should mention that the motion of parcels is also controlled by a maximum Courant number specific for parcels, which is defined similarly to Equation 1 but with replacing the fluid velocity and timestep by the equivalent parcel quantities ($\mathbf{U}^p$ and Δt_p). In OpenFOAM, the maximum Courant number value for parcels is set by default to 0.3. Details on the computation of the Eulerian and Lagrangian time steps were added to Section 2.3.3. Numerics of the manuscript.

2.3 Simulation period

"The period of time is a week, which is really long for blowing snow evaluation."

We chose our one-week simulation period based on the available terrestrial laser scans measured during the MOSAiC expedition. To be able to compare the model results to snow distribution data obtained in the field, we had no choice but to simulate a full week of snow redistribution on sea ice.

To limit the computational effort to a reasonable amount, we selected the dominating wind phases within that week using a friction velocity threshold (0.2 m/s) and came up with the method of computing the erosion and deposition rates during the first seconds of simulation and then using the latter to estimate the total snow distribution after several hours. It is based on the assumption that a flow-particle equilibrium state (steady-state) is reached in the simulations, which we verified by analyzing the low fluctuations of the total particle mass in the system. Figure 1 shows an example of the total mass in the system for the first event described in the manuscript (November 6 2019, from 12:00 to 23:00). We are aware that this method has strong limitations but it seems like the most appropriate choice to simulate such a long period, given the modelling framework employed in our study. We added the following "Although long for the evaluation of blowing snow, we defined our simulation period based on the terrestrial laser scans measurements available during the MOSAiC expedition. To be able to compare the model results to snow distribution data obtained in the field, we had no choice but to simulate a full week of snow redistribution on sea ice." in Section 2.4. Modelling assumptions of the manuscript.

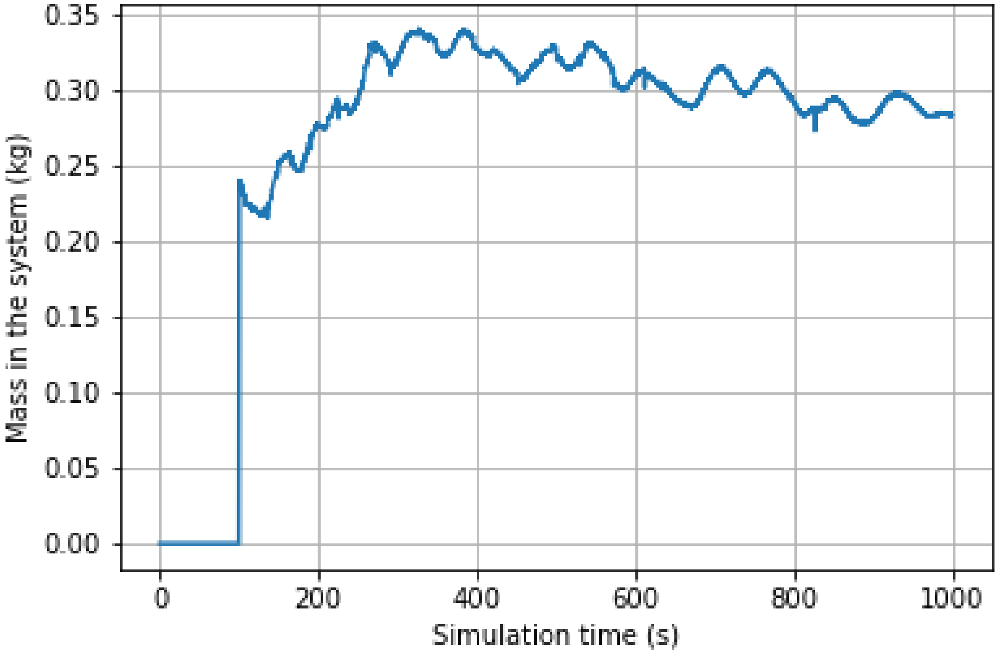

Figure 1: Total snow mass in the system (kg) over simulation time for event I.

2.4 Thermal processes

"The authors may also need to discuss the effects of other snow processes on snow distribution, such as the thermal processes."

