Dear Editor:

We appreciate your immense patience in handling the review process of this MS. After going through the remaining comments from Reviewer 3 thoroughly we have prepared a revised MS, as well as our response on the following page.

As we pointed out in previously, in the current version of the MS, the width of Figure 10 has been reduced from 17 cm to 12 cm so that it will completely fit on the page in manuscript mode. Once accepted for publication, the width of Figure 10 should revert to 17 cm for standardized visualization.

Sincerely yours,

The Authors.

Response to Reviewer 3

> **Comment 1:** *I guess [out lines the program flow of the PALM vsrc module] If this is the case, please say it in the article.*

Response 1: The general architecture of the emission source module has been described in §2.1. The concept has been outlined in LL 112-118. The definition of the hash map and corresponding hash keys are defined in Eqs (2-3). The implementation-specific details can be found in §2.3.2. As an implementation specific issue, the discussion of the sorting and subsequent bisection (binary) search is mentioned in §2.3.2. In the Authors' view, the algorithm behind the present emission module has been described in the necessary detail, and its implementation sufficiently specified without losing generality. To reiterate these points, LL 98-119 in §2.1 of the MS have been revised.

> **Comment 2:** *However, this is *not* what is generally described as a hash table ... The data structure used by the authors could perhaps be described as a binary search tree.*

Response 2: The data structure the Authors employed for the emission module is a hash table. It contains the following components:

1) An (1D) array containing the data, in this case the ϵ_p^m terms defined in Eq (1), and
2) A hash function, as defined in Equation (2), which effectives maps the cell index locations to a unique hash key.

Details on handling and implementation of collision detection have been left out in the description as the hash keys in this case are unique to each cell point and thus unconditionally collision-free. The revisions made in Response 1 also emphasize the uniqueness of the hash keys.

While the Reviewer pointed out differences between the Authors' implementation of the hash table and what are generally described in textbooks, it is very common practice to store hash keys and their associative array indices in memory instead of calculating them on-demand for performance.

Further, in response to the Reviewer's claim on binary search trees, a binary (search) tree is a hierarchical data structure that an array used here is not. To expedite discussion, the rudiments of binary tree can be found in §2.3.1 of Kunth (1997) and Chapter 12 of Cormen et al (2022).

> **Comment 3:** *The addition "representing the number of emission sources." on line 124 does not seem to make sense here, as the N:s in (2) are grid sizes in i,j,k. N_kappa^m might be the number of emission sources.*

Response 3: The phrase "representing the number of emission sources" has been removed.

References:

Cormen et al (2022) "Introduction to Algorithms" 4 Ed MIT Press 312-330.
Kunth (1997) "The Art of Computer Programming" 3 Ed Addison Wesley Longman 308-330.