Response to Community Comment 1 (CC1)

Manuscript: egusphere-2023-2129
Title: A dynamic approach to three-dimensional radiative transfer in numerical weather prediction models: the dynamic TenStream solver v1.0
Authors: Richard Maier, Fabian Jakub, Claudia Emde, Mihail Manev, Aiko Voigt, and Bernhard Mayer

We thank Chiel van Heerwaarden and his group for their comments on our manuscript, which we will respond to below. To structure our response, Chiel's comments are printed on a gray background color, while our answers are displayed on ordinary white background.

I am writing this comment on behalf of our research group that works on understanding interactions between clouds, radiation, and the land surface. One of our main research topics is developing and using large-eddy simulations with coupled 3D radiation, and for that reason, we studied this paper together with great interest. Let me start by congratulating the authors with their paper. The Munich group has pioneered the coupling of large-eddy simulations with 3D radiation with their TenStream solver, and this method is a very interesting further development of the method. Based on our group discussion, we would like to share two suggestions that could help in improving the paper.

Suggestion 1: comparison to alternatives to n-stream methods

It would be nice if the authors could extend their introduction by adding some discussion on alternative methods to the TenStream solver. We believe that in recent years, there has been significant progress in ray tracing of large-eddy simulation fields of cloudy boundary layers, with the papers of Najda Villefranque and colleagues (JAMES, 2019) and Jake Gristey and colleagues (JAS, 2020, GRL 2020) as prominent examples. Also, in our group, we developed a GPU ray tracer, which we coupled to our large-eddy simulation code to study the evolution of shallow cumulus clouds (Veerman et al., 2022, GRL) inspired on earlier work by the Munich group. Then, the recent work of Du an Stechmann (JCP, 2023) on spectral element modeling looks rather promising as well, although coupling with cloud-resolving models remains future work there. To conclude, a more elaborate comparison of n-stream solvers to ray tracing and spectral elements methods could help the reader understand why the authors believe their method is the way to bring 3D radiation to operational weather prediction models.

Thank you for this suggestion. In the preprint version, we kept the introduction rather short. But you are certainly right that we should probably summarize the current state of research on 3D radiative transfer more thoroughly. Therefore, we thankfully used the papers you provided to extend our introduction in that regard.

Suggestion 2: discussion on memory usage of the solver

The authors present a very extensive performance analysis of their method, which shows that they can deliver an excellent speed up with respect to the original TenStream solver. This in itself is a great result, and the way this is achieved – keeping the fluxes in memory – is clever, because it removes the need for global communication and for a linear system solver. The description omits, however, a discussion on the most impactful consequence of keeping fluxes in memory, namely memory usage. We did a back of the envelope calculation: if every flux (10 diffuse, 3 direct, 10 thermal) for every quadrature point needs to be kept in memory, and one uses a set of 54 (SW) and 67 (LW) quadrature points, then the dynamic TenStream solver requires (10+3) * 54 + 10 * 67 = 1372 permanent three-dimensional fields for the solver. While the authors discuss in the final sections the benefit of smaller quadrature-point sets, the exact memory footprint of the dynamic solver with respect to the original TenStream is not discussed. We believe this number is very relevant if the ultimate aim is to include this solver in an operational weather model.

You are absolutely right that we have to save these 1372 three-dimensional flux fields in order for the dynamic TenStream solver to work. However, we do currently not keep these fluxes in memory all the time, but dump them to the hard drive after calculating a spectral band, so that we only have one three-dimensional field of fluxes in the memory at the same time.

Memory usage is thus currently not dominated by storing these 3D fields in memory, but rather by the look-up tables, which we keep in memory all the time in order to be able to quickly access the TenStream coefficients when performing the Gauß-Seidel iterations.

Nevertheless, introducing a time-stepping scheme in contrast to calculating radiation from scratch will always be more memory consuming on the downside.

As we already mentioned in the response to Review Comment 1, a thorough analysis of the computational demands of our new solver however was never within the scope of this paper. The only point that we wanted to make in that regard is that by design incomplete solves lead to a noticeably increase in computational speed. The numbers in section 3.1 were just supposed to give a rough estimation of how fast the new solver is. A detailed investigation of computational speed and memory usage would require a much more thorough analysis of the computational aspects of the solver, whereas the paper is mainly concerned with demonstrating the feasibility of our new method. That is why we decided not to include more detailed computational aspects into the paper.