We replied to the concerns of the first referee about the consideration of sublimation processes (blowing snow and surface sublimation) in our model. The detailed reply can be read in Section 2.1. Snow Sublimation of the document "*Responses to Referee 1 for the manuscript*" posted on the open discussion of the manuscript. We refer to this section which also provides an answer to the comments of the second referee. In summary, given the location and time of interest of our simulations, we consider the sublimation fluxes to be negligible. Additional comments were added to the revised version of the manuscript, such as: “Only pure mechanical fluid-particle interactions are considered here, thus we distinctively evaluate the impact of the horizontal snow transport on the sea ice snow mass balance at a given location. Thermal processes such as blowing snow and snow surface sublimation, although having a big role in the snow mass budget at certain spatio-temporal scales, are assumed to be negligible given the time period and location of interest (Chung et al. (2011); Webster et al. (2021)”. Furthermore, a detailed explanation of our hypothesis (similar to the lines above) is given in Section 2.4. Modelling assumptions of the revised manuscript. The topic is also further discussed in Section 4.

References

Peng, Fabian Kärrholm (2008). “Numerical Modelling of Diesel Spray Injection, Turbulence Interaction and Combustion”. In.

Macpherson, Graham B., Niklas Nordin, and Henry G. Weller (2009). “Particle tracking in unstructured, arbitrary polyhedral meshes for use in CFD and molecular dynamics”. In: *Communications in Numerical Methods in Engineering* 25.3, pp. 263–273. DOI: `https://doi.org/10.1002/cnm.1128`. eprint: `https://onlinelibrary.wiley.com/doi/pdf/10.1002/cnm.1128`. URL: `https://onlinelibrary.wiley.com/doi/abs/10.1002/cnm.1128`.

Smith, William G and Michael P Ebert (2010). *A method for unstructured mesh-to-mesh interpolation*. Tech. rep. Naval Surface Warfare Center Carderock Div Bethesda MD.

Moukalled, F., L. Mangani, and M. Darwish (2015). *The Finite Volume Method in Computational Fluid Dynamics: An Advanced Introduction with OpenFOAM and Matlab*. 1st. Springer Publishing Company, Incorporated. ISBN: 3319168738.

Fangqing, Liu (2016). “A Thorough Description Of How Wall Functions Are Implemented In OpenFOAM”. In: *Proceedings of CFD with OpenSource Software*. URL: `http://www.tfd.chalmers.se/~hani/kurser/OS_CFD_2016`.

Kang, Luyang et al. (2018). “CFD simulation of snow transport over flat, uniformly rough, open terrain: Impact of physical and computational parameters”. In: *Journal of Wind Engineering and Industrial Aerodynamics* 177, pp. 213–226. ISSN: 0167-6105. DOI: `https://doi.org/10.1016/j.jweia.2018.04.014`.

Sharma, V., F. Comola, and M. Lehning (2018). “On the suitability of the Thorpe–Mason model for calculating sublimation of saltating snow”. In: *The Cryosphere* 12.11, pp. 3499–3509. DOI: `10.5194/tc-12-3499-2018`. URL: `https://www.the-cryosphere.net/12/3499/2018/`.

Tofighian, H., E. Amani, and M. Saffar-Avval (2019). “Parcel-number-density control algorithms for the efficient simulation of particle-laden two-phase flows”. In: *Journal of Computational Physics* 387, pp. 569–588. ISSN: 0021-9991. DOI: `https://doi.org/10.1016/j.jcp.2019.02.052`. URL: `https://www.sciencedirect.com/science/article/pii/S002199911930169X`.

Leonard, Eric, Hongtao Qiao, and Saleh Nabi (2021). “A Comparison of Interpolation Methods in Fast Fluid Dynamics”. In: *International High Performance Buildings Conference* Paper 341. ISSN: 0167-6105. DOI: `https://docs.lib.purdue.edu/ihpbc/341`.

Jafari, Mahdi, Varun Sharma, and Michael Lehning (2022). “Convection of water vapour in snowpacks”. In: *Journal of Fluid Mechanics* 934, A38. DOI: `10.1017/jfm.2021.1146`.